Walter Gautschi

Numerical Analysis

Second Edition

Birkhäuser

Birkhäuser

Walter Gautschi

Numerical Analysis

Second Edition

Birkhäuser

Walter Gautschi
Department of Computer Sciences
Purdue University
250 N. University Street
West Lafayette, IN 47907-2066
wgautschi@purdue.edu

ISBN 978-0-8176-8258-3 e-ISBN 978-0-8176-8259-0
DOI 10.1007/978-0-8176-8259-0
Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2011941359

Mathematics Subject Classification (2010): 65-01, 65D05, 65D07, 65D10, 65D25, 65D30, 65D32, 65H04, 65H05, 65H10, 65L04, 65L05, 65L06, 65L10

Printed on acid-free paper

www.birkhauser-science.com

To

Erika

Preface to the Second Edition

In this second edition, the outline of chapters and sections has been preserved. The subtitle "An Introduction", as suggested by several reviewers, has been deleted. The content, however, is brought up to date, both in the text and in the notes. Many passages in the text have been either corrected or improved. Some biographical notes have been added as well as a few exercises and computer assignments. The typographical appearance has also been improved by printing vectors and matrices consistently in boldface types.

With regard to computer language in illustrations and exercises, we now adopt uniformly Matlab. For readers not familiar with Matlab, there are a number of introductory texts available, some, like Moler [2004], Otto and Denier [2005], Stanoyevitch [2005] that combine Matlab with numerical computing, others, like Knight [2000], Higham and Higham [2005], Hunt, Lipsman and Rosenberg [2006], and Driscoll [2009], more exclusively focused on Matlab.

The major novelty, however, is a complete set of detailed solutions to all exercises and machine assignments. The solution manual is available to instructors upon request at the publisher's website http://www.birkhauser-science.com/978-0-8176-8258-3. Selected solutions are also included in the text to give students an idea of what is expected. The bibliography has been expanded to reflect technical advances in the field and to include references to new books and expository accounts. As a result, the text has undergone an expansion in size of about 20%.

West Lafayette, Indiana
November 2011

Walter Gautschi

Preface to the First Edition

The book is designed for use in a graduate program in Numerical Analysis that is structured so as to include a basic introductory course and subsequent more specialized courses. The latter are envisaged to cover such topics as numerical linear algebra, the numerical solution of ordinary and partial differential equations, and perhaps additional topics related to complex analysis, to multidimensional analysis, in particular optimization, and to functional analysis and related functional equations. Viewed in this context, the first four chapters of our book could serve as a text for the basic introductory course, and the remaining three chapters (which indeed are at a distinctly higher level) could provide a text for an advanced course on the numerical solution of ordinary differential equations. In a sense, therefore, the book breaks with tradition in that it does no longer attempt to deal with all major topics of numerical mathematics. It is felt by the author that some of the current subdisciplines, particularly those dealing with linear algebra and partial differential equations, have developed into major fields of study that have attained a degree of autonomy and identity that justifies their treatment in separate books and separate courses on the graduate level. The term "Numerical Analysis" as used in this book, therefore, is to be taken in the narrow sense of the numerical analogue of Mathematical Analysis, comprising such topics as machine arithmetic, the approximation of functions, approximate differentiation and integration, and the approximate solution of nonlinear equations and of ordinary differential equations.

What is being covered, on the other hand, is done so with a view toward stressing basic principles and maintaining simplicity and student-friendliness as far as possible. In this sense, the book is "An Introduction". Topics that, even though important and of current interest, require a level of technicality that transcends the bounds of simplicity striven for, are referenced in detailed bibliographic notes at the end of each chapter. It is hoped, in this way, to place the material treated in proper context and to help, indeed encourage, the reader to pursue advanced modern topics in more depth.

A significant feature of the book is the large collection of exercises that are designed to help the student develop problem-solving skills and to provide interesting extensions of topics treated in the text. Particular attention is given to machine assignments, where the student is encouraged to implement numerical techniques on the computer and to make use of modern software packages.

The author has taught the basic introductory course and the advanced course on ordinary differential equations regularly at Purdue University for the last 30 years or so. The former, typically, was offered both in the fall and spring semesters, to a mixed audience consisting of graduate (and some good undergraduate) students in mathematics, computer science, and engineering, while the latter was taught only in the fall, to a smaller but also mixed audience. Written notes began to materialize in the 1970s, when the author taught the basic course repeatedly in summer courses on Mathematics held in Perugia, Italy. Indeed, for some time, these notes existed only in the Italian language. Over the years, they were progressively expanded, updated, and transposed into English, and along with that, notes for the advanced course were developed. This, briefly, is how the present book evolved.

A long gestation period such as this, of course, is not without dangers, the most notable one being a tendency for the material to become dated. The author tried to counteract this by constantly updating and revising the notes, adding newer developments when deemed appropriate. There are, however, benefits as well: over time, one develops a sense for what is likely to stand the test of time and what may only be of temporary interest, and one selects and deletes accordingly. Another benefit is the steady accumulation of exercises and the opportunity to have them tested on a large and diverse student population.

The purpose of academic teaching, in the author's view, is twofold: to transmit knowledge, and, perhaps more important, to kindle interest and even enthusiasm in the student. Accordingly, the author did not strive for comprehensiveness – even within the boundaries delineated – but rather tried to concentrate on what is essential, interesting and intellectually pleasing, and teachable. In line with this, an attempt has been made to keep the text uncluttered with numerical examples and other illustrative material. Being well aware, however, that mastery of a subject does not come from studying alone but from active participation, the author provided many exercises, including machine projects. Attributions of results to specific authors and citations to the literature have been deliberately omitted from the body of the text. Each chapter, as already mentioned, has a set of appended notes that help the reader to pursue related topics in more depth and to consult the specialized literature. It is here where attributions and historical remarks are made, and where citations to the literature – both textbook and research – appear.

The main text is preceded by a prologue, which is intended to place the book in proper perspective. In addition to other textbooks on the subject, and information on software, it gives a detailed list of topics not treated in this book, but definitely belonging to the vast area of computational mathematics, and it provides ample references to relevant texts. A list of numerical analysis journals is also included.

The reader is expected to have a good background in calculus and advanced calculus. Some passages of the text require a modest degree of acquaintance with linear algebra, complex analysis, or differential equations. These passages, however, can easily be skipped, without loss of continuity, by a student who is not familiar with these subjects.

It is a pleasure to thank the publisher for showing interest in this book and cooperating in producing it. The author is also grateful to Soren Jensen and Manil Suri, who taught from this text, and to an anonymous reader; they all made many helpful suggestions on improving the presentation. He is particularly indebted to Prof. Jensen for substantially helping in preparing the exercises to Chap. 7. The author further acknowledges assistance from Carl de Boor in preparing the notes to Chap. 2 and to Werner C. Rheinboldt for helping with the notes to Chap. 4. Last but not least, he owes a measure of gratitude to Connie Wilson for typing a preliminary version of the text and to Adam Hammer for assisting the author with the more intricate aspects of LaTeX.

West Lafayette, Indiana Walter Gautschi
January 1997

Contents

Prologue

P1 Overview

Numerical Analysis is the branch of mathematics that provides tools and methods for solving mathematical problems in numerical form. The objective is to develop detailed computational procedures, capable of being implemented on electronic computers, and to study their performance characteristics. Related fields are *Scientific Computation*, which explores the application of numerical techniques and computer architectures to concrete problems arising in the sciences and engineering; *Complexity Theory*, which analyzes the number of "operations" and the amount of computer memory required to solve a problem; and *Parallel Computation*, which is concerned with organizing computational procedures in a manner that allows running various parts of the procedures simultaneously on different processors.

The problems dealt with in computational mathematics come from virtually all branches of pure and applied mathematics. There are computational aspects in number theory, combinatorics, abstract algebra, linear algebra, approximation theory, geometry, statistics, optimization, complex analysis, nonlinear equations, differential and other functional equations, and so on. It is clearly impossible to deal with all these topics in a single text of reasonable size. Indeed, the tendency today is to develop specialized texts dealing with one or the other of these topics. In the present text we concentrate on subject matters that are basic to problems in approximation theory, nonlinear equations, and differential equations. Accordingly, we have chapters on machine arithmetic, approximation and interpolation, numerical differentiation and integration, nonlinear equations, one-step and multistep methods for ordinary differential equations, and boundary value problems in ordinary differential equations. Important topics not covered in this text are computational number theory, algebra, and geometry; constructive methods in optimization and complex analysis; numerical linear algebra; and the numerical solution of problems involving partial differential equations and integral equations. Selected texts for these areas are enumerated in Sect. P3.

We now describe briefly the topics treated in this text. Chapter 1 deals with the basic facts of life regarding machine computation. It recognizes that, although present-day computers are extremely powerful in terms of computational speed, reliability, and amount of memory available, they are less than ideal – unless supplemented by appropriate software – when it comes to the precision available, and accuracy attainable, in the execution of elementary arithmetic operations. This raises serious questions as to how arithmetic errors, either present in the input data of a problem or committed during the execution of a solution algorithm, affect the accuracy of the desired results. Concepts and tools required to answer such questions are put forward in this introductory chapter. In Chap. 2, the central theme is the approximation of functions by simpler functions, typically polynomials and piecewise polynomial functions. Approximation in the sense of least squares provides an opportunity to introduce orthogonal polynomials, which are relevant also in connection with problems of numerical integration treated in Chap. 3. A large part of the chapter, however, deals with polynomial interpolation and associated error estimates, which are basic to many numerical procedures for integrating functions and differential equations. Also discussed briefly is inverse interpolation, an idea useful in solving equations.

First applications of interpolation theory are given in Chap. 3, where the tasks presented are the computation of derivatives and definite integrals. Although the formulae developed for derivatives are subject to the detrimental effects of machine arithmetic, they are useful, nevertheless, for purposes of discretizing differential operators. The treatment of numerical integration includes routine procedures, such as the trapezoidal and Simpson's rules, appropriate for well-behaved integrands, as well as the more sophisticated procedures based on Gaussian quadrature to deal with singularities. It is here where orthogonal polynomials reappear. The method of undetermined coefficients is another technique for developing integration formulae. It is applied to approximate general linear functionals, the Peano representation of linear functionals providing an important tool for estimating the error. The chapter ends with a discussion of extrapolation techniques; although applicable to more general problems, they are inserted here since the composite trapezoidal rule together with the Euler–Maclaurin formula provides the best-known application – Romberg integration.

Chapter 4 deals with iterative methods for solving nonlinear equations and systems thereof, the *pièce de résistance* being Newton's method. The emphasis here lies in the study of, and the tools necessary to analyze, convergence. The special case of algebraic equations is also briefly given attention.

Chapter 5 is the first of three chapters devoted to the numerical solution of ordinary differential equations. It concerns itself with one-step methods for solving initial value problems, such as the Runge–Kutta method, and gives a detailed analysis of local and global errors. Also included is a brief introduction to stiff equations and special methods to deal with them. Multistep methods and, in particular, Dahlquist's theory of stability and its applications, is the subject of Chap. 6. The final chapter (Chap. 7) is devoted to boundary value problems and their solution by shooting methods, finite difference techniques, and variational methods.

P2 Numerical Analysis Software

There are many software packages available, both in the public domain and distributed commercially, that deal with numerical analysis algorithms. A widely used source of numerical software is Netlib, accessible at `http://www.netlib.org`.

Large collections of general-purpose numerical algorithms are contained in sources such as Slatec (`http://www.netlib.org/slatec`) and TOMS (ACM Transactions on Mathematical Software). Specialized packages relevant to the topics in the chapters ahead are identified in the "Notes" to each chapter. Likewise, specific files needed to do some of the machine assignments in the Exercises are identified as part of the exercise.

Among the commercial software packages we mention the Visual Numerics (formerly IMSL) and NAG libraries. Interactive systems include HiQ, Macsyma, Maple, Mathcad, Mathematica, and Matlab. Many of these packages, in addition to numerical computation, have symbolic computation and graphics capabilities. Further information is available in the Netlib file `commercial`. For more libraries, and for interactive systems, also see Lozier and Olver [1994, Sect. 3].

In this text we consistently use Matlab as a vehicle for describing algorithms and as the software tool for carrying out some of the exercises and all machine assignments.

P3 Textbooks and Monographs

We provide here an annotated list (ordered alphabetically with respect to authors) of other textbooks on numerical analysis, written at about the same, or higher, level as the present one. Following this, we also mention books and monographs dealing with topics in computational mathematics not covered in our (and many other) books on numerical analysis. Additional books dealing with specialized subject areas, as well as other literature, are referenced in the "Notes" to the individual chapters. We generally restrict ourselves to books written in English and, with a few exceptions, published within the last 25 years or so. Even so, we have had to be selective. (No value judgment is to be implied by our selections or omissions.) A reader with access to the AMS (American Mathematical Society) MathSci Net homepage will have no difficulty in retrieving a more complete list of relevant items, including older texts.

P3.1 Selected Textbooks on Numerical Analysis

Atkinson [1989] A comprehensive in-depth treatment of standard topics short of partial differential equations; includes an appendix describing some of the better-known software packages.

Atkinson and Han [2009] An advanced text on theoretical (as opposed to computational) aspects of numerical analysis, making extensive use of functional analysis.

Bruce, Giblin, and Rippon [1990] A collection of interesting mathematical problems, ranging from number theory and computer-aided design to differential equations, that require the use of computers for their solution.

Cheney and Kincaid [1994] Although an undergraduate text, it covers a broad area, has many examples from science and engineering as well as computer programs; there are many exercises, including machine assignments.

Conte and de Boor [1980] A widely used text for upper-division undergraduate students; written for a broad audience, with algorithmic concerns in the foreground; has Fortran subroutines for many algorithms discussed in the text.

Dahlquist and Björck [2003, 2008] The first (2003) text – a reprint of the 1974 classic – provides a comprehensive introduction to all major fields of numerical analysis, striking a good balance between theoretical issues and more practical ones. The second text expands substantially on the more elementary topics treated in the first and represents the first volume of more to come.

Deuflhard and Hohmann [2003] An introductory text with emphasis on machine computation and algorithms; includes discussions of three-term recurrence relations and stochastic eigenvalue problems (not usually found in textbooks), but no differential equations.

Fröberg [1985] A thorough and exceptionally lucid exposition of all major topics of numerical analysis exclusive of algorithms and computer programs.

Hämmerlin and Hoffmann [1991] Similar to Stoer and Bulirsch [2002] in its emphasis on mathematical theory; has more on approximation theory and multivariate interpolation and integration, but nothing on differential equations.

Householder [2006] A reissue of one of the early mathematical texts on the subject, with coverage limited to systems of linear and nonlinear equations and topics in approximation.

Isaacson and Keller [1994] One of the older but still eminently readable texts, stressing the mathematical analysis of numerical methods.

Kincaid and Cheney [1996] Related to Cheney and Kincaid [1994] but more mathematically oriented and unusually rich in exercises and bibliographic items.

Kress [1998] A rather comprehensive text with a strong functional analysis component.

Neumaier [2001] A text emphasizing robust computation, including interval arithmetic.

Rutishauser [1990] An annotated translation from the German of an older text based on posthumous notes by one of the pioneers of numerical analysis; although the subject matter reflects the state of the art in the early 1970s, the treatment is highly original and is supplemented by translator's notes to each chapter pointing to more recent developments.

Schwarz [1989] A mathematically oriented treatment of all major areas of numerical analysis, including ordinary and partial differential equations.

Stoer and Bulirsch [2002] Fairly comprehensive in coverage; written in a style appealing more to mathematicians than engineers and computer scientists; has many exercises and bibliographic references; serves not only as a textbook but also as a reference work.

Todd [1979, 1977] Rather unique books, emphasizing problem-solving in areas often not covered in other books on numerical analysis.

P3.2 Monographs and Books on Specialized Topics

A collection of outstanding survey papers on specialized topics in numerical analysis is being assembled by Ciarlet and Lions [1990–2003] in handbooks of numerical analysis; nine volumes have appeared so far. Another source of surveys on a variety of topics is *Acta numerica*, an annual series of books edited by Iserles [1992–2010], of which 19 volumes have been published so far. For an authoritative account of the history of numerical analysis from the 16th through the 19th century, the reader is referred to the book by Goldstine [1977]. For more recent history, see Bultheel and Cools, eds. [2010].

The related areas of *Scientific Computing* and *Parallel Computing* are rather more recent fields of study. Basic introductory texts are Scott et al. [2005] and Tveito and Winter [2009]. Texts relevant to linear algebra and differential equations are Schendel [1984], Ortega and Voigt [1985], Ortega [1989], Golub and Ortega [1992], [1993], Van de Velde [1994], Burrage [1995], Heath [1997], Deuflhard and Bornemann [2002], O'Leary [2009], and Quarteroni et al. [2010]. Other texts address topics in optimization, Pardalos et al. [1992] and Gonnet and Scholl [2009]; computational geometry, Akl and Lyons [1993]; and other miscellaneous areas, Crandall [1994], [1996], Köckler [1994], Bellomo and Preziosi [1995], Danaila et al. [2007], and Farin and Hansford [2008]. Interesting historical essays are contained in Nash, ed. [1990]. Matters regarding the *Complexity* of numerical algorithms are discussed in an abstract framework in books by Traub and Woźniakowski [1980] and Traub, Wasilkowski, and Woźniakowski [1983], [1988], with applications to the numerical integration of functions and nonlinear equations, and similarly, applied to elliptic partial differential equations and integral equations, in the book by Werschulz [1991]. Other treatments are those by Kronsjö [1987], Ko [1991], Bini and Pan [1994], Wang et al. [1994], Traub and Werschulz [1998], Ritter [2000], and Novak et al. [2009]. For an in-depth complexity analysis of Newton's method, the reader is encouraged to study Smale's [1987] lecture.

Material on *Computational Number Theory* can be found, at the undergraduate level, in the book by Rosen [2000], which also contains applications to cryptography and computer science, and in Allenby and Redfern [1989], and at a more advanced level in the books by Niven et al. [1991], Cohen [1993], and Bach and Shallit [1996]. Computational methods of factorization are dealt with in the book by Riesel [1994]. Other useful sources are the set of lecture notes by Pohst [1993] on algebraic number theory algorithms, and the proceedings volumes edited by

Pomerance [1990] and Gautschi [1994a, Part II]. For algorithms in *Combinatorics*, see the books by Nijenhuis and Wilf [1978], Hu and Shing [2002], and Cormen et al. [2009]. Various aspects of *Computer Algebra* are treated in the books by Geddes et al. [1992], Mignotte [1992], Davenport et al. [1993], Mishra [1993], Heck [2003], and Cox et al. [2007].

Other relatively new disciplines are *Computational Geometry* and *Geometric Modeling*, *Computer-Aided Design*, and *Computational Topology*, for which relevant texts are, respectively, Preparata and Shamos [1985], Edelsbrunner [1987], Mäntylä [1988], Taylor [1992], McLeod and Baart [1998], Gallier [2000], Cohen et al. [2001], and Salomon [2006]; Hoschek and Lasser [1993], Farin [1997], [1999], and Prautsch et al. [2002]; Edelsbrunner [2006], and Edelsbrunner and Harer [2010]. *Statistical Computing* is covered in general textbooks such as Kennedy and Gentle [1980], Anscombe [1981], Maindonald [1984], Thisted [1988], Monahan [2001], Gentle [2009], and Lange [2010]. More specialized texts are Devroye [1986] and Hörmann et al. [2004] on the generation of nonuniform random variables, Späth [1992] on regression analysis, Heiberger [1989] on the design of experiments, Stewart [1994] on Markov chains, Xiu [2010] on stochastic computing and uncertainty quantification, and Fang and Wang [1994], Manno [1999], Gentle [2003], Liu [2008], Shonkwiler and Mendivil [2009], and Lemieux [2009] on Monte Carlo and number-theoretic methods. Numerical techniques in *Optimization* (including optimal control problems) are discussed in Evtushenko [1985]. An introductory book on unconstrained optimization is Wolfe [1978]; among more advanced and broader texts on optimization techniques we mention Gill et al. [1981], Ciarlet [1989], and Fletcher [2001]. Linear programming is treated in Nazareth [1987] and Panik [1996], linear and quadratic problems in Sima [1996], and the application of conjugate direction methods to problems in optimization in Hestenes [1980]. The most comprehensive text on (numerical and applied) *Complex Analysis* is the three-volume treatise by Henrici [1988, 1991, 1986]. Numerical methods for conformal mapping are also treated in Kythe [1998], Schinzinger and Laura [2003], and Papamichael and Stylianopoulos [2010]. For approximation in the complex domain, the standard text is Gaier [1987]; Stenger [1993] deals with approximation by sinc functions, Stenger [2011] providing some 450 Matlab programs. The book by Iserles and Nørsett [1991] contains interesting discussions on the interface between complex rational approximation and the stability theory of discretized differential equations. The impact of high-precision computation on problems and conjectures involving complex approximation is beautifully illustrated in the set of lectures by Varga [1990].

For an in-depth treatment of many of the preceding topics, also see the four-volume work of Knuth [1975, 1981, 1973, 2005–2006].

Perhaps the most significant topic omitted in our book is numerical linear algebra and its application to solving partial differential equations by finite difference or finite element methods. Fortunately, there are many treatises available that address these areas. For *Numerical Linear Algebra*, we refer to the classic work of Wilkinson [1988] and the book by Golub and Van Loan [1996]. Links and applications of matrix computation to orthogonal polynomials and quadrature are the subject of Golub and Meurant [2010]. Other general texts are Jennings and McKeown [1992], Watkins [2002], [2007], Demmel [1997], Trefethen and Bau [1997], Stewart [1973], [1998], Meurant [1999], White [2007], Allaire and Kaber [2008], and Datta [2010]; Higham [2002], [2008] has a comprehensive treatment of error and stability analyses and the first, equally extensive, treatment of the numerics of matrix functions. Solving linear systems on vector and shared memory parallel computers and the use of linear algebra packages on high-performance computers are discussed in Dongarra et al. [1991], [1998]. The solution of sparse linear systems and the special data structures and pivoting strategies required in direct methods are treated in Østerby and Zlatev [1983], Duff et al. [1989], Zlatev [1991], and Davis [2006], whereas iterative techniques are discussed in the classic texts by Young [2003] and Varga [2000], and in Il'in [1992], Hackbusch [1994], Weiss [1996], Fischer [1996], Brezinski [1997], Greenbaum [1997], Saad [2003], Broyden and Vespucci [2004], Hageman and Young [2004], Meurant [2006], Chan and Jin [2007], Byrne [2008], and Woźnicki [2009]. The books by Branham [1990] and Björck [1996] are devoted especially to least squares problems. For eigenvalues, see Chatelin [1983], [1993], and for a good introduction to the numerical analysis of symmetric eigenvalue problems, see Parlett [1998]. The currently very active investigation of large sparse symmetric and nonsymmetric eigenvalue problems and their solution by Lanczos-type methods has given rise to many books, for example, Cullum and Willoughby [1985], [2002], Meyer [1987], Sehmi [1989], and Saad [1992]. For structured and symplectic eigenvalue problems, see Fassbender [2000] and Kressner [2005], and for inverse eigenvalue problems, Xu [1998] and Chu and Golub [2005]. For readers wishing to test their algorithms on specific matrices, the collection of test matrices in Gregory and Karney [1978] and the "matrix market" on the Web (`http://math.nist.gov./MatrixMarket`) are useful sources.

Even more extensive is the textbook literature on the numerical solution of *Partial Differential Equations*. The field has grown so much that there are currently only a few books that attempt to cover the subject more or less as a whole. Among these are Birkhoff and Lynch [1984] (for elliptic problems), Hall and Porsching [1990], Ames [1992], Celia and Gray [1992], Larsson and Thomée [2003], Quarteroni and Valli [1994], Morton and Mayers [2005], Sewell [2005], Quarteroni [2009], and Tveito and Winter [2009]. Variational and finite element methods seem to have attracted the most attention. An early and still frequently cited reference is the book by Ciarlet [2002] (a reprint of the 1978 original); among more recent texts we mention Beltzer [1990] (using symbolic computation), Křížek and Neittaanmäki [1990], Brezzi and Fortin [1991], Schwab [1998], Kwon and Bang [2000] (using Matlab), Zienkiewicz and Taylor [2000], Axelsson and Barker [2001], Babuška and Strouboulis [2001], Höllig [2003], Monk [2003] (for Maxwell's equation), Ern and Guermonde [2004], Kythe and Wei [2004], Reddy [2004], Chen [2005], Elman et al. [2005], Thomée [2006] (for parabolic equations), Braess [2007], Demkowicz [2007], Brenner and Scott [2008], Bochev and Gunzburger [2009], Efendiev and Hou [2009], and Johnson [2009]. Finite difference methods are treated in Ashyralyev and Sobolevskiĭ [1994], Gustafsson et al. [1995], Thomas [1995], [1999], Samarskii [2001], Strikwerda [2004], LeVeque [2007], and Gustafsson

[2008]; the method of lines in Schiesser [1991]; and the more refined techniques of multigrids and domain decomposition in McCormick [1989], [1992], Bramble [1993], Shaĭdurov [1995], Smith et al. [1996], Quarteroni and Valli [1999], Briggs et al. [2000], Toselli and Widlund [2005], and Mathew [2008]. Problems in potential theory and elasticity are often approached via boundary element methods, for which representative texts are Brebbia and Dominguez [1992], Chen and Zhou [1992], Hall [1994], and Steinbach [2008]. A discussion of conservation laws is given in the classic monograph by Lax [1973] and more recently in LeVeque [1992], Godlewski and Raviart [1996], Kröner [1997], and LeVeque [2002]. Spectral methods, i.e., expansions in (typically) orthogonal polynomials, applied to a variety of problems, were pioneered in the monograph by Gottlieb and Orszag [1977] and have received extensive treatments in more recent texts by Canuto et al. [1988], [2006], [2007], Fornberg [1996], Guo [1998], Trefethen [2000] (in Matlab), Boyd [2001], Peyret [2002], Hesthaven et al. [2007], and Kopriva [2009].

Early, but still relevant, texts on the numerical solution of *Integral Equations* are Atkinson [1976] and Baker [1977]. More recent treatises are Atkinson [1997] and Kythe and Puri [2002]. Volterra integral equations are dealt with by Brunner and van der Houwen [1986] and Brunner [2004], whereas singular integral equations are the subject of Prössdorf and Silbermann [1991].

P4 Journals

Here we list the major journals (in alphabetical order) covering the areas of numerical analysis and mathematical software.

ACM Transactions on Mathematical Software
Applied Numerical Mathematics
BIT Numerical Mathematics
Calcolo
Chinese Journal of Numerical Mathematics and Applications
Computational Mathematics and Mathematical Physics
Computing
IMA Journal on Numerical Analysis
Journal of Computational and Applied Mathematics
Mathematical Modelling and Numerical Analysis
Mathematics of Computation
Numerical Algorithms
Numerische Mathematik
SIAM Journal on Numerical Analysis
SIAM Journal on Scientific Computing

Chapter 1
Machine Arithmetic and Related Matters

The questions addressed in this first chapter are fundamental in the sense that they are relevant in any situation that involves numerical machine computation, regardless of the kind of problem that gave rise to these computations. In the first place, one has to be aware of the rather primitive type of number system available on computers. It is basically a finite system of numbers of finite length, thus a far cry from the idealistic number system familiar to us from mathematical analysis. The passage from a real number to a machine number entails *rounding*, and thus small errors, called *roundoff errors*. Additional errors are introduced when the individual arithmetic operations are carried out on the computer. In themselves, these errors are harmless, but acting in concert and propagating through a lengthy computation, they can have significant – even disastrous – effects.

Most problems involve input data not representable exactly on the computer. Therefore, even before the solution process starts, simply by storing the input in computer memory, the problem is already slightly perturbed, owing to the necessity of rounding the input. It is important, then, to estimate how such small perturbations in the input affect the output, the solution of the problem. This is the question of the (numerical) *condition of a problem*: the problem is called well conditioned if the changes in the solution of the problem are of the same order of magnitude as the perturbations in the input that caused those changes. If, on the other hand, they are much larger, the problem is called ill conditioned. It is desirable to measure by a single number – the *condition number* of the problem – the extent to which the solution is sensitive to perturbations in the input. The larger this number, the more ill conditioned the problem.

Once the solution process starts, additional rounding errors will be committed, which also contaminate the solution. The resulting errors, in contrast to those caused by input errors, depend on the particular solution algorithm. It makes sense, therefore, to also talk about the *condition of an algorithm*, although its analysis is usually quite a bit harder. The quality of the computed solution is then determined by both (essentially the product of) the condition of the problem and the condition of the algorithm.

W. Gautschi, *Numerical Analysis*, DOI 10.1007/978-0-8176-8259-0_1,

1.1 Real Numbers, Machine Numbers, and Rounding

We begin with the number system commonly used in mathematical analysis and confront it with the more primitive number system available to us on any particular computer. We identify the basic constant (the machine precision) that determines the level of precision attainable on such a computer.

1.1.1 Real Numbers

One can introduce real numbers in many different ways. Mathematicians favor the axiomatic approach, which leads them to define the set of real numbers as a "complete Archimedean ordered field." Here we adopt a more pedestrian attitude and consider the set of real numbers $\mathbb{R}$ to consist of positive and negative numbers represented in some appropriate number system and manipulated in the usual manner known from elementary arithmetic. We adopt here the *binary number system*, since it is the one most commonly used on computers. Thus,

$$x \in \mathbb{R} \;\text{ iff }\; x = \pm\,(b_n 2^n + b_{n-1}2^{n-1} + \cdots + b_0 + b_{-1}2^{-1} + b_{-2}2^{-2} + \cdots). \quad (1.1)$$

Here $n \geq 0$ is some integer, and the "binary digits" b_i are either 0 or 1,

$$b_i = 0 \;\text{ or }\; b_i = 1 \;\text{ for all }\; i. \quad (1.2)$$

It is important to note that in general we need infinitely many binary digits to represent a real number. We conveniently write such a number in the abbreviated form (familiar from the decimal number system)

$$x = \pm\,(b_n b_{n-1} \cdots b_0\, .\, b_{-1} b_{-2} b_{-3} \cdots)_2, \quad (1.3)$$

where the subscript 2 at the end is to remind us that we are dealing with a binary number. (Without this subscript, the number could also be read as a decimal number, which would be a source of ambiguity.) The dot in (1.3) – appropriately called the binary point – separates the integer part on the left from the fractional part on the right. Note that representation (1.3) is not unique, for example, $(0.011\overline{1}\ldots)_2 = (0.1)_2$. We regain uniqueness if we always insist on a finite representation, if one exists.

Examples. 1. $(10011.01)_2 = 2^4 + 2^1 + 2^0 + 2^{-2} = 16 + 2 + 1 + \frac{1}{4} = (19.25)_{10}$

2. $$(.0101\overline{01}\ldots)_2 = \sum_{\substack{k=2\\(k\ \text{even})}}^{\infty} 2^{-k} = \sum_{m=1}^{\infty} 2^{-2m} = \frac{1}{4}\sum_{m=0}^{\infty}\left(\frac{1}{4}\right)^m$$

$$= \frac{1}{4}\,\frac{1}{1-\frac{1}{4}} = \frac{1}{3} = (0.33\overline{3}\ldots)_{10}$$

3. $\frac{1}{5} = (0.2)_{10} = (0.0011\overline{0011}\ldots)_2$

   To determine the binary digits on the right, one keeps multiplying by 2 and observing the integer part in the result; if it is zero, the binary digit in question is 0, otherwise 1. In the latter case, the integral part is removed and the process repeated.

The last example is of interest insofar as it shows that to a finite decimal number there may correspond a (nontrivial) infinite binary representation. One cannot assume, therefore, that a finite decimal number is exactly representable on a binary computer. Conversely, however, to a finite binary number there always corresponds a finite decimal representation. (Why?)

1.1.2 Machine Numbers

There are two kinds of machine numbers: floating point and fixed point. The first corresponds to the "scientific notation" in the decimal system, whereby a number is written as a decimal fraction times an integral power of 10. The second allows only for fractions. On a binary computer, one consistently uses powers of 2 instead of 10. More important, the number of binary digits, both in the fraction and in the exponent of 2 (if any), is finite and cannot exceed certain limits that are characteristics of the particular computer at hand.

1.1.2.1 Floating-Point Numbers

We denote by t the number of binary digits allowed by the computer in the fractional part and by s the number of binary digits in the exponent. Then the set of (real) floating-point numbers on that computer will be denoted by $\mathbb{R}(t,s)$. Thus,

$$x \in \mathbb{R}(t,s) \;\text{ iff }\; x = f \cdot 2^e, \tag{1.4}$$

where, in the notation of (1.3),

$$f = \pm\,(.\,b_{-1}b_{-2}\cdots b_{-t})_2, \;\; e = \pm\,(c_{s-1}c_{s-2}\cdots c_0.)_2. \tag{1.5}$$

Here all b_i and c_j are binary digits, that is, either zero or one. The binary fraction f is usually referred to as the *mantissa* of x and the integer e as the *exponent* of x. The number x in (1.4) is said to be *normalized* if in its fraction f we have $b_{-1} = 1$. We assume that all numbers in $\mathbb{R}(t,s)$ are normalized (with the exception of $x = 0$, which is treated as a special number). If $x \neq 0$ were not normalized, we could

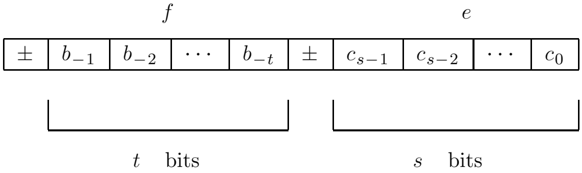


Fig. 1.1 Packing of a floating-point number in a machine register

multiply f by an appropriate power of 2, to normalize it, and adjust the exponent accordingly. This is always possible as long as the adjusted exponent is still in the admissible range.

We can think of a floating-point number (1.4) as being accommodated in a machine register as shown in Fig. 1.1. The figure does not quite correspond to reality, but is close enough to it for our purposes.

Note that the set (1.4) of normalized floating-point numbers is finite and is thus represented by a finite set of points on the real line. What is worse, these points are not uniformly distributed (cf. Ex. 1). This, then, is all we have to work with!

It is immediately clear from (1.4) and (1.5) that the largest and smallest magnitude of a (normalized) floating-point number is given, respectively, by

$$\max_{x\in\mathbb{R}(t,s)} |x| = (1-2^{-t})\,2^{2^s-1}, \quad \min_{x\in\mathbb{R}(t,s)} |x| = 2^{-2^s}. \tag{1.6}$$

On a Sun Sparc workstation, for example, one has $t = 23$, $s = 7$, so that the maximum and minimum in (1.6) are 1.70×10^{38} and 2.94×10^{-39}, respectively. (Because of an asymmetric internal hardware representation of the exponent on these computers, the true range of floating-point numbers is slightly shifted, more like from 1.18×10^{-38} to 3.40×10^{38}.) Matlab arithmetic, essentially double precision, uses $t = 53$ and $s = 10$, which greatly expands the number range from something like 10^{-308} to 10^{+308}.

A real nonzero number whose modulus is not in the range determined by (1.6) cannot be represented on this particular computer. If such a number is produced during the course of a computation, one says that *overflow* has occurred if its modulus is larger than the maximum in (1.6) and *underflow* if it is smaller than the minimum in (1.6). The occurrence of overflow is fatal, and the machine (or its operating system) usually prompts the computation to be interrupted. Underflow is less serious, and one may get away with replacing the delinquent number by zero. However, this is not foolproof. Imagine that at the next step the number that underflowed is to be multiplied by a huge number. If the replacement by zero has been made, the result will always be zero.

To increase the precision, one can use two machine registers to represent a machine number. In effect, one then embeds $\mathbb{R}(t,s) \subset \mathbb{R}(2t,s)$, and calls $x \in \mathbb{R}(2t,s)$ a *double-precision* number.

$\pm$	b_{-1}	b_{-2}	$\cdots$	b_{-t}	$b_{-(t+1)}$	$\cdots$	$b_{-(t+s)}$

Fig. 1.2 Packing of a fixed-point number in a machine register

1.1.2.2 Fixed-Point Numbers

This is the case (1.4) where $e = 0$. That is, fixed-point numbers are binary fractions, $x = f$, hence $|f| < 1$. We can therefore only deal with numbers that are in the interval (–1,1). This, in particular, requires extensive scaling and rescaling to make sure that all initial data, as well as all intermediate and final results, lie in that interval. Such a complication can only be justified in special circumstances where machine time and/or precision is at a premium. Note that on the same computer as considered before, we do not need to allocate space for the exponent in the machine register, and thus have in effect $s + t$ binary digits available for the fraction f, hence more precision; cf. Fig. 1.2.

1.1.2.3 Other Data Structures for Numbers

Complex floating-point numbers consist of pairs of real floating-point numbers, the first of the pair representing the real part and the second the imaginary part. To avoid rounding errors in arithmetic operations altogether, one can employ rational arithmetic, in which each (rational) number is represented by a pair of extended-precision integers – the numerator and denominator of the rational number. The Euclidean algorithm is used to remove common factors. A device that allows keeping track of error propagation and the influence of data errors is interval arithmetic involving intervals guaranteed to contain the desired numbers. In complex arithmetic, one employs rectangular or circular domains.

1.1.3 Rounding

A machine register acts much like the infamous Procrustes bed in Greek mythology. Procrustes was the innkeeper whose inn had only beds of one size. If a fellow came along who was too tall to fit into his beds, he cut off his feet. If the fellow was too short, he stretched him. In the same way, if a real number comes along that is too long, its tail end (not the head) is cutoff; if it is too short, it is padded by zeros at the end.

More specifically, let

$$x \in \mathbb{R}, \quad x = \pm\left(\sum_{k=1}^{\infty} b_{-k}2^{-k}\right)2^{e} \tag{1.7}$$

be the "exact" real number (in normalized floating-point form) and

$$x^* \in \mathbb{R}(t,s), \quad x^* = \pm\left(\sum_{k=1}^{t} b^*_{-k}2^{-k}\right)2^{\mathrm{e}^*} \tag{1.8}$$

the rounded number. One then distinguishes between two methods of rounding, the first being Procrustes' method.

(a) *Chopping*. One takes

$$x^* = \mathrm{chop}(x), \quad \mathrm{e}^* = e, \quad b^*_{-k} = b_{-k} \ \text{ for } \ k = 1, 2, \ldots, t. \tag{1.9}$$

(b) *Symmetric rounding*. This corresponds to the familiar rounding up or rounding down in decimal arithmetic, based on the first discarded decimal digit: if it is larger than or equal to 5, one rounds up; if it is less than 5, one rounds down. In binary arithmetic, the procedure is somewhat simpler, since there are only two possibilities: either the first discarded binary digit is 1, in which case one rounds up, or it is 0, in which case one rounds down. We can write the procedure very simply in terms of the chop operation in (1.9):

$$x^* = \mathrm{rd}(x), \quad \mathrm{rd}(x) := \mathrm{chop}\left(x + \frac{1}{2}\cdot 2^{-t}\cdot 2^{e}\right). \tag{1.10}$$

There is a small error incurred in rounding, which is most easily estimated in the case of chopping. Here the *absolute error* $|x - x^*|$ is

$$\begin{aligned} |x - \mathrm{chop}(x)| &= \left|\pm \sum_{k=t+1}^{\infty} b_{-k}2^{-k}\right| 2^{e} \\ &\le \sum_{k=t+1}^{\infty} 2^{-k}\cdot 2^{e} = 2^{-t}\cdot 2^{e}. \end{aligned}$$

It depends on e (i.e., the magnitude of x), which is the reason why one prefers the *relative error* $|(x - x^*)/x|$ (if $x \neq 0$), which, for normalized x, can be estimated as

$$\left|\frac{x - \mathrm{chop}(x)}{x}\right| \le \frac{2^{-t}\cdot 2^{e}}{\left|\pm\sum_{k=1}^{\infty} b_{-k}2^{-k}\right| 2^{e}} \le \frac{2^{-t}\cdot 2^{e}}{\frac{1}{2}\cdot 2^{e}} = 2\cdot 2^{-t}. \tag{1.11}$$

Similarly, in the case of symmetric rounding, one finds (cf. Ex. 6)

$$\left|\frac{x - \mathrm{rd}(x)}{x}\right| \leq 2^{-t}. \tag{1.12}$$

The number on the right is an important, machine-dependent quantity, called the *machine precision* (or *unit roundoff*),

$$\mathrm{eps} = 2^{-t}; \tag{1.13}$$

it determines the level of precision of any large-scale floating-point computation. In Matlab double-precision arithmetic, one has $t = 53$, so that eps $\approx 1.11 \times 10^{-16}$ (cf. Ex. 5), corresponding to a precision of 15–16 significant decimal digits.

Since it is awkward to work with inequalities, one prefers writing (1.12) equivalently as an equality,

$$\mathrm{rd}(x) = x(1 + \varepsilon), \quad |\varepsilon| \leq \mathrm{eps}, \tag{1.14}$$

and defers dealing with the inequality (for ε) to the very end.

1.2 Machine Arithmetic

The arithmetic used on computers unfortunately does not respect the laws of ordinary arithmetic. Each elementary floating-point operation, in general, generates a small error that may then propagate through subsequent machine operations. As a rule, this error propagation is harmless, except in the case of subtraction, where cancellation effects may seriously compromise the accuracy of the results.

1.2.1 A Model of Machine Arithmetic

Any of the four basic arithmetic operations, when applied to two machine numbers, may produce a result no longer representable on the computer. We have therefore errors also associated with arithmetic operations. Barring the occurrence of overflow or underflow, we may assume as a *model of machine arithmetic* that each arithmetic operation $\circ$ $(= +, -, \times, /)$ produces a correctly rounded result. Thus, if $x, y \in \mathbb{R}(t, s)$ are floating-point machine numbers, and $\mathrm{fl}(x \circ y)$ denotes the machine-produced result of the arithmetic operation $x \circ y$, then

$$\mathrm{fl}(x \circ y) = x \circ y\,(1 + \varepsilon), \quad |\varepsilon| \leq \mathrm{eps}. \tag{1.15}$$

This can be interpreted in a number of ways, for example, in the case of multiplication,

$$\mathrm{fl}(x \times y) = [x(1+\varepsilon)] \times y = x \times [y(1+\varepsilon)] = (x\sqrt{1+\varepsilon}) \times (y\sqrt{1+\varepsilon}) = \cdots .$$

In each equation we identify the computed result as the exact result on data that are slightly perturbed, whereby the respective relative perturbations can be estimated, for example, by $|\varepsilon| \leq \mathrm{eps}$ in the first two equations, and $\sqrt{1+\epsilon} \approx 1 + \frac{1}{2}\varepsilon$, $\left|\frac{1}{2}\varepsilon\right| \leq \frac{1}{2}\mathrm{eps}$ in the third. These are elementary examples of *backward error analysis*, a powerful tool for estimating errors in machine computation (cf. also Sect. 1.3).

Even though a single arithmetic operation causes a small error that can be neglected, a succession of arithmetic operations can well result in a significant error, owing to *error propagation*. It is like the small microorganisms that we all carry in our bodies: if our defense mechanism is in good order, the microorganisms cause no harm, in spite of their large presence. If for some reason our defenses are weakened, then all of a sudden they can play havoc with our health. The same is true in machine computation: the rounding errors, although widespread, will cause little harm unless our computations contain some weak spots that allow rounding errors to take over to the point of completely invalidating the results. We learn about one such weak spot (indeed the only one) in the next section.[1]

1.2.2 Error Propagation in Arithmetic Operations: Cancellation Error

We now study the extent to which the basic arithmetic operations propagate errors already present in their operands. Previously, in Sect. 1.2.1, we assumed the

[1] Rounding errors can also have significant implications in real life. One example, taken from politics, concerns the problem of apportionment: how should the representatives in an assembly, such as the US House of Representatives or the Electoral College, be constituted to fairly reflect the size of population in the various states? If the total number of representatives in the assembly is given, say, A, the total population of the US is P, and the population of State i is p_i, then State i should be allocated

$$r_i = \frac{p_i}{P}A$$

representatives. The problem is that r_i is not an integer, in general. How then should r_i be rounded to an integer r_i^*? One can think of three natural criteria to be imposed: (1) r_i^* should be one of the two integers closest to r_i ("quota condition"). (2) If A is increased, all other things being the same, then r_i^* should not decrease ("house monotonicity"). (3) If p_i is increased, the other p_j remaining constant, then r_i^* should not decrease ("population monotonicity"). Unfortunately, there is no apportionment method that satisfies all three criteria. There is indeed a case in US history when Samuel J. Tilden lost his bid for the presidency in 1876 in favor of Rutherford B. Hayes, purely on the basis of the apportionment method adopted on that occasion (which, incidentally, was not the one prescribed by law at the time).

operands to be exact machine-representable numbers and discussed the errors due to imperfect execution of the arithmetic operations by the computer. We now change our viewpoint and assume that the operands themselves are contaminated by errors, but the arithmetic operations are carried out exactly. (We already know what to do, cf. (1.15), when we are dealing with machine operations.) Our interest is in the errors in the results caused by errors in the data.

(a) *Multiplication*. We consider values $x(1+\varepsilon_x)$ and $y(1+\varepsilon_y)$ of x and y contaminated by relative errors ε_x and ε_y, respectively. What is the relative error in the product? We assume ε_x, ε_y sufficiently small so that quantities of second order, ε_x^2, $\varepsilon_x\varepsilon_y$, ε_y^2 – and even more so, quantities of still higher order – can be neglected against the epsilons themselves. Then

$$x(1+\varepsilon_x)\cdot y(1+\varepsilon_y) = x\cdot y\,(1+\varepsilon_x+\varepsilon_y+\varepsilon_x\varepsilon_y) \approx x\cdot y\,(1+\varepsilon_x+\varepsilon_y).$$

Thus, the relative error $\varepsilon_{x\cdot y}$ in the product is given (at least approximately) by

$$\varepsilon_{x\cdot y} = \varepsilon_x + \varepsilon_y, \tag{1.16}$$

that is, the (relative) errors in the data are being added to produce the (relative) error in the result. We consider this to be acceptable error propagation, and in this sense, multiplication is a *benign* operation.

(b) *Division*. Here we have similarly (if $y \neq 0$)

$$\begin{aligned}\frac{x(1+\varepsilon_x)}{y(1+\varepsilon_y)} &= \frac{x}{y}(1+\varepsilon_x)(1-\varepsilon_y+\varepsilon_y^2-+\cdots)\\ &\approx \frac{x}{y}(1+\varepsilon_x-\varepsilon_y),\end{aligned}$$

that is,

$$\varepsilon_{x/y} = \varepsilon_x - \varepsilon_y. \tag{1.17}$$

Also division is a benign operation.

(c) *Addition and subtraction*. Since x and y can be numbers of arbitrary signs, it suffices to look at addition. We have

$$\begin{aligned}x(1+\varepsilon_x)+y(1+\varepsilon_y) &= x+y+x\varepsilon_x+y\varepsilon_y\\ &= (x+y)\left(1+\frac{x\varepsilon_x+y\varepsilon_y}{x+y}\right),\end{aligned}$$

assuming $x+y \neq 0$. Therefore,

$$\varepsilon_{x+y} = \frac{x}{x+y}\varepsilon_x + \frac{y}{x+y}\varepsilon_y. \tag{1.18}$$

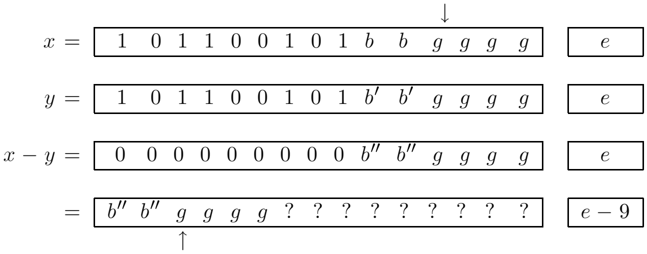

↓
x = 1 0 1 1 0 0 1 0 1 b b g g g g e
y = 1 0 1 1 0 0 1 0 1 b′ b′ g g g g e
x − y = 0 0 0 0 0 0 0 0 0 b″ b″ g g g g e
= b″ b″ g g g g ? ? ? ? ? ? ? ? ? e − 9
↑

Fig. 1.3 The cancellation phenomenon

As before, the error in the result is a linear combination of the errors in the data, but now the coefficients are no longer ± 1 but can assume values that are arbitrarily large. Note first, however, that when x and y have the same sign, then both coefficients are positive and bounded by 1, so that

$$|\varepsilon_{x+y}| \le |\varepsilon_x| + |\varepsilon_y| \quad (x \cdot y > 0); \tag{1.19}$$

addition, in this case, is again a benign operation. It is only when x and y have opposite signs that the coefficients in (1.18) can be arbitrarily large, namely, when $|x + y|$ is arbitrarily small compared to $|x|$ and $|y|$. This happens when x and y are almost equal in absolute value, but opposite in sign. The large magnification of error then occurring in (1.18) is referred to as *cancellation error*. It is the only serious weakness – the Achilles heel, as it were – of numerical computation, and it should be avoided whenever possible. In particular, one should be prepared to encounter cancellation effects not only in single devastating amounts, but also repeatedly over a long period of time involving "small doses" of cancellation. Either way, the end result can be disastrous.

We illustrate the cancellation phenomenon schematically in Fig. 1.3, where b, b', b'' stand for binary digits that are reliable, and the g represent binary digits contaminated by error; these are often called "garbage" digits. Note in Fig. 1.3 that "garbage – garbage = garbage," but, more importantly, that the final normalization of the result moves the first garbage digit from the 12th position to the 3rd.

Cancellation is such a serious matter that we wish to give a number of elementary examples, not only of its occurrence, but also of how it might be avoided.

Examples. 1. An algebraic identity: $(a - b)^2 = a^2 - 2ab + b^2$. Although this is a valid identity in algebra, it is no longer valid in machine arithmetic. Thus, on a 2-decimal-digit computer, with $a = 1.8$, $b = 1.7$, we get, using symmetric rounding,

$$\mathrm{fl}(a^2 - 2ab + b^2) = 3.2 - 6.2 + 2.9 = -0.10$$

instead of the true result 0.010, which we obtain also on our 2-digit computer if we use the left-hand side of the identity. The expanded form of the square thus produces a result which is off by one order of magnitude and on top has the wrong sign.

2. Quadratic equation: $x^2 - 56x + 1 = 0$. The usual formula for a quadratic gives, in 5-decimal arithmetic,

$$x_1 = 28 - \sqrt{783} = 28 - 27.982 = 0.018000,$$

$$x_2 = 28 + \sqrt{783} = 28 + 27.982 = 55.982.$$

   This should be contrasted with the exact roots 0.0178628... and 55.982137... . As can be seen, the smaller of the two is obtained to only two correct decimal digits, owing to cancellation. An easy way out, of course, is to compute x_2 first, which involves a benign addition, and then to compute $x_1 = 1/x_2$ by Vieta's formula, which again involves a benign operation – division. In this way we obtain both roots to full machine accuracy.
3. Compute $y = \sqrt{x+\delta} - \sqrt{x}$, where $x > 0$ and $|\delta|$ is very small. Clearly, the formula as written causes severe cancellation errors, since each square root has to be rounded. Writing instead

$$y = \frac{\delta}{\sqrt{x+\delta} + \sqrt{x}}$$

   completely removes the problem.
4. Compute $y = \cos(x + \delta) - \cos x$, where $|\delta|$ is very small. Here cancellation can be avoided by writing y in the equivalent form

$$y = -2 \sin \frac{\delta}{2} \sin \left(x + \frac{\delta}{2} \right).$$

5. Compute $y = f(x+\delta) - f(x)$, where $|\delta|$ is very small and f a given function. Special tricks, such as those used in the two preceding examples, can no longer be played, but if f is sufficiently smooth in the neighborhood of x, we can use Taylor expansion:

$$y = f'(x)\delta + \frac{1}{2} f''(x)\delta^2 + \cdots .$$

   The terms in this series decrease rapidly when $|\delta|$ is small so that cancellation is no longer a problem.

Addition is an example of a potentially ill-conditioned function (of two variables). It naturally leads us to study the condition of more general functions.

1.3 The Condition of a Problem

A problem typically has an input and an output. The input consists of a set of data, say, the coefficients of some equation, and the output of another set of numbers uniquely determined by the input, say, all the roots of the equation in some prescribed order. If we collect the input in a vector $\boldsymbol{x} \in \mathbb{R}^m$ (assuming the data

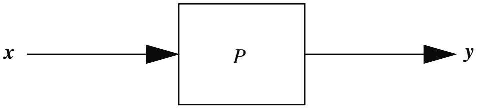


Fig. 1.4 *Black box* representation of a problem

consist of real numbers), and the output in the vector $\boldsymbol{y} \in \mathbb{R}^n$ (also assumed real), we have the black box situation shown in Fig. 1.4, where the box P accepts some input $\boldsymbol{x}$ and then solves the problem for this input to produce the output $\boldsymbol{y}$.

We may thus think of a problem as a map $\boldsymbol{f}$, given by

$$\boldsymbol{f} : \mathbb{R}^m \to \mathbb{R}^n, \quad \boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}). \tag{1.20}$$

(One or both of the spaces $\mathbb{R}^m$, $\mathbb{R}^n$ could be complex spaces without changing in any essential way the discussion that follows.) What we are interested in is the sensitivity of the map $\boldsymbol{f}$ at some given point $\boldsymbol{x}$ to a small perturbation of $\boldsymbol{x}$, that is, how much bigger (or smaller) the perturbation in $\boldsymbol{y}$ is compared to the perturbation in $\boldsymbol{x}$. In particular, we wish to measure the degree of sensitivity by a single number – the *condition number* of the map $\boldsymbol{f}$ at the point $\boldsymbol{x}$. We emphasize that, as we perturb $\boldsymbol{x}$, the function $\boldsymbol{f}$ is always assumed to be evaluated exactly, with infinite precision. The condition of $\boldsymbol{f}$, therefore, is an inherent property of the map $\boldsymbol{f}$ and does not depend on any algorithmic considerations concerning its implementation.

This is not to say that knowledge of the condition of a problem is irrelevant to any algorithmic solution of the problem. On the contrary, the reason is that quite often the *computed* solution $\boldsymbol{y}^*$ of (1.20) (computed in floating-point machine arithmetic, using a specific algorithm) can be demonstrated to be the *exact* solution to a "nearby" problem, that is,

$$\boldsymbol{y}^* = \boldsymbol{f}(\boldsymbol{x}^*), \tag{1.21}$$

where $\boldsymbol{x}^*$ is a vector close to the given data $\boldsymbol{x}$,

$$\boldsymbol{x}^* = \boldsymbol{x} + \boldsymbol{\delta}, \tag{1.22}$$

and moreover, the distance $\|\boldsymbol{\delta}\|$ of $\boldsymbol{x}^*$ to $\boldsymbol{x}$ can be estimated in terms of the machine precision. Therefore, if we know how strongly (or weakly) the map $\boldsymbol{f}$ reacts to a small perturbation, such as $\boldsymbol{\delta}$ in (1.22), we can say something about the error $\boldsymbol{y}^* - \boldsymbol{y}$ in the solution caused by this perturbation. This, indeed, is an important technique of error analysis – known as *backward error analysis* – which was pioneered in the 1950s by J. W. Givens, C. Lanczos, and, above all, J. H. Wilkinson.

Maps $\boldsymbol{f}$ between more general spaces (in particular, function spaces) have also been considered from the point of view of conditioning, but eventually, these spaces have to be reduced to finite-dimensional spaces for practical implementation of the maps in question.

1.3.1 Condition Numbers

We start with the simplest case of a single function of one variable.

The case $m = n = 1$: $y = f(x)$. Assuming first $x \neq 0$, $y \neq 0$, and denoting by Δx a small perturbation of x, we have for the corresponding perturbation Δy by Taylor's formula

$$\Delta y = f(x + \Delta x) - f(x) \approx f'(x)\Delta x, \tag{1.23}$$

assuming that f is differentiable at x. Since our interest is in *relative* errors, we write this in the form

$$\frac{\Delta y}{y} \approx \frac{xf'(x)}{f(x)} \cdot \frac{\Delta x}{x}\,. \tag{1.24}$$

The approximate equality becomes a true equality in the limit as $\Delta x \to 0$. This suggests that the condition of f at x be defined by the quantity

$$(\mathrm{cond}\, f)(x) := \left|\frac{xf'(x)}{f(x)}\right|. \tag{1.25}$$

This number tells us how much larger the relative perturbation in y is compared to the relative perturbation in x.

If $x = 0$ and $y \neq 0$, it is more meaningful to consider the absolute error measure for x and for y still the relative error. This leads to the condition number $|f'(x)/f(x)|$. Similarly for $y = 0$, $x \neq 0$. If $x = y = 0$, the condition number by (1.23) would then simply be $|f'(x)|$.

The case of arbitrary m, n: here we write

$$\boldsymbol{x} = [x_1, x_2, \ldots, x_m]^{\mathrm{T}} \in \mathbb{R}^m, \quad \boldsymbol{y} = [y_1, y_2, \ldots, y_n]^{\mathrm{T}} \in \mathbb{R}^n$$

and exhibit the map $\boldsymbol{f}$ in component form

$$y_\nu = f_\nu(x_1, x_2, \ldots, x_m), \quad \nu = 1, 2, \ldots, n. \tag{1.26}$$

We assume again that each function f_ν has partial derivatives with respect to all m variables at the point $\boldsymbol{x}$. Then the most detailed analysis departs from considering each component y_ν as a function of one single variable, x_μ. In other words, we subject only one variable, x_μ, to a small change and observe the resulting change in just one component, y_ν. Then we can apply (1.25) and obtain

$$\gamma_{\nu\mu}(\boldsymbol{x}) := (\mathrm{cond}_{\nu\mu}\boldsymbol{f})(\boldsymbol{x}) := \left|\frac{x_\mu \frac{\partial f_\nu}{\partial x_\mu}}{f_\nu(\boldsymbol{x})}\right|\,. \tag{1.27}$$

This gives us a whole matrix $\boldsymbol{\Gamma}(\boldsymbol{x}) = [\gamma_{\nu\mu}(\boldsymbol{x})] \in \mathbb{R}_{+}^{n\times m}$ of condition numbers. To obtain a single condition number, we can take any convenient measure of the "magnitude" of the matrix $\boldsymbol{\Gamma}(\boldsymbol{x})$ such as one of the matrix norms defined in (1.30),

$$(\mathrm{cond}\,\boldsymbol{f})(\boldsymbol{x}) = \|\boldsymbol{\Gamma}(\boldsymbol{x})\|, \quad \boldsymbol{\Gamma}(\boldsymbol{x}) = [\gamma_{\nu\mu}(\boldsymbol{x})]. \tag{1.28}$$

The condition so defined, of course, depends on the choice of norm, but the order of magnitude (and that is all that counts) should be more or less the same for any reasonable norm.

If a component of $\boldsymbol{x}$, or of $\boldsymbol{y}$, vanishes, one modifies (1.27) as discussed earlier.

A less-refined analysis can be modeled after the one-dimensional case by defining the relative perturbation of $\boldsymbol{x} \in \mathbb{R}^m$ to mean

$$\frac{\|\Delta\boldsymbol{x}\|_{\mathbb{R}^m}}{\|\boldsymbol{x}\|_{\mathbb{R}^m}}, \quad \Delta\boldsymbol{x} = [\Delta x_1, \Delta x_2, \dots, \Delta x_m]^{\mathrm{T}}, \tag{1.29}$$

where $\Delta\boldsymbol{x}$ is a perturbation vector whose components Δx_μ are small compared to x_μ, and where $\|\cdot\|_{\mathbb{R}^m}$ is some vector norm in $\mathbb{R}^m$. For the perturbation $\Delta\boldsymbol{y}$ caused by $\Delta\boldsymbol{x}$, one defines similarly the relative perturbation $\|\Delta\boldsymbol{y}\|_{\mathbb{R}^n}/\|\boldsymbol{y}\|_{\mathbb{R}^n}$, with a suitable vector norm $\|\cdot\|_{\mathbb{R}^n}$ in $\mathbb{R}^n$. One then tries to relate the relative perturbation in $\boldsymbol{y}$ to the one in $\boldsymbol{x}$.

To carry this out, one needs to define a matrix norm for matrices $\boldsymbol{A} \in \mathbb{R}^{n\times m}$. We choose the so-called "operator norm,"

$$\|\boldsymbol{A}\|_{\mathbb{R}^{n\times m}} := \max_{\substack{x\in\mathbb{R}^m \\ x\neq \mathbf{0}}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_{\mathbb{R}^n}}{\|\boldsymbol{x}\|_{\mathbb{R}^m}}\,. \tag{1.30}$$

In the following we take for the vector norms the "uniform" (or infinity) norm,

$$\|\boldsymbol{x}\|_{\mathbb{R}^m} = \max_{1\leq\mu\leq m} |x_\mu| =: \|\boldsymbol{x}\|_\infty, \quad \|\boldsymbol{y}\|_{\mathbb{R}^n} = \max_{1\leq\nu\leq n} |y_\nu| =: \|\boldsymbol{y}\|_\infty\,. \tag{1.31}$$

It is then easy to show that (cf. Ex. 32)

$$\|\boldsymbol{A}\|_{\mathbb{R}^{n\times m}} = \|\boldsymbol{A}\|_\infty := \max_{1\leq\nu\leq n} \sum_{\mu=1}^{m} |a_{\nu\mu}|, \quad \boldsymbol{A} = [a_{\nu\mu}] \in \mathbb{R}^{n\times m}. \tag{1.32}$$

Now in analogy to (1.23), we have

$$\Delta y_\nu = f_\nu(\boldsymbol{x} + \Delta\boldsymbol{x}) - f_\nu(\boldsymbol{x}) \approx \sum_{\mu=1}^{m} \frac{\partial f_\nu}{\partial x_\mu} \Delta x_\mu$$

with the partial derivatives evaluated at $\boldsymbol{x}$. Therefore, at least approximately,

$$\begin{aligned}|\Delta y_\nu| &\le \sum_{\mu=1}^{m}\left|\frac{\partial f_\nu}{\partial x_\mu}\right| |\Delta x_\mu| \le \max_\mu |\Delta x_\mu| \cdot \sum_{\mu=1}^{m}\left|\frac{\partial f_\nu}{\partial x_\mu}\right| \\ &\le \max_\mu |\Delta x_\mu| \cdot \max_\nu \sum_{\mu=1}^{m}\left|\frac{\partial f_\nu}{\partial x_\mu}\right|.\end{aligned}$$

Since this holds for each $\nu = 1, 2, \dots, n$, it also holds for $\max_\nu |\Delta y_\nu|$, giving, in view of (1.31) and (1.32),

$$\|\Delta \boldsymbol{y}\|_\infty \le \|\Delta \boldsymbol{x}\|_\infty \left\|\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right\|_\infty . \tag{1.33}$$

Here,

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_m} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_m} \\ \cdot & \cdot & \cdots & \cdot \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_m} \end{bmatrix} \in \mathbb{R}^{n\times m} \tag{1.34}$$

is the *Jacobian matrix* of $\boldsymbol{f}$. (This is the analogue of the first derivative for *systems* of functions of *several* variables.) From (1.33) one now immediately obtains for the relative perturbations

$$\frac{\|\Delta \boldsymbol{y}\|_\infty}{\|\boldsymbol{y}\|_\infty} \le \frac{\|\boldsymbol{x}\|_\infty \, \|\partial \boldsymbol{f}/\partial \boldsymbol{x}\|_\infty}{\|\boldsymbol{f}(\boldsymbol{x})\|_\infty} \cdot \frac{\|\Delta \boldsymbol{x}\|_\infty}{\|\boldsymbol{x}\|_\infty} .$$

Although this is an inequality, it is sharp in the sense that equality can be achieved for a suitable perturbation $\Delta \boldsymbol{x}$. We are justified, therefore, in defining a global condition number by

$$(\mathrm{cond}\, \boldsymbol{f})(\boldsymbol{x}) := \frac{\|\boldsymbol{x}\|_\infty \|\partial \boldsymbol{f}/\partial \boldsymbol{x}\|_\infty}{\|\boldsymbol{f}(\boldsymbol{x})\|_\infty} . \tag{1.35}$$

Clearly, in the case $m = n = 1$, definition (1.35) reduces precisely to definition (1.25) (as well as (1.28)) given earlier. In higher dimensions (m and/or n larger than 1), however, the condition number in (1.35) is much cruder than the one in (1.28). This is because norms tend to destroy detail: if $\boldsymbol{x}$, for example, has components of vastly different magnitudes, then $\|\boldsymbol{x}\|_\infty$ is simply equal to the largest of these components, and all the others are ignored. For this reason, some caution is required when using (1.35).

To give an example, consider

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} \dfrac{1}{x_1} + \dfrac{1}{x_2} \\ \dfrac{1}{x_1} - \dfrac{1}{x_2} \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

The components of the condition matrix $\boldsymbol{\Gamma}(\boldsymbol{x})$ in (1.27) are then

$$\gamma_{11} = \left|\frac{x_2}{x_1 + x_2}\right|, \ \gamma_{12} = \left|\frac{x_1}{x_1 + x_2}\right|, \ \gamma_{21} = \left|\frac{x_2}{x_2 - x_1}\right|, \ \gamma_{22} = \left|\frac{x_1}{x_2 - x_1}\right|,$$

indicating ill-conditioning if either $x_1 \approx x_2$ or $x_1 \approx -x_2$ and $|x_1|$ (hence also $|x_2|$) is not small. The global condition number (1.35), on the other hand, since

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(\boldsymbol{x}) = -\frac{1}{x_1^2 x_2^2} \begin{bmatrix} x_2^2 & x_1^2 \\ x_2^2 & -x_1^2 \end{bmatrix},$$

becomes, when L_1 vector and matrix norms are used (cf. Ex. 33),

$$(\operatorname{cond} \boldsymbol{f})(\boldsymbol{x}) = \frac{\|\boldsymbol{x}\|_1 \cdot \dfrac{2}{x_1^2 x_2^2} \max(x_1^2,\ x_2^2)}{\dfrac{1}{|x_1 x_2|}(|x_1 + x_2| + |x_1 - x_2|)} = 2\frac{|x_1| + |x_2|}{|x_1 x_2|} \frac{\max(x_1^2,\ x_2^2)}{|x_1 + x_2| + |x_1 - x_2|}.$$

Here $x_1 \approx x_2$ or $x_1 \approx -x_2$ yields $(\operatorname{cond} \boldsymbol{f})(\boldsymbol{x}) \approx 2$, which is obviously misleading.

1.3.2 Examples

We illustrate the idea of numerical condition in a number of examples, some of which are of considerable interest in applications.

1. Compute $I_n = \displaystyle\int_0^1 \frac{t^n}{t+5}\mathrm{d}t$ for some fixed integer $n \geq 1$. As it stands, the example here deals with a map from the integers to reals and therefore does

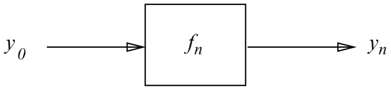


Fig. 1.5 *Black box* for recursion (1.38)

not fit our concept of "problem" in (1.20). However, we propose to compute I_n recursively by relating I_k to I_{k-1} and noting that

$$I_0 = \int_0^1 \frac{\mathrm{d}t}{t+5} = \ln(t+5)\Big|_0^1 = \ln\frac{6}{5}\,. \tag{1.36}$$

To find the recursion, observe that

$$\frac{t}{t+5} = 1 - \frac{5}{t+5}\,.$$

Thus, multiplying both sides by t^{k-1} and integrating from 0 to 1 yields

$$I_k = -5I_{k-1} + \frac{1}{k}, \quad k = 1, 2, \ldots, n. \tag{1.37}$$

We see that I_k is a solution of the (linear, inhomogeneous, first-order) difference equation

$$y_k = -5y_{k-1} + \frac{1}{k}, \quad k = 1, 2, 3, \ldots\,. \tag{1.38}$$

We now have what appears to be a practical scheme to compute I_n: start with $y_0 = I_0$ given by (1.36) and then apply in succession (1.38) for $k = 1, 2, \ldots, n$; then $y_n = I_n$. Recursion (1.38), for any starting value y_0, defines a function,

$$y_n = f_n(y_0). \tag{1.39}$$

We have the black box in Fig. 1.5 and thus a problem $f_n : \mathbb{R} \to \mathbb{R}$. (Here n is a parameter.) We are interested in the condition of f_n at the point $y_0 = I_0$ given by (1.36). Indeed, I_0 in (1.36) is not machine representable and must be rounded to I_0^* before recursion (1.38) can be employed. Even if no further errors are introduced during the recursion, the final result will not be exactly I_n, but some approximation $I_n^* = f_n(I_0^*)$, and we have, at least approximately (actually exactly; see the remark after (1.46)),

$$\left|\frac{I_n^* - I_n}{I_n}\right| = (\mathrm{cond}\, f_n)(I_0)\left|\frac{I_0^* - I_0}{I_0}\right|. \tag{1.40}$$

To compute the condition number, note that f_n is a linear function of y_0. Indeed, if $n = 1$, then

$$y_1 = f_1(y_0) = -5y_0 + 1.$$

If $n = 2$, then

$$y_2 = f_2(y_0) = -5y_1 + \frac{1}{2} = (-5)^2 y_0 - 5 + \frac{1}{2},$$

and so on. In general,

$$y_n = f_n(y_0) = (-5)^n y_0 + p_n,$$

where p_n is some number (independent of y_0). There follows

$$(\mathrm{cond}\, f_n)(y_0) = \left|\frac{y_0 f_n'(y_0)}{y_n}\right| = \left|\frac{y_0(-5)^n}{y_n}\right|. \tag{1.41}$$

Now, if $y_0 = I_0$, then $y_n = I_n$, and from the definition of I_n as an integral it is clear that I_n decreases monotonically in n (and indeed converges monotonically to zero as $n \to \infty$). Therefore,

$$(\mathrm{cond}\, f_n)(I_0) = \frac{I_0 \cdot 5^n}{I_n} > \frac{I_0 \cdot 5^n}{I_0} = 5^n. \tag{1.42}$$

We see that $f_n(y_0)$ is severely ill-conditioned at $y_0 = I_0$, the more so the larger n.

We could have anticipated this result by just looking at the recursion (1.38): we keep multiplying by (–5), which tends to make things bigger, whereas they should get smaller. Thus, there will be continuous cancellation occurring throughout the recursion.

How can we avoid this ill-conditioning? The clue comes from the remark just made: instead of multiplying by a large number, we would prefer dividing by a large number, especially if the results get bigger at the same time. This is accomplished by reversing recurrence (1.38), that is, by choosing an $\nu > n$ and computing

$$y_{k-1} = \frac{1}{5}\left(\frac{1}{k} - y_k\right), \quad k = \nu, \nu - 1, \ldots, n + 1. \tag{1.43}$$

The problem then, of course, is how to compute the starting value y_ν. Before we deal with this, let us observe that we now have a new black box, as shown in Fig. 1.6.

As before, the function involved, g_n, is a linear function of y_ν, and an argument similar to the one leading to (1.41) then gives

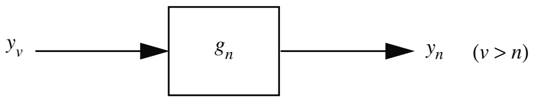


Fig. 1.6 *Black box* for the recursion (1.43)

$$(\operatorname{cond} g_n)(y_\nu) = \left| \frac{y_\nu \left(-\frac{1}{5}\right)^{\nu-n}}{y_n} \right|, \quad \nu > n. \tag{1.44}$$

For $y_\nu = I_\nu$, we get, again by the monotonicity of I_n,

$$(\operatorname{cond} g_n)(I_\nu) < \left(\frac{1}{5}\right)^{\nu-n}, \quad \nu > n. \tag{1.45}$$

In analogy to (1.40), we now have

$$\left| \frac{I_n^* - I_n}{I_n} \right| = (\operatorname{cond} g_n)(I_\nu) \left| \frac{I_\nu^* - I_\nu}{I_\nu} \right| < \left(\frac{1}{5}\right)^{\nu-n} \left| \frac{I_\nu^* - I_\nu}{I_\nu} \right|, \tag{1.46}$$

where I_ν^* is some approximation of I_ν. Actually, I_ν^* does not even have to be close to I_ν for (1.46) to hold, since the function g_n is linear. Thus, we may take $I_\nu^* = 0$, committing a 100% error in the starting value, yet obtaining I_n^* with a relative error

$$\left| \frac{I_n^* - I_n}{I_n} \right| < \left(\frac{1}{5}\right)^{\nu-n}, \quad \nu > n. \tag{1.47}$$

The bound on the right can be made arbitrarily small, say, $\leq \varepsilon$, if we choose ν large enough, for example,

$$\nu \geq n + \frac{\ln \frac{1}{\varepsilon}}{\ln 5}. \tag{1.48}$$

The final procedure, therefore, is: given the desired relative accuracy ε, choose ν to be the smallest integer satisfying (1.48) and then compute

$$\begin{aligned} I_\nu^* &= 0, \\ I_{k-1}^* &= \frac{1}{5}\left(\frac{1}{k} - I_k^*\right), \quad k = \nu, \nu-1, \ldots, n+1. \end{aligned} \tag{1.49}$$

This will produce a sufficiently accurate $I_n^* \approx I_n$, even in the presence of rounding errors committed in (1.49): they, too, will be consistently attenuated.

Similar ideas can be applied to the more important problem of computing solutions to second-order linear recurrence relations such as those satisfied by Bessel functions and many other special functions of mathematical physics.

The procedure of backward recurrence is then closely tied up with the theory of continued fractions.

2. *Algebraic equations*: these are equations involving a polynomial of given degree n,

$$p(x) = 0,\ p(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0, \quad a_0 \neq 0. \tag{1.50}$$

Let ξ be some fixed root of the equation, which we assume to be simple,

$$p(\xi) = 0,\ \ p'(\xi) \neq 0. \tag{1.51}$$

The problem then is to find ξ, given p. The data vector $\boldsymbol{a} = [a_0, a_1, \dots, a_{n-1}]^{\mathrm{T}} \in \mathbb{R}^n$ consists of the coefficients of the polynomial p, and the result is ξ, a real or complex number. Thus, we have

$$\boldsymbol{\xi} :\ \ \mathbb{R}^n \to \mathbb{C},\ \ \xi = \xi(a_0, a_1, \dots, a_{n-1}). \tag{1.52}$$

What is the condition of $\boldsymbol{\xi}$? We adopt the detailed approach of (1.27) and first define

$$\gamma_\nu = (\mathrm{cond}_\nu\, \boldsymbol{\xi})(\boldsymbol{a}) = \left| \frac{a_\nu \frac{\partial \xi}{\partial a_\nu}}{\xi} \right|, \quad \nu = 0, 1, \dots, n-1. \tag{1.53}$$

Then we take a convenient norm, say, the L_1 norm $\|\boldsymbol{\gamma}\|_1 := \sum_{\nu=0}^{n-1} |\gamma_\nu|$ of the vector $\boldsymbol{\gamma} = [\gamma_0, \dots, \gamma_{n-1}]^{\mathrm{T}}$, to define

$$(\mathrm{cond}\, \boldsymbol{\xi})(\boldsymbol{a}) = \sum_{\nu=0}^{n-1} (\mathrm{cond}_\nu\, \boldsymbol{\xi})(\boldsymbol{a}). \tag{1.54}$$

To determine the partial derivative of ξ with respect to a_ν, observe that we have the identity

$$[\xi(a_0, a_1, \dots, a_n)]^n + a_{n-1}[\xi(\cdots)]^{n-1} + \cdots + a_\nu[\xi(\cdots)]^\nu + \cdots + a_0 \equiv 0.$$

Differentiating this with respect to a_ν, we get

$$n[\xi(a_0, a_1, \dots, a_n)]^{n-1}\frac{\partial \xi}{\partial a_\nu} + a_{n-1}(n-1)[\xi(\cdots)]^{n-2}\frac{\partial \xi}{\partial a_\nu} + \cdots$$

$$+ a_\nu \nu[\xi(\cdots)]^{\nu-1}\frac{\partial \xi}{\partial a_\nu} + \cdots + a_1\frac{\partial \xi}{\partial a_\nu} + [\xi(\cdots)]^\nu \equiv 0,$$

where the last term comes from differentiating the first factor in the product $a_\nu \xi^\nu$. The last identity can be written as

$$p'(\xi)\frac{\partial \xi}{\partial a_\nu} + \xi^\nu = 0.$$

Since $p'(\xi) \neq 0$, we can solve for $\partial\xi/\partial a_\nu$ and insert the result in (1.53) and (1.54) to obtain

$$(\mathrm{cond}\,\boldsymbol{\xi})(\boldsymbol{a}) = \frac{1}{|\xi p'(\xi)|}\sum_{\nu=0}^{n-1} |a_\nu|\,|\xi|^\nu. \tag{1.55}$$

We illustrate (1.55) by considering the polynomial p of degree n that has the zeros $1, 2, \ldots, n$,

$$p(x) = \prod_{\nu=1}^{n}(x-\nu) = x^n + a_{n-1}x^{n-1} + \cdots + a_0. \tag{1.56}$$

This is a famous example due to J. H. Wilkinson, who discovered the ill-conditioning of some of the zeros almost by accident. If we let $\xi_\mu = \mu$, $\mu = 1, 2, \ldots, n$, it can be shown that

$$\min_\mu \mathrm{cond}\,\xi_\mu = \mathrm{cond}\,\xi_1 \sim n^2 \;\text{ as }\; n \to \infty,$$

$$\max_\mu \mathrm{cond}\,\xi_\mu \sim \frac{1}{\left(2-\sqrt{2}\right)\pi n}\left(\frac{\sqrt{2}+1}{\sqrt{2}-1}\right)^n \;\text{ as }\; n \to \infty.$$

The worst-conditioned root is ξ_{μ_0} with μ_0 the integer closest to $n/\sqrt{2}$, when n is large. Its condition number grows like $(5.828\ldots)^n$, thus exponentially fast in n. For example, when $n = 20$, then $\mathrm{cond}\,\xi_{\mu_0} = 0.540 \times 10^{14}$.

The example teaches us that the roots of an algebraic equation written in the form (1.50) can be extremely sensitive to small changes in the coefficients a_ν. It would, therefore, be ill-advised to express every polynomial in terms of powers, as in (1.56) and (1.50). This is particularly true for characteristic polynomials of matrices. It is much better here to work with the matrices themselves and try to reduce them (by similarity transformations) to a form that allows the eigenvalues – the roots of the characteristic equation – to be read off relatively easily.

3. *Systems of linear algebraic equations*: given a nonsingular square matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, and a vector $\boldsymbol{b} \in \mathbb{R}^n$, the problem now discussed is solving the system

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}. \tag{1.57}$$

Here the data are the elements of $\boldsymbol{A}$ and $\boldsymbol{b}$, and the result the vector $\boldsymbol{x}$. The map in question is thus $\mathbb{R}^{n^2+n} \to \mathbb{R}^n$. To simplify matters, let us assume that $\boldsymbol{A}$ is a fixed matrix not subject to change, and only the vector $\boldsymbol{b}$ is undergoing perturbations. We then have a map $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ given by

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{b}) := \boldsymbol{A}^{-1}\boldsymbol{b}.$$

It is in fact a linear map. Therefore, $\partial \boldsymbol{f}/\partial \boldsymbol{b} = \boldsymbol{A}^{-1}$, and we get, using (1.35),

$$(\text{cond}\, \boldsymbol{f})(\boldsymbol{b}) = \frac{\|\boldsymbol{b}\|\,\|\boldsymbol{A}^{-1}\|}{\|\boldsymbol{A}^{-1}\boldsymbol{b}\|}\,, \tag{1.58}$$

where we may take any vector norm in $\mathbb{R}^n$ and associated matrix norm (cf. (1.30)). We can write (1.58) alternatively in the form

$$(\text{cond}\, \boldsymbol{f})(\boldsymbol{b}) = \frac{\|\boldsymbol{A}\boldsymbol{x}\|\,\|\boldsymbol{A}^{-1}\|}{\|\boldsymbol{x}\|} \quad (\text{where } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}),$$

and since there is a one-to-one correspondence between $\boldsymbol{x}$ and $\boldsymbol{b}$, we find for the worst condition number

$$\max_{\substack{\boldsymbol{b}\in\mathbb{R}^n \\ \boldsymbol{b}\neq\boldsymbol{0}}} (\text{cond}\, \boldsymbol{f})(\boldsymbol{b}) = \max_{\substack{\boldsymbol{x}\in\mathbb{R}^n \\ \boldsymbol{x}\neq\boldsymbol{0}}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \cdot \|\boldsymbol{A}^{-1}\| = \|\boldsymbol{A}\| \cdot \|\boldsymbol{A}^{-1}\|\,,$$

by definition of the norm of $\boldsymbol{A}$. The number on the far right no longer depends on the particular system (i.e., on $\boldsymbol{b}$) and is called the *condition number* of the matrix $\boldsymbol{A}$. We denote it by

$$\text{cond}\, \boldsymbol{A} := \|\boldsymbol{A}\| \cdot \|\boldsymbol{A}^{-1}\|\,. \tag{1.59}$$

It should be clearly understood, though, that it measures the condition of a linear system with coefficient matrix $\boldsymbol{A}$, and not the condition of other quantities that may depend on $\boldsymbol{A}$, such as eigenvalues.

Although we have considered only perturbations in the right-hand vector $\boldsymbol{b}$, it turns out that the condition number in (1.59) is also relevant when perturbations in the matrix $\boldsymbol{A}$ are allowed, provided they are sufficiently small (so small, for example, that $\|\Delta\boldsymbol{A}\| \cdot \|\boldsymbol{A}^{-1}\| < 1$).

We illustrate (1.59) by several examples.

(a) Hilbert[2] matrix:

[2] David Hilbert (1862–1943) was the most prominent member of the Göttingen school of mathematics. Hilbert's fundamental contributions to almost all parts of mathematics – algebra, number theory, geometry, integral equations, calculus of variations, and foundations – and in particular the 23 now famous problems he proposed in 1900 at the International Congress of Mathematicians in Paris gave a new impetus, and new directions, to 20th-century mathematics. Hilbert is also known for his work in mathematical physics, where among other things he formulated a variationl principle for Einstein's equations in the theory of relativity.

$$\boldsymbol{H}_n = \begin{bmatrix} 1 & \frac{1}{2} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+1} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{1}{n} & \frac{1}{n+1} & \cdots & \frac{1}{2n-1} \end{bmatrix} \in \mathbb{R}^{n\times n}. \tag{1.60}$$

This is clearly a symmetric matrix, and it is also positive definite. Some numerical values for the condition number of $\boldsymbol{H}_n$, computed with the Euclidean norm,[3] are shown in Table 1.1. Their rapid growth is devastating.

Table 1.1 The condition of Hilbert matrices

n	$\text{cond}_2\boldsymbol{H}_n$
10	1.60×10^{13}
20	2.45×10^{28}
40	7.65×10^{58}

A system of order $n = 10$, for example, cannot be solved with any reliability in single precision on a 14-decimal computer. Double precision will be "exhausted" by the time we reach $n = 20$. The Hilbert matrix thus is a prototype of an ill-conditioned matrix. From a result of G. Szegő it can be seen that

$$\text{cond}_2\boldsymbol{H}_n \sim \frac{\left(\sqrt{2}+1\right)^{4n+4}}{2^{15/4}\sqrt{\pi n}} \quad \text{as } n \to \infty.$$

(b) Vandermonde[4] matrices: these are matrices of the form

[3] We have $\text{cond}_2\boldsymbol{H}_n = \lambda_{\max}(\boldsymbol{H}_n)\cdot\lambda_{\max}(\boldsymbol{H}_n^{-1})$, where $\lambda_{\max}(\boldsymbol{A})$ denotes the largest eigenvalue of the (symmetric, positive definite) matrix $\boldsymbol{A}$. The eigenvalues of $\boldsymbol{H}_n$ and $\boldsymbol{H}_n^{-1}$ are easily computed by the Matlab routine `eig`, provided that the inverse of $\boldsymbol{H}_n$ is computed directly from well-known formulae (not by inversion); see MA 9.

[4] Alexandre Théophile Vandermonde (1735–1796), a musician by training, but through acquaintance with Fontaine turned mathematician (temporarily), and even elected to the French Academy of Sciences, produced a total of four mathematical papers within 3 years (1770–1772). Though written by a novice to mathematics, they are not without interest. The first, e.g., made notable contributions to the then emerging theory of equations. By virtue of his fourth paper, he is regarded as the founder of the theory of determinants. What today is referred to as "Vandermonde determinant," however, does not appear anywhere in his writings. As a member of the Academy, he sat in a committee (together with Lagrange, among others) that was to define the unit of length – the meter. Later in his life, he became an ardent supporter of the French revolution.

$$\boldsymbol{V}_n = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ t_1 & t_2 & \cdots & t_n \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ t_1^{n-1} & t_2^{n-1} & \cdots & t_n^{n-1} \end{bmatrix} \in \mathbb{R}^{n \times n}, \tag{1.61}$$

where $t_1, t_2, \dots, t_n$ are parameters, here assumed real. The condition number of these matrices, in the ∞-norm, has been studied at length. Here are some sample results: if the parameters are equally spaced in [–1,1], that is,

$$t_\nu = 1 - \frac{2(\nu - 1)}{n - 1}, \quad \nu = 1, 2, \dots, n,$$

then

$$\text{cond}_\infty \boldsymbol{V}_n \sim \frac{1}{\pi} \mathrm{e}^{-\pi/4} \mathrm{e}^{n\left(\frac{\pi}{4} + \frac{1}{2} \ln 2\right)}, \quad n \to \infty.$$

Numerical values are shown in Table 1.2. They are not growing quite as fast as those for the Hilbert matrix, but still exponentially fast. Worse than exponential growth is observed if one takes harmonic numbers as parameters,

Table 1.2 The condition of Vandermonde matrices

n	$\text{cond}_\infty \boldsymbol{V}_n$
10	1.36×10^4
20	1.05×10^9
40	6.93×10^{18}
80	3.15×10^{38}

$$t_\nu = \frac{1}{\nu}, \quad \nu = 1, 2, \dots, n.$$

Then indeed

$$\text{cond}_\infty \boldsymbol{V}_n > n^{n+1}.$$

Fortunately, there are not many matrices occurring naturally in applications that are *that* ill-conditioned, but moderately to severely ill-conditioned matrices are no rarity in real-life applications.

1.4 The Condition of an Algorithm

We again assume that we are dealing with a problem $\boldsymbol{f}$ given by

$$\boldsymbol{f}: \ \mathbb{R}^m \to \mathbb{R}^n, \ \ \boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}). \tag{1.62}$$

Along with the problem $\boldsymbol{f}$, we are also given an algorithm A that "solves" the problem. That is, given a machine vector $\boldsymbol{x} \in \mathbb{R}^m(t,s)$, the algorithm A produces a vector $\boldsymbol{y}_A$ (in machine arithmetic) that is supposed to approximate $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$. Thus, we have another map $\boldsymbol{f}_A$ describing how the problem $\boldsymbol{f}$ is solved by the algorithm A,

$$\boldsymbol{f}_A : \mathbb{R}^m(t,s) \to \mathbb{R}^n(t,s), \quad \boldsymbol{y}_A = \boldsymbol{f}_A(\boldsymbol{x}). \tag{1.63}$$

In order to be able to analyze $\boldsymbol{f}_A$ in these general terms, we must make a basic assumption, namely, that

$$\begin{gathered} \text{for every } \boldsymbol{x} \in \mathbb{R}^m(t,s), \text{ there holds} \\ \boldsymbol{f}_A(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}_A) \text{ for some } \boldsymbol{x}_A \in \mathbb{R}^m. \end{gathered} \tag{1.64}$$

That is, the computed solution corresponding to some input $\boldsymbol{x}$ is the exact solution for some different input $\boldsymbol{x}_A$ (not necessarily a machine vector and not necessarily uniquely determined) that we hope is close to $\boldsymbol{x}$. The closer we can find an $\boldsymbol{x}_A$ to $\boldsymbol{x}$, the more confidence we should place in the algorithm A. We therefore define the condition of A in terms of the $\boldsymbol{x}_A$ closest to $\boldsymbol{x}$ (if there is more than one), by comparing its relative error with the machine precision eps:

$$(\operatorname{cond} A)(\boldsymbol{x}) = \inf_{\boldsymbol{x}_A} \frac{\|\boldsymbol{x}_A - \boldsymbol{x}\|}{\|\boldsymbol{x}\|} / \text{eps}. \tag{1.65}$$

Here the infimum is over all $\boldsymbol{x}_A$ satisfying $\boldsymbol{y}_A = \boldsymbol{f}(\boldsymbol{x}_A)$. In practice, one can take any such $\boldsymbol{x}_A$ and then obtain an upper bound for the condition number:

$$(\operatorname{cond} A)(\boldsymbol{x}) \le \frac{\|\boldsymbol{x}_A - \boldsymbol{x}\|}{\|\boldsymbol{x}\|} / \text{eps}. \tag{1.66}$$

The vector norm in (1.65), respectively, (1.66), can be chosen as seems convenient.

Here are some very elementary examples.

1. Suppose a library routine for the logarithm function $y = \ln x$, for any positive machine number x, produces a y_A satisfying $y_A = [\ln x](1 + \varepsilon)$, $|\varepsilon| \le 5\,\text{eps}$. What can we say about the condition of the underlying algorithm A? We clearly have

   $$y_A = \ln x_A, \quad \text{where } x_A = x^{1+\varepsilon} \text{ (uniquely).}$$

   Consequently,

   $$\left|\frac{x_A - x}{x}\right| = \left|\frac{x^{1+\varepsilon} - x}{x}\right| = |x^\varepsilon - 1| = \left|\mathrm{e}^{\varepsilon \ln x} - 1\right| \approx |\varepsilon \ln x| \le 5\,|\ln x| \cdot \text{eps},$$

and, therefore, $(\operatorname{cond} A)(x) \le 5\,|\ln x|$. The algorithm A is well conditioned, except in the immediate right-hand vicinity of $x = 0$ and for x very large. (In the latter case, however, x is likely to overflow before A becomes seriously ill-conditioned.)

2. Consider the problem

$$\boldsymbol{f}: \ \mathbb{R}^n \to \mathbb{R}, \quad y = x_1 x_2 \cdots x_n.$$

We solve the problem by the obvious algorithm

$$A: \quad \begin{aligned} p_1 &= x_1, \\ p_k &= \mathrm{fl}(x_k p_{k-1}), \quad k = 2, 3, \ldots, n, \\ y_A &= p_n. \end{aligned}$$

Note that x_1 is machine representable, since for the algorithm A we assume $\boldsymbol{x} \in \mathbb{R}^n(t,s)$.

Now using the basic law of machine arithmetic (cf. (1.15)), we get

$$\begin{aligned} p_1 &= x_1, \\ p_k &= x_k p_{k-1}(1 + \varepsilon_k), \quad k = 2, 3, \ldots, n, \quad |\varepsilon_k| \le \mathrm{eps}, \end{aligned}$$

from which

$$p_n = x_1 x_2 \cdots x_n (1 + \varepsilon_2)(1 + \varepsilon_3) \cdots (1 + \varepsilon_n).$$

Therefore, we can take for example (there is no uniqueness),

$$\boldsymbol{x}_A = [x_1, x_2(1 + \varepsilon_2), \ldots, x_n(1 + \varepsilon_n)]^{\mathrm{T}}.$$

This gives, using the ∞-norm,

$$\frac{\|\boldsymbol{x}_A - \boldsymbol{x}\|_\infty}{\|\boldsymbol{x}\|_\infty \mathrm{eps}} = \frac{\|[0, x_2\varepsilon_2, \ldots, x_n\varepsilon_n]^{\mathrm{T}}\|_\infty}{\|\boldsymbol{x}\|_\infty \mathrm{eps}} \le \frac{\|\boldsymbol{x}\|_\infty \mathrm{eps}}{\|\boldsymbol{x}\|_\infty \mathrm{eps}} = 1,$$

and so, by (1.66), $(\operatorname{cond} A)(\boldsymbol{x}) \le 1$ for any $\boldsymbol{x} \in \mathbb{R}^n(t,s)$. Our algorithm, to nobody's surprise, is perfectly well conditioned.

1.5 Computer Solution of a Problem; Overall Error

The problem to be solved is again

$$\boldsymbol{f}: \ \mathbb{R}^m \to \mathbb{R}^n, \quad \boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}). \tag{1.67}$$

This is the mathematical (idealized) problem, where the data are exact real numbers, and the solution is the mathematically exact solution.

When solving such a problem on a computer, in floating-point arithmetic with precision eps, and using some algorithm A, one first of all rounds the data, and then applies to these rounded data not $\boldsymbol{f}$, but $\boldsymbol{f}_A$:

$$\begin{aligned} \boldsymbol{x}^* &= \text{rounded data}, \quad \frac{\|\boldsymbol{x}^* - \boldsymbol{x}\|}{\|\boldsymbol{x}\|} = \varepsilon, \\ \boldsymbol{y}_A^* &= \boldsymbol{f}_A(\boldsymbol{x}^*). \end{aligned} \tag{1.68}$$

Here ε represents the rounding error in the data. (The error ε could also be due to sources other than rounding, e.g., measurement.) The total error that we wish to estimate is then

$$\frac{\|\boldsymbol{y}_A^* - \boldsymbol{y}\|}{\|\boldsymbol{y}\|}. \tag{1.69}$$

By the basic assumption (1.64) made on the algorithm A, and choosing $\boldsymbol{x}_A^*$ optimally, we have

$$\boldsymbol{f}_A(\boldsymbol{x}^*) = \boldsymbol{f}(\boldsymbol{x}_A^*), \quad \frac{\|\boldsymbol{x}_A^* - \boldsymbol{x}^*\|}{\|\boldsymbol{x}^*\|} = (\operatorname{cond} A)(\boldsymbol{x}^*) \cdot \text{eps}. \tag{1.70}$$

Let $\boldsymbol{y}^* = \boldsymbol{f}(\boldsymbol{x}^*)$. Then, using the triangle inequality, we have

$$\frac{\|\boldsymbol{y}_A^* - \boldsymbol{y}\|}{\|\boldsymbol{y}\|} \le \frac{\|\boldsymbol{y}_A^* - \boldsymbol{y}^*\|}{\|\boldsymbol{y}\|} + \frac{\|\boldsymbol{y}^* - \boldsymbol{y}\|}{\|\boldsymbol{y}\|} \approx \frac{\|\boldsymbol{y}_A^* - \boldsymbol{y}^*\|}{\|\boldsymbol{y}^*\|} + \frac{\|\boldsymbol{y}^* - \boldsymbol{y}\|}{\|\boldsymbol{y}\|},$$

where we have used the (harmless) approximation $\|\boldsymbol{y}\| \approx \|\boldsymbol{y}^*\|$. By virtue of (1.70), we now have for the first term on the right,

$$\begin{aligned} \frac{\|\boldsymbol{y}_A^* - \boldsymbol{y}^*\|}{\|\boldsymbol{y}^*\|} = \frac{\|\boldsymbol{f}_A(\boldsymbol{x}^*) - \boldsymbol{f}(\boldsymbol{x}^*)\|}{\|\boldsymbol{f}(\boldsymbol{x}^*)\|} &= \frac{\|\boldsymbol{f}(\boldsymbol{x}_A^*) - \boldsymbol{f}(\boldsymbol{x}^*)\|}{\|\boldsymbol{f}(\boldsymbol{x}^*)\|} \\ &\le (\operatorname{cond} \boldsymbol{f})(\boldsymbol{x}^*) \cdot \frac{\|\boldsymbol{x}_A^* - \boldsymbol{x}^*\|}{\|\boldsymbol{x}^*\|} \\ &= (\operatorname{cond} \boldsymbol{f})(\boldsymbol{x}^*) \cdot (\operatorname{cond} A)(\boldsymbol{x}^*) \cdot \text{eps}. \end{aligned}$$

For the second term we have

$$\frac{\|\boldsymbol{y}^* - \boldsymbol{y}\|}{\|\boldsymbol{y}\|} = \frac{\|\boldsymbol{f}(\boldsymbol{x}^*) - \boldsymbol{f}(\boldsymbol{x})\|}{\|\boldsymbol{f}(\boldsymbol{x})\|} \leq (\mathrm{cond}\,\boldsymbol{f})(\boldsymbol{x}) \cdot \frac{\|\boldsymbol{x}^* - \boldsymbol{x}\|}{\|\boldsymbol{x}\|} = (\mathrm{cond}\,\boldsymbol{f})(\boldsymbol{x}) \cdot \varepsilon.$$

Assuming finally that $(\mathrm{cond}\,\boldsymbol{f})(\boldsymbol{x}^*) \approx (\mathrm{cond}\,\boldsymbol{f})(\boldsymbol{x})$, we get

$$\frac{\|\boldsymbol{y}_A^* - \boldsymbol{y}\|}{\|\boldsymbol{y}\|} \leq (\mathrm{cond}\,\boldsymbol{f})(\boldsymbol{x})\{\varepsilon + (\mathrm{cond}\,A)(\boldsymbol{x}^*) \cdot \mathrm{eps}\}. \tag{1.71}$$

This shows how the data error and machine precision contribute toward the total error: both are amplified by the condition of the problem, but the latter is further amplified by the condition of the algorithm.

1.6 Notes to Chapter 1

In addition to rounding errors in the data and those committed during the execution of arithmetic operations, there may be other sources of errors not considered in this introductory chapter. One such source of error, which is not entirely dismissible, is a faulty design of the computer chip that executes arithmetic operations. This was brought home in an incident several years ago, when it was discovered (by Thomas Nicely in the course of number-theoretic computations involving reciprocals of twin primes) that the Pentium floating-point divide chip manufactured by Intel can produce erroneous results for certain (extremely rare) bit patterns in the divisor. The incident – rightly so – has stirred up considerable concern and prompted not only remedial actions but also careful analysis of the phenomenon; some relevant articles are those by Coe et al. [1995] and Edelman [1997].

Neither should the occurrence of overflow and proper handling thereof be taken lightly, especially not in real-time applications. Again, a case in point is the failure of the French rocket Ariane 5, which on June 4, 1996, less than a minute into its flight, self-destructed. The failure was eventually traced to an overflow in a floating-point to integer conversion and lack of protection against this occurrence in the rocket's on-board software (cf. Anonymous [1996]).

Many of the topics covered in this chapter, but also the effect of finite precision computation on convergence and stability of mathematical processes, and issues of error analyses are dealt with in Chaitin-Chatelin and Frayssé [1996].

Section 1.1.1. The abstract notion of the real number system is discussed in most texts on real analysis, for example, Hewitt and Stromberg [1975, Chap. 1, Sect. 1.5] or Rudin [1976, Chap. 1]. The development of the concept of real (and complex) numbers has had a long and lively history, extending from pre-Hellenic times to the recent past. Many of the leading thinkers over time contributed to this development. A reader interested in a detailed historical account (and who knows German) is referred to the monograph by Gericke [1970].

Section 1.1.2.1. The notion of the floating-point number system and associated arithmetic, including interval arithmetic, can also be phrased in abstract algebraic terms; see, for example, Kulisch and Miranker [1981]. For a comprehensive treatment of computer arithmetic, including questions of validation, see Kulisch [2008]. A more elementary, but detailed, discussion of floating-point numbers and arithmetic is given in Sterbenz [1974]. There the reader will learn, for example, that computing the average of two floating-point numbers, or solving a quadratic equation, can be fairly intricate tasks if they are to be made foolproof. The quadratic equation problem is also considered at some length in Young and Gregory [1988, Sect. 3.4], where further references are given to earlier work of W. Kahan and G. E. Forsythe.

The basic standard for binary floating-point arithmetic, used on all contemporary computers, is the ANSI/IEEE Standard 754 established in IEEE [1985]. It provides for $t = 23$ bits in the mantissa and $s = 7$ bits in the exponent, in single-precision arithmetic, and has $t = 52$, $s = 11$ in double precision. There is also an "extended precision" for which $t = 63$, $s = 14$, allowing for a number range of approx. 10^{-4964} to 10^{+4964}. A good source for IEEE floating-point arithmetic is Overton [2001].

Section 1.1.2.3. Rational arithmetic is available in all major symbolic computation packages such as Mathematica and Macsyma.

Interval arithmetic has evolved to become an important tool in computations that strive at obtaining guaranteed and sharp inclusion regions for the results of mathematical problems. Basic texts on (real) interval analysis are Moore [1966], [1979], Alefeld and Herzberger [1983], and Moore et al. [2009], whereas complex interval arithmetic is treated in Petković and Petković [1998]. For the newly evolving field of validated numerics we refer to Tucker [2011]. Specific applications such as computing inclusions of the range of functions, of global extrema of functions of one and several variables, and of solutions to systems of linear and nonlinear equations are studied, respectively, in Ratschek and Rokne [1984], [1988], Hansen and Walster [2004], and Neumaier [1990]. Other applications, e.g., to parameter estimation, robust control, and robotics can be found in Jaulin et al. [2001]. Concrete algorithms and codes (in Pascal and C^{++}) for "verified computing" are contained in Hammer et al. [1993], [1995]. Interval arithmetic has been most widely used in processes involving finite-dimensional spaces; for applications to infinite-dimensional problems, notably differential equations, see, however, Eijgenraam [1981] and Kaucher and Miranker [1984]. For a recent expository account, see also Rump [2010].

Section 1.2. For floating-point arithmetic, see the handbook by Muller et al. [2010]. The fact that thoughtless use of mathematical formulae and numerical methods, or inherent sensitivities in a problem, can lead to disastrous results has been known since the early days of computers; see, for example, the old but still relevant papers by Stegun and Abramowitz [1956] and Forsythe [1970]. Nearby singularities can also cause the accuracy to deteriorate unless corrective measures are taken; Forsythe [1958] has an interesting discussion of this.

Section 1.2.1. For the implications of rounding in the problem of apportionment, mentioned in footnote 1, a good reference is Garfunkel and Steen, eds. [1988, Chap. 12, pp.230–249].

Section 1.3.1. An early but basic reference for ideas of conditioning and error analysis in algebraic processes is Wilkinson [1994]. An impressive continuation of this work, containing copious references to the literature, is Higham [2002]. It analyzes the behavior in floating-point arithmetic of virtually all the algebraic processes in current use. Problems of conditioning specifically involving polynomials are discussed in Gautschi [1984]. The condition of general (differentiable) maps has been studied as early as 1966 in Rice [1966].

Section 1.3.2. 1. For a treatment of stability aspects of more general difference equations, and systems thereof, including nonlinear ones, the reader is referred to the monograph by Wimp [1984]. This also contains many applications to special functions. Other relevant texts are Lakshmikantham and Trigiante [2002] and Elaydi [2005].

2. The condition of algebraic equations, although considered already in 1963 by Wilkinson, has been further analyzed by Gautschi [1973]. The circumstances that led to Wilkinson's example (1.56), which he himself describes as "the most traumatic experience in [his] career as a numerical analyst," are related in the essay Wilkinson [1984, Sect. 2]. This reference also deals with errors committed in the evaluation and deflation of polynomials. For the latter, also see Cohen [1994]. The asymptotic estimates for the best- and worst-conditioned roots in Wilkinson's example are from Gautschi [1973]. For the computation of eigenvalues of matrices, the classic treatment is Wilkinson [1988]; more recent accounts are Parlett [1998] for symmetric matrices and Golub and Van Loan [1996, Chap. 7–9] for general matrices.
3. A more complete analysis of the condition of linear systems that also allows for perturbations of the matrix can be found, for example, in the very readable books by Forsythe and Moler [1967, Chap. 8] and Stewart [1973, Chap. 4, Sect. 3]. The asymptotic result of Szegő cited in connection with the Euclidean condition number of the Hilbert matrix is taken from Szegő [1936]. For the explicit inverse of the Hilbert matrix, referred to in footnote 3, see Todd [1954]. The condition of Vandermonde and Vandermonde-like matrices has been studied in a series of papers by the author; for a summary, see Gautschi [1990], and Gautschi [2011b] for optimally scaled and optimally conditioned Vandermonde and Vandermonde-like matrices.

Sections 1.4 and 1.5. The treatment of the condition of algorithms and of the overall error in computer solutions of problems, as given in these sections, seems to be more or less original. Similar ideas, however, can be found in the book by Dahlquist and Björck [2008, Sect. 2.4].

Exercises and Machine Assignments to Chapter 1

Exercises

1. Represent all elements of $\mathbb{R}_+(3,2) = \{x \in \mathbb{R}(3,2) :\ x > 0,\ x \text{ normalized}\}$ as dots on the real axis. For clarity, draw two axes, one from 0 to 8, the other from 0 to $\frac{1}{2}$.
2. (a) What is the distance $d(x)$ of a positive normalized floating-point number $x \in \mathbb{R}(t,s)$ to its next larger floating-point number:

$$d(x) = \min_{\substack{y \in \mathbb{R}(t,s) \\ y > x}} (y - x)\,?$$

   (b) Determine the relative distance $r(x) = d(x)/x$, with x as in (a), and give upper and lower bounds for it.
3. The identity $\mathrm{fl}(1+x) = 1$, $x \geq 0$, is true for $x = 0$ and for x sufficiently small. What is the largest machine number x for which the identity still holds?
4. Consider a miniature binary computer whose floating-point words consist of four binary digits for the mantissa and three binary digits for the exponent (plus sign bits). Let

$$x = (0.1011)_2 \times 2^0, \quad y = (0.1100)_2 \times 2^0.$$

   Mark in the following table whether the machine operation indicated (with the result z assumed normalized) is exact, rounded (i.e., subject to a nonzero rounding error), overflows, or underflows.

Operation	Exact	Rounded	Overflow	Underflow
$z = \mathrm{fl}(x - y)$				
$z = \mathrm{fl}((y - x)^{10})$				
$z = \mathrm{fl}(x + y)$				
$z = \mathrm{fl}(y + (x/4))$				
$z = \mathrm{fl}(x + (y/4))$				

5. The Matlab "machine precision" `eps` is twice the unit roundoff (2×2^{-t}, $t = 53$; cf. Sect. 1.1.3). It can be computed by the following Matlab program (attributed to CLEVE MOLER):

```
%EI_5  Matlab machine precision
%
a=4/3;
b=a-1;
c=b+b+b;
eps0=abs(c-1)
```

Run the program and prove its validity.

6. Prove (1.12).
7. A set S of real numbers is said to possess a metric if there is defined a distance function $d(x, y)$ for any two elements $x, y \in S$ that has the following properties:

   (i) $d(x, y) \geq 0$ and $d(x, y) = 0$ if and only if $x = y$ (positive definiteness);
   (ii) $d(x, y) = d(y, x)$ (symmetry);
   (iii) $d(x, y) \leq d(x, z) + d(z, y)$ (triangle inequality).

   Discuss which of the following error measures is, or is not, a distance function on what set S of real numbers:

   (a) absolute error: $\mathrm{ae}(x, y) = |x - y|$;
   (b) relative error: $\mathrm{re}(x, y) = \left|\frac{x-y}{x}\right|$;
   (c) relative precision (F.W.J. Olver, 1978): $\mathrm{rp}(x, y) = |\ \ln|x| - \ln|y|\ |$.

   If $y = x(1 + \varepsilon)$, show that $\mathrm{rp}(x, y) = O(\varepsilon)$ as $\varepsilon \to 0$.
8. Assume that x_1^*, x_2^* are approximations to x_1, x_2 with relative errors E_1 and E_2, respectively, and that $|E_i| \leq E$, $i = 1, 2$. Assume further that $x_1 \neq x_2$.

   (a) How small must E (in dependence of x_1 and x_2) be to ensure that $x_1^* \neq x_2^*$?
   (b) Taking $\frac{1}{x_1^* - x_2^*}$ to approximate $\frac{1}{x_1 - x_2}$, obtain a bound on the relative error committed, assuming (1) exact arithmetic; (2) machine arithmetic with machine precision eps. (In both cases, neglect higher-order terms in E_1, E_2, eps.)

9. Consider the quadratic equation $x^2 + px + q = 0$ with roots x_1, x_2. As seen in the second Example of Sect. 1.2.2, the absolutely larger root must be computed first, whereupon the other can be accurately obtained from $x_1 x_2 = q$. Suppose one incorporates this idea in a program such as

```
x1=abs(p/2)+sqrt(p*p/4-q);
if p>0, x1=-x1; end
x2=q/x1;
```

   Find two serious flaws with this program as a "general-purpose quadratic equation solver." Take into consideration that the program will be executed in floating-point machine arithmetic. Be specific and support your arguments by examples, if necessary.

10. Suppose, for $|x|$ small, one has an accurate value of $y = \mathrm{e}^x - 1$ (obtained, e.g., by Taylor expansion). Use this value to compute accurately $\sinh x = \frac{1}{2}(\mathrm{e}^x - \mathrm{e}^{-x})$ for small $|x|$.
11. Let $f(x) = \sqrt{1 + x^2} - 1$.

    (a) Explain the difficulty of computing $f(x)$ for a small value of $|x|$ and show how it can be circumvented.
    (b) Compute $(\mathrm{cond}\, f)(x)$ and discuss the conditioning of $f(x)$ for small $|x|$.
    (c) How can the answers to (a) and (b) be reconciled?

12. The nth power of some positive (machine) number x can be computed

    (i) either by repeated multiplication by x, or
    (ii) as $x^n = \mathrm{e}^{n \ln x}$.

    In each case, derive bounds for the relative error due to machine arithmetic, neglecting higher powers of the machine precision against the first power. (Assume that exponentiation and taking logarithms both involve a relative error ε with $|\varepsilon| \leq \mathrm{eps}$.) Based on these bounds, state a criterion (involving x and n) for (i) to be more accurate than (ii).
13. Let $f(x) = (1 - \cos x)/x$, $x \neq 0$.

    (a) Show that direct evaluation of f is inaccurate if $|x|$ is small; assume $\mathrm{fl}(f(x)) = \mathrm{fl}((1 - \mathrm{fl}(\cos x))/x)$, where $\mathrm{fl}(\cos x) = (1 + \varepsilon_c)\cos x$, and estimate the relative error ε_f of $\mathrm{fl}(f(x))$ as $x \to 0$.
    (b) A mathematically equivalent form of f is $f(x) = \sin^2 x/(x(1 + \cos x))$. Carry out a similar analysis as in (a), based on $\mathrm{fl}(f(x)) = \mathrm{fl}([\mathrm{fl}(\sin x)]^2/\mathrm{fl}(x(1 + \mathrm{fl}(\cos x))))$, assuming $\mathrm{fl}(\cos x) = (1 + \varepsilon_c)\cos x$, $\mathrm{fl}(\sin x) = (1 + \varepsilon_s)\sin x$ and retaining only first-order terms in ε_s and ε_c. Discuss the result.
    (c) Determine the condition of $f(x)$. Indicate for what values of x (if any) $f(x)$ is ill-conditioned. ($|x|$ is no longer small, necessarily.)

14. If $z = x + \mathrm{i}y$, then $\sqrt{z} = \left(\frac{r+x}{2}\right)^{1/2} + \mathrm{i}\left(\frac{r-x}{2}\right)^{1/2}$, where $r = (x^2 + y^2)^{1/2}$. Alternatively, $\sqrt{z} = u + \mathrm{i}v$, $u = \left(\frac{r+x}{2}\right)^{1/2}$, $v = y/2u$. Discuss the computational merits of these two (mathematically equivalent) expressions. Illustrate with $z = 4.5 + 0.025\mathrm{i}$, using eight significant decimal places. {*Hint*: you may assume $x > 0$ without restriction of generality. Why?}
15. Consider the numerical evaluation of

$$f(t) = \sum_{n=0}^{\infty} \frac{1}{1 + n^4(t-n)^2(t-n-1)^2},$$

    say, for $t = 20$, and 7-digit accuracy. Discuss the danger involved.
16. Let X_+ be the largest positive machine representable number, and X_- the absolute value of the smallest negative one (so that $-X_- \leq x \leq X_+$ for any machine number x). Determine, approximately, all intervals on $\mathbb{R}$ on which the tangent function overflows.

17. (a) Use Matlab to determine the first value of the integer n for which $n!$ overflows. {*Hint*: use Stirling's formula for $n!$.}
    (b) Do the same as (a), but for x^n, $x = 10, 20, \dots, 100$.
    (c) Discuss how $x^n e^{-x}/n!$ can be computed for large x and n without unnecessarily incurring overflow. {*Hint*: use logarithms and an asymptotic formula for $\ln n!$.}

18. Consider a decimal computer with three (decimal) digits in the floating-point mantissa.

    (a) Estimate the relative error committed in symmetric rounding.
    (b) Let $x_1 = 0.982$, $x_2 = 0.984$ be two machine numbers. Calculate in machine arithmetic the mean $m = \frac{1}{2}(x_1 + x_2)$. Is the computed number between x_1 and x_2?
    (c) Derive sufficient conditions for $x_1 < \mathrm{fl}(m) < x_2$ to hold, where x_1, x_2 are two machine numbers with $0 < x_1 < x_2$.

19. For this problem, assume a binary computer with 12 bits in the floating-point mantissa.

    (a) What is the machine precision eps?
    (b) Let $x = 6/7$ and x^* be the correctly rounded machine approximation to x (symmetric rounding). Exhibit x and x^* as binary numbers.
    (c) Determine (exactly) the relative error ε of x^* as an approximation to x, and calculate the ratio $|\varepsilon|/\mathrm{eps}$.

20. The distributive law of algebra states that

$$(a + b)c = ac + bc.$$

    Discuss to what extent this is violated in machine arithmetic. Assume a computer with machine precision eps and assume that a, b, c are machine-representable numbers.

    (a) Let y_1 be the floating-point number obtained by evaluating $(a + b)c$ (as written) in floating-point arithmetic, and let $y_1 = (a+b)c(1+e_1)$. Estimate $|e_1|$ in terms of eps (neglecting second-order terms in eps).
    (b) Let y_2 be the floating-point number obtained by evaluating $ac + bc$ (as written) in floating-point arithmetic, and let $y_2 = (a+b)c(1+e_2)$. Estimate $|e_2|$ (neglecting second-order terms in eps) in terms of eps (and a, b, and c).
    (c) Identify conditions (if any) under which one of the two y_i is significantly less accurate than the other.

21. Let $x_1, x_2, \dots, x_n$, $n > 1$, be machine numbers. Their product can be computed by the algorithm

$$p_1 = x_1,$$
$$p_k = \mathrm{fl}(x_k p_{k-1}), \quad k = 2, 3, \dots, n.$$

(a) Find an upper bound for the relative error $(p_n - x_1 x_2 \cdots x_n)/(x_1 x_2 \cdots x_n)$ in terms of the machine precision eps and n.

(b) For any integer $r \geq 1$ not too large so as to satisfy $r \cdot \text{eps} < \frac{1}{10}$, show that

$$(1 + \text{eps})^r - 1 < 1.06 \cdot r \cdot \text{eps}.$$

Hence, for n not too large, simplify the answer given in (a). {*Hint*: use the binomial theorem.}

22. Analyze the error propagation in exponentiation, x^α $(x > 0)$:

(a) assuming x exact and α subject to a small relative error ε_α;

(b) assuming α exact and x subject to a small relative error ε_x.

Discuss the possibility of any serious loss of accuracy.

23. Indicate how you would accurately compute

$$(x + y)^{1/4} - y^{1/4}, \quad x > 0, \quad y > 0.$$

24. (a) Let $a = 0.23371258 \times 10^{-4}$, $b = 0.33678429 \times 10^2$, $c = -0.33677811 \times 10^2$. Assuming an 8-decimal-digit computer, determine the sum $s = a + b + c$ either as (1) $\text{fl}(s) = \text{fl}(\text{fl}(a+b)+c)$ or as (2) $\text{fl}(s) = \text{fl}(a+\text{fl}(b+c))$. Explain the discrepancy between the two answers.

(b) For arbitrary machine numbers a, b, c on a computer with machine precision eps, find a criterion on a, b, c for the result of (2) in (a) to be more accurate than the result of (1). {*Hint*: compare bounds on the relative errors, neglecting higher-order terms in eps and assuming $a + b + c \neq 0$; see also MA 7.}

25. Write the expression $a^2 - 2ab \cos\gamma + b^2$ $(a > 0, b > 0)$ as the sum of two positive terms to avoid cancellation errors. Illustrate the advantage gained in the case $a = 16.5$, $b = 15.7$, $\gamma = 5°$, using 3-decimal-digit arithmetic. Is the method foolproof?

26. Determine the condition number for the following functions:

(a) $f(x) = \ln x, \quad x > 0;$

(b) $f(x) = \cos x, \quad |x| < \frac{1}{2}\pi;$

(c) $f(x) = \sin^{-1} x, \quad |x| < 1;$

(d) $f(x) = \sin^{-1} \dfrac{x}{\sqrt{1 + x^2}}.$

Indicate the possibility of ill-conditioning.

27. Compute the condition number of the following functions, and discuss any possible ill-conditioning:

(a) $f(x) = x^{1/n}$ $(x > 0,\ n > 0$ an integer);

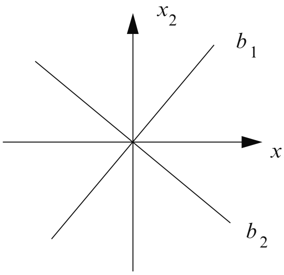
(b) $f(x) = x - \sqrt{x^2 - 1}$ $(x > 1)$;

(c) $\boldsymbol{f}(x_1, x_2) = \sqrt{x_1^2 + x_2^2};$

(d) $\boldsymbol{f}(x_1, x_2) = x_1 + x_2.$

28. (a) Consider the composite function $h(t) = g(f(t))$. Express the condition of h in terms of the condition of g and f. Be careful to state at which points the various condition numbers are to be evaluated.
    (b) Illustrate (a) with $h(t) = \frac{1+\sin t}{1-\sin t}$, $t = \frac{1}{4}\pi$.
29. Show that $(\text{cond}\, f \cdot g)(x) \le (\text{cond}\, f)(x) + (\text{cond}\, g)(x)$. What can be said about $(\text{cond}\, f/g)(x)$?
30. Let $\boldsymbol{f} : \mathbb{R}^2 \to \mathbb{R}$ be given by $y = x_1 + x_2$. Define $(\text{cond}\, \boldsymbol{f})(\boldsymbol{x}) =$ $(\text{cond}_{11}\boldsymbol{f})(\boldsymbol{x})+ (\text{cond}_{12}\boldsymbol{f})(\boldsymbol{x})$, where $\boldsymbol{x} = [x_1, x_2]^{\mathrm{T}}$ (cf. (1.27)).

    (a) Derive a formula for $\kappa(x_1, x_2) = (\text{cond}\, \boldsymbol{f})(\boldsymbol{x})$.
    (b) Show that $\kappa(x_1, x_2)$ as a function of x_1, x_2 is symmetric with respect to both bisectors b_1 and b_2 (see figure).



    (c) Determine the lines in $\mathbb{R}^2$ on which $\kappa(x_1, x_2) = c$, $c \ge 1$ a constant. (Simplify the analysis by using symmetry; cf. part (b).)

31. Let $\|\cdot\|$ be a vector norm in $\mathbb{R}^n$ and denote by the same symbol the associated matrix norm. Show for arbitrary matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times n}$ that

    (a) $\|\boldsymbol{AB}\| \le \|\boldsymbol{A}\|\,\|\boldsymbol{B}\|$;
    (b) $\text{cond}(\boldsymbol{AB}) \le \text{cond}\,\boldsymbol{A} \cdot \text{cond}\,\boldsymbol{B}$.

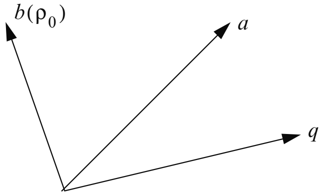
32. Prove (1.32). {*Hint*: let $m_\infty = \max_\nu \sum_\mu |a_{\nu\mu}|$. Show that $\|\boldsymbol{A}\|_\infty \le m_\infty$ as well as $\|\boldsymbol{A}\|_\infty \ge m_\infty$, the latter by taking a special vector $\boldsymbol{x}$ in (1.30).}
33. Let the L_1 norm of a vector $\boldsymbol{y} = [y_\lambda]$ be defined by $\|\boldsymbol{y}\|_1 = \sum_\lambda |y_\lambda|$. For a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times m}$, show that

$$\|\boldsymbol{A}\|_1 := \max_{\substack{\boldsymbol{x}\in\mathbb{R}^m \\ \boldsymbol{x}\ne\boldsymbol{0}}} \frac{\|\boldsymbol{Ax}\|_1}{\|\boldsymbol{x}\|_1} = \max_\mu \sum_\nu |a_{\nu\mu}|,$$

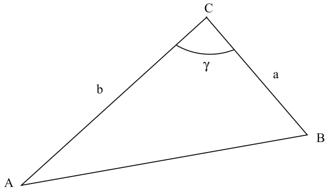
    that is, $\|\boldsymbol{A}\|_1$ is the "maximum column sum." {*Hint*: let $m_1 = \max_\mu \sum_\nu |a_{\nu\mu}|$. Show that $\|\boldsymbol{A}\|_1 \le m_1$ as well as $\|\boldsymbol{A}\|_1 \ge m_1$, the latter by taking for $\boldsymbol{x}$ in (1.30) an appropriate coordinate vector.}
34. Let $\boldsymbol{a}, \boldsymbol{q}$ be linearly independent vectors in $\mathbb{R}^n$ of (Euclidean) length 1. Define $\boldsymbol{b}(\rho) \in \mathbb{R}^n$ as follows:

$$\boldsymbol{b}(\rho) = \boldsymbol{a} - \rho\boldsymbol{q}, \quad \rho \in \mathbb{R}.$$

Compute the condition number of the angle $\alpha(\rho)$ between $\boldsymbol{b}(\rho)$ and $\boldsymbol{q}$ at the value $\rho = \rho_0 = \boldsymbol{q}^{\mathrm{T}}\boldsymbol{a}$. (Then $\boldsymbol{b}(\rho_0) \perp \boldsymbol{q}$; see figure.) Discuss the answer.



35. The area Δ of a triangle ABC is given by $\Delta = \frac{1}{2}ab \sin\gamma$ (see figure). Discuss the numerical condition of Δ.



36. Define, for $x \neq 0$,

$$f_n = f_n(x) = (-1)^n \frac{\mathrm{d}^n}{\mathrm{d}x^n}\left(\frac{\mathrm{e}^{-x}}{x}\right), \quad n = 0, 1, 2, \dots .$$

(a) Show that $\{f_n\}$ satisfies the recursion

$$y_k = \frac{k}{x}\, y_{k-1} + \frac{\mathrm{e}^{-x}}{x}, \quad k = 1, 2, 3, \dots ; \quad y_0 = \frac{\mathrm{e}^{-x}}{x}.$$

{*Hint*: differentiate k times the identity $\mathrm{e}^{-x} = x \cdot (\mathrm{e}^{-x}/x)$.}

(b) Why do you expect the recursion in (a), without doing any analysis, to be numerically stable if $x > 0$? How about $x < 0$?

(c) Support and discuss your answer to (b) by considering y_n as a function of y_0 (which for $y_0 = f_0(x)$ yields $f_n = f_n(x)$) and by showing that the condition number of this function at f_0 is

$$(\mathrm{cond}\, y_n)(f_0) = \frac{1}{|e_n(x)|},$$

where $e_n(x) = 1 + x + x^2/2! + \cdots + x^n/n!$ is the nth partial sum of the exponential series. {*Hint*: use Leibniz's formula to evaluate $f_n(x)$.}

37. Consider the algebraic equation

$$x^n + ax - 1 = 0, \ \ a > 0, \ \ n \geq 2.$$

(a) Show that the equation has exactly one positive root $\xi(a)$.
(b) Obtain a formula for $(\operatorname{cond}\xi)(a)$.
(c) Obtain (good) upper and lower bounds for $(\operatorname{cond}\xi)(a)$.

38. Consider the algebraic equation

$$x^n + x^{n-1} - a = 0, \quad a > 0, \quad n \geq 2.$$

(a) Show that there is exactly one positive root $\xi(a)$.
(b) Show that $\xi(a)$ is well conditioned as a function of a. Indeed, prove

$$(\operatorname{cond}\xi)(a) < \frac{1}{n-1}\,.$$

39. Consider Lambert's equation

$$x\mathrm{e}^x = a$$

for real values of x and a.

(a) Show graphically that the equation has exactly one root $\xi(a) \geq 0$ if $a \geq 0$, exactly two roots $\xi_2(a) < \xi_1(a) < 0$ if $-1/e < a < 0$, a double root -1 if $a = -1/e$, and no root if $a < -1/e$.
(b) Discuss the condition of $\xi(a)$, $\xi_1(a)$, $\xi_2(a)$ as a varies in the respective intervals.

40. Given the natural number n, let $\xi = \xi(a)$ be the unique positive root of the equation $x^n = a\mathrm{e}^{-x}$ $(a > 0)$. Determine the condition of ξ as a function of a; simplify the answer as much as possible. In particular, show that $(\operatorname{cond}\xi)(a) < 1/n$.

41. Let $\boldsymbol{f}(x_1, x_2) = x_1 + x_2$ and consider the algorithm A given as follows:

$$\boldsymbol{f}_A : \mathbb{R}^2(t,s) \to \mathbb{R}(t,s) \quad y_A = \mathrm{fl}(x_1 + x_2).$$

Estimate $\gamma(x_1, x_2) = (\operatorname{cond} A)(\boldsymbol{x})$, using any of the norms

$$\|\boldsymbol{x}\|_1 = |x_1| + |x_2|, \quad \|\boldsymbol{x}\|_2 = \sqrt{x_1^2 + x_2^2}, \quad \|\boldsymbol{x}\|_\infty = \max\ (|x_1|, |x_2|).$$

Discuss the answer in the light of the conditioning of $\boldsymbol{f}$.

42. This problem deals with the function $f(x) = \sqrt{1-x} - 1$, $-\infty < x < 1$.

(a) Compute the condition number $(\operatorname{cond} f)(x)$.
(b) Let A be the algorithm that evaluates $f(x)$ in floating-point arithmetic on a computer with machine precision eps, given an (error-free) floating-point number x. Let ε_1, ε_2, ε_3 be the relative errors due, respectively, to the subtraction in $1-x$, to taking the square root, and to the final subtraction of 1. Assume $|\varepsilon_i| \leq \text{eps}$ $(i = 1, 2, 3)$. Letting $f_A(x)$ be the value of $f(x)$ so computed, write $f_A(x) = f(x_A)$ and $x_A = x(1+\varepsilon_A)$. Express ε_A in terms

of x, ε_1, ε_2, ε_3 (neglecting terms of higher order in the ε_i). Then determine an upper bound for $|\varepsilon_A|$ in terms of x and eps and finally an estimate of $(\mathrm{cond}\, A)(x)$.

(c) Sketch a graph of $(\mathrm{cond}\, f)(x)$ (found in (a)) and a graph of the estimate of $(\mathrm{cond}\, A)(x)$ (found in (b)) as functions of x on $(-\infty, 1)$. Discuss your results.

43. Consider the function $f(x) = 1 - \mathrm{e}^{-x}$ on the interval $0 \le x \le 1$.

(a) Show that $(\mathrm{cond}\, f)(x) \le 1$ on [0,1].

(b) Let A be the algorithm that evaluates $f(x)$ for the machine number x in floating-point arithmetic (with machine precision eps). Assume that the exponential routine returns a correctly rounded answer. Estimate $(\mathrm{cond}\, A)(x)$ for $0 \le x \le 1$, neglecting terms of $O(\mathrm{eps}^2)$. {*Point of information*: $\ln(1+\varepsilon) = \varepsilon + O(\varepsilon^2)$, $\varepsilon \to 0$.}

(c) Plot $(\mathrm{cond}\, f)(x)$ and your estimate of $(\mathrm{cond}\, A)(x)$ as functions of x on [0,1]. Comment on the results.

44. (a) Suppose A is an algorithm that computes the (smooth) function $f(x)$ for a given machine number x, producing $f_A(x) = f(x)(1+\varepsilon_f)$, where $|\varepsilon_f| \le \varphi(x)\mathrm{eps}$ (eps $=$ machine precision). If $0 < (\mathrm{cond}\, f)(x) < \infty$, show that

$$(\mathrm{cond}\, A)(x) \le \frac{\varphi(x)}{(\mathrm{cond}\, f)(x)}$$

if second-order terms in eps are neglected. {*Hint*: set $f_A(x) = f(x_A)$, $x_A = x(1+\varepsilon_A)$, and expand in powers of ε_A, keeping only the first.}

(b) Apply the result of (a) to $f(x) = \frac{1-\cos x}{\sin x}$, $0 < x < \frac{1}{2}\pi$, when evaluated as shown. (You may assume that $\cos x$ and $\sin x$ are computed within a relative error of eps.) Discuss the answer.

(c) Do the same as (b), but for the (mathematically equivalent) function $f(x) = \frac{\sin x}{1+\cos x}$, $0 < x < \frac{1}{2}\pi$.

Machine Assignments

1. Let $x = 1 + \pi/10^6$. Compute the nth power of x for $n = 100{,}000, 200{,}000, \dots, 1{,}000{,}000$ once in single and once in double Matlab precision. Let the two results be p_n and $\mathrm{d}p_n$. Use the latter to determine the relative errors r_n of the former. Print $n, p_n, \mathrm{d}p_n, r_n, r_n/(n \cdot \mathrm{eps0})$, where eps0 is the single-precision `eps`. What should x^n be, approximately, when $n = 1{,}000{,}000$? Comment on the results.

2. Compute the derivative $\mathrm{d}y/\mathrm{d}x$ of the exponential function $y = \mathrm{e}^x$ at $x = 0$ from the difference quotients $d(h) = (\mathrm{e}^h - 1)/h$ with decreasing h. Use

(a) $h = h1 := 2^{-i}, i = 5:5:50$;

(b) $h = h2 := (2.2)^{-i}, i = 5:5:50$.

Print the quantities $i, h1, h2, d1 := d(h1), d2 := d(h2)$, the first and two last ones in `f`-format, the others in `e`-format. Explain what you observe.

3. Consider the following procedure for determining the limit $\lim_{h\to 0}(\mathrm{e}^h - 1)/h$ on a computer. Let

$$d_n = \mathrm{fl}\left(\frac{\mathrm{e}^{2^{-n}} - 1}{2^{-n}}\right) \quad \text{for } n = 0, 1, 2, \ldots$$

and accept as the machine limit the first value satisfying $d_n = d_{n-1}$ $(n \geq 1)$.

(a) Write and run a Matlab routine implementing the procedure.

(b) In $\mathbb{R}(t, s)$-floating-point arithmetic, with rounding by chopping, for what value of n will the correct limit be reached, assuming no underflow (of 2^{-n}) occurs? {*Hint*: use $\mathrm{e}^h = 1 + h + \frac{1}{2}h^2 + \cdots$.} Compare with the experiment made in (a).

(c) On what kind of computer (i.e., under what conditions on s and t) will underflow occur before the limit is reached?

4. Euler's constant $\gamma = 0.57721566490153286\ldots$ is defined as the limit

$$\gamma = \lim_{n\to\infty} \gamma_n, \quad \text{where } \gamma_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} - \ln n.$$

Assuming that $\gamma - \gamma_n \sim cn^{-d}$, $n \to \infty$, for some constants c and $d > 0$, try to determine c and d experimentally on the computer.

5. Letting $\Delta u_n = u_{n+1} - u_n$, one has the easy formula

$$\sum_{n=1}^{N} \Delta u_n = u_{N+1} - u_1.$$

With $u_n = \ln(1 + n)$, compute each side (as it stands) for $N = 1{,}000 : 1{,}000 : 10{,}000$, the left-hand side in Matlab single precision and the right-hand side in double precision. Print the relative discrepancy of the two results. Repeat with $\sum_{n=1}^{N} u_n$: compute the sum in single and double precision and compare the results. Try to explain what you observe.

6. (a) Write a program to compute

$$S_N = \sum_{n=1}^{N}\left[\frac{1}{n} - \frac{1}{n+1}\right] = \sum_{n=1}^{N}\frac{1}{n(n+1)},$$

once using the first summation and once using the (mathematically equivalent) second summation. For $N = 10^k$, $k = 1:7$, print the respective absolute errors. Comment on the results.

(b) Write a program to compute

$$p_N = \prod_{n=1}^{N} \frac{n}{n+1}.$$

For the same values of N as in part (a), print the relative errors. Comment on the results.

7. (a) Prove: *based on best possible relative error bounds, the floating-point addition* $\mathrm{fl}(\mathrm{fl}(x+y)+z)$ *is more accurate than* $\mathrm{fl}(x+\mathrm{fl}(y+z))$ *if and only if* $|x+y| < |y+z|$. As applications, formulate addition rules in the cases

   (a1) $0 < x < y < z$;
   (a2) $x > 0, y < 0, z > 0$;
   (a3) $x < 0, y > 0, z < 0$.

   (b) Consider the nth partial sums of the series defining the zeta function $\zeta(s)$, resp., eta function $\eta(s)$,

$$z_n = \sum_{k=1}^{n} \frac{1}{k^s}, \quad e_n = \sum_{k=1}^{n} (-1)^{k-1} \frac{1}{k^s}.$$

   For $s = 2, 11/3, 5, 7.2, 10$ and $n = 50, 100, 200, 500, 1000$, compute these sums in Matlab single precision, once in forward direction and once in backward direction, and compare the results with Matlab double-precision evaluations. Interpret the results in the light of your answers to part (a), especially (a2) and (a3).

8. Let $n = 10^6$ and

$$s = 10^{11} n + \sum_{k=1}^{n} \ln k.$$

   (a) Determine s analytically and evaluate to 16 decimal digits.
   (b) The following Matlab program computes s in three different (but mathematically equivalent) ways:

```
%MAI_8B
%
n=10^6; s0=10^11*n;
s1=s0;
for k=1:n
  s1=s1+log(k);
end
s2=0;
for k=1:n
   s2=s2+log(k);
```

```
end
s2=s2+s0;
i=1:n;
s3=s0+sum(log(i));
[s1 s2 s3]'
```

Run the program and discuss the results.

9. Write a Matlab program that computes the Euclidean condition number of the Hilbert matrix $\boldsymbol{H}_n$ following the prescription given in footnote 3 of the text.

   (a) The inverse of the Hilbert matrix $\boldsymbol{H}_n$ has elements

   $$\left(\boldsymbol{H}_n^{-1}\right)_{ij} = (-1)^{i+j}(i+j-1)\binom{n+i-1}{n-j}\binom{n+j-1}{n-i}\binom{i+j-2}{i-1}^2$$

   (cf. Note 3 to Sect. 1.3.2). Simplify the expression to avoid factorials of large numbers. {*Hint*: express all binomial coefficients in terms of factorials and simplify.}

   (b) Implement in Matlab the formula obtained in (a) and reproduce Table 1.1 of the text.

10. The (symmetrically truncated) cardinal series of a function f is defined by

    $$C_N(f,h)(x) = \sum_{k=-N}^{N} f(kh)\,\mathrm{sinc}\left(\frac{x-kh}{h}\right),$$

    where $h > 0$ is the spacing of the data and the sinc function is defined by

    $$\mathrm{sinc}(u) = \begin{cases} \dfrac{\sin(\pi u)}{\pi u} & \text{if } u \neq 0, \\ 1 & \text{if } u = 0. \end{cases}$$

    Under appropriate conditions, $C_N(f,h)(x)$ approximates $f(x)$ on $[-Nh, Nh]$.

    (a) Show that

    $$C_N(f,h)(x) = \frac{h}{\pi}\sin\frac{\pi x}{h}\sum_{k=-N}^{N}\frac{(-1)^k}{x-kh}\,f(kh).$$

    Since this requires the evaluation of only one value of the sine function, it provides a more efficient way to evaluate the cardinal series than the original definition.

(b) While the form of C_N given in (a) may be more efficient, it is numerically unstable when x is near one of the abscissae kh. Why?

(c) Find a way to stabilize the formula in (a). {*Hint*: introduce the integer k_0 and the real number t such that $x = (k_0 + t)h$, $|t| \le \frac{1}{2}$.}

(d) Write a program to compute $C_N(f, h)(x)$ according to the formula in (a) and the one developed in (c) for $N = 100$, $h = 0.1$, $f(x) = x\,\exp(-x^2)$, and $x = 0.55$, $x = 0.5 + 10^{-8}$, $x = 0.5 + 10^{-15}$. Print $C_N(f, h)(x)$, $f(x)$, and the error $|C_N(f, h)(x) - f(x)|$ in either case. Compare the results.

11. In the theory of Fourier series, the numbers

$$\lambda_n = \frac{1}{2n+1} + \frac{2}{\pi}\sum_{k=1}^{n}\frac{1}{k}\tan\frac{k\pi}{2n+1}, \quad n = 1, 2, 3, \dots\,,$$

known as *Lebesgue constants*, are of some importance.

(a) Show that the terms in the sum increase monotonically with k. How do the terms for k near n behave when n is large?

(b) Compute λ_n for $n = 1, 10, 10^2, \dots, 10^5$ in Matlab single and double precision and compare the results. Do the same with n replaced by $\lceil n/2 \rceil$. Explain what you observe.

12. Sum the series

$$\text{(a)}\ \sum_{n=0}^{\infty}(-1)^n/n!^2, \quad \text{(b)}\ \sum_{n=0}^{\infty}1/n!^2$$

until there is no more change in the partial sums to within the machine precision. Generate the terms recursively. Print the number of terms required and the value of the sum. (Answers in terms of Bessel functions: (a) $J_0(2)$; cf. Abramowitz and Stegun [1964, (9.1.18)] or Olver et al. [2010, (10.9.1)] and (b) $I_0(2)$; cf. Abramowitz and Stegun [1964, (9.6.16)] or Olver et al. [2010, (10.32.1)].)

13. (P.J. Davis, 1993) Consider the series $\sum_{k=1}^{\infty}\frac{1}{k^{3/2}+k^{1/2}}$. Try to compute the sum to three correct decimal digits.

14. We know from calculus that

$$\lim_{n\to\infty}\left(1 + \frac{1}{n}\right)^n = e.$$

What is the "machine limit"? Explain.

15. Let $f(x) = (n+1)x - 1$. The iteration

$$x_k = f(x_{k-1}), \quad k = 1, 2, \dots, K; \quad x_0 = 1/n,$$

in exact arithmetic converges to the fixed point $1/n$ in one step (Why?). What happens in machine arithmetic? Run a program with $n = 1 : 5$ and $K = 10 : 10 : 50$ and explain quantitatively what you observe.

16. Compute the integral $\int_0^1 \mathrm{e}^x \mathrm{d}x$ from Riemann sums with n equal subintervals, evaluating the integrand at the midpoint of each. Print the Riemann sums for $n = 5{,}000 : 5{,}000 : 100{,}000$ (to 15 decimal digits after the decimal point), together with absolute errors. Comment on the results.
17. Let $y_n = \int_0^1 t^n \mathrm{e}^{-t} \mathrm{d}t$, $n = 0, 1, 2, \ldots$.

    (a) Use integration by parts to obtain a recurrence formula relating y_k to y_{k-1} for $k = 1, 2, 3, \ldots$, and determine the starting value y_0.

    (b) Write and run a Matlab program that generates $y_0, y_1, \ldots, y_{20}$, using the recurrence of (a), and print the results to 15 decimal digits after the decimal point. Explain in detail (quantitatively, using mathematical analysis) what is happening.

    (c) Use the recursion of (a) in reverse order, starting (arbitrarily) with $y_N = 0$. Place into five consecutive columns of a (21×5) matrix Y the values $y_0^{(N)}, y_1^{(N)}, \ldots, y_{20}^{(N)}$ thus obtained for $N = 22, 24, 26, 28, 30$. Determine how much consecutive columns of Y differ from one another by printing

    $$e_i = \max |\, (Y(:, i+1) - Y(:, i))\, ./\, Y(:, i+1)|, \quad i = 1, 2, 3, 4.$$

    Print the last column $Y(:, 5)$ of Y and explain why this represents accurately the column vector of the desired quantities $y_0, y_1, \ldots, y_{20}$.

Selected Solutions to Exercises

14. We may assume $x > 0$, since otherwise we could multiply z by -1 and the result by $-\mathrm{i}$.

    In the first expression for $\sqrt{z}$ there will be a large cancellation error in the imaginary part when $|y|$ is very small, whereas in the second expression all arithmetic operations are benign. Illustration: $z = 4.5 + 0.025\mathrm{i}$ (in eight significant digits)

    $$r = 4.5000694,$$
    $$v = \left(\frac{r - x}{2}\right)^{1/2} = 5.8906706 \times 10^{-3},$$
    $$v = \frac{y}{2\left(\frac{r+x}{2}\right)^{1/2}} = 5.8925338 \times 10^{-3}.$$

    The last five digits in the first evaluation of v are in error!
21. (a) We have

    $$p_1 = x_1,$$
    $$p_k = x_k p_{k-1}(1 + \varepsilon_k), \ \ |\varepsilon_k| \le \text{eps}, \ \ k = 2, 3, \ldots, n.$$

Therefore,

$$
\begin{aligned}
p_2 &= x_1 x_2 (1 + \varepsilon_2), \\
p_3 &= x_1 x_2 x_3 (1 + \varepsilon_2)(1 + \varepsilon_3), \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
p_n &= x_1 x_2 \cdots x_n (1 + \varepsilon_2)(1 + \varepsilon_3) \cdots (1 + \varepsilon_n),
\end{aligned}
$$

so that

$$\frac{p_n - x_1 x_2 \cdots x_n}{x_1 x_2 \cdots x_n} = (1 + \varepsilon_2)(1 + \varepsilon_3) \cdots (1 + \varepsilon_n) - 1 =: E.$$

If $E \geq 0$, then $|E| \leq (1 + \text{eps})^{n-1} - 1$; otherwise, $|E| \leq 1 - (1 - \text{eps})^{n-1}$. Since the first bound is larger than the second, we get

$$|E| \leq (1 + \text{eps})^{n-1} - 1.$$

(b) Using the binomial theorem, one has

$$
\begin{aligned}
(1 + \text{eps})^r - 1 &= \left(1 + \binom{r}{1} \text{eps} + \binom{r}{2} \text{eps}^2 + \cdots + \text{eps}^r\right) - 1 \\
&= r \cdot \text{eps} \left\{ 1 + \frac{r-1}{2} \text{eps} + \frac{(r-1)(r-2)}{3!} \text{eps}^2 + \cdots \right. \\
&\qquad \left. + \frac{(r-1)(r-2)\cdots 1}{r!} \text{eps}^{r-1} \right\}.
\end{aligned}
$$

Since $r \cdot \text{eps} < \dfrac{1}{10}$, one has also

$$(r - k)\,\text{eps} < \frac{1}{10}, \ \ k = 1, 2, \ldots, r - 1,$$

and so

$$
\begin{aligned}
(1 + \text{eps})^r - 1 &< r \cdot \text{eps} \left\{ 1 + \frac{1}{2!} 10^{-1} + \frac{1}{3!} 10^{-2} + \cdots + \frac{1}{r!} 10^{-(r-1)} \right\} \\
&< r \cdot \text{eps} \cdot 10 \left\{ \frac{1}{1!} 10^{-1} + \frac{1}{2!} 10^{-2} + \cdots \right\} \\
&= r \cdot \text{eps} \cdot 10 \{ \text{e}^{10^{-1}} - 1 \} = 1.051709\ldots r \cdot \text{eps} \\
&< 1.06 \cdot r \cdot \text{eps}.
\end{aligned}
$$

Hence, if $(n-1)\text{eps} < 1/10$, the result in (a) can be simplified to

$$|E| \le 1.06(n-1)\text{eps}.$$

34. We have

$$\begin{aligned}
\cos\alpha(\rho) &= \frac{\boldsymbol{q}^{\mathrm{T}}\boldsymbol{b}(\rho)}{\|\boldsymbol{q}\|\cdot\|\boldsymbol{b}(\rho)\|} = \frac{\boldsymbol{q}^{\mathrm{T}}(\boldsymbol{a}-\rho\boldsymbol{q})}{[(\boldsymbol{a}-\rho\boldsymbol{q})^{\mathrm{T}}(\boldsymbol{a}-\rho\boldsymbol{q})]^{1/2}} \\
&= \frac{\rho_0-\rho}{(1-2\rho_0\rho+\rho^2)^{1/2}} =: R(\rho), \\
\alpha(\rho) &= \cos^{-1} R(\rho), \quad R(\rho_0) = 0 \ \text{ if } \ |\rho_0| < 1, \\
\alpha'(\rho) &= \frac{-1}{\sqrt{1-R^2(\rho)}} R'(\rho) \\
&= \frac{-1}{\sqrt{1-R^2(\rho)}} \\
&\quad \times \frac{-(1-2\rho_0\rho+\rho^2)^{1/2} - (\rho_0-\rho)[(1-2\rho_0\rho+\rho^2)^{1/2}]'}{(1-2\rho_0\rho+\rho^2)}.
\end{aligned}$$

For $\rho = \rho_0$, therefore, assuming $|\rho_0| < 1$, we get

$$\begin{aligned}
\alpha'(\rho_0) &= \frac{(1-\rho_0^2)^{1/2}}{1-\rho_0^2} = \frac{1}{(1-\rho_0^2)^{1/2}}, \\
(\operatorname{cond}\alpha)(\rho_0) &= \frac{|\rho_0|\frac{1}{(1-\rho_0^2)^{1/2}}}{\frac{1}{2}\pi} = \frac{2}{\pi}\frac{|\rho_0|}{(1-\rho_0^2)^{1/2}}.
\end{aligned}$$

If $\rho_0 = 0$, i.e., $\boldsymbol{a}$ is already orthogonal to $\boldsymbol{q}$, hence $\boldsymbol{b} = \boldsymbol{a}$, then $(\operatorname{cond}\alpha)(\rho_0) = 0$, as expected. If $|\rho_0| \uparrow 1$, then $(\operatorname{cond}\alpha)(\rho_0) \uparrow \infty$, since in the limit, $\boldsymbol{a}$ is parallel to $\boldsymbol{q}$ and cannot be orthogonalized. In practice, if $|\rho_0|$ is close to 1, the problem of ill-conditioning can be overcome by a single, or possibly repeated, reorthogonalization.

44. (a) Following the *Hint*, we have

$$\begin{aligned}
f_A(x) &= f(x)(1+\varepsilon_f) = f(x_A) \\
&= f(x(1+\varepsilon_A)) = f(x + x\varepsilon_A) \\
&= f(x) + x\varepsilon_A f'(x) + O(\varepsilon_A^2).
\end{aligned}$$

Neglecting the $O(\varepsilon_A^2)$ term, one gets

$$x\varepsilon_A f'(x) = f(x)\varepsilon_f,$$

hence

$$\left|\frac{x_A - x}{x}\right| = |\varepsilon_A| = \left|\frac{f(x)}{xf'(x)}\right| |\varepsilon_f| \le \frac{\varphi(x)}{(\operatorname{cond} f)(x)} \text{ eps},$$

which proves the assertion.

(b) One easily computes

$$(\operatorname{cond} f)(x) = \frac{x}{\sin x}, \quad 0 < x < \frac{\pi}{2}.$$

Furthermore,

$$f_A(x) = \frac{(1 - (\cos x)(1 + \varepsilon_1))(1 + \varepsilon_2)}{(\sin x)(1 + \varepsilon_3)} (1 + \varepsilon_4), \quad |\varepsilon_i| \le \text{eps},$$

where ε_1, ε_2, ε_3, ε_4 are the relative errors committed, respectively, in evaluating the cosine function, the difference in the numerator, the sine function, and the quotient. Neglecting terms of $O(\varepsilon_i^2)$, one obtains by a simple computation that

$$f_A(x) = \frac{1 - \cos x}{\sin x}\left\{1 + \varepsilon_2 + \varepsilon_4 - \varepsilon_3 - \varepsilon_1 \frac{\cos x}{1 - \cos x}\right\},$$

that is,

$$|\varepsilon_f| = \left|\varepsilon_2 + \varepsilon_4 - \varepsilon_3 - \varepsilon_1 \frac{\cos x}{1 - \cos x}\right| \le \left(3 + \frac{\cos x}{1 - \cos x}\right) \text{eps}.$$

Therefore,

$$\varphi(x) = 3 + \frac{\cos x}{1 - \cos x},$$

and one gets

$$(\operatorname{cond} A)(x) \le \frac{\sin x}{x}\left(3 + \frac{\cos x}{1 - \cos x}\right), \quad 0 < x < \frac{\pi}{2}.$$

Obviously, $(\operatorname{cond} A)(x) \to \infty$ as $x \to 0$, whereas $(\operatorname{cond} A)(x) \to 6/\pi$ as $x \to \pi/2$. The algorithm is ill-conditioned near $x = 0$ (cancellation error), but well conditioned near $\pi/2$. The function itself is quite well conditioned,

$$1 \le (\operatorname{cond} f)(x) \le \frac{\pi}{2}.$$

(c) In this case (the condition of f being of course the same),

$$f_A(x) = \frac{(\sin x)(1+\varepsilon_1)}{(1+(\cos x)(1+\varepsilon_2))(1+\varepsilon_3)}\,(1+\varepsilon_4),\ \ |\varepsilon_i| \le \text{ eps},$$

giving

$$f_A(x) = \frac{\sin x}{1+\cos x}\left\{1+\varepsilon_1-\varepsilon_3+\varepsilon_4-\varepsilon_2\,\frac{\cos x}{1+\cos x}\right\},$$

that is,

$$|\varepsilon_f| = \left|\varepsilon_1-\varepsilon_3+\varepsilon_4-\varepsilon_2\,\frac{\cos x}{1+\cos x}\right| \le \left(3+\frac{\cos x}{1+\cos x}\right)\text{eps},$$

and

$$(\text{cond}\,A)(x) \le \frac{\sin x}{x}\left(3+\frac{\cos x}{1+\cos x}\right).$$

Now, A is entirely well conditioned,

$$\frac{6}{\pi} \le (\text{cond}\,A)(x) \le \frac{7}{2},\quad 0 < x < \frac{\pi}{2}.$$

Selected Solutions to Machine Assignments

7. (a) For arbitrary real x, y, z, the first addition can be written as

$$\begin{aligned}\text{fl}(\text{fl}(x+y)+z) &= ((x+y)(1+\varepsilon_1)+z)(1+\varepsilon_2)\\ &\approx x+y+z+(x+y)\varepsilon_1+(x+y+z)\varepsilon_2\\ &= (x+y+z)\left(1+\frac{x+y}{x+y+z}\,\varepsilon_1+\varepsilon_2\right),\end{aligned}$$

where the ε_i are bounded in absolute value by `eps`. The best bound for the relative error is

$$|\text{rel.err.}| \le \left(\frac{|x+y|}{|x+y+z|}+1\right)\text{eps}.$$

Likewise, for the second addition, there holds (interchange x and z)

$$|\text{rel.err.}| \le \left(\frac{|y+z|}{|x+y+z|} + 1 \right) \text{eps}.$$

Based on these two error bounds, the first addition is more accurate than the second if and only if $|x+y| < |y+z|$, as claimed.

Examples

(a1) $0 < x < y < z$. Here,

$$|x+y| = x+y < y+z = |y+z|.$$

Thus, *addition in increasing order* is more accurate.

(a2) $x > 0$, $y < 0$, $z > 0$. Here,

$$|x+y| = |x - |y||\,, \quad |y+z| = |z - |y||\,,$$

and *adding to the negative number y the positive number closer to $|y|$ first* is more accurate.

(a3) $x < 0$, $y > 0$, $z < 0$. Here,

$$|x+y| = |-|x| + y| = ||x| - y|\,,$$
$$|y+z| = |y - |z|| = ||z| - y|\,,$$

and *adding to the positive number y the negative number first whose modulus is closer to y* is more accurate.

(b) PROGRAM

```
%MAI_7B
%
f0='%6.4f %8.1e %9.2e %9.2e %9.2e %9.2e\n';
disp('                    zeta            eta')
disp('   s     n     forw   backw    forw   backw')
for s=[2 11/3 5 7.2 10]
  for n=[50 100 200 500 1000]
    k=1:n;
    z=sum(1./k.^s);
    e=sum((-1).^(k-1)./k.^s);
    zf=single(0); ef=single(0);
    for kf=1:n
      zf=zf+single(1/kf^s);
```

```
         ef=ef+single((-1)^(kf-1)/kf^s);
       end
       zf0=zf; ef0=ef;
       zb=single(0); eb=single(0);
       for kb=n:-1:1
         zb=zb+single(1/kb^s);
         eb=eb+single((-1)^(kb-1)/kb^s);
       end
       zb0=zb; eb0=eb;
       errzf=abs((zf0-z)/z);
       errzb=abs((zb0-z)/z);
       erref=abs((ef0-e)/e);
       erreb=abs((eb0-e)/e);
       fprintf(f0,s,n,errzf,errzb,erref,erreb)
     end
     fprintf('\n')
   end

    OUTPUT

>> MAI_7B
                       zeta                 eta
   s       n       forw     backw      forw     backw
2.0000  5.0e+01  1.14e-07  3.30e-08  5.16e-08  2.09e-08
2.0000  1.0e+02  7.11e-08  1.82e-09  4.35e-08  4.35e-08
2.0000  2.0e+02  9.34e-08  5.20e-08  1.16e-07  2.94e-08
2.0000  5.0e+02  4.52e-08  4.52e-08  3.92e-07  3.01e-08
2.0000  1.0e+03  1.70e-07  4.77e-08  3.85e-07  2.23e-08

3.6667  5.0e+01  2.48e-07  3.30e-08  2.84e-08  3.54e-08
3.6667  1.0e+02  1.26e-07  1.82e-08  5.97e-08  4.07e-09
3.6667  2.0e+02  1.43e-06  3.15e-08  8.21e-08  1.84e-08
3.6667  5.0e+02  1.65e-06  4.06e-08  8.40e-08  2.02e-08
3.6667  1.0e+03  1.67e-06  4.88e-08  8.41e-08  2.03e-08

5.0000  5.0e+01  2.46e-07  1.61e-08  4.04e-08  2.09e-08
5.0000  1.0e+02  2.81e-07  5.08e-08  3.89e-08  2.24e-08
5.0000  2.0e+02  2.83e-07  5.30e-08  3.88e-08  2.25e-08
5.0000  5.0e+02  2.83e-07  5.31e-08  3.88e-08  2.25e-08
5.0000  1.0e+03  2.83e-07  5.31e-08  3.88e-08  2.25e-08

7.2000  5.0e+01  7.20e-09  7.20e-09  5.45e-08  5.52e-09
7.2000  1.0e+02  7.20e-09  7.20e-09  5.45e-08  5.52e-09
7.2000  2.0e+02  7.20e-09  7.20e-09  5.45e-08  5.52e-09
7.2000  5.0e+02  7.20e-09  7.20e-09  5.45e-08  5.52e-09
7.2000  1.0e+03  7.20e-09  7.20e-09  5.45e-08  5.52e-09

10.0000  5.0e+01  1.20e-08  1.20e-08  2.32e-08  2.32e-08
10.0000  1.0e+02  1.20e-08  1.20e-08  2.32e-08  2.32e-08
```

```
10.0000  2.0e+02  1.20e-08  1.20e-08  2.32e-08  2.32e-08
10.0000  5.0e+02  1.20e-08  1.20e-08  2.32e-08  2.32e-08
10.0000  1.0e+03  1.20e-08  1.20e-08  2.32e-08  2.32e-08
>>
```

Interpretation

Rather consistently, backward summation gives more accurate results than forward summation, by as much as two decimal orders (for $s = 11/3$ and $n = 200$ in the case of $\zeta(s)$). For large s (for example $s = 10$) there is no noticeable difference in accuracy. All this (and more) is consistent with the answers given in part (a). Indeed, the series for the zeta function has terms that are all positive and strictly decreasing, so that by (a1) summation in increasing order of the terms, i.e., backward summation, is more accurate. In the case of the eta series, consider

$$x = \frac{1}{k^s},\quad y = -\frac{1}{(k+1)^s},\quad z = \frac{1}{(k+2)^s}.$$

Then

$$|x+y| = \frac{1}{k^s} - \frac{1}{(k+1)^s} = \frac{1}{k^s}\left(1 - \left(\frac{k}{k+1}\right)^s\right),$$

$$|y+z| = \frac{1}{(k+1)^s} - \frac{1}{(k+2)^s} = \frac{1}{(k+1)^s}\left(1 - \left(\frac{k+1}{k+2}\right)^s\right).$$

Since the function $x/(x+1)$ for $x > 0$ increases monotonically, it follows that $|x+y| > |y+z|$ for all $k > 0$, and backward summation is more accurate than forward summation. Moreover,

$$|x+y| = \frac{1}{k^s}\left(1 - \left(1 + \frac{1}{k}\right)^{-s}\right) \sim \frac{s}{k^{s+1}},\quad k \to \infty,$$

$$|y+z| = \frac{1}{k^s}\left(1 + \frac{1}{k}\right)^{-s}\left(1 - \left(1 + \frac{1}{k}\right)^{s}\left(1 + \frac{2}{k}\right)^{-s}\right) \sim \frac{s}{k^{s+1}},\quad k \to \infty,$$

so that the improvement $\frac{|y+z|}{|x+y|}$ of backward over forward summation disappears asymptotically as $k \to \infty$. Also, for large s, the improvement is relatively small, even for only moderately large k.

The same discussion holds for

$$x = -\frac{1}{k^s},\quad y = \frac{1}{(k+1)^s},\quad z = -\frac{1}{(k+2)^s}.$$

11. (a) Let $x = \frac{k\pi}{2n+1}$, so that $0 < x < \frac{\pi}{2}$ if $1 \leq k \leq n$. Then, up to a positive constant factor, the general term of the sum is

$$f(x) = \frac{1}{x}\,\tan x.$$

We show that f increases monotonically: we have

$$[xf(x)]' = \frac{1}{\cos^2 x}\,,$$

hence

$$\begin{aligned} xf'(x) &= \frac{1}{\cos^2 x} - f(x) = \frac{1}{\cos^2 x} - \frac{\sin x}{x\cos x} \\ &= \frac{1}{\cos^2 x}\left\{1 - \frac{1}{x}\cdot\sin x\ \cos x\right\} \\ &= \frac{1}{\cos^2 x}\left\{1 - \frac{\sin 2x}{2x}\right\} > 0. \end{aligned}$$

Thus, the terms in the sum increase monotonically. For n very large, say $n = 10^5$, most terms of the sum are negligibly small except for a few very near n, where they sharply increase to the maximum value $\approx \frac{4}{\pi}$. This can be seen by letting $k = n - r$ for some fixed (small) integer r and large n. Then

$$\frac{n-r}{2n+1} = \frac{1}{2} - \frac{2r+1}{2(2n+1)},$$

and, as $n \to \infty$,

$$\tan\frac{(n-r)\pi}{2n+1} = \tan\left(\frac{\pi}{2} - \frac{\pi}{2}\,\frac{2r+1}{2n+1}\right) = \frac{\cos\left(\frac{\pi}{2}\frac{2r+1}{2n+1}\right)}{\sin\left(\frac{\pi}{2}\frac{2r+1}{2n+1}\right)} \sim \frac{4}{\pi}\,\frac{n}{2r+1}.$$

Therefore,

$$\frac{1}{n-r}\tan\frac{(n-r)\pi}{2n+1} \sim \frac{4}{\pi}\,\frac{1}{2r+1} \quad \text{as } n \to \infty.$$

(b) PROGRAM

```
 %MAI_11B
 %
 f0='%10.0f %12.8f %19.16f %12.4e\n';
 disp('    n      Lebesgue  Lebesgue double      diff')
% disp('    n       truncated single and double Lebesgue')
 for n=[1 10 100 1000 10000 100000]
   den0=single(1/(2*n+1));
```

```
  den=1/(2*n+1);
  k=1:n;
 % k=1:ceil(n/2);
  s0=sum(single(tan(k*pi*den0)./k));
  s=sum(tan(k*pi*den)./k);
  s0=den0+single(2*s0/pi);
  s=den+2*s/pi;
  diff=abs(s-s0);
  fprintf(f0,n,subs(s0),s,diff)
end
```

```
 OUTPUT

>> MAI_11B
      n        Lebesgue    Lebesgue double        diff
         1   1.43599117  1.4359911241769170   4.3845e-08
        10   2.22335672  2.2233569241536819   2.0037e-07
       100   3.13877344  3.1387800926548395   6.6513e-06
      1000   4.07023430  4.0701636043524356   7.0694e-05
     10000   5.00332785  5.0031838616314577   1.4398e-04
    100000   5.92677021  5.9363682125234796   9.5980e-03
>>
```

Comments

Because of the behavior of the terms in the sum, when n is large, the accuracy of the sum is largely determined by the accuracy of the terms very near to n. But there, the argument of the tangent is very close to $\pi/2$. Since

$$(\text{cond}\ \tan)(x) = \frac{x(1+\tan^2 x)}{\tan x}, \quad 0 < x < \pi/2,$$

the tangent is very ill-conditioned for x near $\pi/2$. In fact, for $\varepsilon > 0$ very small,

$$(\text{cond}\ \tan)\left(\frac{\pi}{2}-\varepsilon\right) \sim \frac{\pi}{2}\tan\left(\frac{\pi}{2}-\varepsilon\right) = \frac{\pi}{2}\,\frac{\cos\varepsilon}{\sin\varepsilon} \sim \frac{\pi}{2\varepsilon}.$$

Since $k = n$ corresponds to $\frac{\pi}{2} - \varepsilon = \frac{n}{2n+1}\,\pi$, that is, $\varepsilon = \frac{\pi}{2(2n+1)} \sim \frac{\pi}{4n}$, one has

$$(\text{cond}\ \tan)\left(\frac{\pi}{2}-\varepsilon\right) \sim \frac{\pi}{2\pi/(4n)} = 2n, \quad n \to \infty.$$

So, for $n = 10^5$, for example, we must expect a loss of about five decimal digits. This is confirmed by the numerical results shown above.

The inaccuracy observed cannot be ascribed merely to the large volume of computation. In fact, if we extended the sum only to, say, $k = n/2$, we would escape the ill-conditioning of the tangent and, even for $n = 10^5$, would get more accurate answers. This is shown by the numerical results below.

```
>> MAI_11B
      n         truncated single and double Lebesgue
         1   1.43599117  1.4359911241769170   4.3845e-08
        10   0.57060230  0.5706023117647347   1.3982e-08
       100   0.54363489  0.5436349730685069   8.1558e-08
      1000   0.54078776  0.5407873971420362   3.5930e-07
     10000   0.54050153  0.5405010907553436   4.4418e-07
    100000   0.54047358  0.5404724445910691   1.1358e-06
>>
```

Chapter 2
Approximation and Interpolation

The present chapter is basically concerned with the approximation of functions. The functions in question may be functions defined on a continuum – typically a finite interval – or functions defined only on a finite set of points. The first instance arises, for example, in the context of special functions (elementary or transcendental) that one wishes to evaluate as a part of a subroutine. Since any such evaluation must be reduced to a finite number of arithmetic operations, we must ultimately approximate the function by means of a polynomial or a rational function. The second instance is frequently encountered in the physical sciences when measurements are taken of a certain physical quantity as a function of some other physical quantity (such as time). In either case one wants to approximate the given function "as well as possible" in terms of other simpler functions.

The general scheme of approximation can be described as follows. We are given the function f to be approximated, along with a class Φ of "approximating functions" φ and a "norm" $\|\cdot\|$ measuring the overall magnitude of functions. We are looking for an approximation $\hat{\varphi} \in \Phi$ of f such that

$$\|f - \hat{\varphi}\| \le \|f - \varphi\| \ \text{ for all } \ \varphi \in \Phi. \tag{2.1}$$

The function $\hat{\varphi}$ is called the *best approximation* to f from the class Φ, relative to the norm $\|\cdot\|$.

The class Φ is called a (real) *linear space* if with any two functions φ_1, $\varphi_2 \in \Phi$ it also contains $\varphi_1 + \varphi_2$ and $c\varphi_1$ for any $c \in \mathbb{R}$, hence also any (finite) linear combination of functions $\varphi_i \in \Phi$. Given n "basis functions" $\pi_j \in \Phi$, $j = 1, 2, \dots, n$, we can define a linear space of finite dimension n by

$$\Phi = \Phi_n = \left\{ \varphi : \ \varphi(t) = \sum_{j=1}^{n} c_j \pi_j(t), \ c_j \in \mathbb{R} \right\}. \tag{2.2}$$

Examples of linear spaces Φ. 1. $\Phi = \mathbb{P}_m$: polynomials of degree $\le m$. A basis for $\mathbb{P}_m$ is, for example, $\pi_j(t) = t^{j-1}$, $j = 1, 2, \dots, m+1$, so that $n = m + 1$. Polynomials are the most frequently used "general-purpose" approximants for

W. Gautschi, *Numerical Analysis*, DOI 10.1007/978-0-8176-8259-0_2,

dealing with functions on bounded domains (finite intervals or finite sets of points). One reason is *Weierstrass's theorem*, which states that any continuous function can be approximated on a finite interval as closely as one wishes by a polynomial of sufficiently high degree.

2. $\Phi = \mathbb{S}_m^k(\Delta)$: (polynomial) spline functions of degree m and smoothness class k on the subdivision

$$\Delta: \quad a = t_1 < t_2 < t_3 < \cdots < t_{N-1} < t_N = b$$

of the interval $[a, b]$. These are piecewise polynomials of degree $\le m$ pieced together at the "joints" $t_2, \ldots, t_{N-1}$ in such a way that all derivatives up to and including the kth are continuous on the whole interval $[a, b]$, including the joints:

$$\mathbb{S}_m^k(\Delta) = \{s \in C^k[a,b]: \ s\big|_{[t_i, t_{i+1}]} \in \mathbb{P}_m, \ i = 1, 2, \ldots, N-1\}.$$

We assume here $0 \le k < m$; otherwise, we are back to polynomials $\mathbb{P}_m$ (see Ex. 68). We set $k = -1$ if we allow discontinuities at the joints. The dimension of $\mathbb{S}_m^k(\Delta)$ is $n = (m-k) \cdot (N-2) + m + 1$ (see Ex. 71), but to find a basis is a nontrivial task; for $m = 1$, see Sect. 2.3.2.

3. $\Phi = \mathbb{T}_m\ [0, 2\pi]$: trigonometric polynomials of degree $\le m$ on $[0,\ 2\pi]$. These are linear combinations of the basic harmonics up to and including the mth one, that is,

$$\pi_k(t) = \cos\ (k-1)t, \ \ k = 1, 2, \ldots, m+1;$$

$$\pi_{m+1+k}(t) = \sin kt, \ \ k = 1, 2, \ldots, m,$$

where now $n = 2m + 1$. Such approximants are a natural choice when the function f to be approximated is periodic with period 2π. (If f has period p, one makes a preliminary change of variables $t \mapsto t \cdot p/2\pi$.)

4. $\Phi = \mathbb{E}_n$: exponential sums. For given distinct $\alpha_j > 0$, one takes $\pi_j(t) = \mathrm{e}^{-\alpha_j t}$, $j = 1, 2, \ldots, n$. Exponential sums are often employed on the half-infinite interval $\mathbb{R}_+$: $0 \le t < \infty$, especially if one knows that f decays exponentially as $t \to \infty$.

Note that the important class of rational functions,

$$\Phi = \mathbb{R}_{r,s} = \{\varphi: \ \ \varphi = p/q, \ p \in \mathbb{P}_r, \ q \in \mathbb{P}_s\},$$

is *not* a linear space. (Why not?)

Possible choices of norm – both for continuous and discrete functions – and the type of approximation they generate are summarized in Table 2.1. The continuous case involves an interval $[a, b]$ and a "weight function" $w(t)$ (possibly $w(t) \equiv 1$) defined on $[a, b]$ and positive except for isolated zeros. The discrete case involves a set of N distinct points $t_1, t_2, \ldots, t_N$ along with positive weight factors

Table 2.1 Types of approximation and associated norms

Continuous norm	Approximation	Discrete norm
$\|u\|_\infty = \max\limits_{a\le t\le b}\|u(t)\|$	L_∞ Uniform Chebyshev	$\|u\|_\infty = \max\limits_{1\le i\le N}\|u(t_i)\|$
$\|u\|_1 = \int_a^b \|u(t)\|\mathrm{d}t$	L_1	$\|u\|_1 = \sum\limits_{i=1}^N \|u(t_i)\|$
$\|u\|_{1,w} = \int_a^b \|u(t)\|w(t)\mathrm{d}t$	Weighted L_1	$\|u\|_{1,w} = \sum\limits_{i=1}^N w_i\|u(t_i)\|$
$\|u\|_{2,w} = \left(\int_a^b \|u(t)\|^2 w(t)\mathrm{d}t\right)^{\frac{1}{2}}$	Weighted L_2 Least squares	$\|u\|_{2,w} = \left(\sum\limits_{i=1}^N w_i\|u(t_i)\|^2\right)^{\frac{1}{2}}$

$w_1, w_2, \dots, w_N$ (possibly all equal to 1). The interval $[a,b]$ may be unbounded if the weight function w is such that the integral extended over $[a,b]$, which defines the norm, makes sense.

Hence, we may take any one of the norms in Table 2.1 and combine it with any of the preceding linear spaces Φ to arrive at a meaningful best approximation problem (2.1). In the continuous case, the given function f, and the functions φ of the class Φ, of course, must be defined on $[a,b]$ and such that the norm $\|f-\varphi\|$ makes sense. Likewise, f and φ must be defined at the points t_i in the discrete case.

Note that if the best approximant $\hat{\varphi}$ in the discrete case is such that $\|f-\hat{\varphi}\| = 0$, then $\hat{\varphi}(t_i) = f(t_i)$ for $i = 1, 2, \dots, N$. We then say that $\hat{\varphi}$ *interpolates* f at the points t_i and we refer to this kind of approximation problem as an *interpolation problem*.

The simplest approximation problems are the least squares problem and the interpolation problem, and the easiest space Φ to work with the space of polynomials of given degree. These are indeed the problems we concentrate on in this chapter. In the case of the least squares problem, however, we admit general linear spaces Φ of approximants, and also in the case of the interpolation problem, we include polynomial splines in addition to straight polynomials.

Before we start with the least squares problem, we introduce a notational device that allows us to treat the continuous and the discrete case simultaneously. We define, in the continuous case,

$$\lambda(t) = \begin{cases} 0 \ \text{ if } \ t < a \ \ (\text{whenever } -\infty < a), \\ \displaystyle\int_a^t w(\tau)\mathrm{d}\tau \ \text{ if } \ a \le t \le b, \\ \displaystyle\int_a^b w(\tau)\mathrm{d}\tau \ \text{ if } \ t > b \ \ (\text{whenever } b < \infty). \end{cases} \tag{2.3}$$

Then we can write, for any (say, continuous) function u,

$$\int_{\mathbb{R}} u(t)\mathrm{d}\lambda(t) = \int_a^b u(t)w(t)\mathrm{d}t, \tag{2.4}$$

since $\mathrm{d}\lambda(t) \equiv 0$ "outside" $[a,b]$, and $\mathrm{d}\lambda(t) = w(t)\mathrm{d}t$ inside. We call $\mathrm{d}\lambda$ a *continuous* (positive) *measure*. The *discrete measure* (also called "Dirac measure") associated with the point set $\{t_1, t_2, \dots, t_N\}$ is a measure $\mathrm{d}\lambda$ that is nonzero only at the points t_i and has the value w_i there. Thus, in this case,

$$\int_{\mathbb{R}} u(t)\mathrm{d}\lambda(t) = \sum_{i=1}^{N} w_i u(t_i). \tag{2.5}$$

(A more precise definition can be given in terms of Stieltjes integrals, if we define $\lambda(t)$ to be a *step function* having jump w_i at t_i.) In particular, we can define the L_2 norm as

$$\|u\|_{2,\mathrm{d}\lambda} = \left(\int_{\mathbb{R}} |u(t)|^2 \mathrm{d}\lambda(t)\right)^{\frac{1}{2}}, \tag{2.6}$$

and obtain the continuous or the discrete norm depending on whether λ is taken to be as in (2.3), or a step function, as in (2.5).

We call the *support* of $\mathrm{d}\lambda$ – and denote it by $\operatorname{supp}\mathrm{d}\lambda$ – the interval $[a,b]$ in the continuous case (assuming w positive on $[a,b]$ except for isolated zeros), and the set $\{t_1, t_2, \dots, t_N\}$ in the discrete case. We say that the set of functions $\pi_j(t)$ in (2.2) is *linearly independent* on the support of $\mathrm{d}\lambda$ if

$$\sum_{j=1}^{n} c_j \pi_j(t) \equiv 0 \text{ for all } t \in \operatorname{supp}\mathrm{d}\lambda \text{ implies } c_1 = c_2 = \dots = c_n = 0. \tag{2.7}$$

Example: the powers $\pi_j(t) = t^{j-1}$, $j = 1, 2, \dots, n$.

Here $\sum_{j=1}^{n} c_j \pi_j(t) = p_{n-1}(t)$ is a polynomial of degree $\leq n-1$. Suppose, first, that $\operatorname{supp}\mathrm{d}\lambda = [a,b]$. Then the identity in (2.7) says that $p_{n-1}(t) \equiv 0$ on $[a,b]$. Clearly, this implies $c_1 = c_2 = \dots = c_n = 0$, so that the powers are linearly independent on $\operatorname{supp}\mathrm{d}\lambda = [a,b]$. If, on the other hand, $\operatorname{supp}\mathrm{d}\lambda = \{t_1, t_2, \dots, t_N\}$, then the premise in (2.7) says that $p_{n-1}(t_i) = 0$, $i = 1, 2, \dots, N$; that is, p_{n-1} has N distinct zeros t_i. This implies $p_{n-1} \equiv 0$ only if $N \geq n$. Otherwise, $p_{n-1}(t) = \prod_{i=1}^{N}(t - t_i) \in \mathbb{P}_{n-1}$ would satisfy $p_{n-1}(t_i) = 0$, $i = 1, 2, \dots, N$, without being identically zero. Thus, we have linear independence on $\operatorname{supp}\mathrm{d}\lambda = \{t_1, t_2, \dots, t_N\}$ if and only if $N \geq n$.

2.1 Least Squares Approximation

We specialize the best approximation problem (2.1) by taking as norm the L_2 norm

$$\|u\|_{2,\mathrm{d}\lambda} = \left(\int_{\mathbb{R}} |u(t)|^2 \mathrm{d}\lambda(t) \right)^{\frac{1}{2}}, \tag{2.8}$$

where $\mathrm{d}\lambda$ is either a continuous measure (cf. (2.3)) or a discrete measure (cf. (2.5)), and by using approximants φ from an n-dimensional linear space

$$\Phi = \Phi_n = \left\{ \varphi : \;\; \varphi(t) = \sum_{j=1}^{n} c_j \pi_j(t),\; c_j \in \mathbb{R} \right\}. \tag{2.9}$$

Here the basis functions π_j are assumed linearly independent on $\operatorname{supp} \mathrm{d}\lambda$ (cf. (2.7)). We furthermore assume, of course, that the integral in (2.8) is meaningful whenever $u = \pi_j$ or $u = f$, the given function to be approximated.

The solution of the least squares problem is most easily expressed in terms of orthogonal systems π_j relative to an appropriate inner product. We therefore begin with a discussion of inner products.

2.1.1 Inner Products

Given a continuous or discrete measure $\mathrm{d}\lambda$, as introduced earlier, and given any two functions u, v having a finite norm (2.8), we can define the *inner product*

$$(u, v) = \int_{\mathbb{R}} u(t) v(t) \mathrm{d}\lambda(t). \tag{2.10}$$

(Schwarz's inequality $|(u,v)| \le \|u\|_{2,\mathrm{d}\lambda} \cdot \|v\|_{2,\mathrm{d}\lambda}$, cf. Ex. 6, tells us that the integral in (2.10) is well defined.) The inner product (2.10) has the following obvious (but useful) properties:

1. symmetry: $(u, v) = (v, u)$;
2. homogeneity: $(\alpha u, v) = \alpha(u, v)$, $\alpha \in \mathbb{R}$;
3. additivity: $(u + v, w) = (u, w) + (v, w)$; and
4. positive definiteness: $(u, u) \ge 0$, with equality holding if and only if $u \equiv 0$ on $\operatorname{supp} \mathrm{d}\lambda$.

Homogeneity and additivity together give *linearity*,

$$(\alpha_1 u_1 + \alpha_2 u_2, v) = \alpha_1 (u_1, v) + \alpha_2 (u_2, v) \tag{2.11}$$

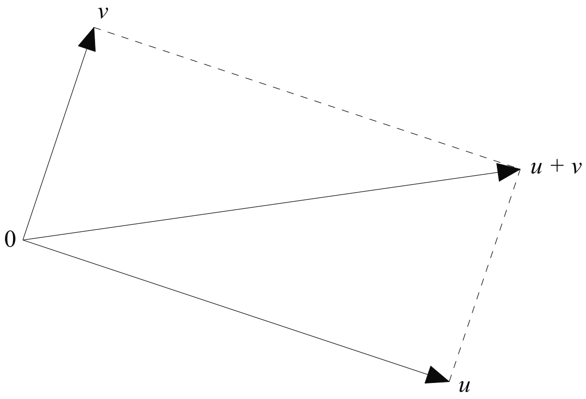


Fig. 2.1 Orthogonal vectors and their sum

in the first variable and, by symmetry, also in the second. Moreover, (2.11) easily extends to linear combinations of arbitrary finite length. Note also that

$$\|u\|^2_{2,\,\mathrm{d}\lambda} = (u,u). \tag{2.12}$$

We say that u and v are *orthogonal* if

$$(u,v) = 0. \tag{2.13}$$

This is always trivially true if either u or v vanishes identically on supp $\mathrm{d}\lambda$.

It is now a simple exercise, for example, to prove the *Theorem of Pythagoras*:

$$\text{if } \ (u,v) = 0, \ \text{ then } \ \|u+v\|^2 = \|u\|^2 + \|v\|^2, \tag{2.14}$$

where $\|\cdot\| = \|\cdot\|_{2,\mathrm{d}\lambda}$. (From now on we use this abbreviated notation for the norm.) Indeed,

$$\begin{aligned}\|u+v\|^2 &= (u+v,u+v) = (u,u) + (u,v) + (v,u) + (v,v)\\ &= \|u\|^2 + 2(u,v) + \|v\|^2 = \|u\|^2 + \|v\|^2,\end{aligned}$$

where the first equality is a definition, the second follows from additivity, the third from symmetry, and the last from orthogonality. Interpreting functions u, v as "vectors," we can picture the configuration of u, v (orthogonal) and $u+v$ as in Fig. 2.1.

More generally, we may consider an *orthogonal systems* $\{u_k\}_{k=1}^n$:

$$\begin{gathered}(u_i,u_j) = 0 \ \text{ if } \ i \neq j, \ \ u_k \not\equiv 0 \ \text{ on supp } \mathrm{d}\lambda;\\ i,j = 1,2,\ldots,n; \ \ k = 1,2,\ldots,n.\end{gathered} \tag{2.15}$$

For such a system we have the *Generalized Theorem of Pythagoras*,

$$\left\| \sum_{k=1}^{n} \alpha_k u_k \right\|^2 = \sum_{k=1}^{n} |\alpha_k|^2 \|u_k\|^2. \tag{2.16}$$

The proof is essentially the same as before. An important consequence of (2.16) is that *every orthogonal system is linearly independent* on the support of $\mathrm{d}\lambda$. Indeed, if the left-hand side of (2.16) vanishes, then so does the right-hand side, and this, since $\|u_k\|^2 > 0$ by assumption, implies $\alpha_1 = \alpha_2 = \cdots = \alpha_n = 0$.

2.1.2 The Normal Equations

We are now in a position to solve the least squares approximation problem. By (2.12), we can write the L_2 error, or rather its square, in the form:

$$E^2[\varphi] := \|\varphi - f\|^2 = (\varphi - f, \varphi - f) = (\varphi, \varphi) - 2(\varphi, f) + (f, f).$$

Inserting φ here from (2.9) gives

$$E^2[\varphi] = \int_{\mathbb{R}} \left(\sum_{j=1}^{n} c_j \pi_j(t) \right)^2 \mathrm{d}\lambda(t) - 2 \int_{\mathbb{R}} \left(\sum_{j=1}^{n} c_j \pi_j(t) \right) f(t) \mathrm{d}\lambda(t) + \int_{\mathbb{R}} f^2(t) \mathrm{d}\lambda(t). \tag{2.17}$$

The squared L_2 error, therefore, is a quadratic function of the coefficients $c_1, c_2, \ldots, c_n$ of φ. The problem of best L_2 approximation thus amounts to minimizing a quadratic function of n variables. This is a standard problem of calculus and is solved by setting all partial derivatives equal to zero. This yields a system of *linear* algebraic equations. Indeed, differentiating partially with respect to c_i under the integral sign in (2.17) gives

$$\frac{\partial}{\partial c_i} E^2[\varphi] = 2 \int_{\mathbb{R}} \left(\sum_{j=1}^{n} c_j \pi_j(t) \right) \pi_i(t) \mathrm{d}\lambda(t) - 2 \int_{\mathbb{R}} \pi_i(t) f(t) \mathrm{d}\lambda(t),$$

and setting this equal to zero, interchanging integration and summation in the process, we get

$$\sum_{j=1}^{n} (\pi_i, \pi_j) c_j = (\pi_i, f), \quad i = 1, 2, \ldots, n. \tag{2.18}$$

These are called the *normal equations* for the least squares problem. They form a linear system of the form

$$\boldsymbol{A}\boldsymbol{c} = \boldsymbol{b}, \tag{2.19}$$

where the matrix $\boldsymbol{A}$ and the vector $\boldsymbol{b}$ have elements

$$\boldsymbol{A} = [a_{ij}],\ a_{ij} = (\pi_i, \pi_j);\ \ \boldsymbol{b} = [b_i],\ b_i = (\pi_i, f). \tag{2.20}$$

By symmetry of the inner product, $\boldsymbol{A}$ is a symmetric matrix. Moreover, $\boldsymbol{A}$ is positive definite; that is,

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}x_i x_j > 0 \ \text{ if } \ \boldsymbol{x} \neq [0, 0, \ldots, 0]^{\mathrm{T}}. \tag{2.21}$$

The quadratic function in (2.21) is called a *quadratic form* (since it is homogeneous of degree 2). Positive definiteness of $\boldsymbol{A}$ thus says that the quadratic form whose coefficients are the elements of $\boldsymbol{A}$ is always nonnegative, and zero only if all variables x_i vanish.

To prove (2.21), all we have to do is insert the definition of the a_{ij} and use the elementary properties 1–4 of the inner product:

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j (\pi_i, \pi_j) = \sum_{i=1}^{n}\sum_{j=1}^{n} (x_i\pi_i, x_j\pi_j) = \left\| \sum_{i=1}^{n} x_i\pi_i \right\|^2 .$$

This clearly is nonnegative. It is zero only if $\sum_{i=1}^{n} x_i\pi_i \equiv 0$ on $\operatorname{supp} \mathrm{d}\lambda$, which, by the assumption of linear independence of the π_i, implies $x_1 = x_2 = \cdots = x_n = 0$.

Now it is a well-known fact of linear algebra that a symmetric positive definite matrix $\boldsymbol{A}$ is nonsingular. Indeed, its determinant, as well as all its leading principal minor determinants, are strictly positive. It follows that the system (2.18) of normal equations has a unique solution. Does this solution correspond to a minimum of $E[\varphi]$ in (2.17)? Calculus tells us that for this to be the case, the Hessian matrix $\boldsymbol{H} = [\partial^2 E^2/\partial c_i \partial c_j]$ has to be positive definite. But $\boldsymbol{H} = 2\boldsymbol{A}$, since E^2 is a quadratic function. Therefore, $\boldsymbol{H}$, with $\boldsymbol{A}$, is indeed positive definite, and the solution of the normal equations gives us the desired minimum. The least squares approximation problem thus has a unique solution, given by

$$\hat{\varphi}(t) = \sum_{j=1}^{n} \hat{c}_j \pi_j(t), \tag{2.22}$$

where $\hat{\boldsymbol{c}} = [\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_n]^{\mathrm{T}}$ is the solution vector of the normal equation (2.18).

This completely settles the least squares approximation problem in theory. How about in practice?

Assuming a general set of (linearly independent) basis functions, we can see the following possible difficulties.

1. The system (2.18) may be *ill-conditioned*. A simple example is provided by $\operatorname{supp} \mathrm{d}\lambda = [0, 1]$, $\mathrm{d}\lambda(t) = \mathrm{d}t$ on [0,1], and $\pi_j(t) = t^{j-1}$, $j = 1, 2, \ldots, n$. Then

$$(\pi_i, \pi_j) = \int_0^1 t^{i+j-2}\,\mathrm{d}t = \frac{1}{i+j-1},\ \ i, j = 1, 2, \ldots, n;$$

that is, the matrix $\boldsymbol{A}$ in (2.18) is precisely the Hilbert matrix (cf. Chap. 1, (1.60)). The resulting severe ill-conditioning of the normal equations in this example is entirely due to an unfortunate choice of basis functions – the powers. These become almost linearly dependent, more so the larger the exponent (cf. Ex. 38). Another source of degradation lies in the element $b_j = \int_0^1 \pi_j(t)f(t)\mathrm{d}t$ of the right-hand vector $\boldsymbol{b}$ in (2.18). When j is large, the power $\pi_j = t^{j-1}$ behaves very much like a discontinuous function on [0,1]: it is practically zero for much of the interval until it shoots up to the value 1 at the right endpoint. This has the unfortunate consequence that a good deal of information about f is lost when one forms the integral defining b_j. A polynomial π_j that oscillates rapidly on [0,1] would seem to be preferable from this point of view, since it would "engage" the function f more vigorously over all of the interval [0,1].

2. The second disadvantage is the fact that all coefficients $\hat{c}_j$ in (2.22) depend on n; that is, $\hat{c}_j = \hat{c}_j^{(n)}$, $j = 1, 2, \dots, n$. Increasing n, for example, will give an enlarged system of normal equations with a completely new solution vector. We refer to this as the *nonpermanence* of the coefficients $\hat{c}_j$.

Both defects 1 and 2 can be eliminated (or at least attenuated in the case of 1) in one stroke: select for the basis functions π_j an orthogonal system,

$$(\pi_i, \pi_j) = 0 \ \text{ if } \ i \neq j; \ \ (\pi_j, \pi_j) = \|\pi_j\|^2 > 0. \tag{2.23}$$

Then the system of normal equations becomes diagonal and is solved immediately by

$$\hat{c}_j = \frac{(\pi_j, f)}{(\pi_j, \pi_j)}, \quad j = 1, 2, \dots, n. \tag{2.24}$$

Clearly, each of these coefficients $\hat{c}_j$ is independent of n, and once computed, remains the same for any larger n. We now have *permanence* of the coefficients. Also, we do not have to go through the trouble of solving a linear system of equations, but instead can use the formula (2.24) directly. This does not mean that there are no numerical problems associated with (2.24). Indeed, it is typical that the denominators $\|\pi_j\|^2$ in (2.24) decrease rapidly with increasing j, whereas the integrand in the numerator (or the individual terms in the case of a discrete inner product) are of the same magnitude as f. Yet the coefficients $\hat{c}_j$ also are expected to decrease rapidly. Therefore, cancellation errors must occur when one computes the inner product in the numerator. The cancellation problem can be alleviated somewhat by computing $\hat{c}_j$ in the alternative form

$$\hat{c}_j = \frac{1}{(\pi_j, \pi_j)} \left(f - \sum_{k=1}^{j-1} \hat{c}_k \pi_k, \pi_j \right), \quad j = 1, 2, \dots, n, \tag{2.25}$$

where the empty sum (when $j = 1$) is taken to be zero, as usual. Clearly, by orthogonality of the π_j, (2.25) is equivalent to (2.24) mathematically, but not necessarily numerically.

An algorithm for computing $\hat{c}_j$ from (2.25), and at the same time $\hat{\varphi}(t)$, is as follows:

```
$s_0 = 0$,
for $j = 1, 2, \dots, n$ do
    $\hat{c}_j = \dfrac{1}{\|\pi_j\|^2}\,(f - s_{j-1}, \pi_j)$
    $s_j = s_{j-1} + \hat{c}_j \pi_j(t)$.
```

This produces the coefficients $\hat{c}_1, \hat{c}_2, \dots, \hat{c}_n$ as well as $\hat{\varphi}(t) = s_n$.

Any system $\{\hat{\pi}_j\}$ that is linearly independent on the support of $\mathrm{d}\lambda$ can be orthogonalized (with respect to the measure $\mathrm{d}\lambda$) by a device known as the *Gram*[1]–Schmidt [2]*procedure*. One takes

$$\pi_1 = \hat{\pi}_1$$

and, for $j = 2, 3, \dots$, recursively forms

$$\pi_j = \hat{\pi}_j - \sum_{k=1}^{j-1} c_k \pi_k, \quad c_k = \frac{(\hat{\pi}_j, \pi_k)}{(\pi_k, \pi_k)}.$$

Then each π_j so determined is orthogonal to all preceding ones.

2.1.3 Least Squares Error; Convergence

We have seen in Sect. 2.1.2 that if the class $\Phi = \Phi_n$ consists of n functions π_j, $j = 1, 2, \dots, n$, that are linearly independent on the support of some measure $\mathrm{d}\lambda$, then the least squares problem for this measure,

$$\min_{\varphi \in \Phi_n} \|f - \varphi\|_{2,\mathrm{d}\lambda} = \|f - \hat{\varphi}\|_{2,\mathrm{d}\lambda}, \tag{2.26}$$

[1] Jórgen Pedersen Gram (1850–1916) was a farmer's son who studied at the University of Copenhagen. After graduation, he entered an insurance company as computer assistant and, moving up the ranks, eventually became its director. He was interested in series expansions of special functions and also contributed to Chebyshev and least squares approximation. The "Gram determinant" was introduced by him in connection with his study of linear independence.

[2] Erhard Schmidt (1876–1959), a student of Hilbert, became a prominent member of the Berlin School of Mathematics, where he founded the Institute of Applied Mathematics. He is considered one of the originators of Functional Analysis, having contributed substantially to the theory of Hilbert spaces. His work on linear and nonlinear integral equations is of lasting interest, as is his contribution to linear algebraic systems of infinite dimension. He is also known for his proof of the Jordan curve theorem. His procedure of orthogonalization was published in 1907 and today also carries the name of Gram. It was known, however, already to Laplace.

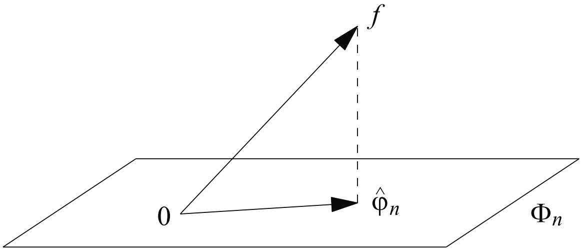


Fig. 2.2 Least squares approximation as orthogonal projection

has a unique solution $\hat{\varphi} = \hat{\varphi}_n$ given by (2.22). There are many ways we can select a basis π_j in Φ_n and, therefore, many ways the solution $\hat{\varphi}_n$ can be represented. Nevertheless, it is always one and the same function. The least squares error – the quantity on the right-hand side of (2.26) – therefore is independent of the choice of basis functions (although the calculation of the least squares solution, as mentioned previously, is not). In studying this error, we may thus assume, without restricting generality, that the basis π_j is an orthogonal system. (Every linearly independent system can be orthogonalized by the Gram–Schmidt orthogonalization procedure; cf. Sect. 2.1.2.) We then have (cf. (2.24))

$$\hat{\varphi}_n(t) = \sum_{j=1}^{n} \hat{c}_j \pi_j(t), \quad \hat{c}_j = \frac{(\pi_j, f)}{(\pi_j, \pi_j)}. \tag{2.27}$$

We first note that the error $f - \hat{\varphi}_n$ is orthogonal to the space Φ_n; that is,

$$(f - \hat{\varphi}_n, \varphi) = 0 \text{ for all } \varphi \in \Phi_n, \tag{2.28}$$

where the inner product is the one in (2.10). Since φ is a linear combination of the π_k, it suffices to show (2.28) for each $\varphi = \pi_k$, $k = 1, 2, \dots, n$. Inserting $\hat{\varphi}_n$ from (2.27) in the left-hand side of (2.28), and using orthogonality, we find indeed

$$(f - \hat{\varphi}_n, \pi_k) = \left(f - \sum_{j=1}^{n} \hat{c}_j \pi_j, \pi_k \right) = (f, \pi_k) - \hat{c}_k(\pi_k, \pi_k) = 0,$$

the last equation following from the formula for $\hat{c}_k$ in (2.27). The result (2.28) has a simple geometric interpretation. If we picture functions as vectors, and the space Φ_n as a plane, then for any f that "sticks out" of the plane Φ_n, the least squares approximant $\hat{\varphi}_n$ is the *orthogonal projection* of f onto Φ_n; see Fig. 2.2.

In particular, choosing $\varphi = \hat{\varphi}_n$ in (2.28), we get

$$(f - \hat{\varphi}_n, \hat{\varphi}_n) = 0$$

and, therefore, since $f = (f - \hat{\varphi}_n) + \hat{\varphi}_n$, by the Theorem of Pythagoras (cf. (2.14)) and its generalization (cf. (2.16)),

$$\begin{aligned}\|f\|^2 &= \|f - \hat{\varphi}_n\|^2 + \|\hat{\varphi}_n\|^2 \\ &= \|f - \hat{\varphi}_n\|^2 + \left\| \sum_{j=1}^{n} \hat{c}_j \pi_j \right\|^2 \\ &= \|f - \hat{\varphi}_n\|^2 + \sum_{j=1}^{n} |\hat{c}_j|^2 \|\pi_j\|^2.\end{aligned}$$

Solving for the first term on the right-hand side, we get

$$\|f - \hat{\varphi}_n\| = \left\{ \|f\|^2 - \sum_{j=1}^{n} |\hat{c}_j|^2 \|\pi_j\|^2 \right\}^{\frac{1}{2}}, \quad \hat{c}_j = \frac{(\pi_j, f)}{(\pi_j, \pi_j)}. \tag{2.29}$$

Note that the expression in braces must necessarily be nonnegative.

The formula (2.29) for the error is interesting theoretically, but of limited practical use. Note, indeed, that as the error approaches the level of the machine precision eps, computing the error from the right-hand side of (2.29) cannot produce anything smaller than $\sqrt{\text{eps}}$ because of inevitable rounding errors committed during the subtraction in the radicand. (They may even produce a negative result for the radicand.) Using instead the definition,

$$\|f - \hat{\varphi}_n\| = \left\{ \int_{\mathbb{R}} [f(t) - \hat{\varphi}_n(t)]^2 \mathrm{d}\lambda(t) \right\}^{\frac{1}{2}},$$

along, perhaps, with a suitable (positive) quadrature rule (cf. Chap. 3, Sect. 3.2), is guaranteed to produce a nonnegative result that may potentially be as small as $O(\text{eps})$.

If we are now given a sequence of linear spaces Φ_n, $n = 1, 2, 3, \ldots$, as defined in (2.2), then clearly

$$\|f - \hat{\varphi}_1\| \geq \|f - \hat{\varphi}_2\| \geq \|f - \hat{\varphi}_3\| \geq \cdots,$$

which follows not only from (2.29), but more directly from the fact that $\Phi_1 \subset \Phi_2 \subset \Phi_3 \subset \cdots$. If there are infinitely many such spaces, then the sequence of L_2 errors, being monotonically decreasing, must converge to a limit. Is this limit zero? If so, we say that the least squares approximation process *converges* (in the mean) as $n \to \infty$. It is obvious from (2.29) that a necessary and sufficient condition for this is

$$\sum_{j=1}^{\infty} |\hat{c}_j|^2 \|\pi_j\|^2 = \|f\|^2. \tag{2.30}$$

An equivalent way of stating convergence is as follows: given any f with $\|f\| < \infty$, that is, any f in the $L_{2,\mathrm{d}\lambda}$ space, and given any $\varepsilon > 0$, no matter how small, there exists an integer $n = n_\varepsilon$ and a function $\varphi^* \in \Phi_n$ such that $\|f - \varphi^*\| \le \varepsilon$. A class of spaces Φ_n having this property is said to be *complete* with respect to the norm $\|\cdot\| = \|\cdot\|_{2,\mathrm{d}\lambda}$. One therefore calls (2.30) also the *completeness relation*.

For a finite interval $[a,b]$, one can define completeness of $\{\Phi_n\}$ also for the uniform norm $\|\cdot\| = \|\cdot\|_\infty$ on $[a,b]$. One then assumes $f \in C[a,b]$ and also $\pi_j \in C[a,b]$ for all basis functions in all classes Φ_n, and one calls $\{\Phi_n\}$ complete in the norm $\|\cdot\|_\infty$ if for any $f \in C[a,b]$ and any $\varepsilon > 0$ there is an $n = n_\varepsilon$ and a $\varphi^* \in \Phi_n$ such that $\|f - \varphi^*\|_\infty \le \varepsilon$. It is easy to see that completeness of $\{\Phi_n\}$ in the norm $\|\cdot\|_\infty$ (on $[a,b]$) implies completeness of $\{\Phi_n\}$ in the L_2 norm $\|\cdot\|_{2,\mathrm{d}\lambda}$, where $\operatorname{supp}\mathrm{d}\lambda = [a,b]$, and hence convergence of the least squares approximation process. Indeed, let $\varepsilon > 0$ be arbitrary and let n and $\varphi^* \in \Phi_n$ be such that

$$\|f - \varphi^*\|_\infty \le \frac{\varepsilon}{\left(\int_{\mathbb{R}} \mathrm{d}\lambda(t)\right)^{\frac{1}{2}}}.$$

This is possible by assumption. Then

$$\begin{aligned}\|f - \varphi^*\|_{2,\mathrm{d}\lambda} &= \left(\int_{\mathbb{R}} [f(t) - \varphi^*(t)]^2 \mathrm{d}\lambda(t)\right)^{\frac{1}{2}} \\ &\le \|f - \varphi^*\|_\infty \left(\int_{\mathbb{R}} \mathrm{d}\lambda(t)\right)^{\frac{1}{2}} \le \frac{\varepsilon}{\left(\int_{\mathbb{R}} \mathrm{d}\lambda(t)\right)^{\frac{1}{2}}} \left(\int_{\mathbb{R}} \mathrm{d}\lambda(t)\right)^{\frac{1}{2}} = \varepsilon,\end{aligned}$$

as claimed.

Example: $\Phi_n = \mathbb{P}_{n-1}$.

Here completeness of $\{\Phi_n\}$ in the norm $\|\cdot\|_\infty$ (on a finite interval $[a,b]$) is a consequence of Weierstrass's Approximation Theorem. Thus, polynomial least squares approximation on a finite interval always converges (in the mean).

2.1.4 Examples of Orthogonal Systems

There are many orthogonal systems in use. The prototype of them all is the system of trigonometric functions known from Fourier analysis. Other widely used systems involve algebraic polynomials. We restrict ourselves here to these two particular examples of orthogonal systems.

1. *Trigonometric functions*: 1, $\cos t$, $\cos 2t$, $\cos 3t$, ..., $\sin t$, $\sin 2t$, $\sin 3t$, ... These are the basic harmonics; they are mutually orthogonal on the interval $[0, 2\pi]$ with respect to the equally weighted measure on $[0,2\pi]$,

$$d\lambda(t) = \begin{cases} dt & \text{on } [0, 2\pi], \\ 0 & \text{otherwise.} \end{cases} \tag{2.31}$$

We verify this for the sine functions: for $k, \ell = 1, 2, 3, \dots$ we have

$$\int_0^{2\pi} \sin kt \cdot \sin \ell t \, dt = -\frac{1}{2} \int_0^{2\pi} [\cos(k+\ell)t - \cos(k-\ell)t] \, dt.$$

The right-hand side is equal to

$$-\frac{1}{2}\left[\frac{\sin(k+\ell)t}{k+\ell} - \frac{\sin(k-\ell)t}{k-\ell}\right]_0^{2\pi} = 0,$$

when $k \neq \ell$, and equal to π otherwise. Thus,

$$\int_0^{2\pi} \sin kt \cdot \sin \ell t \, dt = \begin{cases} 0 & \text{if } k \neq \ell, \\ \pi & \text{if } k = \ell, \end{cases} \quad k, \ell = 1, 2, 3, \dots. \tag{2.32}$$

Similarly, one shows that

$$\int_0^{2\pi} \cos kt \cdot \cos \ell t \, dt = \begin{cases} 0 & \text{if } k \neq \ell, \\ 2\pi & \text{if } k = \ell = 0, \\ \pi & \text{if } k = \ell > 0, \end{cases} \quad k, \ell = 0, 1, 2, \dots \tag{2.33}$$

and

$$\int_0^{2\pi} \sin kt \cdot \cos \ell t \, dt = 0, \quad k = 1, 2, 3, \dots, \quad \ell = 0, 1, 2, \dots . \tag{2.34}$$

The theory of *Fourier series* is concerned with the expansion of a given 2π-periodic function in terms of these trigonometric functions,

$$f(t) = \sum_{k=0}^{\infty} a_k \cos kt + \sum_{k=1}^{\infty} b_k \sin kt. \tag{2.35}$$

Using (2.32)–(2.34), one formally obtains

$$\begin{aligned} a_0 &= \frac{1}{2\pi}\int_0^{2\pi} f(t)\,\mathrm{d}t, \quad a_k = \frac{1}{\pi}\int_0^{2\pi} f(t)\cos kt\,\mathrm{d}t, \quad k = 1,2,\dots, \\ b_k &= \frac{1}{\pi}\int_0^{2\pi} f(t)\sin kt\,\mathrm{d}t, \quad k = 1,2,\dots, \end{aligned} \tag{2.36}$$

which are known as *Fourier coefficients* of f. They are precisely the coefficients (2.24) for the system π_j consisting of our trigonometric functions. By extension, one therefore calls the coefficients $\hat{c}_j$ in (2.24), for *any* orthogonal system π_j, the Fourier coefficients of f relative to this system. In particular, we now recognize the truncated Fourier series (the series on the right-hand side of (2.35) truncated at $k = m$, with a_k, b_k given by (2.36)) as the best L_2 approximation to f from the class of trigonometric polynomials of degree $\le m$ relative to the norm (cf. (2.31))

$$\|u\|_2 = \left(\int_0^{2\pi} |u(t)|^2 \mathrm{d}t\right)^{\frac{1}{2}}.$$

2. *Orthogonal polynomials*: given a measure $\mathrm{d}\lambda$ as introduced in (2.3)–(2.5), we know from the example immediately following (2.7) that any finite number of consecutive powers $1, t, t^2, \dots$ are linearly independent on $[a,b]$, if $\operatorname{supp}\mathrm{d}\lambda = [a,b]$, whereas the finite set $1, t, \dots, t^{N-1}$ is linearly independent on $\operatorname{supp}\mathrm{d}\lambda = \{t_1, t_2, \dots, t_N\}$. Since a linearly independent set can be orthogonalized by Gram–Schmidt (cf. Sect. 2.1.2), any measure $\mathrm{d}\lambda$ of the type considered generates a unique set of (monic) polynomials $\pi_j(t) = \pi_j(t;\mathrm{d}\lambda)$, $j = 0,1,2,\dots$, satisfying

$$\begin{aligned} &\deg \pi_j = j, \quad j = 0,1,2,\dots, \\ &\int_{\mathbb{R}} \pi_k(t)\pi_\ell(t)\mathrm{d}\lambda(t) = 0 \ \text{ if } k \ne \ell. \end{aligned} \tag{2.37}$$

These are called *orthogonal polynomials* relative to the measure $\mathrm{d}\lambda$. (We slightly deviate from the notation in Sects. 2.1.2 and 2.1.3 by letting the index j start from zero.) The set π_j is infinite if $\operatorname{supp}\mathrm{d}\lambda = [a,b]$, and consists of exactly N polynomials $\pi_0, \pi_1, \dots, \pi_{N-1}$ if $\operatorname{supp}\mathrm{d}\lambda = \{t_1, t_2, \dots, t_N\}$. The latter are referred to as *discrete orthogonal polynomials.*

It is an important fact that three consecutive orthogonal polynomials are linearly related. Specifically, there are real constants $\alpha_k = \alpha_k(\mathrm{d}\lambda)$ and positive constants $\beta_k = \beta_k(\mathrm{d}\lambda)$ (depending on the measure $\mathrm{d}\lambda$) such that

$$\begin{aligned} &\pi_{k+1}(t) = (t-\alpha_k)\pi_k(t) - \beta_k\pi_{k-1}(t), \quad k = 0,1,2,\dots, \\ &\pi_{-1}(t) = 0, \quad \pi_0(t) = 1. \end{aligned} \tag{2.38}$$

(It is understood that (2.38) holds for all integers $k \geq 0$ if $\operatorname{supp} d\lambda = [a,b]$, and only for $0 \leq k < N-1$ if $\operatorname{supp} d\lambda = \{t_1, t_2, \ldots, t_N\}$.)

To prove (2.38) and, at the same time identify the coefficients α_k, β_k, we note that

$$\pi_{k+1}(t) - t\pi_k(t)$$

is a polynomial of degree $\leq k$, since the leading terms cancel (the polynomials π_j are assumed monic). Since an orthogonal system is linearly independent (cf. the remark after (2.16)), we can express this polynomial as a linear combination of π_0, $\pi_1, \ldots, \pi_k$. We choose to write this linear combination in the form:

$$\pi_{k+1}(t) - t\pi_k(t) = -\alpha_k \pi_k(t) - \beta_k \pi_{k-1}(t) + \sum_{j=0}^{k-2} \gamma_{k,j} \pi_j(t) \tag{2.39}$$

(with the understanding that empty sums are zero). Now multiply both sides of (2.39) by π_k in the sense of the inner product $(\cdot\ ,\ \cdot)$ defined in (2.10). By orthogonality, this gives $(-t\pi_k, \pi_k) = -\alpha_k(\pi_k, \pi_k)$; that is,

$$\alpha_k = \frac{(t\pi_k, \pi_k)}{(\pi_k, \pi_k)}, \quad k = 0, 1, 2, \ldots. \tag{2.40}$$

Similarly, forming the inner product of (2.39) with π_{k-1} gives $(-t\pi_k, \pi_{k-1}) = -\beta_k(\pi_{k-1}, \pi_{k-1})$. Since $(t\pi_k, \pi_{k-1}) = (\pi_k, t\pi_{k-1})$ and $t\pi_{k-1}$ differs from π_k by a polynomial of degree $< k$, we obtain by orthogonality $(t\pi_k, \pi_{k-1}) = (\pi_k, \pi_k)$; hence

$$\beta_k = \frac{(\pi_k, \pi_k)}{(\pi_{k-1}, \pi_{k-1})}, \quad k = 1, 2, \ldots. \tag{2.41}$$

Finally, multiplication of (2.39) by π_ℓ, $\ell < k-1$, yields

$$\gamma_{k,\ell} = 0, \quad \ell = 0, 1, \ldots, k-2. \tag{2.42}$$

Solving (2.39) for π_{k+1} then establishes (2.38), with α_k, β_k defined by (2.40) and (2.41), respectively. Clearly, $\beta_k > 0$. By convention, $\beta_0 = \int_{\mathbb{R}} d\lambda(t) = \int_{\mathbb{R}} \pi_0^2(t) d\lambda(t)$.

The recursion (2.38) provides us with a practical scheme of generating orthogonal polynomials. Indeed, since $\pi_0 = 1$, we can compute α_0 by (2.40) with $k = 0$. This allows us to compute $\pi_1(t)$ for any t, using (2.38) with $k = 0$. Knowing π_0, π_1, we can go back to (2.40) and (2.41) and compute, respectively, α_1 and β_1. This gives us access to π_2 via (2.38) with $k = 1$. Proceeding in this fashion, using alternately (2.40), (2.41), and (2.38), we can generate as many orthogonal polynomials as are desired. This procedure – called *Stieltjes's procedure* – is particularly well suited for discrete orthogonal polynomials, since the inner product is then a finite sum, $(u,v) = \sum_{i=1}^{N} w_i u(t_i) v(t_i)$ (cf. (2.5)), so that the computation of the α_k, β_k from

(2.40) and (2.41) is straightforward. In the continuous case, the computation of the inner product requires integration, which complicates matters. Fortunately, for many important special measures $\mathrm{d}\lambda(t) = w(t)\mathrm{d}t$, the recursion coefficients are explicitly known (cf. Chap. 3, Table 3.1). In these cases, it is again straightforward to generate the orthogonal polynomials by (2.38).

The special case of *symmetry* (i.e., $\mathrm{d}\lambda(t) = w(t)\mathrm{d}t$ with $w(-t) = w(t)$ and $\mathrm{supp}(\mathrm{d}\lambda)$ symmetric with respect to the origin) deserves special mention. In this case, defining $p_k(t) = (-1)^k \pi_k(-t)$, one obtains by a simple change of variables that $(p_k, p_\ell) = (-1)^{k+\ell}(\pi_k, \pi_\ell) = 0$ if $k \neq \ell$. Since p_k is monic, it follows by uniqueness that $p_k(t) \equiv \pi_k(t)$; that is,

$$(-1)^k \pi_k(-t) \equiv \pi_k(t) \quad (\mathrm{d}\lambda \text{ symmetric}). \tag{2.43}$$

Thus, if k is even, then π_k is an even polynomial, that is, a polynomial in t^2. Likewise, when k is odd, π_k contains only odd powers of t. As a consequence,

$$\alpha_k = 0 \quad \text{for all } k \geq 0 \quad (\mathrm{d}\lambda \text{ symmetric}), \tag{2.44}$$

which also follows from (2.40), since the numerator on the right-hand side of this equation is an integral of an odd function over a symmetric set of points.

Example: Legendre[3] polynomials.

We may introduce the monic Legendre polynomials by

$$\pi_k(t) = (-1)^k \frac{k!}{(2k)!} \frac{\mathrm{d}^k}{\mathrm{d}t^k}(1-t^2)^k, \quad k = 0, 1, 2, \dots, \tag{2.45}$$

which is known as the *Rodrigues formula*.

We first verify orthogonality on the interval $[-1, 1]$ relative to the measure $\mathrm{d}\lambda(t) = \mathrm{d}t$. For any ℓ with $0 \leq \ell < k$, repeated integration by parts gives

$$\begin{aligned}\int_{-1}^{1} \frac{\mathrm{d}^k}{\mathrm{d}t^k}(1-t^2)^k \cdot t^\ell \mathrm{d}t &= \sum_{m=0}^{\ell} (-1)^m \ell(\ell-1)\cdots(\ell-m+1) t^{\ell-m} \\ &\quad \times \left. \frac{\mathrm{d}^{k-m-1}}{\mathrm{d}t^{k-m-1}}(1-t^2)^k \right|_{-1}^{1} = 0,\end{aligned}$$

the last equation since $0 \leq k - m - 1 < k$. Thus,

$$(\pi_k, p) = 0 \quad \text{for every } p \in \mathbb{P}_{k-1},$$

[3] Adrien Marie Legendre (1752–1833) was a French mathematician active in Paris, best known not only for his treatise on elliptic integrals but also famous for his work in number theory and geometry. He is considered as the originator (in 1805) of the method of least squares, although Gauss had already used it in 1794, but published it only in 1809.

proving orthogonality. Writing (by symmetry)

$$\pi_k(t) = t^k + \mu_k t^{k-2} + \cdots, \quad k \geq 2,$$

and noting (again by symmetry) that the recurrence relation has the form

$$\pi_{k+1}(t) = t\pi_k(t) - \beta_k \pi_{k-1}(t),$$

we obtain

$$\beta_k = \frac{t\pi_k(t) - \pi_{k+1}(t)}{\pi_{k-1}(t)},$$

which is valid for all t. In particular, as $t \to \infty$,

$$\beta_k = \lim_{t\to\infty} \frac{t\pi_k(t) - \pi_{k+1}(t)}{\pi_{k-1}(t)} = \lim_{t\to\infty} \frac{(\mu_k - \mu_{k+1})t^{k-1} + \cdots}{t^{k-1} + \cdots} = \mu_k - \mu_{k+1}.$$

(If $k = 1$, set $\mu_1 = 0$.) From Rodrigues's formula, however, we find

$$\begin{aligned}\pi_k(t) = \frac{k!}{(2k)!}\frac{\mathrm{d}^k}{\mathrm{d}t^k}(t^{2k} - kt^{2k-2} + \cdots) &= \frac{k!}{(2k)!}\big(2k(2k-1)\cdots(k+1)t^k \\ &\quad -k\cdot(2k-2)(2k-3)\cdots(k-1)t^{k-2} + \cdots\big) \\ &= t^k - \frac{k(k-1)}{2(2k-1)}t^{k-2} + \cdots,\end{aligned}$$

so that

$$\mu_k = -\frac{k(k-1)}{2(2k-1)}, \quad k \geq 2.$$

Therefore,

$$\beta_k = \mu_k - \mu_{k+1} = -\frac{k(k-1)}{2(2k-1)} + \frac{(k+1)k}{2(2k+1)} = \frac{k}{2}\,\frac{2k}{(2k+1)(2k-1)};$$

that is, since $\mu_1 = 0$,

$$\beta_k = \frac{1}{4 - k^{-2}}, \quad k \geq 1. \tag{2.46}$$

We conclude with two remarks concerning discrete measures $\mathrm{d}\lambda$ with $\operatorname{supp}\mathrm{d}\lambda = \{t_1, t_2, \ldots, t_N\}$. As before, the L_2 errors decrease monotonically, but the last one is now zero, since there is a polynomial of degree $\leq N-1$ that interpolates f at the N points $t_1, t_2, \ldots, t_N$ (cf. Sect. 2.1.2). Thus,

$$\|f - \hat{\varphi}_0\| \geq \|f - \hat{\varphi}_1\| \geq \cdots \geq \|f - \hat{\varphi}_{N-1}\| = 0, \tag{2.47}$$

where $\hat{\varphi}_n$ is the L_2 approximant of degree $\le n$,

$$\hat{\varphi}_n(t) = \sum_{j=0}^{n} \hat{c}_j \pi_j(t; \mathrm{d}\lambda), \quad \hat{c}_j = \frac{(\pi_j, f)}{(\pi_j, \pi_j)}. \tag{2.48}$$

We see that the polynomial $\hat{\varphi}_{N-1}$ solves the interpolation problem for $\mathbb{P}_{N-1}$. Using (2.48) with $n = N - 1$ to obtain the interpolation polynomial, however, is a roundabout way of solving the interpolation problem. We learn of more direct ways in the next section.

2.2 Polynomial Interpolation

We now wish to approximate functions by matching their values at given points. Using polynomials as approximants gives rise to the following problem: given $n+1$ distinct points $x_0, x_1, \dots, x_n$ and values $f_i = f(x_i)$ of some function f at these points, find a polynomial $p \in \mathbb{P}_n$ such that

$$p(x_i) = f_i, \;\; i = 0, 1, 2, \dots, n.$$

Since we have to satisfy $n+1$ conditions, and have at our disposal $n+1$ degrees of freedom – the coefficients of p – we expect the problem to have a unique solution. Other questions of interest, in addition to existence and uniqueness, are different ways of representing and computing the polynomial p, what can be said about the error $e(x) = f(x) - p(x)$ when $x \ne x_i$, $i = 0, 1, \dots, n$, and the quality of approximation $f(x) \approx p(x)$ when the number of points, and hence the degree of p, is allowed to increase indefinitely. Although these questions are not of the utmost interest in themselves, the results discussed here are widely used in the development of approximate methods for more important practical tasks such as solving initial and boundary value problems for ordinary and partial differential equations. It is in view of these and other applications that we study polynomial interpolation.

The simplest example is *linear interpolation*, that is, the case $n = 1$. Here, it is obvious from Fig. 2.3 that the interpolation problem has a unique solution. It is also clear that the error $e(x)$ can be as large as one likes (or dislikes) if nothing is known about f other than its two values at x_0 and x_1.

One way of writing down the linear interpolant p is as a weighted average of f_0 and f_1 (already taught in high school),

$$p(x) = \frac{x - x_1}{x_0 - x_1} f_0 + \frac{x - x_0}{x_1 - x_0} f_1.$$

This is the way Lagrange expressed p in the general case (cf. Sect. 2.1.2). However, we can write p also in Taylor's form, noting that its derivative at x_0 is equal to the "difference quotient,"

$$p(x) = f_0 + \frac{f_1 - f_0}{x_1 - x_0}(x - x_0).$$

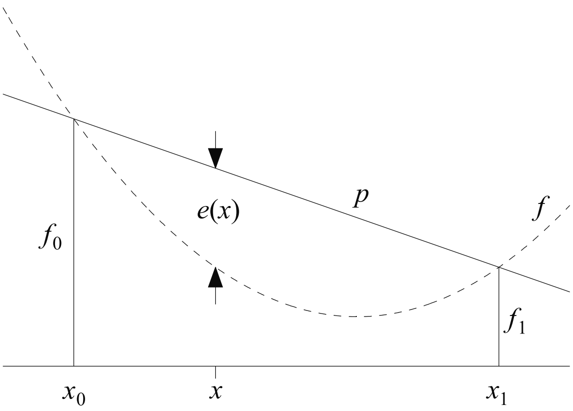


Fig. 2.3 Linear interpolation

This indeed is a prototype of Newton's form of the interpolation polynomial (cf. Sect. 2.2.6).

Interpolating to function values is referred to as *Lagrange interpolation*. More generally, we may wish to interpolate to function and consecutive derivative values of some function. This is called *Hermite interpolation*. It turns out that the latter can be solved as a limit case of the former (cf. Sect. 2.2.7).

2.2.1 Lagrange Interpolation Formula: Interpolation Operator

We prove the existence of the interpolation polynomial by simply writing it down. It is clear, indeed, that

$$\ell_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}, \quad i = 0, 1, \ldots, n, \tag{2.49}$$

is a polynomial of degree n that interpolates to 1 at $x = x_i$ and to 0 at all the other points. Multiplying it by f_i produces the correct value at x_i, and then adding up the resulting polynomials,

$$p(x) = \sum_{i=0}^{n} f_i \ell_i(x),$$

produces a polynomial, still of degree $\leq n$, that has the desired interpolation properties. To prove this formally, note that

$$\ell_i(x_k) = \delta_{ik} = \begin{cases} 1 & \text{if} \quad i = k, \\ 0 & \text{if} \quad i \neq k, \end{cases} \quad i,k = 0,1,\ldots,n. \tag{2.50}$$

Therefore,

$$p(x_k) = \sum_{i=0}^{n} f_i \ell_i(x_k) = \sum_{i=0}^{n} f_i \delta_{ik} = f_k, \;\; k = 0,1,\ldots,n.$$

This establishes the *existence* of the interpolation polynomial. To prove *uniqueness*, assume that there are two polynomials of degree $\leq n$, say, p and p^*, both interpolating to f at x_i, $i = 0,1,\ldots,n$. Then

$$d(x) = p(x) - p^*(x)$$

is a polynomial of degree $\leq n$ that satisfies

$$d(x_i) = f_i - f_i = 0, \;\; i = 0,1,\ldots,n.$$

In other words, d has $n+1$ distinct zeros x_i. There is only *one* polynomial in $\mathbb{P}_n$ with that many zeros, namely, $d(x) \equiv 0$. Therefore, $p^*(x) \equiv p(x)$.

We denote the unique polynomial $p \in \mathbb{P}_n$ interpolating f at the (distinct) points $x_0, x_1, \ldots, x_n$ by

$$p_n(f; x_0, x_1, \ldots, x_n; x) = p_n(f; x), \tag{2.51}$$

where we use the long form on the left-hand side if we want to place in evidence the points at which interpolation takes place, and the short form on the right-hand side if the choice of these points is clear from the context. We thus have what is called the *Lagrange*[4] *interpolation formula*

$$p_n(f; x) = \sum_{i=0}^{n} f(x_i)\ell_i(x), \tag{2.52}$$

with the $\ell_i(x)$ – the *elementary Lagrange interpolation polynomials* – defined in (2.49).

[4] Joseph Louis Lagrange (1736–1813), born in Turin, became, through correspondence with Euler, his protégé. In 1766 he indeed succeeded Euler in Berlin. He returned to Paris in 1787. Clairaut wrote of the young Lagrange: " … a young man, no less remarkable for his talents than for his modesty; his temperament is mild and melancholic; he knows no other pleasure than study." Lagrange made fundamental contributions to the calculus of variations and to number theory, and worked also on many problems in analysis. He is widely known for his representation of the remainder term in Taylor's formula. The interpolation formula appeared in 1794. His *Mécanique Analytique*, published in 1788, made him one of the founders of analytic mechanics.

It is useful to look at Lagrange interpolation in terms of a (linear) operator P_n from (say) the space of continuous functions to the space of polynomials $\mathbb{P}_n$,

$$P_n : \ C[a,b] \to \mathbb{P}_n, \ \ p(\cdot) = p_n(f; \cdot). \tag{2.53}$$

The interval $[a,b]$ here is any interval containing all points x_i, $i = 0, 1, \dots, n$. The operator P_n has the following properties:

1. $P_n(\alpha f) = \alpha P_n f, \ \ \alpha \in \mathbb{R}$ (homogeneity);
2. $P_n(f + g) = P_n f + P_n g$ (additivity).

Combining 1 and 2 shows that P_n is a linear operator,

$$P_n(\alpha f + \beta g) = \alpha P_n f + \beta P_n g, \ \ \alpha, \beta \in \mathbb{R}.$$

3. $P_n f = f$ for all $f \in \mathbb{P}_n$.

The last property – an immediate consequence of uniqueness of the interpolation polynomial – says that P_n leaves polynomials of degree $\leq n$ unchanged, and hence is a *projection* operator.

A norm of the linear operator P_n can be defined (similarly as for matrices, cf. Chap. 1, (1.30)) by

$$\|P_n\| = \max_{f \in C[a,b]} \frac{\|P_n f\|}{\|f\|}, \tag{2.54}$$

where on the right-hand side one takes any convenient norm for functions. Taking the L_∞ norm (cf. Table 2.1), one obtains from Lagrange's formula (2.52)

$$\begin{aligned} \|p_n(f; \cdot)\|_\infty &= \max_{a \leq x \leq b} \left| \sum_{i=0}^{n} f(x_i)\ell_i(x) \right| \\ &\leq \|f\|_\infty \max_{a \leq x \leq b} \sum_{i=0}^{n} |\ell_i(x)|. \end{aligned} \tag{2.55}$$

Indeed, equality holds for some continuous function f; cf. Ex. 30. Therefore,

$$\|P_n\|_\infty = \Lambda_n, \tag{2.56}$$

where

$$\Lambda_n = \|\lambda_n\|_\infty, \ \ \lambda_n(x) = \sum_{i=0}^{n} |\ell_i(x)|. \tag{2.57}$$

The function $\lambda_n(x)$ and its maximum Λ_n are called, respectively, the *Lebesgue*[5] *function* and *Lebesgue constant* for Lagrange interpolation. They provide a first estimate for the interpolation error: let $\mathcal{E}_n(f)$ be the *best* (uniform) *approximation* of f on $[a,b]$ by polynomials of degree $\leq n$,

$$\mathcal{E}_n(f) = \min_{p \in \mathbb{P}_n} \|f - p\|_\infty = \|f - \hat{p}_n\|_\infty, \tag{2.58}$$

where $\hat{p}_n$ is the nth-degree polynomial of best uniform approximation to f. Then, using the basic properties 1–3 of P_n, in particular, the projection property 3, and (2.55) and (2.57), one finds

$$\begin{aligned}\|f - p_n(f;\cdot)\|_\infty &= \|f - \hat{p}_n - p_n(f - \hat{p}_n;\cdot)\|_\infty \\ &\leq \|f - \hat{p}_n\|_\infty + \Lambda_n \|f - \hat{p}_n\|_\infty;\end{aligned}$$

that is,

$$\|f - p_n(f;\cdot)\|_\infty \leq (1 + \Lambda_n)\mathcal{E}_n(f). \tag{2.59}$$

Thus, the better f can be approximated by polynomials of degree $\leq n$, the smaller the interpolation error. Unfortunately, Λ_n is not uniformly bounded: no matter how one chooses the nodes $x_i = x_i^{(n)}$, $i = 0, 1, \ldots, n$, one can show that always $\Lambda_n > O(\log n)$ as $n \to \infty$. It is not possible, therefore, to conclude from Weierstrass's approximation theorem (i.e., from $\mathcal{E}_n(f) \to 0$, $n \to \infty$) that Lagrange interpolation converges uniformly on $[a,b]$ for any continuous function, not even for judiciously selected nodes; indeed, one knows that it does not.

2.2.2 *Interpolation Error*

As noted earlier, we need to make some assumptions about the function f in order to be able to estimate the error of interpolation, $f(x) - p_n(f;x)$, for any $x \neq x_i$ in $[a,b]$. In (2.59) we made an assumption in terms of how well f can be approximated on $[a,b]$ by polynomials of degree $\leq n$. Now we make an assumption on the magnitude of some appropriate derivative of f.

[5] Henri Leon Lebesgue (1875–1941) was a French mathematician best known for his work on the theory of real functions, notably the concepts of measure and integral that now bear his name. These became fundamental in many areas of mathematics such as functional analysis, Fourier analysis, and probability theory. He has also made interesting contributions to the calculus of variations, the theory of dimension, and set theory.

It is not difficult to guess how the formula for the error should look: since the error is zero at each x_i, $i = 0, 1, \dots, n$, we ought to see a factor of the form $(x-x_0)(x-x_1)\cdots(x-x_n)$. On the other hand, by the projection property 3 in Sect. 2.2.1, the error is also zero (even identically so) if $f \in \mathbb{P}_n$, which suggests another factor – the $(n+1)$st derivative of f. But evaluated where? Certainly not at x, since f would then have to satisfy a differential equation. So let us say that $f^{(n+1)}$ is evaluated at some point $\xi = \xi(x)$, which is unknown but must be expected to depend on x. Now if we test the formula so far conjectured on the simplest nontrivial polynomial, $f(x) = x^{n+1}$, we discover that a factor $1/(n+1)!$ is missing. So, our final (educated) guess is the formula

$$f(x) - p_n(f;x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}\prod_{i=0}^{n}(x-x_i), \quad x \in [a,b]. \tag{2.60}$$

Here $\xi(x)$ is some number in the open interval (a,b), but otherwise unspecified,

$$a < \xi(x) < b. \tag{2.61}$$

The statement (2.60) and (2.61) is, in fact, correct if we assume that $f \in C^{n+1}[a,b]$. An elegant proof of it, due to Cauchy,[6] goes as follows. We can assume $x \neq x_i$ for $i = 0, 1, \dots, n$, since otherwise (2.60) would be trivially true for any $\xi(x)$. So, fix $x \in [a,b]$ in this manner, and define a function F of the new variable t as follows:

$$F(t) = f(t) - p_n(f;t) - \frac{f(x) - p_n(f;x)}{\displaystyle\prod_{i=0}^{n}(x-x_i)}\prod_{i=0}^{n}(t-x_i). \tag{2.62}$$

Clearly, $F \in C^{n+1}[a,b]$. Furthermore,

$$F(x_i) = 0, \; i = 0, 1, \dots, n; \quad F(x) = 0.$$

Thus, F has $n+2$ distinct zeros in $[a,b]$. Applying repeatedly Rolle's Theorem, we conclude that

[6] Augustin Louis Cauchy (1789–1857), active in Paris, is truly the father of modern analysis. He provided a firm foundation for analysis by basing it on a rigorous concept of limit. He is also the creator of complex analysis, of which "Cauchy's formula" (cf. (2.70)) is a centerpiece. In addition, Cauchy's name is attached to pioneering contributions to the theory of ordinary and partial differential equations, in particular, regarding questions of existence and uniqueness. As with many great mathematicians of the eighteenth and nineteenth centuries, his work also encompasses geometry, algebra, number theory, and mechanics, as well as theoretical physics.

$$\begin{array}{lllll}
F' & \text{has at least} & n+1 & \text{distinct zeros} & \text{in } (a,b) \\
F'' & \text{has at least} & n & \text{distinct zeros} & \text{in } (a,b) \\
F''' & \text{has at least} & n-1 & \text{distinct zeros} & \text{in } (a,b) \\
\cdots & \cdots\cdots\cdots & \cdots & \cdots & \cdots\cdots\cdots \\
F^{(n+1)} & \text{has at least} & 1 & \text{zero} & \text{in } (a,b)
\end{array}$$

since $F^{(n+1)}$ is still continuous on $[a,b]$. Denote by $\xi(x)$ a zero of $F^{(n+1)}$ whose existence we just established. It certainly satisfies (2.61) and, of course, will depend on x. Now differentiating F in (2.62) $n+1$ times with respect to t, and then setting $t = \xi(x)$, we get

$$0 = f^{(n+1)}(\xi(x)) - \frac{f(x) - p_n(f;x)}{\prod_{i=0}^{n}(x - x_i)} \cdot (n+1)!,$$

which, when solved for $f(x) - p_n(f;x)$, gives precisely (2.60). Actually, what we have shown is that $\xi(x)$ is contained in the span of $x_0, x_1, \ldots, x_n, x$, that is, in the interior of the smallest closed interval containing $x_0, x_1, \ldots, x_n$ and x.

Examples. 1. Linear interpolation ($n = 1$). Assume that $x_0 \le x \le x_1$; that is, $[a,b] = [x_0, x_1]$, and let $h = x_1 - x_0$. Then by (2.60) and (2.61),

$$f(x) - p_1(f;x) = (x - x_0)(x - x_1)\frac{f''(\xi)}{2}, \quad x_0 < \xi < x_1,$$

and an easy computation gives

$$\|f - p_1(f;\cdot)\|_\infty \le \frac{M_2}{8}h^2, \quad M_2 = \|f''\|_\infty. \tag{2.63}$$

Here the ∞-norm refers to the interval $[x_0, x_1]$. Thus, on small intervals of length h, the error for linear interpolation is $O(h^2)$.

2. Quadratic interpolation ($n = 2$) on *equally spaced* points x_0, $x_1 = x_0 + h$, $x_2 = x_0 + 2h$. We now have, for $x \in [x_0, x_2]$,

$$f(x) - p_2(f;x) = (x - x_0)(x - x_1)(x - x_2)\frac{f'''(\xi)}{6}, \quad x_0 < \xi < x_2,$$

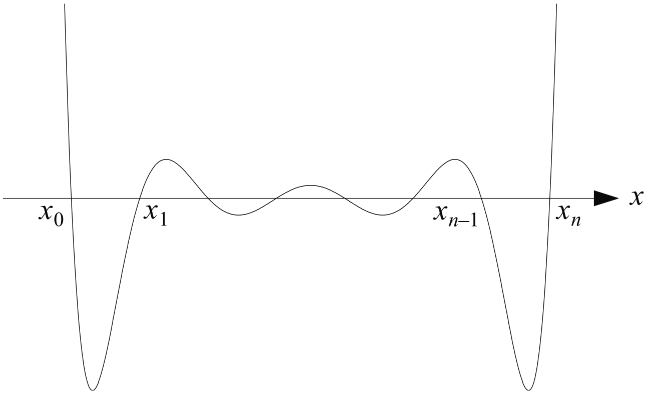


Fig. 2.4 Interpolation error for eight equally spaced points

and (cf. Ex. 43(a))

$$\|f - p_2(f;\cdot)\|_\infty \le \frac{M_3}{9\sqrt{3}}h^3, \quad M_3 = \|f'''\|_\infty,$$

giving an error of $O(h^3)$.

3. nth-degree interpolation on *equally spaced* points $x_i = x_0 + ih, i = 0, 1, \dots, n$. When h is small, and $x_0 \le x \le x_n$, then $\xi(x)$ in (2.60) is constrained to a relatively small interval and $f^{(n+1)}(\xi(x))$ cannot vary a great deal. The behavior of the error, therefore, is mainly determined by the product $\prod_{i=0}^{n}(x - x_i)$, the graph of which, for $n = 7$, is shown in Fig. 2.4. We clearly have symmetry with respect to the midpoint $(x_0 + x_n)/2$. It can also be shown that the relative extrema decrease monotonically in modulus as one moves from the endpoints to the center (cf. Ex. 29(c)).

   It is evident that the oscillations become more violent as n increases. In particular, the curve is extremely steep at the endpoints, and takes off to ∞ rapidly as x moves away from the interval $[x_0, x_n]$. Although it is true that the curve representing the interpolation error is scaled by a factor of $O(h^{n+1})$, it is also clear that one ought to interpolate near the center zone of the interval $[x_0, x_n]$, if at all possible, and should avoid interpolation near the end zones, or even *extra*polation outside the interval. The highly oscillatory nature of the error curve, when n is large, also casts some legitimate doubts about convergence of the interpolation process as $n \to \infty$. This is studied in the next section.

2.2.3 Convergence

We first must define what we mean by "convergence." We assume that we are given a triangular array of interpolation nodes $x_i = x_i^{(n)}$, exactly $n + 1$ distinct nodes for each $n = 0, 1, 2, \dots$:

$$\begin{array}{lllll} x_0^{(0)} & & & & \\ x_0^{(1)} & x_1^{(1)} & & & \\ x_0^{(2)} & x_1^{(2)} & x_2^{(2)} & & \\ \cdot\ \cdot\ \cdot & \cdot\ \cdot\ \cdot & \cdot\ \cdot\ \cdot & \cdot\ \cdot\ \cdot & \\ x_0^{(n)} & x_1^{(n)} & x_2^{(n)} & \cdots & x_n^{(n)} \\ \cdot\ \cdot\ \cdot & \cdot\ \cdot\ \cdot & \cdot\ \cdot\ \cdot & \cdot\ \cdot\ \cdot & \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \end{array} \tag{2.64}$$

We further assume that all nodes $x_i^{(n)}$ are contained in some finite interval $[a, b]$. Then, for each n, we define

$$p_n(x) = p_n(f; x_0^{(n)}, x_1^{(n)}, \dots, x_n^{(n)}; x), \quad x \in [a, b]. \tag{2.65}$$

We say that Lagrange interpolation based on the triangular array of nodes (2.64) *converges* if

$$p_n(x) \to f(x) \text{ as } n \to \infty, \tag{2.66}$$

uniformly for $x \in [a, b]$.

Convergence clearly depends on the behavior of the kth derivative $f^{(k)}$ of f as $k \to \infty$. We assume that $f \in C^\infty[a, b]$, and that

$$|f^{(k)}(x)| \le M_k \text{ for } a \le x \le b, \quad k = 0, 1, 2, \dots. \tag{2.67}$$

Since $|x - x_i^{(n)}| \le b - a$ whenever $x \in [a, b]$ and $x_i^{(n)} \in [a, b]$, we have

$$|(x - x_0^{(n)})(x - x_1^{(n)}) \cdots (x - x_n^{(n)})| \le (b - a)^{n+1}, \tag{2.68}$$

so that by (2.60)

$$|f(x) - p_n(x)| \le (b - a)^{n+1} \frac{M_{n+1}}{(n+1)!}, \quad x \in [a, b].$$

We therefore have convergence if

$$\lim_{k\to\infty} \frac{(b-a)^k}{k!} M_k = 0. \tag{2.69}$$

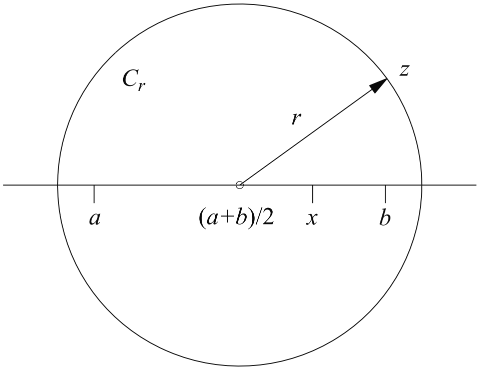


Fig. 2.5 The circular disk C_r

We now show that (2.69) is true if f is analytic in a sufficiently large region in the complex plane containing the interval $[a,b]$. Specifically, let C_r be the circular (closed) disk with center at the midpoint of $[a,b]$ and radius r, and assume, for the time being, that $r > \frac{1}{2}(b-a)$, so that $[a,b] \subset C_r$. Assume f analytic in C_r. Then we can estimate the derivative in (2.67) by Cauchy's Formula,

$$f^{(k)}(x) = \frac{k!}{2\pi \mathrm{i}} \oint_{\partial C_r} \frac{f(z)}{(z-x)^{k+1}}\,\mathrm{d}z, \quad x \in [a,b]. \tag{2.70}$$

Noting that $|z-x| \geq r - \frac{1}{2}(b-a)$ (cf. Fig. 2.5), we obtain

$$|f^{(k)}(x)| \leq \frac{k!}{2\pi} \frac{\max\limits_{z\in\partial C_r} |f(z)|}{[r-\frac{1}{2}(b-a)]^{k+1}} \cdot 2\pi r.$$

Therefore, we can take for M_k in (2.67)

$$M_k = \frac{r}{r-\frac{1}{2}(b-a)} \max_{z\in\partial C_r} |f(z)| \cdot \frac{k!}{[r-\frac{1}{2}(b-a)]^k}, \tag{2.71}$$

and (2.69) holds if

$$\left(\frac{b-a}{r-\frac{1}{2}(b-a)}\right)^k \to 0 \text{ as } k \to \infty,$$

that is, if $b-a < r - \frac{1}{2}(b-a)$, or, equivalently,

$$r > \tfrac{3}{2}(b-a). \tag{2.72}$$

We have shown that *Lagrange interpolation converges* (uniformly on $[a,b]$) *for an arbitrary triangular set of nodes* (2.64) (all contained in $[a,b]$) *if f is analytic in the circular disk C_r centered at $(a+b)/2$ and having radius r sufficiently large so that* (2.72) *holds*.

Since our derivation of this result used rather crude estimates (see, in particular, (2.68)), the required domain of analyticity for f that we found is certainly not sharp. Using more refined methods, one can prove the following. Let $\mathrm{d}\mu(t)$ be the "limit distribution" of the interpolation nodes, that is, let

$$\int_a^x \mathrm{d}\mu(t), \quad a < x \leq b,$$

be the ratio of the number of nodes $x_i^{(n)}$ in $[a,x]$ to the total number, $n+1$, of nodes, asymptotically as $n \to \infty$. (When the nodes are uniformly distributed over the interval $[a,b]$, then $\mathrm{d}\mu(t) = \mathrm{d}t/(b-a)$.) A curve of *constant logarithmic potential* is the locus of all complex $z \in \mathbb{C}$ such that

$$u(z) = \gamma, \quad u(z) = \int_a^b \ln \frac{1}{|z-t|}\,\mathrm{d}\mu(t),$$

where γ is a constant. For large negative γ, these curves look like circles with large radii and center at $(a+b)/2$. As γ increases, the curves "shrink" toward the interval $[a,b]$. Let

$$\Gamma = \sup \gamma,$$

where the supremum is taken over all curves $u(z) = \gamma$ containing $[a,b]$ in their interior. The important domain (replacing C_r) is then the domain

$$C_\Gamma = \{z \in \mathbb{C} : \ u(z) \geq \Gamma\}, \tag{2.73}$$

in the sense that *if f is analytic in any domain C containing C_Γ in its interior* (no matter how closely C covers C_Γ), *then*

$$|f(z) - p_n(f;z)| \to 0 \ \ as \ \ n \to \infty \tag{2.74}$$

uniformly for $z \in C_\Gamma$.

Examples. 1. Equally distributed nodes: $\mathrm{d}\mu(t) = \mathrm{d}t/(b-a)$, $a \leq t \leq b$. In this case, C_Γ is a lens-shaped domain with tips at a and b, as shown in Fig. 2.6. Thus, we have uniform convergence in C_Γ (not just on $[a,b]$, as before) provided f is analytic in a region slightly larger than C_Γ.

2. Arc sine distribution on [–1,1]: $\mathrm{d}\mu(t) = \frac{1}{\pi}\frac{\mathrm{d}t}{\sqrt{1-t^2}}$. Here the nodes are more densely distributed near the endpoints of the interval [–1,1]. It turns out that in this case C_Γ = [–1,1], so that Lagrange interpolation converges uniformly on [–1,1] if f is "analytic on [1,1]," that is, analytic in any region, no matter how thin, that contains the interval [–1,1] in its interior.

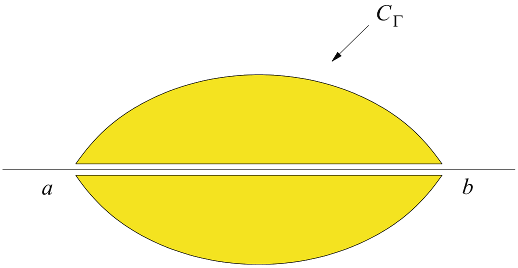


Fig. 2.6 The domain C_Γ for uniformly distributed nodes

3. Runge's[7] example:

$$f(x) = \frac{1}{1+x^2}, \quad -5 \le x \le 5,$$

$$x_k^{(n)} = -5 + k\frac{10}{n}, \quad k = 0, 1, 2, \ldots, n. \tag{2.75}$$

   Here the nodes are equally spaced, hence asymptotically equally distributed. Note that $f(z)$ has poles at $z = \pm\, \mathrm{i}$. These poles lie definitely inside the region C_Γ in Fig. 2.6 for the interval [–5,5], so that f is *not* analytic in C_Γ. For this reason, we can no longer expect convergence on the whole interval [–5,5]. It has been shown, indeed, that

$$\lim_{n\to\infty} |f(x) - p_n(f;x)| = \begin{cases} 0 & \text{if } |x| < 3.633\ldots, \\ \infty & \text{if } |x| > 3.633\ldots. \end{cases} \tag{2.76}$$

   We have convergence in the central zone of the interval [–5,5], but divergence in the lateral zones. With Fig. 2.4 kept in mind, this is perhaps not all that surprising (cf. MA 7(b)).

4. Bernstein's[8] example:

$$f(x) = |x|, \quad -1 \le x \le 1,$$

$$x_k^{(n)} = -1 + \frac{2k}{n}, \quad k = 0, 1, 2, \ldots, n. \tag{2.77}$$

[7] Carl David Tolme Runge (1856–1927) was active in the famous Göttingen school of mathematics and is one of the early pioneers of numerical mathematics. He is best known for the Runge–Kutta formula in ordinary differential equations (cf. Chap. 5, Sect. 5.6.5), for which he provided the basic idea. He made also notable contributions to approximation theory in the complex plane.

[8] Sergei Natanovič Bernštein (1880–1968) made major contributions to polynomial approximation, continuing in the tradition of his countryman Chebyshev. He is also known for his work on partial differential equations and probability theory.

Here analyticity of f is completely gone, f being not even differentiable at $x = 0$. Accordingly, one finds that

$$\lim_{n\to\infty} |f(x) - p_n(f;x)| = \infty \;\text{ for every }\; x \in [-1,1],$$
$$\text{except }\; x = -1,\; x = 0,\; \text{ and }\; x = 1. \tag{2.78}$$

The fact that $x = \pm\, 1$ are exceptional points is trivial, since they are interpolation nodes, where the error is zero. The same is true for $x = 0$ when n is even, but not if n is odd.

The failure of convergence in the last two examples can only in part be blamed on insufficient regularity of f. Another culprit is the equidistribution of the nodes. There are indeed better distributions, for example, the arc sine distribution of Example 2. An instance of the latter is discussed in the next section.

We add one more example, which involves *complex* nodes, and for which the preceding theory, therefore, no longer applies. We prove convergence directly.

5. Interpolation at the roots of unity (Fejér[9]): $z_k = \exp(k2\pi \mathrm{i}/n)$, $k = 1, 2, \ldots, n$. We show that

$$p_{n-1}(f;z) \to f(z),\;\; n \to \infty,\;\; \text{for any }\; |z| < 1, \tag{2.79}$$

uniformly in any disk $|z| \le \rho < 1$, provided f is analytic in $|z| < 1$ and continuous on $|z| \le 1$.

We have

$$\omega_n(z) := \prod_{k=1}^{n} (z - z_k) = z^n - 1,\;\; \omega_n'(z_k) = n z_k^{n-1} = \frac{n}{z_k},$$

so that the elementary Lagrange polynomials are

$$\ell_k(z) = \frac{\omega_n(z)}{\omega_n'(z_k)(z - z_k)} = \frac{z^n - 1}{\frac{n}{z_k}(z - z_k)}$$
$$= \frac{z_k}{n}\frac{1}{z_k - z} + z^n \frac{z_k}{(z - z_k)n}.$$

[9] Leopold Fejér (1880–1959) was a leading Hungarian mathematician of the twentieth century. Interestingly, Fejér had great difficulties in mathematics at the elementary and lower secondary school level, and even required private tutoring. It was an inspiring teacher in the upper-level secondary school who awoke Fejér's interest and passion for mathematics. He went on to discover – still a university student – an important result on the summability of Fourier series, which made him famous overnight. He continued to make further contributions to the theory of Fourier series, but also occupied himself with problems of approximation and interpolation in the real as well as complex domain. He in turn was an inspiring teacher to the next generation of Hungarian mathematicians.

Therefore,

$$p_{n-1}(f;z) = \sum_{k=1}^{n} \frac{f(z_k)}{z_k - z}\frac{z_k}{n} + z^n \sum_{k=1}^{n} \frac{f(z_k)}{z - z_k}\frac{z_k}{n}. \tag{2.80}$$

We interpret the first sum as a Riemann sum of an integral extended over the unit circle:

$$\sum_{k=1}^{n} \frac{f(z_k)}{z_k - z}\frac{z_k}{n} = \frac{1}{2\pi\mathrm{i}} \sum_{k=1}^{n} \frac{f(\mathrm{e}^{\mathrm{i}k2\pi/n})}{\mathrm{e}^{\mathrm{i}k2\pi/n} - z}\,\mathrm{i}\mathrm{e}^{\mathrm{i}k2\pi/n} \cdot \frac{2\pi}{n}$$

$$\to \frac{1}{2\pi\mathrm{i}} \int_0^{2\pi} \frac{f(\mathrm{e}^{\mathrm{i}\theta})}{\mathrm{e}^{\mathrm{i}\theta} - z}\,\mathrm{i}\mathrm{e}^{\mathrm{i}\theta}\mathrm{d}\theta = \frac{1}{2\pi\mathrm{i}} \oint_{|\zeta|=1} \frac{f(\zeta)\mathrm{d}\zeta}{\zeta - z} \quad \text{as } n \to \infty.$$

The last expression, by Cauchy's Formula, however, is precisely $f(z)$. The second term in (2.80), being just $-z^n$ times the first, converges to zero, uniformly in $|z| \le \rho < 1$.

2.2.4 Chebyshev Polynomials and Nodes

The choice of nodes, as we saw in the previous section, distinctly influences the convergence character of the interpolation process. We now discuss a choice of points – the *Chebyshev points* – which leads to very favorable convergence properties. These points are useful, not only for interpolation, but also for other purposes (integration, collocation, etc.). We consider them on the canonical interval [–1,1], but they can be defined on any finite interval $[a,b]$ by means of a linear transformation of variables that maps [–1,1] onto $[a,b]$.

We begin with developing the Chebyshev polynomials. They arise from the fact that the cosine of a multiple argument is a polynomial in the cosine of the simple argument; more precisely,

$$\cos n\theta = T_n(\cos\theta), \quad T_n \in \mathbb{P}_n. \tag{2.81}$$

This is a consequence of the well-known trigonometric identity

$$\cos(k+1)\theta + \cos(k-1)\theta = 2\cos\theta\cos k\theta,$$

which, when solved for the first term, gives

$$\cos(k+1)\theta = 2\cos\theta\cos k\theta - \cos(k-1)\theta. \tag{2.82}$$

Therefore, if $\cos m\theta$ is a polynomial of degree m in $\cos\theta$ for all $m \le k$, then the same is true for $m = k+1$. Mathematical induction then proves (2.81). At the same time, it follows from (2.81) and (2.82), if we set $\cos\theta = x$, that

$$\begin{aligned} T_{k+1}(x) &= 2xT_k(x) - T_{k-1}(x), \;\; k = 1, 2, 3, \ldots, \\ T_0(x) &= 1, \;\; T_1(x) = x. \end{aligned} \tag{2.83}$$

The polynomials T_m so defined are called the *Chebyshev polynomials* (of the first kind). Thus, for example,

$$T_2(x) = 2x^2 - 1,$$

$$T_3(x) = 4x^3 - 3x,$$

$$T_4(x) = 8x^4 - 8x^2 + 1,$$

and so on.

Clearly, these polynomials are defined not only for x in [–1,1], but also for arbitrary real or complex x. It is only that on the interval [–1,1] they satisfy the identity (2.81) (where θ is real).

It is evident from (2.83) that the leading coefficient of T_n is 2^{n-1} (if $n \geq 1$); the *monic Chebyshev polynomial* of degree n, therefore, is

$$\overset{\circ}{T}_n(x) = \frac{1}{2^{n-1}} T_n(x), \; n \geq 1; \quad \overset{\circ}{T}_0 = T_0. \tag{2.84}$$

The basic identity (2.81) allows us to immediately obtain the zeros $x_k = x_k^{(n)}$ of T_n: indeed, $\cos n\theta = 0$ if $n\theta = (2k-1)\pi/2$, so that

$$x_k^{(n)} = \cos \theta_k^{(n)}, \;\; \theta_k^{(n)} = \frac{2k-1}{2n}\pi, \;\; k = 1, 2, \ldots, n. \tag{2.85}$$

All zeros of T_n are thus real, distinct, and contained in the open interval (–1,1). They are the projections onto the real line of equally spaced points on the unit circle; cf. Fig. 2.7 for the case $n = 4$.

In terms of the zeros $x_k^{(n)}$ of T_n, we can write the monic polynomial in factored form as

$$\overset{\circ}{T}_n(x) = \prod_{k=1}^{n} \left(x - x_k^{(n)} \right). \tag{2.86}$$

As we let θ increase from 0 to π, hence $x = \cos\theta$ decrease from $+1$ to -1, (2.81) shows that $T_n(x)$ oscillates between +1 and –1, attaining these extreme values at

$$y_k^{(n)} = \cos \eta_k^{(n)}, \;\; \eta_k^{(n)} = k\frac{\pi}{n}, \;\; k = 0, 1, 2, \ldots, n. \tag{2.87}$$

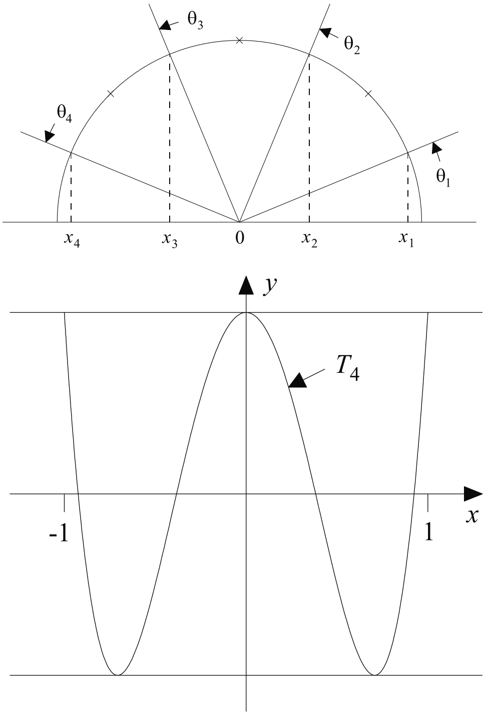


Fig. 2.7 The Chebyshev polynomial $y = T_4(x)$

In summary, then,

$$T_n\left(x_k^{(n)}\right) = 0 \text{ for } x_k^{(n)} = \cos\frac{2k-1}{2n}\pi, \quad k = 1, 2, \ldots, n; \tag{2.88}$$

$$T_n\left(y_k^{(n)}\right) = (-1)^k \text{ for } y_k^{(n)} = \cos\frac{k}{n}\pi, \quad k = 0, 1, 2, \ldots, n. \tag{2.89}$$

Chebyshev polynomials owe their importance and usefulness to the following theorem, due to Chebyshev.[10]

Theorem 2.2.1. *For an arbitrary monic polynomial $\overset{\circ}{p}_n$ of degree n, there holds*

$$\max_{-1\le x\le 1} |\overset{\circ}{p}_n(x)| \ge \max_{-1\le x\le 1} |\overset{\circ}{T}_n(x)| = \frac{1}{2^{n-1}}, \quad n \ge 1, \tag{2.90}$$

where $\overset{\circ}{T}_n$ is the monic Chebyshev polynomial (2.84) *of degree n.*

[10] Pafnuti Levovich Chebyshev (1821–1894) was the most prominent member of the St. Petersburg school of mathematics. He made pioneering contributions to number theory, probability theory, and approximation theory. He is regarded as the founder of constructive function theory, but also worked in mechanics, notably the theory of mechanisms, and in ballistics.

Proof (by contradiction). Assume, contrary to (2.90), that

$$\max_{-1\le x\le 1} |\overset{\circ}{p}_n(x)| < \frac{1}{2^{n-1}}. \tag{2.91}$$

Then the polynomial $d_n(x) = \overset{\circ}{T}_n(x) - \overset{\circ}{p}_n(x)$ (a polynomial of degree $\le n-1$), satisfies

$$d_n\left(y_0^{(n)}\right) > 0,\ d_n\left(y_1^{(n)}\right) < 0,\ d_n\left(y_2^{(n)}\right) > 0, \dots,\ (-1)^n d_n\left(y_n^{(n)}\right) > 0. \tag{2.92}$$

Thus d_n changes sign at least n times, and hence has at least n distinct real zeros. But having degree $\le n-1$, it must vanish identically, $d_n(x) \equiv 0$. This contradicts (2.92); thus (2.91) cannot be true. □

The result (2.90) can be given the following interesting interpretation: *the best uniform approximation* (on the interval [–1,1]) *to* $f(x) = x^n$ *from polynomials in* $\mathbb{P}_{n-1}$ *is given by* $x^n - \overset{\circ}{T}_n(x)$, that is, by the aggregate of terms of degree $\le n-1$ in $\overset{\circ}{T}_n$ taken with the minus sign. From the theory of uniform polynomial approximation it is known that the best approximant is unique. Therefore, equality in (2.90) can only hold if $\overset{\circ}{p}_n = \overset{\circ}{T}_n$.

What is the significance of Chebyshev polynomials for interpolation? Recall (cf. (2.60)) that the interpolation error (on [–1,1], for a function $f \in C^{n+1}[-1,1]$), is given by

$$f(x) - p_n(f;x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \cdot \prod_{i=0}^{n}(x - x_i),\quad x \in [-1,1]. \tag{2.93}$$

The first factor is essentially independent of the choice of the nodes x_i. It is true that $\xi(x)$ *does* depend on the x_i, but we usually estimate $f^{(n+1)}$ by $\|f^{(n+1)}\|_\infty$, which removes this dependence. On the other hand, the product in the second factor, including its norm

$$\left\| \prod_{i=0}^{n}(\cdot - x_i) \right\|_\infty, \tag{2.94}$$

depends strongly on the x_i. It makes sense, therefore, to try to minimize (2.94) over all $x_i \in [-1,1]$. Since the product in (2.94) is a monic polynomial of degree $n+1$, it follows from Theorem 2.2.1 that *the optimal nodes* $x_i = \hat{x}_i^{(n)}$ in (2.93) *are precisely the zeros of* T_{n+1}; that is,

$$\hat{x}_i^{(n)} = \cos\frac{2i+1}{2n+2}\pi,\quad i = 0, 1, 2, \dots, n. \tag{2.95}$$

For these nodes, we then have (cf. (2.90))

$$\|f(\cdot) - p_n(f;\cdot)\|_\infty \le \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} \cdot \frac{1}{2^n}. \tag{2.96}$$

One ought to compare the last factor in (2.96) with the much cruder bound given in (2.68), which, in the case of the interval [–1,1], is 2^{n+1}.

Since by (2.93) the error curve $y = f - p_n(f;\cdot)$ for Chebyshev points (2.95) is essentially *equilibrated* (modulo the variation in the factor $f^{(n+1)}$), and thus free of the violent oscillations we saw for equally spaced points, we would expect more favorable convergence properties for the triangular array (2.64) consisting of Chebyshev nodes. Indeed, one can prove, for example, that

$$p_n(f;\hat{x}_0^{(n)},\hat{x}_1^{(n)},\dots,\hat{x}_n^{(n)};x) \to f(x) \ \text{ as } \ n \to \infty, \tag{2.97}$$

uniformly on [–1,1], provided only that $f \in C^1[-1,1]$. Thus we do not need analyticity of f for (2.97) to hold.

We finally remark – as already suggested by the recurrence relation (2.83) – that Chebyshev polynomials are a special case of orthogonal polynomials. Indeed, the measure in question is precisely (up to an unimportant constant factor) the arc sine measure

$$\mathrm{d}\lambda(x) = \frac{\mathrm{d}x}{\sqrt{1-x^2}} \ \text{ on } \ [-1,1] \tag{2.98}$$

already mentioned in Example 2 of Sect. 2.1.4. This is easily verified from (2.81) and the orthogonality of the cosines (cf. Sect. 2.1.4, (2.33)):

$$\int_{-1}^{1} T_k(x)T_\ell(x)\frac{\mathrm{d}x}{\sqrt{1-x^2}} = \int_0^\pi T_k(\cos\theta)T_\ell(\cos\theta)\mathrm{d}\theta$$

$$= \int_0^\pi \cos k\theta \cos \ell\theta \mathrm{d}\theta = \begin{cases} 0 & \text{if} \quad k \ne \ell, \\ \pi & \text{if} \quad k = \ell = 0, \\ \frac{1}{2}\pi & \text{if} \quad k = \ell > 0. \end{cases} \tag{2.99}$$

The Fourier expansion in Chebyshev polynomials (essentially the Fourier cosine expansion) is therefore given by

$$f(x) = {\sum_{j=1}^{\infty}}' c_j T_j(x) = \frac{1}{2}c_0 + \sum_{j=1}^{\infty} c_j T_j(x), \tag{2.100}$$

where

$$c_j = \frac{2}{\pi}\int_{-1}^{1} f(x)T_j(x)\frac{\mathrm{d}x}{\sqrt{1-x^2}}, \quad j = 0, 1, 2, \ldots. \tag{2.101}$$

Truncating (2.100) with the term of degree n gives a useful polynomial approximation of degree n,

$$\tau_n(x) = {\sum_{j=0}^{n}}{}' \, c_j T_j(x),$$

having an error

$$f(x) - \tau_n(x) = \sum_{j=n+1}^{\infty} c_j T_j(x) \approx c_{n+1}T_{n+1}(x). \tag{2.102}$$

The approximation on the far right is better the faster the Fourier coefficients c_j tend to zero. The error (2.102), therefore, essentially oscillates between $+c_{n+1}$ and $-c_{n+1}$ as x varies on the interval [–1,1], and thus is of "uniform" size. This is in stark contrast to Taylor's expansion at $x = 0$, where the nth-degree partial sum has an error proportional to x^{n+1} on [–1,1].

2.2.5 Barycentric Formula

Lagrange's formula (2.52) is attractive more for theoretical purposes than for practical computational work. It can be rewritten, however, in a form that makes it efficient computationally, and that also allows additional interpolation nodes to be added with ease. Having the latter feature in mind, we now assume a sequential set $x_0, x_1, x_2, \ldots$ of interpolation nodes and denote by $p_n(f;\,\cdot)$ the polynomial of degree $\leq n$ interpolating to f at the first $n+1$ of them. We do not assume that the x_i are in any particular order, as long as they are mutually distinct.

We introduce a triangular array of auxiliary quantities defined by

$$\lambda_0^{(0)} = 1,\ \lambda_i^{(n)} = \prod_{\substack{j=0\\ j\neq i}}^{n} \frac{1}{x_i - x_j}, \quad i = 0, 1, \ldots, n; \quad n = 1, 2, 3, \ldots. \tag{2.103}$$

The elementary Lagrange interpolation polynomials of degree n, (2.49), can then be written in the form

$$\ell_i(x) = \frac{\lambda_i^{(n)}}{x - x_i}\,\omega_n(x),\ i = 0, 1, \ldots, n; \quad \omega_n(x) = \prod_{j=0}^{n}(x - x_j). \tag{2.104}$$

Dividing Lagrange's formula through by $1 \equiv \sum_{i=0}^{n} \ell_i(x)$, one finds

$$p_n(f;x) = \sum_{i=0}^{n} f_i \ell_i(x) = \frac{\displaystyle\sum_{i=0}^{n} f_i \ell_i(x)}{\displaystyle\sum_{i=0}^{n} \ell_i(x)} = \frac{\displaystyle\sum_{i=0}^{n} f_i \frac{\lambda_i^{(n)}}{x - x_i} \omega_n(x)}{\displaystyle\sum_{i=0}^{n} \frac{\lambda_i^{(n)}}{x - x_i} \omega_n(x)},$$

that is,

$$p_n(f;x) = \frac{\displaystyle\sum_{i=0}^{n} \frac{\lambda_i^{(n)}}{x - x_i} f_i}{\displaystyle\sum_{i=0}^{n} \frac{\lambda_i^{(n)}}{x - x_i}}, \quad x \neq x_i \text{ for } i = 0, 1, \ldots, n. \tag{2.105}$$

This expresses the interpolation polynomial as a weighted average of the function values $f_i = f(x_i)$ and is, therefore, called the *barycentric formula* – a slight misnomer, since the weights are not necessarily all positive. The auxiliary quantities $\lambda_i^{(n)}$ involved in (2.105) are those in the row numbered n of the triangular array (2.103). Once they have been calculated, the evaluation of $p_n(f;x)$ by (2.105), for any fixed x, is straightforward and cheap. Note, however, that when x is sufficiently close to some x_i, the right-hand side of (2.105) should be replaced by f_i.

Comparison with (2.52) shows that

$$\ell_i(x) = \frac{\dfrac{\lambda_i^{(n)}}{x - x_i}}{\displaystyle\sum_{j=0}^{n} \frac{\lambda_j^{(n)}}{x - x_j}}, \quad i = 0, 1, \ldots, n. \tag{2.106}$$

In order to arrive at an efficient algorithm for computing the required quantities $\lambda_i^{(n)}$, we first note that, for $k \geq 1$,

$$\lambda_i^{(k)} = \frac{\lambda_i^{(k-1)}}{x_i - x_k}, \quad i = 0, 1, \ldots, k-1. \tag{2.107}$$

The last quantity $\lambda_k^{(k)}$ missing in (2.107) is best computed directly from the definition (2.103),

$$\lambda_k^{(k)} = \frac{1}{\prod_{j=0}^{k-1}(x_k - x_j)}, \quad k \geq 1.$$

We thus arrive at the following algorithm:

$$\begin{aligned}
&\lambda_0^{(0)} = 1,\\
&\text{for } k = 1, 2, \ldots, n \ \text{ do}\\
&\left[\begin{aligned}
\lambda_i^{(k)} &= \frac{\lambda_i^{(k-1)}}{x_i - x_k}, \quad i = 0, 1, \ldots, k-1,\\
\lambda_k^{(k)} &= \frac{1}{\prod_{j=0}^{k-1}(x_k - x_j)}.
\end{aligned}\right.
\end{aligned} \tag{2.108}$$

This requires $\frac{1}{2}n(n+1)$ subtractions, $\frac{1}{2}(n-1)n$ multiplications, and $\frac{1}{2}n(n+3)$ divisions for computing the $n+1$ quantities $\lambda_0^{(n)}, \lambda_1^{(n)}, \ldots, \lambda_n^{(n)}$ in (2.105). Therefore, (2.106) in combination with (2.108) is more efficient than (2.49), which requires $O(n^3)$ operations to evaluate. It is also quite stable, since only benign arithmetic operations are involved (disregarding the formation of differences such as $x - x_i$, which occur in both formulae).

If we decide to incorporate the next data point (x_{n+1}, f_{n+1}), all we need to do is extend the k-loop in (2.108) through $n+1$, that is, generate the next row of auxiliary quantities $\lambda_0^{(n+1)}, \lambda_1^{(n+1)}, \ldots, \lambda_{n+1}^{(n+1)}$. We are then ready to compute $p_{n+1}(f;x)$ from (2.105) with n replaced by $n+1$.

2.2.6 Newton's[11] Formula

This is another way of organizing the work in Sect. 2.2.5. Although the computational effort remains essentially the same, it becomes easier to treat "confluent" interpolation points, that is, multiple points in which not only the function values, but also consecutive derivative values, are given (cf. Sect. 2.2.7).

Using the same setup as in Sect. 2.2.5, we denote

$$p_n(x) = p_n(f; x_0, x_1, \ldots, x_n; x), \quad n = 0, 1, 2, \ldots. \tag{2.109}$$

[11] Sir Isaac Newton (1643–1727) was an eminent figure of seventeenth century mathematics and physics. Not only did he lay the foundations of modern physics, but he was also one of the coinventors of differential calculus. Another was Leibniz, with whom he became entangled in a bitter and life-long priority dispute. His most influential work was the *Philosophiae Naturalis Principia Mathematica*, often called simply the *Principia*, one of the greatest work on physics and astronomy ever written. Therein one finds not only his ideas on interpolation, but also his suggestion to use the interpolating polynomial for purposes of integration (cf. Chap. 3, Sect. 3.2.2).

We clearly have

$$\begin{aligned} p_0(x) &= a_0, \\ p_n(x) &= p_{n-1}(x) + a_n(x-x_0)(x-x_1)\cdots(x-x_{n-1}), \\ n &= 1,2,3,\dots, \end{aligned} \tag{2.110}$$

for some constants a_0, a_1, $a_2, \dots$. This gives rise to a new form of the interpolation polynomial,

$$\begin{aligned} p_n(f;x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) \\ + \cdots + a_n(x-x_0)(x-x_1)\cdots(x-x_{n-1}), \end{aligned} \tag{2.111}$$

which is called *Newton's form*. The constants involved can be determined, in principle, by the interpolation conditions

$$\begin{aligned} f_0 &= a_0, \\ f_1 &= a_0 + a_1(x_1-x_0), \\ f_2 &= a_0 + a_1(x_2-x_0) + a_2(x_2-x_0)(x_2-x_1), \end{aligned}$$

and so on, which represent a triangular, nonsingular (why?) system of linear algebraic equations. This uniquely determines the constants; for example,

$$\begin{aligned} a_0 &= f_0, \\ a_1 &= \frac{f_1-f_0}{x_1-x_0}, \\ a_2 &= \frac{f_2-a_0-a_1(x_2-x_0)}{(x_2-x_0)(x_2-x_1)}, \end{aligned}$$

and so on. Evidently, a_n is a linear combination of f_0, $f_1, \dots, f_n$, with coefficients that depend on x_0, $x_1, \dots, x_n$. We use the notation

$$a_n = [x_0, x_1, \dots, x_n]f, \quad n = 0,1,2,\dots, \tag{2.112}$$

for this linear combination, and call the right-hand side the nth *divided difference* of f relative to the nodes x_0, $x_1, \dots, x_n$. Considered as a function of these $n+1$ variables, the divided difference is a *symmetric function*; that is, permuting the variables in any way does not affect the value of the function. This is a direct consequence of the fact that a_n in (2.111) is the *leading coefficient of* $p_n(f;x)$: the interpolation polynomial $p_n(f;\cdot)$ surely does not depend on the order in which we write down the interpolation conditions.

The name “divided difference” comes from the useful property

$$[x_0, x_1, x_2, \ldots, x_k]f = \frac{[x_1, x_2, \ldots, x_k]f - [x_0, x_1, \ldots, x_{k-1}]f}{x_k - x_0} \tag{2.113}$$

expressing the kth divided difference as a difference of $(k-1)$st divided differences, divided by a difference of the x_i. Since we have symmetry, the order in which the variables are written down is immaterial; what is important is that the two divided differences (of the same order $k-1$) in the numerator have $k-1$ of the x_i in common. The “extra” one in the first term, and the “extra” one in the second, are precisely the x_i that appear in the denominator, in the same order.

To prove (2.113), let

$$r(x) = p_{k-1}(f; x_1, x_2, \ldots, x_k; x)$$

and

$$s(x) = p_{k-1}(f; x_0, x_1, \ldots, x_{k-1}; x).$$

Then

$$p_k(f; x_0, x_1, \ldots, x_k; x) = r(x) + \frac{x - x_k}{x_k - x_0}[r(x) - s(x)]. \tag{2.114}$$

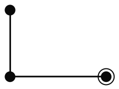
Indeed, the polynomial on the right-hand side has clearly degree $\le k$ and takes on the correct value f_i at x_i, $i = 0, 1, \ldots, k$. For example, if $i \ne 0$ and $i \ne k$,

$$r(x_i) + \frac{x_i - x_k}{x_k - x_0}[r(x_i) - s(x_i)] = f_i + \frac{x_i - x_k}{x_k - x_0}[f_i - f_i] = f_i,$$

and similarly for $i = 0$ and for $i = k$. By uniqueness of the interpolation polynomial, this implies (2.114). Now equating the leading coefficients on both sides of (2.114) immediately gives (2.113).

Equation (2.113) can be used to generate the *table of divided differences*:

$$\begin{array}{ll|lll}
x & f & & & \\
\hline
x_0 & f_0 & & & \\
x_1 & f_1 & [x_0, x_1]f & & \\
x_2 & f_2 & [x_1, x_2]f & [x_0, x_1, x_2]f & \\
x_3 & f_3 & [x_2, x_3]f & [x_1, x_2, x_3]f & [x_0, x_1, x_2, x_3]f \\
. & . & \cdots & \cdots & \cdots
\end{array} \tag{2.115}$$



The divided differences are here arranged in such a manner that their computation proceeds according to one single rule: *each entry is the difference of the entry immediately to the left and the one above it, divided by the difference of the x-value* *horizontally to the left and the one opposite the f-value found by going diagonally up*. Each entry, therefore, is calculated from its two neighbors immediately to the left, which is expressed by the computing stencil in (2.115).

The divided differences $a_0,\ a_1, \dots, a_n$ (cf. (2.112)) that occur in Newton's formula (2.111) are precisely the first $n + 1$ *diagonal entries* in the table of divided differences. Their computation requires $n(n + 1)$ additions and $\frac{1}{2}n(n + 1)$ divisions, essentially the same effort that was required in computing the auxiliary quantities $\lambda_i^{(n)}$ in the barycentric formula (cf. Ex. 61). Adding another data point $(x_{n+1},\ f_{n+1})$ requires the generation of the next line of divided differences. The last entry of this line is a_{n+1}, and we can update $p_n(f; x)$ by adding to it the term $a_{n+1}(x - x_0)(x - x_1) \cdots (x - x_n)$ to get p_{n+1} (cf. (2.110)).

Example.

x	f			
0	3			
1	4	(4–3)/(1–0) = 1		
2	7	(7–4)/(2–1) = 3	(3–1)/(2–0) = 1	
4	19	(19–7)/(4–2) = 6	(6–3)/(4–1) = 1	(1–1)/(4–0) = 0

The cubic interpolation polynomial is

$$\begin{aligned} p_3(f;x) &= 3 + 1 \cdot (x - 0) + 1 \cdot (x - 0)(x - 1) + 0 \cdot (x - 0)(x - 1)(x - 2) \\ &= 3 + x + x(x - 1) = 3 + x^2, \end{aligned}$$

which indeed is the function tabulated. Note that the leading coefficient of $p_3(f;\,\cdot)$ is zero, which is why the last divided difference turned out to be 0.

Newton's formula also yields a new representation for the error term in Lagrange interpolation. Let t temporarily denote an arbitrary "node" not equal to any of the $x_0, x_1, \dots, x_n$. Then we have,

$$\begin{aligned} &p_{n+1}(f; x_0, x_1, \dots, x_n, t; x) \\ &\quad = p_n(f;x) + [x_0, x_1, \dots, x_n, t]f \cdot \prod_{i=0}^{n}(x - x_i). \end{aligned}$$

Now put $x = t$; since the polynomial on the left-hand side interpolates to f at t, we get

$$f(t) = p_n(f;t) + [x_0, x_1, \dots, x_n, t]f \cdot \prod_{i=0}^{n}(t - x_i).$$

Writing again x for t (which was arbitrary, after all), we find

$$f(x) - p_n(f;x) = [x_0, x_1, \dots, x_n, x]f \cdot \prod_{i=0}^{n}(x - x_i). \tag{2.116}$$

This is the new formula for the interpolation error. Note that it involves no derivative of f, only function values. The trouble is, that $f(x)$ is one of them. Indeed, (2.116) is basically a tautology since, when everything is written out explicitly, the formula evaporates to $0 = 0$, which is correct, but not overly exciting.

In spite of this seeming emptiness of (2.116), we can draw from it an interesting and very useful conclusion. (For another application, see Chap. 3, Ex. 2.) Indeed, compare it with the earlier formula (2.60); one obtains

$$[x_0, x_1, \ldots, x_n, x]f = \frac{f^{(n+1)}(\xi(x))}{(n+1)!},$$

where $x_0, x_1, \ldots, x_n, x$ are arbitrary distinct points in $[a,b]$ and $f \in C^{n+1}[a,b]$. Moreover, $\xi(x)$ is strictly between the smallest and largest of these points (cf. the proof of (2.60)). We can now write $x = x_{n+1}$, and then replace $n+1$ by n to get

$$[x_0, x_1, \ldots, x_n]f = \frac{1}{n!} f^{(n)}(\xi). \tag{2.117}$$

Thus, for any $n+1$ distinct points in $[a,b]$ and any $f \in C^n[a,b]$, *the divided difference of f of order n is the* nth *scaled derivative of f at some* (unknown) *intermediate point*. If we now let all x_i, $i \geq 1$, tend to x_0, then ξ, being trapped between them, must also tend to x_0, and, since $f^{(n)}$ is continuous at x_0, we obtain

$$[\underbrace{x_0, x_0, \ldots, x_0}_{n+1 \text{ times}}]f = \frac{1}{n!} f^{(n)}(x_0). \tag{2.118}$$

This suggests that *the* nth *divided difference at* n+1 *"confluent"* (i.e., identical) *points be defined to be the* nth *derivative at this point divided by* $n!$. This allows us, in the next section, to solve the Hermite interpolation problem.

2.2.7 Hermite[12] Interpolation

The general Hermite interpolation problem consists of the following: given $K+1$ distinct points $x_0, x_1, \ldots, x_K$ in $[a,b]$ and corresponding integers $m_k \geq 1$, and given a function $f \in C^{M-1}[a,b]$, with $M = \max_k m_k$, find a polynomial p of lowest degree such that, for $k = 0, 1, \ldots, K$,

$$p^{(\mu)}(x_k) = f_k^{(\mu)}, \quad \mu = 0, 1, \ldots, m_k - 1, \tag{2.119}$$

where $f_k^{(\mu)} = f^{(\mu)}(x_k)$ is the μth derivative of f at x_k.

[12] Charles Hermite (1822–1901) was a leading French mathematician. An Academician in Paris, known for his extensive work in number theory, algebra, and analysis, he is famous for his proof in 1873 of the transcendental nature of the number e. He was also a mentor of the Dutch mathematician Stieltjes.

The problem can be thought of as a limiting case of Lagrange interpolation if we consider x_k to be a *point of multiplicity* m_k, that is, obtained by a confluence of m_k distinct points into a single point x_k. We can imagine setting up the table of divided differences, and Newton's interpolation formula, just before the confluence takes place, and then simply "go to the limit." To do this in practice requires that each point x_k be entered exactly m_k times in the first column of the table of divided differences. The formula (2.118) then allows us to *initialize* the divided differences for these points. For example, if $m_k = 4$, then

$$\begin{array}{lllll} x & f & & & \\ \cline{1-2} \cdot & \cdot & \cdot & \cdot & \cdot \\ x_k & f_k & & & \\ x_k & f_k & f_k' & & \\ x_k & f_k & f_k' & \frac{1}{2} f_k'' & \\ x_k & f_k & f_k' & \frac{1}{2} f_k'' & \frac{1}{6} f_k''' \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{array} \tag{2.120}$$

Doing this initialization for each k, we are then ready to complete the table of divided differences in the usual way. (There will be no zero divisors; they have been taken care of during the initialization.) We obtain a table with $m_0 + m_1 + \cdots + m_K$ entries in the first column, and hence an interpolation polynomial of degree $\leq n = m_0 + m_1 + \cdots + m_K - 1$, which, as in the Lagrange case, is unique. The $n + 1$ diagonal entries in the table give us the coefficients in Newton's formula, as before, except that in the product terms of the formula, some of the factors are repeated. Also the error term of interpolation remains in force, with the repetition of factors properly accounted for.

We illustrate the procedure with two simple examples.

1. Find $p \in \mathbb{P}_3$ such that

$$p(x_0) = f_0,\ p'(x_0) = f_0',\ p''(x_0) = f_0'',\ p'''(x_0) = f_0'''.$$

Here $K = 0$, $m_0 = 4$, that is, we have a single quadruple point. The table of divided differences is precisely the one in (2.120) (with $k = 0$); hence Newton's formula becomes

$$p(x) = f_0 + (x - x_0) f_0' + \frac{1}{2}(x - x_0)^2 f_0'' + \frac{1}{6}(x - x_0)^3 f_0''',$$

which is nothing but the Taylor polynomial of degree 3. Thus Taylor's polynomial is a special case of a Hermite interpolation polynomial. The error term of interpolation, furthermore, gives us

$$f(x) - p(x) = \frac{1}{24}(x - x_0)^4 f^{(4)}(\xi), \quad \xi \text{ between } x_0 \text{ and } x,$$

which is Lagrange's form of the remainder term in Taylor's formula.

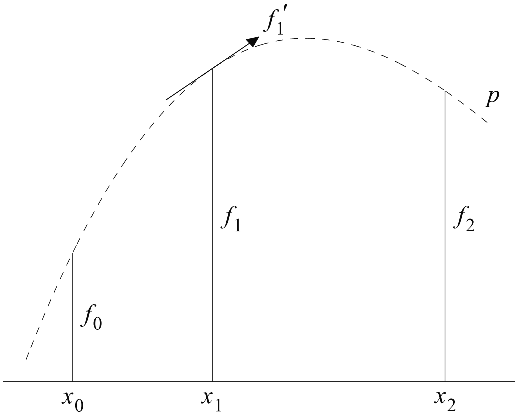


Fig. 2.8 A Hermite interpolation problem

2. Find $p \in \mathbb{P}_3$ such that

$$p(x_0) = f_0, \;\; p(x_1) = f_1, \;\; p'(x_1) = f_1', \;\; p(x_2) = f_2,$$

where $x_0 < x_1 < x_2$ (cf. Fig. 2.8).

The table of divided differences now has the form:

x	f			
x_0	f_0			
x_1	f_1	$[x_0, x_1]f$		
x_1	f_1	f_1'	$[x_0, x_1, x_1]f$	
x_2	f_2	$[x_1, x_2]f$	$[x_1, x_1, x_2]f$	$[x_0, x_1, x_1, x_2]f$.

If we denote the diagonal entries, as before, by a_0, a_1, a_2, a_3, Newton's formula takes the form

$$p(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + a_3(x - x_0)(x - x_1)^2,$$

and the error formula becomes

$$f(x) - p(x) = (x - x_0)(x - x_1)^2(x - x_2)\frac{f^{(4)}(\xi)}{4!}, \;\; x_0 < \xi < x_2.$$

For equally spaced points, say, $x_0 = x_1 - h$, $x_2 = x_1 + h$, we have, if $x = x_1 + th$, $-1 \le t \le 1$,

$$|(x - x_0)(x - x_1)^2(x - x_2)| = |(t^2 - 1)t^2 \cdot h^4| \le \tfrac{1}{4}h^4,$$

and so

$$\|f - p\|_\infty \le \frac{1}{4} h^4 \frac{\|f^{(4)}\|_\infty}{24} = \frac{h^4}{96} \|f^{(4)}\|_\infty,$$

with the ∞-norm referring to the interval $[x_0, x_2]$.

2.2.8 Inverse Interpolation

An interesting application of interpolation – and, in particular, of Newton's formula – is to the solution of a nonlinear equation,

$$f(x) = 0. \tag{2.121}$$

Here f is a given (nonlinear) function, and we are interested in a root α of the equation for which we already have two approximations,

$$x_0 \approx \alpha, \quad x_1 \approx \alpha.$$

We assume further that near the root α, the function f is monotone, so that

$$y = f(x) \text{ has an inverse } x = f^{-1}(y).$$

Denote, for short,

$$g(y) = f^{-1}(y).$$

Since $\alpha = g(0)$, our problem is to evaluate $g(0)$. From our two approximations, we can compute $y_0 = f(x_0)$ and $y_1 = f(x_1)$, giving $x_0 = g(y_0)$, $x_1 = g(y_1)$. Hence, we can start a table of divided differences for the inverse function g:

y	g	
y_0	x_0	
y_1	x_1	$[y_0, y_1]g$

Wanting to compute $g(0)$, we can get a first improved approximation by linear interpolation,

$$x_2 = x_0 + (0 - y_0)[y_0, y_1]g = x_0 - y_0[y_0, y_1]g.$$

Now evaluating $y_2 = f(x_2)$, we get $x_2 = g(y_2)$. Hence, the table of divided differences can be updated and becomes

y	g		
y_0	x_0		
y_1	x_1	$[y_0, y_1]g$	
y_2	x_2	$[y_1, y_2]g$	$[y_0, y_1, y_2]g$

This allows us to use quadratic interpolation to get, again with Newton's formula,

$$x_3 = x_2 + (0 - y_0)(0 - y_1)[y_0, y_1, y_2]g = x_2 + y_0 y_1 [y_0, y_1, y_2]g$$

and then

$$y_3 = f(x_3), \quad \text{and} \quad x_3 = g(y_3).$$

Since y_0, y_1 are small, the product $y_0 y_1$ is even smaller, making the correction term added to the linear interpolant x_2 quite small. If necessary, we can continue updating the difference table,

y	g			
y_0	x_0			
y_1	x_1	$[y_0, y_1]g$		
y_2	x_2	$[y_1, y_2]g$	$[y_0, y_1, y_2]g$	
y_3	x_3	$[y_2, y_3]g$	$[y_1, y_2, y_3]g$	$[y_0, y_1, y_2, y_3]g$

and computing

$$x_4 = x_3 - y_0 y_1 y_2 [y_0, y_1, y_2, y_3]g, \quad y_4 = f(x_4), \quad x_4 = g(y_4),$$

giving us another data point to generate the next row of divided differences, and so on. In general, the process will converge rapidly: $x_k \to \alpha$ as $k \to \infty$. The precise analysis of convergence, however, is not simple because of the complicated structure of the successive derivatives of the inverse function $g = f^{-1}$.

2.3 Approximation and Interpolation by Spline Functions

Our concern in Sect. 2.1.1 was with approximation of functions by a single polynomial over a finite interval $[a,b]$. When more accuracy was wanted, we simply increased the degree of the polynomial, and under suitable assumptions the approximation indeed can be made as accurate as one wishes by choosing the degree of the approximating polynomial sufficiently large.

However, there are other ways to control accuracy. One is to impose a subdivision Δ upon the interval $[a,b]$,

$$\Delta: \ \ a = x_1 < x_2 < x_3 < \cdots < x_{n-1} < x_n = b, \tag{2.122}$$

and use *low-degree* polynomials on each subinterval $[x_i, x_{i+1}]$ ($i = 1,2,\ldots, n-1$) to approximate the given function. The rationale behind this is the recognition that on a sufficiently small interval, functions can be approximated arbitrarily well by polynomials of low degree, even degree 1, or zero, for that matter. Thus, measuring the "fineness" of the subdivision Δ by

$$|\Delta| = \max_{1\le i\le n-1} \Delta x_i, \ \ \Delta x_i = x_{i+1} - x_i, \tag{2.123}$$

we try to control (increase) the accuracy by varying (decreasing) $|\Delta|$, keeping the degrees of the polynomial pieces uniformly low.

To discuss these approximation processes, we make use of the class of functions (cf. Example 2 at the beginning of Chap. 2)

$$\mathbb{S}_m^k(\Delta) = \left\{ s : \ s \ \in C^k[a,b], \ s\big|_{[x_i,x_{i+1}]} \in \mathbb{P}_m, \ \ i = 1,2,\ldots,n-1 \right\}, \tag{2.124}$$

where $m \ge 0, k \ge 0$ are given nonnegative integers. We refer to $\mathbb{S}_m^k(\Delta)$ as the *spline functions of degree m and smoothness class k* relative to the subdivision Δ. (If the subdivision is understood from the context, we omit Δ in the notation on the left-hand side of (2.124).) The point in the continuity assumption of (2.124), of course, is that the kth derivative of s is to be continuous *everywhere* on $[a,b]$, in particular, also at the subdivision points x_i ($i = 2,\ldots,n-1$) of Δ. One extreme case is $k = m$, in which case $s \in \mathbb{S}_m^m$ necessarily consists of just one single polynomial of degree m on the whole interval $[a,b]$; that is, $\mathbb{S}_m^m = \mathbb{P}_m$ (see Ex. 68). Since we want to get away from $\mathbb{P}_m$, we assume $k < m$. The other extreme is the case where no continuity at all (at the subdivision points x_i) is required; we then put $k = -1$. Thus $\mathbb{S}_m^{-1}(\Delta)$ is the class of piecewise polynomials of degree $\le m$, where the polynomial pieces can be completely disjoint (see Fig. 2.9).

We begin with the simplest case – piecewise linear approximation – that is, the case $m = 1$ (hence $k = 0$).

2.3.1 Interpolation by Piecewise Linear Functions

The problem here is to find an $s \in \mathbb{S}_1^0(\Delta)$ such that, for a given function f defined on $[a,b]$, we have

$$s(x_i) = f_i \ \text{ where } \ f_i = f(x_i), \ \ i = 1,2,\ldots,n. \tag{2.125}$$

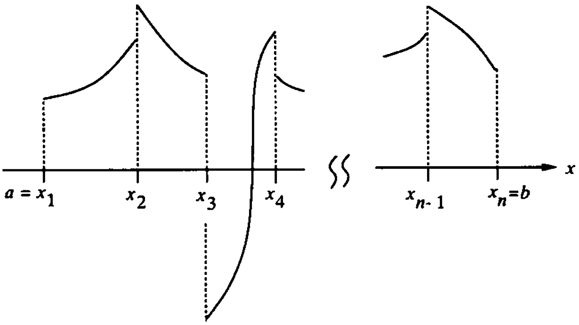


Fig. 2.9 A function $s \in \mathbb{S}_m^{-1}$

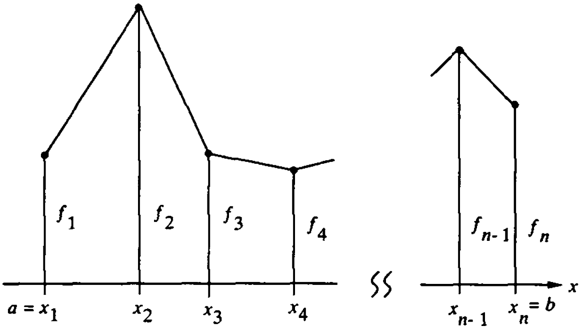


Fig. 2.10 Piecewise linear interpolation.

We conveniently let the interpolation nodes coincide with the points x_i of the subdivision Δ in (2.122). This simplifies matters, but is not necessary (cf. Ex. 75). The solution then indeed is trivial; see Fig. 2.10. If we denote the (obviously unique) interpolant by $s(\cdot) = s_1(f;\cdot)$, then the formula of linear interpolation gives

$$s_1(f;x) = f_i + (x - x_i)[x_i, x_{i+1}]f \quad \text{for } x_i \le x \le x_{i+1}, \quad i = 1, 2, \dots, n-1. \tag{2.126}$$

A bit more interesting is the analysis of the error. This, too, however, is quite straightforward, once we note that $s_1(f;\cdot)$ on $[x_i, x_{i+1}]$ is simply the linear interpolant to f. Thus, from the theory of (linear) interpolation,

$$f(x) - s_1(f;x) = (x - x_i)(x - x_{i+1})[x_i, x_{i+1}, x]f \quad \text{for } x \in [x_i, x_{i+1}];$$

hence, if $f \in C^2[a,b]$,

$$|f(x) - s_1(f;x)| \le \frac{(\Delta x_i)^2}{8} \max_{[x_i, x_{i+1}]} |f''|, \quad x \in [x_i, x_{i+1}].$$

It then follows immediately that

$$\|f(\cdot) - s_1(f;\,\cdot)\|_\infty \le \tfrac{1}{8}\,|\Delta|^2 \|f''\|_\infty, \tag{2.127}$$

where the maximum norms are those on $[a,b]$; that is, $\|g\|_\infty = \max\limits_{[a,b]} |g|$. This shows that the error indeed can be made arbitrarily small, uniformly on $[a,b]$, by taking $|\Delta|$ sufficiently small. Making $|\Delta|$ smaller, of course, increases the number of polynomial pieces, and with it, the volume of data.

It is easy to show (see Ex. 80(b)) that

$$\operatorname{dist}_\infty(f, \mathbb{S}_1^0) \le \|f(\cdot) - s_1(f;\,\cdot)\|_\infty \le 2\operatorname{dist}_\infty(f, \mathbb{S}_1^0), \tag{2.128}$$

where, for any set of functions $\mathbb{S}$,

$$\operatorname{dist}_\infty(f, \mathbb{S}) := \inf_{s\in\mathbb{S}} \|f - s\|_\infty.$$

In other words, the piecewise linear interpolant $s_1(f;\,\cdot)$ is a nearly optimal approximation, its error differing from the error of the best approximant to f from $\mathbb{S}_1^0$ by at most a factor of 2.

2.3.2 A Basis for $\mathbb{S}_1^0(\Delta)$

What is the dimension of the space $\mathbb{S}_1^0(\Delta)$? In other words, how many degrees of freedom do we have? If, for the moment, we ignore the continuity requirement (i.e., if we look at $\mathbb{S}_1^{-1}(\Delta)$), then each linear piece has two degrees of freedom, and there are $n-1$ pieces; so $\dim \mathbb{S}_1^{-1}(\Delta) = 2n-2$. Each continuity requirement imposes one equation, and hence reduces the degree of freedom by 1. Since continuity must be enforced only at the interior subdivision points x_i, $i = 2,\dots,n-1$, we find that $\dim \mathbb{S}_1^0(\Delta) = 2n-2-(n-2) = n$. So we expect that a basis of $\mathbb{S}_1^0(\Delta)$ must consist of exactly n basis functions.

We now define n such functions. For notational convenience, we let $x_0 = x_1$ and $x_{n+1} = x_n$; then, for $i = 1,2,\dots,n$, we define

$$B_i(x) = \begin{cases} \dfrac{x - x_{i-1}}{x_i - x_{i-1}} & \text{if } x_{i-1} \le x \le x_i, \\[2ex] \dfrac{x_{i+1} - x}{x_{i+1} - x_i} & \text{if } x_i \le x \le x_{i+1}, \\[2ex] 0 & \text{otherwise.} \end{cases} \tag{2.129}$$

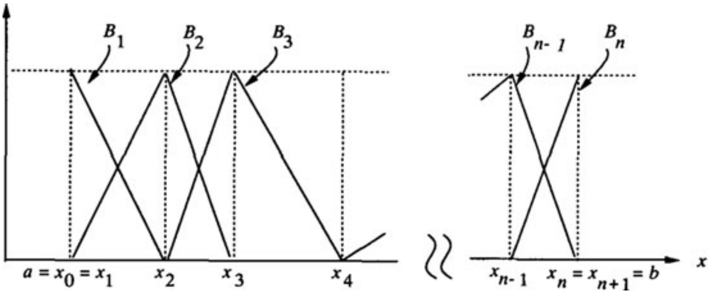


Fig. 2.11 The functions B_i

Note that the first equation, when $i = 1$, and the second, when $i = n$, are to be ignored, since x in both cases is restricted to a single point and the ratio in question has the meaningless form 0/0. (It is the other ratio that provides the necessary information in these cases.) The functions B_i may be referred to as "hat functions" (Chinese hats), but note that the first and last hat is cut in half. The functions B_i are depicted in Fig. 2.11. We expect these functions to form a basis of $\mathbb{S}_1^0(\Delta)$. To prove this, we must show:

(a) the functions $\{B_i\}_{i=1}^n$ are linearly independent and
(b) they span the space $\mathbb{S}_1^0(\Delta)$.

Both these properties follow from the basic fact that

$$B_i(x_j) = \delta_{ij} = \begin{cases} 1 & \text{if} \quad i = j, \\ \\ 0 & \text{if} \quad i \neq j, \end{cases} \tag{2.130}$$

which one easily reads from Fig. 2.11. To show (a), assume there is a linear combination of the B_i that vanishes identically on $[a,b]$,

$$s(x) = \sum_{i=1}^{n} c_i B_i(x), \quad s(x) \equiv 0 \ \text{ on } \ [a, b]. \tag{2.131}$$

Putting $x = x_j$ in (2.131) and using (2.130) then gives $c_j = 0$. Since this holds for each $j = 1, 2, \dots, n$, we see that only the trivial linear combination (with all $c_i = 0$) can vanish identically. To prove (b), let $s \in \mathbb{S}_1^0(\Delta)$ be given arbitrarily. We must show that s can be represented as a linear combination of the B_i. We claim that, indeed,

$$s(x) = \sum_{i=1}^{n} s(x_i) B_i(x). \tag{2.132}$$

This is so, because the function on the right-hand side has the same values as s at each x_j, and therefore, being in $\mathbb{S}_1^0(\Delta)$, must coincide with s.

Equation (2.132), which holds for every $s \in \mathbb{S}_1^0(\Delta)$, may be thought of as the analogue of the Lagrange interpolation formula for polynomials. The role of the elementary Lagrange polynomials ℓ_i is now played by the B_i.

2.3.3 Least Squares Approximation

As an application of the basis $\{B_i\}$, we consider the problem of least squares approximation on $[a,b]$ by functions in $\mathbb{S}_1^0(\Delta)$. The discrete L_2 approximation problem with data given at the points x_i ($i = 1, 2, \dots, n$), of course, has the trivial solution $s_1(f;\cdot)$, which drives the error to zero at each data point. We therefore consider only the continuous problem: given $f \in C[a,b]$, find $\hat{s}_1(f;\cdot) \in \mathbb{S}_1^0(\Delta)$ such that

$$\int_a^b [f(x) - \hat{s}_1(f;x)]^2 \mathrm{d}x \le \int_a^b [f(x) - s(x)]^2 \mathrm{d}x \;\text{ for all }\; s \in \mathbb{S}_1^0(\Delta). \tag{2.133}$$

Writing

$$\hat{s}_1(f;x) = \sum_{i=1}^n \hat{c}_i B_i(x), \tag{2.134}$$

we know from the general theory of Sect. 2.1 that the coefficients $\hat{c}_i$ must satisfy the normal equations

$$\sum_{j=1}^n \left[\int_a^b B_i(x) B_j(x) \mathrm{d}x \right] \hat{c}_j = \int_a^b B_i(x) f(x) \mathrm{d}x, \;\; i = 1, 2, \dots, n. \tag{2.135}$$

Now the fact that B_i is nonzero only on (x_{i-1}, x_{i+1}) implies that $\int_a^b B_i(x) \cdot B_j(x)\mathrm{d}x = 0$ if $|i - j| > 1$; that is, the system (2.135) is tridiagonal. An easy computation (cf. Ex. 77) indeed yields

$$\frac{1}{6}\Delta x_{i-1} \cdot \hat{c}_{i-1} + \frac{1}{3}(\Delta x_{i-1} + \Delta x_i)\hat{c}_i + \frac{1}{6}\Delta x_i \cdot \hat{c}_{i+1} = b_i, \; i = 1, 2, \dots, n, \tag{2.136}$$

where $b_i = \int_a^b B_i(x) f(x) \mathrm{d}x = \int_{x_{i-1}}^{x_{i+1}} B_i(x) f(x) \mathrm{d}x$. Note, by our convention, that $\Delta x_0 = 0$ and $\Delta x_n = 0$, so that (2.136) is in fact a tridiagonal system for the unknowns $\hat{c}_1, \hat{c}_2, \dots, \hat{c}_n$. Its matrix is given by

$$\begin{bmatrix} \frac{1}{3}\Delta x_1 & \frac{1}{6}\Delta x_1 & & & 0 \\ \frac{1}{6}\Delta x_1 & \frac{1}{3}(\Delta x_1 + \Delta x_2) & \frac{1}{6}\Delta x_2 & & \\ & \frac{1}{6}\Delta x_2 & & \ddots & \\ & & \ddots & \ddots & \frac{1}{6}\Delta x_{n-1} \\ 0 & & & \frac{1}{6}\Delta x_{n-1} & \frac{1}{3}\Delta x_{n-1} \end{bmatrix}.$$

As it must be, by the general theory of Sect. 2.1, the matrix is symmetric and positive definite, but it is also diagonally dominant, each diagonal element exceeding by a factor of 2 the sum of the (positive) off-diagonal elements in the same row. The system (2.136) can therefore be solved easily, rapidly, and accurately by the Gauss elimination procedure, and there is no need for pivoting.

Like the interpolant $s_1(f;\,\cdot\,)$, the least squares approximant $\hat{s}_1(f;\,\cdot\,)$, too, can be shown to be nearly optimal, in that

$$\operatorname{dist}_\infty(f, \mathbb{S}_1^0) \le \|f(\,\cdot\,) - \hat{s}_1(f;\,\cdot\,)\|_\infty \le 4\,\operatorname{dist}_\infty(f, \mathbb{S}_1^0). \tag{2.137}$$

The spread is now by a factor of 4, rather than 2, as in (2.128).

2.3.4 Interpolation by Cubic Splines

The most widely used splines are cubic splines, in particular, cubic spline interpolants. We first discuss the interpolation problem for splines $s \in \mathbb{S}_3^1(\Delta)$. Continuity of the first derivative of any cubic spline interpolant $s_3(f;\,\cdot\,)$ can be enforced by prescribing the values of the first derivative at each point x_i, $i = 1, 2, \dots, n$. Thus let $m_1, m_2, \dots, m_n$ be arbitrary given numbers, and denote

$$s_3(f;\,\cdot\,)|_{[x_i, x_{i+1}]} = p_i(x), \;\; i = 1, 2, \dots, n-1. \tag{2.138}$$

Then, we enforce $s_3'(f; x_i) = m_i$, $i = 1, 2, \dots, n$, by selecting each piece p_i of $s_3(f;\,\cdot\,)$ to be the (unique) solution of a Hermite interpolation problem, namely,

$$\begin{aligned} p_i(x_i) &= f_i, \quad p_i(x_{i+1}) = f_{i+1}, \\ p_i'(x_i) &= m_i, \quad p_i'(x_{i+1}) = m_{i+1}, \end{aligned} \qquad i = 1, 2, \dots, n-1. \tag{2.139}$$

We solve (2.139) by Newton's interpolation formula. The required divided differences are:

$$\begin{array}{lllll}
x_i & f_i & & & \\
x_i & f_i & m_i & & \\
x_{i+1} & f_{i+1} & [x_i, x_{i+1}]f & \dfrac{[x_i, x_{i+1}]f - m_i}{\Delta x_i} & \\
x_{i+1} & f_{i+1} & m_{i+1} & \dfrac{m_{i+1} - [x_i, x_{i+1}]f}{\Delta x_i} & \dfrac{m_{i+1} + m_i - 2[x_i, x_{i+1}]f}{(\Delta x_i)^2}
\end{array}$$

and the interpolation polynomial (in Newton's form) is

$$\begin{aligned} p_i(x) = f_i + (x - x_i)m_i + (x - x_i)^2 \, \frac{[x_i, x_{i+1}]f - m_i}{\Delta x_i} \\ + \, (x - x_i)^2(x - x_{i+1}) \, \frac{m_{i+1} + m_i - 2[x_i, x_{i+1}]f}{(\Delta x_i)^2}. \end{aligned}$$

Alternatively, in Taylor's form, we can write

$$\begin{aligned} & p_i(x) = c_{i,0} + c_{i,1}(x - x_i) + c_{i,2}(x - x_i)^2 + c_{i,3}(x - x_i)^3, \\ & \quad x_i \le x \le x_{i+1}, \end{aligned} \tag{2.140}$$

where, by noting that $x - x_{i+1} = x - x_i - \Delta x_i$,

$$\begin{aligned} c_{i,0} &= f_i, \\ c_{i,1} &= m_i, \\ c_{i,2} &= \frac{[x_i, x_{i+1}]f - m_i}{\Delta x_i} - c_{i,3} \cdot \Delta x_i, \\ c_{i,3} &= \frac{m_{i+1} + m_i - 2[x_i, x_{i+1}]f}{(\Delta x_i)^2}. \end{aligned} \tag{2.141}$$

Thus to compute $s_3(f; x)$ for any given $x \in [a, b]$ that is not an interpolation node, one first locates the interval $[x_i, x_{i+1}]$ containing x and then computes the corresponding piece (2.138) by (2.140) and (2.141).

We now discuss some possible choices of the parameters $m_1, m_2, \ldots, m_n$.

(a) *Piecewise cubic Hermite interpolation*. Here one selects $m_i = f'(x_i)$, assuming that these derivative values are known. This gives rise to a strictly *local* scheme, in that each piece p_i can be determined independently from the others. Furthermore, the error of interpolation is easily estimated, since from the theory of interpolation,

$$f(x) - p_i(x) = (x - x_i)^2(x - x_{i+1})^2[x_i, x_i, x_{i+1}, x_{i+1}, x]f, \; x_i \le x \le x_{i+1};$$

hence, if $f \in C^4[a,b]$,

$$|f(x) - p_i(x)| \le \left(\frac{1}{2}\Delta x_i\right)^4 \max_{[x_i,x_{i+1}]} \frac{|f^{(4)}|}{4!}, \quad x_i \le x \le x_{i+1}.$$

There follows:

$$\|f(\cdot) - s_3(f;\cdot)\|_\infty \le \tfrac{1}{384}|\Delta|^4\,\|f^{(4)}\|_\infty. \tag{2.142}$$

In the case of equally spaced points x_i, one has $|\Delta| = (b-a)/(n-1)$ and, therefore,

$$\|f(\cdot) - s_3(f;\cdot)\|_\infty = O(n^{-4}) \text{ as } n \to \infty. \tag{2.143}$$

This is quite satisfactory, but note that the derivative of f must be known at each point x_i, and the interpolant is only in $C^1[a,b]$.

As to the derivative values, one could approximate them by the derivatives of $p_2(f; x_{i-1}, x_i, x_{i+1}; x)$ at $x = x_i$, which requires only function values of f, except at the endpoints, where again the derivatives of f are involved, the points $a = x_0 = x_1$ and $b = x_n = x_{n+1}$ being double points (cf. Ex. 78). It can be shown that this degrades the accuracy to $O(|\Delta|^3)$.

(b) *Cubic spline interpolation*. Here we require $s_3(f;\cdot) \in \mathbb{S}_3^2(\Delta)$, that is, continuity of the second derivative. In terms of the pieces (2.138) of $s_3(f;\cdot)$, this means that

$$p''_{i-1}(x_i) = p''_i(x_i), \quad i = 2, 3, \dots, n-1, \tag{2.144}$$

and translates into a condition for the Taylor coefficients in (2.140), namely,

$$2\,c_{i-1,2} + 6\,c_{i-1,3} \cdot \Delta x_{i-1} = 2\,c_{i,2}, \quad i = 2, 3, \dots, n-1.$$

Plugging in the explicit values (2.141) for these coefficients, we arrive at the linear system

$$(\Delta x_i)m_{i-1} + 2(\Delta x_{i-1} + \Delta x_i)m_i + (\Delta x_{i-1})m_{i+1} = b_i, \quad i = 2, 3, \dots, n-1, \tag{2.145}$$

where

$$b_i = 3\{(\Delta x_i)[x_{i-1}, x_i]f + (\Delta x_{i-1})[x_i, x_{i+1}]f\}. \tag{2.146}$$

These are $n-2$ linear equations in the n unknowns $m_1, m_2, \dots, m_n$. Once m_1 and m_n have been chosen in some way, the system again becomes tridiagonal in the remaining unknowns and hence is readily solved by Gauss elimination. Here are some possible choices of m_1 and m_n.

(b.1) *Complete splines*: $m_1 = f'(a)$, $m_n = f'(b)$. It is known that for this spline

$$\|f^{(r)}(\cdot)-s^{(r)}(f;\,\cdot)\|_\infty \le c_r|\Delta|^{4-r}\|f^{(4)}\|_\infty, r = 0, 1, 2, 3, \text{ if } f \in C^4[a,b], \tag{2.147}$$

where $c_0 = \frac{5}{384}$, $c_1 = \frac{1}{24}$, $c_2 = \frac{3}{8}$, and c_3 is a constant depending on the mesh ratio $\frac{|\Delta|}{\min_i \Delta x_i}$. Rather remarkably, the bound for $r = 0$ is only five times larger than the bound (2.142) for the piecewise cubic Hermite interpolant, which requires derivative values of f at all interpolation nodes x_i, not just at the endpoints a and b.

(b.2) *Matching of the second derivatives at the endpoints*: $s_3''(f;a) = f''(a)$, $s_3''(f;b) = f''(b)$. Each of these conditions gives rise to an additional equation, namely,

$$\begin{aligned} 2m_1 + m_2 &= 3[x_1,x_2]f - \frac{1}{2}f''(a)\Delta x_1, \\ m_{n-1} + 2m_n &= 3[x_{n-1},x_n]f + \frac{1}{2}f''(b)\Delta x_{n-1}. \end{aligned} \tag{2.148}$$

One conveniently adjoins the first equation to the top of the system (2.145), and the second to the bottom, thereby preserving the tridiagonal structure of the system.

(b.3) *Natural cubic spline*: $s''(f;a) = s''(f;b) = 0$. This again produces two additional equations, which can be obtained from (2.148) by putting there $f''(a) = f''(b) = 0$. They are adjoined to the system (2.145) as described in (b.2). The nice thing about this spline is that it requires only function values of f – no derivatives! – but the price one pays is a degradation of the accuracy to $O(|\Delta|^2)$ near the endpoints (unless indeed $f''(a) = f''(b) = 0$).

(b.4) *"Not-a-knot spline"* (C. de Boor): here we require $p_1(x) \equiv p_2(x)$ and $p_{n-2}(x) \equiv p_{n-1}(x)$; that is, the first two pieces of the spline should be the same polynomial, and similarly for the last two pieces. In effect, this means that the first interior knot x_2, and the last one x_{n-1}, both are inactive (hence the name). This again gives rise to two supplementary equations expressing continuity of $s_3'''(f;x)$ at $x = x_2$ and $x = x_{n-1}$ (cf. Ex. 79).

2.3.5 Minimality Properties of Cubic Spline Interpolants

The complete and natural splines defined in (b.1) and (b.3) of the preceding section have interesting optimality properties. To formulate them, it is convenient to consider not only the subdivision Δ in (2.122), but also the subdivision

$$\Delta' : \quad a = x_0 = x_1 < x_2 < x_3 < \cdots < x_{n-1} < x_n = x_{n+1} = b, \tag{2.149}$$

in which the endpoints are double knots. This means that whenever we interpolate on Δ', we interpolate not only to function values at all interior points but also to the function as well as first derivative values at the endpoints.

The first of the two theorems relates to the complete cubic spline interpolant, $s_{\text{compl}}(f\,;\,\cdot\,)$.

Theorem 2.3.1. *For any function $g \in C^2[a,b]$ that interpolates f on Δ', there holds*

$$\int_a^b [g''(x)]^2 \mathrm{d}x \geq \int_a^b [s''_{\text{compl}}(f\,;x)]^2 \mathrm{d}x, \tag{2.150}$$

with equality if and only if $g(\,\cdot\,) = s_{\text{compl}}(f\,;\,\cdot\,)$.

Note that $s_{\text{compl}}(f\,;\,\cdot\,)$ in Theorem 2.3.1 also interpolates f on Δ', and among all such interpolants its second derivative has the smallest L_2 norm.

Proof of Theorem 2.3.1. We write (for short) $s_{\text{compl}} = s$. The theorem follows, once we have shown that

$$\int_a^b [g''(x)]^2 \mathrm{d}x = \int_a^b [g''(x) - s''(x)]^2 \mathrm{d}x + \int_a^b [s''(x)]^2 \mathrm{d}x. \tag{2.151}$$

Indeed, this immediately implies (2.150), and equality in (2.150) holds if and only if $g''(x) - s''(x) \equiv 0$, which, integrating twice from a to x and using the interpolation properties of s and g at $x = a$ gives $g(x) \equiv s(x)$.

To complete the proof, note that (2.151) is equivalent to

$$\int_a^b s''(x)[g''(x) - s''(x)]\mathrm{d}x = 0. \tag{2.152}$$

Integrating by parts, we get

$$\begin{aligned}
&\int_a^b s''(x)[g''(x) - s''(x)]\mathrm{d}x \\
&\quad = s''(x)[g'(x) - s'(x)]\Big|_a^b - \int_a^b s'''(x)[g'(x) - s'(x)]\mathrm{d}x \\
&\quad = -\int_a^b s'''(x)[g'(x) - s'(x)]\mathrm{d}x,
\end{aligned} \tag{2.153}$$

since $s'(b) = g'(b) = f'(b)$, and similarly at $x = a$. But s''' is piecewise constant, so

$$\begin{aligned}
&\int_a^b s'''(x)[g'(x) - s'(x)]\mathrm{d}x \\
&\quad = \sum_{\nu=1}^{n-1} s'''(x_\nu + 0) \int_{x_\nu}^{x_{\nu+1}} [g'(x) - s'(x)]\mathrm{d}x \\
&\quad = \sum_{\nu=1}^{n-1} s'''(x_\nu + 0)[g(x_{\nu+1}) - s(x_{\nu+1}) - (g(x_\nu) - s(x_\nu))] = 0,
\end{aligned}$$

since both s and g interpolate to f on Δ. This proves (2.152) and hence the theorem. □

For interpolation on Δ, the distinction of being optimal goes to the natural cubic spline interpolant $s_{\text{nat}}(f;\,\cdot\,)$. This is the content of the second theorem.

Theorem 2.3.2. *For any function $g \in C^2[a,b]$ that interpolates f on Δ (not Δ'), there holds*

$$\int_a^b [g''(x)]^2 \mathrm{d}x \geq \int_a^b [s''_{nat}(f;x)]^2 \mathrm{d}x, \tag{2.154}$$

with equality if and only if $g(\,\cdot\,) = s_{nat}(f;\,\cdot\,)$.

The proof of Theorem 2.3.2 is virtually the same as that of Theorem 2.3.1, since (2.153) holds again, this time because $s''(b) = s''(a) = 0$. □

Putting $g(\,\cdot\,) = s_{\text{compl}}(f;\,\cdot\,)$ in Theorem 2.3.2 immediately gives

$$\int_a^b [s''_{\text{compl}}(f;x)]^2 \mathrm{d}x \geq \int_a^b [s''_{\text{nat}}(f;x)]^2 \mathrm{d}x. \tag{2.155}$$

Therefore, in a sense, the natural cubic spline is the "smoothest" interpolant.

The property expressed in Theorem 2.3.2 is the origin of the name "spline." A spline is a flexible strip of wood used in drawing curves. If its shape is given by the equation $y = g(x)$, $a \leq x \leq b$, and if the spline is constrained to pass through the points (x_i, g_i), then it assumes a form that minimizes the bending energy

$$\int_a^b \frac{[g''(x)]^2 \mathrm{d}x}{(1 + [g'(x)]^2)^3}$$

over all functions g similarly constrained. For slowly varying g ($\|g'\|_\infty \ll 1$), this is nearly the same as the minimum property of Theorem 2.3.2.

2.4 Notes to Chapter 2

There are many excellent texts on the general problem of best approximation as exemplified by (2.1). One that emphasizes uniform approximation by polynomials is Feinerman and Newman [1974]; apart from the basic theory of best polynomial approximation, it also contains no fewer than four proofs of the fundamental theorem of Weierstrass. For approximation in the L_∞ and L_1 norm, which is related to linear programming, a number of constructive methods, notably the Remez algorithms and exchange algorithms, are known, both for polynomial and rational approximation. Early, but still very readable, expositions are given in Cheney [1998] and Rivlin [1981], and more recent accounts in Watson [1980] and Powell [1981]. Nearly-best polynomial and rational approximations are widely used in computer routines for special functions; for a survey of work in this area, up to about 1975, see Gautschi [1975a], and for subsequent work, van der Laan and Temme [1984] and Németh [1992]. Much relevant material is also contained in the books by Luke [1975] and [1977]. The numerical approximation and software for special functions is the subject of Gil et al. [2007]; exhaustive documentation can also be found in Lozier and Olver [1994]. A package for some of the more esoteric functions is described in MacLeod [1996]. For an extensive (and mathematically demanding) treatment of rational approximation, the reader is referred to Petrushev and Popov [1987], and for L_1 approximation, to Pinkus [1989]. Methods of nonlinear approximation, including approximation by exponential sums, are studied in Braess [1986]. Other basic texts on approximation and interpolation are Natanson [1964, 1965, 1965] and Davis [1975] from the 1960s, and the more recent books by DeVore and Lorentz [1993] and its sequel, Lorentz et al. [1996]. A large variety of problems of interpolation and approximation by rational functions (including polynomials) in the complex plane is studied in Walsh [1969]. An example of a linear space Φ containing a denumerable set of nonrational basis functions are the sinc functions – scaled translates of $\frac{\sin \pi t}{\pi t}$. They are of importance in the Shannon sampling and interpolation theory (see, e.g., Zayed [1993]) and are also useful for approximation on infinite or semi-infinite domains in the complex plane; see Stenger [1993], [2000] and Kowalski et al. [1995] for an extensive discussion of this. A reader interested in issues of current interest related to multivariate approximation can get a good start by consulting Cheney [1986].

Rich and valuable sources on polynomials and their numerous properties of interest in applied analysis are Milovanović et al. [1994] and Borwein and Erdélyi [1995]. Spline functions – in name and as a basic tool of approximation – were introduced in 1946 by Schoenberg [1946]; also see Schoenberg [1973]. They have generated enormous interest, owing both to their interesting mathematical theory and practical usefulness. There are now many texts available, treating splines from various points of view. A selected list is Ahlberg et al. [1967], Nürnberger [1989], and Schumaker [2007] for the basic theory, de Boor [2001] and Späth [1995] for more practical aspects including algorithms, Atteia [1992] for an abstract treatment based on Hilbert kernels, Bartels et al. [1987] and Dierckx [1993] for applications to computer graphics and geometric modeling, and Chui [1988], de Boor et al. [1993], and Bojanov et al. [1993] for multivariate splines. The standard text on trigonometric series still is Zygmund [2002] .

Section 2.1. Historically, the least squares principle evolved in the context of discrete linear approximation. The principle was first enunciated by Legendre in 1805 in a treatise on celestial mechanics (Legendre [1805]), although Gauss used it earlier in 1794, but published the method only in 1809 (in a paper also on celestial mechanics). For Gauss's subsequent treatises, published in 1821–1826, see the English translation in Gauss [1995]. The statistical justification of least squares as a minimum variance (unbiased) estimator is due to Gauss. If one were to disregard probabilistic arguments, then, as Gauss already remarked (Goldstine [1977, p. 212]), one could try to minimize the sum of any even (positive) power of the errors, and even let this power go to infinity, in which case one would minimize the maximum error. But by these principles " ... we should be led into the most complicated calculations." Interestingly, Laplace at about the same time also proposed discrete L_1 approximation (under the side condition that all errors add up to zero). A reader interested in the history of least squares may wish to consult the article by Sheynin [1993].

The choice of weights w_i in the discrete L_2 norm $\|\cdot\|_{2,w}$ can be motivated on statistical grounds if one assumes that the errors in the data $f(x_i)$ are uncorrelated and have zero mean and variances σ_i^2; an appropriate choice then is $w_i = \sigma_i^{-2}$.

The discrete problem of minimizing $\|f - \varphi\|_{2,w}$ over functions φ in Φ as given by (2.2) can be rephrased in terms of an overdetermined system of linear equations, $\boldsymbol{Pc} = \boldsymbol{f}$, where $\boldsymbol{P} = [\pi_j(x_i)]$ is a rectangular matrix of size $N \times n$, and $\boldsymbol{f} = [f(x_i)]$ the data vector of dimension N. If $\boldsymbol{r} = \boldsymbol{f} - \boldsymbol{Pc}$, $\boldsymbol{r} = [r_i]$ denotes the residual vector, one tries to find the coefficient vector $\boldsymbol{c} \in \mathbb{R}^n$ such that $\sum_i w_i r_i^2$ is as small as possible. There is a vast literature dealing with overdetermined systems involving more general (full or sparse) matrices and their solution by the method of least squares. A large arsenal of modern techniques of matrix computation can be brought to bear on this problem; see, for example, Björck [1996] for an extensive discussion. In the special case considered here, the method of (discrete) orthogonal polynomials, however, is more efficient. It has its origin in the work of Chebyshev [1859]; a more contemporary exposition, including computational and statistical issues, is given in Forsythe [1957].

There are interesting variations on the theme of polynomial least squares approximation. One is to minimize $\|f - p\|_{2,\mathrm{d}\lambda}$ among all polynomials in $\mathbb{P}_n$ subject to interpolatory constraints at $m+1$ given points, where $m < n$. It turns out that this can be reduced to an unconstrained least squares problem, but for a different measure $\mathrm{d}\lambda$ and a different function f; cf. Gautschi [1996, Sect. 2.1]. Something similar is true for approximation by rational functions with a prescribed denominator polynomial. A more substantial variation consists in wanting to approximate simultaneously a function f and its first s derivatives. In the most general setting, this would require the minimization of $\int_{\mathbb{R}} \sum_{\sigma=0}^{s} [f^{(\sigma)}(t) - p^{(\sigma)}(t)]^2 \mathrm{d}\lambda_\sigma(t)$ among all polynomials $p \in \mathbb{P}_n$, where $\mathrm{d}\lambda_\sigma$ are given (continuous or discrete) positive measures. The problem can be solved, as in Sect. 2.1.2, by orthogonal polynomials, but they are now orthogonal with respect to the inner product $(u,v)_{H_s} = \sum_{\sigma=0}^{s} \int_{\mathbb{R}} u^{(\sigma)}(t) v^{(\sigma)}(t) \mathrm{d}\lambda_\sigma(t)$ – a so-called Sobolev inner product. This gives rise to Sobolev orthogonal polynomials; see Gautschi [2004, Sect. 1.7] for some history on this problem and relevant literature.

Section 2.1.2. The alternative form (2.25) of computing the coefficients $\hat{c}_j$ was suggested in the 1972 edition of Conte and de Boor [1980] and is further discussed by Shampine [1975]. The Gram–Schmidt procedure described at the end of this section is now called the classical Gram–Schmidt procedure. There are other, modified, versions of Gram–Schmidt that are computationally more effective; see, for example, Björck [1996, pp. 61ff].

Section 2.1.4. The standard text on Fourier series, as already mentioned, is Zygmund [2002], and on orthogonal polynomials, Szegö [1975]. Not only is it true that orthogonal polynomials satisfy a three-term recurrence relation (2.38), but the converse is also true: any system $\{\pi_k\}$ of monic polynomials satisfying (2.38) for all $k \geq 0$, with real coefficients α_k and $\beta_k > 0$, is necessarily orthogonal with respect to some (in general unknown) positive measure. This is known as Favard's Theorem (cf., e.g., Natanson [1965], Vol. 2, Chap. 8, Sect. 6]). The computation of orthogonal polynomials, when the recursion coefficients are not known explicitly, is not an easy task; a number of methods are surveyed in Gautschi [1996]; see also Gautschi [2004, Chap. 2]. Orthogonal systems in $L_2(\mathbb{R})$ that have become prominent in recent years are wavelets, which are functions of the form $\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k)$, $j,k = 0, \pm 1, \pm 2, \ldots$, with ψ a "mother wavelet" – square integrable on $\mathbb{R}$ and (usually) satisfying $\int_{\mathbb{R}} \psi(t)\mathrm{d}t = 0$. Among the growing textbook and monograph literature on this subject, we mention Chui [1992], Daubechies [1992], Walter [1994], Wickerhauser [1994], Hernández and Weiss [1996], Resnikoff and Wells [1998], Burrus et al. [1998], and Novikov et al. [2010].

Section 2.2. Although interpolation by polynomials and spline functions is most common, it is sometimes appropriate to use other systems of approximants for interpolation, for example, trigonometric polynomials or rational functions. Trigonometric interpolation at equally spaced points is closely related to discrete Fourier analysis and hence accessible to the Fast Fourier Transform (FFT). For this, and also for rational interpolation algorithms, see, for example, Stoer and Bulirsch [2002, Sects. 2.1.1 and 2.2]. For the fast Fourier transform and some of its important applications, see Henrici [1979a] and Van Loan [1992].

Besides Lagrange and Hermite interpolation, other types of interpolation processes have been studied in the literature. Among these are Fejér–Hermite interpolation, where one interpolates to given function values and requires the derivative to vanish at these points, and Birkhoff (also called lacunary) interpolation, which is similar to Hermite interpolation, but derivatives of only preselected orders are being interpolated. Remarkably, Fejér–Hermite interpolation at the Chebyshev points (defined in Sect. 2.2.4) converges for every continuous function $f \in C[-1,1]$. The convergence theory of Lagrange and Fejér–Hermite interpolation is the subject of a monograph by Szabados and Vértesi [1990]. The most comprehensive work on Birkhoff interpolation is the book by G. G. Lorentz et al. [1983]. A more recent monograph by R. A. Lorentz [1992] deals with multivariate Birkhoff interpolation.

Section 2.2.1. The growth of the Lebesgue constants Λ_n is at least $O(\log n)$ as $n \to \infty$; specifically, $\Lambda_n > \frac{2}{\pi}\log n + c$ for any triangular array of interpolation nodes (cf. Sect. 2.1.4), where the constant c can be expressed in terms of Euler's constant γ (cf. Chap. 1, MA 4) by $c = \frac{2}{\pi}\left(\log\frac{8}{\pi} + \gamma\right) = 0.9625228\ldots$; see Rivlin [1990, Theorem 1.2]. The Chebyshev points achieve the optimal order $O(\log n)$; for them, $\Lambda_n \leq \frac{2}{\pi}\log n + 1$ (Rivlin [1990, Theorem 1.2]). Equally spaced nodes, on the other hand, lead to exponential growth of the Lebesgue constants inasmuch as $\Lambda_n \sim 2^{n+1}/(\mathrm{e}n\log n)$ for $n \to \infty$; see Trefethen and Weideman [1991] for some history on this result and Brutman [1997a] for a recent survey on Lebesgue constants. The very last statement of Sect. 2.1.2 is the content of Faber's Theorem (see, e.g., Natanson [1965, Vol. 3, Chap. 2, Theorem 2]), which says that, no matter how one chooses the triangular array of nodes (2.64) in $[a, b]$, there is always a continuous function $f \in C[a, b]$ for which the Lagrange interpolation process does not converge uniformly to f. Indeed, there is an $f \in C[a, b]$ for which Lagrange interpolation diverges almost everywhere in $[a, b]$; see Erdös and Vértesi [1980]. Compare this with Fejér–Hermite interpolation.

Section 2.2.3. A more complete discussion of how the convergence domain of Lagrange interpolation in the complex plane depends on the limit distribution of the interpolation nodes can be found in Krylov [1962, Chap. 12, Sect. 2].

Runge's example is further elucidated in Epperson [1987]. For an analysis of Bernstein's example, we refer to Natanson [1965, Vol. 3, Chap. 2, Sect. 2]. The same divergence phenomenon, incidentally, is exhibited also for a large class of nonequally spaced nodes; see Brutman and Passow [1995]. The proof of Example 5 follows Fejér [1918].

Section 2.2.4. The Chebyshev polynomial arguably is one of the most interesting polynomials from the point of view not only of approximation theory, but also of algebra and number theory. In Rivlin's words, it "... is like a fine jewel that reveals different characteristics under illumination from various positions." In his text, Rivlin [1990] gives ample testimony in support of this view. Another text, unfortunately available only in Russian (or Polish), is Paszkowski [1983], which has an exhaustive account of analytic properties of Chebyshev polynomials as well as numerical applications.

The convergence result stated in (2.97) follows from (2.59) and the logarithmic growth of Λ_n, since $\mathcal{E}_n(f) \log n \to 0$ for $f \in C^1[-1, 1]$ by Jackson's theorems (cf. Cheney [1998, p. 147]). A more rigorous estimate for the error in (2.102) is $\mathcal{E}_n(f) \leq \|\tau_n - f\|_\infty \leq \left(4 + \frac{4}{\pi^2} \log n\right) \mathcal{E}_n(f)$ (Rivlin [1990, Theorem 3.3]), where the infinity norm refers to the interval $[-1, 1]$ and $\mathcal{E}_n(f)$ is the best uniform approximation of f on $[-1, 1]$ by polynomials of degree n.

Section 2.2.5. A precursor of the algorithm (2.108) expressing $\lambda_k^{(k)}$ in the form of a sum rather than a product, and thus susceptible to serious cancellation errors, was proposed in Werner [1984]. The more stable algorithm given in the text is due to Berrut and Trefethen [2004]. Barycentric formulae have been developed also for trigonometric interpolation (see Henrici [1979b] for uniform, and Salzer [1949] and Berrut [1984] for nonuniform distributions of the nodes), and for cardinal (sinc-) interpolation (Berrut [1989]); for the latter, see also Gautschi [2001] and Chap. 1, MA 10.

Section 2.2.7. There are explicit formulae, analogous to Lagrange's formula, for Hermite interpolation in the most general case; see, for example, Stoer and Bulirsch [2002, Sect. 2.1.5]. For the important special case $m_k = 2$, see also Chap. 3, Ex. 34(a).

Section 2.2.8. To estimate the error of inverse interpolation, using an appropriate version of (2.60), one needs the derivatives of the inverse function f^{-1}. A general expression for the nth derivative of f^{-1} in terms of the first n derivatives of f is derived in Ostrowski [1973, Appendix C].

Section 2.3. The definition of the class of spline functions $\mathbb{S}_m^k(\Delta)$ can be refined to $\mathbb{S}_m^{\boldsymbol{k}}(\Delta)$, where $\boldsymbol{k}^{\mathrm{T}} = [k_2, k_3, \dots, k_{n-1}]$ is a vector with integer components $k_i \geq -1$ specifying the degree of smoothness at the interior knots x_i; that is, $s^{(j)}(x_i + 0) - s^{(j)}(x_i - 0) = 0$ for $j = 0, 1, \dots, k_i$. Then $\mathbb{S}_m^k(\Delta)$ as defined in (2.124) becomes $\mathbb{S}_m^{\boldsymbol{k}}(\Delta)$ with $\boldsymbol{k} = [k, k, \dots, k]$.

Section 2.3.1. As simple as the procedure of piecewise linear interpolation may seem, it can be applied to advantage in numerical Fourier analysis, for example. In trying to compute the (complex) Fourier coefficients $c_n(f) = \frac{1}{2\pi}\int_0^{2\pi} f(x)\mathrm{e}^{-\mathrm{i}nx}\mathrm{d}x$ of a 2π-periodic function f, one often approximates them by the "discrete Fourier transform" $\hat{c}_n(f) = \frac{1}{N}\sum_{k=0}^{N-1} f(x_k)\mathrm{e}^{-\mathrm{i}nx_k}$, where $x_k = k\frac{2\pi}{N}$. This can be computed efficiently (for large N) by the Fast Fourier Transform. Note, however, that $\hat{c}_n(f)$ is periodic in n with period N, whereas the true Fourier coefficients $c_n(f)$ tend to zero as $n \to \infty$. To remove this deficiency, one can approximate f by some (simple) function φ and thereby approximate $c_n(f)$ by $c_n(\varphi)$. Then $c_n(\varphi)$ will indeed tend to zero as $n \to \infty$. The simplest choice for φ is precisely the piecewise linear interpolant $\varphi = s_1(f;\cdot)$ (relative to the uniform partition of $[0, 2\pi]$ into N subintervals). One then finds, rather remarkably (see Chap. 3, Ex. 14), that $c_n(\varphi)$ is a multiple of the discrete Fourier transform, namely, $c_n(f) = \tau_n\hat{c}_n(f)$, where $\tau_n = \left(\frac{\sin(n\pi/N)}{n\pi/N}\right)^2$; this still allows the application of the FFT but corrects the behavior of $\hat{c}_n(f)$ at infinity. The same modification of the discrete Fourier transform by an "attenuation factor" τ_n occurs for many other approximation processes $f \approx \varphi$; see Gautschi [1971/1972] for a general theory (and history) of attenuation factors.

The near optimality of the piecewise linear interpolant $s_1(f;\cdot)$, as expressed by the inequalities in (2.128), is noted by de Boor [2001, p. 31].

Section 2.3.2. The basis (2.129) for $\mathbb{S}_1^0(\Delta)$ is a special case of a B-spline basis that can be defined for any space of spline functions $\mathbb{S}_m^k(\Delta)$ previously introduced (cf. de Boor [2001, Theorem IX(44)]. The B-splines are formed by means of divided differences of order $m+1$ applied to the truncated power $(t-x)_+^m$ (considered as a function of t). Like the basis in (2.129), each basis function of a B-spline basis is supported on at most $m+1$ consecutive intervals of Δ and is positive on the interior of the support.

Section 2.3.3. A proof of the near optimality of the piecewise linear least squares approximant $\hat{s}_1(f;\cdot)$, as expressed by the inequalities (2.137), can be found in de Boor [2001, p. 32]. For smoothing and least squares approximation procedures involving cubic splines, see, for example, de Boor [2001, Chap. XIV].

Section 2.3.4. (a) For the remark in the last paragraph of (a), see de Boor [2001, Chap. 4, Problem 3].

(b.1) The error bounds in (2.147), which for $r = 0$ and $r = 1$ are asymptotically sharp, are due to Hall and Meyer [1976].

(b.2) The cubic spline interpolant matching second derivatives at the endpoints satisfies the same error bounds as in (2.147) for $r = 0, 1, 2$, with constants $c_0 = \frac{3}{64}$,

$c_1 = \frac{3}{16}$, and $c_2 = \frac{3}{8}$; see Kershaw [1971, Theorem 2]. The same is shown also for periodic spline interpolants s, satisfying $s^{(r)}(a) = s^{(r)}(b)$ for $r = 0, 1, 2$.

(b.3) Even though the natural spline interpolant, in general, converges only with order $|\Delta|^2$ (e.g., for uniform partitions Δ), it has been shown by Atkinson [1968] that the order of convergence is $|\Delta|^4$ on any compact interval contained in the open interval (a, b), and by Kershaw [1971] even on intervals extending (in a sense made precise) to $[a, b]$ as $|\Delta| \to 0$. On such intervals, in fact, the natural spline interpolant s provides approximations to any $f \in C^4[a, b]$ with errors satisfying $\|f^{(r)} - s^{(r)}\|_\infty \leq 8c_r K|\Delta|^{4-r}$, where $K = 2 + \frac{3}{8}\|f^{(4)}\|_\infty$ and $c_0 = \frac{1}{8}$, $c_1 = \frac{1}{2}$, and $c_2 = 1$.

(b.4) The error of the "not-a-knot" spline interpolant is of the same order as the error of the complete spline; it follows from Beatson [1986, (2.49)] that for functions $f \in C^4[a, b]$, one has $\|f^{(r)} - s^{(r)}\|_\infty \leq c_r|\Delta|^{4-r}\|f^{(4)}\|_\infty$, $r = 0$, 1, 2 (at least when $n \geq 6$), where c_r are constants independent of f and Δ. The same bounds are valid for other schemes that depend only on function values, for example, the scheme with m_1 equal to the first (or second) derivative of $p_3(f; x_1, x_2, x_3, x_4; \cdot)$ at $x = a$, and similarly for m_n. The first of these schemes (using first-order derivatives of p_3) is in fact the one recommended by Beatson and Chacko [1989, 1992] for general-purpose interpolation. Numerical experiments in Beatson and Chacko [1989] suggest values of approximately 1 for the constants c_r in the preceding error estimates. In Beatson and Chacko [1992] further comparisons are made among many other cubic spline interpolation schemes.

Section 2.3.5. The minimum norm property of natural splines (Theorem 2.3.1) and its proof based on the identity (2.151), called "the first integral relation" in Ahlberg et al. [1967], is due to Holladay [1957], who derived it in the context of numerical quadrature. "Much of the present-day theory of splines began with this theorem" (Ahlberg et al. [1967, p. 3]). An elegant alternative proof of (2.152), and hence of the theorem, can be based (cf. de Boor [2001, pp. 64–66]) on the Peano representation (see Chap. 3, Sect. 3.2.6) of the second divided difference of $g - s$, that is, $[x_{i-1}, x_i, x_{i+1}](g - s) = \int_{\mathbb{R}} K(t)(g''(t) - s''(t))\mathrm{d}t$, by noting that the Peano kernel K, up to a constant, is the B-spline B_i defined in (2.129). Since the left-hand side is zero by the interpolation properties of g and s, it follows from the preceding equation that $g'' - s''$ is orthogonal to the span of the B_i, hence to s'', which lies in this span.

Exercises and Machine Assignments to Chapter 2

Exercises

1. Suppose you want to approximate the function

$$f(t) = \begin{cases} -1 & \text{if} \quad -1 \leq t < 0, \\ 0 & \text{if} \quad t = 0, \\ 1 & \text{if} \quad 0 < t \leq 1 \end{cases}$$

by a constant function $\varphi(x) = c$:

(a) on [–1,1] in the continuous L_1 norm,
(b) on $\{t_1, t_2, \dots, t_N\}$ in the discrete L_1 norm,
(c) on [–1,1] in the continuous L_2 norm,
(d) on $\{t_1, t_2, \dots, t_N\}$ in the discrete L_2 norm,
(e) on [–1,1] in the ∞-norm,
(f) on $\{t_1, t_2, \dots, t_N\}$ in the discrete ∞-norm.

The weighting in all norms is uniform (i.e., $w(t) \equiv 1$, $w_i = 1$) and $t_i = -1 + \frac{2(i-1)}{N-1}$, $i = 1, 2, \dots, N$. Determine the best constant c (or constants c, if there is nonuniqueness) and the minimum error.

2. Consider the data

$$f(t_i) = 1, \;\; i = 1, 2, \dots, N-1; \quad f(t_N) = y \gg 1.$$

(a) Determine the discrete L_∞ approximant to f by means of a constant c (polynomial of degree zero).
(b) Do the same for discrete (equally weighted) least square approximation.
(c) Compare and discuss the results, especially as $N \to \infty$.

3. Let $x_0, x_1, \dots, x_n$ be pairwise distinct points in $[a, b]$, $-\infty < a < b < \infty$, and $f \in C^1[a, b]$. Show that, given any $\varepsilon > 0$, there exists a polynomial p such that $\|f - p\|_\infty < \varepsilon$ and, at the same time, $p(x_i) = f(x_i)$, $i = 0, 1, \dots, n$. Here $\|u\|_\infty = \max_{a \le x \le b} |u(x)|$. {*Hint*: write $p = p_n(f; \cdot) + \omega_n q$, where $p_n(f; \cdot)$ is the interpolation polynomial of degree n (cf. Sect. 2.2.1, (2.51)), $\omega_n(x) = \prod_{i=0}^{n}(x - x_i)$, $q \in \mathbb{P}$, and apply Weierstrass's approximation theorem.}

4. Consider the function $f(t) = t^\alpha$ on $0 \le t \le 1$, where $\alpha > 0$. Suppose we want to approximate f best in the L_p norm by a constant c, $0 < c < 1$, that is, minimize the L_p error

$$E_p(c) = \|t^\alpha - c\|_p = \left(\int_0^1 |t^\alpha - c|^p \mathrm{d}t \right)^{1/p}$$

as a function of c. Find the optimal $c = c_p$ for $p = \infty$, $p = 2$, and $p = 1$, and determine $E_p(c_p)$ for each of these p-values.

5. Taylor expansion yields the simple approximation $\mathrm{e}^x \approx 1 + x$, $0 \le x \le 1$. Suppose you want to improve this by seeking an approximation of the form $\mathrm{e}^x \approx 1 + cx$, $0 \le x \le 1$, for some suitable c.

(a) How must c be chosen if the approximation is to be optimal in the (continuous, equally weighted) least squares sense?
(b) Sketch the error curves $e_1(x) := \mathrm{e}^x - (1 + x)$ and $e_2(x) := \mathrm{e}^x - (1 + cx)$ with c as obtained in (a) and determine $\max_{0 \le x \le 1} |e_1(x)|$ and $\max_{0 \le x \le 1} |e_2(x)|$.
(c) Solve the analogous problem with three instead of two terms in the modified Taylor expansion: $\mathrm{e}^x \approx 1 + c_1 x + c_2 x^2$, and provide error curves for $e_1(x) = \mathrm{e}^x - 1 - x - \frac{1}{2}x^2$ and $e_2(x) = \mathrm{e}^x - 1 - c_1 x - c_2 x^2$.

6. Prove Schwarz's inequality

$$|(u,v)| \le \|u\| \cdot \|v\|$$

for the inner product (2.10). {*Hint*: use the nonnegativity of $\|u + tv\|^2$, $t \in \mathbb{R}$.}

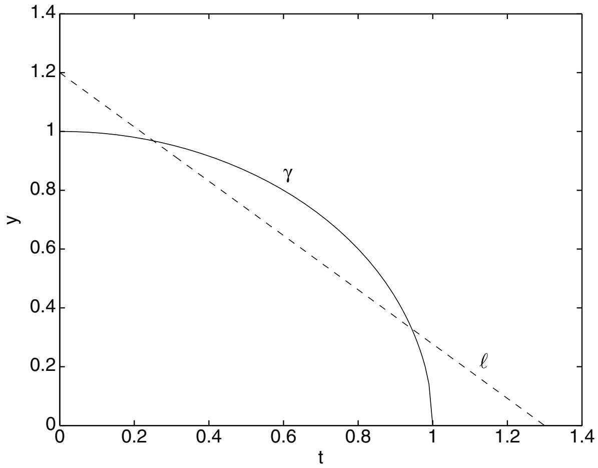
7. Discuss uniqueness and nonuniqueness of the least squares approximant to a function f in the case of a discrete set $T = \{t_1, t_2\}$ (i.e., $N = 2$) and $\Phi_n = \mathbb{P}_{n-1}$ (polynomials of degree $\le n-1$). In case of nonuniqueness, determine *all* solutions.

8. Determine the least squares approximation

$$\varphi(t) = \frac{c_1}{1+t} + \frac{c_2}{(1+t)^2}, \quad 0 \le t \le 1,$$

to the exponential function $f(t) = \mathrm{e}^{-t}$, assuming $\mathrm{d}\lambda(t) = \mathrm{d}t$ on [0,1]. Determine the condition number $\mathrm{cond}_\infty \boldsymbol{A} = \|\boldsymbol{A}\|_\infty \|\boldsymbol{A}^{-1}\|_\infty$ of the coefficient matrix $\boldsymbol{A}$ of the normal equations. Calculate the error $f(t) - \varphi(t)$ at $t = 0$, $t = 1/2$, and $t = 1$. {*Point of information*: the integral $\int_1^\infty t^{-m}\mathrm{e}^{-xt}\mathrm{d}t = E_m(x)$ is known as the "mth exponential integral"; cf. Abramowitz and Stegun [1964, (5.1.4)] or Olver et al. [2010, (8.19.3)].}

9. Approximate the circular quarter arc γ given by the equation $y(t) = \sqrt{1-t^2}$, $0 \le t \le 1$ (see figure) by a straight line ℓ in the least squares sense, using either the weight function $w(t) = (1-t^2)^{-1/2}$, $0 \le t \le 1$, or $w(t) = 1$, $0 \le t \le 1$. Where does ℓ intersect the coordinate axes in these two cases? {*Points of information*: $\int_0^{\pi/2} \cos^2\theta \mathrm{d}\theta = \frac{\pi}{4}$, $\int_0^{\pi/2} \cos^3\theta \mathrm{d}\theta = \frac{2}{3}$.}



10. (a) Let the class Φ_n of approximating functions have the following properties. Each $\varphi \in \Phi_n$ is defined on an interval $[a,b]$ symmetric with respect to the

origin (i.e., $a = -b$), and $\varphi(t) \in \Phi_n$ implies $\varphi(-t) \in \Phi_n$. Let $\mathrm{d}\lambda(t) = \omega(t)\mathrm{d}t$, with $\omega(t)$ an even function on $[a, b]$ (i.e., $\omega(-t) = \omega(t)$). Show: if f is an even function on $[a, b]$, then so is its least squares approximant, $\hat{\varphi}_n$, on $[a, b]$ from Φ_n.

(b) Consider the "hat function" $f(t) = \begin{cases} 1 - t & \text{if } 0 \le t \le 1, \\ 1 + t & \text{if } -1 \le t \le 0. \end{cases}$

Determine its least squares approximation on [–1,1] by a polynomial of degree ≤ 2. (Use $\mathrm{d}\lambda(t) = \mathrm{d}t$.) Simplify your calculation by using part (a). Determine where the error vanishes.

11. Suppose you want to approximate the step function

$$f(t) = \begin{cases} 1 & \text{if } 0 \le t \le 1, \\ 0 & \text{if } t > 1 \end{cases}$$

on the positive line $\mathbb{R}_+$ by a linear combination of exponentials $\pi_j(t) = \mathrm{e}^{-jt}$, $j = 1, 2, \ldots, n$, in the (continuous, equally weighted) least squares sense.

(a) Derive the normal equations. How is the matrix related to the Hilbert matrix?

(b) Use Matlab to solve the normal equations for $n = 1, 2, \ldots, 8$. Print n, the Euclidean condition number of the matrix (supplied by the Matlab function `cond.m`), along with the solution. Plot the approximations vs. the exact function for $1 \le n \le 4$.

12. Let $\pi_j(t) = (t - a_j)^{-1}$, $j = 1, 2, \ldots, n$, where a_j are distinct real numbers with $|a_j| > 1$, $j = 1, 2, \ldots, n$. For $\mathrm{d}\lambda(t) = \mathrm{d}t$ on $-1 \le t \le 1$ and $\mathrm{d}\lambda(t) = 0$, $t \notin [-1, 1]$, determine the matrix of the normal equations for the least squares problem $\int_{\mathbb{R}} (f - \varphi)^2 \mathrm{d}\lambda(t) = \min$, $\varphi = \sum_{j=1}^{n} c_j \pi_j$. Can the sytem $\{\pi_j\}_{j=1}^{n}$, $n > 1$, be an orthogonal system for suitable choices of the constants a_j? Explain.

13. Given an integer $n \ge 1$, consider the subdivision Δ_n of the interval $[0, 1]$ into n equal subintervals of length $1/n$. Let $\pi_j(t)$, $j = 0, 1, \ldots, n$, be the function having the value 1 at $t = j/n$, decreasing on either side linearly to zero at the neighboring subdivision points (if any), and being zero elsewhere.

(a) Draw a picture of these functions. Describe in words the meaning of a linear combination $\pi(t) = \sum_{j=0}^{n} c_j \pi_j(t)$.

(b) Determine $\pi_j(k/n)$ for $j, k = 0, 1, \ldots, n$.

(c) Show that the system $\{\pi_j(t)\}_{j=0}^{n}$ is linearly independent on the interval $0 \le t \le 1$. Is it also linearly independent on the set of subdivision points 0, $\frac{1}{n}, \frac{2}{n}, \ldots, \frac{n-1}{n}$, 1 of Δ_n? Explain.

(d) Compute the matrix of the normal equations for $\{\pi_j\}$, assuming $\mathrm{d}\lambda(t) = \mathrm{d}t$ on [0,1]. That is, compute the $(n + 1) \times (n + 1)$ matrix $\boldsymbol{A} = [a_{ij}]$, where $a_{ij} = \int_0^1 \pi_i(t)\pi_j(t)\mathrm{d}t$.

14. Even though the function $f(t) = \ln(1/t)$ becomes infinite as $t \to 0$, it can be approximated on [0,1] arbitrarily well by polynomials of sufficiently high degree in the (continuous, equally weighted) least squares sense. Show this by proving

$$e_{n,2} := \min_{p\in\mathbb{P}_n} \|f - p\|_2 = \frac{1}{n+1}.$$

{*Hint*: use the following known facts about the "shifted" Legendre polynomial $\pi_j(t)$ of degree j (orthogonal on [0,1] with respect to the weight function $w \equiv 1$ and normalized to satisfy $\pi_j(1) = 1$):

$$\int_0^1 \pi_j^2(t)\mathrm{d}t = \frac{1}{2j+1}, \quad j \geq 0; \quad \int_0^1 \pi_j(t)\ln(1/t)\mathrm{d}t = \begin{cases} 1 & \text{if } j = 0, \\ \frac{(-1)^j}{j(j+1)} & \text{if } j > 0. \end{cases}$$

The first relation is well known from the theory of orthogonal polynomials (see, e.g., Sect. 1.5.1, p. 27 of Gautschi [2004]); the second is due to Blue [1979].}

15. Let $\mathrm{d}\lambda$ be a continuous (positive) measure on $[a,b]$ and $n \geq 1$ a given integer. Assume f continuous on $[a,b]$ and not a polynomial of degree $\leq n-1$. Let $\hat{p}_{n-1} \in \mathbb{P}_{n-1}$ be the least squares approximant to f on $[a,b]$ from polynomials of degree $\leq n-1$:

$$\int_a^b [\hat{p}_{n-1}(t) - f(t)]^2\mathrm{d}\lambda(t) \leq \int_a^b [p(t) - f(t)]^2\mathrm{d}\lambda(t), \quad \text{all } p \in \mathbb{P}_{n-1}.$$

Prove: the error $e_n(t) = \hat{p}_{n-1}(t) - f(t)$ changes sign at least n times in $[a,b]$. {*Hint*: assume the contrary and develop a contradiction.}

16. Let f be a given function on [0,1] satisfying $f(0) = 0,\ f(1) = 1$.

    (a) Reduce the problem of approximating f on [0,1] in the (continuous, equally weighted) least squares sense by a quadratic polynomial p satisfying $p(0) = 0,\ p(1) = 1$ to an unconstrained least squares problem (for a different function).

    (b) Apply the result of (a) to $f(t) = t^r,\ r > 2$. Plot the approximation against the exact function for $r = 3$.

17. Suppose you want to approximate $f(t)$ on $[a,b]$ by a function of the form $r(t) = \pi(t)/q(t)$ in the least squares sense with weight function w, where $\pi \in \mathbb{P}_n$ and q is a *given* function (e.g., a polynomial) such that $q(t) > 0$ on $[a,b]$. Formulate this problem as an ordinary polynomial least squares problem for an appropriate new function $\overline{f}$ and new weight function $\overline{w}$.

18. The Bernstein polynomials of degree n are defined by

$$B_j^n(t) = \binom{n}{j} t^j(1-t)^{n-j}, \quad j = 0, 1, \ldots, n,$$

and are usually employed on the interval $0 \leq t \leq 1$.

(a) Show that $B_0^n(0) = 1$, and for $j = 1, 2, \dots, n$

$$\left.\frac{\mathrm{d}^r}{\mathrm{d}t^r} B_j^n(t)\right|_{t=0} = 0, \;\; r = 0, 1, \dots, j-1; \;\; \left.\frac{\mathrm{d}^j}{\mathrm{d}t^j} B_j^n(t)\right|_{t=0} \neq 0.$$

(b) What are the analogous properties at $t = 1$, and how are they most easily derived?

(c) Prepare a plot of the fourth-degree polynomials $B_j^4(t)$, $j = 0, 1, \dots, 4$, $0 \le t \le 1$.

(d) Use (a) to show that the system $\{B_j^n(t)\}_{j=0}^n$ is linearly independent on [0,1] and spans the space $\mathbb{P}_n$.

(e) Show that $\sum_{j=0}^n B_j^n(t) \equiv 1$. {*Hint*: use the binomial theorem.}

19. Prove that, if $\{\pi_j\}_{j=1}^n$ is linearly dependent on the support of $\mathrm{d}\lambda$, then the matrix $\boldsymbol{A} = [a_{ij}]$, where $a_{ij} = (\pi_i, \pi_j)_{\mathrm{d}\lambda} = \int_{\mathbb{R}} \pi_i(t)\pi_j(t)\mathrm{d}\lambda(t)$, is singular.

20. Given the recursion relation $\pi_{k+1}(t) = (t - \alpha_k)\pi_k(t) - \beta_k \pi_{k-1}(t)$, $k = 0, 1, 2, \dots$, for the (monic) orthogonal polynomials $\{\pi_j(\,\cdot\,; \mathrm{d}\lambda)\}$, and defining $\beta_0 = \int_{\mathbb{R}} \mathrm{d}\lambda(t)$, show that $\|\pi_k\|^2 = \beta_0\beta_1 \cdots \beta_k$, $k = 0, 1, 2, \dots$.

21. (a) Derive the three-term recurrence relation

$$\sqrt{\beta_{k+1}}\tilde{\pi}_{k+1}(t) = (t - \alpha_k)\tilde{\pi}_k(t) - \sqrt{\beta_k}\tilde{\pi}_{k-1}, \quad k = 0, 1, 2, \dots,$$

$$\tilde{\pi}_{-1}(t) = 0, \quad \tilde{\pi}_0 = 1/\sqrt{\beta_0}$$

for the orthonormal polynomials $\tilde{\pi}_k = \pi_k / \|\pi_k\|$, $k = 0, 1, 2, \dots$.

(b) Use the result of (a) to derive the *Christoffel–Darboux* formula

$$\sum_{k=0}^n \tilde{\pi}_k(x)\tilde{\pi}_k(t) = \sqrt{\beta_{n+1}}\, \frac{\tilde{\pi}_{n+1}(x)\tilde{\pi}_n(t) - \tilde{\pi}_n(x)\tilde{\pi}_{n+1}(t)}{x - t}.$$

22. (a) Let $\pi_n(\,\cdot\,) = \pi_n(\,\cdot\,; \mathrm{d}\lambda)$ be the (monic) orthogonal polynomial of degree n relative to the positive measure $\mathrm{d}\lambda$ on $\mathbb{R}$. Show:

$$\int_{\mathbb{R}} \pi_n^2(t; \mathrm{d}\lambda)\mathrm{d}\lambda(t) \le \int_{\mathbb{R}} p^2(t)\mathrm{d}\lambda(t), \quad \text{all } p \in \overset{\circ}{\mathbb{P}}_n,$$

where $\overset{\circ}{\mathbb{P}}_n$ is the class of monic polynomials of degree n. Discuss the case of equality. {*Hint*: represent p in terms of $\pi_j(\,\cdot\,; \mathrm{d}\lambda)$, $j = 0, 1, \dots, n$.}

(b) If $\mathrm{d}\lambda(t) = \mathrm{d}\lambda_N(t)$ is a discrete measure with exactly N support points $t_1, t_2, \dots, t_N$, and $\pi_j(t) = \pi_j(\,\cdot\,; \mathrm{d}\lambda_N)$, $j = 0, 1, \dots, N-1$, are the corresponding (monic) orthogonal polynomials, let $\pi_N(t) = (t - \alpha_{N-1})\pi_{N-1}(t) - \beta_{N-1}\pi_{N-2}(t)$, with α_{N-1}, β_{N-1} defined as in Sect. 2.1.4(2). Show that $\pi_N(t_j) = 0$ for $j = 1, 2, \dots, N$.

23. Let $\{\pi_j\}_{j=0}^n$ be a system of orthogonal polynomials, not necessarily monic, relative to the (positive) measure $\mathrm{d}\lambda$. For some a_{ij}, define

$$p_i(t) = \sum_{j=0}^n a_{ij}\pi_j(t), \quad i = 1, 2, \ldots, n,$$

   (a) Derive conditions on the matrix $\boldsymbol{A} = [a_{ij}]$ which ensure that the system $\{p_i\}_{i=0}^n$ is also a system of orthogonal polynomials.
   (b) Assuming all π_j monic and $\{p_i\}_{i=0}^n$ an orthogonal system, show that each p_i is monic if and only if $\boldsymbol{A} = \boldsymbol{I}$ is the identity matrix.
   (c) Prove the same as in (b), with "monic" replaced by "orthonormal" throughout.

24. Let $(u, v) = \sum_{k=1}^N w_k u(t_k)v(t_k)$ be a discrete inner product on the interval [–1,1] with $-1 \le t_1 < t_2 < \cdots < t_N \le 1$, and let α_k, β_k be the recursion coefficients for the (monic) orthogonal polynomials $\{\pi_k(t)\}_{k=0}^{N-1}$ associated with (u, v):

$$\begin{cases} \pi_{k+1}(t) = (t - \alpha_k)\pi_k(t) - \beta_k\pi_{k-1}(t), \\ \qquad\qquad\qquad\qquad\qquad\qquad k = 0, 1, 2, \ldots, N-2, \\ \pi_0(t) = 1, \quad \pi_{-1}(t) = 0. \end{cases}$$

   Let $x = \frac{b-a}{2}t + \frac{a+b}{2}$ map the interval [–1,1] to $[a, b]$, and the points $t_k \in [-1, 1]$ to $x_k \in [a, b]$. Define $(u, v)^* = \sum_{k=1}^N w_k u(x_k)v(x_k)$, and let $\{\pi_k^*(x)\}_{k=0}^{N-1}$ be the (monic) orthogonal polynomials associated with $(u, v)^*$. Express the recursion coefficients α_k^*, β_k^* for the $\{\pi_k^*\}$ in terms of those for $\{\pi_k\}$. {*Hint*: first show that $\pi_k^*(x) = (\frac{b-a}{2})^k\pi_k(\frac{2}{b-a}(x - \frac{a+b}{2}))$.}

25. Let

$$(\star) \qquad \begin{cases} \pi_{k+1}(t) = (t - \alpha_k)\pi_k(t) - \beta_k\pi_{k-1}(t), \\ \qquad\qquad\qquad\qquad\qquad\qquad k = 0, 1, 2, \ldots, n-1, \\ \pi_0(t) = 1, \quad \pi_{-1}(t) = 0 \end{cases}$$

   and consider

$$p_n(t) = \sum_{j=0}^n c_j\pi_j(t).$$

   Show that p_n can be computed by the following algorithm (*Clenshaw's algorithm*):

$$(\star\star)\qquad \begin{cases} u_n = c_n, \;\; u_{n+1} = 0, \\ u_k = (t - \alpha_k)u_{k+1} - \beta_{k+1}u_{k+2} + c_k, \\ \qquad\qquad\qquad\qquad k = n-1, n-2, \dots, 0, \\ p_n = u_0. \end{cases}$$

{*Hint*: write ($\star$) in matrix form in terms of the vector $\boldsymbol{\pi}^{\mathrm{T}} = [\pi_0, \pi_1, \dots, \pi_n]$ and a unit triangular matrix. Do likewise for ($\star\star$).}

26. Show that the elementary Lagrange interpolation polynomials $\ell_i(x)$ are invariant with respect to any linear transformation of the independent variable.
27. Use Matlab to prepare plots of the Lebesgue function for interpolation, $\lambda_n(x)$, $-1 \le x \le 1$, for $n = 5, 10, 20$, with the interpolation nodes x_i being given by

    (a) $x_i = -1 + \frac{2i}{n}, \;\; i = 0, 1, 2, \dots, n;$

    (b) $x_i = \cos \frac{2i+1}{2n+2}\pi, \;\; i = 0, 1, 2, \dots, n.$

    Compute $\lambda_n(x)$ on a grid obtained by dividing each interval $[x_{i-1}, x_i]$, $i = 1, 2, \dots, n$, into 20 equal subintervals. Plot $\log_{10} \lambda_n(x)$ in case (a), and $\lambda_n(x)$ in case (b). Comment on the results.
28. Let $\omega_n(x) = \prod_{k=0}^{n} (x - k)$ and denote by x_n the location of the extremum of ω_n on [0,1], that is, the unique x in [0,1], where $\omega_n'(x) = 0$.

    (a) Prove or disprove that $x_n \to 0$ as $n \to \infty$.

    (b) Investigate the monotonicity of x_n as n increases.
29. Consider equidistant sampling points $x_k = k$ $(k = 0, 1, \dots, n)$ and $\omega_n(x) = \prod_{k=0}^{n} (x - k)$, $0 \le x \le n$.

    (a) Show that $\omega_n(x) = (-1)^{n+1}\omega_n(n - x)$. What kind of symmetry does this imply?

    (b) Show that $|\,\omega_n(x)| < |\,\omega_n(x + 1)|$ for nonintegral $x > (n - 1)/2$.

    (c) Show that the relative maxima of $|\,\omega_n(x)|$ increase monotonically (from the center of $[0, n]$ outward).
30. Let

    $$\lambda_n(x) = \sum_{i=0}^{n} |\ell_i(x)|$$

    be the Lebesgue function for polynomial interpolation at the distinct points $x_i \in [a, b]$, $i = 0, 1, \dots, n$, and $\Lambda_n = \|\lambda_n\|_\infty = \max_{a \le x \le b} |\lambda_n(x)|$ the Lebesgue constant. Let $p_n(f;\,\cdot\,)$ be the polynomial of degree $\le n$ interpolating f at the nodes x_i. Show that in the inequality

    $$\|p_n(f;\,\cdot\,)\|_\infty \le \Lambda_n \|f\|_\infty, \qquad f \in C[a, b],$$

equality can be attained for some $f = \varphi \in C[a,b]$. {*Hint*: let $\|\lambda_n\|_\infty = \lambda_n(x_\infty)$; take $\varphi \in C[a,b]$ piecewise linear and such that $\varphi(x_i) = \operatorname{sgn} \ell_i(x_\infty)$, $i = 0, 1, \dots, n$.}

31. (a) Let $x_0, x_1, \dots, x_n$ be $n+1$ distinct points in $[a,b]$ and $f_i = f(x_i)$, $i = 0, 1, \dots, n$, for some function f. Let $f_i^* = f_i + \varepsilon_i$, where $|\varepsilon_i| \le \varepsilon$. Use the Lagrange interpolation formula to show that $|p_n(f^*; x) - p_n(f; x)| \le \varepsilon\lambda_n(x)$, $a \le x \le b$, where $\lambda_n(x)$ is the Lebesgue function (cf. Ex. 30).
    (b) Show: $\lambda_n(x_j) = 1$ for $j = 0, 1, \dots, n$.
    (c) For quadratic interpolation at three equally spaced points, show that $\lambda_2(x) \le 1.25$ for any x between the three points.
    (d) Obtain $\lambda_2(x)$ for $x_0 = 0$, $x_1 = 1$, $x_2 = p$, where $p \gg 1$, and determine $\max_{1 \le x \le p} \lambda_2(x)$. How fast does this maximum grow with p? {*Hint*: to simplify the algebra, note from (b) that $\lambda_2(x)$ on $1 \le x \le p$ must be of the form $\lambda_2(x) = 1 + c(x-1)(p-x)$ for some constant c.}
32. In a table of the Bessel function $J_0(x) = \frac{1}{\pi}\int_0^\pi \cos(x \sin\theta)\mathrm{d}\theta$, where x is incremented in steps of size h, how small must h be chosen if the table is to be "linearly interpolable" with error less that 10^{-6} in absolute value? {*Point of information*: $\int_0^{\pi/2} \sin^2\theta \mathrm{d}\theta = \frac{\pi}{4}$.}
33. Suppose you have a table of the logarithm function $\ln x$ for positive integer values of x, and you compute $\ln 11.1$ by quadratic interpolation at $x_0 = 10$, $x_1 = 11$, $x_2 = 12$. Estimate the relative error incurred.
34. The "Airy function" $y(x) = \mathrm{Ai}(x)$ is a solution of the differential equation $y'' = xy$ satisfying appropriate initial conditions. It is known that $\mathrm{Ai}(x)$ on $[0, \infty)$ is monotonically decreasing to zero and $\mathrm{Ai}'(x)$ monotonically increasing to zero. Suppose you have a table of Ai and Ai$'$ (with tabular step h) and you want to interpolate

    (a) linearly between x_0 and x_1,
    (b) quadratically between x_0, x_1, and x_2,

    where x_0, $x_1 = x_0 + h$, $x_2 = x_0 + 2h$ are (positive) tabular arguments. Determine close upper bounds for the respective errors in terms of quantities $y_k = y(x_k)$, $y_k' = y'(x_k)$, $k = 0, 1, 2$, contained in the table.
35. The error in linear interpolation of f at x_0, x_1 is known to be

    $$f(x) - p_1(f; x) = (x - x_0)(x - x_1)\,\frac{f''(\xi(x))}{2}, \quad x_0 < x < x_1,$$

    if $f \in C^2[x_0, x_1]$. Determine $\xi(x)$ explicitly in the case $f(x) = \frac{1}{x}$, $x_0 = 1$, $x_1 = 2$, and find $\max_{1 \le x \le 2} \xi(x)$ and $\min_{1 \le x \le 2} \xi(x)$.
36. (a) Let $p_n(f; x)$ be the interpolation polynomial of degree $\le n$ interpolating $f(x) = \mathrm{e}^x$ at the points $x_i = i/n$, $i = 0, 1, 2, \dots, n$. Derive an upper bound for

    $$\max_{0 \le x \le 1} |\mathrm{e}^x - p_n(f; x)|$$

and determine the smallest n guaranteeing an error less than 10^{-6} on [0, 1]. {*Hint*: first show that for any integer i with $0 \le i \le n$ one has $\max_{0\le x\le 1} |(x - \frac{i}{n})(x - \frac{n-i}{n})| \le \frac{1}{4}$.}

(b) Solve the analogous problem for the nth-degree Taylor polynomial $t_n(x) = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!}$, and compare the result with the one in (a).

37. Let $x_0 < x_1 < x_2 < \cdots < x_n$ and $H = \max_{0\le i\le n-1}(x_{i+1} - x_i)$. Defining $\omega_n(x) = \prod_{i=0}^{n}(x-x_i)$, find an upper bound for $\|\omega_n\|_\infty = \max_{x_0\le x\le x_n} |\omega_n(x)|$ in terms of H and n. {*Hint*: assume $x_j \le x \le x_{j+1}$ for some $0 \le j < n$ and estimate $(x - x_j)(x - x_{j+1})$ and $\prod_{\substack{i\neq j \\ i\neq j+1}} (x - x_i)$ separately.}

38. Show that the power x^n on the interval $-1 \le x \le 1$ can be uniformly approximated by a linear combination of powers $1, x, x^2, \ldots, x^{n-1}$ with error $\le 2^{-(n-1)}$. In this sense, the powers of x become "less and less linearly independent" on [–1,1] with growing exponent n.

39. Determine

$$\min \max_{a\le x\le b} |a_0x^n + a_1x^{n-1} + \cdots + a_n|, \quad n \ge 1,$$

where the minimum is taken over all real $a_0, a_1, \ldots, a_n$ with $a_0 \neq 0$. {*Hint*: use Theorem 2.2.1.}

40. Let $a > 1$ and $\mathbb{P}_n^a = \{p \in \mathbb{P}_n : p(a) = 1\}$. Define $\hat{p}_n \in \mathbb{P}_n^a$ by $\hat{p}_n(x) = \frac{T_n(x)}{T_n(a)}$, where T_n is the Chebyshev polynomial of degree n, and let $\|\cdot\|_\infty$ denote the maximum norm on the interval $[-1, 1]$. Prove:

$$\|\hat{p}_n\|_\infty \le \|p\|_\infty \quad \text{for all } \ p \in \mathbb{P}_n^a.$$

{*Hint*: imitate the proof of Theorem 2.2.1.}

41. Let

$$f(x) = \int_5^\infty \frac{\mathrm{e}^{-t}}{t-x}\mathrm{d}t, \quad -1 \le x \le 1,$$

and let $p_{n-1}(f; \cdot)$ be the polynomial of degree $\le n-1$ interpolating f at the n Chebyshev points $x_\nu = \cos(\frac{2\nu-1}{2n}\pi)$, $\nu = 1, 2, \ldots, n$. Derive an upper bound for $\max_{-1\le x\le 1} |f(x) - p_{n-1}(f, x)|$.

42. Let f be a positive function defined on $[a, b]$ and assume

$$\min_{a\le x\le b} |f(x)| = m_0, \quad \max_{a\le x\le b} |f^{(k)}(x)| = M_k, \ \ k = 0, 1, 2, \ldots.$$

(a) Denote by $p_{n-1}(f; \cdot)$ the polynomial of degree $\le n - 1$ interpolating f at the n Chebyshev points (relative to the interval $[a, b]$). Estimate the maximum relative error $r_n = \max_{a\le x\le b} |(f(x) - p_{n-1}(f; x))/f(x)|$.

(b) Apply the result of (a) to $f(x) = \ln x$ on $I_r = \{\mathrm{e}^r \le x \le \mathrm{e}^{r+1}\}$, $r \ge 1$ an integer. In particular, show that $r_n \le \alpha(r, n)c^n$, where $0 < c < 1$ and α is slowly varying. Exhibit c.

(c) (This relates to the function $f(x) = \ln x$ of part (b).) How does one compute $f(\overline{x})$, $\overline{x} \in I_s$, from $f(x)$, $x \in I_r$?

43. (a) For quadratic interpolation on equally spaced points x_0, $x_1 = x_0 + h$, $x_2 = x_0 + 2h$, derive an upper bound for $\|f - p_2(f;\cdot)\|_\infty$ involving $\|f'''\|_\infty$ and h. (Here $\|u\|_\infty = \max_{x_0 \le x \le x_2} |u(x)|$.)
    (b) Compare the bound obtained in (a) with the analogous one for interpolation at the three Chebyshev points on $[x_0, x_2]$.
44. (a) Suppose the function $f(x) = \ln(2 + x)$, $-1 \le x \le 1$, is interpolated by a polynomial p_n of degree $\le n$ at the Chebyshev points $x_k = \cos\left(\frac{2k+1}{2n+2}\pi\right)$, $k = 0, 1, \dots, n$. Derive a bound for the maximum error $\|f - p_n\|_\infty = \max_{-1 \le x \le 1} |f(x) - p_n(x)|$.
    (b) Compare the result of (a) with bounds for $\|f - t_n\|_\infty$, where $t_n(x)$ is the nth-degree Taylor polynomial of f and where either Lagrange's form of the remainder is used or the full Taylor expansion of f.
45. Consider $f(t) = \cos^{-1} t$, $-1 \le t \le 1$. Obtain the least squares approximation $\hat{\varphi}_n \in \mathbb{P}_n$ of f relative to the weight function $w(t) = (1 - t)^{-\frac{1}{2}}$; that is, find the solution $\varphi = \hat{\varphi}_n$ of

$$\text{minimize} \quad \left\{ \int_{-1}^{1} [f(t) - \varphi(t)]^2 \frac{\mathrm{d}t}{\sqrt{1 - t^2}} \ : \ \varphi \in \mathbb{P}_n \right\}.$$

    Express $\hat{\varphi}_n$ in terms of Chebyshev polynomials $\pi_j(t) = T_j(t)$.
46. Compute $T_n'(0)$, where T_n is the Chebyshev polynomial of degree n.
47. Prove that the system of Chebyshev polynomials $\{T_k : 0 \le k < n\}$ is orthogonal with respect to the discrete inner product $(u, v) = \sum_{\nu=1}^{n} u(x_\nu)v(x_\nu)$, where x_ν are the Chebyshev points $x_\nu = \cos\frac{2\nu-1}{2n}\pi$.
48. Let $T_k(x)$ denote the Chebyshev polynomial of degree k. Clearly, $T_n(T_m(x))$ is a polynomial of degree $n \cdot m$. Identify it.
49. Let T_n denote the Chebyshev polynomial of degree $n \ge 2$. The equation

$$x = T_n(x)$$

    is an algebraic equation of degree n and hence has exactly n roots. Identify them.
50. For any x with $0 \le x \le 1$ show that $T_n(2x - 1) = T_{2n}(\sqrt{x})$.
51. Let $f(x)$ be defined for all $x \in \mathbb{R}$ and infinitely often differentiable on $\mathbb{R}$. Assume further that

$$|f^{(m)}(x)| \le 1, \quad \text{all } x \in \mathbb{R},\ m = 1, 2, 3, \dots.$$

    Let $h > 0$ and p_{2n-1} be the polynomial of degree $< 2n$ interpolating f at the $2n$ points $x = kh$, $k = \pm 1, \pm 2, \dots, \pm n$. For what values of h is it true that

$$\lim_{n \to \infty} p_{2n-1}(0) = f(0)\,?$$

(Note that $x = 0$ is *not* an interpolation node.) Explain why the convergence theory discussed in Sect. 2.2.3 does not apply here. {*Point of information*: $n! \sim \sqrt{2\pi n}(n/\mathrm{e})^n$ as $n \to \infty$ (Stirling's formula).}

52. (a) Let $x_i^C = \cos\left(\frac{2i+1}{2n+2}\pi\right)$, $i = 0, 1, \ldots, n$, be Chebyshev points on $[-1, 1]$. Obtain the analogous Chebyshev points t_i^C on $[a, b]$ (where $a < b$) and find an upper bound of $\prod_{i=0}^{n}(t - t_i^C)$ for $a \le t \le b$.
    (b) Consider $f(t) = \ln t$ on $[a, b]$, $0 < a < b$, and let $p_n(t) = p_n(f; t_0^{(n)}, t_1^{(n)}, \ldots, t_n^{(n)}; t)$. Given $a > 0$, how large can b be chosen such that $\lim_{n\to\infty} p_n(t) = f(t)$ for arbitrary nodes $t_i^{(n)} \in [a, b]$ and arbitrary $t \in [a, b]$?
    (c) Repeat (b), but with $t_i^{(n)} = t_i^C$ (see (a)).
53. Let $\mathbb{P}_m^+$ be the set of all polynomials of degree $\le m$ that are nonnegative on the real line,

$$\mathbb{P}_m^+ = \{p : \ \ p \in \mathbb{P}_m,\ p(x) \ge 0 \text{ for all } x \in \mathbb{R}\}.$$

Consider the following interpolation problem: find $p \in \mathbb{P}_m^+$ such that $p(x_i) = f_i$, $i = 0, 1, \ldots, n$, where $f_i \ge 0$ and x_i are distinct points on $\mathbb{R}$.

    (a) Show that, if $m = 2n$, the problem admits a solution for arbitrary $f_i \ge 0$.
    (b) Prove: if a solution is to exist for arbitrary $f_i \ge 0$, then, necessarily, $m \ge 2n$. {*Hint*: consider $f_0 = 1,\ f_1 = f_2 = \cdots = f_n = 0$.}

54. Defining forward differences by $\Delta f(x) = f(x + h) - f(x)$, $\Delta^2 f(x) = \Delta(\Delta f(x)) = f(x + 2h) - 2f(x + h) + f(x)$, and so on, show that

$$\Delta^k f(x) = k!h^k [x_0, x_1, \ldots, x_k]f,$$

where $x_j = x + jh,\ j = 0, 1, 2, \ldots$. Prove an analogous formula for backward differences.

55. Let $f(x) = x^7$. Compute the fifth divided difference [0,1,1,1,2,2]f of f. It is known that this divided difference is expressible in terms of the fifth derivative of f evaluated at some ξ, $0 < \xi < 2$ (cf. (2.117)). Determine ξ.
56. In this problem $f(x) = \mathrm{e}^x$ throughout.

    (a) Prove: for any real number t, one has

$$[t, t + 1, \ldots, t + n]f = \frac{(\mathrm{e} - 1)^n}{n!}\,\mathrm{e}^t.$$

{*Hint*: use induction on n.}

    (b) From (2.117) we know that

$$[0, 1, \ldots, n]f = \frac{f^{(n)}(\xi)}{n!}, \quad 0 < \xi < n.$$

Use the result in (a) to determine ξ. Is ξ located to the left or to the right of the midpoint $n/2$?

57. (Euler, 1734) Let $x_k = 10^k$, $k = 0, 1, 2, 3, \ldots$, and $f(x) = \log_{10} x$.
   (a) Show that

$$[x_0, x_1, \ldots, x_n]f = \frac{(-1)^{n-1}}{10^{n(n-1)/2}(10^n - 1)}, \quad n = 1, 2, 3, \ldots.$$

   {*Hint*: prove more generally

$$[x_r, x_{r+1}, \ldots, x_{r+n}]f = \frac{(-1)^{n-1}}{10^{rn+n(n-1)/2}(10^n - 1)}, \quad r \geq 0,$$

   by induction on n.}
   (b) Use Newton's interpolation formula to determine $p_n(x) = p_n(f; x_0, x_1, \ldots, x_n; x)$. Show that $\lim_{n\to\infty} p_n(x)$ exists for $1 \leq x < 10$. Is the limit equal to $\log_{10} x$? (Check, e.g., for $x = 9$.)

58. Show that

$$\frac{\partial}{\partial x_0}[x_0, x_1, \ldots, x_n]f = [x_0, x_0, x_1, \ldots, x_n]f,$$

   assuming f is differentiable at x_0. What about the partial derivative with respect to one of the other variables?

59. (a) For $n + 1$ distinct nodes x_ν, show that

$$[x_0, x_1, \ldots, x_n]f = \sum_{\nu=0}^{n} \frac{f(x_\nu)}{\prod_{\mu\neq\nu}(x_\nu - x_\mu)}.$$

   (b) Show that

$$[x_0, x_1, \ldots, x_n](fg_j) = [x_0, x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n]f,$$

   where $g_j(x) = x - x_j$.

60. (Mikeladze, 1941) Assuming $x_0, x_1, \ldots, x_n$ mutually distinct, show that

$$[\underbrace{x_0, x_0, \ldots, x_0}_{m \text{ times}}, x_1, x_2, \ldots, x_n]f$$
$$= \frac{[\overbrace{x_0, \ldots, x_0}^{m \text{ times}}]f}{\prod_{\mu=1}^{n}(x_0 - x_\mu)} + \sum_{\nu=1}^{n} \frac{[\overbrace{x_0, \ldots, x_0}^{(m-1) \text{ times}}, x_\nu]f}{\prod_{\substack{\mu=0 \\ \mu\neq\nu}}^{n}(x_\nu - x_\mu)}.$$

   {*Hint*: use induction on m.}

61. Determine the number of additions and the number of multiplications/divisions required

   (a) to compute all divided differences for $n + 1$ data points,
   (b) to compute all auxiliary quantities $\lambda_i^{(n)}$ in (2.103), and

(c) to compute $p_n(f;\cdot)$ (efficiently) from Newton's formula (2.111), once the divided differences are available. Compare with the analogous count for the barycentric formula (2.105), assuming all auxiliary quantities available. Overall, which, if any, of the two formulae can be computed more economically?

62. Consider the data $f(0) = 5$, $f(1) = 3$, $f(3) = 5$, $f(4) = 12$.

(a) Obtain the appropriate interpolation polynomial $p_3(f;x)$ in Newton's form.

(b) The data suggest that f has a minimum between $x = 1$ and $x = 3$. Find an approximate value for the location $x_{\min}$ of the minimum.

63. Let $f(x) = (1+a)^x$, $|a| < 1$. Show that $p_n(f;0,1,\dots,n;x)$ is the truncation of the binomial series for f to $n+1$ terms. {*Hint*: use Newton's form of the interpolation polynomial.}

64. Suppose f is a function on [0,3] for which one knows that

$$f(0) = 1,\ \ f(1) = 2,\ \ f'(1) = -1,\ \ f(3) = f'(3) = 0.$$

(a) Estimate $f(2)$, using Hermite interpolation.

(b) Estimate the maximum possible error of the answer given in (a) if one knows, in addition, that $f \in C^5[0,3]$ and $|f^{(5)}(x)| \le M$ on [0,3]. Express the answer in terms of M.

65. (a) Use Hermite interpolation to find a polynomial of lowest degree satisfying $p(-1) = p'(-1) = 0$, $p(0) = 1$, $p(1) = p'(1) = 0$. Simplify your expression for p as much as possible.

(b) Suppose the polynomial p of (a) is used to approximate the function $f(x) = [\cos(\pi x/2)]^2$ on $-1 \le x \le 1$.

(b1) Express the error $e(x) = f(x) - p(x)$ (for some *fixed* x in $[-1,1]$) in terms of an appropriate derivative of f.

(b2) Find an upper bound for $|e(x)|$ (still for a *fixed* $x \in [-1,1]$).

(b3) Estimate $\max_{-1\le x\le 1} |e(x)|$.

66. Consider the problem of finding a polynomial $p \in \mathbb{P}_n$ such that

$$p(x_0) = f_0,\ \ p'(x_i) = f_i',\quad i = 1,2,\dots,n,$$

where x_i, $i = 1,2,\dots,n$, are distinct nodes. (It is not excluded that $x_1 = x_0$.) This is neither a Lagrange nor a Hermite interpolation problem (why not?). Nevertheless, show that the problem has a unique solution and describe how it can be obtained.

67. Let

$$f(t) = \begin{cases} 0 & \text{if}\quad 0 \le t \le \frac{1}{2}, \\ 1 & \text{if}\quad \frac{1}{2} \le t \le 1. \end{cases}$$

(a) Find the linear least squares approximant $\hat{p}_1$ to f on $[0,1]$, that is, the polynomial $p_1 \in \mathbb{P}_1$ for which

$$\int_0^1 [p_1(t) - f(t)]^2 \mathrm{d}t = \min.$$

Use the normal equations with $\pi_0(t) = 1,\ \pi_1(t) = t$.

(b) Can you do better with continuous piecewise linear functions (relative to the partition $[0,1] = [0,\frac{1}{2}] \cup [\frac{1}{2},1]$)? Use the normal equations for the B-spline basis $B_0,\ B_1,\ B_2$ (cf. Sect. 2.2.2 and Ex. 13).

68. Show that $\mathbb{S}_m^m(\Delta) = \mathbb{P}_m$.
69. Let Δ be the subdivision

$$\Delta = [0,1] \cup [1,2] \cup [2,3]$$

of the interval [0,3]. Define the function s by

$$s(x) = \begin{cases} 2 - x(3 - 3x + x^2) & \text{if} \quad 0 \le x \le 1, \\ 1 & \text{if} \quad 1 \le x \le 2, \\ \frac{1}{4}\, x^2(3 - x) & \text{if} \quad 2 \le x \le 3. \end{cases}$$

To which class $\mathbb{S}_m^k(\Delta)$ does s belong?

70. In

$$s(x) = \begin{cases} p(x) & \text{if } 0 \le x \le 1, \\ (2-x)^3 & \text{if } 1 \le x \le 2 \end{cases}$$

determine $p \in \mathbb{P}_3$ such that $s(0) = 0$ and s is a cubic spline in $\mathbb{S}_3^2(\Delta)$ on the subdivision $\Delta = [0,1] \cup [1,2]$ of the interval [0,2]. Do you get a natural spline?

71. Let Δ: $a = x_1 < x_2 < x_3 < \cdots < x_n = b$ be a subdivision of $[a,b]$ into $n-1$ subintervals. What is the dimension of the space $\mathbb{S}_m^k = \{s \in C^k[a,b]\colon s|_{[x_i,x_{i+1}]} \in \mathbb{P}_m,\ i = 1,2,\ldots,n-1\}$?
72. Given the subdivision $\Delta:\ a = x_1 < x_2 < \cdots < x_n = b$ of $[a,b]$, determine a basis of "hat functions" for the space $\boldsymbol{S} = \{s \in \mathbb{S}_1^0 :\ s(a) = s(b) = 0\}$.
73. Let $\Delta:\ a = x_1 < x_2 < x_3 < \cdots < x_{n-1} < x_n = b$ be a subdivision of $[a,b]$ into $n-1$ subintervals. Suppose we are given values $f_i = f(x_i)$ of some function $f(x)$ at the points $x = x_i,\ i = 1,2,\ldots,n$. In this problem $s \in \mathbb{S}_2^1$ is a quadratic spline in $C^1[a,b]$ that interpolates f on Δ, that is, $s(x_i) = f_i$, $i = 1,2,\ldots,n$.

(a) Explain why one expects an additional condition to be required in order to determine s uniquely.

(b) Define $m_i = s'(x_i),\ i = 1,2,\ldots,n-1$. Determine $p_i := s\,\big|_{[x_i,x_{i+1}]}$, $i = 1,2,\ldots,n-1$, in terms of f_i, f_{i+1}, and m_i.

(c) Suppose one takes $m_1 = f'(a)$. (According to (a), this determines s uniquely.) Show how $m_2, m_3, \dots, m_{n-1}$ can be computed.

74. Let the subdivision Δ of $[a,b]$ be given by

$$\Delta: \ a = x_1 < x_2 < x_3 < \cdots < x_{n-1} < x_n = b, \ \ n \geq 2,$$

and let $f_i = f(x_i)$, $i = 1, 2, \dots, n$, for some function f. Suppose you want to interpolate this data by a quintic spline $s_5(f; \cdot)$ (a piecewise fifth-degree polynomial of smoothness class $C^4[a,b]$). By counting the number of parameters at your disposal and the number of conditions imposed, state how many additional conditions (if any) you expect are needed to make $s_5(f; \cdot)$ unique.

75. Let

$$\Delta: \quad a = x_1 < x_2 < x_3 < \cdots < x_{n-1} < x_n = b.$$

Consider the following problem: given $n-1$ numbers f_ν and $n-1$ points ξ_ν with $x_\nu < \xi_\nu < x_{\nu+1}$ ($\nu = 1, 2, \dots, n-1$), find a piecewise linear function $s \in \mathbb{S}_1^0(\Delta)$ such that

$$s(\xi_\nu) = f_\nu \quad (\nu = 1, 2, \dots, n-1), \quad s(x_1) = s(x_n).$$

Representing s in terms of the basis B_1, $B_2, \dots, B_n$ of "hat functions," determine the structure of the linear system of equations that you obtain for the coefficients c_j in $s(x) = \sum_{j=1}^{n} c_j B_j(x)$. Describe how you would solve the system.

76. Let $s_1(x) = 1 + c(x+1)^3$, $-1 \leq x \leq 0$, where c is a (real) parameter. Determine $s_2(x)$ on $0 \leq x \leq 1$ so that

$$s(x) := \begin{cases} s_1(x) & \text{if} \quad -1 \leq x \leq 0, \\ s_2(x) & \text{if} \quad 0 \leq x \leq 1 \end{cases}$$

is a natural cubic spline on $[-1, 1]$ with knots at $-1, 0, 1$. How must c be chosen if one wants $s(1) = -1$?

77. Derive (2.136).

78. Determine the quantities m_i in the variant of piecewise cubic Hermite interpolation mentioned at the end of Sect. 2.3.4(a).

79. (a) Derive the two extra equations for m_1, $m_2, \dots, m_n$ that result from the "not-a-knot" condition (Sect. 2.3.4, (b.4)) imposed on the cubic spline interpolant $s \in \mathbb{S}_3^2(\Delta)$ (with Δ as in Ex. 73).

    (b) Adjoin the first of these equations to the top and the second to the bottom of the system of $n-2$ equations derived in Sect. 2.3.4(b). Then apply elementary row operations to produce a tridiagonal system. Display the new matrix elements in the first and last equations, simplified as much as possible.

(c) Is the tridiagonal system so obtained diagonally dominant?

80. Let $\mathbb{S}_1^0(\Delta)$ be the class of continuous piecewise linear functions relative to the subdivision $a = x_1 < x_2 < \cdots < x_n = b$. Let $\|g\|_\infty = \max_{a \le x \le b} |g(x)|$, and denote by $s_1(g;\,\cdot\,)$ the piecewise linear interpolant (from $\mathbb{S}_1^0(\Delta)$) to g.

    (a) Show: $\|s_1(g;\,\cdot\,)\|_\infty \le \|g\|_\infty$ for any $g \in C[a,b]$.
    (b) Show: $\|f - s_1(f;\,\cdot\,)\|_\infty \le 2\|f - s\|_\infty$ for any $s \in \mathbb{S}_1^0$, $f \in C[a,b]$. {*Hint*: use additivity of $s_1(f;\,\cdot\,)$ with respect to f.}
    (c) Interpret the result in (b) when s is the best uniform spline approximant to f.

81. Consider the interval $[a,b] = [-1,1]$ and its subdivision $\Delta = [-1,0] \cup [0,1]$, and let $f(x) = \cos\frac{\pi}{2}x$, $-1 \le x \le 1$.

    (a) Determine the natural cubic spline interpolant to f on Δ.
    (b) Illustrate Theorem 2.3.2 by taking in turn $g(x) = p_2(f;-1,0,1;x)$ and $g(x) = f(x)$.
    (c) Discuss analogously the complete cubic spline interpolant to f on Δ' (cf. (2.149)) and the choices $g(x) = p_3(f;-1,0,1,1;x)$ and $g(x) = f(x)$.

Machine Assignments

1. (a) A simple-minded approach to best uniform approximation of a function $f(x)$ on [0,1] by a linear function $ax + b$ is to first discretize the problem and then, for various (appropriate) trial values of a, solve the problem of (discrete) uniform approximation of $f(x) - ax$ by a constant b (which admits an easy solution). Write a program to implement this idea.
    (b) Run your program for $f(x) = \mathrm{e}^x$, $f(x) = 1/(1+x)$, $f(x) = \sin\frac{\pi}{2}x$, $f(x) = x^\alpha$ $(\alpha = 2,3,4,5)$. Print the respective optimal values of a and b and the associated minimum error. What do you find particularly interesting in the results (if anything)?
    (c) Give a heuristic explanation (and hence exact values) for the results, using the known fact that the error curve for the optimal linear approximation attains its maximum modulus at three consecutive points $0 \le x_0 < x_1 < x_2 \le 1$ with alternating signs (*Principle of Alternation*).
2. (a) Determine the $(n+1)\times(n+1)$ matrix $\boldsymbol{A} = [a_{ij}]$, $a_{ij} = (B_i^n, B_j^n)$, of the normal equations relative to the Bernstein basis

$$B_j^n(t) = \binom{n}{j} t^j (1-t)^{n-j}, \quad j = 0,1,\ldots,n,$$

    and weight function $w(t) \equiv 1$ on [0,1]. {*Point of information*: $\int_0^1 t^k (1-t)^\ell \mathrm{d}t = k!\ell!/(k+\ell+1)!$}.

(b) Use Matlab to solve the normal equations of (a) for $n = 5 : 5 : 25$, when the function to be approximated is $f(t) \equiv 1$. What should the exact answer be? For each n, print the infinity norm of the error vector and an estimate of the condition number of $\boldsymbol{A}$. Comment on your results.

3. Compute discrete least squares approximations to the function $f(t) = \sin\left(\frac{\pi}{2}t\right)$ on $0 \le t \le 1$ by polynomials of the form

$$\varphi_n(t) = t + t(1-t)\sum_{j=1}^{n} c_j t^{j-1},\ n = 1(1)5,$$

using N abscissae $t_k = k/(N+1)$, $k = 1, 2, \ldots, N$, and equal weights 1. Note that $\varphi_n(0) = 0$, $\varphi_n(1) = 1$ are the exact values of f at $t = 0$ and $t = 1$, respectively. $\{$*Hint*: approximate $f(t) - t$ by a linear combination of $\pi_j(t) = t^j(1-t)$; $j = 1, 2, \ldots, n$.$\}$ Write a Matlab program for solving the normal equations $\boldsymbol{Ac} = \boldsymbol{b}$, $\boldsymbol{A} = [(\pi_i, \pi_j)]$, $\boldsymbol{b} = [(\pi_i, f - t)]$, $\boldsymbol{c} = [c_j]$, that does the computation in both single and double precision. For each $n = 1, 2, \ldots, 5$ output the following:

- the condition number of the system (computed in double precision);
- the maximum relative error in the coefficients, $\max_{1 \le j \le n} |(c_j^s - c_j^d)/c_j^d|$, where c_j^s are the single-precision values of c_j and c_j^d the double-precision values;
- the minimum and maximum error (computed in double precision),

$$e_{\min} = \min_{1 \le k \le N} |\varphi_n(t_k) - f(t_k)|,\ \ e_{\max} = \max_{1 \le k \le N} |\varphi_n(t_k) - f(t_k)|.$$

Make two runs:

(a) $N = 5, 10, 20$; (b) $N = 4$.

Comment on the results.

4. Write a program for discrete polynomial least squares approximation of a function f defined on [–1,1], using the inner product

$$(u, v) = \frac{2}{N+1}\sum_{i=0}^{N} u(t_i)v(t_i),\ \ t_i = -1 + \frac{2i}{N}.$$

Follow these steps.

(a) The recurrence coefficients for the appropriate (monic) orthogonal polynomials $\{\pi_k(t)\}$ are known explicitly:

$$\alpha_k = 0,\ \ k = 0, 1, \ldots, N;\quad \beta_0 = 2,$$

$$\beta_k = \left(1 + \frac{1}{N}\right)^2 \left(1 - \left(\frac{k}{N+1}\right)^2\right) \left(4 - \frac{1}{k^2}\right)^{-1},\ \ k = 1, 2, \ldots, N.$$

(You do not have to prove this.) Define $\gamma_k = \|\pi_k\|^2 = (\pi_k, \pi_k)$, which is known to be equal to $\beta_0\beta_1 \cdots \beta_k$ (cf. Ex. 20).

(b) Using the recurrence formula with coefficients α_k, β_k given in (a), generate an array $\boldsymbol{\pi}$ of dimension $(N+2, N+1)$ containing $\pi_k(t_\ell)$, $k = 0, 1, \ldots, N+1$; $\ell = 0, 1, \ldots, N$. (Here k is the row index and ℓ the column index.) Define $\mu_k = \max_{0 \le \ell \le N} |\pi_k(t_\ell)|$, $k = 1, 2, \ldots, N+1$. Print β_k, γ_k, and μ_{k+1} for $k = 0, 1, 2, \ldots, N$, where $N = 10$. Comment on the results.

(c) With $\hat{p}_n(t) = \sum_{k=0}^{n} \hat{c}_k \pi_k(t)$, $n = 0, 1, \ldots, N$, denoting the least squares approximation of degree $\le n$ to the function f on [–1,1], define

$$\|e_n\|_2 = \|\hat{p}_n - f\| = (\hat{p}_n - f, \hat{p}_n - f)^{1/2},$$
$$\|e_n\|_\infty = \max_{0 \le i \le N} |\hat{p}_n(t_i) - f(t_i)|.$$

Using the array $\boldsymbol{\pi}$ generated in part (b), compute $\hat{c}_n$, $\|e_n\|_2$, $\|e_n\|_\infty$, $n = 0, 1, \ldots, N$, for the following four functions:

$$f(t) = \mathrm{e}^{-t}, \quad f(t) = \ln(2+t), \quad f(t) = \sqrt{1+t}, \quad f(t) = |t|.$$

Be sure you compute $\|e_n\|_2$ as accurately as possible. For $N = 10$ and for each f, print $\hat{c}_n$, $\|e_n\|_2$, and $\|e_n\|_\infty$ for $n = 0, 1, 2, \ldots, N$. Comment on your results. In particular, from the information provided in the output, discuss to what extent the computed coefficients $\hat{c}_k$ may be corrupted by rounding errors.

5. (a) A Sobolev-type least squares approximation problem results if the inner product is defined by

$$(u, v) = \int_{\mathbb{R}} u(t)v(t)\mathrm{d}\lambda_0(t) + \int_{\mathbb{R}} u'(t)v'(t)\mathrm{d}\lambda_1(t),$$

where $\mathrm{d}\lambda_0$, $\mathrm{d}\lambda_1$ are positive measures. What does this type of approximation try to accomplish?

(b) Letting $\mathrm{d}\lambda_0(t) = \mathrm{d}t$, $\mathrm{d}\lambda_1(t) = \lambda \mathrm{d}t$ on [0,2], where $\lambda > 0$ is a parameter, set up the normal equations for the Sobolev-type approximation in (a) of the function $f(t) = \mathrm{e}^{-t^2}$ on [0,2] by means of a polynomial of degree $n-1$. Use the basis $\pi_j(t) = t^{j-1}$, $j = 1, 2, \ldots, n$. {*Hint*: express the components b_i of the right-hand vector of the normal equations in terms of the "incomplete gamma function" $\gamma(a, x) = \int_0^x t^{a-1}\mathrm{e}^{-t}\mathrm{d}t$ with $x = 4$, $a = i/2$.}

(c) Use Matlab to solve the normal equations for $n = 2:5$ and $\lambda = 0, .5, 1, 2$. Print

$$\|\hat{\varphi}_n - f\|_\infty \quad \text{and} \quad \|\hat{\varphi}_n' - f'\|_\infty, \quad n = 2, 3, 4, 5$$

(or a suitable approximation thereof) along with the condition numbers of the normal equations. {Use the following values for the incomplete gamma function: $\gamma(\frac{1}{2},4) = 1.764162781524843$, $\gamma(1,4) = 0.9816843611112658$, $\gamma(\frac{3}{2},4) = 0.8454501129849537$, $\gamma(2,4) = 0.9084218055563291$, $\gamma(\frac{5}{2},4) = 1.121650058367554$.} Comment on the results.

6. With $\omega_n(x) = \prod_{k=0}^{n}(x-k)$, let M_n be the largest, and m_n the smallest, relative maximum of $|\,\omega_n(x)|$. For $n = 5:5:30$ calculate M_n, m_n, and M_n/m_n, using Newton's method (cf. Chap 4, Sect. 4.6), and print also the respective number of iterations.

7. (a) Write a subroutine that produces the value of the interpolation polynomial $p_n(f; x_0, x_1, \dots, x_n; t)$ at any real t, where $n \geq 0$ is a given integer, x_i are $n+1$ distinct nodes, and f is any function available in the form of a function subroutine. Use Newton's interpolation formula and exercise frugality in the use of memory space when generating the divided differences. It is possible, indeed, to generate them "in place" in a single array of dimension $n+1$ that originally contains the values $f(x_i)$, $i = 0, 1, \dots, n$. {*Hint*: generate the divided differences from the bottom up.}

   (b) Run your routine on the function $f(t) = \frac{1}{1+t^2}$, $-5 \leq t \leq 5$, using $x_i = -5 + 10\frac{i}{n}$, $i = 0, 1, \dots, n$, and $n = 2:2:8$ (Runge's example). Plot the polynomials against the exact function.

8. (a) Write a Matlab function `y=tridiag(n,a,b,c,v)` for solving a tridiagonal (nonsymmetric) system

$$\begin{bmatrix} a_1 & c_1 & & & 0 \\ b_1 & a_2 & c_2 & & \\ & b_2 & a_3 & c_3 & \\ & & \ddots & \ddots & \ddots \\ & & & & c_{n-1} \\ 0 & & & b_{n-1} & a_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_{n-1} \\ v_n \end{bmatrix}$$

   by Gauss elimination without pivoting. Keep the program short.

   (b) Write a program for computing the natural spline interpolant $s_{\text{nat}}(f;\,\cdot)$ on an arbitrary partition $a = x_1 < x_2 < x_3 < \cdots < x_{n-1} < x_n = b$ of $[a,b]$. Print $\{i, \text{errmax}(i);\, i = 1, 2, \dots, n-1\}$, where

$$\text{errmax}(i) = \max_{1 \leq j \leq N} \left| s_{\text{nat}}(f; x_{i,j}) - f(x_{i,j}) \right|, \quad x_{i,j} = x_i + \frac{j-1}{N-1}\,\Delta x_i.$$

   (You will need the function `tridiag`.) Test the program for cases in which the error is zero (what are these, and why?).

   (c) Write a second program for computing the complete cubic spline interpolant $s_{\text{compl}}(f;\,\cdot)$ by modifying the program in (b) with a minimum of changes. Highlight the changes in the program listing. Apply (and justify) a test similar to that of (b).

(d) Run the programs in (b) and (c) for $[a, b] = [0, 1]$, $n = 11$, $N = 51$, and

(i) $x_i = \frac{i-1}{n-1}$, $i = 1, 2, \dots, n$; $f(x) = \mathrm{e}^{-x}$ and $f(x) = x^{5/2}$;

(ii) $x_i = \left(\frac{i-1}{n-1}\right)^2$, $i = 1, 2, \dots, n$; $f(x) = x^{5/2}$.

Comment on the results.

Selected Solutions to Exercises

11. (a) We have

$$(\pi_r, \pi_s) = \int_0^\infty \mathrm{e}^{-(r+s)t}\mathrm{d}t = -\frac{1}{r+s}\,\mathrm{e}^{-(r+s)t}\Big|_0^\infty = \frac{1}{r+s},$$

$$(\pi_r, f) = \int_0^1 \mathrm{e}^{-rt}\mathrm{d}t = -\frac{1}{r}\,\mathrm{e}^{-rt}\Big|_0^1 = \frac{1}{r}(1-\mathrm{e}^{-r}).$$

The normal equations, therefore, are

$$\sum_{s=1}^{n} \frac{1}{r+s}\,c_s = \frac{1}{r}\,(1-\mathrm{e}^{-r}), \quad r = 1, 2, \dots, n.$$

The matrix is the Hilbert matrix of order $n+1$ with the first column and last row removed.

(b) PROGRAM

```
%EXII_11B
%
f0='%8.0f %12.4e\n';
f1='%45.14e\n';
disp('       n      cond       solution')
for n=1:8
  A=hilb(n+1);
  A(:,1)=[];
  A(n+1,:)=[];
  x=(1:n)';
  b=(1-exp(-x))./x;
  c=A\b;
  cd=cond(A);
  fprintf(f0,n,cd)
  fprintf(f1,c)
  for i=1:201
    t=.01*(i-1);
    fa(i,n)=sum(c.*exp(-x*t));
  end
```

```
end
for i=1:11
  tf(i)=.1*(i-1);
  f(i)=1;
end
for i=1:201
  tfa(i)=.01*(i-1);
end
plot(tf,f);
hold on
plot(ones(size(tf)),tf);
plot(tfa,fa(:,1),’:’);
plot(tfa,fa(:,2),’-.’);
plot(tfa,fa(:,3),’--’);
plot(tfa,fa(:,4),’-’);
axis([0 2 0 1.5]);
hold off
```

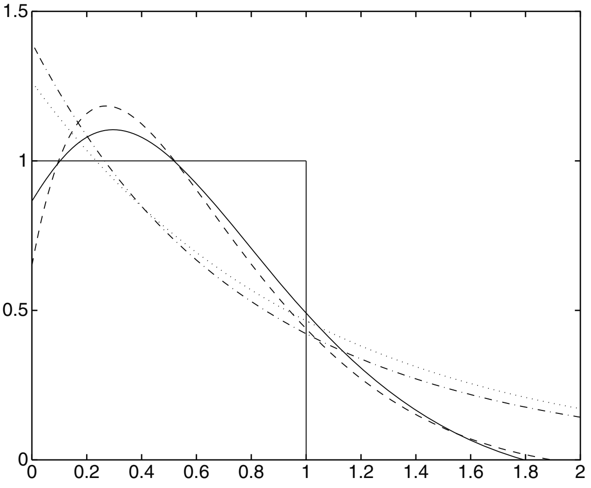
OUTPUT

```
>> EXII_11B
     n      cond             solution
     1   1.0000e+00
                       1.26424111765712e+00
     2   3.8474e+01
                       1.00219345775339e+00
                       3.93071489855589e-01
     3   1.3533e+03
                      -1.23430987802214e+00
                       9.33908483295774e+00
                      -7.45501111925180e+00
     4   4.5880e+04
                      -2.09728726098036e+00
                       1.58114152051443e+01
                      -2.03996718636248e+01
                       7.55105210088422e+00
     5   1.5350e+06
                       2.95960905289307e-01
                      -1.29075627900844e+01
                       8.01167511196597e+01
                      -1.26470845210147e+02
                       6.03098537899591e+01
     6   5.1098e+07
                       2.68879580265092e+00
```

```
                          -5.47821734938751e+01
                           3.03448008206274e+02
                          -6.28966173654415e+02
                           5.62805182233655e+02
                          -1.84248287095832e+02

 7   1.6978e+09
                           1.19410815562677e+00
                          -1.89096699709436e+01
                           3.44042318216034e+01
                           2.67846414188988e+02
                          -9.16935587561045e+02
                           9.99544328640241e+02
                          -3.66412000082403e+02
 8   5.6392e+10
                          -2.39677086853911e+00
                           9.42030165764484e+01
                          -1.09672261269167e+03
                           5.45217770865201e+03
                          -1.33593305457727e+04
                           1.71746576145770e+04
                          -1.11498207678274e+04
                           2.88841304234284e+03

>>
```

The condition numbers here are even a bit larger than the condition numbers of the Hilbert matrices of the same order (cf. Chap. 1, MA 9).

PLOTS



dotted line: n=1, *dashdotted line*: $n = 2$, *dashed line*: n=3, *solid line* n=4

16. (a) Let $p(t) = a_0 + a_1 t + a_2 t^2$. Then p satisfies the constraints if and only if $a_0 = 0,\ a_0 + a_1 + a_2 = 1$, that is,

$$p(t) = t^2 + a_1 t(1 - t).$$

Therefore, we need to minimize

$$\int_0^1 [f(t) - p(t)]^2 \mathrm{d}t = \int_0^1 [f(t) - t^2 - a_1 t(1 - t)]^2 \mathrm{d}t.$$

This is an unconstrained least squares problem for approximating the function $f(t) - t^2$ by a multiple of $\pi_1(t) = t(1 - t)$. The normal equation is

$$a_1 \int_0^1 [t(1 - t)]^2 \mathrm{d}t = \int_0^1 (f(t) - t^2) t(1 - t) \mathrm{d}t,$$

and yields the solution

$$\hat{p}(t) = t^2 + \hat{a}_1 t(1 - t),$$

where

$$\hat{a}_1 = \frac{\int_0^1 (f(t) - t^2) t(1 - t) \mathrm{d}t}{\int_0^1 [t(1 - t)]^2 \mathrm{d}t} = 30 \int_0^1 f(t) t(1 - t) \mathrm{d}t - \frac{3}{2}.$$

(b) If $f(t) = t^r$, then

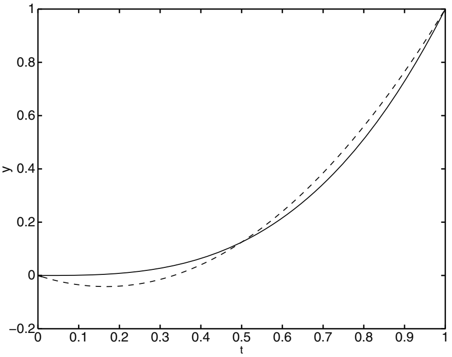
$$\hat{a}_1 = 30 \int_0^1 t^{r+1}(1 - t) \mathrm{d}t - \frac{3}{2} = \frac{30}{(r + 2)(r + 3)} - \frac{3}{2}$$

and

$$\begin{aligned} \hat{p}(t) &= t^2 + \left(\frac{30}{(r + 2)(r + 3)} - \frac{3}{2} \right) t(1 - t) \\ &= t \left\{ \frac{30}{(r + 2)(r + 3)} - \frac{3}{2} + \left(\frac{5}{2} - \frac{30}{(r + 2)(r + 3)} \right) t \right\}. \end{aligned}$$

For $r = 3$, this gives $\hat{p}(t) = \frac{1}{2} t(3t - 1)$.

Plot:



solid line: $y = t^3$
dashed line: $y = .5t(3t - 1)$

27. PROGRAM

```
%EXII_27 Lebesgue functions
%
n=5;
%n=10;
%n=20;
i=1:n+1; mu=1:n+1;
% equally spaced points
x=-1+2*(i-1)/n;
%
% Chebyshev points
%x=cos((2*(i-1)+1)*pi/(2*n+2));
%
iplot=0;
for k=2:n+1
%for k=1:n+2
  for j=1:21
    iplot=iplot+1;
    t(iplot)=x(k-1)+(j-1)*(x(k)-x(k-1))/20;
%     if k==1
%       t(iplot)=1+(j-1)*(x(1)-1)/20;
%     elseif k<=n+1
%       t(iplot)=x(k-1)+(j-1)*(x(k)-x(k-1))/20;
%     else
%       t(iplot)=x(n+1)+(j-1)*(-1-x(n+1))/20;
%     end
```

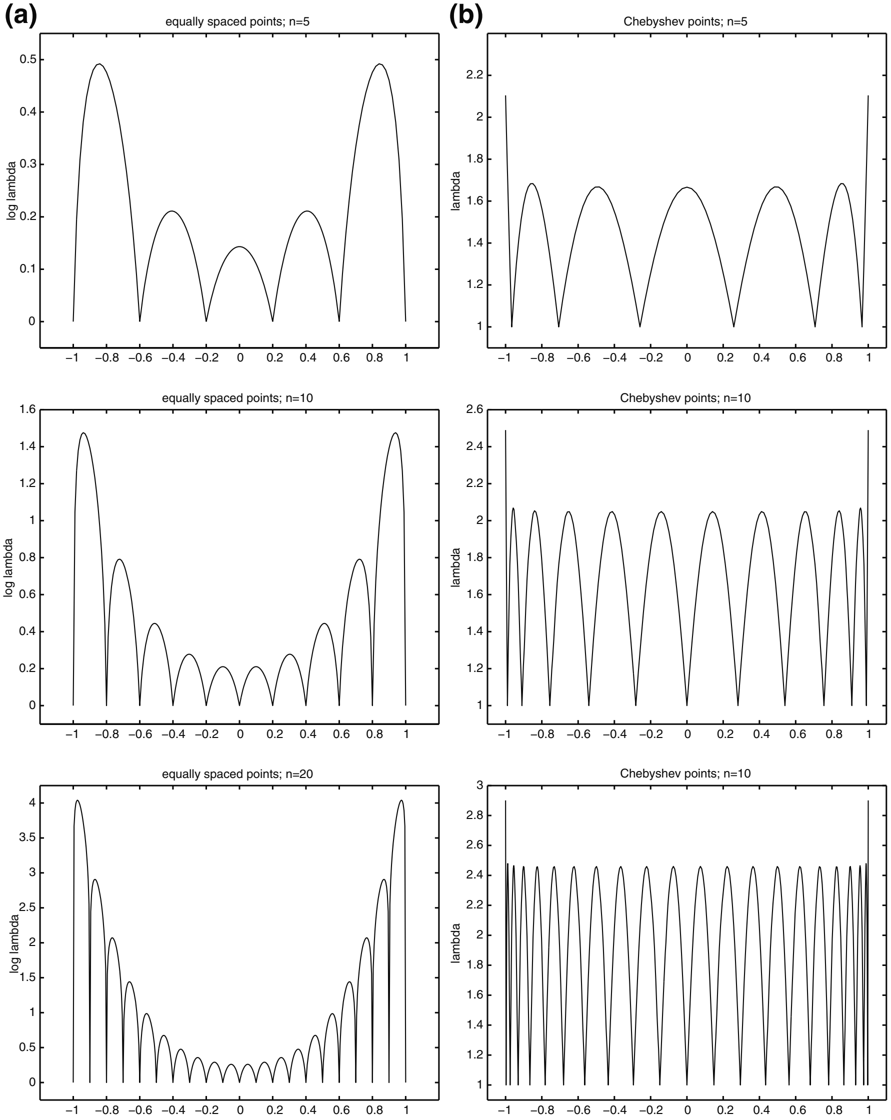
```
    s=0;
    for nu=1:n+1
      mu0=find(mu-nu);
      p=prod((t(iplot)-x(mu0))./(x(nu)-x(mu0)));
      s=s+abs(p);
    end
    leb(iplot)=s;
  end
end
plot(t,log10(leb))
%plot(t,leb)
axis([-1.2 1.2 -.05 .55])
%axis([-1.2 1.2 -.1 1.6])
%axis([-1.2 1.2 -.25 4.25])
%axis([-1.1 1.1 .9 2.4])
%axis([-1.1 1.1 .9 2.6])
%axis([-1.1 1.1 .9 3])
title('equally spaced points; n=5',
   'Fontsize',14)
%title('equally spaced points; n=10',
   'Fontsize',14)
%title('equally spaced points; n=20',
   'Fontsize',14)
%title('Chebyshev points; n=5','Fontsize',14)
%title('Chebyshev points; n=10','Fontsize',14)
ylabel('log lambda','Fontsize',14)
%ylabel('lambda','Fontsize',14)

 OUTPUT
 (on the next page)
```

At the interpolation nodes x_i, one clearly has $\lambda_n(x_i) = 1$. The local maxima of λ_n between successive interpolation nodes are almost equal, and relatively small, in case (b), but become huge near the endpoints of $[-1, 1]$ in case (a). In case (b), the global maxima occur at the endpoints ± 1.

36. We first prove the assertion of the *Hint*. One easily verifies that the function $\left|\left(x - \frac{i}{n}\right)\left(x - \frac{n-i}{n}\right)\right|$ on $[0, 1]$ is symmetric with respect to the midpoint $\frac{1}{2}$. Being quadratic, its maximum must occur either at $x = 0$ or at $x = \frac{1}{2}$, and hence is the larger of $\frac{i(n-i)}{n^2}$ and $\frac{(n-2i)^2}{4n^2}$. The former attains its maximum at $i = \frac{n}{2}$, the latter at $i = 0$ (and $i = n$). Either one equals $\frac{1}{4}$. Thus,

$$\max_{0 \le x \le 1} \left|\left(x - \frac{i}{n}\right)\left(x - \frac{n-i}{n}\right)\right| \le \frac{1}{4} \quad \text{for } i = 0, 1, \dots, n,$$

as claimed.



(a) We have

$$e^x - p_n(f;x) = \frac{e^{\xi(x)}}{(n+1)!}\prod_{k=0}^{n}\left(x-\frac{k}{n}\right), \quad 0 < \xi(x) < 1.$$

Here we use

$$\prod_{k=0}^{n}\left|x-\frac{k}{n}\right| = \sqrt{\prod_{i=0}^{n}\left|x-\frac{i}{n}\right|\left|x-\frac{n-i}{n}\right|}$$

along with the assertion of the *Hint* to obtain

$$\max_{0\le x\le 1}|\mathrm{e}^x - p_n(f;x)| \le \frac{\mathrm{e}}{(n+1)!}\left(\frac{1}{4}\right)^{\frac{n+1}{2}} = \frac{\mathrm{e}}{2^{n+1}(n+1)!}.$$

The smallest n making the upper bound $\le 10^{-6}$ is $n = 7$.

(b) From Taylor's formula,

$$\mathrm{e}^x - t_n(x) = \frac{\mathrm{e}^{\xi(x)}}{(n+1)!}\,x^{n+1}, \quad 0 < \xi(x) < 1.$$

Thus,

$$|\mathrm{e}^x - t_n(x)| \le \frac{\mathrm{e}}{(n+1)!}.$$

This bound is larger than the one in (a) by a factor of 2^{n+1}. Accordingly, for it to be $\le 10^{-6}$ now requires $n = 9$.

47. We have, for $0 \le k, \ell < n$,

$$\begin{aligned}(T_k, T_\ell) &= \sum_{\nu=1}^{n} T_k(x_\nu)T_\ell(x_\nu) = \sum_{\nu=1}^{n}\cos\left(k\,\frac{2\nu-1}{2n}\pi\right)\cos\left(\ell\,\frac{2\nu-1}{2n}\pi\right)\\ &= \frac{1}{2}\sum_{\nu=1}^{n}\left[\cos\left((k+\ell)\frac{2\nu-1}{2n}\pi\right) + \cos\left((k-\ell)\frac{2\nu-1}{2n}\pi\right)\right]\\ &= \frac{1}{2}\,\mathrm{Re}\left\{\sum_{\nu=1}^{n}\mathrm{e}^{\mathrm{i}(k+\ell)\frac{2\nu-1}{2n}\pi} + \sum_{\nu=1}^{n}\mathrm{e}^{\mathrm{i}(k-\ell)\frac{2\nu-1}{2n}\pi}\right\}\\ &= \frac{1}{2}\,\mathrm{Re}\left\{\mathrm{e}^{\mathrm{i}(k+\ell)\frac{\pi}{2n}}\sum_{\nu=1}^{n}\mathrm{e}^{\mathrm{i}(k+\ell)\frac{\nu-1}{n}\pi} + \mathrm{e}^{\mathrm{i}(k-\ell)\frac{\pi}{2n}}\sum_{\nu=1}^{n}\mathrm{e}^{\mathrm{i}(k-\ell)\frac{\nu-1}{n}\pi}\right\}.\end{aligned}$$

Assume $k \ne \ell$. Both sums in the last equation are finite geometric series and can thus be summed explicitly. One gets

$$\begin{aligned}(T_k, T_\ell) &= \frac{1}{2}\,\mathrm{Re}\left\{\mathrm{e}^{\mathrm{i}(k+\ell)\frac{\pi}{2n}}\,\frac{1-\mathrm{e}^{\mathrm{i}(k+\ell)\pi}}{1-\mathrm{e}^{i\frac{k+\ell}{n}\pi}} + \mathrm{e}^{\mathrm{i}(k-\ell)\frac{\pi}{2n}}\,\frac{1-\mathrm{e}^{\mathrm{i}(k-\ell)\pi}}{1-\mathrm{e}^{\mathrm{i}\frac{k-\ell}{n}\pi}}\right\}\\ &= \frac{1}{2}\,\mathrm{Re}\left\{\frac{i[1-\mathrm{e}^{\mathrm{i}(k+\ell)\pi}]}{2\sin\frac{k+\ell}{2n}\pi} + \frac{i[1-\mathrm{e}^{\mathrm{i}(k-\ell)\pi}]}{2\sin\frac{k-\ell}{2n}\pi}\right\},\end{aligned}$$

where the denominators are not zero by the assumption on k and ℓ. Now if $k+\ell$ (and hence also $k-\ell$) is even, both expressions in brackets are zero. If $k+\ell$ (and hence also $k-\ell$) is odd, the numerators are both $2i$, hence the real part equals zero. In either case, $(T_k, T_\ell) = 0$, as claimed.

An easy argument also shows that the value of the inner product is $\frac{n}{2}$ if $k = \ell > 0$, and n if $k = \ell = 0$.

The result follows more easily from the continuous orthogonality (cf. Sect. 2.2.4, (2.99)) by applying the Gauss–Chebyshev quadrature formula (cf. Chap. 3, Sect. 3.2.3 and Ex. 36).

57. (a) For $n = 1$, the assertion of the *Hint* is true for all $r \geq 0$ since

$$[x_r, x_{r+1}] = \frac{\log_{10} x_{r+1} - \log_{10} x_r}{x_{r+1} - x_r} = \frac{(r+1) - r}{10^r(10-1)} = \frac{1}{9 \cdot 10^r}.$$

Thus, assume the assertion to be true for some n and all $r \geq 0$. Then, by the property (2.113) of divided differences,

$$\begin{aligned}
&[x_r, x_{r+1}, \dots, x_{r+n}, x_{r+n+1}]f \\
&\quad = \frac{[x_{r+1}, x_{r+2}, \dots, x_{r+n+1}]f - [x_r, x_{r+1}, \dots, x_{r+n}]f}{x_{r+n+1} - x_r} \\
&\quad = \frac{(-1)^{n-1}}{10^{rn+n(n-1)/2}(10^n - 1)} \, \frac{1 - 10^n}{10^n(10^{r+n+1} - 10^r)} \\
&\quad = \frac{(-1)^n}{10^{rn+n(n-1)/2} \, 10^{n+r}(10^{n+1} - 1)} \\
&\quad = \frac{(-1)^n}{10^{r(n+1)+n(n+1)/2} \, (10^{n+1} - 1)},
\end{aligned}$$

which is precisely the assumed assertion with n replaced by $n+1$.

(b) Let $a_k = [x_0, x_1, \dots, x_k]f$. By Newton's formula, noting that $a_0 = \log_{10} 1 = 0$, we have

$$\begin{aligned}
p_n(x) &= \sum_{k=1}^{n} a_k (x-1)(x-10)\cdots(x-10^{k-1}) \\
&= \sum_{k=1}^{n} \frac{(-1)^{k-1}}{10^{k(k-1)/2}(10^k - 1)} \prod_{\ell=0}^{k-1} (x - 10^\ell) \\
&= -\sum_{k=1}^{n} \frac{1}{10^{k(k-1)/2}(10^k - 1)} \prod_{\ell=0}^{k-1} (10^\ell - x)
\end{aligned}$$

$$= -\sum_{k=1}^{n} \frac{1}{10^k - 1} \prod_{\ell=0}^{k-1} (1 - x/10^\ell)$$

$$= -\sum_{k=1}^{n} t_k(x),$$

where

$$t_k(x) := \frac{1}{10^k - 1} \prod_{\ell=0}^{k-1} (1 - x/10^\ell).$$

For $1 \le x < 10$, we have

$$|t_k(x)| < \left| \frac{(1-x)(1-x/10^{k-1})}{10^k - 1} \right| < \frac{9}{10^k - 1} \left(1 - \frac{1}{10^{k-1}}\right) < \frac{9}{10^k}.$$

Thus, the infinite series $\sum_{k=1}^{\infty} t_k(x)$ is majorized by the convergent geometric series $9 \sum_{k=1}^{\infty} 10^{-k}$ and therefore also converges. However, for $x = 9$, one computes

$$-\sum_{k=1}^{\infty} t_k(9) = \sum_{k=1}^{\infty} \frac{8}{10^k - 1} \prod_{\ell=1}^{k-1} (1 - 9/10^k)$$

$$= 0.89777\ldots < \log_{10}(9) = 0.95424\ldots.$$

(For an analysis of the discrepancy, see Gautschi [2008].)

75. Let $s(x) = \sum_{j=1}^{n} c_j B_j(x)$. Then, with points ξ_ν as defined, the first $n-1$ conditions imposed on s can be written as

$$c_1 B_1(\xi_1) + c_2 B_2(\xi_1) = f_1,$$

$$c_2 B_2(\xi_2) + c_3 B_3(\xi_2) = f_2,$$

$$\cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ldots\ .$$

$$c_{n-1} B_{n-1}(\xi_{n-1}) + c_n B_n(\xi_{n-1}) = f_{n-1}.$$

The last condition imposed is, since $B_1(x_1) = B_n(x_n) = 1$,

$$c_1 - c_n = 0.$$

The matrix of the system has the following structure:

$$\begin{bmatrix} \times & \times & & & & \\ & \times & \times & & & \\ & & \times & \times & & \\ & & & \ddots & \ddots & \\ & & & & \times & \times \\ 1 & & & & & -1 \end{bmatrix}$$

Note that $B_j(\xi_j) \neq 0$ for $j = 1, 2, \ldots, n-1$.

Solution: (1) Subtract a suitable multiple of the first equation from the last equation to create a zero in position $(n, 1)$. This produces a fill-in in position $(n, 2)$. (2) Subtract a suitable multiple of the second equation from the last to create a zero in position $(n, 2)$. This produces a fill-in in position $(n, 3)$, etc. After $n-1$ such operations one obtains a nonsingular upper bidiagonal system, which is quickly solved by back substitution.

79. (a) From Sect. 2.2.4, (2.140) and (2.141), the spline on $[x_i, x_{i+1}]$ is

$$(*) \qquad p_i(x) = c_{i,0} + c_{i,1}(x - x_i) + c_{i,2}(x - x_i)^2 + c_{i,3}(x - x_i)^3,$$

where

$$\binom{*}{*} \qquad \begin{aligned} c_{i,0} = f_i, \;\; c_{i,1} = m_i, \;\; c_{i,2} &= \frac{[x_i, x_{i+1}]f - m_i}{\Delta x_i} - c_{i,3}\Delta x_i, \\ c_{i,3} &= \frac{m_{i+1} + m_i - 2[x_i, x_{i+1}]f}{(\Delta x_i)^2}. \end{aligned}$$

The two "not-a-knot" conditions are $p_1'''(x_2) = p_2'''(x_2)$, $p_{n-2}'''(x_{n-1}) = p_{n-1}'''(x_{n-1})$. By (*), this yields

$$c_{1,3} = c_{2,3}, \quad c_{n-2,3} = c_{n-1,3}.$$

Substituting from $\binom{*}{*}$, the first equality becomes

$$\frac{m_2 + m_1 - 2[x_1, x_2]f}{(\Delta x_1)^2} = \frac{m_3 + m_2 - 2[x_2, x_3]f}{(\Delta x_2)^2},$$

or, after some elementary manipulations,

$$
\text{(1)}\qquad \begin{aligned} & m_1 + \left(1 - \left(\frac{\Delta x_1}{\Delta x_2}\right)^2\right) m_2 - \left(\frac{\Delta x_1}{\Delta x_2}\right)^2 m_3 \\ & = 2\left([x_1, x_2]f - \left(\frac{\Delta x_1}{\Delta x_2}\right)^2 [x_2, x_3]f\right) =: b_1. \end{aligned}
$$

Similarly, the second equality becomes

$$
(n)\qquad \begin{aligned} & m_{n-2} + \left(1 - \left(\frac{\Delta x_{n-2}}{\Delta x_{n-1}}\right)^2\right) m_{n-1} - \left(\frac{\Delta x_{n-2}}{\Delta x_{n-1}}\right)^2 m_n \\ & = 2\left([x_{n-2}, x_{n-1}]f - \left(\frac{\Delta x_{n-2}}{\Delta x_{n-1}}\right)^2 [x_{n-1}, x_n]f\right) =: b_n. \end{aligned}
$$

(b) The first equation (for $i = 2$) from Sect. 2.2.4, (2.145), is

$$
\text{(2)}\qquad \Delta x_2\, m_1 + 2(\Delta x_1 + \Delta x_2)\, m_2 + \Delta x_1\, m_3 = b_2.
$$

Multiply (2) by $\dfrac{\Delta x_1}{(\Delta x_2)^2}$ and add to (1) to get the new pair of equations

$$
\begin{cases} \left(1 + \dfrac{\Delta x_1}{\Delta x_2}\right) m_1 + \left(1 + \dfrac{\Delta x_1}{\Delta x_2}\right)^2 m_2 = b_1 + \dfrac{\Delta x_1}{(\Delta x_2)^2}\, b_2, \\ \Delta x_2\, m_1 + 2(\Delta x_1 + \Delta x_2)\, m_2 + \Delta x_1\, m_3 = b_2. \end{cases}
$$

This is the beginning of a tridiagonal system.

Similarly, the last equation (for $i = n - 1$) from Sect. 2.2.4, (2.145), is

$$
(n-1)\quad \Delta x_{n-1}\, m_{n-2} + 2(\Delta x_{n-2} + \Delta x_{n-1})\, m_{n-1} + \Delta x_{n-2}\, m_n = b_{n-1}.
$$

Multiply Eq. $(n - 1)$ by $\frac{1}{\Delta x_{n-1}}$ and subtract from (n); then the last two equations become

$$
\begin{cases} \Delta x_{n-1}\, m_{n-2} + 2(\Delta x_{n-2} + \Delta x_{n-1})\, m_{n-1} + \Delta x_{n-2}\, m_n = b_{n-1} \\ -\left(1 + \dfrac{\Delta x_{n-2}}{\Delta x_{n-1}}\right)^2 m_{n-1} - \dfrac{\Delta x_{n-2}}{\Delta x_{n-1}}\left(1 + \dfrac{\Delta x_{n-2}}{\Delta x_{n-1}}\right) m_n = b_n - \dfrac{1}{\Delta x_{n-1}}\, b_{n-1}. \end{cases}
$$

This is the end of the tridiagonal system.

(c) No: the system is *not* diagonally dominant, since in the first equation the diagonal element $1 + \frac{\Delta x_1}{\Delta x_2}$ is less than the other remaining element $\left(1 + \frac{\Delta x_1}{\Delta x_2}\right)^2$.

Selected Solutions to Machine Assignments

4. (a), (b) and (c)

PROGRAMS

```
%MAII_4ABC
%
%(a)
%
function [beta,gamma,mu,coeff,L2err, ...
   maxerr]=MAII_4ABC(N)
P=zeros(N+2,N+1); i=0:N; t=-1+2*i/N;
beta(1)=2; b=(1+1/N)^2; gamma(1)=2;
for k=1:N
  beta(k+1)=b*(1-(k/(N+1))^2) ...
    /(4-1/k^2);
  gamma(k+1)=beta(k+1)*gamma(k);
end
%
%(b) and (c)
%
P(1,:)=1; P(2,:)=t; mu(1)=max(abs(t));
for k=2:N+1
  P(k+1,:)=t.*P(k,:)-beta(k)*P(k-1,:);
  mu(k)=max(abs(P(k+1,:)));
end
for n=0:N
  coeff(n+1)=2*sum(P(n+1,:) ...
     .*f(t))/((N+1)*gamma(n+1));
end
for n=0:N
  emax=0; e2=0;
  for k=1:N+1
    e=abs(sum(coeff(1:n+1)’ ...
      .*P(1:n+1,k))-f(t(k)));
    if e>emax, emax=e; end
    e2=e2+e^2;
  end
  L2err(n+1)=sqrt(2*e2/(N+1));
  maxerr(n+1)=emax;
end

function y=f(x)
y=exp(-x);
```

```
%y=log(2+x);
%y=sqrt(1+x);
%y=abs(x);

%RUNMAII_4ABC  Driver program for
% MAII_4ABC
%
f0='%4.0f %20.15f %23.15e %23.15e\n';
f1='%12.0f %23.15e %12.4e %12.4e\n';
disp(['   k          beta(k)' ...
  '                gamma(k)' ...
  '                 mu(k+1)'])
N=10;
[beta,gamma,mu,coeff,L2err,maxerr] ...
  =MAII_4ABC(N);
for k=1:N+1
  fprintf(f0,k-1,beta(k),gamma(k), ...
  mu(k))
end
fprintf('\n')
disp(['        n      coefficients' ...
  '      L2 error     max error'])
for k=1:N+1
  fprintf(f1,k-1,coeff(k),L2err(k), ...
  maxerr(k))
end
```

```
OUTPUT

>> runMAII_4ABC
   k         beta(k)               gamma(k)                 mu(k+1)
   0    2.000000000000000   2.000000000000000e+00   1.000000000000000e+00
   1    0.400000000000000   8.000000000000002e-01   5.999999999999999e-01
   2    0.312000000000000   2.496000000000001e-01   2.879999999999998e-01
   3    0.288000000000000   7.188480000000004e-02   1.152000000000001e-01
   4    0.266666666666667   1.916928000000001e-02   7.680000000000001e-02
   5    0.242424242424242   4.647098181818186e-03   3.351272727272728e-02
   6    0.213986013986014   9.944140165289268e-04   1.488738461538463e-02
   7    0.180923076923077   1.799124436058489e-04   5.269231888111895e-03
   8    0.143058823529412   2.573806252055439e-05   1.834253163307282e-03
   9    0.100309597523220   2.581774692464279e-06   5.068331109138542e-04
  10    0.052631578947368   1.358828785507516e-07   3.771414197649772e-17

   n       coefficients          L2 error       max error
   0   1.212203623058161e+00   1.0422e+00     1.5061e+00     f(t)=exp(-t)
   1  -1.123748299778268e+00   2.7551e-01     3.8233e-01
   2   5.430255798492442e-01   4.7978e-02     5.6515e-02
   3  -1.774967744700318e-01   6.0942e-03     5.7637e-03
   4   4.380735440218084e-02   5.9354e-04     7.1705e-04
   5  -8.681309127655327e-03   4.5392e-05     5.0328e-05
   6   1.436822757642024e-03  2.7410e-06     3.1757e-06
   7  -2.041262174764023e-04  1.2894e-07     1.3676e-07
   8   2.539983220653184e-05  4.5185e-09     5.2244e-09
   9  -2.811356175489033e-06  1.0329e-10     1.4201e-10
  10   2.800374538854939e-07  6.2576e-14     7.9492e-14

   n       coefficients        L2 error    max error
   0   6.379455015198038e-01  4.8331e-01   6.3795e-01     f(t)=ln(2+t)
   1   5.341350646266596e-01  7.3165e-02   1.0381e-01
   2  -1.436962628313260e-01  1.4112e-02   1.7593e-02
   3   5.149163971020859e-02  2.9261e-03   3.2053e-03
   4  -2.066713722106771e-02  6.1199e-04   8.2444e-04
   5   8.790920250468457e-03  1.2409e-04   1.4929e-04
   6  -3.863610725685766e-03  2.3550e-05   3.0261e-05
   7   1.730112538790927e-03  4.0112e-06   4.5316e-06
   8  -7.825109066954439e-04  5.7410e-07   6.9079e-07
   9   3.553730415235034e-04  5.9485e-08   8.1789e-08
  10  -1.613718817644612e-04  2.1657e-14   2.7534e-14

   n       coefficients        L2 error    max error
   0   9.134654065768736e-01  5.7547e-01   9.1347e-01   f(t)=sqrt(1+t)
   1   6.165636213969754e-01  1.6444e-01   2.9690e-01
   2  -2.799173478370132e-01  8.6512e-02   1.2895e-01
   3   2.654751232178156e-01  4.9173e-02   7.8888e-02
   4  -2.969055755002559e-01  2.6985e-02   4.4684e-02
   5   3.416320385824934e-01  1.3632e-02   1.8447e-02
   6  -3.862066935480817e-01  6.1238e-03   9.0935e-03
   7   4.215059528049150e-01  2.3530e-03   3.0774e-03
   8  -4.411293520434504e-01  7.2661e-04   9.1223e-04
   9   4.416771232533437e-01  1.5591e-04   2.1437e-04
  10  -4.229517654415031e-01  2.7984e-14   3.5305e-14

   n       coefficients        L2 error    max error
   0   5.454545454545454e-01  4.5272e-01   5.4545e-01   f(t)=|t|
   1   5.046468293750710e-17  4.5272e-01   5.4545e-01
   2   8.741258741258736e-01  1.1933e-01   1.9580e-01
   3   0.000000000000000e+00  1.1933e-01   1.9580e-01
   4  -7.284382284382317e-01  6.3786e-02   1.1189e-01
   5  -5.429698379835253e-16  6.3786e-02   1.1189e-01
   6   1.531862745098003e+00  4.1655e-02   6.9107e-02
   7  -3.506197370654419e-15  4.1655e-02   6.9107e-02
   8  -6.118812656642364e+00  2.7776e-02   3.8191e-02
   9   2.061545898679865e-13  2.7776e-02   3.8191e-02
  10   7.535204475298005e+01  3.9494e-14   5.0709e-14
>>
```

Comments

- Note that the last entry in the mu column vanishes, confirming that π_{N+1} vanishes at all the $N + 1$ nodes t_ℓ (cf. Ex. 22(b)).
- The calculation of $\hat{c}_n$ is subject to severe cancellation errors as n increases. Indeed, from the formula for the coefficient $\hat{c}_n$ (cf. (2.24)),

$$\hat{c}_n = 2\sum_{\ell=0}^{N} f(t_\ell)\pi_n(t_\ell)/((N+1)\gamma_n)),$$

one expects $\gamma_n \cdot \hat{c}_n$, being a "mean value" of the quantities $\pi_n(t_\ell)$, to have the order of magnitude of these quantitites, i.e., of μ_n, unless there is considerable cancellation in the summation, in which case $\gamma_n \cdot \hat{c}_n$ is much smaller in absolute value than μ_n. That, in fact, is clearly observed in our output when n gets large.

- The maximum error for $n = N$ should be zero since π_N interpolates. This is confirmed reasonably well in the output.
- e^{-t}: Note the rapid convergence. This is because the exponential function is an entire function, hence very smooth.
- $\ln(2 + t)$: Remarkably good convergence in spite of the logarithmic singularity at $t = -2$, a distance of 1 from the left endpoint of [–1,1].
- $\sqrt{1+t}$: Slow convergence because of $f'(t) \to \infty$ as $t \to -1$. There is a branch-point singularity at $t = -1$.
- $|t|$: Extremely slow convergence since f is not differentiable at $t = 0$. Since f is *even*, the approximation for n odd is exactly the same as the one for the preceding even n. This is evident from the L_2 and maximum errors and from the vanishing of the odd-numbered coefficients.

7. (a) PROGRAM

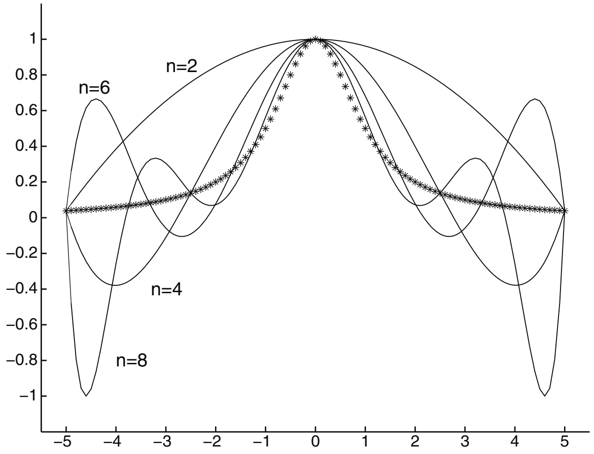
```
%MAII_7AB
%
hold on
it=(0:100)'; t=-5+it/10;
y=1./(1+t.^2);
plot(t,y,'k*')
axis([-5.5 5.5 -1.2 1.2])
for n=2:2:8;
  i=(0:n)'; it=(0:100)';
  x=-5+10*i/n; t=-5+it/10;
  y=pnewt(n,x,t);
  plot(t,y)
end
hold off

%PNEWT
%
function y=pnewt(n,x,t)
d=zeros(n+1,1);
d=f(x);
if n==0
  y=d(1);
  return
end
for j=1:n
```

```
  for i=n:-1:j
    d(i+1)=(d(i+1)-d(i))/(x(i+1)-x(i+1-j));
  end
end
y=d(n+1);
for i=n:-1:1
  y=d(i)+(t-x(i)).*y;
end

function y=f(x)
y=1./(1+x.^2);
```

(b)

```
 OUTPUT
```

The interpolation polynomials are drawn as solid lines, the exact function as black stars.



The "Runge phenomenon", i.e., the violent oscillations of the interpolants near the end points, is clearly evident.

8. (a)

```
 PROGRAM

%TRIDIAG
%
%   Gauss elimination without pivoting for a nxn (not
%   necessarily symmetric) tridiagonal system with nonzero
%   diagonal elements a, subdiagonal elements b, superdiagonal
%   element c, and right-hand vector v. The solution vector
%   is y. The vectors a and v will undergo changes by the
%   routine.
%
```

```
function y=tridiag(n,a,b,c,v)
y=zeros(n,1);
for i=2:n
  r=b(i-1)/a(i-1);
  a(i)=a(i)-r*c(i-1);
  v(i)=v(i)-r*v(i-1);
end
y(n)=v(n)/a(n);
for i=n-1:-1:1
  y(i)=(v(i)-c(i)*y(i+1))/a(i);
end
```

(b) The natural spline on the interval $[x_i, x_{i+1}]$ is (cf. (2.140), (2.141))

$$s_{\text{nat}}(x) = c_{i,0} + c_{i,1}(x - x_i) + c_{i,2}(x - x_i)^2 + c_{i,3}(x - x_i)^3, \quad x_i \le x \le x_{i+1},$$

where

$$\begin{aligned}
c_{i,0} &= f_i, \\
c_{i,1} &= m_i, \\
c_{i,2} &= \frac{[x_i, x_{i+1}]f - m_i}{\Delta x_i} - c_{i,3}\Delta x_i, \\
c_{i,3} &= \frac{m_{i+1} + m_i - 2[x_i, x_{i+1}]f}{(\Delta x_i)^2},
\end{aligned}$$

and the vector $\boldsymbol{m} = [m_1, m_2, \dots, m_n]^{\mathrm{T}}$ satisfies the tridiagonal system of equations (cf. Sect. 2.2.4, (b.3))

$$\begin{aligned}
2m_1 + m_2 &= b_1 \\
(\Delta x_2)m_1 + 2(\Delta x_1 + \Delta x_2)m_2 + (\Delta x_1)m_3 &= b_2 \\
\cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad & \cdots\cdots \\
(\Delta x_{n-1})m_{n-2} + 2(\Delta x_{n-2} + \Delta x_{n-1})m_{n-1} + (\Delta x_{n-2})m_n &= b_{n-1} \\
m_{n-1} + 2m_n &= b_n
\end{aligned}$$

where

$$\begin{aligned}
b_1 &= 3[x_1, x_2]f \\
b_2 &= 3\{[(\Delta x_2)[x_1, x_2]f + (\Delta x_1)[x_2, x_3]f\} \\
\cdots \quad & \cdots \quad \cdots \quad \cdots \\
b_{n-1} &= 3\{(\Delta x_{n-1})[x_{n-2}, x_{n-1}]f + (\Delta x_{n-2})[x_{n+1}, x_n]f\} \\
b_n &= 3[x_{n-1}, x_n]f
\end{aligned}$$

PROGRAM (for natural spline)

```
%MAII_8B
%
f0=’%8.0f %12.4e\n’;
n=11; N=51;
a=zeros(n,1); b=zeros(n-1,1); c=b;
i=(1:n)’; j=(1:N)’;
x=(i-1)/(n-1);
%x=((i-1)/(n-1)).^2;
f=exp(-x);
%f=sqrt(x).^5;
dx=x(2:n)-x(1:n-1); df=(f(2:n)-f(1:n-1))./dx;
a(1)=2; a(n)=2; b(n-1)=1; c(1)=1;
v(1)=3*df(1); v(n)=3*df(n-1);
a(2:n-1)=2*(dx(1:n-2)+dx(2:n-1));
b(1:n-2)=dx(2:n-1); c(2:n-1)=dx(1:n-2);
v(2:n-1)=3*(dx(2:n-1).*df(1:n-2)+dx(1:n-2).*df(2:n-1));
m=tridiag(n,a,b,c,v);
c0=f(1:n-1); c1=m(1:n-1);
c3=(m(2:n)+m(1:n-1)-2*df)./(dx.^2);
c2=(df-m(1:n-1))./dx-c3.*dx;
emax=zeros(n-1,1);
for i=1:n-1
  xx=x(i)+((j-1)/(N-1))*dx(i);
  t=xx-x(i);
  s=c3(i);
  s=t.*s+c2(i);
  s=t.*s+c1(i);
  s=t.*s+c0(i);
  emax(i)=max(abs(s-exp(-xx)));
%  emax(i)=max(abs(s-sqrt(xx).^5));
  fprintf(f0,i,emax(i))
end
```

(c) For the complete spline, only two small changes need to be made, as indicated by comment lines in the program below.

PROGRAM (for complete spline)

```
%MAII_8C
%
f0=’%8.0f %12.4e\n’;
n=11; N=51;
a=zeros(n,1); b=zeros(n-1,1); c=b;
i=(1:n)’; j=(1:N)’;
x=(i-1)/(n-1);
% x=((i-1)/(n-1)).^2;
f=exp(-x);
```

```
   %f=sqrt(x).^5;
%
% The next statement is new and does not occur in the
% program of (b)
%
   fder_1=-1; fder_n=-exp(-1);
   %fder_1=0; fder_n=5/2;
   dx=x(2:n)-x(1:n-1); df=(f(2:n)-f(1:n-1))./dx;
%
% The next two lines differ from the corresponding lines
% in the program of (b)
%
   a(1)=1; a(n)=1; b(n-1)=0; c(1)=0;
   v(1)=fder_1; v(n)=fder_n;
   a(2:n-1)=2*(dx(1:n-2)+dx(2:n-1));
   b(1:n-2)=dx(2:n-1); c(2:n-1)=dx(1:n-2);
   v(2:n-1)=3*(dx(2:n-1).*df(1:n-2)+dx(1:n-2).*df(2:n-1));
   m=tridiag(n,a,b,c,v);
   c0=f(1:n-1); c1=m(1:n-1);
   c3=(m(2:n)+m(1:n-1)-2*df)./(dx.^2);
   c2=(df-m(1:n-1))./dx-c3.*dx;
   emax=zeros(n-1,1);
   for i=1:n-1
     xx=x(i)+((j-1)/(N-1))*dx(i);
     t=xx-x(i);
     s=c3(i);
     s=t.*s+c2(i);
     s=t.*s+c1(i);
     s=t.*s+c0(i);
     emax(i)=max(abs(s-exp(-xx)));
   %  emax(i)=max(abs(s-sqrt(xx).^5));
     fprintf(f0,i,emax(i))
   end
```

(d) OUTPUT

```
>> MAII_8B                                >> MAII_8C
  1   4.9030e-04  f(x)=exp(-x)             2.5589e-07  f(x)=exp(-x)
  2   1.3163e-04  natural                  2.2123e-07  complete
  3   3.5026e-05  spline                   2.0294e-07  spline
  4   9.5467e-06  uniform                  1.8288e-07  uniform
  5   2.2047e-06  partition                1.6568e-07  partition
  6   4.2094e-07                           1.4984e-07
  7   3.4559e-06                           1.3568e-07
  8   1.2809e-05                           1.2247e-07
  9   4.8441e-05                           1.1190e-07
 10   1.8036e-04                           9.7227e-08
>>                                        >>
>> MAII_8B                                >> MAII_8C
```

```
  1   2.0524e-04  f(x)=x^(5/2)          4.4346e-05  f(x)=x^(5/2)
  2   5.3392e-05  natural               9.9999e-06  complete
  3   1.6192e-05  spline                4.7121e-06  spline
  4   2.7607e-06  uniform               3.9073e-07  uniform
  5   1.2880e-06  partition             9.8538e-07  partition
  6   9.8059e-06                        5.2629e-07
  7   3.4951e-05                        4.7131e-07
  8   1.3252e-04                        3.6500e-07
  9   4.9310e-04                        3.1193e-07
 10   1.8416e-03                        2.4850e-07
>>                                    >>
>> MAII_8B                            >> MAII_8C
  1   6.6901e-07  f(x)=x^(5/2)          1.0809e-07  f(x)=x^(5/2)
  2   2.3550e-07  natural               6.0552e-07  complete
  3   1.1749e-06  spline                9.1261e-07  spline
  4   1.6950e-06  nonuniform            1.3558e-06  nonuniform
  5   1.5853e-06  partition             1.7319e-06  partition
  6   1.6441e-05                        2.1329e-06
  7   6.8027e-05                        2.5242e-06
  8   3.2950e-04                        2.9138e-06
  9   1.4755e-03                        3.3225e-06
 10   6.5448e-03                        3.6393e-06
>>                                    >>
```

Comments

- Testing: For the natural spline, the error should be exactly zero if f is any linear function. (Not for arbitrary cubics, since f'' does not vanish at $x = 0$ and $x = 1$, unless f is linear.) For the complete spline, the error is zero for any cubic, if one sets $m_1 = f'(0)$ and $m_n = f'(1)$. Example: $f(x) = x^3$, $m_1 = 0$, $m_n = 3$.
- The natural spline for the uniform partition is relatively inaccurate near the endpoints, as expected.
- The complete spline is uniformly accurate for $f(x) = \mathrm{e}^{-x}$ but still relatively inaccurate near $x = 0$ for $f(x) = x^{5/2}$ on account of the “square root” singularity (of f''') at $x = 0$.
- Nonuniform partition (for $f(x) = x^{5/2}$): The natural spline is accurate near $x = 0$ because of the nodes being more dense there, but is still inaccurate at the other end. The complete spline is remarkably accurate at both ends, as well as elsewhere.

Chapter 3
Numerical Differentiation and Integration

Differentiation and integration are infinitary concepts of calculus; that is, they are defined by means of a limit process – the limit of the difference quotient in the first instance, the limit of Riemann sums in the second. Since limit processes cannot be carried out on the computer, we must replace them by finite processes. The tools to do so come from the theory of polynomial interpolation (Chap. 2, Sect. 2.2). They not only provide us with approximate formulae for the limits in question, but also permit us to estimate the errors committed and discuss convergence.

3.1 Numerical Differentiation

For simplicity, we consider only the first derivative; analogous techniques apply to higher-order derivatives.

The problem can be formulated as follows: for a given differentiable function f, approximate the derivative $f'(x_0)$ in terms of the values of f at x_0 and at nearby points $x_1, x_2, \ldots, x_n$ (not necessarily equally spaced or in natural order). Estimate the error of the approximation obtained.

In Sect. 3.1.1, we solve this problem by means of interpolation. Examples are given in Sect. 3.1.2, and the problematic nature of numerical differentiation in the presence of rounding errors is briefly discussed in Sect. 3.1.3.

3.1.1 A General Differentiation Formula for Unequally Spaced Points

The idea is simply to differentiate not $f(\cdot)$, but its interpolation polynomial $p_n(f; x_0, \ldots, x_n; \cdot)$. By carrying along the error term of interpolation, we can analyze the error committed.

W. Gautschi, *Numerical Analysis*, DOI 10.1007/978-0-8176-8259-0_3,

Thus, recall from Chap. 2, Sect. 2.2, that, given the $n+1$ distinct points $x_0, x_1, \dots, x_n$, we have

$$f(x) = p_n(f;x) + r_n(x), \tag{3.1}$$

where the interpolation polynomial can be written in Newton's form

$$\begin{aligned} p_n(f;x) = f_0 &+ (x-x_0)[x_0,x_1]f + (x-x_0)(x-x_1)[x_0,x_1,x_2]f + \cdots \\ &+ (x-x_0)(x-x_1)\cdots(x-x_{n-1})[x_0,x_1,\dots,x_n]f, \end{aligned} \tag{3.2}$$

and the error term in the form

$$r_n(x) = (x-x_0)(x-x_1)\cdots(x-x_n)\,\frac{f^{(n+1)}(\xi(x))}{(n+1)!}, \tag{3.3}$$

assuming (as we do) that f has a continuous $(n+1)$st derivative in an interval that contains all x_i and x. Differentiating (3.2) with respect to x and then putting $x = x_0$ gives

$$\begin{aligned} p_n'(f;x_0) = [x_0,x_1]f &+ (x_0-x_1)[x_0,x_1,x_2]f + \cdots \\ &+ (x_0-x_1)(x_0-x_2)\cdots(x_0-x_{n-1})[x_0,x_1,\dots,x_n]f. \end{aligned} \tag{3.4}$$

Similarly, from (3.3) (assuming that f has in fact $n+2$ continuous derivatives in an appropriate interval), we get

$$r_n'(x_0) = (x_0-x_1)(x_0-x_2)\cdots(x_0-x_n)\,\frac{f^{(n+1)}(\xi(x_0))}{(n+1)!}. \tag{3.5}$$

Therefore, differentiating (3.1), we find

$$f'(x_0) = p_n'(f;x_0) + e_n, \tag{3.6}$$

where the first term on the right, given by (3.4), represents the desired approximation, and the second,

$$e_n = r_n'(x_0), \tag{3.7}$$

given by (3.5), the respective error. If $H = \max_i |x_0 - x_i|$, we clearly obtain from (3.5) that

$$e_n = O(H^n) \quad \text{as} \quad H \to 0. \tag{3.8}$$

We can thus get approximation formulae of arbitrarily high order, but those with large n are of limited practical use; cf. Sect. 3.1.3.

3.1.2 Examples

The most important uses of differentiation formulae are made in the discretization of differential equations – ordinary or partial. In these applications, the spacing of the points is usually uniform, but unequally distributed points arise when partial differential operators are to be discretized near the boundary of the domain of interest.

1. $n = 1, x_1 = x_0 + h$. Here,

$$p_1'(f; x_0) = [x_0, x_1]f = \frac{f_1 - f_0}{h},$$

and (3.6) in conjunction with (3.7) and (3.5) gives

$$f'(x_0) = \frac{f_1 - f_0}{h} + e_1, \quad e_1 = -h\,\frac{f''(\xi)}{2}, \tag{3.9}$$

provided $f \in C^3[x_0, x_1]$. (Taylor's formula, actually, shows that $f \in C^2[x_0, x_1]$ suffices.) Thus, the error is of $O(h)$ as $h \to 0$.

2. $n = 2, x_1 = x_0 + h, x_2 = x_0 - h$. We also use the suggestive notation $x_2 = x_{-1}$, $f_2 = f_{-1}$. Here,

$$p_2'(f; x_0) = [x_0, x_1]f + (x_0 - x_1)[x_0, x_1, x_2]f. \tag{3.10}$$

The table of divided differences is:

$$\begin{array}{llll} x_{-1} & f_{-1} & & \\ x_0 & f_0 & \dfrac{f_0 - f_{-1}}{h} & \\ x_1 & f_1 & \dfrac{f_1 - f_0}{h} & \dfrac{f_1 - 2f_0 + f_{-1}}{2h^2}. \end{array}$$

Therefore,

$$p_2'(f; x_0) = \frac{f_1 - f_0}{h} - h\,\frac{f_1 - 2f_0 + f_{-1}}{2h^2} = \frac{f_1 - f_{-1}}{2h},$$

and (3.6), (3.7), and (3.5) give, if $f \in C^3[x_{-1}, x_1]$,

$$f'(x_0) = \frac{f_1 - f_{-1}}{2h} + e_2, \quad e_2 = -h^2\,\frac{f'''(\xi)}{6}. \tag{3.11}$$

Both approximations (3.9) and (3.11) are difference quotients; the former, however, is "one-sided" whereas the latter is "symmetric." As can be seen, the symmetric difference quotient is one order more accurate than the one-sided difference quotient.

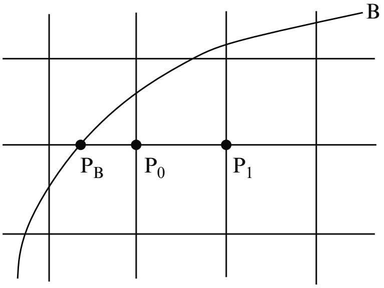


Fig. 3.1 Partial derivative near the boundary

3. $n = 2, x_1 = x_0 + h, x_2 = x_0 + 2h$. In this case, we have the following table of divided differences,

$$\begin{array}{llll} x_0 & f_0 & & \\ x_1 & f_1 & \dfrac{f_1 - f_0}{h} & \\ x_2 & f_2 & \dfrac{f_2 - f_1}{h} & \dfrac{f_2 - 2f_1 + f_0}{2h^2} \end{array}$$

and (3.10) now gives

$$p_2'(f; x_0) = \frac{f_1 - f_0}{h} - h\,\frac{f_2 - 2f_1 + f_0}{2h^2} = \frac{-f_2 + 4f_1 - 3f_0}{2h};$$

hence, by (3.7) and (3.5),

$$f'(x_0) = \frac{-f_2 + 4f_1 - 3f_0}{2h} + e_2, \quad e_2 = h^2\,\frac{f'''(\xi)}{3}. \tag{3.12}$$

Compared to (3.11), this formula also is accurate to $O(h^2)$, but the error is now about twice as large, in modulus, than before. One always pays for destroying symmetry!

4. For a function $u = u(x, y)$ of two variables, approximate $\partial u/\partial x$ "near the boundary."

Consider the points $P_0(x_0, y_0)$, $P_1(x_0 + h, y_0)$, $P_B(x_0 - \beta h, y_0)$, $0 < \beta < 1$ (see Fig. 3.1); the problem is to approximate $(\partial u/\partial x)(P_0)$ in terms of $u_0 = u(P_0)$, $u_1 = u(P_1)$, $u_B = u(P_B)$.

The relevant table of divided differences is:

$$\begin{array}{llll} x_B & u_B & & \\ x_0 & u_0 & \dfrac{u_0 - u_B}{\beta h} & \\ x_1 & u_1 & \dfrac{u_1 - u_0}{h} & \dfrac{\beta(u_1 - u_0) - (u_0 - u_B)}{\beta h(1 + \beta)h} \end{array}.$$

Thus,

$$p_2'(u; P_0) = \frac{u_1 - u_0}{h} - h\,\frac{\beta(u_1 - u_0) - (u_0 - u_B)}{\beta h(1+\beta)h} = \frac{\beta^2 u_1 + (1-\beta^2)u_0 - u_B}{\beta(1+\beta)h},$$

and the error is given by

$$e_2 = -\frac{\beta}{6}\,h^2\,\frac{\partial^3 u}{\partial x^3}(\xi, y_0).$$

3.1.3 Numerical Differentiation with Perturbed Data

Formulae for numerical differentiation become more accurate as the spacing h between evaluation points is made smaller, provided the function to be differentiated is sufficiently smooth. This, however, is true only in theory, since in practice the data are usually inaccurate, if for no other reason than rounding, and the problem of cancellation becomes more acute as h gets smaller. There will be a point of diminishing returns, beyond which the errors increase rather than decrease.

To give a simple analysis of this, take the symmetric differentiation formula (3.11),

$$f'(x_0) = \frac{f_1 - f_{-1}}{2h} + e_2,\quad e_2 = -h^2\,\frac{f'''(\xi)}{6}. \tag{3.13}$$

Suppose now that what are known are not the exact values $f_{\pm 1} = f(x_0 \pm h)$, but slight perturbations of them, say,

$$f_1^* = f_1 + \varepsilon_1,\; f_{-1}^* = f_{-1} + \varepsilon_{-1},\quad |\varepsilon_{\pm 1}| \le \varepsilon. \tag{3.14}$$

Then our formula (3.13) becomes

$$f'(x_0) = \frac{f_1^* - f_{-1}^*}{2h} - \frac{\varepsilon_1 - \varepsilon_{-1}}{2h} + e_2. \tag{3.15}$$

Here, the first term on the right is what we actually compute (assuming, for simplicity, that h is machine-representable and roundoff errors in forming the difference quotient are neglected). The corresponding error, therefore, is

$$E_2 = f'(x_0) - \frac{f_1^* - f_{-1}^*}{2h} = -\frac{\varepsilon_1 - \varepsilon_{-1}}{2h} + e_2$$

and can be estimated by

$$|E_2| \le \frac{\varepsilon}{h} + \frac{M_3}{6}\,h^2,\quad M_3 = \max_{[x_{-1},x_1]} |f'''|. \tag{3.16}$$

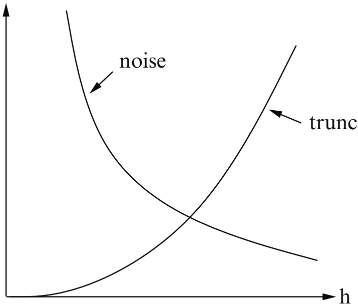


Fig. 3.2 Truncation and noise error in numerical differentiation

The bound on the right is best possible. It consists of two parts, the term ε/h, which is due to noise in the data, and $\frac{1}{6}M_3h^2$, which is the truncation error introduced by replacing the derivative by a finite difference expression. Their behavior is shown in Fig. 3.2.

If we denote the bound in (3.16) by $E(h)$,

$$E(h) = \frac{\varepsilon}{h} + \frac{M_3}{6}h^2, \tag{3.17}$$

then by determining its minimum, one finds

$$E(h) \geq E(h_0), \quad h_0 = \left(\frac{3\varepsilon}{M_3}\right)^{1/3},$$

and

$$E(h_0) = \frac{3}{2}\left(\frac{M_3}{3}\right)^{1/3}\varepsilon^{2/3}. \tag{3.18}$$

This shows that even in the best of circumstances, the error is $O(\varepsilon^{2/3})$, not $O(\varepsilon)$, as one would hope. This represents a significant loss of accuracy.

The same problem persists, indeed is more severe, in higher-order formulae. The only way one can escape from this dilemma is to use not *difference* formulae, but *summation* formulae, that is, integration. But to do this, one has to go into the complex plane and assume that the definition of f can be extended into a domain of the complex plane containing x_0. Then one can use Cauchy's theorem,

$$f'(x_0) = \frac{1}{2\pi\mathrm{i}}\oint_C \frac{f(z)}{(z-x_0)^2}\,\mathrm{d}z = \frac{1}{2\pi r}\int_0^{2\pi} \mathrm{e}^{-\mathrm{i}\theta} f(x_0 + r\mathrm{e}^{\mathrm{i}\theta})\mathrm{d}\theta, \tag{3.19}$$

in combination with numerical integration (cf. Sect. 3.2). Here, C was taken to be a circular contour about x_0 with radius r, with r chosen such that $z = x_0 + r\mathrm{e}^{\mathrm{i}\theta}$ remains in the domain of analyticity of f. Since the result is real, one can replace the integrand by its real part.

3.2 Numerical Integration

The basic problem is to calculate the definite integral of a given function f, extended over a finite interval $[a,b]$. If f is well behaved, this is a routine problem for which the simplest integration rules, such as the composite trapezoidal or Simpson's rule (Sect. 3.2.1) will be quite adequate, the former having an edge over the latter if f is periodic with period $b-a$. Complications arise if f has an integrable singularity, or the interval of integration extends to infinity (which is just another manifestation of singular behavior). By breaking up the integral, if necessary, into several pieces, it can be assumed that the singularity, if its location is known, is at one (or both) ends of the interval $[a, b]$. Such "improper" integrals can usually be treated by weighted quadrature; that is, one incorporates the singularity into a weight function, which then becomes one factor of the integrand, leaving the other factor well behaved. The most important example of this is Gaussian quadrature relative to such a weight function (Sects. 3.2.2–3.2.4). Finally, it is possible to accelerate the convergence of quadrature schemes by suitable recombinations. The best-known example of this is Romberg integration (Sect. 3.2.7).

3.2.1 The Composite Trapezoidal and Simpson's Rules

These may be regarded as the workhorses of numerical integration. They will do the job when the interval is finite and the integrand unproblematic. The trapezoidal rule is sometimes surprisingly effective even on infinite intervals.

Both rules are obtained by applying the simplest kind of interpolation on subintervals of the decomposition

$$a = x_0 < x_1 < x_2 < \cdots < x_{n-1} < x_n = b,\ x_k = a + kh,\ \ h = \frac{b-a}{n} \tag{3.20}$$

of the interval $[a,b]$. In the trapezoidal rule, one interpolates linearly on each subinterval $[x_k, x_{k+1}]$, and obtains

$$\int_{x_k}^{x_{k+1}} f(x)\mathrm{d}x = \int_{x_k}^{x_{k+1}} p_1(f;x)\mathrm{d}x + \int_{x_k}^{x_{k+1}} R_1(x)\mathrm{d}x, \tag{3.21}$$

where

$$\begin{aligned} p_1(f;x) &= f_k + (x-x_k)[x_k, x_{k+1}]f, \\ R_1(x) &= (x-x_k)(x-x_{k+1})\,\frac{f''(\xi(x))}{2}. \end{aligned} \tag{3.22}$$

Here, $f_k = f(x_k)$, and we assumed that $f \in C^2[a,b]$. The first integral on the right of (3.21) is easily obtained as the area of a trapezoid with "bases" f_k, f_{k+1} and "height" h, or else by direct integration of $p_1(f;\,\cdot\,)$ in (3.22). To the second integral, we can apply the Mean Value Theorem of integration, since $(x - x_k)(x - x_{k+1})$ has constant (negative) sign on $[x_k, x_{k+1}]$. The result is

$$\int_{x_k}^{x_{k+1}} f(x)\mathrm{d}x = \frac{h}{2}\,(f_k + f_{k+1}) - \frac{1}{12}\,h^3 f''(\xi_k), \tag{3.23}$$

where $x_k < \xi_k < x_{k+1}$. This is the *elementary trapezoidal rule*. Summing over all subintervals gives the *composite trapezoidal rule*

$$\int_a^b f(x)\mathrm{d}x = h\left(\frac{1}{2}\,f_0 + f_1 + \cdots + f_{n-1} + \frac{1}{2}\,f_n\right) + E_n^{\mathrm{T}}(f), \tag{3.24}$$

with error term

$$E_n^{\mathrm{T}}(f) = -\frac{1}{12}\,h^3 \sum_{k=0}^{n-1} f''(\xi_k).$$

This is not a particularly elegant expression for the error. We can simplify it by writing

$$E_n^{\mathrm{T}}(f) = -\frac{1}{12}\,h^2 \cdot (b-a)\left[\frac{1}{n}\sum_{k=0}^{n-1} f''(\xi_k)\right]$$

and noting that the expression in brackets is a mean value of second-derivative values, hence certainly contained between the algebraically smallest and largest value of the second derivative f'' on $[a,b]$. Since the function f'' was assumed continuous on $[a,b]$, it takes on every value between its smallest and largest, in particular, the bracketed value in question, at some interior point, say, ξ, of $[a,b]$. Consequently,

$$E_n^{\mathrm{T}}(f) = -\frac{1}{12}(b-a)\,h^2 f''(\xi), \quad a < \xi < b. \tag{3.25}$$

Since f'' is bounded in absolute value on $[a,b]$, this shows that $E_n^{\mathrm{T}}(f) = O(h^2)$ as $h \to 0$. In particular, the composite trapezoidal rule converges as $h \to 0$ (or, equivalently, $n \to \infty$) in (3.24), provided $f \in C^2[a,b]$.

It should be noted that (3.25) holds only for real-valued functions f and cannot be applied to complex-valued functions; cf. (3.29).

One expects an improvement if instead of linear interpolation one uses quadratic interpolation over two consecutive subintervals. This gives rise to the composite Simpson's formula.[1] Its "elementary" version, analogous to (3.23), is

$$\int_{x_k}^{x_{k+2}} f(x)\mathrm{d}x = \frac{h}{3}(f_k + 4f_{k+1} + f_{k+2}) - \frac{1}{90}\,h^5 f^{(4)}(\xi_k),\ x_k < \xi_k < x_{k+2}, \tag{3.26}$$

[1] Thomas Simpson (1710–1761) was an English mathematician, self-educated, and author of many textbooks popular at the time. Simpson published his formula in 1743, but it was already known to Cavalieri [1639], Gregory [1668], and Cotes [1722], among others.

where it has been assumed that $f \in C^4[a,b]$. The remainder term shown in (3.26) does not come about as easily as before in (3.23), since the Mean Value Theorem is no longer applicable, the factor $(x-x_k)\,(x-x_{k+1})\,(x-x_{k+2})$ changing sign at the midpoint of $[x_k, x_{k+2}]$. However, an alternative derivation of (3.26), using Hermite interpolation (cf. Ex. 9), not only produces the desired error term, but also explains its unexpectedly large order $O(h^5)$. If n is even, we can sum up all $n/2$ contributions in (3.26) and obtain the *composite Simpson's rule*,

$$\int_a^b f(x)\mathrm{d}x = \frac{h}{3}\,(f_0 + 4f_1 + 2f_2 + 4f_3 + 2f_4 + \cdots + 4f_{n-1} + f_n) + E_n^{\mathrm{S}}(f),$$

$$E_n^{\mathrm{S}}(f) = -\frac{1}{180}(b-a)\,h^4 f^{(4)}(\xi), \;\; a < \xi < b. \tag{3.27}$$

The error term in (3.27) is the result of a simplification similar to the one previously carried out for the trapezoidal rule (cf. Ex. 9(c)). Comparing it with the one in (3.25), we see that we gained *two* orders of accuracy without any appreciable increase in work (same number of function evaluations). This is the reason why Simpson's rule has long been, and continues to be, one of the most popular general-purpose integration methods.

The composite trapezoidal rule, nevertheless, has its own advantages. Although it integrates exactly polynomials of degree 1 only, it does much better with *trigonometric* polynomials. Suppose, indeed (for simplicity), that the interval [a,b] is [0,2π], and denote by $\mathbb{T}_m[0,2\pi]$ the class of trigonometric polynomials of degree m,

$$\mathbb{T}_m[0,2\pi] = \{t(x):\; t(x) = a_0 + a_1\cos x + a_2\cos 2x + \cdots + a_m\cos mx$$
$$+\; b_1\sin x + b_2\sin 2x + \cdots + b_m\sin mx\}\,.$$

Then

$$E_n^{\mathrm{T}}(f) = 0 \;\text{ for all }\; f \in \mathbb{T}_{n-1}[0,2\pi]. \tag{3.28}$$

This is most easily verified by taking for f the complex exponential $e_\nu(x) = \mathrm{e}^{\mathrm{i}\nu x}$ $(= \cos\nu x + \mathrm{i}\sin\nu x),\; \nu = 0,1,2,\ldots:$

$$E_n^{\mathrm{T}}(e_\nu) = \int_0^{2\pi} e_\nu(x)\mathrm{d}x - \frac{2\pi}{n}\left[\frac{1}{2}\,e_\nu(0) + \sum_{k=1}^{n-1} e_\nu(k\cdot 2\pi/n) + \frac{1}{2}\,e_\nu(2\pi)\right]$$
$$= \int_0^{2\pi} \mathrm{e}^{\mathrm{i}\nu x}\mathrm{d}x - \frac{2\pi}{n}\sum_{k=0}^{n-1} \mathrm{e}^{\mathrm{i}\nu k\cdot 2\pi/n}.$$

When $\nu = 0$, this is clearly zero, and otherwise, since $\int_0^{2\pi}\mathrm{e}^{\mathrm{i}\nu x}\mathrm{d}x = (i\,\nu)^{-1}$ $\cdot\,\mathrm{e}^{\mathrm{i}\nu x}\big|_0^{2\pi} = 0$,

$$E_n^{\mathrm{T}}(e_\nu) = \begin{cases} -2\pi & \text{if } \nu = 0 \;(\mathrm{mod}\, n),\;\; \nu > 0, \\ -\dfrac{2\pi}{n}\,\dfrac{1-\mathrm{e}^{\mathrm{i}\nu n\cdot 2\pi/n}}{1-\mathrm{e}^{\mathrm{i}\nu\cdot 2\pi/n}} = 0 \;\text{ if }\; \nu \neq 0 \;(\mathrm{mod}\, n). \end{cases} \tag{3.29}$$

In particular, $E_n^{\mathrm{T}}(e_\nu) = 0$ for $\nu = 0, 1, \ldots, n-1$, which proves (3.28). Taking real and imaginary parts in (3.29) gives

$$E_n^{\mathrm{T}}(\cos\nu\,\cdot) = \left\{\begin{matrix} -2\pi,\ \nu = 0\ (\mathrm{mod}\, n), \quad \nu \neq 0, \\ 0 \text{ otherwise}, \end{matrix}\right\}, \qquad E_n^{\mathrm{T}}(\sin\nu\,\cdot) = 0.$$

Therefore, if f is 2π-periodic and has a uniformly convergent Fourier expansion

$$f(x) = \sum_{\nu=0}^{\infty} [a_\nu(f)\cos\nu x + b_\nu(f)\sin\nu x], \tag{3.30}$$

where $a_\nu(f)$, $b_\nu(f)$ are the "Fourier coefficients" of f, then

$$\begin{aligned} E_n^{\mathrm{T}}(f) &= \sum_{\nu=0}^{\infty} \left[a_\nu(f)E_n^{\mathrm{T}}(\cos\nu\,\cdot) + b_\nu(f)E_n^{\mathrm{T}}(\sin\nu\,\cdot)\right] \\ &= -2\pi\sum_{\ell=1}^{\infty} a_{\ell\cdot n}(f). \end{aligned} \tag{3.31}$$

From the theory of Fourier series, it is known that the Fourier coefficients of f go to zero faster the smoother f is. More precisely, if $f \in C^r[\mathbb{R}]$, then $a_\nu(f) = O(\nu^{-r})$ as $\nu \to \infty$ (and similarly for $b_\nu(f)$). Since by (3.31), $E_n^{\mathrm{T}}(f) \approx -2\pi a_n(f)$, it follows that

$$E_n^{\mathrm{T}}(f) = O(n^{-r}) \ \text{ as } \ n \to \infty \ \ (f \in C^r[\mathbb{R}],\ \ 2\pi\text{−periodic}), \tag{3.32}$$

which, if $r > 2$, is better than $E_n^{\mathrm{T}}(f) = O(n^{-2})$, valid for nonperiodic functions f. In particular, if $r = \infty$, then the trapezoidal rule converges faster than any power of n^{-1}. It should be noted, however, that f must be smooth on the whole real line $\mathbb{R}$. (See (3.19) for an example.) Starting with a function $f \in C^r[0, 2\pi]$ and extending it periodically to $\mathbb{R}$ will *not* in general produce a function $f \in C^r[\mathbb{R}]$.

Another instance in which the composite trapezoidal rule excels is for functions f defined on $\mathbb{R}$ and having the following properties for some $r \geq 1$,

$$\begin{aligned} &f \in C^{2r+1}[\mathbb{R}], \quad \int_{\mathbb{R}} |f^{(2r+1)}(x)|\mathrm{d}x < \infty, \\ &\lim_{x\to-\infty} f^{(2\rho-1)}(x) = \lim_{x\to\infty} f^{(2\rho-1)}(x) = 0, \ \ \rho = 1, 2, \ldots, r. \end{aligned} \tag{3.33}$$

In this case, it can be shown that

$$\int_{\mathbb{R}} f(x)\mathrm{d}x = h\sum_{k=-\infty}^{\infty} f(kh) + E(f;h) \tag{3.34}$$

has an error $E(f;h)$ satisfying $E(f;h) = O(h^{2r+1})$, $h \to 0$. Therefore, again, if (3.33) holds for all $r = 1, 2, 3, \ldots$, then the error goes to zero faster than any power of h.

3.2.2 (Weighted) Newton–Cotes and Gauss Formulae

A weighted quadrature formula is a formula of the type

$$\int_a^b f(t)w(t)\mathrm{d}t = \sum_{k=1}^{n} w_k f(t_k) + E_n(f), \tag{3.35}$$

where w is a positive (or at least nonnegative) "weight function," assumed integrable over (a,b). The interval (a,b) may now be finite or infinite. If it is infinite, we must make sure that the integral in (3.35) is well defined, at least when f is a polynomial. We achieve this by requiring that all moments of the weight function,

$$\mu_s = \int_a^b t^s w(t)\mathrm{d}t, \quad s = 0, 1, 2, \ldots, \tag{3.36}$$

exist and be finite.

We say that the quadrature formula (3.35) has (polynomial) *degree of exactness* d if

$$E_n(f) = 0 \;\text{ for all }\; f \in \mathbb{P}_d; \tag{3.37}$$

that is, the formula has zero error whenever f is a polynomial of degree $\leq d$. We call (3.35) *interpolatory*, if it has degree of exactness $d = n - 1$. Interpolatory formulae are precisely those "obtained by interpolation," that is, for which

$$\sum_{k=1}^{n} w_k f(t_k) = \int_a^b p_{n-1}(f; t_1, \ldots, t_n; t)w(t)\mathrm{d}t, \tag{3.38}$$

or, equivalently,

$$w_k = \int_a^b \ell_k(t)w(t)\mathrm{d}t, \quad k = 1, 2, \ldots, n, \tag{3.39}$$

where

$$\ell_k(t) = \prod_{\substack{\ell=1 \\ \ell \neq k}}^{n} \frac{t - t_\ell}{t_k - t_\ell} \tag{3.40}$$

are the elementary Lagrange interpolation polynomials associated with the nodes $t_1, t_2, \ldots, t_n$. The fact that (3.35) with w_k given by (3.39) has degree of exactness $d = n - 1$ is evident, since for any $f \in \mathbb{P}_{n-1}$ we have $p_{n-1}(f; \cdot) \equiv f(\cdot)$ in (3.38). Conversely, if (3.35) has degree of exactness $d = n-1$, then putting $f(t) = \ell_r(t)$ in (3.35) gives $\int_a^b \ell_r(t)w(t)\mathrm{d}t = \sum_{k=1}^{n} w_k \ell_r(t_k) = w_r$, $r = 1, 2, \ldots, n$, that is, (3.39).

We see, therefore, that given any n distinct nodes $t_1, t_2, \ldots, t_n$, it is always possible to construct a formula of type (3.35), which is exact for all polynomials of degree $\leq n-1$. In the case $w(t) \equiv 1$ on [–1,1], and t_k equally spaced on [–1,1], the feasibility of such a construction was already alluded to by Newton in 1687 and implemented in detail by Cotes[2] around 1712. By extension, we call the formula (3.35), with the t_k prescribed and the w_k given by (3.39), a *Newton–Cotes formula*.

The question naturally arises whether we can do better, that is, whether we can achieve $d > n-1$ by a judicious choice of the nodes t_k (the weights w_k being necessarily given by (3.39)). The answer is surprisingly simple and direct. To formulate it, we introduce the node polynomial

$$\omega_n(t) = \prod_{k=1}^{n} (t - t_k). \tag{3.41}$$

Theorem 3.2.1. *Given an integer k with $0 \leq k \leq n$, the quadrature formula (3.35) has degree of exactness $d = n-1+k$ if and only if both of the following conditions are satisfied.*

(a) *The formula (3.35) is interpolatory.*
(b) *The node polynomial ω_n in (3.41) satisfies $\int_a^b \omega_n(t)p(t)w(t)\mathrm{d}t = 0$ for all $p \in \mathbb{P}_{k-1}$.*

The condition in (b) imposes k conditions on the nodes $t_1, t_2, \ldots, t_n$ of (3.35). (If $k = 0$, there is no restriction since, as we know, we can always get $d = n-1$.) In effect, ω_n must be orthogonal to $\mathbb{P}_{k-1}$ relative to the weight function w. Since $w(t) \geq 0$, we have necessarily $k \leq n$; otherwise, ω_n would have to be orthogonal to $\mathbb{P}_n$, in particular, orthogonal to itself, which is impossible. Thus, $k = n$ is optimal, giving rise to a quadrature rule of maximum degree of exactness $d_{\max} = 2n-1$. Condition (b) then amounts to orthogonality of ω_n to all polynomials of lower degree; that is, $\omega_n(\,\cdot\,) = \pi_n(\,\cdot\,; w)$ is precisely the nth-degree orthogonal polynomial belonging to the weight function w (cf. Chap. 2, Sect. 2.1.4(2)). This optimal formula is called the *Gaussian quadrature formula* associated with the weight function w. Its nodes, therefore, are the zeros of $\pi_n(\,\cdot\,; w)$, and the weights w_k are given as in (3.39); thus,

$$\left.\begin{aligned} &\pi_n(t_k; w) = 0, \\ &w_k = \int_a^b \frac{\pi_n(t; w)}{(t-t_k)\pi_n'(t_k; w)}\, w(t)\mathrm{d}t, \end{aligned}\right\} \quad k = 1, 2, \ldots, n. \tag{3.42}$$

[2] Roger Cotes (1682–1716), precocious son of an English country pastor, was entrusted with the preparation of the second edition of Newton's *Principia*. He worked out in detail Newton's idea of numerical integration and published the coefficients – now known as Cotes numbers – of the n-point formula for all $n < 11$. Upon his death at the early age of 33, Newton said of him: "If he had lived, we might have known something."

The formula was developed in 1814 by Gauss[3] for the special case $w(t) \equiv 1$ on [–1,1], and extended to more general weight functions by Christoffel[4] in 1877. It is, therefore, also referred to as the *Gauss–Christoffel quadrature formula.*

Proof of Theorem 3.2.1. We first prove the *necessity* of (a) and (b). Since, by assumption, the degree of exactness is $d = n - 1 + k \geq n - 1$, condition (a) is trivial. Condition (b) also follows immediately, since, for any $p \in \mathbb{P}_{k-1}$, the product $\omega_n p$ is in $\mathbb{P}_{n-1+k}$; hence,

$$\int_a^b \omega_n(t)p(t)w(t)\mathrm{d}t = \sum_{k=1}^n w_k \omega_n(t_k)p(t_k),$$

which vanishes, since $\omega_n(t_k) = 0$ for $k = 1, 2, \ldots, n$.

To prove the *sufficiency* of (a), (b), we must show that for any $p \in \mathbb{P}_{n-1+k}$ we have $E_n(p) = 0$ in (3.35). Given any such p, divide it by ω_n, so that

$$p = q\omega_n + r, \quad q \in \mathbb{P}_{k-1}, \quad r \in \mathbb{P}_{n-1},$$

where q is the quotient and r the remainder. There follows

$$\int_a^b p(t)w(t)\mathrm{d}t = \int_a^b q(t)\omega_n(t)w(t)\mathrm{d}t + \int_a^b r(t)w(t)\mathrm{d}t.$$

The first integral on the right, by (b), is zero, since $q \in \mathbb{P}_{k-1}$, whereas the second, by (a), since $r \in \mathbb{P}_{n-1}$, equals

$$\sum_{k=1}^n w_k r(t_k) = \sum_{k=1}^n w_k[p(t_k) - q(t_k)\omega_n(t_k)] = \sum_{k=1}^n w_k p(t_k),$$

the last equality following again from $\omega_n(t_k) = 0$, $k = 1, 2, \ldots, n$. This completes the proof. □

[3] Carl Friedrich Gauss (1777–1855) was one of the greatest mathematicians of the 19th century – and perhaps of all time. He spent almost his entire life in Göttingen, where he was director of the observatory for some 40 years. Already as a student in Göttingen, Gauss discovered that the regular 17-gon can be constructed by compass and ruler, thereby settling a problem that had been open since antiquity. His dissertation gave the first proof of the Fundamental Theorem of Algebra (that an algebraic equation of degree n has exactly n roots). He went on to make fundamental contributions to number theory, differential and non-Euclidean geometry, elliptic and hypergeometric functions, celestial mechanics, geodesy, and various branches of physics, notably magnetism and optics. His computational efforts in celestial mechanics and geodesy, based on the principle of least squares, required the solution (by hand) of large systems of linear equations, for which he used what today are known as Gauss elimination and relaxation methods. Gauss's work on quadrature builds upon the earlier work of Newton and Cotes.

[4] Elvin Bruno Christoffel (1829–1900) was active for short periods of time in Berlin and Zurich and, for the rest of his life, in Strasbourg. He is best known for his work in geometry, in particular, tensor analysis, which became important in Einstein's theory of relativity.

The case $k = n$ of Theorem 3.2.1 (i.e., the Gauss quadrature rule) is discussed further in Sect. 3.2.3. Here, we still mention two special cases with $k < n$, which are of some practical interest. The first is the *Gauss–Radau*[5] quadrature formula in which one endpoint, say, a, is finite and serves as a quadrature node, say, $t_1 = a$. The maximum degree of exactness attainable then is $d = 2n-2$ and corresponds to $k = n-1$ in Theorem 3.2.1. Part (b) of that theorem tells us that the remaining nodes $t_2, \ldots, t_n$ must be the zeros of $\pi_{n-1}(\,\cdot\,; w_a)$, where $w_a(t) = (t-a)w(t)$. Similarly, in the *Gauss–Lobatto*[6] formula, both endpoints are finite and serve as nodes, say, $t_1 = a$, $t_n = b$, and the remaining nodes $t_2, \ldots, t_{n-1}$ are taken to be the zeros of $\pi_{n-2}(\,\cdot\,; w_{a,b})$, $w_{a,b}(t) = (t-a)(b-t)w(t)$, thus achieving maximum degree of exactness $d = 2n-3$.

Example: Two-point Newton–Cotes vs. two-point Gauss

We compare the Newton–Cotes with the Gauss formula in the case $n = 2$ and for the weight function $w(t) = t^{-1/2}$ on [0,1]. The two prescribed nodes in the Newton–Cotes formula are taken to be the endpoints; thus,

$$\int_0^1 t^{-1/2} f(t)\mathrm{d}t \approx \begin{cases} w_1^{\mathrm{NC}} f(0) + w_2^{\mathrm{NC}} f(1) \text{ (Newton-Cotes)}, \\[2ex] w_1^{\mathrm{G}} f(t_1) + w_2^{\mathrm{G}} f(t_2) \quad \text{(Gauss)}. \end{cases}$$

To get the coefficients in the Newton–Cotes formula, we use (3.39), where

$$\ell_1(t) = \frac{t-1}{0-1} = 1-t, \;\; \ell_2(t) = \frac{t-0}{1-0} = t.$$

This gives

$$w_1^{\mathrm{NC}} = \int_0^1 t^{-1/2}\ell_1(t)\mathrm{d}t = \int_0^1 (t^{-1/2} - t^{1/2})\mathrm{d}t = \left(2t^{1/2} - \frac{2}{3}t^{3/2}\right)\Bigg|_0^1 = \frac{4}{3},$$
$$w_2^{\mathrm{NC}} = \int_0^1 t^{-1/2}\ell_2(t)\mathrm{d}t = \int_0^1 t^{1/2}\mathrm{d}t = \frac{2}{3}t^{3/2}\Bigg|_0^1 = \frac{2}{3}.$$

[5] Jean-Charles-Rodolphe Radau (1835–1911) was born in Germany but spent most of his life in France. He was strongly attracted to classical music (the French composer Jaques Offenbach was one of his acquaintances) as he was to celestial mechanics. A gifted writer, he composed many popular articles on topics of scientific interest. He was a person working quietly by himself and staying away from the spotlight.

[6] Rehuel Lobatto (1797–1866), a Dutch mathematician of Portuguese ancestry, although very gifted in his youth, stopped short of attaining an academic degree at the University of Amsterdam. He had to wait, and be satisfied with a low-level government position, until 1842 when he was appointed a professor of mathematics at the Technical University of Delft. His mathematical work is relatively unknown, but he has written several textbooks, one of which, on calculus, published in 1851, contains the quadrature rule now named after him.

Thus,

$$\int_0^1 t^{-1/2} f(t)\mathrm{d}t = \frac{2}{3}\,(2f(0) + f(1)) + E_2^{\mathrm{NC}}(f). \tag{3.43}$$

Note how the square root singularity at the origin causes the value $f(0)$ to receive a weight twice as large as that of $f(1)$.

To develop the Gauss formula, we first construct the required orthogonal polynomial

$$\pi_2(t) = t^2 - p_1 t + p_2.$$

Since it is orthogonal to the constant 1, and to t, we get

$$\begin{aligned}
0 &= \int_0^1 t^{-1/2}\pi_2(t)\mathrm{d}t = \int_0^1 \left(t^{3/2} - p_1 t^{1/2} + p_2 t^{-1/2}\right)\mathrm{d}t \\
&= \left(\frac{2}{5}\,t^{5/2} - \frac{2}{3}\,p_1 t^{3/2} + 2p_2 t^{1/2}\right)\Bigg|_0^1 = \frac{2}{5} - \frac{2}{3}\,p_1 + 2p_2, \\
0 &= \int_0^1 t^{-1/2}\cdot t\pi_2(t)\mathrm{d}t = \int_0^1 \left(t^{5/2} - p_1 t^{3/2} + p_2 t^{1/2}\right)\mathrm{d}t \\
&= \left(\frac{2}{7}\,t^{7/2} - \frac{2}{5}\,p_1 t^{5/2} + \frac{2}{3}\,p_2 t^{3/2}\right)\Bigg|_0^1 = \frac{2}{7} - \frac{2}{5}\,p_1 + \frac{2}{3}\,p_2,
\end{aligned}$$

that is, the linear system

$$\begin{aligned}
\frac{1}{3}p_1 - p_2 &= \frac{1}{5}, \\
\frac{1}{5}p_1 - \frac{1}{3}\,p_2 &= \frac{1}{7}.
\end{aligned}$$

The solution is $p_1 = \frac{6}{7}$, $p_2 = \frac{3}{35}$; thus,

$$\pi_2(t) = t^2 - \frac{6}{7}\,t + \frac{3}{35}\,.$$

The Gauss nodes – the zeros of π_2 – are therefore, to ten decimal places,

$$t_1 = \frac{1}{7}\left(3 - 2\sqrt{\frac{6}{5}}\right) = 0.1155871100,\;\; t_2 = \frac{1}{7}\left(3 + 2\sqrt{\frac{6}{5}}\right) = 0.7415557471.$$

For the weights w_1^{G}, w_2^{G}, we could use again (3.39), but it is simpler to set up a linear system of equations, which expresses the fact that the formula is exact for $f(t) \equiv 1$ and $f(t) \equiv t$:

$$\begin{aligned} w_1^{\mathrm{G}} + w_2^{\mathrm{G}} &= \int_0^1 t^{-1/2} \mathrm{d}t = 2, \\ t_1 w_1^{\mathrm{G}} + t_2 w_2^{\mathrm{G}} &= \int_0^1 t^{-1/2} \cdot t \mathrm{d}t = \frac{2}{3}. \end{aligned}$$

This yields

$$w_1^{\mathrm{G}} = \frac{-2t_2 + \frac{2}{3}}{t_1 - t_2}, \quad w_2^{\mathrm{G}} = \frac{2t_1 - \frac{2}{3}}{t_1 - t_2},$$

or, with the values of t_1, t_2 substituted from the preceding,

$$w_1^{\mathrm{G}} = 1 + \frac{1}{3}\sqrt{\frac{5}{6}} = 1.3042903097, \quad w_2^{\mathrm{G}} = 1 - \frac{1}{3}\sqrt{\frac{5}{6}} = 0.6957096903.$$

Again, w_1^{G} is larger than w_2^{G}, this time by a factor 1.87476….

Summarizing, we obtain the Gauss formula

$$\begin{aligned} \int_0^1 t^{-1/2} f(t) \mathrm{d}t = \left(1 + \frac{1}{3}\sqrt{\frac{5}{6}}\right) f\left(\frac{1}{7}\left(3 - 2\sqrt{\frac{6}{5}}\right)\right) + \left(1 - \frac{1}{3}\sqrt{\frac{5}{6}}\right) \\ \cdot f\left(\frac{1}{7}\left(3 + 2\sqrt{\frac{6}{5}}\right)\right) + E_2^{\mathrm{G}}(f). \end{aligned} \tag{3.44}$$

To illustrate, consider $f(t) = \cos\left(\frac{1}{2}\pi t\right)$; that is,

$$I = \int_0^1 t^{-1/2} \cos\left(\frac{1}{2}\pi t\right) \mathrm{d}t = 2C(1) = 1.5597868008\ldots.$$

$\big(C(x)$ is the Fresnel integral $\int_0^x \cos\left(\frac{1}{2}\pi t^2\right)\mathrm{d}t.\big)$ Newton–Cotes and Gauss give the following approximations,

$$\begin{aligned} I^{\mathrm{NC}} &= \frac{4}{3} = 1.3333\ldots, \\ I^{\mathrm{G}} &= 1.2828510665 + 0.2747384931 = 1.5575895596. \end{aligned}$$

The respective errors are

$$E_2^{\mathrm{NC}} = 0.226453\ldots, \quad E_2^{\mathrm{G}} = 0.002197\ldots,$$

demonstrating the superiority of the Gauss formula even for $n = 2$.

3.2.3 Properties of Gaussian Quadrature Rules

The Gaussian quadrature rule (3.35) and (3.42), in addition to being optimal, has some interesting and useful properties. The more important ones are now listed, with most of the proofs relegated to the exercises.

(a) All nodes t_k are real, distinct, and contained in the open interval (a,b). This is a well-known property satisfied by the zeros of orthogonal polynomials (cf. Ex. 32).

(b) All weights w_k are positive. The formula (3.39) for the weights gives no clue as to their signs; however, an ingenious observation of Stieltjes[7] proves it almost immediately. Indeed,

$$0 < \int_a^b \ell_j^2(t)w(t)\mathrm{d}t = \sum_{k=1}^n w_k \ell_j^2(t_k) = w_j, \quad j = 1,2,\dots,n,$$

the first equality following since $\ell_j^2 \in \mathbb{P}_{2n-2}$ and the degree of exactness is $d = 2n-1$.

(c) If $[a,b]$ is a finite interval, then the Gauss formula converges for any continuous function; that is, $E_n(f) \to 0$ as $n \to \infty$ whenever $f \in C[a,b]$. This is basically a consequence of the Weierstrass Approximation Theorem, which implies that, if $\hat{p}_{2n-1}(f;\cdot)$ denotes the polynomial of degree $2n-1$ that approximates f best on $[a,b]$ in the uniform norm, then

$$\lim_{n\to\infty} \|f(\cdot) - \hat{p}_{2n-1}(f;\cdot)\|_\infty = 0.$$

Since $E_n(\hat{p}_{2n-1}) = 0$ (polynomial degree of exactness $d = 2n-1$), it follows that

$$\begin{aligned}|E_n(f)| &= |E_n(f - \hat{p}_{2n-1})| \\ &= \left| \int_a^b [f(t) - \hat{p}_{2n-1}(f;t)]w(t)\mathrm{d}t - \sum_{k=1}^n w_k[f(t_k) - \hat{p}_{2n-1}(f;t_k)] \right|\end{aligned}$$

[7] Thomas Jan Stieltjes (1856–1894), born in the Netherlands, studied at the Technical Institute of Delft, but never finished to get his degree because of a deep-seated aversion to examinations. He nevertheless got a job at the Observatory of Leiden as a "computer assistant for astronomical calculations." His early publications caught the attention of Hermite, who was able to eventually secure a university position for Stieltjes in Toulouse. A life-long friendship evolved between these two great men, of which two volumes of their correspondence (see Hermite and Stieltjes [1905]) gives vivid testimony (and still makes fascinating reading). Stieltjes is best known for his work on continued fractions and the moment problem, which, among other things, led him to invent a new concept of integral, which now bears his name. He died of tuberculosis at the young age of 38.

$$\begin{aligned}&\le \int_a^b |f(t) - \hat{p}_{2n-1}(f;t)|w(t)\mathrm{d}t + \sum_{k=1}^n w_k|f(t_k) - \hat{p}_{2n-1}(f;t_k)|\\ &\le \|f(\cdot) - \hat{p}_{2n-1}(f;\cdot)\|_\infty \left[\int_a^b w(t)\mathrm{d}t + \sum_{k=1}^n w_k\right].\end{aligned}$$

Here, the positivity of the weight w_k has been used crucially. Noting that

$$\sum_{k=1}^n w_k = \int_a^b w(t)\mathrm{d}t = \mu_0 \quad \text{(cf. (3.36))},$$

we thus conclude

$$|E_n(f)| \le 2\mu_0\|f - \hat{p}_{2n-1}\|_\infty \to 0 \text{ as } n \to \infty.$$

(d) The nodes in the n-point Gauss formula separate those of the $(n+1)$-point formula. (Cf. Ex. 33).

The next property forms the basis of an efficient algorithm for computing Gaussian quadrature formulae.

(e) Let $\alpha_k = \alpha_k(w)$, $\beta_k = \beta_k(w)$ be the recurrence coefficients for the orthogonal polynomials $\pi_k(\,\cdot\,) = \pi_k(\,\cdot\,; w)$; that is (cf. Chap. 2, Sect. 2.1.4(2)),

$$\begin{aligned}\pi_{k+1}(t) = (t - \alpha_k)\pi_k(t) - \beta_k\pi_{k-1}(t),\ \ k = 0, 1, 2\ldots,\\ \pi_0(t) = 1,\ \ \pi_{-1}(t) = 0,\end{aligned} \tag{3.45}$$

with β_0 (as is customary) defined by $\beta_0 = \int_a^b w(t)\mathrm{d}t$ (= μ_0). The nth-order *Jacobi matrix* for the weight function w is a tridiagonal symmetric matrix defined by

$$\boldsymbol{J}_n = \boldsymbol{J}_n(w) = \begin{bmatrix} \alpha_0 & \sqrt{\beta_1} & & & 0 \\ \sqrt{\beta_1} & \alpha_1 & \sqrt{\beta_2} & & \\ & \sqrt{\beta_2} & & \ddots & \\ & & \ddots & \ddots & \sqrt{\beta_{n-1}} \\ 0 & & & \sqrt{\beta_{n-1}} & \alpha_{n-1} \end{bmatrix}.$$

Then the nodes t_k are the eigenvalues of $\boldsymbol{J}_n$ (cf. Ex. 44(a)),

$$\boldsymbol{J}_n\boldsymbol{v}_k = t_k\boldsymbol{v}_k,\ \ \boldsymbol{v}_k^{\mathrm{T}}\boldsymbol{v}_k = 1,\ \ k = 1, 2, \ldots, n, \tag{3.46}$$

and the weights w_k expressible in terms of the first component $v_{k,1}$ of the corresponding (normalized) eigenvectors $\boldsymbol{v}_k$ by (cf. Ex. 44(b))

$$w_k = \beta_0 v_{k,1}^2, \quad k = 1, 2, \ldots, n. \tag{3.47}$$

Thus, to compute the Gauss formula, we must solve an eigenvalue/eigenvector problem for a symmetric tridiagonal matrix. This is a routine problem in numerical linear algebra, and very efficient methods (the QR algorithm, for example) are known for solving it.

(f) Markov[8] observed in 1885 that the Gauss quadrature formula can also be obtained by Hermite interpolation on the nodes t_k, each counting as a double node, if one requires that after integration all coefficients of the derivative terms should be zero (cf. Ex. 34). An interesting consequence of this new interpretation is the following expression for the remainder term, which follows directly from the error term of Hermite interpolation (cf. Chap. 2, Sect. 2.2.7) and the Mean Value Theorem of integration,

$$E_n(f) = \frac{f^{(2n)}(\xi)}{(2n)!} \int_a^b [\pi_n(t; w)]^2 w(t) \mathrm{d}t, \quad a < \xi < b. \tag{3.48}$$

Here, $\pi_n(\,\cdot\,; w)$ is the orthogonal polynomial, with leading coefficient 1, relative to the weight function w. It is assumed, of course, that $f \in C^{2n}[a, b]$.

We conclude this section with a table of some classical weight functions, their corresponding orthogonal polynomials, and the recursion coefficients α_k, β_k for generating the orthogonal polynomials as in (3.45) (see Table 3.1 on the next page). We also include the standard notations for these polynomials (these usually do not refer to the monic polynomials). Note that the recurrence coefficients α_k are all zero for even weight functions on intervals symmetric with respect to the origin (cf. Chap. 2, Sect. 2.1.4(2)). For Jacobi polynomials, the recursion coefficients are explicitly known, but the formulae are a bit lengthy and are not given here (they can be found, e.g., in Gautschi [2004], Table 1.1).

[8] Andrey Andreyevich Markov (1856–1922), a student of Chebyshev, was active in St. Petersburg. While his early work was in number theory and analysis, he is best known for his work in probability theory, where he studied certain discrete random processes now known as Markov chains.

[9] Carl Gustav Jacob Jacobi (1804–1851) was a contemporary of Gauss and with him one of the most important 19th-century mathematicians in Germany. His name is connected with elliptic functions, partial differential equations of dynamics, calculus of variations, celestial mechanics; functional determinants also bear his name. In his work on celestial mechanics he invented what is now called the Jacobi method for solving linear algebraic system.

[10] Edmond Laguerre (1834–1886) was a French mathematician active in Paris, who made essential contributions to geometry, algebra, and analysis.

Table 3.1 Classical orthogonal polynomials

$w(t)$	$[a,b]$	Orth. Pol.	Notation	α_k	β_k
1	[–1,1]	Legendre	P_n	0	$2\ (k=0)$ $(4-k^{-2})^{-1}(k>0)$
$(1-t^2)^{-\frac{1}{2}}$	[–1,1]	Chebyshev #1	T_n	0	$\pi\ (k=0)$ $\frac{1}{2}\ (k=1)$ $\frac{1}{4}\ (k>1)$
$(1-t^2)^{\frac{1}{2}}$	[–1,1]	Chebyshev #2	U_n	0	$\frac{1}{2}\pi\ (k=0)$ $\frac{1}{4}\ (k>0)$
$(1-t)^\alpha\ (1+t)^\beta$ $\alpha>-1, \beta>-1$	[–1,1]	Jacobi[9]	$P_n^{(\alpha,\beta)}$	known	known
$t^\alpha \mathrm{e}^{-t}, \alpha>-1$	$[0,\infty]$	Laguerre[10]	$L_n^{(\alpha)}$	$2k+\alpha+1$	$\Gamma(1+\alpha)\ (k=0)$ $k(k+\alpha)\ (k>0)$
e^{-t^2}	$[-\infty,\infty]$	Hermite	H_n	0	$\sqrt{\pi}\ (k=0)$ $\frac{1}{2}k\ (k>0)$

3.2.4 Some Applications of the Gauss Quadrature Rule

In many applications, the integrals to be computed have the weight function w already built in. In others, one has to figure out for oneself what the most appropriate weight function should be. Several examples of this are given in this section. We begin, however, with the easy exercise of transforming the Gauss–Jacobi quadrature rule from an *arbitrary* finite interval to the canonical interval [–1,1].

(a) *The Gauss–Jacobi formula for the interval* $[a,b]$. We assume $[a,b]$ a finite interval. What is essential about the weight function in the Jacobi case is the fact that it has an algebraic singularity (with exponent α) at the right endpoint, and an algebraic singularity (with exponent β) at the left endpoint. The integral in question, therefore, is

$$\int_a^b (b-x)^\alpha (x-a)^\beta g(x)\mathrm{d}x.$$

A linear transformation of variables

$$x=\frac{1}{2}(b-a)t+\frac{1}{2}(b+a), \quad \mathrm{d}x=\frac{1}{2}(b-a)\mathrm{d}t,$$

maps the x-interval $[a,b]$ onto the t-interval $[-1,1]$, and the integral becomes

$$\int_{-1}^{1}\left(\frac{1}{2}(b-a)(1-t)\right)^{\alpha}\left(\frac{1}{2}(b-a)(1+t)\right)^{\beta}$$
$$\times\, g\left(\frac{1}{2}(b-a)t+\frac{1}{2}(b+a)\right)\frac{1}{2}(b-a)\mathrm{d}t$$
$$=\left(\frac{1}{2}(b-a)\right)^{\alpha+\beta+1}\int_{-1}^{1}(1-t)^{\alpha}(1+t)^{\beta}f(t)\mathrm{d}t,$$

where we have set

$$f(t)=g\left(\frac{1}{2}(b-a)t+\frac{1}{2}(b+a)\right),\quad -1\le t\le 1.$$

The last integral is now in standard form for the application of the Gauss–Jacobi quadrature formula. One thus finds

$$\int_{a}^{b}(b-x)^{\alpha}(x-a)^{\beta}g(x)\mathrm{d}x$$
$$=\left(\frac{1}{2}(b-a)\right)^{\alpha+\beta+1}\sum_{k=1}^{n}w_k^{\mathrm{J}}g\left(\frac{1}{2}(b-a)t_k^{\mathrm{J}}+\frac{1}{2}(b+a)\right)+E_n(g), \tag{3.49}$$

where t_k^{J}, w_k^{J} are the (standard) Gauss–Jacobi nodes and weights. Since with g, also f is a polynomial, and both have the same degree, the formula (3.49) is exact for all $g \in \mathbb{P}_{2n-1}$.

(b) *Iterated integrals*. Let I denote the integral operator

$$(Ig)(t) := \int_{0}^{t} g(\tau)\mathrm{d}\tau,$$

and I^p its pth power (the identity operator if $p=0$). Then

$$(I^{p+1}g)(1)=\int_{0}^{1}(I^p g)(t)\mathrm{d}t$$

is the pth *iterated integral* of g. It is a well-known fact from calculus that an iterated integral can be written as a simple integral, namely,

$$(I^{p+1}g)(1)=\int_{0}^{1}\frac{(1-t)^p}{p!}g(t)\mathrm{d}t. \tag{3.50}$$

Thus, to the integral on the right of (3.50) we could apply any of the standard integration procedures, such as the composite trapezoidal or Simpson's rule. If g is smooth, this works well for p relatively small. As p gets larger, the factor $(1-t)^p$ becomes rather unpleasant, since as $p \to \infty$ it approaches the discontinuous function equal to 1 at $t = 0$, and 0 elsewhere on (0,1]. This adversely affects the performance of any standard quadrature scheme. However, noting that $(1-t)^p$ is a Jacobi weight on the interval [0,1], with parameters $\alpha = p$, $\beta = 0$, we can apply (3.49); that is,

$$(I^{p+1}g)(1) = \frac{1}{2^{p+1}p!} \sum_{k=1}^{n} w_k^{\mathrm{J}} g\left(\frac{1}{2}(1+t_k^{\mathrm{J}})\right) + E_n(g), \tag{3.51}$$

and get accurate results with moderate values of n, even when p is quite large (cf. MA 5).

(c) *Integration over* $\mathbb{R}$. Compute $\int_{-\infty}^{\infty} F(x)\mathrm{d}x$, assuming that, for some $a > 0$,

$$F(x) \sim \mathrm{e}^{-ax^2}, \quad x \to \pm\infty. \tag{3.52}$$

Instead of a weight function we are given here information about the asymptotic behavior of the integrand. It is natural, then, to introduce the new function

$$f(t) = \mathrm{e}^{t^2} F\left(\frac{t}{\sqrt{a}}\right), \tag{3.53}$$

which tends to 1 as $t \to \pm\infty$. The change of variables $x = t/\sqrt{a}$ then gives

$$\int_{-\infty}^{\infty} F(x)\mathrm{d}x = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} F\left(\frac{t}{\sqrt{a}}\right)\mathrm{d}t = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} \mathrm{e}^{-t^2} f(t)\mathrm{d}t.$$

The last integral is now in a form suitable for the application of the Gauss–Hermite formula. There results

$$\int_{-\infty}^{\infty} F(x)\mathrm{d}x = \frac{1}{\sqrt{a}} \sum_{k=1}^{n} w_k^{\mathrm{H}} \mathrm{e}^{(t_k^{\mathrm{H}})^2} F\left(\frac{t_k^{\mathrm{H}}}{\sqrt{a}}\right) + E_n(f), \tag{3.54}$$

where t_k^{H}, w_k^{H} are the Gauss–Hermite nodes and weights. The remainder $E_n(f)$ vanishes whenever f is a polynomial of degree $\leq 2n-1$; that is,

$$F(x) = \mathrm{e}^{-ax^2} p(x), \quad p \in \mathbb{P}_{2n-1}.$$

Since the coefficients in (3.54) involve the products $w_k^{\mathrm{H}} \cdot \exp((t_k^{\mathrm{H}})^2)$, some tables of Gauss–Hermite formulae also provide these products, in addition to the nodes and weights.

(d) *Integration over* $\mathbb{R}_+$. Compute $\int_0^{\infty} F(x)\mathrm{d}x$, assuming that

$$F(x) \sim \begin{cases} x^p & \text{as } x \downarrow 0 \quad (p > -1), \\ x^{-q} & \text{as } x \to \infty \ (q > 1). \end{cases} \tag{3.55}$$

Similarly as in (c), we now define f by

$$F(x) = \frac{x^p}{(1+x)^{p+q}}\, f(x), \quad x \in \mathbb{R}_+, \tag{3.56}$$

so as to again have $f(x) \to 1$ as $x \downarrow 0$ and $x \to \infty$. The change of variables

$$x = \frac{1+t}{1-t}, \quad \mathrm{d}x = \frac{2\mathrm{d}t}{(1-t)^2}$$

then yields

$$\begin{aligned}\int_0^\infty F(x)\mathrm{d}x &= \int_{-1}^1 \left(\frac{1+t}{1-t}\right)^p \left(\frac{2}{1-t}\right)^{-(p+q)} f\left(\frac{1+t}{1-t}\right) \frac{2\mathrm{d}t}{(1-t)^2} \\ &= \frac{1}{2^{p+q-1}} \int_{-1}^1 (1-t)^{q-2}(1+t)^p g(t)\mathrm{d}t,\end{aligned}$$

where

$$g(t) = f\left(\frac{1+t}{1-t}\right).$$

This calls for Gauss–Jacobi quadrature with parameters $\alpha = q-2$, $\beta = p$:

$$\int_0^\infty F(x)\mathrm{d}x = \frac{1}{2^{p+q-1}}\left(\sum_{k=1}^n w_k^{\mathrm{J}} g(t_k^{\mathrm{J}}) + E_n(g)\right).$$

It remains to re-express g in terms of f, and f in terms of F, to obtain the final formula

$$\begin{aligned}\int_0^\infty F(x)\mathrm{d}x =& 2\sum_{k=1}^n w_k^{\mathrm{J}} \left(1+t_k^{\mathrm{J}}\right)^{-p} \left(1-t_k^{\mathrm{J}}\right)^{-q} F\left(\frac{1+t_k^{\mathrm{J}}}{1-t_k^{\mathrm{J}}}\right) \\ &+ 2^{-p-q+1} E_n(g).\end{aligned} \tag{3.57}$$

This is exact whenever $g(t)$ is a polynomial of degree $\le 2n-1$, for example a polynomial of that degree in the variable $1-t$; hence, since $f(x) = g\left(\frac{x-1}{x+1}\right)$, for any $F(x)$ of the form

$$F(x) = \frac{x^p}{(1+x)^{p+q+\lambda}}, \quad \lambda = 0, 1, 2, \ldots, 2n-1.$$

3.2.5 Approximation of Linear Functionals: Method of Interpolation vs. Method of Undetermined Coefficients

Up until now, we heavily relied on interpolation to obtain approximations to derivatives, integrals, and the like. This is not the only possible approach, however, to construct approximation formulae, and indeed not usually the simplest one. Another is the method of undetermined coefficients. Both approaches can be described in a vastly more general setting, in which a given linear functional is to be approximated by other linear functionals so as to have exactness on a suitable finite-dimensional function space. Although both approaches yield the same formulae, the mechanics involved are quite different.

Before stating the general approximation problem, we illustrate the two methods on some simple examples.

Example. Obtain an approximation formula of the type

$$\int_0^1 f(x)\mathrm{d}x \approx a_1 f(0) + a_2 f(1). \tag{3.58}$$

Method of interpolation. Instead of integrating f, we integrate the (linear) polynomial interpolating f at $x = 0$ and $x = 1$. This gives

$$\begin{aligned}\int_0^1 f(x)\mathrm{d}x &\approx \int_0^1 p_1(f;0,1;x)\mathrm{d}x \\ &= \int_0^1 [(1-x)f(0) + xf(1)]\mathrm{d}x = \frac{1}{2}\,[f(0) + f(1)],\end{aligned}$$

the trapezoidal rule, to nobody's surprise.

Method of undetermined coefficients. We simply require (3.58) to be exact for all linear functions. This is the same as requiring equality in (3.58) when $f(x) = 1$ and $f(x) = x$. (Then by linearity, one gets equality also for $f(x) = c_0 + c_1 x$ for arbitrary constants c_0, c_1.) This immediately produces the linear system

$$\begin{aligned}1 &= a_1 + a_2, \\ \frac{1}{2} &= a_2,\end{aligned}$$

hence $a_1 = a_2 = \frac{1}{2}$, as before.

Example. Find a formula of the type

$$\int_0^1 \sqrt{x}\, f(x)\mathrm{d}x \approx a_1 f(0) + a_2 \int_0^1 f(x)\mathrm{d}x. \tag{3.59}$$

Such a formula may come in handy if we already know the integral on the right and want to use this information, together with the value of f at $x = 0$, to approximate the weighted integral on the left.

Method of interpolation. "Interpolation" here means interpolation to the given data – the value of f at $x = 0$ and the integral of f from 0 to 1. We are thus seeking a polynomial $p \in \mathbb{P}_1$ such that

$$p(0) = f(0), \quad \int_0^1 p(x)\mathrm{d}x = \int_0^1 f(x)\mathrm{d}x, \tag{3.60}$$

which we then substitute in place of f in the left-hand integral of (3.59). If we let $p(x) = c_0 + c_1 x$, the interpolation conditions (3.60) give

$$\begin{aligned} c_0 &= f(0), \\ c_0 + \frac{1}{2}c_1 &= \int_0^1 f(x)\mathrm{d}x; \end{aligned}$$

hence

$$c_0 = f(0), \quad c_1 = 2\left\{\int_0^1 f(x)\mathrm{d}x - f(0)\right\}.$$

Therefore,

$$\begin{aligned} \int_0^1 \sqrt{x} f(x)\mathrm{d}x &\approx \int_0^1 \sqrt{x} p(x)\mathrm{d}x = \int_0^1 \sqrt{x}(c_0 + c_1 x)\mathrm{d}x \\ &= \frac{2}{3}c_0 + \frac{2}{5}c_1 = \frac{2}{3}f(0) + \frac{2}{5}\cdot 2\left\{\int_0^1 f(x)\mathrm{d}x - f(0)\right\}; \end{aligned}$$

that is,

$$\int_0^1 \sqrt{x} f(x)\mathrm{d}x \approx -\frac{2}{15}f(0) + \frac{4}{5}\int_0^1 f(x)\mathrm{d}x. \tag{3.61}$$

Method of undetermined coefficients. Equality in (3.59) for $f(x) = 1$ and $f(x) = x$ immediately yields

$$\begin{aligned} \frac{2}{3} &= a_1 + a_2, \\ \frac{2}{5} &= \frac{1}{2}a_2; \end{aligned}$$

hence $a_1 = -\frac{2}{15}$, $a_2 = \frac{4}{5}$, the same result as in (3.61), but produced incomparably faster.

In both examples, we insisted on exactness for polynomials (of degree 1). In place of polynomials, we could have chosen other classes of functions, as long as we make sure that their dimension matches the number of "free parameters."

The essence of the two examples consists of showing how a certain linear functional can be approximated in terms of other (presumably simpler) linear functionals by forming a suitable linear combination of the latter. We recall that a *linear functional* L on a function space $\mathbb{F}$ is a map

$$L : \mathbb{F} \to \mathbb{R}, \tag{3.62}$$

which satisfies the conditions of additivity and homogeneity:

$$L(f+g) = Lf + Lg, \ \text{ all } \ f,\ g \in \mathbb{F}, \tag{3.63}$$

and

$$L(cf) = cLf, \ \text{ all } \ c \in \mathbb{R},\ f \in \mathbb{F}. \tag{3.64}$$

The function class $\mathbb{F}$, of course, must be a *linear space*, that is, closed under addition and multiplication by a scalar: $f, g \in \mathbb{F}$ implies $f + g \in \mathbb{F}$, and $f \in \mathbb{F}$ implies $cf \in \mathbb{F}$ for any $c \in \mathbb{R}$.

Here are some examples of linear functionals and appropriate spaces $\mathbb{F}$ on which they live:

(a) $Lf = f(0)$; $\mathbb{F} = \{f : f \text{ is defined at } x = 0\}$.
(b) $Lf = f''\left(\frac{1}{2}\right)$; $\mathbb{F} = \left\{f : f \text{ has a second derivative at } x = \frac{1}{2}\right\}$.
(c) $Lf = \int_0^1 f(x)\mathrm{d}x$; $\mathbb{F} = C[0,1]$ (or, more generally, f is Riemann integrable, or Lebesgue integrable).
(d) $Lf = \int_0^1 f(x)w(x)\mathrm{d}x$, where w is a given (integrable) "weight function;" $\mathbb{F} = C[0,1]$.
(e) Any linear combination (with constant coefficients) of the preceding linear functionals.

Examples of *nonlinear functionals* are

(a′) $Kf = |f(0)|$.
(b′) $Kf = \displaystyle\int_0^1 [f(x)]^2\mathrm{d}x$, and so on.

We are now ready to formulate the *general approximation problem*: given a linear functional L on $\mathbb{F}$ (to be approximated), n special linear functionals $L_1, L_2, \ldots, L_n$ on $\mathbb{F}$ and their values (the "data") $\ell_i = L_i f$, $i = 1, 2, \ldots, n$, applied to some function f, and given a linear subspace $\Phi \subset \mathbb{F}$ with $\dim \Phi = n$, we want to find an approximation formula of the type

$$Lf \approx \sum_{i=1}^{n} a_i L_i f \tag{3.65}$$

that is exact (i.e., holds with equality) whenever $f \in \Phi$.

It is natural (since we want to "interpolate") to make the following.
Assumption: the "interpolation problem"

$$\text{find } \varphi \in \Phi \text{ such that } L_i\varphi = s_i, \ \ i = 1, 2, \dots, n, \tag{3.66}$$

has a unique solution, $\varphi(\cdot) = \varphi(\boldsymbol{s}; \cdot)$, for arbitrary $\boldsymbol{s} = [s_1, s_2, \dots, s_n]^{\mathrm{T}}$.

We can express our assumption more explicitly in terms of a given *basis* $\varphi_1, \varphi_2, \dots, \varphi_n$ of Φ and the associated "Gram matrix"

$$\boldsymbol{G} = [L_i\varphi_j] = \begin{bmatrix} L_1\varphi_1 & L_1\varphi_2 & \cdots & L_1\varphi_n \\ L_2\varphi_1 & L_2\varphi_2 & \cdots & L_2\varphi_n \\ \cdots & \cdots & \cdots & \cdots \\ L_n\varphi_1 & L_n\varphi_2 & \cdots & L_n\varphi_n \end{bmatrix} \in \mathbb{R}^{n\times n}. \tag{3.67}$$

What we require is that

$$\det \boldsymbol{G} \neq 0. \tag{3.68}$$

It is easily seen (cf. Ex. 49) that this condition is independent of the particular choice of basis. To show that unique solvability of (3.66) and (3.68) are equivalent, we express φ in (3.66) as a linear combination of the basis functions,

$$\varphi = \sum_{j=1}^{n} c_j \varphi_j \tag{3.69}$$

and note that the interpolation conditions

$$L_i\left(\sum_{j=1}^{n} c_j\varphi_j\right) = s_i, \ \ i = 1, 2, \dots, n,$$

by the linearity of the functionals L_i, can be written in the form

$$\sum_{j=1}^{n} c_j L_i \varphi_j = s_i, \ \ i = 1, 2, \dots, n;$$

that is,

$$\boldsymbol{G}\boldsymbol{c} = \boldsymbol{s}, \ \ \boldsymbol{c} = [c_1, c_2, \dots, c_n]^{\mathrm{T}}, \ \ \boldsymbol{s} = [s_1, s_2, \dots, s_n]^{\mathrm{T}}. \tag{3.70}$$

This has a unique solution for arbitrary $\boldsymbol{s}$ if and only if (3.68) holds.

Method of interpolation. We solve the general approximation problem "by interpolation,"

$$Lf \approx L\varphi(\boldsymbol{\ell}; \cdot), \ \ \boldsymbol{\ell} = [\ell_1, \ell_2, \dots, \ell_n]^{\mathrm{T}}, \ \ \ell_i = L_i f. \tag{3.71}$$

In other words, we apply L not to f, but to $\varphi(\boldsymbol{\ell};\,\cdot\,)$ – the solution of the interpolation problem (3.66) in which $\boldsymbol{s} = \boldsymbol{\ell}$, the given "data." Our assumption guarantees that $\varphi(\boldsymbol{\ell};\,\cdot\,)$ is uniquely determined. In particular, if $f \in \Phi$, then (3.71) holds with equality, since trivially $\varphi(\boldsymbol{\ell};\,\cdot\,) \equiv f(\,\cdot\,)$ in this case. Thus, our approximation (3.71) already satisfies the exactness condition required for (3.65). It remains only to show that (3.71) indeed produces an approximation of the form (3.65). To do so, observe that the interpolant in (3.71) is

$$\varphi(\boldsymbol{\ell};\,\cdot\,) = \sum_{j=1}^{n} c_j \varphi_j(\,\cdot\,),$$

where the vector $\boldsymbol{c} = [c_1, c_2, \dots, c_n]^{\mathrm{T}}$ satisfies (3.70) with $\boldsymbol{s} = \boldsymbol{\ell}$,

$$\boldsymbol{G}\boldsymbol{c} = \boldsymbol{\ell}, \quad \boldsymbol{\ell} = [L_1 f, L_2 f, \dots, L_n f]^{\mathrm{T}}.$$

Writing

$$\lambda_j = L\varphi_j, \quad j = 1, 2, \dots, n; \quad \boldsymbol{\lambda} = [\lambda_1, \lambda_2, \dots, \lambda_n]^{\mathrm{T}}, \tag{3.72}$$

we have by the linearity of L,

$$L\varphi(\boldsymbol{\ell};\,\cdot\,) = \sum_{j=1}^{n} c_j L\varphi_j = \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{c} = \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{G}^{-1}\boldsymbol{\ell} = \left[(\boldsymbol{G}^{\mathrm{T}})^{-1}\boldsymbol{\lambda}\right]^{\mathrm{T}}\boldsymbol{\ell};$$

that is,

$$L\varphi(\boldsymbol{\ell};\,\cdot\,) = \sum_{i=1}^{n} a_i L_i f, \quad \boldsymbol{a} = [a_1, a_2, \dots, a_n]^{\mathrm{T}} = (\boldsymbol{G}^{\mathrm{T}})^{-1}\boldsymbol{\lambda}. \tag{3.73}$$

Method of undetermined coefficients. Here, we determine the coefficients a_i in (3.65) such that equality holds for all $f \in \Phi$, which, by the linearity of both L and L_i is equivalent to equality for $f = \varphi_1$, $f = \varphi_2, \dots, f = \varphi_n$; that is,

$$\left(\sum_{j=1}^{n} a_j L_j\right)\varphi_i = L\varphi_i, \quad i = 1, 2, \dots, n,$$

or, by (3.72),

$$\sum_{j=1}^{n} a_j L_j \varphi_i = \lambda_i, \quad i = 1, 2, \dots, n.$$

Evidently, the matrix of this system is $\boldsymbol{G}^{\mathrm{T}}$, so that

$$\boldsymbol{a} = [a_1, a_2, \dots, a_n]^{\mathrm{T}} = (\boldsymbol{G}^{\mathrm{T}})^{-1}\boldsymbol{\lambda},$$

in agreement with (3.73). Thus, *the method of interpolation and the method of undetermined coefficients are mathematically equivalent* – they produce exactly the same approximation.

3.2.6 Peano Representation of Linear Functionals

It may be argued that the method of interpolation, at least in the case of polynomials (i.e., $\Phi = \mathbb{P}_d$), is more powerful than the method of undetermined coefficients because it also yields an expression for the error term (if we carry along the remainder term of interpolation). The method of undetermined coefficients, in contrast, generates only the coefficients in the approximation and gives no clue as to the approximation error.

There is, however, a device due to Peano[11] that allows us to discuss the error after the approximation has been found. The point is that the error,

$$Ef = Lf - \sum_{i=1}^{n} a_i L_i f, \tag{3.74}$$

is itself a linear functional, one that annihilates all polynomials, say, of degree d,

$$Ep = 0, \;\text{ all }\; p \in \mathbb{P}_d. \tag{3.75}$$

Now suppose that $\mathbb{F}$ consists of all functions f having a continuous $(r+1)$st derivative on the finite interval $[a,b]$, $\mathbb{F} = C^{r+1}[a,b]$, $r \le d$. Then by Taylor's theorem with the remainder in integral form , we have

$$\begin{aligned} f(x) =& f(a) + (x-a)\,\frac{f'(a)}{1!} + \cdots + (x-a)^r\,\frac{f^{(r)}(a)}{r!} \\ &+ \frac{1}{r!}\int_a^x (x-t)^r f^{(r+1)}(t)\mathrm{d}t. \end{aligned} \tag{3.76}$$

The last integral can be extended to $t = b$ if we replace $x - t$ by 0 when $t > x$:

$$(x-t)_+ = \begin{cases} x-t & \text{if } x-t \ge 0, \\ 0 & \text{if } x-t < 0. \end{cases}$$

[11] Giuseppe Peano (1858–1932), an Italian mathematician active in Turin, made fundamental contributions to mathematical logic, set theory, and the foundations of mathematics. General existence theorems in ordinary differential equations also bear his name. He created his own mathematical language, using symbols of the algebra of logic, and even promoted (and used) a simplified Latin (his "latino") as a world language for scientific publication.

Thus,

$$\int_a^x (x-t)^r f^{(r+1)}(t)\mathrm{d}t = \int_a^b (x-t)_+^r f^{(r+1)}(t)\mathrm{d}t. \tag{3.77}$$

Now by applying the linear functional E to both sides of (3.76), with the integral written as in (3.77), yields by linearity of E and (3.75)

$$\begin{aligned} Ef &= \frac{1}{r!}\, E\left\{\int_a^b (x-t)_+^r f^{(r+1)}(t)\mathrm{d}t\right\} \\ &= \frac{1}{r!}\int_a^b [E_{(x)}(x-t)_+^r] f^{(r+1)}(t)\mathrm{d}t, \end{aligned}$$

provided the interchange of E with the integral is legitimate. (For most functionals it is.) The subscript x in $E_{(x)}$ is to indicate that E acts on the variable x (not t). Defining

$$K_r(t) = \frac{1}{r!}\, E_{(x)}(x-t)_+^r, \quad t \in \mathbb{R}, \tag{3.78}$$

we thus have the following representation for the error E,

$$Ef = \int_a^b K_r(t) f^{(r+1)}(t)\mathrm{d}t. \tag{3.79}$$

This is called the *Peano representation* of the functional E, and K_r the rth *Peano kernel* for E.

If the functional E makes reference only to values of x in $[a,b]$ (e.g., Ef may involve values of f or of a derivative of f at some points in $[a,b]$, or integration over $[a,b]$), then it follows from (3.78) that $K_r(t) = 0$ for $t \notin [a,b]$ (cf. Ex. 50). In this case, the integral in (3.79) can be extended over the whole real line.

The functional E is called *definite of order r* if its Peano kernel K_r is of the same sign. (We then also say that K_r is definite.) For such functionals E, we can use the Mean Value Theorem of integration to write (3.79) in the form

$$Ef = f^{(r+1)}(\tau)\int_a^b K_r(t)\mathrm{d}t, \quad a < \tau < b \quad (E \text{ definite of order } r).$$

The integral on the right is easily evaluated by putting $f(t) = t^{r+1}/(r+1)!$ in (3.79). This gives

$$Ef = e_{r+1} f^{(r+1)}(\tau), \quad e_{r+1} = E\,\frac{t^{r+1}}{(r+1)!} \quad (E \text{ definite of order } r). \tag{3.80}$$

Since $e_{r+1} \neq 0$ by definiteness of K_r, we must have $r = d$ by virtue of (3.75), and so

$$|Ef| \leq |e_{d+1}|\, \|f^{(d+1)}\|_\infty. \tag{3.81}$$

Conversely, a functional E satisfying (3.80) with $e_{r+1} \neq 0$ is necessarily definite of order r (see Ex. 51). For nondefinite functionals E, we must estimate by

$$|Ef| \leq \|f^{(r+1)}\|_\infty \int_a^b |K_r(t)|\mathrm{d}t, \tag{3.82}$$

which, in view of the form (3.78) of K_r, can be rather laborious.

As an example, consider the formula obtained in (3.61); here

$$Ef = \int_0^1 \sqrt{x} f(x)\mathrm{d}x + \frac{2}{15} f(0) - \frac{4}{5}\int_0^1 f(x)\mathrm{d}x,$$

and $Ef = 0$ for all $f \in \mathbb{P}_1$ ($d = 1$). Assuming that $f \in C^2[0,1]$ ($r = d = 1$), we thus have by (3.79),

$$Ef = \int_0^1 K_1(t) f''(t)\mathrm{d}t, \quad K_1(t) = E_{(x)}(x-t)_+.$$

Furthermore,

$$\begin{aligned}
K_1(t) &= \int_0^1 \sqrt{x}(x-t)_+\mathrm{d}x + \frac{2}{15}\cdot 0 - \frac{4}{5}\int_0^1 (x-t)_+\mathrm{d}x \\
&= \int_t^1 \sqrt{x}(x-t)\mathrm{d}x - \frac{4}{5}\int_t^1 (x-t)\mathrm{d}x \\
&= \frac{2}{5}x^{\frac{5}{2}}\Big|_t^1 - t\cdot\frac{2}{3}x^{\frac{3}{2}}\Big|_t^1 - \frac{4}{5}\left(\frac{1}{2}x^2\Big|_t^1 - t(1-t)\right) \\
&= \frac{2}{5}\left(1-t^{\frac{5}{2}}\right) - \frac{2}{3}t\left(1-t^{\frac{3}{2}}\right) - \frac{2}{5}(1-t^2) + \frac{4}{5}t(1-t) \\
&= \frac{4}{15}t^{\frac{5}{2}} - \frac{2}{5}t^2 + \frac{2}{15}t \\
&= \frac{2}{15}t\left(2t^{\frac{3}{2}} - 3t + 1\right).
\end{aligned}$$

Now the function in parentheses, say, $q(t)$, satisfies $q(0) = 1$, $q(1) = 0$, $q'(t) = -3(1-t^{1/2}) < 0$ for $0 < t < 1$. There follows $q(t) \geq 0$ on [0,1], and the kernel K_1 is (positive) definite. Furthermore,

$$\begin{aligned}
e_2 = E\left(\frac{t^2}{2!}\right) &= \int_0^1 \sqrt{x}\,\frac{x^2}{2}\mathrm{d}x + \frac{2}{15}\cdot 0 - \frac{4}{5}\int_0^1 \frac{x^2}{2}\mathrm{d}x \\
&= \frac{1}{2}\,\frac{2}{7}x^{\frac{7}{2}}\Big|_0^1 - \frac{2}{5}\,\frac{1}{3}x^3\Big|_0^1 \\
&= \frac{1}{7} - \frac{2}{15} = \frac{1}{105},
\end{aligned}$$

so that finally, by (3.80),

$$Ef = \frac{1}{105} f''(\tau), \;\; 0 < \tau < 1.$$

3.2.7 Extrapolation Methods

Many methods of approximation depend on a positive parameter, say, h, which controls the accuracy of the method. As $h \downarrow 0$, the approximations typically converge to the exact solution. An example of this is the composite trapezoidal rule of integration, where h is the spacing of the quadrature nodes. Other important examples are finite difference methods in ordinary and partial differential equations. In practice, one usually computes several approximations to a solution, corresponding to different values of the parameter h. It is then natural to try "extrapolating to the limit $h = 0$," that is, constructing a linear combination of these approximations that is more accurate than either of them. This is the basic idea behind extrapolation methods. We apply it here to the composite trapezoidal rule, which gives rise to an interesting and powerful integration technique known as Romberg integration.

To develop the general principle, suppose the approximation in question is $A(h)$, a scalar-valued function of h, and thus $A(0)$ the exact solution. The approximation may be defined only for a set of discrete values of h, which, however, have $h = 0$ as a limit point (e.g., $h = (b-a)/n$, $n = 1, 2, 3, \dots$, in the case of the composite trapezoidal rule). We call these *admissible* values of h. When in the following we write $h \to 0$, we mean that h goes to zero over these admissible values. About the approximation $A(h)$, we first assume that there exist constants a_1, a_2 independent of h, and two positive numbers p, p' with $p' > p$, such that

$$A(h) = a_0 + a_1 h^p + O(h^{p'}), \;\; h \to 0, \; a_1 \neq 0. \tag{3.83}$$

The order term here has the usual meaning of a quantity bounded (for all sufficiently small h) by a constant times $h^{p'}$, where the constant does not depend on h. We only assume the existence of such constants a_0, a_1; their values are usually not known. Indeed, $a_0 = A(0)$, for example, is the exact solution. The value p, on the other hand, is assumed to be known.

Now let $q < 1$ be a fixed positive number, and h and $q^{-1}h$ admissible parameters. Then we have

$$\left.\begin{aligned} A(h) &= a_0 + a_1 h^p + O(h^{p'}), \\ A(q^{-1}h) &= a_0 + a_1 q^{-p} h^p + O(h^{p'}), \end{aligned}\right. \;\; h \to 0.$$

Eliminating the middle terms on the right, we find

$$a_0 = A(0) = A(h) + \frac{A(h) - A(q^{-1}h)}{q^{-p} - 1} + O(h^{p'}), \;\; h \to 0.$$

Thus, from two approximations, $A(h)$, $A(q^{-1}h)$, whose errors are both $O(h^p)$, we obtain an improved approximation,

$$A_{\text{impr}}(h) = A(h) + \frac{A(h) - A(q^{-1}h)}{q^{-p} - 1}, \tag{3.84}$$

with a smaller error of $O(h^{p'})$. The passage from A to A_{impr} is called *Richardson*[12] *extrapolation*.

If we want to repeat this process, we have to know more about the approximation $A(h)$; there must be more terms to be eliminated. This leads to the idea of an asymptotic expansion: we say that $A(h)$ admits an *asymptotic expansion*

$$A(h) = a_0 + a_1 h^{p_1} + a_2 h^{p_2} + \cdots, \quad 0 < p_1 < p_2 < \cdots, \quad h \to 0 \tag{3.85}$$

(with coefficients a_i independent of h), if for each $k = 1, 2, \ldots$

$$A(h) - (a_0 + a_1 h^{p_1} + \cdots + a_k h^{p_k}) = O(h^{p_{k+1}}), \quad h \to 0. \tag{3.86}$$

We emphasize that (3.85) need not (and in fact usually does not) converge for any fixed $h > 0$; all that is required is (3.86). If (3.86) holds only for finitely many k, say, $k = 1, 2, \ldots, K$, then the expansion (3.85) is finite and is referred to as an *asymptotic approximation* to $K + 1$ terms.

It is now clear that if $A(h)$ admits an asymptotic expansion (or approximation), we can successively eliminate the terms of the expansion exactly as we did for a 2-term approximation, thereby obtaining a (finite or infinite) sequence of successively improved approximations. We formulate this in the form of a theorem.

Theorem 3.2.2. (Repeated Richardson extrapolation). *Let $A(h)$ admit the asymptotic expansion (3.85) and define, for some fixed positive $q < 1$,*

$$\begin{aligned} A_1(h) &= A(h), \\ A_{k+1}(h) &= A_k(h) + \frac{A_k(h) - A_k(q^{-1}h)}{q^{-p_k} - 1}, \quad k = 1, 2, \ldots. \end{aligned} \tag{3.87}$$

[12] Lewis Fry Richardson (1881–1953), born, educated, and active in England, did pioneering work in numerical weather prediction, proposing to solve the hydrodynamical and thermodynamical equations of meteorology by finite difference methods. Although this was the precomputer age, Richardson envisaged that the job could be done "with tier upon tier of human computers fitted into an Albert Hall structure" (P.S. Sheppard in *Nature*, vol. 172, 1953, p. 1127). He also did a penetrating study of atmospheric turbulence, where a nondimensional quantity introduced by him is now called "Richardson's number." At the age of 50, he earned a degree in psychology and began to develop a scientific theory of international relations. He was elected a Fellow of the Royal Society in 1926.

$$\begin{array}{llll}
A_{0,0} & & & \\
A_{1,0} & A_{1,1} & & \\
A_{2,0} & A_{2,1} & A_{2,2} & \\
A_{3,0} & A_{3,1} & A_{3,2} & A_{3,3} \\
\dots & \dots & \dots & \dots
\end{array}$$

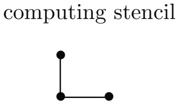


Fig. 3.3 Extrapolation algorithm

Then for each $n = 1, 2, 3, \dots, A_n(h)$ *admits an asymptotic expansion*

$$A_n(h) = a_0 + a_n^{(n)} h^{p_n} + a_{n+1}^{(n)} h^{p_{n+1}} + \cdots, \quad h \to 0, \tag{3.88}$$

with certain coefficients $a_n^{(n)}$, $a_{n+1}^{(n)}, \dots$ *not depending on h.*

We remark that if (3.85) is only an approximation to $K + 1$ terms, then the recursion (3.87) is applicable only for $k = 1, 2, \dots, K$, and (3.88) holds for $n = 1, 2, \dots, K$, whereas for $n = K + 1$ one has $A_{K+1}(h) = a_0 + O(h^{p_{K+1}})$.

It is easily seen from (3.87) that $A_{k+1}(h)$ is a linear combination of $A(h)$, $A(q^{-1}h)$, $A(q^{-2}h), \dots, A(q^{-k}h)$, where it was tacitly assumed that h, $q^{-1}h$, $q^{-2}h, \dots$ are admissible values of the parameter.

We now rework Theorem 3.2.2 into a practical algorithm. To do so, we assume that we initially compute $A(h)$ for a succession of parameter values

$$h_0, \ qh_0, \ q^2h_0, \dots \ (q < 1),$$

all being admissible. Then we define

$$A_{m,k} = A_{k+1}(q^m h_0), \quad m, k = 0, 1, 2, \dots. \tag{3.89}$$

The idea behind (3.89) is to provide two mechanisms for improving the accuracy: one is to increase m, which reduces the parameter h, the other is increasing k, which engages a more accurate approximation. Ideally, one employs both mechanisms simultaneously, which suggests that the diagonal entries $A_{m,m}$ are the ones of most interest.

Putting $h = q^m h_0$ in (3.87) produces the *extrapolation algorithm*

$$\begin{aligned}
A_{m,k} &= A_{m,k-1} + \frac{A_{m,k-1} - A_{m-1,k-1}}{q^{-p_k} - 1}, \quad m \geq k \geq 1, \\
A_{m,0} &= A(q^m h_0).
\end{aligned} \tag{3.90}$$

This allows us to compute the triangular scheme of approximations shown in Fig. 3.3.

Each entry is computed in terms of its neighbors horizontally, and diagonally above, to the left, as indicated in the computing stencil of Fig. 3.3. The entries in the first column are the approximations initially computed. The generation of the triangular scheme, once the first column has been computed, is extremely cheap, yet can dramatically improve the accuracy, especially down the diagonal. If (3.85) is a finite asymptotic approximation to $K+1$ terms, then the array in Fig. 3.3 has a trapezoidal shape, consisting of $K+1$ columns (including the one with $k=0$).

We now apply (3.90) and Theorem 3.2.2 to the case of the composite trapezoidal rule,

$$a_0 = \int_a^b f(x)\mathrm{d}x, \tag{3.91}$$

$$A(h) = h\left\{\frac{1}{2}f(a) + \sum_{k=1}^{n-1} f(a+kh) + \frac{1}{2}f(b)\right\}, \quad h = \frac{b-a}{n}. \tag{3.92}$$

The development of an asymptotic expansion for $A(h)$ in (3.92) is far from trivial. In fact, the result is the content of a well-known formula due to Euler[13] and Maclaurin.[14]

Before stating it, we need to define the *Bernoulli*[15] *numbers* B_k; these are the coefficients in the expansion

$$\frac{z}{\mathrm{e}^z - 1} = \sum_{k=0}^{\infty} \frac{B_k}{k!} z^k, \quad |z| < 2\pi. \tag{3.93}$$

[13] Leonhard Euler (1707–1783) was the son of a minister interested in mathematics who followed lectures of Jakob Bernoulli at the University of Basel. Euler himself was allowed to see Johann Bernoulli on Saturday afternoons for private tutoring. At the age of 20, after he was unsuccessful in obtaining a professorship in physics at the University of Basel, anectodically because of a lottery system then in use (Euler lost), he emigrated to St. Petersburg; later, he moved on to Berlin, and then back again to St. Petersburg. Euler unquestionably was the most prolific mathematician of the 18th century, working in virtually all branches of the differential and integral calculus and, in particular, being one of the founders of the calculus of variations. He also did pioneering work in the applied sciences, notably hydrodynamics, mechanics of deformable materials and rigid bodies, optics, astronomy, and the theory of the spinning top. Not even his blindness at the age of 59 managed to break his phenomenal productivity. Euler's collected works are still being edited, 71 volumes having already been published.

[14] Colin Maclaurin (1698–1764) was a Scottish mathematician who applied the new infinitesimal calculus to various problems in geometry. He is best known for his power series expansion, but also contributed to the theory of equations.

[15] Jakob Bernoulli (1654–1705), the elder brother of Johann Bernoulli, was active in Basel. He was one of the first to appreciate Leibniz's and Newton's differential and integral calculus and enriched it by many original contributions of his own, often in (not always amicable) competition with his younger brother. He is also known in probability theory for his "law of large numbers."

It is known that

$$\begin{aligned} &B_0 = 1, \quad B_2 = \frac{1}{6}, \quad B_4 = -\frac{1}{30}, \quad B_6 = \frac{1}{42}, \ldots, \\ &B_1 = -\frac{1}{2}, \quad B_3 = B_5 = \cdots = 0. \end{aligned} \tag{3.94}$$

Furthermore,

$$|B_k| \sim 2\,\frac{k!}{(2\pi)^k} \quad \text{as} \quad k \text{ (even)} \to \infty. \tag{3.95}$$

We now state without proof[16]

Theorem 3.2.3. (Euler–Maclaurin formula). *Let $A(h)$ be defined by (3.92), where $f \in C^{2K+1}[a,b]$ for some integer $K \geq 1$. Then*

$$A(h) = a_0 + a_1 h^2 + a_2 h^4 + \cdots + a_K h^{2K} + O(h^{2K+1}), \quad h \to 0, \tag{3.96}$$

[16] A heuristic derivation of the formal expansion (3.96) and (3.97), very much in the spirit of Euler, may proceed as follows. We start from Taylor's expansion (where $x, x+h \in [a,b]$)

$$f(x+h) - f(x) = \sum_{k=1}^{\infty} \frac{h^k D^k}{k!} f(x) = (\mathrm{e}^{hD} - 1) f(x), \quad D = \frac{\mathrm{d}}{\mathrm{d}x}.$$

Solving formally for $f(x)$, we get, using (3.93),

$$f(x) = (\mathrm{e}^{hD} - 1)^{-1}[f(x+h) - f(x)] = \sum_{r=0}^{\infty} \frac{B_r}{r!}(hD)^{r-1}[f(x+h) - f(x)];$$

that is, in view of (3.94),

$$f(x) = [(hD)^{-1} - \frac{1}{2} + \sum_{r=2}^{\infty} \frac{B_r}{r!}(hD)^{r-1}][f(x+h) - f(x)]$$

$$= (hD)^{-1}[f(x+h) - f(x)] - \frac{1}{2}[f(x+h) - f(x)] + \sum_{r=2}^{\infty} \frac{B_r}{r!} h^{r-1}[f^{(r-1)}(x+h) - f^{(r-1)}(x)]$$

$$= \frac{1}{h}\int_x^{x+h} f(t)\mathrm{d}t - \frac{1}{2}[f(x+h) - f(x)] + \sum_{k=1}^{\infty} \frac{B_{2k}}{(2k)!} h^{2k-1}[f^{(2k-1)}(x+h) - f^{(2k-1)}(x)].$$

Therefore, bringing the second term to the left-hand side, and multiplying through by h,

$$\frac{h}{2}[f(x+h) + f(x)] = \int_x^{x+h} f(t)\mathrm{d}t + \sum_{k=1}^{\infty} \frac{B_{2k}}{(2k)!} h^{2k}[f^{(2k-1)}(x+h) - f^{(2k-1)}(x)].$$

Now letting $x = a + ih$ and summing over i from 0 to $n-1$ gives

$$A(h) = \int_a^b f(t)\mathrm{d}t + \sum_{k=1}^{\infty} \frac{B_{2k}}{(2k)!} h^{2k}[f^{(2k-1)}(b) - f^{(2k-1)}(a)].$$

where a_0 is given by (3.91) and

$$a_k = \frac{B_{2k}}{(2k)!}\,[f^{(2k-1)}(b) - f^{(2k-1)}(a)], \quad k = 1, 2, \ldots, K. \tag{3.97}$$

Thus, under the assumption of Theorem 3.2.3, we have in (3.96) an asymptotic approximation to $K+1$ terms for $A(h)$, with

$$p_k = 2k, \quad k = 1, 2, \ldots, K, \quad p_{K+1} = 2K + 1, \tag{3.98}$$

and an asymptotic expansion, if $K = \infty$ (i.e., if f has continuous derivatives of any order). Choosing $h_0 = b - a$, $q = \frac{1}{2}$ in (3.89), which is certainly permissible, the scheme (3.90) becomes

$$A_{m,k} = A_{m,k-1} + \frac{A_{m,k-1} - A_{m-1,k-1}}{4^k - 1}, \quad m \geq k \geq 1,$$

and is known as the *Romberg integration* scheme. The choice $q = \frac{1}{2}$ is particularly convenient here, since in the generation of $A_{m,0}$ we can reuse function values already computed (cf. MA 6).

There is an important instance in which the application of Romberg integration would be pointless, namely, when all coefficients $a_1, a_2, \ldots$ in (3.96) are zero. This is the case when f is periodic with period $b - a$, and smooth on $\mathbb{R}$. Indeed, we already know that the composite trapezoidal rule is then exceptionally accurate (cf. Sect. 3.2.1, (3.32)).

3.3 Notes to Chapter 3

Section 3.1. Here we are dealing strictly with numerical differentiation, that is, with the problem of obtaining approximations to derivatives that can be used for numerical evaluation. The problem of symbolic differentiation, where the goal is to obtain analytic expressions for derivatives of functions given in analytic form, is handled by most computer algebra systems such as Mathematica and Macsyma, and we refer to texts in this area cited in Sect. P3.2 under *Computer Algebra*. Another important approach to differentiation is what is referred to as automatic differentiation. Here, the objective is to create a program (i.e., a piece of software) for computing the derivatives of a function given in the form of a program or algorithm. Notable applications are to optimization (calculation of Jacobian and Hessian matrices) and to the solution of ordinary differential equations by Taylor expansion. For an early paper on this subject, see Kedem [1980], for a good cross-section of current activity, Griewank and Corliss [1991] and a more recent exposition, Griewank and Walther [2008]. Automatic differentiation in the context of Matlab object-oriented programming is discussed in Neidinger [2010].

Section 3.1.1. The interpolatory formulae for differentiation derived here are analogous to the Newton–Cotes formulae for integration (cf. Sect. 3.2.2) in the sense that one fixes $n+1$ distinct nodes $x_0, x_1, \dots, x_n$ and interpolates on them by a polynomial of degree n to ensure polynomial degree of exactness n. In analogy to Gauss quadrature formulae, one might well ask whether by a suitable choice of nodes one could substantially increase the degree of precision. If x_0 is the point at which f' is to be approximated, there is actually no compelling reason for including it among the points where f is evaluated. One may thus consider approximating $f'(x_0)$ by $L(f; x_0, h) = h^{-1} \sum_{i=0}^{n} w_i f(x_0 + t_i h)$ and choosing the (real) numbers w_i, t_i so that the integer d for which $Ef = f'(x_0) - L(f; x_0, h) = 0$, all $f \in \mathbb{P}_d$, is as large as possible. This, however, is where the analogy with Gauss ends. For one thing, the t_i need to be normalized somehow, since multiplying all t_i by a constant c and dividing w_i by c yields essentially the same formula. One way of normalization is to require that $|t_i - t_j| \geq 1$ for all $i \neq j$. More important, the possible improvement over $d = n$ (which is achievable by interpolation) is disappointing: the best we can do is obtain $d = n+1$. This can be shown by a simple matrix argument; see Ash et al. [1984]. Among the formulae with $d = n+1$ (which are not unique), one may define an optimal one that minimizes the absolute value of the coefficient in the leading term of the truncation error Ef. These have been derived for each n in Ash et al. [1984], not only for the first, but also for the second derivative. They seem to be optimal also in the presence of noise (for $n = 2$, see Ash and Jones [1981, Theorem 3.2.3]), but are still subject to the magnification of noise as exemplified in (3.18). To strike a balance between errors due to truncation and those due to noise, an appropriate step h may be found adaptively; see, for example, Stepleman and Winarsky [1979] and Oliver [1980].

For the sth derivative and its approximation by a formula L as in the preceding paragraph, where h^{-1} is to be replaced by h^{-s}, one may alternatively wish to minimize the "condition mumber" $\sum_{i=0}^{n} |w_i|$. Interestingly, if n and s have the same parity, the optimum is achieved by the extreme points of the nth-degree Chebyshev polynomial T_n; see Rivlin [1975] and Miel and Mooney [1985].

One can do better, especially for high-order derivatives, if one allows the t_i and w_i to be complex and assumes analyticity for f. For the sth derivative, it is then possible (cf. Lyness [1968]) to achieve degree of exactness $n+s$ by choosing the t_i to be the $n+1$ roots of unity; specifically, one applies the trapezoidal rule to Cauchy's integral for the sth derivative (see (3.19) for $s = 1$). A more sophisticated use of these trapezoidal sums is made in Lyness and Moler [1967]. For practical implementations of these ideas, and algorithms, see Lyness and Sande [1971] and Fornberg [1981].

Considering the derivative of a function f on some interval, say, [0, 1], as the solution on this interval of the (trivial) integral equation $\int_0^x u(t)\mathrm{d}t = f(x)$, one can try to combat noise in the data by applying "Tikhonov regularization" to this operator equation; this approach is studied, for example, in King and Murio [1986].

Section 3.2. The standard text on the numerical evaluation of integrals – simple as well as multiple – is Davis and Rabinowitz [2007]. It contains a valuable bibliography of over 1500 items. Other useful texts are Krylov [1962], Brass [1977], Engels [1980], and Evans [1993], the last being more practical and application-oriented than the others and containing a detailed discussion of oscillatory integrals. A broadly based, but concise, text is Krommer and Ueberhuber [1998]. Most quadrature rules are designed to integrate exactly polynomials up to some degree d, that is, all solutions of the differential equation $u^{(d+1)} = 0$. One can generalize and require the rules to be exact for all solutions of a linear homogeneous differential equation of order $d + 1$. This is the approach taken in the book by Ghizzetti and Ossicini [1970]. There are classes of quadrature rules, perhaps more of theoretical than practical interest, that are not covered in our text. One such concerns rules, first studied by Chebyshev, whose weights are all equal (which minimizes the effects of small random errors in the function values), or whose weights have small variance. Surveys on these are given in Gautschi [1976a] and Förster [1993]. Another class includes optimal quadrature formulae, which minimize, for prescribed or variable nodes, the supremum of the modulus of the remainder term taken over some suitable class of functions. For these, and their close relationship to "monosplines," we refer to Nikol'skiĭ [1988] or Levin and Girshovich [1979].

Symbolic integration is inherently more difficult than symbolic differentiation since integrals are often not expressible in analytic form, even if the integrand is an elementary function. A great amount of attention, however, has been given to the problem of automatic integration. Here, the user is required to specify the limits of integration, to provide a routine for evaluating the integrand function, and to indicate an error tolerance (absolute, relative, or mixed) and an upper bound for the number of function evaluations to be performed. The automatic integrator is then expected either to return an answer satisfying the user's criteria, or to report that the error criterion could not be satisfied within the desired volume of computation. In the latter event, a user-friendly integrator will offer a best-possible estimate for the value of the integral along with an estimate of the error. A popular collection of automatic integrators is Quadpack (see Piessens et al. [1983]), and a good description of the internal workings of automatic integration routines can be found in Davis and Rabinowitz [2007, Chap. 6].

For the important (and difficult) problem of multiple integration and related computational tools, we must refer to special texts, for example, Stroud [1971], Mysovskikh [1981] (for readers familiar with Russian), Sloan and Joe [1994], and Sobolev and Vaskevich [1997]. An update to Stroud [1971] is available in Cools and Rabinowitz [1993]. Monte Carlo methods are widely used in statistical physics and finance to compute high-dimensional integrals; texts discussing these methods are Niederreiter [1992], Sobol' [1994], Evans and Swartz [2000], and Kalos and Whitlock [2008].

In dealing with definite integrals, one should never lose sight of the many analytical tools available that may help in evaluating or approximating integrals. The reader will find the old, but still pertinent, essay of Abramowitz [1954] informative in this respect, and may also wish to consult Zwillinger [1992a].

Section 3.2.1. The result relating to (3.34), in a slightly different form, is proved in Davis and Rabinowitz [2007, p. 209]. Their proof carries over to integrals of the form $\int_{\mathbb{R}} f(x)\mathrm{d}x$, if one first lets $n \to \infty$, and then $a \to -\infty$, in their proof.

Section 3.2.2. The classical Newton–Cotes formulae (3.35) involve equally spaced nodes (on $[-1, 1]$) and weight function $w \equiv 1$. They are useful only for relatively small values of n, since for large n the weights become large and oscillatory in sign. Choosing Chebyshev points instead removes this obstacle and gives rise to the Fejér quadrature rule. Close relatives are the Filippi rule, which uses the local extreme points of T_{n+1}, and the Clenshaw–Curtis rule, which uses all extreme points (including ± 1) of T_{n-1} as nodes. All three quadrature rules have weights that can be explicitly expressed in terms of trigonometric functions, and they are all positive. The latter has been proved for the first two rules by Fejér [1933], and for the last by Imhof [1963]. Formulas with Chebyshev points of the third and fourth kind, with or without the endpoints, are studied in Notaris [1997], [1998]. Algorithms for accurately computing weighted Newton–Cotes formulae are discussed in Kautsky and Elhay [1982] and Gautschi [1997] (see also Gautschi [2011a, Sect.4.1] for an improvement); for a computer program, see Elhay and Kautsky [1987].

It is difficult to trace the origin of Theorem 3.2.1, but in essence, Jacobi already was aware of it in 1826, and the idea of the proof, using division by the node polynomial, is his (Jacobi [1826]). There are other noteworthy applications of Theorem 3.2.1. One is to quadrature rules with $2n + 1$ points, where n of them are Gauss points and the remaining $n+1$ are to be selected, together with all the weights, so as to make the degree of exactness as large as possible. These are called Gauss–Kronrod formulae (cf. Ex. 19) and have found use in automatic integration routines. Interest has focused on the polynomial of degree $n + 1$ – the Stieltjes polynomial – whose zeros are the $n + 1$ nodes added to the Gauss nodes. In particular, this polynomial must be orthogonal to all polynomials of lower degree with respect to the "weight function" $\pi_n(t; w)w(t)$. The oscillatory character of this weight function poses intriguing questions regarding the location relative to the Gauss nodes, or even the reality, of the added nodes. For a discussion of these and related matters, see the surveys in Gautschi [1988] and Notaris [1994].

There is a theorem analogous to Theorem 3.2.1 that deals with quadrature rules having multiple nodes. The simplest one, first studied by Turán [1950], has constant multiplicity $2s + 1$ ($s \geq 0$) for each node; that is, on the right of (3.35), there are also terms involving derivatives up to order $2s$ for each node t_k (cf. Ex. 20 for $s = 1$). If one applies Gauss's principle of maximum algebraic degree of exactness to them, one is led to define the t_k as the zeros of a polynomial of degree n whose $(2s + 1)$st power is orthogonal to all lower-degree polynomials (cf. Gautschi [1981, Sect. 2.2.1]). This gives rise to what are called s-orthogonal polynomials and to generalizations thereof pertaining to multiplicities that vary from node to node; a good reference for this is Ghizzetti and Ossicini [1970, Chap. 3, Sect. 3.9]; see also Gautschi [1981, Sect. 2.2] and Chap. 4, Sect. 4.1.4.

Another class of Gauss-type formulae, where exactness is required not only for polynomials (if at all), but also for rational functions (with prescribed poles), has been developed by Van Assche and Vanherwegen [1993] and Gautschi [1993]. They are particularly useful if the integrand has poles near the interval of integration. One can, of course, require exactness for still other systems of functions; for literature on this, see Gautschi [1981, Sect. 2.3.3]. Exactness of Gauss formulae for parabolic splines is dicussed in Nikolov and Simian [2011].

Section 3.2.3. (c) The convergence theory for Gauss formulae on infinite intervals is more delicate. Some general results can be found in the book by Freud [1971, Chap. 3, Sect. 3.1].

(e) The importance for Gauss quadrature rules of eigenvalues and eigenvectors of the Jacobi matrix and related computational algorithms was first elaborated by Golub and Welsch [1969], although the idea is older. Similar eigenvalue techniques apply to Gauss–Radau and Gauss–Lobatto formulae (Golub [1973]), and indeed to Gauss–Kronrod formulae as well (Laurie [1997]).

A list of published tables of Gauss formulae for various classical and nonclassical weight functions is contained in Gautschi [1981, Sect. 5.4], where one also finds a detailed history of Gauss–Christoffel quadrature rules and extensions thereof. A table of recurrence coefficients, in particular, also for Jacobi weight functions, can be found in the Appendix to Chihara [1978] and in Gautschi [2004, Sect. 1.5]. For practical purposes, it is important to be able to automatically generate Gauss formulae as needed, even if the Jacobi matrix for them is unknown (and must itself be computed). A major first step in this direction is the Fortran computer package in Gautschi [1994b] based on earlier work of the author in Gautschi [1982], and the more recent Matlab packages `OPQ, SOPQ` on the Web at the URL cited in Gautschi [2004, Preface].

Section 3.2.4. Other applications of classical Gaussian quadrature rules, notably to product integration of multiple integrals, are described in the book by Stroud and Secrest [1966], which also contains extensive high-precision tables of Gauss formulae. Prominent use of Gaussian quadrature, especially with Jacobi weight functions, is made in the evaluation of Cauchy principal value integrals in connection with singular integral equations; for this, see, for example, Gautschi [1981, Sect. 3.2]. A number of problems in approximation theory that can be solved by nonclassical Gaussian quadrature rules are discussed in Gautschi [1996].

Section 3.2.7. A classical account of Romberg integration – one that made this procedure popular – is Bauer et al. [1963]. The basic idea, however, can be traced back to nineteenth-century mathematics, and even beyond. An extensive survey not only of the history, but also of the applications and modifications of extrapolation methods, is given in Joyce [1971] and supplemented in Rabinowitz [1992]. See also Engels [1979] and Dutka [1984] for additional historical accounts. Romberg schemes for other sequences of composite trapezoidal rules are discussed in Fischer [2002].

Richardson extrapolation is just one of many techniques to accelerate the convergence of sequences. For others, we refer to the books by Wimp [1981], Brezinski and Redivo-Zaglia [1991] (containing also computer programs), and Sidi [2003]. A book with emphasis on linear extrapolation methods and the existence of related asymptotic expansions is Walz [1996].

Exercises and Machine Assignments to Chapter 3

Exercises

1. From (3.4)–(3.6) with $n = 3$ we know that

$$\begin{aligned} f'(x_0) =& [x_0, x_1]f + (x_0 - x_1)[x_0, x_1, x_2]f \\ &+ (x_0 - x_1)(x_0 - x_2)[x_0, x_1, x_2, x_3]f + e_3, \end{aligned}$$

where $e_3 = (x_0 - x_1)(x_0 - x_2)(x_0 - x_3)\frac{f^{(4)}(\xi)}{4!}$. Apply this to

$$x_0 = 0, \quad x_1 = \frac{1}{8}, \quad x_2 = \frac{1}{4}, \quad x_3 = \frac{1}{2}$$

and express $f'(0)$ as a linear combination of $f_k = f(x_k)$, $k = 0, 1, 2, 3$, and e_3. Also, estimate the error e_3 in terms of $M_4 = \max_{0 \le x \le \frac{1}{2}} |f^{(4)}(x)|$.

2. Derive a formula for the error term $r_n'(x)$ of numerical differentiation analogous to (3.5) but for $x \neq x_0$. {*Hint*: use Chap. 2, (2.116) in combination with Chap. 2, Ex. 58.}

3. Let x_i, $i = 0, 1, \ldots, n$, be $n + 1$ distinct points with $H = \max_{1 \le i \le n} |x_i - x_0|$ small.

   (a) Show that for $k = 0, 1, \ldots, n$ one has

$$\left.\frac{\mathrm{d}^k}{\mathrm{d}x^k} \prod_{i=1}^{n} (x - x_i)\right|_{x=x_0} = O(H^{n-k}) \quad \text{as} \ \ H \to 0.$$

   (b) Prove that

$$f^{(n)}(x_0) = n!\,[x_0, x_1, \ldots, x_n]f + e_n,$$

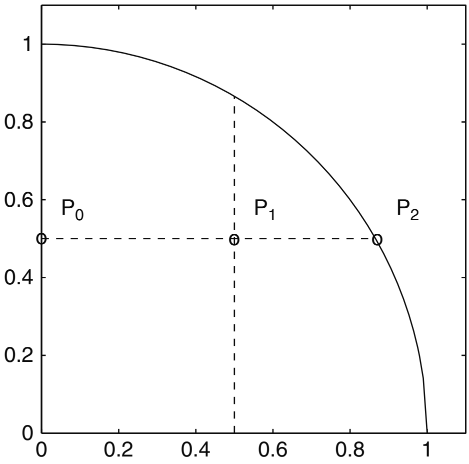
   where

$$e_n = \begin{cases} O(H^2) & \text{if} \ \ x_0 = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} x_i, \\ O(H) & \text{otherwise}, \end{cases}$$

   assuming that f is sufficiently often (how often?) differentiable in the interval spanned by the x_i. {*Hint*: use the Newton interpolation formula with remainder, in combination with Leibniz's rule of differentiation.}

   (c) Specialize the formula in (b) to equally spaced points x_i with spacing h and express the result in terms of either the nth forward difference $\Delta^n f_0$ or the nth backward difference $\nabla^n f_n$ of the values $f_i = f(x_i)$. {Here, $\Delta f_0 = f_1 - f_0$, $\Delta^2 f_0 = \Delta(\Delta f_0) = \Delta f_1 - \Delta f_0 = f_2 - 2f_1 + f_0$, etc., and similarly for ∇f_1, $\nabla^2 f_2$, and so on.}

4. Approximate $\partial u/\partial x|_{P_1}$ in terms of $u_0 = u(P_0)$, $u_1 = u(P_1)$, $u_2 = u(P_2)$ (see figure, where the curve represents a quarter arc of the unit circle). Estimate the error.



5. (a) Use the central difference quotient approximation $f'(x) \approx [f(x+h) - f(x-h)]/(2h)$ of the first derivative to obtain an approximation of $\dfrac{\partial^2 u}{\partial x \partial y}(x, y)$ for a function u of two variables.
   (b) Use Taylor expansion of a function of two variables to show that the error of the approximation derived in (a) is $O(h^2)$.
6. Consider the integral $I = \int_{-1}^{1} |x| \mathrm{d}x$, whose exact value is evidently 1. Suppose I is approximated (as it stands) by the composite trapezoidal rule $T(h)$ with $h = 2/n, n = 1, 2, 3, \ldots$.

   (a) Show (without any computation) that $T(2/n) = 1$ if n is even.
   (b) Determine $T(2/n)$ for n odd and comment on the speed of convergence.

7. Let

$$I(h) = \int_0^h f(x)\mathrm{d}x, \quad T(h) = \frac{h}{2}\,[f(0) + f(h)].$$

   (a) Evaluate $I(h)$, $T(h)$, and $E(h) = I(h) - T(h)$ explicitly for $f(x) = x^2 + x^{5/2}$.
   (b) Repeat for $f(x) = x^2 + x^{1/2}$. Explain the discrepancy that you will observe in the order of the error terms.

8. (a) Derive the "midpoint rule" of integration

$$\int_{x_k}^{x_k+h} f(x)\mathrm{d}x = hf\left(x_k + \frac{1}{2}h\right) + \frac{1}{24}h^3 f''(\xi), \quad x_k < \xi < x_k + h.$$

{*Hint*: use Taylor's theorem centered at $x_k + \frac{1}{2}h$.}

(b) Obtain the composite midpoint rule for $\int_a^b f(x)\mathrm{d}x$, including the error term, subdividing $[a, b]$ into n subintervals of length $h = \frac{b-a}{n}$.

9. (a) Show that the elementary Simpson's rule can be obtained as follows:

$$\int_{-1}^{1} y(t)\mathrm{d}t = \int_{-1}^{1} p_3(y; -1, 0, 0, 1; t)\mathrm{d}t + E^{\mathrm{S}}(y).$$

(b) Obtain a formula for the remainder $E^{\mathrm{S}}(y)$, assuming $y \in C^4[-1, 1]$.

(c) Using (a) and (b), derive the composite Simpson's rule for $\int_a^b f(x)\mathrm{d}x$, including the remainder term.

10. Let $E_n^{\mathrm{S}}(f)$ be the remainder term of the composite Simpson's rule for $\int_0^{2\pi} f(x)\mathrm{d}x$ using n subintervals (n even). Evaluate $E_n^{\mathrm{S}}(f)$ for $f(x) = \mathrm{e}^{\mathrm{i}mx}$ ($m = 0, 1, \dots$). Hence determine for what values of d Simpson's rule integrates exactly (on $[0, 2\pi]$) trigonometric polynomials of degree d.

11. Estimate the number of subintervals required to obtain $\int_0^1 \mathrm{e}^{-x^2}\mathrm{d}x$ to 6 correct decimal places (absolute error $\leq \frac{1}{2} \times 10^{-6}$)

(a) by means of the composite trapezoidal rule,

(b) by means of the composite Simpson's rule.

12. Let f be an arbitrary (continuous) function on [0,1] satisfying $f(x) + f(1-x) \equiv 1$ for $0 \leq x \leq 1$.

(a) Show that $\int_0^1 f(x)\mathrm{d}x = \frac{1}{2}$.

(b) Show that the composite trapezoidal rule for computing $\int_0^1 f(x)\mathrm{d}x$ is exact.

(c) Show, with as little computation as possible, that the composite Simpson's rule and more general symmetric rules are also exact.

13. (a) Construct a trapezoidal-like formula

$$\int_0^h f(x)\mathrm{d}x = af(0) + bf(h) + E(f), \quad 0 < h < \pi,$$

which is exact for $f(x) = \cos x$ and $f(x) = \sin x$. Does this formula integrate constants exactly?

(b) Show that a similar formula holds for $\int_c^{c+h} g(t)\mathrm{d}t$.

14. Given the subdivision Δ of $[0, 2\pi]$ into N equal subintervals, $0 = x_0 < x_1 < x_2 < \cdots < x_{N-1} < x_N = 2\pi$, $x_k = kh$, $h = 2\pi/N$, and a (2π)-periodic function f, construct a quadrature rule for the mth (complex) Fourier coefficients of f,
$$\frac{1}{2\pi}\int_0^{2\pi} f(x)\mathrm{e}^{-\mathrm{i}mx}\mathrm{d}x,$$
by approximating f by the spline interpolant $s_1(f;\cdot)$ from $\mathbb{S}_1^0(\Delta)$. Write the result in the form of a "modified" composite trapezoidal approximation. {*Hint*: express $s_1(f;\cdot)$ in terms of the hat functions defined in Chap. 2, (2.129).}

15. The composite trapezoidal rule for computing $\int_0^1 f(x)\mathrm{d}x$ can be generalized to subdivisions
$$\Delta: \quad 0 = x_0 < x_1 < x_2 < \cdots < x_{n-1} < x_n = 1$$
of the interval [0,1] in subintervals of arbitrary length $\Delta x_i = x_{i+1} - x_i$, $i = 0, 1, \ldots, n-1$, by approximating
$$\int_0^1 f(x)\mathrm{d}x \approx \int_0^1 s_1(f;x)\mathrm{d}x,$$
where $s_1(f;\cdot) \in \mathbb{S}_1^0(\Delta)$ is the piecewise linear continuous spline interpolating f at $x_0, x_1, \ldots, x_n$.
    (a) Use the basis of hat functions $B_0, B_1, \ldots, B_n$ to represent $s_1(f;\cdot)$ and calculate $\int_0^1 s_1(f;x)\mathrm{d}x$.
    (b) Discuss the error $E(f) = \int_0^1 f(x)\mathrm{d}x - \int_0^1 s_1(f;x)\mathrm{d}x$. In particular, find a formula of the type $E(f) = \text{const} \cdot f''(\xi)$, $0 < \xi < 1$, where the constant depends only on Δ.

16. (a) Construct the weighted Newton–Cotes formula
$$\int_0^1 f(x)x^\alpha \mathrm{d}x = a_0 f(0) + a_1 f(1) + E(f), \quad \alpha > -1.$$
Explain why the formula obtained makes good sense.
    (b) Derive an expression for the error term $E(f)$ in terms of an appropriate derivative of f.
    (c) From the formulae in (a) and (b) derive an approximate integration formula for $\int_0^h g(t)t^\alpha \mathrm{d}t$ ($h > 0$ small), including an expression for the error term.

17. (a) Construct the weighted Newton–Cotes formula
$$\int_0^1 f(x) \cdot x\ln(1/x)\mathrm{d}x \approx a_0 f(0) + a_1 f(1).$$
{*Hint:* use $\int_0^1 x^r \ln(1/x)\mathrm{d}x = (r+1)^{-2}$, $r = 0, 1, 2, \ldots$.}

(b) Discuss how the formula in (a) can be used to approximate $\int_0^h g(t)\cdot t\ln(1/t)\mathrm{d}t$ for small $h>0$. {*Hint*: make a change of variables.}

18. Let s be the function defined by

$$s(x)=\begin{cases}(x+1)^3 & \text{if } -1\le x\le 0,\\ (1-x)^3 & \text{if } 0\le x\le 1.\end{cases}$$

(a) With Δ denoting the subdivision of $[-1,1]$ into the two subintervals $[-1,0]$ and $[0,1]$, to what class $\mathbb{S}_m^k(\Delta)$ does the spline s belong?
(b) Estimate the error of the composite trapezoidal rule applied to $\int_{-1}^1 s(x)\mathrm{d}x$, when $[-1,1]$ is divided into n subintervals of equal length $h=2/n$ and n is even.
(c) What is the error of the composite Simpson's rule applied to $\int_{-1}^1 s(x)\mathrm{d}x$, with the same subdivision of $[-1,1]$ as in (b)?
(d) What is the error resulting from applying the 2-point Gauss–Legendre rule to $\int_{-1}^0 s(x)\mathrm{d}x$ and $\int_0^1 s(x)\mathrm{d}x$ separately and summing?

19. (Gauss–Kronrod rule) Let $\pi_n(\,\cdot\,;w)$ be the (monic) orthogonal polynomial of degree n relative to a nonnegative weight function w on $[a,b]$, and $t_k^{(n)}$ its zeros. Use Theorem 3.2.1 to determine conditions on w_k, w_k^*, t_k^* for the quadrature rule

$$\int_a^b f(t)w(t)\mathrm{d}t=\sum_{k=1}^n w_k f(t_k^{(n)})+\sum_{k=1}^{n+1} w_k^* f(t_k^*)+E_n(f)$$

to have degree of exactness at least $3n+1$; that is, $E_n(f)=0$ for all $f\in\mathbb{P}_{3n+1}$.

20. (Turán quadrature formula) Let w be a nonnegative weight function on $[a,b]$. Prove: the quadrature formula

$$\int_a^b f(t)w(t)\mathrm{d}t=\sum_{k=1}^n [w_k f(t_k)+w_k' f'(t_k)+w_k'' f''(t_k)]+E_n(f)$$

has degree of exactness $d=4n-1$ if and only if the following conditions are satisfied:

(a) The formula is (Hermite-) interpolatory; that is, $E_n(f)=0$ if $f\in\mathbb{P}_{3n-1}$.
(b) The node polynomial $\omega_n(t)=\Pi_{k=1}^n(t-t_k)$ satisfies

$$\int_a^b [\omega_n(t)]^3 p(t)w(t)\mathrm{d}t=0 \quad \text{for all } p\in\mathbb{P}_{n-1}.$$

{*Hint*: simulate the proof of Theorem 3.2.1.}

21. Consider $s>1$ weight functions $w_\sigma(t)$, $\sigma=1,2,\dots,s$, integers m_σ such that $\sum_{\sigma=1}^s m_\sigma=n$, and s quadrature rules

$$Q_\sigma:\quad \int_a^b f(t)w_\sigma(t)\mathrm{d}t=\sum_{k=1}^n w_{k,\sigma} f(t_k)+E_{n,\sigma}(f),\quad \sigma=1,2,\dots,s,$$

which share n common nodes t_k but have individual weights $w_{k,\sigma}$. State necessary and sufficient conditions for Q_σ to have degree of exactness $n + m_\sigma - 1$, $\sigma = 1, 2, \dots, s$, and explain why this is likely to be optimal.

22. Consider a quadrature formula of the form

$$\int_0^1 f(x)\mathrm{d}x \approx a_0 f(0) + a_1 f'(0) + \sum_{k=1}^{n} w_k f(x_k) + b_0 f(1).$$

   (a) Call the formula "Hermite-interpolatory" if the right-hand side is obtained by integrating on the left instead of f the (Hermite) interpolation polynomial p satisfying

$$\begin{aligned} p(0) = f(0), \quad p'(0) = f'(0), \quad p(1) = f(1), \\ p(x_k) = f(x_k), \quad k = 1, 2, \dots, n. \end{aligned}$$

   What degree of exactness does the formula have in this case (regardless of how the nodes x_k are chosen, as long as they are mutually distinct and strictly inside the interval [0, 1])?

   (b) What is the maximum degree of exactness expected to be if all coefficients and nodes x_k are allowed to be freely chosen?

   (c) Show that for the maximum degree of exactness to be achieved, it is necessary that $\{x_k\}$ are the zeros of the polynomial π_n of degree n which is orthogonal on [0, 1] with respect to the weight function $w(x) = x^2(1-x)$. Identify this polynomial in terms of one of the classical orthogonal polynomials.

   (d) Show that the choice of the x_k in (c) together with the requirement of the quadrature formula to be Hermite-interpolatory is sufficient for the maximum degree of exactness to be attained.

23. Show that the Gauss–Radau as well as the Gauss–Lobatto formulae are positive if the weight function w is nonnegative and not identically zero. $\{$*Hint*: modify the proof given for the Gauss formula in Sect. 3.2.3(b).$\}$ What are the implications with regard to convergence as $n \to \infty$ of the formulae?

24. (Fejér, 1933). Let t_k, $k = 1, 2, \dots, n$, be the zeros of

$$\omega_n(t) = P_n(t) + \alpha P_{n-1}(t) + \beta P_{n-2}(t), \quad n \geq 2,$$

where $\{P_k\}$ are the Legendre polynomials, and assume $\alpha \in \mathbb{R}$, $\beta \leq 0$, and the zeros t_k real and pairwise distinct. Show that the Newton–Cotes formula

$$\int_{-1}^{1} f(t)\mathrm{d}t = \sum_{k=1}^{n} w_k f(t_k) + E_n(f), \quad E_n(\mathbb{P}_{n-1}) = 0,$$

has all weights positive: $w_k > 0$ for $k = 1, 2, \dots, n$. $\{$*Hint*: define $\Delta_k(t) = [\ell_k(t)]^2 - \ell_k(t)$ and show that $\int_{-1}^{1} \Delta_k(t)\mathrm{d}t \leq 0$.$\}$

25. (a) Determine by Newton's interpolation formula the quadratic polynomial p interpolating f at $x = 0$ and $x = 1$ and f' at $x = 0$. Also, express the error in terms of an appropriate derivative (assumed continuous on [0,1]).
    (b) Based on the result of (a), derive an integration formula of the type
    $$\int_0^1 f(x)\mathrm{d}x = a_0 f(0) + a_1 f(1) + b_0 f'(0) + E(f).$$
    Determine a_0, a_1, b_0 and an appropriate expression for $E(f)$.
    (c) Transform the result of (b) to obtain an integration rule, with remainder, for $\int_c^{c+h} y(t)\mathrm{d}t$, where $h > 0$. {Do not rederive this rule from scratch.}
26. Imitate the procedures used in the Example of Chap. 2, Sect. 2.1.4(2) for monic Legendre polynomials to show orthogonality on $[0,\infty)$ relative to the Laguerre measure $\mathrm{d}\lambda(t) = t^\alpha \mathrm{e}^{-t}\mathrm{d}t$, $\alpha > -1$, of the (monic) polynomials
    $$\pi_k(t) = (-1)^k t^{-\alpha} \mathrm{e}^t \frac{\mathrm{d}^k}{\mathrm{d}t^k}(t^{\alpha+k}\mathrm{e}^{-t}), \quad k = 0, 1, 2, \ldots,$$
    and to derive explicit formulae for the recursion coefficients α_k, β_k. {*Hint*: express α_k and β_k in terms of the coefficients λ_k, μ_k in $\pi_k(t) = t^k + \lambda_k t^{k-1} + \mu_k t^{k-2} + \cdots$.}
27. Show that
    $$\pi_k(t) = \frac{(-1)^k}{2^k} \mathrm{e}^{t^2} \frac{\mathrm{d}^k}{\mathrm{d}t^k}(\mathrm{e}^{-t^2}), \quad k = 0, 1, 2, \ldots ,$$
    are the monic orthogonal polynomials on $\mathbb{R}$ relative to the Hermite measure $\mathrm{d}\lambda(t) = \mathrm{e}^{-t^2}\mathrm{d}t$. Use this "Rodrigues formula" directly to derive the recurrence relation for the (monic) Hermite polynomials.
28. (a) Construct the quadratic (monic) polynomial $\pi_2(\,\cdot\,; w)$ orthogonal on $(0,\infty)$ with respect to the weight function $w(t) = \mathrm{e}^{-t}$. {*Hint*: use $\int_0^\infty t^m \mathrm{e}^{-t}\mathrm{d}t = m!$.}
    (b) Obtain the two-point Gauss–Laguerre quadrature formula,
    $$\int_0^\infty f(t)\mathrm{e}^{-t}\mathrm{d}t = w_1 f(t_1) + w_2 f(t_2) + E_2(f),$$
    including a representation for the remainder $E_2(f)$.
    (c) Apply the formula in (b) to approximate $I = \int_0^\infty \mathrm{e}^{-t}\mathrm{d}t/(t+1)$. Use the remainder term $E_2(f)$ to estimate the error, and compare your estimate with the true error {use $I = 0.596347361\ldots$}. Knowing the true error, identify the unknown quantity $\xi > 0$ contained in the error term $E_2(f)$.
29. Derive the 2-point Gauss–Hermite quadrature formula,
    $$\int_{-\infty}^\infty f(t)\mathrm{e}^{-t^2}\mathrm{d}t = w_1 f(t_1) + w_2 f(t_2) + E_2(f),$$
    including an expression for the remainder $E_2(f)$. {*Hint*: use $\int_0^\infty t^{2n}\mathrm{e}^{-t^2}\mathrm{d}t = \frac{(2n)!}{n!2^{2n}}\frac{\sqrt{\pi}}{2}$, $n = 0, 1, 2, \ldots$.}

30. Let $\pi_n(\,\cdot\,;w)$ be the nth-degree orthogonal polynomial with respect to the weight function w on $[a,b]$, $t_1, t_2, \ldots, t_n$ its n zeros, and w_1, $w_2, \ldots, w_n$ the n Gauss weights.

    (a) Assuming $n > 1$, show that the n polynomials π_0, $\pi_1, \ldots, \pi_{n-1}$ are also orthogonal with respect to the *discrete* inner product $(u,v) = \sum_{\nu=1}^{n} w_\nu u(t_\nu)v(t_\nu)$.

    (b) With $\ell_i(t) = \prod_{k\neq i}[(t-t_k)/(t_i-t_k)]$, $i = 1,2,\ldots,n$, denoting the elementary Lagrange interpolation polynomials associated with the nodes $t_1, t_2, \ldots, t_n$, show that

$$\int_a^b \ell_i(t)\ell_k(t)w(t)\mathrm{d}t = 0 \;\text{ if }\; i \neq k.$$

31. Consider a quadrature formula of the type

$$\int_0^\infty \mathrm{e}^{-x} f(x)\mathrm{d}x = af(0) + bf(c) + E(f).$$

    (a) Find a, b, c such that the formula has degree of exactness $d = 2$. Can you identify the formula so obtained? {*Point of information*: $\int_0^\infty \mathrm{e}^{-x}x^r\,\mathrm{d}x = r!$}

    (b) Let $p_2(x) = p_2(f;0,2,2;x)$ be the Hermite interpolation polynomial interpolating f at the (simple) point $x = 0$ and the double point $x = 2$. Determine $\int_0^\infty \mathrm{e}^{-x}p_2(x)\mathrm{d}x$ and compare with the result in (a).

    (c) Obtain the remainder $E(f)$ in the form $E(f) = \text{const}\cdot f'''(\xi)$, $\xi > 0$.

32. In this problem, $\pi_j(\,\cdot\,;w)$ denotes the monic polynomial of degree j orthogonal on the interval $[a,b]$ relative to a weight function $w \geq 0$.

    (a) Show that $\pi_n(\,\cdot\,;w)$, $n > 0$, has at least one real zero in the interior of $[a,b]$ at which π_n changes sign.

    (b) Prove that all zeros of $\pi_n(\,\cdot\,;w)$ are real, simple, and contained in the interior of $[a,b]$. {*Hint*: put $r_0 = \max\{r \geq 1\colon t_{k_1}^{(n)}, t_{k_2}^{(n)}, \ldots, t_{k_r}^{(n)}$ are distinct real zeros of π_n in (a,b) at each of which π_n changes sign}. Show that $r_0 = n$.}

33. Prove that the zeros of $\pi_n(\,\cdot\,;w)$ interlace with those of $\pi_{n+1}(\,\cdot\,;w)$.

34. Consider the Hermite interpolation problem: Find $p \in \mathbb{P}_{2n-1}$ such that

$$(*) \qquad p(\tau_\nu) = f_\nu, \;\; p'(\tau_\nu) = f'_\nu, \;\; \nu = 1,2,\ldots,n.$$

There are "elementary Hermite interpolation polynomials" h_ν, k_ν such that the solution of (*) can be expressed (in analogy to Lagrange's formula) in the form

$$p(t) = \sum_{\nu=1}^{n}[h_\nu(t)f_\nu + k_\nu(t)f'_\nu].$$

(a) Seek h_ν and k_ν in the form

$$h_\nu(t) = (a_\nu + b_\nu t)\ell_\nu^2(t), \;\; k_\nu(t) = (c_\nu + d_\nu t)l_\nu^2(t),$$

where ℓ_ν are the elementary Lagrange polynomials. Determine the constants $a_\nu, b_\nu, c_\nu, d_\nu$.

(b) Obtain the quadrature rule

$$\int_a^b f(t)w(t)\mathrm{d}t = \sum_{\nu=1}^n [\lambda_\nu f(\tau_\nu) + \mu_\nu f'(\tau_\nu)] + E_n(f)$$

with the property that $E_n(f) = 0$ for all $f \in \mathbb{P}_{2n-1}$.

(c) What conditions on the node polynomial $\omega_n(t) = \prod_{\nu=1}^n (t - \tau_\nu)$ (or on the nodes τ_ν) must be imposed in order that $\mu_\nu = 0$ for $\nu = 1, 2, \dots, n$?

35. Show that $\int_0^1 (1-t)^{-1/2} f(t)\mathrm{d}t$, when f is smooth, can be computed accurately by Gauss–Legendre quadrature. {*Hint*: substitute $1 - t = x^2$.}

36. The Gaussian quadrature rule for the (Chebyshev) weight function $w(t) = (1-t^2)^{-1/2}$ is known to be

$$\int_{-1}^1 f(t)(1-t^2)^{-1/2}\mathrm{d}t \approx \frac{\pi}{n}\sum_{k=1}^n f(t_k^{\mathrm{C}}), \quad t_k^{\mathrm{C}} = \cos\left(\frac{2k-1}{2n}\pi\right).$$

(The nodes t_k^{C} are the n Chebyshev points.) Use this fact to show that the unit disk has area π.

37. Assuming f is a well-behaved function, discuss how the following integrals can be approximated by *standard* Gauss-type rules (i.e., with canonical intervals and weight functions).

(a) $\int_a^b f(x)\mathrm{d}x$ $(a < b)$.

(b) $\int_1^\infty \mathrm{e}^{-ax} f(x)\mathrm{d}x$ $\;(a > 0)$.

(c) $\int_{-\infty}^\infty \mathrm{e}^{-(ax^2+bx)} f(x)\mathrm{d}x$ $(a > 0)$. {*Hint*: complete the square.}

(d) $\int_0^\infty \frac{\mathrm{e}^{-xt}}{y+t}\mathrm{d}t$, $\; x > 0, \; y > 0$. Is the approximation you get for the integral too small or too large? Explain.

38. (a) Let $w(t)$ be an even weight function on $[a, b]$, $a < b$, $a + b = 0$, i.e., $w(-t) = w(t)$ on $[a, b]$. Show that $(-1)^n \pi_n(-t; w) \equiv \pi_n(t; w)$, i.e., the (monic) nth-degree orthogonal polynomial relative to the weight function w is even [odd] for n even [odd].

(b) Show that the Gauss formula

$$\int_a^b f(t)w(t)\mathrm{d}t = \sum_{\nu=1}^n w_\nu f(t_\nu) + E_n(f)$$

for an even weight function w is symmetric, i.e.,

$$t_{n+1-\nu} = -t_\nu, \quad w_{n+1-\nu} = w_\nu, \qquad \nu = 1, 2, \ldots, n.$$

(c) Let w be the "hat function"

$$w(t) = \begin{cases} 1 + t & \text{if } -1 \le t \le 0, \\ 1 - t & \text{if } 0 \le t \le 1. \end{cases}$$

Obtain the 2-point Gaussian quadrature formula $\int_{-1}^{1} f(t)w(t)\mathrm{d}t = w_1 f(t_1) + w_2 f(t_2) + E_2(f)$ for this weight function w, including an expression for the error term under a suitable regularity assumption on f. {*Hint*: use (a) and (b) to simplify the calculations.}

39. Let $t_k^{(2n)}$ be the nodes, ordered monotonically, of the $(2n)$-point Gauss–Legendre quadrature rule and $w_k^{(2n)}$ the associated weights. Show that, for any $p \in \mathbb{P}_{2n-1}$, one has

$$\int_0^1 t^{-1/2} p(t)\mathrm{d}t = 2\sum_{k=1}^{n} w_k^{(2n)} p([t_k^{(2n)}]^2).$$

40. Let f be a smooth function on $[0, \pi]$. Explain how best to evaluate

$$I_{\alpha,\beta}(f) = \int_0^\pi f(\theta)\left[\cos\frac{1}{2}\theta\right]^\alpha \left[\sin\frac{1}{2}\theta\right]^\beta \mathrm{d}\theta, \quad \alpha > -1, \;\; \beta > -1.$$

41. Let $Q_n f$, $Q_{n^*} f$ be n-point, resp. n^*-point quadrature rules for $If = \int_a^b f(t)w(t)\mathrm{d}t$ and $Q_{n^*} f$ at least twice as accurate as $Q_n f$, i.e.,

$$|Q_{n^*} f - If| \le \frac{1}{2}|Q_n f - If|.$$

Show that the error of $Q_{n^*} f$ then satisfies

$$|Q_{n^*} f - If| \le |Q_n f - Q_{n^*} f|.$$

42. Given a nonnegative weight function w on $[-1, 1]$ and $x > 1$, let

$$(G) \qquad \int_{-1}^{1} f(t)\frac{w(t)}{x^2 - t^2}\mathrm{d}t = \sum_{k=1}^{n} w_k^{\mathrm{G}} f(t_k^{\mathrm{G}}) + E_n^{\mathrm{G}}(f)$$

be the n-point Gaussian quadrature formula for the weight function $\frac{w(t)}{x^2 - t^2}$. (Note that t_k^{G}, w_k^{G} both depend on n and x.) Consider the quadrature rule

$$\int_{-1}^{1} g(t)w(t)\mathrm{d}t = \sum_{k=1}^{n} w_k g(t_k) + E_n(g),$$

where $t_k = t_k^{\mathrm{G}},\ w_k = [x^2 - (t_k^{\mathrm{G}})^2]w_k^{\mathrm{G}}$. Prove:

(a) $E_n(g) = 0$ if $g(t) = \frac{1}{t \pm x}$.
(b) If $n \geq 2$, then $E_n(g) = 0$ whenever g is a polynomial of degree $\leq 2n - 3$.

43. Let ξ_ν, $\nu = 1, 2, \dots, 2n$, be $2n$ preassigned distinct numbers satisfying $-1 < \xi_\nu < 1$, and let w be a positive weight function on [–1,1]. Define $\omega_{2n}(x) = \prod_{\nu=1}^{2n}(1 + \xi_\nu x)$. (Note that ω_{2n} is positive on [–1,1].) Let x_k^{G}, w_k^{G} be the nodes and weights of the n-point Gauss formula for the weight function $w^*(x) = \frac{w(x)}{\omega_{2n}(x)}$:

$$\int_{-1}^{1} p(x)w^*(x)\mathrm{d}x = \sum_{k=1}^{n} w_k^{\mathrm{G}} p(x_k^{\mathrm{G}}), \quad p \in \mathbb{P}_{2n-1}.$$

Define $x_k^* = x_k^{\mathrm{G}}$, $w_k^* = w_k^{\mathrm{G}}\omega_{2n}(x_k^{\mathrm{G}})$. Show that the quadrature formula

$$\int_{-1}^{1} f(x)w(x)\mathrm{d}x = \sum_{k=1}^{n} w_k^* f(x_k^*) + E_n^*(f)$$

is exact for the $2n$ rational functions

$$f(x) = \frac{1}{1 + \xi_\nu x}, \quad \nu = 1, 2, \dots, 2n.$$

44. (a) Prove (3.46).
(b) Prove (3.47). {*Hint*: use the Christoffel–Darboux formula of Chap. 2, Ex. 21(b).}

45. Prove (3.50). {*Hint*: prove, more generally, $(I^{p+1}g)(s) = \int_0^s \frac{(s-t)^p}{p!} g(t)\mathrm{d}t$.}

46. (a) Use the method of undetermined coefficients to obtain an integration rule (having degree of exactness $d = 2$) of the form

$$\int_0^1 y(s)\mathrm{d}s \approx ay(0) + by(1) - c[y'(1) - y'(0)].$$

(b) Transform the rule in (a) into one appropriate for approximating $\int_x^{x+h} f(t)\mathrm{d}t$.
(c) Obtain a composite integration rule based on the formula in (b) for approximating $\int_a^b f(t)\mathrm{d}t$. Interpret the result.

47. Determine the quadrature formula of the type

$$\int_{-1}^{1} f(t)\mathrm{d}t = \alpha_{-1}\int_{-1}^{-1/2} f(t)\mathrm{d}t + \alpha_0 f(0) + \alpha_1 \int_{1/2}^{1} f(t)\mathrm{d}t + E(f)$$

having maximum degree of exactness d. What is the value of d ?

48. (a) Determine the quadratic spline $s_2(x)$ on $[-1, 1]$ with a single knot at $x = 0$ and such that $s_2(x) \equiv 0$ on $[-1, 0]$ and $s_2(1) = 1$.
    (b) Consider a function $s(x)$ of the form

$$s(x) = c_0 + c_1x + c_2x^2 + c_3s_2(x), \quad c_i = \text{const},$$

where $s_2(x)$ is as defined in (a). What kind of function is s? Determine s such that

$$s(-1) = f_{-1}, \ \ s(0) = f_0, \ \ s'(0) = f'_0, \ \ s(1) = f_1,$$

where $f_{-1} = f(-1), \ f_0 = f(0), \ f'_0 = f'(0), \ f_1 = f(1)$ for some function f on $[-1, 1]$.
    (c) What quadrature rule does one obtain if one approximates $\int_{-1}^{1} f(x)\mathrm{d}x$ by $\int_{-1}^{1} s(x)\mathrm{d}x$, with s as obtained in (b)?
49. Prove that the condition (3.68) does not depend on the choice of the basis $\varphi_1, \varphi_2, \dots, \varphi_n$.
50. Let E be a linear functional that annihilates all polynomials of degree $d \geq 0$. Show that the Peano kernel $K_r(t)$, $r \leq d$, of E vanishes for $t \notin [a, b]$, where $[a, b]$ is the interval of function values referenced by E.
51. Show that a linear functional E satisfying $Ef = e_{r+1}f^{(r+1)}(\bar{t})$, $\bar{t} \in [a, b]$, $e_{r+1} \neq 0$, for any $f \in C^{r+1}[a, b]$, is necessarily definite of order r if it has a continuous Peano kernel K_r.
52. Let E be a linear functional that annihilates all polynomials of degree d. Show that none of the Peano kernels $K_0, K_1, \dots, K_{d-1}$ of E can be definite.
53. Suppose in (3.61) the function f is known to be only once continuously differentiable, i.e., $f \in C^1[0, 1]$.

    (a) Derive the appropriate Peano representation of the error functional Ef.
    (b) Obtain an estimate of the form $|Ef| \leq c_0\|f'\|_\infty$.

54. Assume, in Simpson's rule

$$\int_{-1}^{1} f(x)\mathrm{d}x = \frac{1}{3}[f(-1) + 4f(0) + f(1)] + E^{\mathrm{S}}(f),$$

that f is only of class $C^2[-1, 1]$ instead of class $C^4[-1, 1]$ as normally assumed.

    (a) Find an error estimate of the type

$$|E^S(f)| \leq \text{const} \cdot \|f''\|_\infty, \ \ \|f''\|_\infty = \max_{-1 \leq x \leq 1} |f''(x)|.$$

{*Hint*: apply the appropriate Peano representation of $E^{\mathrm{S}}(f)$.}
    (b) Transform the result in (a) to obtain Simpson's formula, with remainder estimate, for the integral

$$\int_{c-h}^{c+h} g(t)\mathrm{d}t, \quad g \in C^2[c-h, c+h], \quad h > 0.$$

(c) How does the estimate in (a) compare with the analogous error estimate for two applications of the trapezoidal rule,

$$\int_{-1}^{1} f(x)\mathrm{d}x = \frac{1}{2}\,[f(-1) + 2f(0) + f(1)] + E_2^{\mathrm{T}}(f)?$$

55. Determine the Peano kernel $K_1(t)$ on $[a, b]$ of the error functional for the composite trapezoidal rule over the interval $[a, b]$ subdivided into n subintervals of equal length.

56. Consider the trapezoidal formula "with mean values,"

$$\int_0^1 f(x)\mathrm{d}x = \frac{1}{2}\left[\frac{1}{\varepsilon}\int_0^{\varepsilon} f(x)\mathrm{d}x + \frac{1}{\varepsilon}\int_{1-\varepsilon}^{1} f(x)\mathrm{d}x\right] + E(f), \quad 0 < \varepsilon < \frac{1}{2}.$$

(a) Determine the degree of exactness of this formula.

(b) Express the remainder $E(f)$ by means of its Peano kernel K_1 in terms of f'', assuming $f \in C^2[0, 1]$.

(c) Show that the Peano kernel K_1 is definite, and thus express the remainder in the form $E(f) = e_2 f''(\tau)$, $0 < \tau < 1$.

(d) Consider (and explain) the limit cases $\varepsilon \downarrow 0$ and $\varepsilon \to \frac{1}{2}$.

57. (a) Use the method of undetermined coefficients to construct a quadrature formula of the type

$$\int_0^1 f(x)\mathrm{d}x = af(0) + bf(1) + cf''(\gamma) + E(f)$$

having maximum degree of exactness d, the variables being a, b, c, and γ.

(b) Show that the Peano kernel K_d of the error functional E of the formula obtained in (a) is definite, and hence express the remainder in the form $E(f) = e_{d+1} f^{(d+1)}(\xi)$, $0 < \xi < 1$.

58. (a) Use the method of undetermined coefficients to construct a quadrature formula of the type

$$\int_0^1 f(x)\mathrm{d}x = -\alpha f'(0) + \beta f\left(\frac{1}{2}\right) + \alpha f'(1) + E(f)$$

that has maximum degree of exactness.

(b) What is the precise degree of exactness of the formula obtained in (a)?

(c) Use the Peano kernel of the error functional E to express $E(f)$ in terms of the appropriate derivative of f reflecting the result of (b).

(d) Transform the formula in (a) to one that is appropriate to evaluate $\int_c^{c+h} g(t)\mathrm{d}t$, and then obtain the corresponding composite formula for $\int_a^b g(t)\mathrm{d}t$, using n subintervals of equal length, and derive an error term. Interpret your result.

59. Consider a quadrature rule of the form

$$\int_0^1 x^\alpha f(x)\mathrm{d}x \approx Af(0) + B\int_0^1 f(x)\mathrm{d}x, \;\; \alpha > -1, \;\; \alpha \neq 0.$$

(a) Determine A and B such that the formula has degree of exactness $d = 1$.

(b) Let $E(f)$ be the error functional of the rule determined in (a). Show that the Peano kernel $K_1(t) = E_{(x)}((x-t)_+)$ of E is positive definite if $\alpha > 0$, and negative definite if $\alpha < 0$.

(c) Based on the result of (b), determine the constant e_2 in $E(f) = e_2 f''(\xi)$, $0 < \xi < 1$.

60. (a) Consider a quadrature formula of the type

$$(^*) \qquad \int_0^1 f(x)\mathrm{d}x = \alpha f(x_1) + \beta[f(1) - f(0)] + E(f)$$

and determine α, β, x_1 such that the degree of exactness is as large as possible. What is the maximum degree attainable?

(b) Use interpolation theory to obtain a bound on $|E(f)|$ in terms of $\|f^{(r)}\|_\infty = \max_{0\le x\le 1}|f^{(r)}(x)|$ for some suitable r.

(c) Adapt (*), including the bound on $|E(f)|$, to an integral of the form $\int_c^{c+h} f(t)\mathrm{d}t$, where c is some constant and $h > 0$.

(d) Apply the result of (c) to develop a composite quadrature rule for $\int_a^b f(t)\mathrm{d}t$ by subdividing $[a,b]$ into n subintervals of equal length $h = \frac{b-a}{n}$. Find a bound for the total error.

61. Construct a quadrature rule

$$\int_0^1 x^\alpha f(x)\mathrm{d}x \approx a_1\int_0^1 f(x)\mathrm{d}x + a_2\int_0^1 xf(x)\mathrm{d}x, \;\; 0 < \alpha < 1,$$

(a) which is exact for all polynomials p of degree ≤ 1;

(b) which is exact for all $f(x) = x^{1/2}p(x)$, $p \in \mathbb{P}_1$.

62. Let

$$a = x_0 < x_1 < x_2 < \cdots < x_{n-1} < x_n = b, \quad x_k = a + kh, \;\; h = \frac{b-a}{n},$$

be a subdivision of $[a,b]$ into n equal subintervals.

(a) Derive an elementary quadrature formula for the integral $\int_{x_k}^{x_{k+1}} f(x)\,\mathrm{d}x$, including a remainder term, by approximating f by the cubic Hermite interpolation polynomial $p_3(f;\, x_k,\, x_k,\, x_{k+1},\, x_{k+1};\, x)$ and then integrating over $[x_k,\, x_{k+1}]$. Interpret the result.

(b) Develop the formula obtained in (a) into a composite quadrature rule, with remainder term, for the integral $\int_a^b f(x)\mathrm{d}x$.

63. (a) Given a function $g(x, y)$ on the unit square $0 \le x \le 1$, $0 \le y \le 1$, determine a "bilinear polynomial" $p(x, y) = a + bx + cy + dxy$ such that p has the same values as g at the four corners of the square.

(b) Use (a) to obtain a cubature formula for $\int_0^1 \int_0^1 g(x, y)\mathrm{d}x\mathrm{d}y$ that involves the values of g at the four corners of the unit square. What rule does this reduce to if g is a function of x only (i.e., does not depend on y)?

(c) Use (b) to find a "composite cubature rule" for $\int_0^1 \int_0^1 g(x, y)\mathrm{d}x\mathrm{d}y$ involving the values $g_{i,j} = g(ih, jh)$, $i, j = 0, 1, \dots, n$, where $h = 1/n$.

64. (a) Let $d_1(h) = (f(h) - f(0))/h$, $h > 0$, be the difference quotient of f at the origin. Describe how the extrapolation method based on a suitable expansion of $d_1(h)$ can be used to approximate $f'(0)$ to successively higher accuracy.

(b) Develop a similar method for calculating $f''(0)$, based on $d_2(h) = [f(h) - 2f(0) + f(-h)]/h^2$.

Machine Assignments

1. Let $f(x) = \frac{1}{1-\pi x}$ and $f_i = f(ih)$, $i = -2, -1, 0, 1, 2$. In terms of the four backward differences

$$\nabla f_1 = f_1 - f_0,\ \ \nabla^2 f_1 = f_1 - 2f_0 + f_{-1},$$
$$\nabla^3 f_2 = f_2 - 3f_1 + 3f_0 - f_{-1},\ \ \nabla^4 f_2 = f_2 - 4f_1 + 6f_0 - 4f_{-1} + f_{-2},$$

define

$$e_n(h) = f^{(n)}(0) - \frac{1}{h^n}\,\nabla^n f_{\lfloor \frac{n+1}{2} \rfloor},\quad n = 1, 2, 3, 4.$$

Try to determine the order of convergence of $e_n(h)$ as $h \to 0$ by printing, for $n = 1, \dots, 4$,

$$e_n(h_k)\ \text{ and }\ r_k := \frac{e_n(h_k)}{e_n(h_{k-1})},\quad k = 1, 2, \dots, 10,$$

where $h_k = \dfrac{1}{4} \cdot 2^{-k}$, $k \ge 0$. Comment on the results.

2. Let

$$f(z) = \tan z, \quad |z| < \frac{1}{2}\pi.$$

(a) Express

$$y_m(\theta) = \mathrm{Re}\,\{\mathrm{e}^{-\mathrm{i}m\theta} f(x_0 + r\mathrm{e}^{\mathrm{i}\theta})\}, \;\; 0 < r < \frac{1}{2}\pi, \;\; m = 1, 2, 3, \ldots$$

explicitly as a function of θ. {*Hint*: use Euler's identities $\sin z = (\mathrm{e}^{\mathrm{i}z} - \mathrm{e}^{-iz})/(2i)$, $\cos z = (\mathrm{e}^{\mathrm{i}z} + \mathrm{e}^{-iz})/2$, valid for arbitrary complex z.}

(b) Obtain the analogue to (3.19) for the mth derivative and thus write $f^{(m)}(x_0)$ as a definite integral over $[0, 2\pi]$.

(c) Use Matlab to compute $f^{(m)}(0)$ for $m = 1 : 5$ using the integral in (b) in conjunction with the composite trapezoidal rule (cf. Sect. 3.2.1) relative to a subdivision of $[0, 2\pi]$ into n subintervals. Use $r = \frac{1}{12}\rho\pi$, $\rho = 1, 2, 3, 4, 5$, and $n = 5 : 5 : 50$. For each m print a table whose columns contain r, n, the trapezoidal approximation $t_n^{(m)}$, and the (absolute) error, in this order. Comment on the results; in particular, try to explain the convergence behavior as r increases and the difference in behavior for n even and n odd. {*Hint*: prepare plots of the integrand; you may use the Matlab routine `spline` for cubic spline interpolation to do this.}

(d) Do the same as (c), but for $f^{(m)}(\frac{7}{16}\pi)$ and $r = \frac{1}{32}\pi$.

(e) Write and run a Matlab program for approximating $f^{(m)}(0)$, $m = 1 : 5$, by central difference formulae with steps $h = 1, \frac{1}{5}, \frac{1}{25}, \frac{1}{125}, \frac{1}{625}$. Comment on the results.

(f) Do the same as (e), but for $f^{(m)}(\frac{7}{16}\pi)$ and $h = \frac{1}{32}\pi, \frac{1}{160}\pi, \frac{1}{800}\pi, \frac{1}{4000}\pi$.

3. Given n distinct real nodes $x_k = x_k^{(n)}$, the interpolatory quadrature rule

$$(\mathrm{WNC}_n) \qquad \int_a^b f(x)w(x)\mathrm{d}x = \sum_{k=1}^n w_k^{(n)} f(x_k^{(n)}), \quad \text{all } f \in \mathbb{P}_{n-1},$$

is called a weighted (by the weight function w) Newton–Cotes formula (cf. Sect. 3.2.2). The weights $w_k^{(n)}$ can be generated by n_g-point Gauss integration, $n_g = \lfloor (n+1)/2 \rfloor$, of the elementary Lagrange interpolation polynomials (see (3.39)),

$$(\mathrm{W}_n) \qquad w_k^{(n)} = \int_a^b \ell_k(x)w(x)\mathrm{d}x, \quad \ell_k(x) = \prod_{\substack{\ell=1 \\ \ell \neq k}}^n \frac{x - x_\ell}{x_k - x_\ell}.$$

This is implemented in the `OPQ` routine `NewtCotes.m` downloadable from the web site mentioned in MA 4. For reasons of economy, it uses the barycentric form (see Chap. 2, Sect. 2.2.5, (2.106)) of the Lagrange polynomials and the algorithm (*ibid.*, (2.108)) to compute the auxiliary quantities $\lambda_i^{(k)}$. Use the routine `NewtCotes.m` to explore the positivity of (WNC$_n$), i.e., $w_k^{(n)} > 0$, $k = 1, 2, \ldots, n$.

(a) Write a Matlab function `y=posNC(n,ab,ab0,eps0)`, which checks the positivity of the n-point Newton–Cotes formula with the abscissae being the zeros of the Jacobi polynomial $P_n^{(\alpha,\beta)}$ with parameters α, β, and using integration relative to the Jacobi weight function $w = w^{(\alpha_0,\beta_0)}$ with other parameters α_0, β_0. The former are selected by the `OPQ` routine `ab=r_jacobi(n,alpha,beta)`, from which the Jacobi abscissae can be obtained via the `OPQ` function `xw=gauss(n,ab)` as the first column of the $n \times 2$ array `xw`. The weight function w is provided by the routine `ab0=r_jacobi(floor((n+1)/2),alpha0,beta0)`, which allows us to generate the required n_g-point Gaussian quadrature rule by the routine `xw=gauss(ng,ab0)`. The input parameter `eps0`, needed in the routine `NewtCotes.m`, is a number close to, but larger than, the machine precision `eps`, for example $\varepsilon_0 = 0.5 \times 10^{-14}$. Arrange the output parameter `y` to have the value 1 if all n weights of the Newton–Cotes formula (WNC$_n$) are positive, and the value 0 otherwise.

Use your routine for all $n \le N = 50$, $\alpha_0 = \beta_0 = 0, -1/2, 1/2$, and $\alpha = -1 + h : h : \alpha^+$, $\beta = \alpha : h : \beta^+$, where $\alpha^+ = \beta^+ = 3, 1.5, 4$ and $h = 0.05, 0.025, 0.05$ for the three values of α_0, β_0, respectively. Prepare plots in which a red plus sign is placed at the point (α, β) of the (α, β)-plane if positivity holds for all $n \le N$, and a blue dot otherwise. Explain why it suffices to consider only $\beta \ge \alpha$. {*Hint*: use the reflection formula $P_n^{(\beta,\alpha)}(x) = (-1)^n P_n^{(\alpha,\beta)}(-x)$ for Jacobi polynomials.} In a second set of plots show the exact upper boundary of the positivity domain created in the first plots; compute it by a bisection-type method (cf. Chap. 4, Sect. 4.3.1). (Running the programs for $N = 50$ may take a while. You may want to experiment with smaller values of N to see how the positivity domains vary.)

(b) The plots in (a) suggest that n-point Newton–Cotes formulae are positive for all $n \le N = 50$ on the line $0 \le \beta = \alpha$ up to a point $\alpha = \alpha_{\max}$. Use the same bisection-type method as in (a) to determine $\alpha_{\max}$ for the three values of α_0, β_0 and for $N = 20, 50, 100$ in each case.

(c) Repeat (a) with $\alpha = \beta = 0, -1/2, 1/2$ (Gauss–Legendre and Chebyshev abscissae of the first and second kinds) and $\alpha_0 = -0.95 : 0.05 : \alpha_0^+$, $\beta_0 = \alpha_0 : 0.05 : \beta_0^+$, where $\alpha_0^+ = \beta_0^+ = 3.5, 3, 4$.

(d) Repeat (a) with $\alpha^+ = \beta^+ = 6, 4, 6$, but for weighted $(n+2)$-point Newton–Cotes formulae that contain as nodes the points ±1 in addition to the n Jacobi abscissae.

(e) The plots in (d) suggest that the closed $(n+2)$-point Newton–Cotes formulae are positive for all $n \le N = 50$ on some line $\alpha_{\min} < \alpha = \beta < \alpha_{\max}$. Determine $\alpha_{\min}$ and $\alpha_{\max}$ similarly as $\alpha_{\max}$ in (b).

(f) Repeat (c) for the weighted closed $(n+2)$-point Newton–Cotes formula of (d) with $\alpha_0 = -1 + h : h : \alpha_0^+$, $\beta_0 = \alpha_0 : h : \beta_0^+$, where $\alpha_0^+ = \beta_0^+ = 0.5, -0.2, 1$ and $h = 0.01, 0.01, 0.02$ for the three values of α, β.

4. Below are a number of suggestions as to how the following integrals may be computed,

$$I_c = \int_0^1 \frac{\cos x}{\sqrt{x}} \mathrm{d}x, \quad I_s = \int_0^1 \frac{\sin x}{\sqrt{x}} \mathrm{d}x.$$

(a) Use the composite trapezoidal rule with n intervals of equal length $h = 1/n$, "ignoring" the singularity at $x = 0$ (i.e., arbitrarily using zero as the value of the integrand at $x = 0$).
(b) Use the composite trapezoidal rule over the interval $[h, 1]$ with $n-1$ intervals of length $h = 1/n$ in combination with a weighted Newton–Cotes rule with weight function $w(x) = x^{-1/2}$ over the interval $[0, h]$. {Adapt the formula (3.43) to the interval $[0, h]$.}
(c) Make the change of variables $x = t^2$ and apply the composite trapezoidal rule to the resulting integrals.
(d) Use Gauss–Legendre quadrature on the integrals obtained in (c).
(e) Use Gauss–Jacobi quadrature with parameters $\alpha = 0$ and $\beta = -\frac{1}{2}$ directly on the integrals I_c and I_s.

{As a *point of information*, $I_c = \sqrt{2\pi}\, C\left(\sqrt{\frac{2}{\pi}}\right) = 1.809048475800\ldots$, $I_s = \sqrt{2\pi}\, S\left(\sqrt{\frac{2}{\pi}}\right) = 0.620536603446\ldots$, where $C(x)$, $S(x)$ are the Fresnel integrals.}

Implement and run the proposed methods for $n = 100 : 100 : 1000$ in (a) and (b), for $n = 20 : 20 : 200$ in (c), and for $n = 1 : 10$ in (d) and (e). Try to explain the results you obtain. {To get the required subroutines for Gaussian quadrature, download the `OPQ` routines `r_jacobi.m` and `gauss.m` from the web site `http://www.cs.purdue.edu/archives/2002/wxg/codes/OPQ.html`.}

5. For a natural number p let

$$I_p = \int_0^1 (1-t)^p f(t) \mathrm{d}t$$

be (except for the factor $1/p!$) the pth iterated integral of f; cf. (3.50). Compare the composite trapezoidal rule based on n subintervals with the n-point Gauss–Jacobi rule on $[0, 1]$ with parameters $\alpha = p$ and $\beta = 0$. Take, for example, $f(t) = \tan t$ and $p = 5 : 5 : 20$, and let $n = 10 : 10 : 50$ in the case of the trapezoidal rule, and $n = 2 : 2 : 10$ for the Gauss rule. {See MA 4 for instructions on how to download routines for generating the Gaussian quadrature rules.}

6. (a) Let $h_k = (b-a)/2^k$, $k = 0, 1, 2, \ldots$. Denote by

$$T_{h_k}(f) = h_k\left(\frac{1}{2}f(a) + \sum_{r=1}^{2^k-1} f(a + rh_k) + \frac{1}{2}f(b)\right)$$

the composite trapezoidal rule and by

$$M_{h_k}(f) = h_k \sum_{r=1}^{2^k} f\left(a + \left(r - \frac{1}{2}\right)h_k\right)$$

the composite midpoint rule, both relative to a subdivision of $[a, b]$ into 2^k subintervals. Show that the first column $T_{k,0}$ of the Romberg array $\{T_{k,m}\}$ can be generated recursively as follows:

$$\begin{aligned} T_{0,0} &= \frac{b-a}{2}\,[f(a) + f(b)], \\ T_{k+1,0} &= \frac{1}{2}\,[T_{k,0} + M_{h_k}(f)], \quad k = 0, 1, 2, \ldots. \end{aligned}$$

(b) Write a Matlab function for computing $\int_a^b f(x)\mathrm{d}x$ by the Romberg integration scheme, with $h_k = (b-a)/2^k$, $k = 0, 1, \ldots, n-1$.
Formal parameters: `a, b, n`; include f as a subfunction.
Output variable: the $n \times n$ Romberg array `T`.
Order of computation: Generate `T` row by row; generate the trapezoidal sums recursively as in part (a).
Program size: Keep it down to about 20 lines of Matlab code.
Output: $\mathtt{T}_{k,0}$, $\mathtt{T}_{k,k}$, $k = 0, 1, \ldots, n-1$.

(c) Call your subroutine (with $n = 10$) to approximate the following integrals.

1. $\displaystyle\int_1^2 \frac{\mathrm{e}^x}{x}\,\mathrm{d}x$ ("exponential integral")
2. $\displaystyle\int_0^1 \frac{\sin x}{x}\,\mathrm{d}x$ ("sine integral")
3. $\displaystyle\frac{1}{\pi}\int_0^{\pi} \cos(yx)\mathrm{d}x, \quad y = 1.7$
4. $\displaystyle\frac{1}{\pi}\int_0^{\pi} \cos(y \sin x)\mathrm{d}x, \quad y = 1.7$
5. $\displaystyle\int_0^1 \sqrt{1-x^2}\,\mathrm{d}x$

6. $\displaystyle\int_0^2 f(x)\mathrm{d}x, \quad f(x) = \begin{cases} x, & 0 \le x \le \sqrt{2}, \\ \dfrac{\sqrt{2}}{2-\sqrt{2}}\,(2-x), & \sqrt{2} \le x \le 2 \end{cases}$

7. $\displaystyle\int_0^2 f(x)\mathrm{d}x, \quad f(x) = \begin{cases} x, & 0 \le x \le \dfrac{3}{4}, \\ \dfrac{3}{5}\,(2-x), & \dfrac{3}{4} \le x \le 2 \end{cases}$

(d) Comment on the behavior of the Romberg scheme in each of the seven cases in part (c).

Selected Solutions to Exercises

3. (a) The kth derivative of $\prod_{i=1}^{n}(x - x_i)$ is a sum of products, each containing $n - k$ factors $x - x_i$. Thus, if $x = x_0$, each term of the sum is $O(H^{n-k})$, hence also the sum itself.

   (b) By Lagrange interpolation, we have

$$f(x) = p_n(f;x) + R_n(x),$$

where $p_n(f;x) = p_n(f;x_0,x_1,\dots,x_n;x)$, in Newton's form, is given by

$$\begin{aligned} p_n(f;x) =& f_0 + (x-x_0)[x_0,x_1]f + \cdots \\ & + \prod_{i=0}^{n-1}(x-x_i)\cdot[x_0,x_1,\dots,x_n]f, \end{aligned}$$

and

$$R_n(x) = \prod_{i=0}^{n}(x-x_i)\frac{f^{(n+1)}(\xi(x))}{(n+1)!},$$

assuming $f \in C^{n+1}$ in the span I of $x_0,x_1,\dots,x_n,x$. Differentiating n times at $x = x_0$ gives

$$(*) \qquad f^{(n)}(x_0) = \frac{\mathrm{d}^n}{\mathrm{d}x^n}\,p_n(f;x)\bigg|_{x=x_0} + R_n^{(n)}(x_0).$$

Clearly,

$$(**) \qquad \frac{\mathrm{d}^n}{\mathrm{d}x^n}\,p_n(f;x)\bigg|_{x=x_0} = n!\,[x_0,x_1,\dots,x_n].$$

Assuming that $f \in C^{2n+1}(I)$, we can apply Leibniz's rule of differentiating n times the product

$$R_n(x) = (x - x_0) \cdot \left\{ \prod_{i=1}^{n} (x - x_i) \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \right\} .$$

If we evaluate the result at $x = x_0$, we get

$$R_n^{(n)}(x_0) = n \frac{\mathrm{d}^{n-1}}{\mathrm{d}x^{n-1}} \left\{ \prod_{i=1}^{n} (x - x_i) \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \right\} \Bigg|_{x=x_0} .$$

Using Leibniz's rule again, and the result of (a), we obtain

$$R_n^{(n)}(x_0) = n \frac{\mathrm{d}^{n-1}}{\mathrm{d}x^{n-1}} \prod_{i=1}^{n} (x - x_i) \Bigg|_{x=x_0} \cdot \frac{f^{(n+1)}(\xi_0)}{(n+1)!} + O(H^2),$$

where $\xi_0 = \xi(x_0)$. Since

$$\prod_{i=1}^{n} (x - x_i) = x^n - \sigma_1 x^{n-1} + \cdots , \quad \sigma_1 = \sum_{i=1}^{n} x_i ,$$

we have

$$\frac{\mathrm{d}^{n-1}}{\mathrm{d}x^{n-1}} \prod_{i=1}^{n} (x - x_i) \Bigg|_{x=x_0} = n!\, x_0 - (n-1)!\, \sigma_1 = (n-1)!\, (n x_0 - \sigma_1),$$

so that

$$(***) \qquad R_n^{(n)}(x_0) = n!\, (n x_0 - \sigma_1) \frac{f^{(n+1)}(\xi_0)}{(n+1)!} + O(H^2).$$

Since

$$n x_0 - \sigma_1 = \sum_{i=1}^{n} (x_0 - x_i) = O(H) \text{ if } n x_0 \neq \sigma_1 ,$$

the assertion follows by combining (*)–(***).

(c) One has

$$[x_0, x_1, \ldots, x_n] f = \frac{1}{n! h^n} \Delta^n f_0 = \frac{1}{n! h^n} \nabla^n f_n .$$

This is proved by induction on n. We show it for the forward difference only; for the backward difference the proof is analogous. Since the claim is obviously true for $n = 1$, suppose it is true for some n. Then by definition of divided differences, and the induction hypothesis,

$$\begin{aligned}[x_0, x_1, \ldots, x_n, x_{n+1}]f &= \frac{[x_1, \ldots, x_{n+1}]f - [x_0, \ldots, x_n]f}{x_{n+1} - x_0} \\ &= \frac{1}{n!h^n}\,\frac{\Delta^n f_1 - \Delta^n f_0}{(n+1)h} \\ &= \frac{1}{(n+1)!h^{n+1}}\,\Delta^{n+1} f_0,\end{aligned}$$

which is the claim for $n + 1$. Therefore, since $\frac{1}{n}\sum_{i=1}^{n} x_i = x_0 + (n+1)h/2 \neq x_0$, we get

$$f^{(n)}(x_0) = \frac{1}{h^n}\,\Delta^n f_0 + O(h) = \frac{1}{h^n}\nabla^n f_n + O(h).$$

14. With a slight difference in notation, one has from (2.129) of Chap. 2 that

$$\begin{aligned}&hB_0(x) = x_1 - x,\ x_0 \le x \le x_1; \\ &hB_k(x) = \begin{cases} x - x_{k-1} & \text{if } x_{k-1} \le x \le x_k, \\ x_{k+1} - x & \text{if } x_k \le x \le x_{k+1}, \end{cases} \quad k = 1, 2, \ldots, N-1; \\ &hB_N(x) = x - x_{N-1},\ x_{N-1} \le x \le x_N.\end{aligned}$$

From this, one gets, with obvious changes of variables,

$$\begin{aligned}&\int_{x_0}^{x_1} B_0(x)\mathrm{e}^{-\mathrm{i}mx}\mathrm{d}x + \int_{x_{N-1}}^{x_N} B_N(x)\mathrm{e}^{-\mathrm{i}mx}\mathrm{d}x \\ &\quad = h\int_0^1 (1-t)\mathrm{e}^{-\mathrm{i}mth}\mathrm{d}t + h\int_0^1 (1-t)\mathrm{e}^{-\mathrm{i}m(2\pi - th)}\mathrm{d}t \\ &\quad = 2h\int_0^1 (1-t)\cos(mth)\mathrm{d}t,\end{aligned}$$

and, for $k = 1, 2, \ldots, N-1$,

$$\begin{aligned}&\int_{x_{k-1}}^{x_{k+1}} B_k(x)\mathrm{e}^{-\mathrm{i}mx}\mathrm{d}x \\ &\quad = \frac{1}{h}\int_{x_{k-1}}^{x_k} (x - x_{k-1})\mathrm{e}^{-\mathrm{i}mx}\mathrm{d}x + \frac{1}{h}\int_{x_k}^{x_{k+1}} (x_{k+1} - x)\mathrm{e}^{-\mathrm{i}mx}\mathrm{d}x\end{aligned}$$

$$= h\int_0^1 (1-t)\mathrm{e}^{-\mathrm{i}m(x_k-th)}\mathrm{d}t + h\int_0^1 (1-t)\mathrm{e}^{-\mathrm{i}m(x_k+th)}\mathrm{d}t$$
$$= 2h\mathrm{e}^{-\mathrm{i}mx_k}\int_0^1 (1-t)\cos(mth)\mathrm{d}t.$$

Since (cf. Chap. 2, (2.132)), with $f_k = f(x_k)$,

$$s_1(f;x) = f_0B_0(x) + \sum_{k=1}^{N-1} f_kB_k(x) + f_NB_N(x),$$

and $f_N = f_0$, we get

$$\begin{aligned}
&\frac{1}{2\pi}\int_0^{2\pi} s_1(f;x)\mathrm{e}^{-\mathrm{i}mx}\mathrm{d}x\\
&\quad= \frac{1}{2\pi}f_0\left(\int_{x_0}^{x_1} B_0(x)\mathrm{e}^{-\mathrm{i}mx}\mathrm{d}x + \int_{x_{N-1}}^{x_N} B_N(x)\mathrm{e}^{-\mathrm{i}mx}\mathrm{d}x\right)\\
&\qquad+ \frac{1}{2\pi}\sum_{k=1}^{N-1} f_k\int_{x_{k-1}}^{x_{k+1}} B_k(x)\mathrm{e}^{-\mathrm{i}mx}\mathrm{d}x\\
&\quad= \frac{2h}{2\pi}\sum_{k=0}^{N-1} f_k\mathrm{e}^{-\mathrm{i}mx_k}\int_0^1 (1-t)\cos(mth)\mathrm{d}t\\
&\quad= \tau_m\cdot\frac{1}{N}\sum_{k=0}^{N-1} f_k\mathrm{e}^{-\mathrm{i}mx_k},
\end{aligned}$$

where

$$\tau_m = 2\int_0^1 (1-t)\cos(mth)\mathrm{d}t.$$

Integration by parts yields

$$\tau_m = \frac{2}{(mh)^2}(1-\cos mh) = \frac{\sin^2\left(\frac{1}{2}mh\right)}{\left(\frac{1}{2}mh\right)^2},$$

hence,

$$\tau_m = \left[\frac{\sin(m\pi/N)}{m\pi/N}\right]^2.$$

The factor τ_m, which modifies the composite trapezoidal sum, may be interpreted as an "attenuation factor," since as $m \to \infty$ it tends to zero, whereas the composite trapezoidal sums, being periodic in m with period N, cycle through N values. For a theory of attenuation factors in Fourier analysis, see Gautschi [1971/1972].

22. (a) Since the Hermite interpolant reproduces f exactly if $f \in \mathbb{P}_{n+2}$, the formula has degree of exactness $d = n + 2$.

(b) There are $2n+3$ free parameters available to make the formula exact for the the first $2n+3$ powers x^r, $r = 0, 1, \dots, 2n+2$. Thus, the maximum degree of exactness is expected to be $d = 2n + 2$.

(c) Letting $\omega_n(x) = \prod_{k=1}^{n}(x - x_k)$, we have, for any $p \in \mathbb{P}_{n-1}$,

$$\int_0^1 \omega_n(x)p(x)x^2(1-x)\mathrm{d}x = 0,$$

since the integrand is a polynomial of degree $\leq 2n + 2$, and hence the integral is equal to the quadrature sum. The latter, however, vanishes, since the integrand vanishes together with its first derivative at $x = 0$ and also vanishes at $x = 1$, and $\omega_n(x_k) = 0$ for $k = 1, 2, \dots, n$. This shows that $\omega_n(\cdot) = \pi_n(\cdot\,; x^2(1-x)\mathrm{d}x)$, the Jacobi polynomial relative to the interval $[0, 1]$, with parameters $\alpha = 1,\ \beta = 2$.

(d) Let $f(x) \in \mathbb{P}_{2n+2}$. Divide $f(x)$ by $x^2(1-x)\omega_n(x)$:

$$f(x) = x^2(1-x)\omega_n(x)q(x) + r(x), \quad q \in \mathbb{P}_{n-1}, \quad r \in \mathbb{P}_{n+2}.$$

Then

$$\int_0^1 f(x)\mathrm{d}x = \int_0^1 \omega_n(x)q(x)x^2(1-x)\mathrm{d}x + \int_0^1 r(x)\mathrm{d}x.$$

By the orthogonality assumption and the fact that $q \in \mathbb{P}_{n-1}$, the first integral vanishes. For the second integral, since $r \in \mathbb{P}_{n+2}$ and the quadrature formula is Hermite interpolatory, we have

$$\int_0^1 r(x)\mathrm{d}x = a_0 r(0) + a_1 r'(0) + \sum_{k=1}^{n} w_k r(x_k) + b_0 r(1),$$

and using

$$r(0) = f(0), \;\; r'(0) = f'(0), \;\; r(1) = f(1),$$
$$r(x_k) = f(x_k) - x_k^2(1-x_k)\omega_n(x_k)q(x_k) = f(x_k),$$

since again, $\omega_n(x_k) = 0$, we get

$$\int_0^1 f(x)\mathrm{d}x = a_0 f(0) + a_1 f'(0) + \sum_{k=1}^{n} w_k f(x_k) + b_0 f(1).$$

This shows that the formula has degree of exactness $2n + 2$.

54. (a) Putting

$$E(f) = \int_{-1}^{1} f(x)\mathrm{d}x - \frac{1}{3}[f(-1) + 4f(0) + f(1)],$$

one has, if $f \in C^2[-1, 1]$, the Peano representation

$$E(f) = \int_{-1}^{1} K_1(t) f''(t)\mathrm{d}t,$$

where, for $-1 \leq t \leq 1$,

$$\begin{aligned}
K_1(t) &= E_{(x)}((x-t)_+) \\
&= \int_t^1 (x-t)\mathrm{d}x - \frac{1}{3}\left[(-1-t)_+ + 4(-t)_+ + (1-t)_+\right] \\
&= \frac{1}{2}(1-t)^2 - \frac{1}{3}\cdot 0 - \frac{4}{3}\left\{\begin{matrix} 0 & \text{if } t \geq 0 \\ -t & \text{if } t < 0 \end{matrix}\right\} - \frac{1}{3}(1-t) \\
&= \begin{cases} \dfrac{1}{6}(1-t)(1-3t) & \text{if } t \geq 0, \\ \dfrac{1}{6}(1+t)(1+3t) & \text{if } t < 0. \end{cases}
\end{aligned}$$

It can be seen that $K_1(-t) = K_1(t)$ and $K(\frac{1}{3}) = K(1) = 0$ (see figure on the next page). Therefore, $|E(f)| \leq \|f''\|_\infty \int_{-1}^{1} |K_1(t)|\mathrm{d}t$.

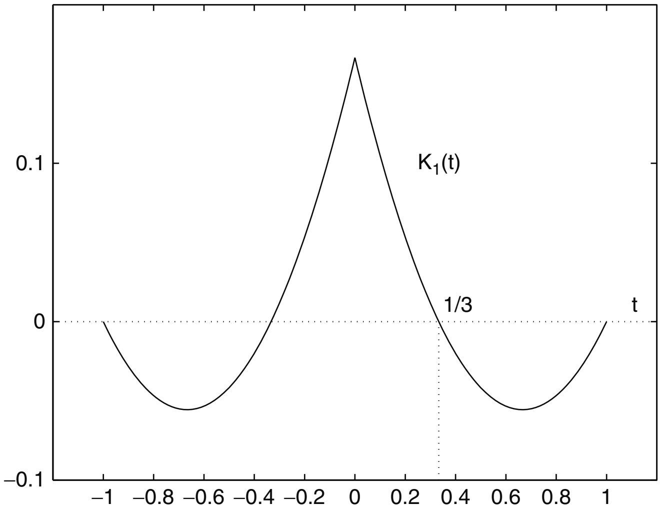
Now,

$$\int_{-1}^{1} |K_1(t)|\mathrm{d}t = 2\int_0^1 |K_1(t)|\mathrm{d}t = 2\left(\int_0^{1/3} K_1(t)\mathrm{d}t - \int_{1/3}^{1} K_1(t)\mathrm{d}t\right)$$

$$\begin{aligned}
&= \frac{1}{3}\left(\int_0^{1/3} (1-t)(1-3t)\mathrm{d}t + \int_{1/3}^{1} (1-t)(3t-1)\mathrm{d}t\right) \\
&= \frac{1}{3}\left(\int_0^{1/3} (1-4t+3t^2)\mathrm{d}t + \int_{1/3}^{1} (-1+4t-3t^2)\mathrm{d}t\right) \\
&= \frac{1}{3}\left(\frac{1}{3} - \frac{2}{9} + \frac{1}{27}\right) + \frac{1}{3}\left(-1 + \frac{1}{3} + 2 - \frac{2}{9} - 1 + \frac{1}{27}\right) = \frac{8}{81}.
\end{aligned}$$

Thus,

$$|E(f)| \leq \frac{8}{81}\|f''\|_\infty.$$



(b) With the substitution $t = c + xh$, and applying (a), one obtains

$$\int_{c-h}^{c+h} g(t)\mathrm{d}t = h\int_{-1}^{1} g(c + xh)\mathrm{d}x$$
$$= \frac{h}{3}\,[g(c-h) + 4g(c) + g(c+h)] + E_h(g),$$

so that

$$|E_h(g)| \le \frac{8}{81}\,h \max_{-1\le x\le 1}\left|\frac{\mathrm{d}^2}{\mathrm{d}x^2}\,g(c + xh)\right|.$$

Since $\dfrac{\mathrm{d}^2}{\mathrm{d}x^2}\,g(c + xh) = h^2 g''(c + xh)$, one gets

$$|E_h(g)| \le \frac{8}{81}\,h^3\|g''\|_{\infty[c-h,c+h]}.$$

(c) The Peano kernel $K_1(t)$ in this case is (negative) definite, namely

$$K_1(t) = \begin{cases} \dfrac{1}{2}t(1+t) & \text{if } \ -1 \le t \le 0, \\ -\dfrac{1}{2}t(1-t) & \text{if } \ 0 \le t \le 1, \end{cases}$$

giving

$$E_2^{\mathrm{T}}(f) = f''(\tau)\int_{-1}^{1} K_1(t)\mathrm{d}t = -\frac{1}{6}f''(\tau),$$

hence

$$|E_2^{\mathrm{T}}(f)| \le \frac{1}{6}\|f''\|_\infty.$$

This is only a little worse than the bound in (a).

56. (a) An easy calculation shows that $E(f) = 0$ if $f(x) \equiv 1$ and $f(x) \equiv x$. For $f(x) \equiv x^2$, one obtains $E(f) = -\frac{1}{6}\,(1-\varepsilon)(1-2\varepsilon) < 0$ if $0 < \varepsilon < \frac{1}{2}$. It follows that $d = 1$.

(b) With

$$E(f) = \int_0^1 f(x)\mathrm{d}x - \frac{1}{2\varepsilon}\int_0^\varepsilon f(x)\mathrm{d}x - \frac{1}{2\varepsilon}\int_{1-\varepsilon}^1 f(x)\mathrm{d}x,$$

one has

$$K_1(t) = E_{(x)}((x-t)_+) = \int_t^1 (x-t)\mathrm{d}x - \frac{1}{2\varepsilon}\int_0^\varepsilon (x-t)_+\mathrm{d}x$$
$$-\frac{1}{2\varepsilon}\int_{1-\varepsilon}^1 (x-t)_+\mathrm{d}x.$$

Now

$$\int_t^1 (x-t)\mathrm{d}x = \frac{1}{2}(1-t)^2,$$

$$\int_0^\varepsilon (x-t)_+\mathrm{d}x = \begin{cases} 0 \;\text{ if }\; t > \varepsilon \\ \int_t^\varepsilon (x-t)\mathrm{d}x \;\text{ if }\; 0 \le t \le \varepsilon \end{cases} = \begin{cases} 0 \\ \frac{1}{2}\,(\varepsilon-t)^2, \end{cases}$$

$$\int_{1-\varepsilon}^1 (x-t)_+\mathrm{d}x = \begin{cases} \int_t^1 (x-t)\mathrm{d}x \;\text{ if }\; t > 1-\varepsilon \\ \int_{1-\varepsilon}^1 (x-t)\mathrm{d}x \;\text{ if }\; t \le 1-\varepsilon \end{cases}$$

$$= \begin{cases} \frac{1}{2}\,(1-t)^2 \\ \frac{1}{2}\,[(1-t)^2 - (1-\varepsilon-t)^2] \end{cases} = \begin{cases} \frac{1}{2}\,(1-t)^2 \\ \varepsilon\left(1-t-\frac{1}{2}\,\varepsilon\right). \end{cases}$$

Therefore, if $0 \le t \le \varepsilon$, then

$$K_1(t) = \frac{1}{2}\,(1-t)^2 - \frac{1}{2\varepsilon}\cdot\frac{1}{2}\,(\varepsilon-t)^2 - \frac{1}{2\varepsilon}\cdot\varepsilon\left(1-t-\frac{1}{2}\,\varepsilon\right),$$

which, after some algebra, reduces to

$$(1)\qquad K_1(t) = \frac{1}{2}\,t^2\left(1 - \frac{1}{2\varepsilon}\right), \quad 0 \le t \le \varepsilon.$$

If $\varepsilon \le t \le 1-\varepsilon$, then

$$K_1(t) = \frac{1}{2}\,(1-t)^2 - \frac{1}{2\varepsilon}\cdot\varepsilon\left(1 - t - \frac{1}{2}\,\varepsilon\right),$$

that is,

$$(2)\qquad K_1(t) = -\frac{1}{2}\,t(1-t) + \frac{1}{4}\,\varepsilon, \quad \varepsilon \le t \le 1-\varepsilon.$$

Finally, if $t \ge 1-\varepsilon$, then

$$K_1(t) = \frac{1}{2}(1-t)^2 - \frac{1}{2\varepsilon}\cdot\frac{1}{2}\,(1-t)^2,$$

that is,

$$(3)\qquad K_1(t) = \frac{1}{2}\,(1-t)^2\left(1 - \frac{1}{2\varepsilon}\right), \quad 1-\varepsilon \le t \le 1.$$

Therefore,

$$(4)\qquad E(f) = \int_0^1 K_1(t)f''(t)\mathrm{d}t,$$

with K_1 as above in (1)–(3).

(c) Since $1 - \frac{1}{2\varepsilon} < 0$ when $0 < \varepsilon < \frac{1}{2}$, it follows from (1) and (3) that

$$K_1(t) \le 0 \ \text{ if } \ 0 \le t \le \varepsilon \ \text{ or } \ 1-\varepsilon \le t \le 1.$$

For $\varepsilon \le t \le 1-\varepsilon$, one has from (2) that

$$K_1(t) = -\frac{1}{2}\,t(1-t) + \frac{1}{4}\,\varepsilon \le -\frac{1}{2}\,\varepsilon(1-\varepsilon) + \frac{1}{4}\,\varepsilon = -\frac{1}{4}\,\varepsilon(1-2\varepsilon) < 0,$$

since $0 < \varepsilon < \dfrac{1}{2}$. Altogether, therefore,

$$K_1(t) \le 0 \ \text{ for } \ 0 \le t \le 1,$$

and K_1 is negative definite. Consequently,

$$E(f) = e_2 f''(\tau), \ \ 0 < \tau < 1,$$

with

$$e_2 = E\left(\frac{x^2}{2}\right) = -\frac{1}{12}(1-2\varepsilon)(1-\varepsilon),$$

as follows by an elementary calculation.

(d) The limit case $\varepsilon \downarrow 0$ gives us the known results for the (ordinary) trapezoidal formula. If $\varepsilon \to \frac{1}{2}$, then $e_2 \to 0$, that is, $E(f) \to 0$. This is consistent with the fact that, for $\varepsilon \to \frac{1}{2}$,

$$\frac{1}{2}\left[\frac{1}{\varepsilon}\int_0^{\varepsilon} f(x)\mathrm{d}x + \frac{1}{\varepsilon}\int_{1-\varepsilon}^{1} f(x)\mathrm{d}x\right] \to \int_0^1 f(x)\mathrm{d}x.$$

60. (a) The remainder vanishes for the first three powers $1, x, x^2$ of x if

$$\alpha = 1,$$
$$x_1 + \beta = \frac{1}{2},$$
$$x_1^2 + \beta = \frac{1}{3}.$$

Eliminating β from the last two equations gives the quadratic equation $x_1^2 - x_1 + \frac{1}{6} = 0$, which has two solutions,

$$x_1 = \frac{1}{2}\left(1 \pm \frac{1}{\sqrt{3}}\right),$$

both located in $(0, 1)$. Thus,

$$\alpha = 1, \quad x_1 = \frac{1}{2}\left(1 \pm \frac{1}{\sqrt{3}}\right), \quad \beta = \mp\frac{1}{2\sqrt{3}}.$$

With these values one gets, since $x_1 = \frac{1}{2} - \beta$, $x_1^2 = \frac{1}{3} - \beta$, and $\beta^2 = \frac{1}{12}$,

$$E(x^3) = \frac{1}{4} - x_1^3 - \beta = \frac{1}{4} - \left(\frac{1}{2} - \beta\right)\left(\frac{1}{3} - \beta\right) - \beta = -\frac{1}{6}\beta = \pm\frac{1}{12}\frac{1}{\sqrt{3}} \neq 0,$$

so the maximum attainable degree of exactness is $d = 2$.

(b) Each of the two quadrature formulae obtained in (a) (having three nodes and degree of exactness 2) is interpolatory. Therefore, if $f \in C^3[0, 1]$, then

$$E(f) = \int_0^1 [f(x) - p_2(f; 0, x_1, 1; x)]\mathrm{d}x = \int_0^1 x(x-1)(x-x_1)\frac{f^{(3)}(\xi(x))}{6}\mathrm{d}x.$$

There follows

$$|E(f)| \le \frac{1}{6}\cdot\gamma\cdot\|f^{(3)}\|_\infty, \quad \text{where } \gamma = \int_0^1 x(1-x)|x - x_1|\mathrm{d}x.$$

The numerical factor γ can be written as

$$\gamma = \int_0^{x_1} x(1-x)(x_1-x)\mathrm{d}x + \int_{x_1}^1 x(1-x)(x-x_1)\mathrm{d}x,$$

where the first integral evaluates to

$$\frac{1}{6}x_1^3 - \frac{1}{12}x_1^4$$

and the second to

$$\frac{1}{12} - \frac{1}{6}x_1 + \frac{1}{6}x_1^3 - \frac{1}{12}x_1^4.$$

Thus,

$$\gamma = \frac{1}{12} - \frac{1}{6}x_1 + \frac{1}{3}x_1^3 - \frac{1}{6}x_1^4.$$

Using repeatedly the equation $x_1^2 = x_1 - \frac{1}{6}$, one can eliminate all higher powers of x_1:

$$\begin{aligned}
\gamma &= \frac{1}{12} - \frac{1}{6}x_1 + \frac{1}{3}x_1^3 - \frac{1}{6}x_1^2\left(x_1 - \frac{1}{6}\right) \\
&= \frac{1}{12} - \frac{1}{6}x_1 + \frac{1}{36}x_1^2 + \frac{1}{6}x_1^3 \\
&= \frac{1}{12} - \frac{1}{6}x_1 + \frac{1}{36}\left(x_1 - \frac{1}{6}\right) + \frac{1}{6}x_1\left(x_1 - \frac{1}{6}\right) \\
&= \frac{1}{12} - \frac{1}{6}x_1 + \frac{1}{6}x_1^2 - \frac{1}{216} \\
&= \frac{1}{12} - \frac{1}{6}x_1 + \frac{1}{6}\left(x_1 - \frac{1}{6}\right) - \frac{1}{216} \\
&= \frac{1}{12} - \frac{1}{36} - \frac{1}{216} \\
&= \frac{11}{216}.
\end{aligned}$$

The value of γ, being independent of x_1, holds for both choices of x_1. Thus,

$$|E(f)| \le \frac{11}{1296}\|f^{(3)}\|.$$

(c) By the work in (a),

$$\begin{aligned}\int_c^{c+h} f(t)\mathrm{d}t &= h\int_0^1 f(c+xh)\mathrm{d}x \\ &= h\{f(c+x_1h)+\beta[f(c+h)-f(c)]\}+E_h(f)(c),\end{aligned}$$

with β and x_1 as determined in (a). From the result in (b),

$$|E_h(f)(c)| \le \frac{11}{1296}h^4\|f^{(3)}\|_{\infty[c,c+h]}.$$

(d) Letting $h=(b-a)/n,\ t_k=a+kh,\ f_k=f(t_k),\ k=0,1,2,\dots,n$, we have

$$\int_a^b f(t)\mathrm{d}t = \sum_{k=0}^{n-1}\int_{t_k}^{t_k+h} f(t)\mathrm{d}t$$

$$\begin{aligned}&= h[f(t_0+x_1h)+\beta(f_1-f_0)+f(t_1+x_1h)+\beta(f_2-f_1)+\cdots \\ &\quad + f(t_{n-1}+x_1h)+\beta(f_n-f_{n-1})]+\sum_{k=0}^{n-1}E_h(f)(t_k) \\ &= h\left\{\sum_{k=0}^{n-1}f(t_k+x_1h)+\beta[f(b)-f(a)]\right\}+E_n(f),\end{aligned}$$

where, by the result in (c),

$$\begin{aligned}|E_n(f)| &\le \sum_{k=0}^{n-1}|E_h(f)(t_k)| \le \frac{11}{1296}\sum_{k=0}^{n-1}h^4\|f^{(3)}\|_{\infty[t_k,t_{k+1}]} \\ &\le \frac{11}{1296}nh\cdot h^3\left(\frac{1}{n}\sum_{k=0}^{n-1}\|f^{(3)}\|_{\infty[t_k,t_{k+1}]}\right) \\ &= \frac{11}{1296}(b-a)h^3\|f^{(3)}\|_\infty.\end{aligned}$$

63. (a) If $p(x,y)=g(x,y)$ is to hold at the four corner points of the square, we must have

$$\begin{aligned}a &= g(0,0), \\ a+b &= g(1,0), \\ a+c &= g(0,1), \\ a+b+c+d &= g(1,1).\end{aligned}$$

Solving for a, b, c, and d gives

$$\begin{aligned} a &= g(0,0), \\ b &= g(1,0) - g(0,0), \\ c &= g(0,1) - g(0,0), \\ d &= g(1,1) - g(1,0) - g(0,1) + g(0,0). \end{aligned}$$

(b) If instead of g we integrate p (as determined in (a)), we get the approximation

$$\begin{aligned} \int_0^1 \int_0^1 g(x,y)\mathrm{d}x\mathrm{d}y &\approx a \int_0^1 \int_0^1 \mathrm{d}x\mathrm{d}y + b \int_0^1 \mathrm{d}y \int_0^1 x\mathrm{d}x \\ &+ c \int_0^1 \mathrm{d}x \int_0^1 y\mathrm{d}y + d \int_0^1 x\mathrm{d}x \int_0^1 y\mathrm{d}y = a + \frac{1}{2}b + \frac{1}{2}c + \frac{1}{4}d, \end{aligned}$$

hence, substituting from (a),

$$\begin{aligned} \int_0^1 \int_0^1 g(x,y)\mathrm{d}x\mathrm{d}y &\approx g(0,0) + \frac{1}{2}[g(1,0) - g(0,0)] \\ &+ \frac{1}{2}[g(0,1) - g(0,0)] + \frac{1}{4}[g(1,1) - g(1,0) - g(0,1) + g(0,0)], \end{aligned}$$

that is,

$$\int_0^1 \int_0^1 g(x,y)\mathrm{d}x\mathrm{d}y \approx \frac{1}{4}[g(0,0) + g(1,0) + g(0,1) + g(1,1)].$$

If $g(x,y) = g(x)$, this reduces to the trapezoidal rule.

(c) The grid square

$$Q_{ij} = \{(x,y) : \ ih \le x \le (i+1)h, \ \ jh \le y \le (j+1)h\}$$

is mapped by

$$\begin{aligned} x &= h(i+u), \\ y &= h(j+v) \end{aligned}$$

onto the unit square $0 \le u \le 1, 0 \le v \le 1$. Since

$$\frac{\partial(x,y)}{\partial(u,v)} = \begin{vmatrix} h & 0 \\ 0 & h \end{vmatrix} = h^2,$$

we get

$$\int\int_{Q_{i,j}} g(x,y)\mathrm{d}x\mathrm{d}y = \int_0^1\int_0^1 g(h(i+u),h(j+v))\cdot h^2\mathrm{d}u\mathrm{d}v$$
$$\approx \frac{h^2}{4}[g_{i,j}+g_{i+1,j}+g_{j+1,i}+g_{j+1,j+1}].$$

Summing over all grid squares gives

$$\int_0^1\int_0^1 g(x,y)\mathrm{d}x\mathrm{d}y \approx h^2\left\{\sum_{(i,j)\in\mathcal{S}} g_{i,j} + \frac{1}{2}\sum_{(i,j)\in\mathcal{B}} g_{i,j} + \frac{1}{4}\sum_{(i,j)\in\mathcal{C}} g_{i,j}\right\},$$

where $\mathcal{S}$, $\mathcal{B}$, $\mathcal{C}$ denote, respectively, the sets of interior grid points, interior boundary points, and (four) corner points.

Selected Solutions to Machine Assignments

3.(a) PROGRAMS

```
%MAIII_3A  Boundary of positivity domain
%
N=50;
eps0=.5e-14;
[abound,bbound]=posdomainNC(N);
ab0=r_jacobi(floor((N+1)/2));
%ab0=r_jacobi(floor((N+1)/2),-1/2);
%ab0=r_jacobi(floor((N+1)/2),1/2);
ib=find(bbound-abound);
for i=1:size(ib,1)
  ap(i)=abound(i);
  bhigh=bbound(i); blow=abound(i);
  while bhigh-blow>.5e-5
    bm=(bhigh+blow)/2;
    ab=r_jacobi(N,abound(i),bm);
    y=1;
    for n=1:N
      pos=posNC(n,ab,ab0,eps0);
      if pos==0
        y=0;
        break
      end
    end
    if y==0
      bhigh=bm;
    else
      blow=bm;
```

```
       end
     end
     bp(i)=bm;
   end
   figure
   hold on
   plot(ap,bp)
   axis('square')
   plot([-1 3],[-1 3])
   %plot([-1 1.5],[-1 1.5])
   %plot([-1 4],[-1 4])
   hold off


% POSDOMAINNC
%
% Positivity domain for Newton-Cotes formulae with
% Jacobi abscissae and integration relative to the
% weight function 1 and Chebyshev weight functions
% of the first and second kind.
%
function [abound,bbound]=posdomainNC(N)
hold on
abound=zeros(80,1); bbound=zeros(80,1);
%abound=zeros(100,1); bbound=zeros(100,1);
%abound=zeros(100,1); bbound=zeros(100,1);
eps0=.5e-14; i=0;
for a=-.95:.05:3
%for a=-.975:.025:1.5
%for a=-.95:.05:4
  i=i+1; abound(i)=a; k=0;
  for b=a:.05:3
%  for b=a:.025:1.5
%  for b=a:.05:4
    ab=r_jacobi(N,a,b);
    ab0=r_jacobi(floor((N+1)/2));
%    ab0=r_jacobi(floor((N+1)/2),-1/2);
%    ab0=r_jacobi(floor((N+1)/2),1/2);
    y=1;
    for n=1:N
      pos=posNC(n,ab,ab0,eps0);
      if pos==0
        y=0;
        break
      end
    end
    if y==0
      plot(a,b,':')
      axis('square')
      k=k+1;
      if k==1, bbound(i)=b; end
    else
      plot(a,b,'r+')
    end
  end
end
hold off
```

```
% POSNC
%
% Positivity of the n-point Newton-Cotes formula
% with abscissae equal to the zeros of the orthogonal
% polynomial associated with the nx2 recurrence matrix
% ab and integration being with respect to the measure
% identified by the floor((n+1)/2)x2 recurrence matrix
% ab0. The input parameter eps0, required in the routine
% NewtCotes.m, is a number larger than, but close to,
% the machine precision.
%
function y=posNC(n,ab,ab0,eps0)
y=0;
xw=gauss(n,ab); zn=zeros(n,1);
w=NewtCotes(n,xw(:,1),ab0,eps0);
if w>zn, y=1; end
```

Why does $\beta \geq \alpha$ suffice? To simplify notation, let

$$\begin{aligned} p_n(x) &:= P_n^{(\alpha,\beta)}(x), \;\; w(x) = (1-x)^\alpha(1+x)^\beta; \\ P_n(x) &= P_n^{(\beta,\alpha)}(x), \;\; W(x) = (1-x)^\beta(1+x)^\alpha. \end{aligned}$$

If x_ν denote the zeros of p_n, and X_ν those of P_n, one has $X_\nu = -x_\nu$ by the reflection formula for Jacobi polynomials (see *Hint*). With

$$\ell_\nu(x) = \frac{p_n(x)}{(x-x_\nu)p_n'(x_\nu)}, \;\; L_\nu(x) = \frac{P_n(x)}{(x-X_\nu)P_n'(X_\nu)},$$

there follows

$$\begin{aligned} W_\nu = \int_{-1}^{1} L_\nu(x)W(x) &= \int_{-1}^{1} \frac{P_n(x)}{(x-X_\nu)P_n'(X_\nu)}\, W(x)\mathrm{d}x \\ &= \int_{-1}^{1} \frac{(-1)^n p_n(-x)}{(x+x_\nu)(-1)^{n+1}p_n'(x_\nu)}\,(1-x)^\beta(1+x)^\alpha\mathrm{d}x \\ &= -\int_{-1}^{1} \frac{p_n(-x)}{(x+x_\nu)p_n'(x_\nu)}\,(1-x)^\beta(1+x)^\alpha\mathrm{d}x \\ &= \int_{1}^{-1} \frac{p_n(t)}{(-t+x_\nu)p_n'(x_\nu)}\,(1-t)^\alpha(1+t)^\beta\mathrm{d}t \\ &= -\int_{-1}^{1} \frac{p_n(t)}{-(t-x_\nu)p_n'(x_\nu)}\,(1-t)^\alpha(1+t)^\beta\mathrm{d}t = w_\nu, \end{aligned}$$

showing that the two Newton–Cotes weights are the same.

For relevant literature, see Askey and Fitch [1968], Askey [1972], [1979], Micchelli [1980], Sottas [1982], [1988], [1989], and Gautschi [2011a, Sect. 4.1].

In the case of constant weight function $= 1$, it appears that with increasing N the slanted portion of the upper boundary of the positivity domain slightly turns downward, and the horizontal portion slowly moves down. It is likely that in the limit $N \to \infty$ the slope of the slanted portion tends to 1 and the height of the horizontal portion to 3/2, which would be consistent with a conjecture of Askey (1979) (except for the slanted portion of the boundary). On the line $\beta = 1.55$, for example, spotchecking with $a = -0.4, 0 : 0.5 : 1.5$ revealed nonpositivity of the n-point Newton–Cotes formula when $n = 800$ (but not when $n = 700$).

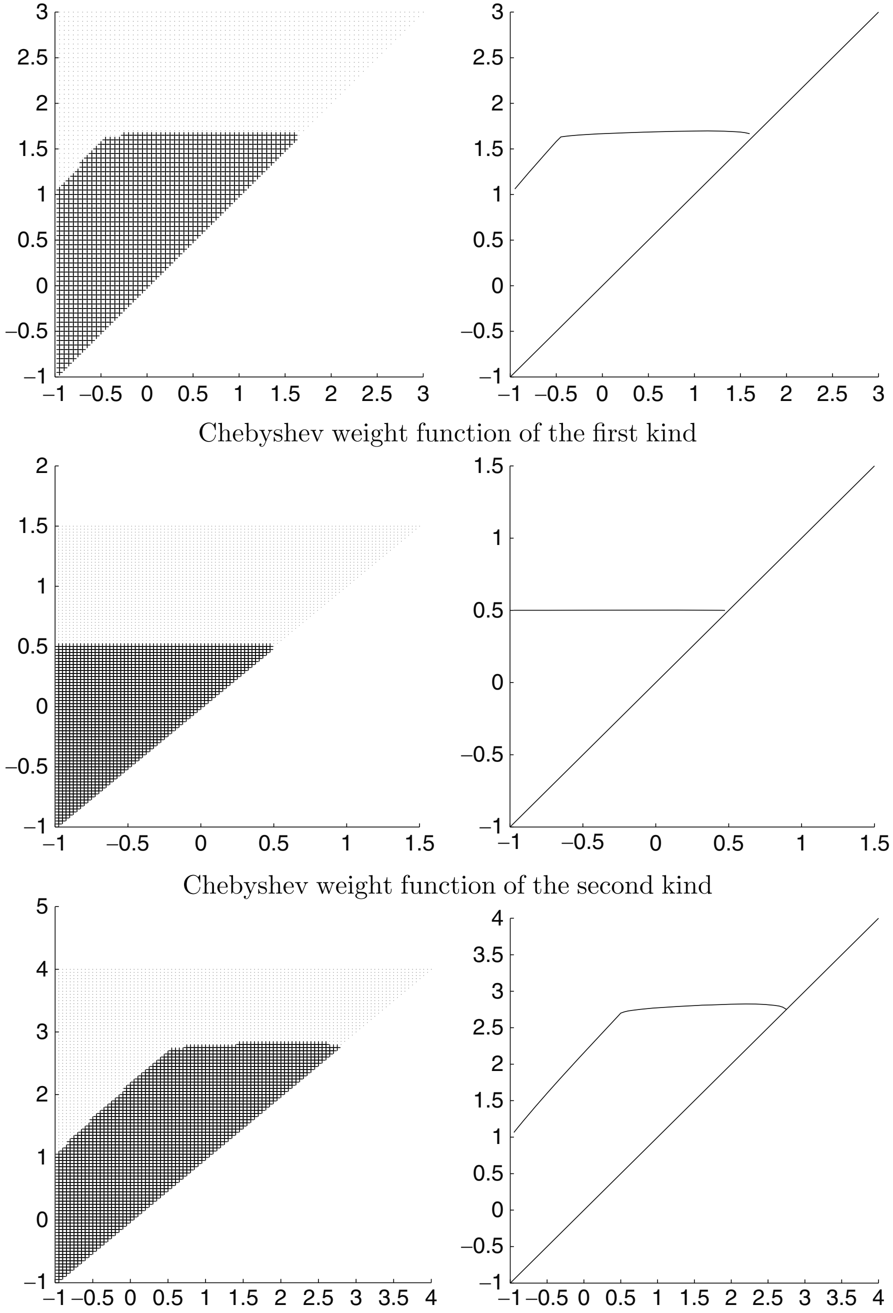
In the case of the Chebyshev weight function of the first kind, the height of the upper boundary is practically constant equal to 1/2 (already for $N = 10$ and more so for larger values of N). This is in agreement with a result proved by Micchelli (1980) in the case of Gegenbauer abscissae, $\alpha = \beta$.

In the case of the Chebyshev weight function of the second kind, the slope of the slanted portion of the boundary curve, as in the first case, seems to tend (slowly) to 1, and the height of the horizontal portion to 2.5. (Cf. also the third column in the output to MAIII_3B, which seems to confirm this, given the slowness of convergence as $N \to \infty$.)

(b) PROGRAM

```
%MAIII_3B Upper bound of Gegenbauer positivity
           interval
%
f0='%8.0f %11.6f\n';
disp('      N     alpha_max')
eps0=.5e-14;
ab0=r_jacobi(50);
%ab0=r_jacobi(50,-1/2);
%ab0=r_jacobi(50,1/2);
for N=[20 50 100]
  ahigh=2; alow=1.5;
%  ahigh=0.6; alow=0.4;
%  ahigh=3.3; alow=2.6;
  while ahigh-alow>.5e-6
    a=(ahigh+alow)/2;
    ab=r_jacobi(N,a);
    y=1;
    for n=1:N
      pos=posNC(n,ab,ab0,eps0);
      if pos==0
        y=0;
        break
```

PLOTS (N=50) for 3(a)



Weight function = 1

Chebyshev weight function of the first kind

Chebyshev weight function of the second kind

```
      end
    end
    if y==0
      ahigh=a;
    else
      alow=a;
    end
  end
  fprintf(f0,N,a)
end
```

OUTPUT

```
>> MAIII_3B
      weight function = 1        Chebyshev #1       Chebyshev #2

      N       alpha_max          alpha_max          alpha_max
      20      1.700560           0.500000           2.863848
      50      1.643718           0.500000           2.750502
     100      1.617770           0.500000           2.700881
>>
```

(c) PROGRAMS

```
%MAIII_3C   Boundary of positivity domain
%
N=50;
eps0=.5e-14;
[a0bound,b0bound]=posdomainNC0(N);
ab=r_jacobi(N);
%ab=r_jacobi(N,-1/2);
%ab=r_jacobi(N,1/2);
ib0=find(b0bound-a0bound);
for i=ib0(1):ib0(1)+size(ib0,1)-1
  a0p(i)=a0bound(i);
  b0high=b0bound(i); b0low=a0bound(i);
  while b0high-b0low>.5e-5
    b0m=(b0high+b0low)/2;
    ab0=r_jacobi(floor((N+1)/2),a0bound(i),b0m);
    y=1;
    for n=1:N
      pos=posNC(n,ab,ab0,eps0);
      if pos==0
        y=0;
        break
      end
    end
    if y==0
```

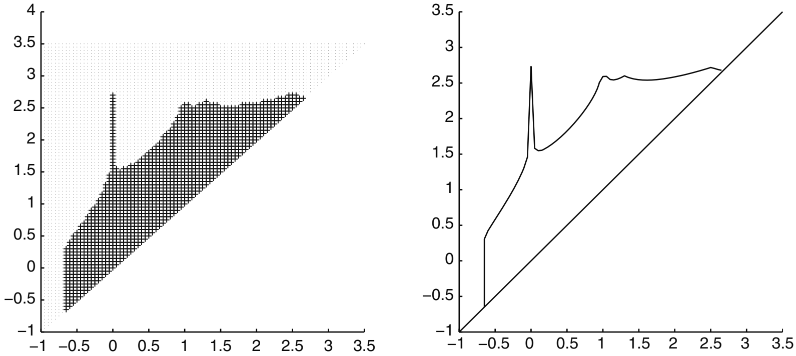
```
        b0high=b0m;
      else
        b0low=b0m;
      end
    end
    b0p(i)=b0m;
  end
  figure
  hold on
  plot([a0p(ib0(1)) a0p(ib0(1))],[a0p(ib0(1))
   b0p(ib0(1))])
  axis('square')
  plot(a0p(ib0(1):size(a0p,2)),b0p(ib0(1):size(b0p,2)))
  plot([-1 3.5],[-1 3.5])
  %plot([-1 3],[-1 3])
  %plot([-1 4],[-1 4])
  hold off

  %POSDOMAINNC0
  %
  %Positivity domain for Newton-Cotes
  %formulae with Gauss-Legendre and Chebyshev abscissae
  %and integration relative to Jacobi weight functions.
  %
  function [a0bound,b0bound]=posdomainNC0(N)
  hold on
  a0bound=zeros(90,1); b0bound=zeros(90,1);
  %a0bound=zeros(80,1); b0bound=zeros(80,1);
  %a0bound=zeros(100,1); b0bound=zeros(100,1);
  eps0=.5e-14; i=0;
  ab=r_jacobi(N);
  %ab=r_jacobi(N,-1/2);
  %ab=r_jacobi(N,1/2);
  for a0=-.95:.05:3.5
  %for a0=-.95:.05:3
  %for a0=-.95:.05:4
    i=i+1; a0bound(i)=a0; k=0;
    for b0=a0:.05:3.5
  %  for b0=a0:.05:3
  %  for b0=a0:.05:4
      ab0=r_jacobi(floor((N+1)/2),a0,b0);
      y=1;
      for n=1:N
        pos=posNC(n,ab,ab0,eps0);
        if pos==0
          y=0;
          break
        end
      end
      if y==0
        plot(a0,b0,':')
        axis('square')
        k=k+1;
        if k==1, b0bound(i)=b0; end
      else
        plot(a0,b0,'r+')
      end
    end
  end
  hold off
```

(d) PROGRAMS

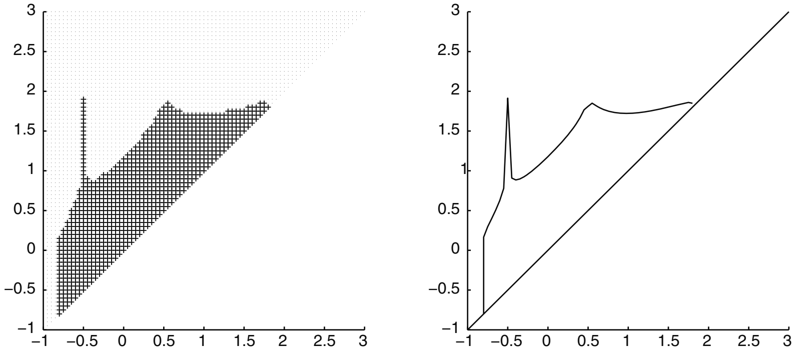
```
%MAIII_3D   Boundary of positivity domain
%
N=50;
eps0=.5e-14;
[abound,bbound]=posdomainNC1(N);
ab0=r_jacobi(floor((N+3)/2));
%ab0=r_jacobi(floor((N+3)/2),-1/2);
%ab0=r_jacobi(floor((N+3)/2),1/2);
ib=find(bbound-abound);
for i=ib(1):size(ib,1)
  ap(i-ib(1)+1)=abound(i);
  bhigh=bbound(i); blow=abound(i);
  while bhigh-blow>.5e-5
    bm=(bhigh+blow)/2;
    ab=r_jacobi(N,abound(i),bm);
    y=1;
    for n=1:N
      pos=posNC1(n,ab,ab0,eps0);
      if pos==0
        y=0;
        break
      end
    end
    if y==0
      bhigh=bm;
    else
      blow=bm;
    end
  end
  bp(i-ib(1)+1)=bm;
end
figure
hold on
plot(ap,bp)
axis('square')
plot([0 0],[0 bp(1)])
```

PLOTS (N=50) for $3(\mathrm{c})$

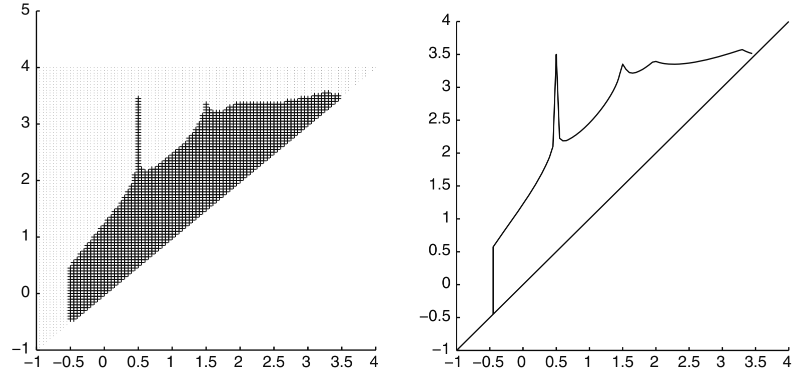
Gauss-Legendre abscissae



Chebyshev abscissae of the first kind



Chebyshev abscissae of the second kind



```
bpf=bp(size(ib,1)-ib(1)+1);
plot([ap(size(ib,1)-ib(1)+1),bpf],[bpf,bpf])
plot([-1 6],[-1 6])
%plot([-1 4],[-1 4])
%plot([-1 6],[-1 6])
hold off

%POSDOMAINNC1
%
%   Positivity domain for closed Newton-Cotes formulae
%   with Jacobi abscissae and integration relative to
%   the weight function 1 and Chebyshev weight functions
%   of the first and second kind.
%
function [abound,bbound]=posdomainNC1(N)
hold on
abound=zeros(140,1); bbound=zeros(140,1);
%abound=zeros(100,1); bbound=zeros(100,1);
%abound=zeros(140,1); bbound=zeros(140,1);
eps0=.5e-14; i=0;
for a=-.95:.05:6
%for a=-.95:.05:4
%for a=-.95:.05:6
  i=i+1; abound(i)=a; k=0;
  for b=a:.05:6
%  for b=a:.05:4
%  for b=a:.05:6
    ab=r_jacobi(N,a,b);
    ab0=r_jacobi(floor((N+3)/2));
%     ab0=r_jacobi(floor((N+3)/2),-1/2);
%     ab0=r_jacobi(floor((N+3)/2),1/2);
    y=1;
    for n=1:N
      pos=posNC1(n,ab,ab0,eps0);
      if pos==0
        y=0;
        break
      end
    end
    if y==0
      plot(a,b,':')
      axis('square')
      k=k+1;
      if k==1, bbound(i)=b; end
    else
      plot(a,b,'r+')
    end
  end
end
```

```
hold off

%POSNC1
%
%   Positivity of the (n+2)-point Newton-Cotes formula
%   with abscissae equal to +- 1 and the n zeros of the
%   orthogonal polynomial associated with the nx2
%   recurrence matrix ab and integration being with
%   respect to the measure identified by the floor
%   ((n+3)/2)x2 recurrence matrix ab0. The input
%   parameter  eps0, required in the routione
%   NewtCotes.m, is a number larger than, but close to,
%   the machine precision.
%
function y=posNC1(n,ab,ab0,eps0)
y=0;
xw=gauss(n,ab); zn=zeros(n+2,1);
xw1=[-1;xw(:,1);1];
w1=NewtCotes(n+2,xw1,ab0,eps0);
if w1>zn, y=1; end
```

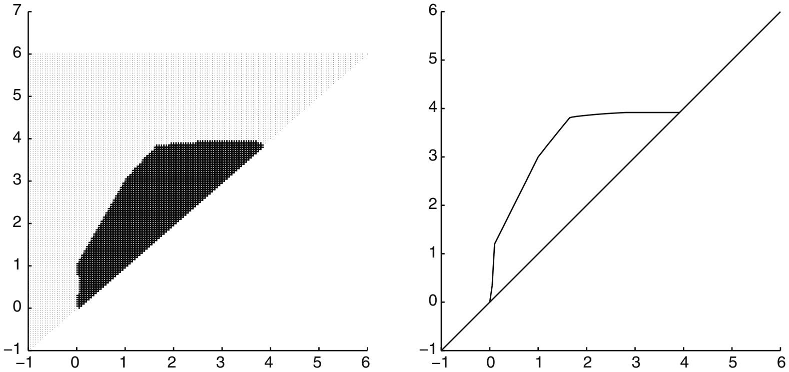
For relevant literature, see Notaris [2002], [2003].

(e) PROGRAM

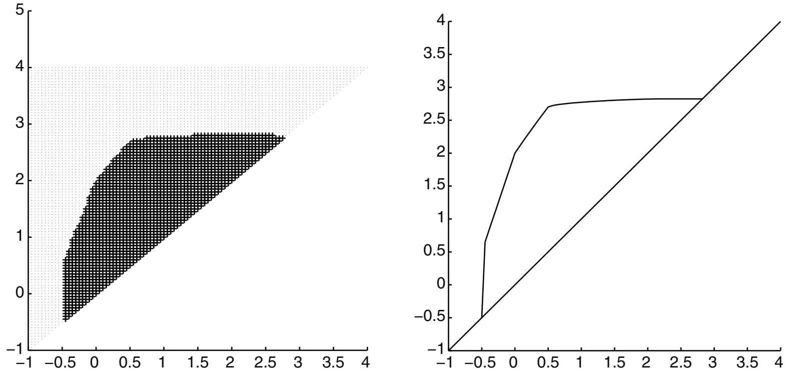
```
%MAIII_3E   Bounds for the Gegenbauer
  positivity intervals
%
f0='%8.0f %11.6f\n';
disp('      N     alpha_max')
%disp('      N     alpha_min')
eps0=.5e-14;
ab0=r_jacobi(51);
%ab0=r_jacobi(51,-1/2);
%ab0=r_jacobi(51,1/2);
for N=[20 50 100]
  ahigh=4.5; alow=3.5;
%  ahigh=3; alow=2.5;
%  ahigh=5.3; alow=4.5;
%  ahigh=.1; alow=-.1;
%  ahigh=-.4; alow=-.6;
%  ahigh=.6; alow=.4;
  while ahigh-alow>.5e-6
    a=(ahigh+alow)/2;
    ab=r_jacobi(N,a);
    y=1;
    for n=1:N
      pos=posNC1(n,ab,ab0,eps0);
```

PLOTS (N=50) for (3d)

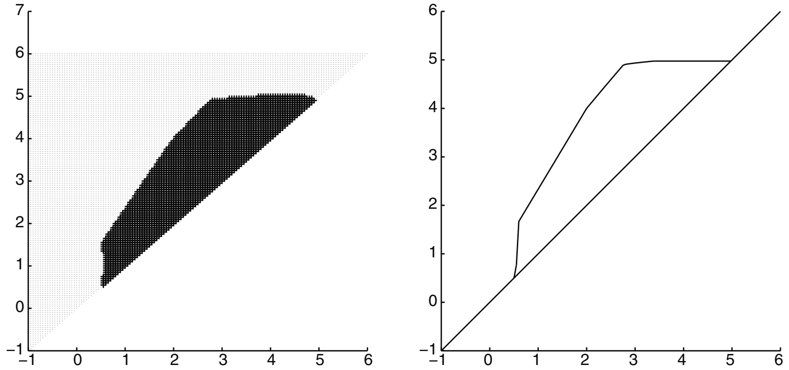
Weight function = 1



Chebyshev weight function of the first kind



Chebyshev weight function of the second kind



```
      if pos==0
        y=0;
        break
      end
    end
    if y==0
      ahigh=a;
%       alow=a;
    else
      alow=a;
%       ahigh=a;
    end
  end
  fprintf(f0,N,a)
end
```

In the case of constant weight function = 1 and Gegenbauer abscissae, one has positivity of the closed Newton–Cotes formulae when $0 \le \alpha = \beta \le 3.5$ by a result of Kütz (cf. Notaris [2002, p. 144]). This is consistent with our plot on the previous page and the output of MAIII_3E below. Also the positivity domain proved in Notaris [2002, Theorem 2.1(a),(b)] is indeed a small subdomain of the respective domain in our plot.

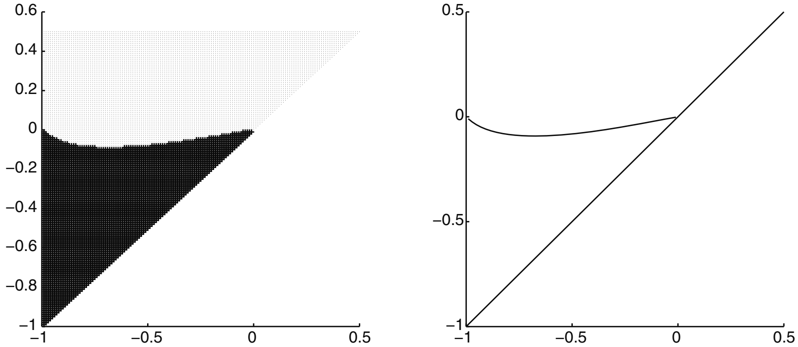
```
OUTPUT

>> MAIII_3E

 weight function = 1          Chebyshev #1          Chebyshev #2

N      alpha_min/max          alpha_min/max         alpha_min/max
20   0.000000   4.012030   -0.500000   2.863848   0.500000   5.152099
50   0.000000   3.841891   -0.500000   2,750502   0.500000   4.924714
100  0.000000   3.769501   -0.500000   2.700881   0.500000   4.829984
>>
```

(f)

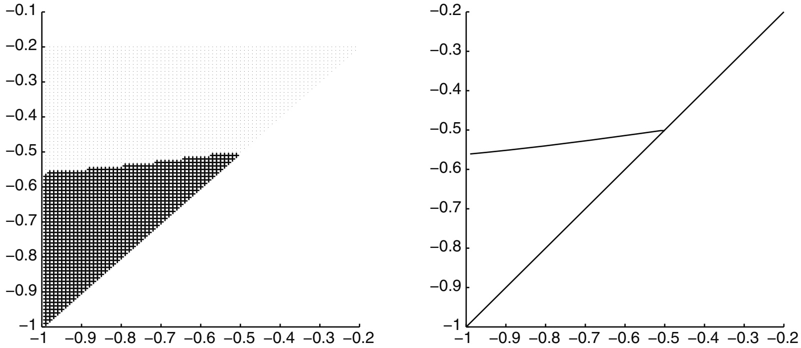
```
PROGRAMS

%MAIII_3F   Boundary of positivity domain
%
N=50;
eps0=.5e-14;
[a0bound,b0bound]=posdomainNC01(N);
ab=r_jacobi(N);
%ab=r_jacobi(N,-1/2);
%ab=r_jacobi(N,1/2);
ib0=find(b0bound-a0bound);
for i=ib0(1):ib0(1)+size(ib0,1)-1
  a0p(i)=a0bound(i);
  b0high=b0bound(i); b0low=a0bound(i);
```

PLOTS (N=50) for 3(f)

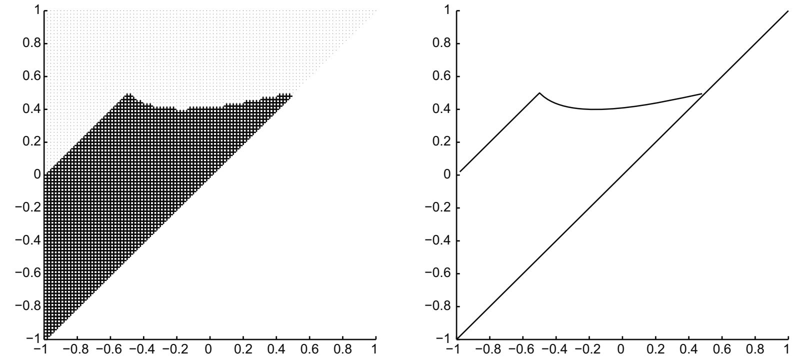
Gauss–Legendre abscissae



Chebyshev abscissae of the first kind



Chebyshev abscissae of the second kind



```
  while b0high-b0low>.5e-5
    b0m=(b0high+b0low)/2;
    ab0=r_jacobi(floor((N+3)/2),a0bound(i),b0m);
    y=1;
    for n=1:N
      pos=posNC1(n,ab,ab0,eps0);
      if pos==0
        y=0;
        break
      end
    end
    if y==0
      b0high=b0m;
    else
      b0low=b0m;
    end
  end
  b0p(i)=b0m;
end
figure
hold on
plot(a0p(ib0(1):size(a0p,2)),b0p(ib0(1):size(b0p,2)))
axis('square')
plot([-1 .5],[-1 .5])
%plot([-1 -.2],[-1 -.2])
%plot([-1 1],[-1 1])
hold off

%POSDOMAINNC01
%
%   Positivity domain for closed Newton-Cotes formulae with
%   Gauss--Legendre and Chebyshev abscissae and integration
%   relative to Jacobi weight functions.
%

function [a0bound,b0bound]=posdomainNC01(N)
hold on
a0bound=zeros(150,1); b0bound=zeros(150,1);
%a0bound=zeros(80,1); b0bound=zeros(80,1);
%a0bound=zeros(100,1); b0bound=zeros(100,1);
eps0=.5e-14; i=0;
ab=r_jacobi(N);
%ab=r_jacobi(N,-1/2);
%ab=r_jacobi(N,1/2);
for a0=-.99:.01:.5
%for a0=-.99:.01:-.2
%for a0=-.98:.02:1
  i=i+1; a0bound(i)=a0; k=0;
  for b0=a0:.01:.5
```

```
%  for b0=a0:.01:-.2
%  for b0=a0:.02:1
    ab0=r_jacobi(floor((N+3)/2),a0,b0);
    y=1;
    for n=1:N
      pos=posNC1(n,ab,ab0,eps0);
      if pos==0
        y=0;
        break
      end
    end
    if y==0
      plot(a0,b0,':')
      axis('square')
      k=k+1;
      if k==1, b0bound(i)=b0; end
    else
      plot(a0,b0,'r+')
    end
  end
end
hold off
```

6. (a) It suffices to derive the second relation:

$$\begin{aligned}
T_{k+1,0} &= h_{k+1}\left\{\frac{1}{2}\,f(a)+\frac{1}{2}\,f(b)+\sum_{\ell=1}^{2^{k+1}-1} f(a+\ell h_{k+1})\right\}\\
&= \frac{1}{2}\,h_k\cdot\frac{1}{2}\,[f(a)+f(b)]+\frac{1}{2}\,h_k\sum_{\substack{\ell=2\\(\ell\text{ even})}}^{2^{k+1}-2} f(a+\ell h_{k+1})\\
&\quad+\frac{1}{2}\,h_k\sum_{\substack{\ell=1\\(\ell\text{ odd})}}^{2^{k+1}-1} f(a+\ell h_{k+1})\\
&= \frac{1}{2}\,h_k\left\{\frac{1}{2}\,f(a)+\sum_{j=1}^{2^k-1} f(a+2jh_{k+1})+\frac{1}{2}\,f(b)\right\}\\
&\quad+\frac{1}{2}\,h_k\sum_{j=1}^{2^k} f(a+(2j-1)h_{k+1})\\
&= \frac{1}{2}\,h_k\left\{\frac{1}{2}\,f(a)+\sum_{j=1}^{2^k-1} f(a+jh_k)+\frac{1}{2}\,f(b)\right\}
\end{aligned}$$

$$\begin{aligned} &+\frac{1}{2}\,h_k\sum_{j=1}^{2^k} f\left(a+\left(j-\frac{1}{2}\right)h_k\right)\\ &=\frac{1}{2}\,T_{k,0}+\frac{1}{2}\,M_{h_k}(f). \end{aligned}$$

(b)–(d) PROGRAM

```
%MAIII_6BC
%
f0='%6.0f %19.15f %19.15f\n’;
disp(’     i        T(i,1)               T(i,i)’)
n=10;
a=1; b=2;
%a=0; b=1;
%a=0; b=pi;
%a=0; b=1;
%a=0; b=2;
T=romberg(a,b,n);
for i=1:n
  fprintf(f0,i,T(i,1),T(i,i))
end

function T=romberg(a,b,n)
T=zeros(1:n);
h=b-a; m=1; T(1,1)=h*(f(a)+f(b))/2;
for i=2:n
  h=h/2; m=2*m; mm1=m-1;
  k=(1:2:mm1)’;
  T(i,1)=T(i-1,1)/2+h*sum(f(a+h*k));
  l=1;
  for k=2:i
    l=4*l;
    T(i,k)=T(i,k-1)+(T(i,k-1)-T(i-1,k-1))/(l-1);
  end
end

function y=f(x)
y=exp(x)./x;
%y=1;
%if x~=zeros(size(x,1),1)
%  y=sin(x)./x;
%end
%y=cos(1.7*x)/pi;
%y=cos(1.7*sin(x))/pi;
%y=sqrt(1-x.^2);
%y=zeros(size(x,1),1);
%for i=1:size(x,1)
```

```
%  if x(i)<=sqrt(2)
%    y(i)=x(i);
%  else
%    y(i)=sqrt(2)*(2-x(i))/(2-sqrt(2));
%  end
%end
%y=zeros(size(x,1),1);
%for i=1:size(x,1)
%  if x(i)<=3/4
%    y(i)=x(i);
%  else
%    y(i)=3*(2-x(i))/5;
%  end
%end
```

OUTPUT

```
>> MAIII_6BC
     i        T(i,1)              T(i,i)
     1   3.206404938962185   3.206404938962185  (i) Ei(2)-Ei(1)
     2   3.097098826260448   3.060663455359868
     3   3.068704101194839   3.059144242004954
     4   3.061519689433579   3.059116836818692
     5   3.059717728013521   3.059116541002761
     6   3.059266861956402   3.059116539648306
     7   3.059154121802282   3.059116539645955
     8   3.059125935283745   3.059116539645953
     9   3.059118888561571   3.059116539645952
    10   3.059117126875243   3.059116539645953
>>
>> MAIII_6BC
     i        T(i,1)              T(i,i)
     1   0.920735492403948   0.920735492403948  (ii) SI(1)
     2   0.939793284806177   0.946145882273587
     3   0.944513521665390   0.946083004063674
     4   0.945690863582701   0.946083070387223
     5   0.945985029934386   0.946083070367181
     6   0.946058560962768   0.946083070367183
     7   0.946076943060063   0.946083070367183
     8   0.946081538543152   0.946083070367183
     9   0.946082687411347   0.946083070367183
    10   0.946082974628235   0.946083070367183
>>
>> MAIII_6BC
     i        T(i,1)              T(i,i)
     1   0.793892626146237   0.793892626146237  (iii) sin(1.7*pi)/pi
     2  -0.048556949021066  -0.329373474076833
     3  -0.128279145629103  -0.143218526971000
     4  -0.145813060666924  -0.151575238486816
     5  -0.150072135492423  -0.151480973833174
     6  -0.151129454951574  -0.151481239834844
     7  -0.151393324105836  -0.151481239647168
     8  -0.151459262675203  -0.151481239647201
     9  -0.151475745523767  -0.151481239647201
    10  -0.151479866123815  -0.151481239647201
>>
>> MAIII_6BC
     i        T(i,1)              T(i,i)
     1   1.000000000000000   1.000000000000000  (iv) J_0(1.7)
     2   0.435577752852238   0.247437003802984
     3   0.397997329127638   0.394672755713868
     4   0.397984859446116   0.398880460382129
     5   0.397984859446110   0.397968405925570
     6   0.397984859446110   0.397984921711269
     7   0.397984859446110   0.397984859403064
     8   0.397984859446110   0.397984859446102
     9   0.397984859446109   0.397984859446109
    10   0.397984859446110   0.397984859446110
>>
>> MAIII_6BC
     i        T(i,1)              T(i,i)
     1   0.500000000000000   0.500000000000000  (v) pi/4
     2   0.683012701892219   0.744016935856292
     3   0.748927267025610   0.772690912262104
     4   0.772454786089293   0.781054541057592
     5   0.780813259456935   0.783876545840612
     6   0.783775605719283   0.784861687334472
     7   0.784824228194921   0.785208669629317
     8   0.785195198099154   0.785331191417285
     9   0.785326395739308   0.785374488842346
    10   0.785372788179914   0.785389793759148
>>
>> MAIII_6BC
     i        T(i,1)              T(i,i)
     1   0.000000000000000   0.000000000000000   (vi) sqrt(2)
     2   1.000000000000000   1.333333333333333
     3   1.353553390593274   1.480609266621545
     4   1.390165042944955   1.396451590456853
     5   1.408470869120796   1.415741434555130
     6   1.412654727760896   1.413984394417376
     7   1.413896991372851   1.414335276603458
     8   1.414109407799875   1.414168137287926
     9   1.414211586644613   1.414251685496006
    10   1.414212593986806   1.414209909989944
>>
>> MAIII_6BC
     i        T(i,1)              T(i,i)
     1   0.000000000000000   0.000000000000000  (vii) 3/4
```

```
          2    0.600000000000000    0.800000000000000
          3    0.700000000000000    0.728888888888889
          4    0.750000000000000    0.769523809523810
          5    0.750000000000000    0.748489262371615
          6    0.750000000000000    0.750024807644598
          7    0.750000000000000    0.749999901904365
          8    0.750000000000000    0.750000000096088
          9    0.750000000000000    0.749999999999976
         10    0.750000000000000    0.750000000000000
>>
```

Comments

1. The Romberg scheme is effective here, since integration is over a smooth nonperiodic function.
2. Same as (i).
3. Same as (i).
4. Romberg is worse than the trapezoidal rule, because the integrand is a smooth periodic function with period π, and integration is over the full period.
5. Romberg only slightly better than the trapezoidal rule, and both converge slowly. The reason is the singularity at $x = 1$ (where the derivative is infinite).
6. Romberg does not provide any improvement, since the derivative of f is discontinuous at an irrational point ($\sqrt{2}$).
7. The trapezoidal rule, in contrast to the Romberg scheme, is exact after the third step, because 3/4 then becomes, and remains, a meshpoint, separating two linear pieces of f. Romberg, however, eventually catches up.

Chapter 4
Nonlinear Equations

The problems discussed in this chapter may be written generically in the form

$$f(x) = 0, \tag{4.1}$$

but allow different interpretations depending on the meaning of x and f. The simplest case is a *single equation* in a single unknown, in which case f is a given function of a real or complex variable, and we are trying to find values of this variable for which f vanishes. Such values are called *roots of the equation* (4.1), or *zeros of the function* f. If x in (4.1) is a vector, say, $x = [x_1, x_2, \dots , x_d]^{\mathrm{T}} \in \mathbb{R}^d$, and f is also a vector, each component of which is a function of d variables $x_1, x_2, \dots , x_d$, then (4.1) represents a *system of equations*. It is said to be a *nonlinear system* if at least one component of f depends nonlinearly on at least one of the variables x_1, $x_2, \dots , x_d$. If all components of f are linear functions of $x_1, x_2, \dots , x_d$, then we call (4.1) a *system of linear algebraic equations*, which (if $d > 1$) is of considerable interest in itself, but is not discussed in this chapter. Still more generally, (4.1) could represent a *functional equation*, if x is an element in some function space and f a (linear or nonlinear) operator acting on this space. In each of these interpretations, the zero on the right of (4.1), of course, has a different meaning: the number zero in the first case, the zero vector in the second, and the function identically equal to zero in the last case.

Much of this chapter is devoted to single nonlinear equations. Such equations are often encountered in the analysis of vibrating systems, where the roots correspond to critical frequencies (resonance). The special case of *algebraic equations*, where f in (4.1) is a polynomial, is also of considerable importance and merits special treatment. Systems of nonlinear equations are briefly considered at the end of the chapter.

W. Gautschi, *Numerical Analysis*, DOI 10.1007/978-0-8176-8259-0_4,

4.1 Examples

4.1.1 A Transcendental Equation

Nonalgebraic equations are referred to as being “transcendental.” An example is

$$\cos x \cosh x - 1 = 0 \tag{4.2}$$

and is typical for equations arising in problems of resonance. Before one starts computing roots, it is helpful to gather some qualitative properties about them: are there any symmetries among the roots? How many roots are there? Where approximately are they located? With regard to symmetry, one notes immediately from (4.2) that the roots are located symmetrically with respect to the origin: if α is a root, so is $-\alpha$. Also, $\alpha = 0$ is a trivial root (which is uninteresting in applications). It suffices therefore to consider positive roots.

A quick way to get insight into the number and location of roots of (4.2) is to divide the equation by $\cos x$ and to rewrite it in the form

$$\cosh x = \frac{1}{\cos x}. \tag{4.3}$$

No roots are being lost by this transformation, since clearly $\cos x \neq 0$ at any root $x = \alpha$. Now one graphs the function on the right and the function on the left and observes where the two graphs intersect. The respective abscissae of intersection are the desired (real) roots of (4.2). This is illustrated in Fig. 4.1 (not drawn to scale). It is evident from this figure that there are infinitely many positive roots. Indeed, each interval $[(2n - \frac{1}{2})\pi,\ (2n + \frac{1}{2})\pi]$, $n = 1, 2, 3, \dots$, has exactly two roots, $\alpha_n < \beta_n$, with α_n rapidly approaching the left endpoint, and β_n the right endpoint, as n increases. These account for all positive roots and thus, by symmetry, for all nonvanishing real roots. In applications, it is likely that only the smallest positive root, α_1, will be of interest.

4.1.2 A Two-Point Boundary Value Problem

Here we are looking for a function $y \in C^2[0, 1]$ satisfying the differential equation

$$y'' = g(x, y, y'), \quad 0 \le x \le 1 \tag{4.4}$$

and the boundary conditions

$$y(0) = y_0, \quad y(1) = y_1, \tag{4.5}$$

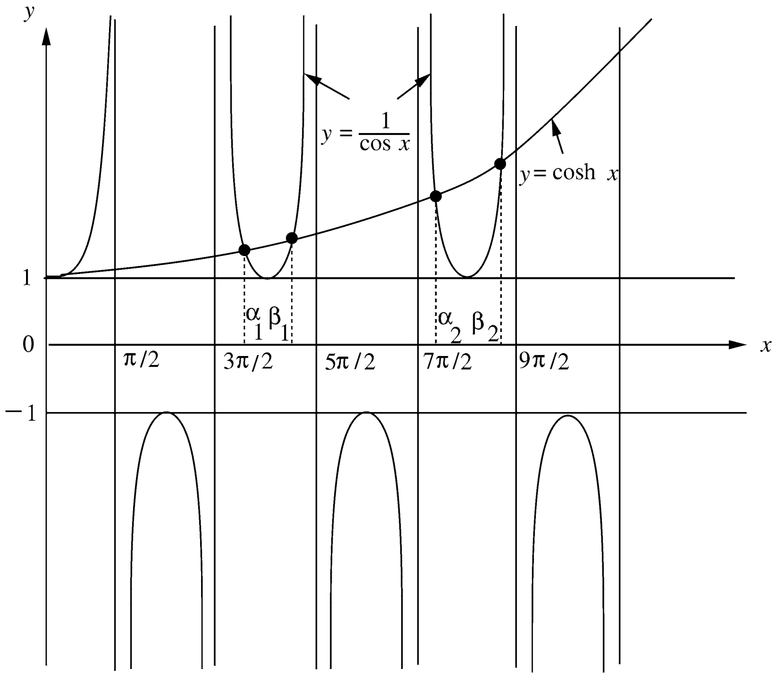


Fig. 4.1 Graphical interpretation of (4.3)

where g is a given (typically nonlinear) function on $[0, 1] \times \mathbb{R} \times \mathbb{R}$, and y_0, y_1 are given numbers. At first sight, this does not look like a problem of the form (4.1), but it can be reduced to it if one introduces the associated initial value problem

$$\begin{aligned} u'' &= g(x, u, u'), \quad 0 \le x \le 1, \\ u(0) &= y_0, \quad u'(0) = s, \end{aligned} \tag{4.6}$$

where s (for "slope") is an unknown to be determined. Suppose, indeed, that for each s, (4.6) has a unique solution that exists on the whole interval [0,1]. Denote it by $u(x) = u(x; s)$. Then problem (4.4), (4.5) is equivalent to problem

$$u(1; s) - y_1 = 0 \tag{4.7}$$

in the sense that to each solution of (4.7) there corresponds a solution of (4.4), (4.5) and vice versa (cf. Chap. 7, Sect. 7.1.2). Thus, by defining

$$f(s) := u(1; s) - y_1, \tag{4.8}$$

we have precisely a problem of the form (4.1). It is to be noted, however, that $f(s)$ is not given explicitly as a function of s; rather, to evaluate $f(s)$ for any s, one has to solve the initial value problem (4.6) over the whole interval [0,1] to find the value of $u(x; s)$ at $x = 1$, and hence of $f(s)$ in (4.8).

A very natural way to go about solving (4.4), (4.5) is to evaluate $f(s)$ for some initial guess s. If $f(s)$ is, say, positive, we lower the value of s until we find one for which $f(s)$ is negative. Then we have two slopes s: one that "overshoots" the target and one that "undershoots" it. We now take as our next aim the average of these slopes and "shoot" again. Depending on whether we hit above the target or below, we discard the first or second initial slope and continue to repeat the same procedure. In the terminology of boundary value problems, this is called the *shooting method*. To shoot is tantamount to solving an initial value problem for a second-order differential equation, which in fact is the equation of the trajectory a bullet would traverse if it were fired from a gun. In the terminology of this chapter, it is called the *bisection method* (cf. Sect. 4.3.1).

4.1.3 A Nonlinear Integral Equation

Suppose we want to find a solution $y \in C[0, 1]$ of the integral equation

$$y(x) - \int_0^1 K(x,t) f(t, y(t)) \mathrm{d}t = a(x), \quad 0 \le x \le 1, \tag{4.9}$$

where K, the "kernel" of the equation, is a given (integrable) function on $[0, 1] \times [0, 1]$, f a given function on $[0, 1] \times \mathbb{R}$, typically nonlinear in the second argument, and a also a given function on $[0, 1]$. One way to approximately solve (4.9) is to approximate the kernel by a degenerate kernel,

$$K(x,t) \approx k_n(x,t), \quad k_n(x,t) = \sum_{i=1}^{n} c_i(t) \pi_i(x). \tag{4.10}$$

We may think of the degenerate kernel as coming from truncating (to n terms) an infinite expansion of $K(x,t)$ in a system of basis functions $\{\pi_i(x)\}$, with coefficients c_i depending only on t. Replacing K in (4.9) by k_n then yields an approximate solution y_n, which is to satisfy

$$y_n(x) - \int_0^1 k_n(x,t) f(t, y_n(t)) \mathrm{d}t = a(x), \quad 0 \le x \le 1. \tag{4.11}$$

If we substitute (4.10) into (4.11) and define

$$\alpha_i = \int_0^1 c_i(t) f(t, y_n(t)) \mathrm{d}t, \quad i = 1, 2, \ldots, n, \tag{4.12}$$

we can write y_n in the form

$$y_n(x) = a(x) + \sum_{i=1}^{n} \alpha_i \pi_i(x). \tag{4.13}$$

All that remains to be done is to compute the coefficients α_i in this representation. It is at this point where one is led to a system of nonlinear equations. Indeed, by (4.12), the α_i must satisfy

$$\alpha_i - \int_0^1 c_i(t) f(t, a(t) + \sum_{j=1}^{n} \alpha_j \pi_j(t))\mathrm{d}t = 0, \quad i = 1, 2, \dots, n, \tag{4.14}$$

where the left-hand sides are functions f_i of $\alpha_1, \alpha_2, \dots, \alpha_n$ which can be evaluated by numerical integration. It is seen how techniques of approximation (to obtain k_n) discussed in Chap. 2, and techniques of integration (to compute the integrals in (4.14) discussed in Chap. 3, usefully combine to provide an approximate solution of (4.9).

4.1.4 s-Orthogonal Polynomials

In Ex. 20(b) of Chap. 3 we encountered an instance ($s = 1$) of "power orthogonality," that is, a (monic) polynomial π_n of degree n satisfying

$$\int_{\mathbb{R}} [\pi_n(t)]^{2s+1} p(t)\mathrm{d}\lambda(t) = 0, \quad \text{all } \ p \in \mathbb{P}_{n-1}. \tag{4.15}$$

This is called an s-orthogonal polynomial relative to the (positive) measure $\mathrm{d}\lambda$. We can reinterpret power orthogonality as ordinary orthogonality

$$(\pi_n, p)_{\mathrm{d}\lambda_n^s} := \int_{\mathbb{R}} \pi_n(t) p(t) \pi_n^{2s}(t)\mathrm{d}\lambda(t) = 0,$$

but relative to the (positive) measure $\mathrm{d}\lambda_n^s(t) = \pi_n^{2s}(t)\mathrm{d}\lambda(t)$ depending on π_n. Thus, orthogonality is defined implicitly. The point, however, is that if we denote by $\{\pi_{k,n}\}_{k=0}^n$ the first $n+1$ orthogonal polynomials relative to $\mathrm{d}\lambda_n^s$, we have $\pi_n = \pi_{n,n}$, and we can formally generate $\pi_{n,n}$ by a three-term recurrence relation:

$$\pi_{k+1,n}(t) = (t - \alpha_k)\pi_{k,n} - \beta_k \pi_{k-1,n}, \quad k = 0, 1, \dots, n-1, \tag{4.16}$$

where $\pi_{-1,n}(t) = 0$, $\pi_{0,n}(t) = 1$. The coefficients $\alpha_0, \alpha_1, \dots, \alpha_{n-1}$; β_0, $\beta_1, \dots,$ β_{n-1} are unknown and must be determined. Here is how a system of $2n$ nonlinear equations can be constructed for them:

From Chap. 2, (2.40) and (2.41), one has

$$\alpha_k = \frac{(t\pi_{k,n}, \pi_{k,n})_{\mathrm{d}\lambda_n^s}}{(\pi_{k,n}, \pi_{k,n})_{\mathrm{d}\lambda_n^s}}, \quad k = 0, 1, \dots, n-1,$$

$$\beta_0 = (1,1)_{\mathrm{d}\lambda_n^s}, \ \beta_k = \frac{(\pi_{k,n}, \ \pi_{k,n})_{\mathrm{d}\lambda_n^s}}{(\pi_{k-1,n}, \pi_{k-1,n})_{\mathrm{d}\lambda_n^s}}, \quad k = 1, \dots, n-1.$$

Consequently, clearing denominators,

$$\begin{aligned} f_0 &:= \beta_0 - \int_{\mathbb{R}} \pi_{n,n}^{2s}(t)\mathrm{d}\lambda(t) = 0, \\ f_{2\nu+1} &:= \int_{\mathbb{R}} (\alpha_\nu - t)\pi_{\nu,n}^2(t)\pi_{n,n}^{2s}(t)\mathrm{d}\lambda(t) = 0, \quad \nu = 0, 1, \dots, n-1, \\ f_{2\nu} &:= \int_{\mathbb{R}} [\beta_\nu \pi_{\nu-1,n}^2(t) - \pi_{\nu,n}^2(t)]\pi_{n,n}^{2s}(t)\mathrm{d}\lambda(t) = 0, \quad \nu = 1, \dots, n-1. \end{aligned} \tag{4.17}$$

Each of the $\pi_{\nu,n}$, $\nu = 1, 2, \dots, n$, depends on $\alpha_0, \dots, \alpha_{\nu-1}; \beta_1, \dots, \beta_{\nu-1}$ via the three-term recurrence relation (4.16). Therefore, we have $2n$ equations depending nonlinearly on the $2n$ unknowns $\alpha_0, \dots, \alpha_{n-1}; \beta_0, \dots, \beta_{n-1}$:

$$\boldsymbol{f}(\boldsymbol{\rho}) = \mathbf{0}, \quad \boldsymbol{\rho}^{\mathrm{T}} = [\alpha_0, \dots, \alpha_{n-1}; \beta_0, \dots, \beta_{n-1}].$$

Since the components of $\boldsymbol{f}$ are integrals of polynomials of degree at most $2(s+1)n-1$, they can be computed exactly by an $(s+1)n$-point Gauss quadrature rule relative to the measure $\mathrm{d}\lambda$ (cf. MA 9).

4.2 Iteration, Convergence, and Efficiency

Even the simplest of nonlinear equations – for example, algebraic equations – are known to not admit solutions that are expressible rationally in terms of the data. It is therefore impossible, in general, to compute roots of nonlinear equations in a finite number of arithmetic operations. What is required is an *iterative method*, that is, a procedure that generates an infinite sequence of approximations, $\{x_n\}_{n=0}^{\infty}$, such that

$$\lim_{n\to\infty} x_n = \alpha \tag{4.18}$$

for some root α of the equation. In case of a system of equations, both x_n and α are vectors of appropriate dimension, and convergence is to be understood in the sense of componentwise convergence.

Although convergence of an iterative process is certainly desirable, it takes more than just convergence to make it practical. What one wants is fast convergence. A basic concept to measure the speed of convergence is the *order of convergence*.

Definition 4.2.1. *Linear convergence*. One says that x_n converges to α (at least) linearly if

$$|x_n - \alpha| \le \varepsilon_n, \tag{4.19}$$

where $\{\varepsilon_n\}$ is a positive sequence satisfying

$$\lim_{n\to\infty} \frac{\varepsilon_{n+1}}{\varepsilon_n} = c, \quad 0 < c < 1. \tag{4.20}$$

If (4.19) and (4.20) hold with the inequality in (4.19) replaced by an equality, then c is called the *asymptotic error constant*.

The phrase "at least" in this definition relates to the fact that we have only inequality in (4.19), which in practice is all we can usually ascertain. So, strictly speaking, it is the *bounds* ε_n that converge linearly, meaning that eventually (e.g., for n large enough) each of these error bounds is approximately a constant fraction of the preceding one.

For linearly convergent sequences there is a simple device, called *Aitken's* Δ^2-*process*, that can be used to speed up convergence. One defines

$$x_n' = x_n - \frac{(\Delta x_n)^2}{\Delta^2 x_n}, \tag{4.21}$$

where $\Delta x_n = x_{n+1} - x_n$, $\Delta^2 x_n = \Delta(\Delta x_n) = x_{n+2} - 2x_{n+1} + x_n$. The sequence $\{x_n'\}$ then converges faster than $\{x_n\}$ in the sense that

$$\frac{x_n' - \alpha}{x_n - \alpha} \to 0 \text{ as } n \to \infty, \tag{4.22}$$

where $\alpha = \lim_{n\to\infty} x_n$ (cf. Ex. 6).

Definition 4.2.2. *Convergence of order p*. One says that x_n converges to α with (at least) order $p \ge 1$ if (4.19) holds with

$$\lim_{n\to\infty} \frac{\varepsilon_{n+1}}{\varepsilon_n^p} = c, \quad c > 0. \tag{4.23}$$

(If $p = 1$, one must assume, in addition, $c < 1$.)

Thus, convergence of order 1 is the same as linear convergence, whereas convergence of order $p > 1$ is faster. Note that in this latter case there is no restriction on the constant c: once ε_n is small enough, it will be the exponent p that takes care of convergence. The constant c is again referred to as the asymptotic error constant if we have equality in (4.19).

The same definitions apply also to vector-valued sequences; one only needs to replace absolute values in (4.19) by (any) vector norm.

The classification of convergence with respect to order is still rather crude, as there are types of convergence that "fall between the cracks." Thus, a sequence $\{\varepsilon_n\}$ may converge to 0 more slowly than linearly, for example, such that $c = 1$ in (4.20). We may call this type of convergence *sublinear*. Likewise, $c = 0$ in (4.20) gives rise to *superlinear* convergence, if (4.23) does not hold for any $p > 1$ (cf. also Ex. 4).

It is instructive to examine the behavior of ε_n if instead of the limit relations (4.20) and (4.23) we had strict equality from some n on, say,

$$\frac{\varepsilon_{n+1}}{\varepsilon_n^p} = c, \quad n = n_0, n_0+1, n_0+2, \dots. \tag{4.24}$$

For n_0 large enough, this is almost true. A simple induction argument then shows that

$$\varepsilon_{n_0+k} = c^{\frac{p^k-1}{p-1}}\, \varepsilon_{n_0}^{p^k}, \quad k = 0, 1, 2, \dots, \tag{4.25}$$

which certainly holds for $p > 1$, but also for $p = 1$ in the limit as $p \downarrow 1$:

$$\varepsilon_{n_0+k} = c^k \varepsilon_{n_0}, \quad k = 0, 1, 2, \dots\ (p = 1). \tag{4.26}$$

Assuming then ε_{n_0} sufficiently small so that the approximation x_{n_0} has several correct decimal digits, we write $\varepsilon_{n_0+k} = 10^{-\delta_k}\varepsilon_{n_0}$. Then δ_k, according to (4.19), approximately represents the number of additional correct decimal digits in the approximation x_{n_0+k} (as compared to x_{n_0}). Taking logarithms in (4.26) and (4.25) gives

$$\delta_k = \begin{cases} k \log \dfrac{1}{c} & \text{if } p = 1, \\[2ex] p^k \left[\dfrac{1-p^{-k}}{p-1} \log \dfrac{1}{c} + (1-p^{-k}) \log \dfrac{1}{\varepsilon_{n_0}} \right] & \text{if } p > 1; \end{cases}$$

hence, as $k \to \infty$,

$$\delta_k \sim C_1 k \ \ (p = 1), \quad \delta_k \sim C_p p^k \ \ (p > 1), \tag{4.27}$$

where $C_1 = \log \frac{1}{c} > 0$ if $p = 1$ and $C_p = \frac{1}{p-1} \log \frac{1}{c} + \log \frac{1}{\varepsilon_{n_0}}$. (We assume here that n_0 is large enough, and hence ε_{n_0} small enough, to have $C_p > 0$.) This shows that the number of correct decimal digits increases linearly with k, when $p = 1$, but exponentially when $p > 1$. In the latter case, $\delta_{k+1}/\delta_k \sim p$, meaning that ultimately (for large k) the number of correct decimal digits increases, per iteration step, by a factor of p.

If each iteration step requires m units of work (a "unit of work" typically is the work involved in computing a function value or a value of one of its derivatives), then the *efficiency index* of the iteration may be defined by $\lim_{k\to\infty}[\delta_{k+1}/\delta_k]^{1/m} = p^{1/m}$. It provides a common basis on which to compare different iterative methods with one another (cf. Ex. 7). Methods that converge linearly have efficiency index 1.

Practical computation requires the employment of a *stopping rule* that terminates the iteration once the desired accuracy is (or is believed to be) attained. Ideally, one stops as soon as the absolute value (or norm) of the error $x_n - \alpha$ is smaller than a prescribed error tolerance. Since α is not known, one commonly replaces $x_n - \alpha$ by $x_n - x_{n-1}$ and requires

$$\|x_n - x_{n-1}\| \le \text{tol}, \tag{4.28}$$

where

$$\text{tol} = \|x_n\|\epsilon_r + \epsilon_a \tag{4.29}$$

with ϵ_r, ϵ_a prescribed tolerances. As a safety measure, one might require (4.28) not just for one but for a few consecutive values of n. Choosing $\epsilon_\mathrm{r} = 0$ or $\epsilon_\mathrm{a} = 0$ will make (4.29) an absolute resp. relative error tolerance. It is prudent, however, to use a "mixed error tolerance," say, $\epsilon_\mathrm{r} = \epsilon_\mathrm{a} = \epsilon$. Then, if $\|x_n\|$ is small or moderately large, one effectively controls the absolute error, whereas for $\|x_n\|$ very large, it is in effect the relative error that is controlled.

4.3 The Methods of Bisection and Sturm Sequences

Both these methods generate a sequence of nested intervals $[a_n, b_n]$, $n = 0, 1, 2, \dots$, whereby each interval is guaranteed to contain at least one root of the equation. As $n \to \infty$, the length of these intervals tends to 0, so that in the limit exactly one (isolated) root is captured. The first method applies to any equation (4.1) with f a continuous function, but has no built-in control of steering the iteration to any particular (real) root if there is more than one. The second method does have such a control mechanism, but applies only to a restricted class of equations, for example, the characteristic equation of a symmetric tridiagonal matrix.

4.3.1 Bisection Method

We assume that two numbers a, b with $a < b$ are known such that

$$f \in C[a,b], \quad f(a) < 0, \quad f(b) > 0. \tag{4.30}$$

What is essential here is that f has opposite signs at the endpoints of $[a, b]$; the particular sign combination in (4.30) is not essential as it can always be obtained, if necessary, by multiplying f by -1. Assumptions (4.30), in particular, guarantee that f has at least one 0 in (a, b).

By repeatedly bisecting the interval and discarding endpoints in such a manner that the sign property in (4.30) remains preserved, it is possible to generate a sequence of nested intervals whose lengths are continuously halved and each of which contains a zero of f.

Specifically, the procedure is as follows. Define $a_1 = a$, $b_1 = b$. Then

```
for $n = 1, 2, 3, \ldots$ do
    $x_n = \frac{1}{2}(a_n + b_n)$
    if $f(x_n) < 0$ then $a_{n+1} = x_n,\ b_{n+1} = b_n$ else
        $a_{n+1} = a_n,\ b_{n+1} = x_n$.
```

Since $b_n - a_n = 2^{-(n-1)}(b - a)$, $n = 1, 2, 3, \ldots$, and x_n is the midpoint of $[a_n, b_n]$, if α is the root eventually captured, we have

$$|x_n - \alpha| \le \frac{1}{2}(b_n - a_n) = \frac{b-a}{2^n}. \tag{4.31}$$

Thus, (4.19) holds with $\varepsilon_n = 2^{-n}(b - a)$ and

$$\frac{\varepsilon_{n+1}}{\varepsilon_n} = \frac{1}{2}, \quad \text{all } n. \tag{4.32}$$

This shows that the bisection method converges (at least) linearly with asymptotic error constant (for the bound ε_n) equal to $\frac{1}{2}$.

Given an (absolute) error tolerance tol > 0, the error in (4.31) will be less than or equal to tol if

$$\frac{b-a}{2^n} \le \text{tol}.$$

Solved explicitly for n, this will be satisfied if

$$n = \left\lceil \frac{\log \frac{b-a}{\text{tol}}}{\log 2} \right\rceil, \tag{4.33}$$

where $\lceil x \rceil$ denotes the "ceiling" of x (i.e., the smallest integer $\ge x$). Thus, we know a priori how many steps are necessary to achieve a prescribed accuracy.

It should be noted that when $f(x_n)$ approaches the level of machine precision, its sign may be computed incorrectly, hence the wrong half of the current interval may be chosen. Normally, this should be of little concern since by this time, the interval has already become sufficiently small. In this sense, bisection is a method that is robust at the level of machine precision. Problems may occur when f is very flat near the root α, in which case, however, the root is numerically not well determined (cf. MA 1).

When implementing the procedure on a computer, it is clearly unnecessary to provide arrays to store a_n, b_n, x_n; one simply keeps overwriting. Assuming a and b have been initialized and tol assigned, one could use the following Matlab program (in symbolic/variable-precision mode to allow for high accuracies):

```
%SBISEC  Symbolic bisection method
%
function [ntol,x]=sbisec(dig,a,b,tol)
ntol=ceil(log((b-a)/tol)/log(2));
digits(dig);
a=vpa(a); b=vpa(b);
for n=1:ntol
  x=vpa((a+b)/2);
  fx=subs(f(x));
  if fx<0
    a=x;
  else
    b=x;
  end
end

function y=f(x)
y=cos(x)*cosh(x)-1;
```

As an example, we run the program to compute the smallest positive root of (4.2) that is, of $f(x) = 0$ with $f(x) = \cos x \cosh x - 1$ (see the subfunction appended to sbisec.m). By taking $a = \frac{3}{2}\pi$, $b = 2\pi$ (cf. Fig. 4.1), we enclose exactly one root, α_1, from the start, and bisection is guaranteed to converge to α_1. This is implemented in the program TABLE4_3_1.m for tolerances tol $= \frac{1}{2} \times 10^{-7}$, $\frac{1}{2} \times 10^{-15}$, and $\frac{1}{2} \times 10^{-33}$; the results, rounded to the appropriate number of digits,

```
%TABLE4_3_1  Bisection method to solve Eq.(4.2)
%
a=3*pi/2; b=2*pi; dig=35;
for tol=[.5e-7 .5e-15 .5e-33]
  [n,x]=sbisec(dig,a,b,tol)
end
```

Table 4.1 The bisection method applied to (4.2)

n	α_1
25	4.7300408
52	4.730040744862704
112	4.730040744862704026024048100833885

are shown in Table 4.1[1]. As can be seen, bisection requires a fair amount of work (over a 100 iterations) to obtain α_1 to high precision. Had we run the program with initial interval $[a, b]$, $a = \frac{3}{2}\pi$, $b = 4\pi$, which contains three roots, we would have converged to the third root, α_2, with about the same amount of work.

4.3.2 *Method of Sturm Sequences*

There are situations in which it is desirable to be able to select one particular root among many and have the iterative scheme converge to it. This is the case, for example, in orthogonal polynomials, where we know that all zeros are real and distinct (cf. Chap. 3, Sect. 3.2.3(a)). It may well be that we are interested in the second-largest or third-largest zero and should be able to compute it without computing any of the others. This is indeed possible if we combine bisection with the theorem of Sturm.[2]

Thus, consider

$$f(x) = \pi_d(x), \tag{4.34}$$

where π_d is a polynomial of degree d orthogonal with respect to some positive measure. We know (cf. Chap. 3, Sect. 3.2.3(e)) that π_d is the characteristic polynomial of a symmetric tridiagonal matrix and can be computed recursively by a three-term recurrence relation

$$\begin{aligned} \pi_0(x) &= 1, \quad \pi_1(x) = x - \alpha_0, \\ \pi_{k+1}(x) &= (x - \alpha_k)\pi_k(x) - \beta_k \pi_{k-1}(x), \quad k = 1, 2, \ldots, d-1, \end{aligned} \tag{4.35}$$

with all β_k positive. Recursion (4.35) not only is useful to compute $\pi_d(x)$ for any fixed x, but also has the following interesting property due to Sturm: *Let $\sigma(x)$ be the number of sign changes* (zeros do not count) *in the sequence of numbers*

[1] The 33-digit result given in the first edition of this text is erroneous, being accurate only to 19 digits after the decimal point.

[2] Jacques Charles François Sturm(1803–1855), a Swiss analyst and theoretical physicist of Alsatian parentage, is best known for his theorem on Sturm sequences, discovered in 1829, and his theory of Sturm–Liouville differential equations, published in 1834, which earned him the Grand Prix des Sciences Mathématiques. He also contributed significantly to differential and projective geometry. A member of the French Academy of Sciences since 1836, he succeeded Poisson in the chair of mechanics at the École Polytechnique in Paris in 1839.

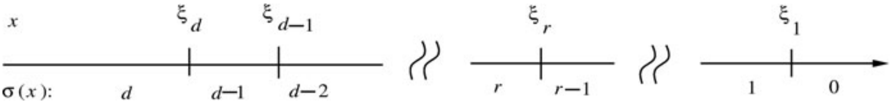


Fig. 4.2 Sturm's theorem

$$\pi_d(x),\ \pi_{d-1}(x), \ldots, \pi_1(x),\ \pi_0(x). \tag{4.36}$$

Then, for any two numbers a, b with $a < b$, the number of real zeros of π_d in the interval $a < x \le b$ is equal to $\sigma(a) - \sigma(b)$.

Since $\pi_k(x) = x^k + \cdots$, it is clear that $\sigma(-\infty) = d$, $\sigma(+\infty) = 0$, so that indeed the number of real zeros of π_d is $\sigma(-\infty) - \sigma(+\infty) = d$. Moreover, if $\xi_1 > \xi_2 > \cdots > \xi_d$ denote the zeros of π_d in decreasing order, we have the behavior of $\sigma(x)$ as shown in Fig. 4.2.

It is now easy to see that

$$\sigma(x) \le r - 1 \text{ iff } x \ge \xi_r. \tag{4.37}$$

Indeed, suppose that $x \ge \xi_r$. Then $\{\#\text{zeros} \le x\} \ge d + 1 - r$; hence, by Sturm's theorem, $\sigma(-\infty) - \sigma(x) = d - \sigma(x) = \{\#\text{zeros} \le x\} \ge d + 1 - r$; that is, $\sigma(x) \le r - 1$. Conversely, if $\sigma(x) \le r - 1$, then, again by Sturm's theorem, $\{\#\text{zeros} \le x\} = d - \sigma(x) \ge d + 1 - r$, which implies $x \ge \xi_r$ (cf. Fig. 4.2).

The basic idea now is to control the bisection process not, as before, by checking the sign of $\pi_d(x)$, but rather, by checking the inequality (4.37) to see whether we are on the right or left side of the zero ξ_r. To initialize the procedure, we need two values $a_1 = a$, $b_1 = b$ such that $a < \xi_d$ and $b > \xi_1$. These are trivially obtained as the endpoints of the interval of orthogonality for π_d, if it is finite. More generally, one can apply Gershgorin's theorem[3] to the Jacobi matrix $\boldsymbol{J}_d$ associated with (4.35) (cf. Chap. 3, Sect. 3.2.3(e)) by recalling that the zeros of π_d are precisely the eigenvalues of $\boldsymbol{J}_d$. In this way, a can be chosen to be the smallest and b the largest of the numbers $\alpha_0 \pm \sqrt{\beta_1}$, $\alpha_1 \pm (\sqrt{\beta_1} + \sqrt{\beta_2}), \ldots, \alpha_{d-2} \pm (\sqrt{\beta_{d-2}} + \sqrt{\beta_{d-1}})$, $\alpha_{d-1} \pm \sqrt{\beta_{d-1}}$. The method of Sturm sequences then proceeds as follows, for any given r with $1 \le r \le d$:

```
for $n = 1, 2, 3, \ldots$ do
    $x_n = \frac{1}{2}(a_n + b_n)$
    if $\sigma(x_n) > r - 1$ then $a_{n+1} = x_n,\ b_{n+1} = b_n$ else
        $a_{n+1} = a_n,\ b_{n+1} = x_n.$
```

[3] Gershgorin's theorem states that the eigenvalues of a matrix $\boldsymbol{A} = [a_{ij}]$ of order d are located in the union of the disks $\{z \in \mathbb{C} : |z - a_{ii}| \le r_i\}$, $i = 1, 2, \ldots d$, where $r_i = \sum_{j \ne i} |a_{ij}|$.

Since initially $\sigma(a) = d > r-1, \sigma(b) = 0 \le r-1$, it follows by construction that

$$\sigma(a_n) > r-1, \ \ \sigma(b_n) \le r-1, \quad \text{all } n = 1, 2, 3, \dots, \tag{4.38}$$

meaning that $\xi_r \in [a_n, b_n]$ for all $n = 1, 2, 3, \dots$. Moreover, as in the bisection method, $b_n - a_n = 2^{-(n-1)}(b-a)$, so that $|x_n - \xi_r| \le \varepsilon_n$ with $\varepsilon_n = 2^{-n}(b-a)$. The method converges (at least) linearly to the root ξ_r. A computer implementation can be modeled after the one for the bisection method by modifying the `if--else` statement appropriately.

4.4 Method of False Position

As in the method of bisection, we assume two numbers $a < b$ such that

$$f \in C[a,b], \quad f(a)f(b) < 0 \tag{4.39}$$

and generate a sequence of nested intervals $[a_n, b_n]$, $n = 1, 2, 3, \dots$, with $a_1 = a$, $b_1 = b$, such that $f(a_n)f(b_n) < 0$. Unlike the bisection method, however, we are not taking the midpoint of $[a_n, b_n]$ to determine the next interval, but rather the solution $x = x_n$ of the linear equation

$$p_1(f; a_n, b_n; x) = 0, \tag{4.40}$$

where $p_1(f; a_n, b_n; \cdot\,)$ is the linear interpolant of f at a_n and b_n. This would appear to be more flexible than bisection, as x_n will come to lie closer to the endpoint at which $|f|$ is smaller. Also, if f is a linear function, we obtain the root in one step rather than in an infinite number of steps. This explains the somewhat strange name given to this method (cf. Notes to Section 4.4).

More explicitly, the method proceeds as follows: define $a_1 = a$, $b_1 = b$. Then,

```
for $n = 1, 2, 3, \dots$ do
    $x_n = a_n - \frac{a_n - b_n}{f(a_n) - f(b_n)} f(a_n)$
    if $f(x_n)f(a_n) > 0$ then $a_{n+1} = x_n$, $b_{n+1} = b_n$ else
        $a_{n+1} = a_n$, $b_{n+1} = x_n$.
```

One may terminate the iteration as soon as $b_n - a_n \le$ tol or $|f(x_n)| \le$ tol, where tol is a prescribed error tolerance.

As in the bisection method, when implementing the method on a computer, the a and b can be overwritten. On the other hand, it is no longer known a priori how many iterations it takes to achieve the desired accuracy. It is prudent, then, to put a limit on the number of iterations. An implementation in (symbolic/variable-precision) Matlab may look as follows.

```
%SFALSEPOS  Symbolic method of false position
%
function [n,x]=sfalsepos(dig,a,b,tol,nmax)
n=0;
digits(dig);
a=vpa(a); b=vpa(b);
fa=f(a); fb=f(b);
while subs(b-a)>=tol
  n=n+1;
  if n>nmax
    fprintf('n exceeds nmax')
    return
  end
  x=a-(a-b)*fa/(fa-fb);
  fx=f(x);
  if abs(subs(fx))<tol, return; end
  if subs(fx*fa)>0
    a=x; fa=fx;
  else
    b=x; fb=fx;
  end
end
```

The convergence behavior is most easily analyzed if we assume that f is convex or concave on $[a,b]$. To fix ideas, suppose f is convex, say,

$$f''(x) > 0 \quad \text{for } a \le x \le b \quad \text{and} \quad f(a) < 0,\ f(b) > 0. \tag{4.41}$$

Then f has exactly one zero, α, in $[a,b]$. Moreover, the secant connecting $f(a)$ and $f(b)$ lies entirely above the graph of $y = f(x)$ and hence intersects the real line to the left of α. This will be the case for all subsequent secants, which means that the point $x = b$ remains fixed while the other endpoint a gets continuously updated, producing a monotonically increasing sequence of approximations defined by

$$x_{n+1} = x_n - \frac{x_n - b}{f(x_n) - f(b)}\, f(x_n), \quad n = 1, 2, 3, \ldots, \tag{4.42}$$

where $x_1 = a$. Any such sequence, being bounded from above by α, necessarily converges, and letting $n \to \infty$ in (4.42) shows immediately that $f(x_n) \to 0$, that is, $x_n \to \alpha$. To determine the speed of convergence, we subtract α from both sides of (4.42) and use the fact that $f(\alpha) = 0$:

$$x_{n+1} - \alpha = x_n - \alpha - \frac{x_n - b}{f(x_n) - f(b)}\,[f(x_n) - f(\alpha)].$$

Now divide by $x_n - \alpha$ to get

$$\frac{x_{n+1} - \alpha}{x_n - \alpha} = 1 - \frac{x_n - b}{f(x_n) - f(b)}\,\frac{f(x_n) - f(\alpha)}{x_n - \alpha}\,.$$

Letting here $n \to \infty$ and using the fact that $x_n \to \alpha$, we obtain

$$\lim_{n\to\infty} \frac{x_{n+1} - \alpha}{x_n - \alpha} = 1 - (b - \alpha)\,\frac{f'(\alpha)}{f(b)}. \tag{4.43}$$

Thus, we have linear convergence with asymptotic error constant equal to

$$c = 1 - (b - \alpha)\,\frac{f'(\alpha)}{f(b)}. \tag{4.44}$$

It is clear on geometric grounds that $0 < c < 1$ under the assumptions made. An analogous result will hold, with a constant $|c| < 1$, provided f is either convex or concave on $[a, b]$ and has opposite signs at the endpoints a and b. One of these endpoints then remains fixed while the other moves monotonically to the root α.

If f does not satisfy these convexity properties on the whole interval but is such that $f \in C^2[a, b]$ and $f''(\alpha) \neq 0$ at the root α eventually approached, then the convergence behavior described sets in for n large enough, since f'' has constant sign in a neighborhood of α, and x_n will eventually come to lie in this neighborhood.

The fact that one of the "false positions" remains at a fixed point from some n on may speak against this method, especially if this occurs early in the iteration, or even from the beginning, as under the assumptions (4.41) made previously. Thus, in the case of (4.2), for example, when $a = \frac{3}{2}\pi$ and $b = 2\pi$, we have $f''(x) = -2\sin x \sinh x > 0$ on $[a, b]$, and $f(a) = -1$, $f(b) = \cosh(2\pi) - 1 > 0$, so that we are precisely in a case where (4.41) holds. Accordingly, we have found that to compute the root α_1 to within a tolerance of 0.5×10^{-7}, 0.5×10^{-15}, and 0.5×10^{-33}, we now need respectively, 42, 87, and 188 iterations, as compared to 25, 52, and 112 in the case of the bisection method.

Exceptionally slow convergence is likely to occur when f is very flat near α, the point a is nearby, and b further away. In this case, b will typically remain fixed while a is slowly creeping toward α (cf. MA 1).

4.5 Secant Method

The secant method is a simple variant of the method of false position in which it is no longer required that the function f has opposite signs at the end points of each interval generated, not even the initial interval. In other words, one starts with two arbitrary initial approximations $x_0 \neq x_1$ and continues with

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}\, f(x_n), \quad n = 1, 2, 3, \ldots . \tag{4.45}$$

This precludes the formation of a fixed false position, as in the method of false positions, and hence suggests potentially faster convergence. On the other hand, we can no longer be sure that each interval $[x_{n-1}, x_n]$ contains at least one root. It will turn out that, if the method converges, it does so with an order of convergence larger than 1 (but less than 2). However, it converges only "locally," that is, only if the initial approximations x_0, x_1 are sufficiently close to a root.

This can be seen by relating the three consecutive errors of x_{n+1}, x_n, and x_{n-1} as follows. Subtract α on both sides of (4.45), and use $f(\alpha) = 0$, to get

$$\begin{aligned} x_{n+1} - \alpha &= x_n - \alpha - \frac{f(x_n)}{[x_{n-1}, x_n]f} \\ &= (x_n - \alpha)\left(1 - \frac{f(x_n) - f(\alpha)}{(x_n - \alpha)[x_{n-1}, x_n]f}\right) \\ &= (x_n - \alpha)\left(1 - \frac{[x_n, \alpha]f}{[x_{n-1}, x_n]f}\right) \\ &= (x_n - \alpha)\, \frac{[x_{n-1}, x_n]f - [x_n, \alpha]f}{[x_{n-1}, x_n]f}; \end{aligned}$$

hence, by the definition of divided differences,

$$x_{n+1} - \alpha = (x_n - \alpha)(x_{n-1} - \alpha)\, \frac{[x_{n-1}, x_n, \alpha]f}{[x_{n-1}, x_n]f}, \quad n = 1, 2, \ldots . \tag{4.46}$$

This is the fundamental relation holding between three consecutive errors.

From (4.46) it follows immediately that if α is a simple root,

$$f(\alpha) = 0, \quad f'(\alpha) \neq 0, \tag{4.47}$$

and if $x_n \to \alpha$, then convergence is faster than linear, at least if $f \in C^2$ near α. Indeed,

$$\lim_{n\to\infty} \frac{x_{n+1} - \alpha}{x_n - \alpha} = 0,$$

since the divided differences in the numerator and denominator of (4.46) converge to $\frac{1}{2}\,f''(\alpha)$ and $f'(\alpha)$, respectively. But just how fast is convergence?

We can discover the order of convergence (assuming convergence) by a simple heuristic argument: we replace the ratio of divided differences in (4.46) by a constant, which is almost true if n is large. Letting then $e_k = |x_k - \alpha|$, we have

$$e_{n+1} = e_n e_{n-1} \cdot C, \quad C > 0.$$

Multiplying both sides by C and defining $E_n = Ce_n$ gives

$$E_{n+1} = E_n E_{n-1}, \quad E_n \to 0.$$

Taking logarithms on both sides, and defining $y_n = \log \frac{1}{E_n}$, we get

$$y_{n+1} = y_n + y_{n-1}, \tag{4.48}$$

the well-known difference equation for the Fibonacci sequence. Its characteristic equation is $t^2 - t - 1 = 0$, which has the two roots t_1, t_2 with

$$t_1 = \frac{1}{2}\,(1+\sqrt{5}), \quad t_2 = \frac{1}{2}\,(1-\sqrt{5}),$$

and $t_1 > 1$, $|t_2| < 1$. The general solution of (4.48), therefore, is

$$y_n = c_1 t_1^n + c_2 t_2^n, \quad c_1,\ c_2 \text{ constant.}$$

Since $y_n \to \infty$, we have $c_1 \neq 0$ and $y_n \sim c_1 t_1^n$ as $n \to \infty$, which translates into $\frac{1}{E_n} \sim \mathrm{e}^{c_1 t_1^n}$, $\frac{1}{e_n} \sim C\mathrm{e}^{c_1 t_1^n}$, and thus

$$\frac{e_{n+1}}{e_n^{t_1}} \sim \frac{C^{t_1}\mathrm{e}^{c_1 t_1^n \cdot t_1}}{C\mathrm{e}^{c_1 t_1^{n+1}}} = C^{t_1 - 1}, \quad n \to \infty.$$

The order of convergence, therefore, is $t_1 = \frac{1}{2}\,(1+\sqrt{5}) = 1.61803\ldots$ (the golden ratio).

We now give a rigorous proof of this and begin with a proof of *local convergence*.

Theorem 4.5.1. *Let α be a simple zero of f. Let $I_\varepsilon = \{x \in \mathbb{R} :\ |x-\alpha| \le \varepsilon\}$ and assume $f \in C^2[I_\varepsilon]$. Define, for sufficiently small ε,*

$$M(\varepsilon) = \max_{\substack{s \in I_\varepsilon \\ t \in I_\varepsilon}} \left|\frac{f''(s)}{2f'(t)}\right|. \tag{4.49}$$

Assume ε so small that

$$\varepsilon M(\varepsilon) < 1. \tag{4.50}$$

Then the secant method converges to the unique root $\alpha \in I_\varepsilon$ *for any starting values* $x_0 \neq x_1$ *with* $x_0 \in I_\varepsilon$, $x_1 \in I_\varepsilon$.

Note that $\lim_{\varepsilon \to 0} M(\varepsilon) = \left| \frac{f''(\alpha)}{2f'(\alpha)} \right| < \infty$, so that (4.50) can certainly be satisfied for ε small enough. The local nature of convergence is thus quantified by the requirement $x_0, x_1 \in I_\varepsilon$.

Proof of Theorem 4.5.1. First of all, observe that α is the only zero of f in I_ε. This follows from Taylor's formula applied at $x = \alpha$:

$$f(x) = f(\alpha) + (x - \alpha) f'(\alpha) + \frac{(x - \alpha)^2}{2} f''(\xi),$$

where $f(\alpha) = 0$ and ξ is between x and α. Thus, if $x \in I_\varepsilon$, then so is ξ, and we have

$$f(x) = (x - \alpha) f'(\alpha) \left\{ 1 + \frac{x - \alpha}{2} \frac{f''(\xi)}{f'(\alpha)} \right\}, \quad \xi \in I_\varepsilon.$$

Here, if $x \neq \alpha$, all three factors are different from zero, the last one since by assumption

$$\left| \frac{x - \alpha}{2} \frac{f''(\xi)}{f'(\alpha)} \right| \leq \varepsilon M(\varepsilon) < 1.$$

Thus, f on I_ε can only vanish at $x = \alpha$.

Next we show that all $x_n \in I_\varepsilon$ and two consecutive iterates are distinct, unless $f(x_n) = 0$ for some n, in which case $x_n = \alpha$ and the method converges in a finite number of steps. We prove this by induction: assume that $x_{n-1} \in I_\varepsilon$, $x_n \in I_\varepsilon$ for some n and $x_n \neq x_{n-1}$. (By assumption, this is true for $n = 1$.) Then, from known properties of divided differences, and by our assumption that $f \in C^2[I_\varepsilon]$, we have

$$[x_{n-1}, x_n]f = f'(\xi_1), \; [x_{n-1}, x_n, \alpha]f = \frac{1}{2} f''(\xi_2), \quad \xi_i \in I_\varepsilon, \; i = 1, 2.$$

Therefore, by (4.46),

$$|x_{n+1} - \alpha| \leq \varepsilon^2 \left| \frac{f''(\xi_2)}{2f'(\xi_1)} \right| \leq \varepsilon \cdot \varepsilon M(\varepsilon) < \varepsilon,$$

showing that $x_{n+1} \in I_\varepsilon$. Furthermore, by (4.45), $x_{n+1} \neq x_n$, unless $f(x_n) = 0$, hence $x_n = \alpha$.

Finally, again using (4.46), we have

$$|x_{n+1} - \alpha| \leq |x_n - \alpha| \, \varepsilon M(\varepsilon), \quad n = 1, 2, 3, \ldots,$$

which, applied repeatedly, yields

$$|x_n - \alpha| \le [\varepsilon M(\varepsilon)]^{n-1} |x_1 - \alpha|.$$

Since $\varepsilon M(\varepsilon) < 1$, it follows that $x_n \to \alpha$ as $n \to \infty$. □

We next prove that the order of convergence is indeed what we derived it to be heuristically.

Theorem 4.5.2. *The secant method is locally convergent with order of convergence* (*at least*) $p = \frac{1}{2}(1 + \sqrt{5}) = 1.61803\ldots$.

Proof. Local convergence is the content of Theorem 4.5.1 and so needs no further proof. We assume that $x_0, x_1 \in I_\varepsilon$, where ε satisfies (4.50), and that all x_n are distinct. Then we know that $x_n \ne \alpha$ for all n, and $x_n \to \alpha$ as $n \to \infty$.

Now the number p in the theorem satisfies

$$p^2 = p + 1. \tag{4.51}$$

From (4.46), we have

$$|x_{n+1} - \alpha| \le |x_n - \alpha| \, |x_{n-1} - \alpha| \cdot M, \tag{4.52}$$

where we write simply M for $M(\varepsilon)$. Define

$$E_n = M|x_n - \alpha|. \tag{4.53}$$

Then, multiplying (4.52) by M, we get

$$E_{n+1} \le E_n E_{n-1}.$$

It follows easily by induction that

$$E_n \le E^{p^n}, \quad E = \max\,(E_0, E_1^{1/p}). \tag{4.54}$$

Indeed, this is trivially true for $n = 0$ and $n = 1$. Suppose (4.54) holds for n as well as for $n - 1$. Then

$$E_{n+1} \le E_n E_{n-1} \le E^{p^n} E^{p^{n-1}} = E^{p^{n-1}(p+1)} = E^{p^{n-1} \cdot p^2} = E^{p^{n+1}},$$

where (4.51) has been used. This proves (4.54) for $n + 1$, and hence for all n. It now follows from (4.53) that

$$|x_n - \alpha| \le \varepsilon_n, \quad \varepsilon_n = \frac{1}{M} E^{p^n}.$$

Since $E_0 = M|x_0 - \alpha| \le \varepsilon M(\varepsilon) < 1$, and the same holds for E_1, we have $E < 1$. Now it suffices to note that

$$\frac{\varepsilon_{n+1}}{\varepsilon_n^p} = M^{p-1}\,\frac{E^{p^{n+1}}}{E^{p^n\cdot p}} = M^{p-1}, \quad \text{all } n,$$

to establish the theorem. □

The method is easily programmed for a computer. In (symbolic/variable-precision) Matlab, for example, we could use the following program:

```
%SSECANT  Symbolic secant method
%
function [n,x]=ssecant(dig,a,b,tol,nmax)
n=0;
digits(dig);
a=vpa(a); b=vpa(b);
fa=f(a);
x=b;
while subs(abs(x-a))>=tol
  n=n+1;
  if n>nmax
    fprintf(’n exceeds nmax’)
    return
  end
  b=a;
  fb=fa;
  a=x;
  fa=f(x);
  x=a-(a-b)*f(a)/(fa-fb);
end
```

It is assumed here that $a = x_0$, $b = x_1$ and that the iteration (4.45) is terminated as soon as $|x_{n+1} - x_n| <$ tol or $n >$ nmax. The routine produces not only the approximation x to the root but also the number n of iterations required to obtain it to within an error of tol.

Since only one evaluation of f is required in each iteration step (the statement `fa=f(x)` in the preceding program), the secant method has efficiency index $p = 1.61803\ldots$ (cf. Sect. 4.2).

To illustrate the considerable gain in speed attainable by the secant method, we again apply the routine to the equation (4.2) with $x_0 = \frac{3}{2}\pi$, $x_1 = 2\pi$. In view of the convexity of f, both x_2 and x_3 come to lie to the left of the root $\alpha = \alpha_1$.

Let $\varepsilon = \alpha - \frac{3}{2}\pi$ and I_ε be the interval defined in Theorem 4.5.1. Its right endpoint is $2\alpha - \frac{3}{2}\pi$, which is $<\frac{7}{4}\pi$. Indeed, $\frac{7}{4}\pi - (2\alpha - \frac{3}{2}\pi) = 2(\frac{7}{8}\pi - \alpha + \frac{3}{4}\pi) = 2(\frac{13}{8}\pi - \alpha) > 0$, since $\alpha < \frac{13}{8}\pi$ on account of $f(\frac{13}{8}\pi) = 30.545\ldots > 0$. On the interval $[\frac{3}{2}\pi, \frac{7}{4}\pi]$, and hence also on I_ε, we have

$$\left|\frac{f''(s)}{2f'(t)}\right| \leq \frac{\sinh\left(\frac{7}{4}\pi\right)}{2\cosh\left(\frac{3}{2}\pi\right)} = 1.0965\ldots,$$

which, multiplied by $\frac{1}{4}\pi$ – the length of the interval – is $0.86121\ldots < 1$. All the more, therefore, $\varepsilon M(\varepsilon) < 1$. By Theorem 4.5.1 and its proof, it follows that all subsequent approximations $x_3, x_4, \ldots$ remain in this interval and converge to α_1, the order of convergence being $p = \frac{1}{2}(1+\sqrt{5})$ by Theorem 4.5.2. To obtain the root α_1 to within the three tolerances used earlier, we find that 6, 7, and 9 iterations, respectively, suffice. This should be contrasted with 25, 52, and 112 iterations (cf. Sect. 4.3.1) for the bisection method.

In contrast to bisection, the secant method is *not* robust at the level of machine precision and may even fail if `fa` happens to become equal to `fb`. The method is, therefore, rarely used on its own, but more often in combination with the method of bisection,; see, for example, Dekker [1969] and Brent [2002, Chap. 4].

4.6 Newton's Method

Newton's method can be thought of as a limit case of the secant method (4.45) if we let there x_{n-1} move into x_n. The result is

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, 2, \ldots, \tag{4.55}$$

where x_0 is some appropriate initial approximation. Another more fruitful interpretation is that of *linearization of the equation* $f(x) = 0$ *at* $x = x_n$. In other words, we replace $f(x)$ for x near x_n by the linear approximation $f(x) \approx f_a(x) := f(x_n) + (x - x_n)f'(x_n)$ obtained by truncating the Taylor expansion of f centered at x_n after the linear term and then solve the resulting linear equation, $f_a(x) = 0$, calling the solution x_{n+1}. This again leads to (4.55). Viewed in this manner, Newton's method can be vastly generalized to nonlinear equations of all kinds, not only single equations as in (4.55), but also systems of nonlinear equations (cf. Sect. 4.9.2) and even functional equations, in which case the derivative f' is to be understood as a Fréchet derivative.

It is useful to begin with a few simple examples of single equations to get a feel for how Newton's method may behave.

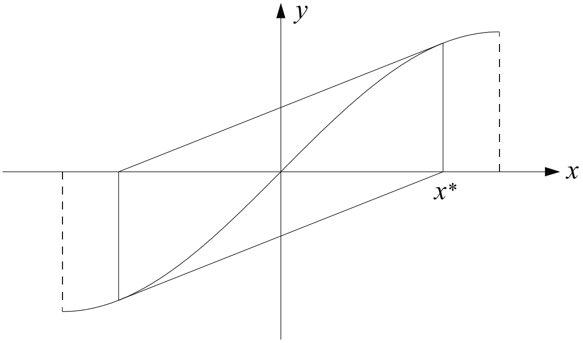


Fig. 4.3 A cycle in Newton's method

Example. The square root $\alpha = \sqrt{a}$, $a > 0$. The equation here is $f(x) = 0$ with

$$f(x) = x^2 - a. \tag{4.56}$$

Equation (4.55) then immediately gives

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right), \quad n = 0, 1, 2, \ldots \tag{4.57}$$

(a method already used by the Babylonians long before Newton). Because of the convexity of f it is clear that iteration (4.57) converges to the positive square root for each $x_0 > 0$ and is monotonically decreasing (except for the first step in the case $0 < x_0 < \alpha$). We have here an elementary example of *global convergence*.

Example. $f(x) = \sin x$, $|x| < \frac{1}{2}\pi$. There is exactly one root in this interval, the trivial root $\alpha = 0$. Newton's method becomes

$$x_{n+1} = x_n - \tan x_n, \quad n = 0, 1, 2, \ldots . \tag{4.58}$$

It exhibits the following amusing phenomenon (cf. Fig. 4.3). If $x_0 = x^*$, where

$$\tan x^* = 2x^*, \tag{4.59}$$

then $x_1 = -x^*$, $x_2 = x^*$, that is, after two steps of Newton's method we end up where we started. This is called a *cycle*.

For this starting value, Newton's method does not converge, let alone to $\alpha = 0$. It does converge, however, for any starting value x_0 with $|x_0| < x^*$, generating a sequence of alternately increasing and decreasing approximations x_n converging necessarily to $\alpha = 0$. The value of the critical number x^* can itself be computed by Newton's method applied to (4.59). The result is $x^* = 1.16556\ldots$. In a sense, we have here an example of *local convergence*, since convergence cannot hold for all $x_0 \in [-\frac{1}{2}\pi, \frac{1}{2}\pi]$. (If $x_0 = \frac{1}{2}\pi$, we even get thrown off to ∞.)

Example. $f(x) = x^{20} - 1, x > 0$. This has exactly one positive (simple) root $\alpha = 1$. Newton's method yields the iteration

$$x_{n+1} = \frac{19}{20} x_n + \frac{1}{20 x_n^{19}}, \quad n = 0, 1, 2, \dots, \tag{4.60}$$

which provides a good example to illustrate one of the dangers in Newton's method: unless one starts sufficiently close to the desired root, it may take a long while to approach it. Thus, suppose we take $x_0 = \frac{1}{2}$, then $x_1 \approx \frac{2^{19}}{20} = 2.62144 \times 10^4$, a huge number. What is worse, it is going to be a slow ride back to the vicinity of $\alpha = 1$, since for x_n very large, one has

$$x_{n+1} \approx \frac{19}{20} x_n, \quad x_n \gg 1.$$

At each step the approximation is reduced only by a fraction $\frac{19}{20} = 0.95$. It takes about 200 steps to get back to near the desired root. But once we come close to $\alpha = 1$, the iteration speeds up dramatically and converges to the root quadratically (see Theorem 4.6.1). Since f is again convex, we actually have global convergence on $\mathbb{R}_+$, but, as we have seen, this is of little comfort.

Example. Let $f \in C^2[a, b]$ be such that

$$\begin{cases} f \text{ is convex (or concave) on } [a, b]; \\ f(a) f(b) < 0; \\ \text{the tangents at the endpoints of } [a, b] \\ \quad \text{intersect the real line within } [a, b]. \end{cases} \tag{4.61}$$

In this case, it is clear on geometric grounds that Newton's method converges globally, that is, for any $x_0 \in [a, b]$. Note that the tangent condition in (4.61) is automatically satisfied at *one* of the endpoints.

The following is a (symbolic/variable-precision) Matlab routine implementing Newton's method and returning not only an approximation x to the root, but also the number n of iterations required to obtain it. The initial approximation is input by `a`, and `tol` is the error tolerance. For reasons of safety, we limit the number of iterations to `nmax`. Two function routines `f,fd` evaluating f and f' have to be appended as subfunctions.

```
%SNEWTON  Symbolic Newton's method
%
function [n,x]=snewton(dig,a,tol,nmax)
n=0;
digits(dig);
a=vpa(a);
x0=a+1; x=a;
```

```
while subs(abs(x-x0))>=tol
  n=n+1;
  if n>nmax
    fprintf('n exceeds nmax')
    return
  end
  x0=x;
  x=x0-f(x0)/fd(x0);
end

function y=f(x)
y=cos(x)*cosh(x)-1;

function y=fd(x)
y=-sin(x)*cosh(x)+cos(x)*sinh(x);
```

Our test equation (4.2) provides a good illustration for (4.61). The function $f(x) = \cos x \cosh x - 1$ is convex on $[\frac{3}{2}\pi, 2\pi]$ and the tangents at both endpoints intersect the real line inside the interval $[\frac{3}{2}\pi, 2\pi]$. This is obvious, by convexity, for the right endpoint, and for the left endpoint the tangent intersects the real line at $x_\ell = \frac{3}{2}\pi + \frac{1}{\cosh(3\pi/2)} = 4.7303\ldots < 2\pi$. Moreover, since the point of intersection of the tangent at the right endpoint $x_r = \frac{2\pi \sinh 2\pi - \cosh 2\pi + 1}{\sinh 2\pi} = 5.2869\ldots$ is to the right of x_ℓ Newton's method converges faster if started at the left endpoint and yields α_1 in 4, 5, and 6 iterations for the three tolerances 0.5×10^{-7}, 0.5×10^{-15}, and 0.5×10^{-33}, respectively. This is slightly, but not much, faster than the secant method. We see shortly, however, that the efficiency index is smaller for Newton's method than for the secant method.

To study the error in Newton's iteration, subtract α – a presumed simple root of the equation – from both sides of (4.55) to get

$$\begin{aligned} x_{n+1} - \alpha &= x_n - \alpha - \frac{f(x_n)}{f'(x_n)} \\ &= (x_n - \alpha)\left(1 - \frac{f(x_n) - f(\alpha)}{(x_n - \alpha) f'(x_n)}\right) \\ &= (x_n - \alpha)\left(1 - \frac{[x_n, \alpha]f}{[x_n, x_n]f}\right) \\ &= (x_n - \alpha)^2 \, \frac{[x_n, x_n, \alpha]f}{[x_n, x_n]f}. \end{aligned} \tag{4.62}$$

Therefore, if $x_n \to \alpha$, then

$$\lim_{n\to\infty} \frac{x_{n+1} - \alpha}{(x_n - \alpha)^2} = \frac{f''(\alpha)}{2f'(\alpha)}, \tag{4.63}$$

that is, Newton's method converges quadratically if $f''(\alpha) \neq 0$. Since it requires at each step one function evaluation and one derivative evaluation, the efficiency index is $\sqrt{2} = 1.41421\ldots$, which is less than the one for the secant method. Formula (4.63) suggests writing the equation $f(x) = 0$ in alternative, but equivalent, ways so as to reduce the asymptotic error constant on the right of (4.63).

The proof of *local convergence* of Newton's method is virtually the same as the one for the secant method (cf. Theorem 4.5.1). We only state the result and refer to Ex. 28 for a proof

Theorem 4.6.1. *Let α be a simple root of the equation $f(x) = 0$ and let $I_\varepsilon = \{x \in \mathbb{R} : |x - \alpha| \leq \varepsilon\}$. Assume that $f \in C^2[I_\varepsilon]$. Define*

$$M(\varepsilon) = \max_{\substack{s \in I_\varepsilon \\ t \in I_\varepsilon}} \left| \frac{f''(s)}{2f'(t)} \right|. \tag{4.64}$$

If ε is so small that

$$2\varepsilon M(\varepsilon) < 1, \tag{4.65}$$

then for every $x_0 \in I_\varepsilon$, Newton's method is well defined and converges quadratically to the only root $\alpha \in I_\varepsilon$. (The extra factor 2 in (4.65) comes from the requirement that $f'(x) \neq 0$ for $x \in I_\varepsilon$.)

An interesting variant of Newton's method is *Steffensen's method*

$$x_{n+1} = x_n - \frac{f(x_n)}{g(x_n)}, \quad g(x_n) = \frac{f(x_n + f(x_n)) - f(x_n)}{f(x_n)}, \tag{4.66}$$

which shares with Newton's method the property of second-order convergence but does not require the derivative of f. Instead, it replaces $f'(x_n)$ in (4.55) by the difference quotient $g(x_n) = [f(x_n + h_n) - f(x_n)]/h_n$, where $h_n = f(x_n)$.

4.7 Fixed Point Iteration

Often, in applications, a nonlinear equation presents itself in the form of a *fixed point problem*: find x such that

$$x = \varphi(x). \tag{4.67}$$

A number α satisfying this equation is called a *fixed point of φ*. Any equation $f(x) = 0$, in fact, can (in many different ways) be written equivalently in the form (4.67). For example, if $f'(x) \neq 0$ in the interval of interest, we can take

$$\varphi(x) = x - \frac{f(x)}{f'(x)}. \tag{4.68}$$

If x_0 is an initial approximation of a fixed point α of (4.67), the *fixed point iteration* generates a sequence of approximants by

$$x_{n+1} = \varphi(x_n), \quad n = 0, 1, 2, \dots . \tag{4.69}$$

If it converges, it clearly converges to a fixed point of φ if φ is continuous. Note that (4.69) is precisely Newton's method for solving $f(x) = 0$ if φ is defined by (4.68). So Newton's method can be viewed as a fixed point iteration, but not the secant method (why not?).

For any iteration of the form (4.69), assuming that $x_n \to \alpha$ as $n \to \infty$, it is straightforward to determine the order of convergence. Suppose, indeed, that at the fixed point α we have

$$\varphi'(\alpha) = \varphi''(\alpha) = \cdots = \varphi^{(p-1)}(\alpha) = 0, \quad \varphi^{(p)}(\alpha) \neq 0. \tag{4.70}$$

(We tacitly assume $\varphi \in C^p$ near α.) This defines the integer $p \geq 1$. We then have by Taylor's theorem

$$\begin{aligned}\varphi(x_n) = \varphi(\alpha) + (x_n - \alpha)\varphi'(\alpha) + \cdots + \frac{(x_n-\alpha)^{p-1}}{(p-1)!}\,\varphi^{(p-1)}(\alpha)\\ + \frac{(x_n-\alpha)^p}{p!}\,\varphi^{(p)}(\xi_n) = \varphi(\alpha) + \frac{(x_n-\alpha)^p}{p!}\,\varphi^{(p)}(\xi_n),\end{aligned}$$

where ξ_n is between α and x_n. Since $\varphi(x_n) = x_{n+1}$ and $\varphi(\alpha) = \alpha$, we get

$$\frac{x_{n+1}-\alpha}{(x_n-\alpha)^p} = \frac{1}{p!}\,\varphi^{(p)}(\xi_n).$$

As $x_n \to \alpha$, since ξ_n is trapped between x_n and α, we conclude, by the continuity of $\varphi^{(p)}$ at α, that

$$\lim_{n\to\infty} \frac{x_{n+1}-\alpha}{(x_n-\alpha)^p} = \frac{1}{p!}\,\varphi^{(p)}(\alpha) \neq 0. \tag{4.71}$$

This shows that convergence is exactly of the order p, and

$$c = \frac{1}{p!}\,\varphi^{(p)}(\alpha) \tag{4.72}$$

is the asymptotic error constant. Combining this with the usual local convergence argument, we obtain the following result.

Theorem 4.7.1. *Let α be a fixed point of φ and $I_\varepsilon = \{x \in \mathbb{R}\colon |x-\alpha| \leq \varepsilon\}$. Assume $\varphi \in C^p[I_\varepsilon]$ satisfies (4.70). If*

$$M(\varepsilon) := \max_{t\in I_\varepsilon} |\varphi'(t)| < 1, \tag{4.73}$$

then the fixed point iteration (4.69) converges to α *for any* $x_0 \in I_\varepsilon$. *The order of convergence is p and the asymptotic error constant given by (4.72).*

Applying the theorem to (4.68) recovers second-order convergence of Newton's method. Indeed, with φ given in (4.68), we have

$$\varphi'(x) = 1 - \frac{[f'(x)]^2 - f(x)f''(x)}{[f'(x)]^2} = f(x)\,\frac{f''(x)}{[f'(x)]^2};$$

hence $\varphi'(\alpha) = 0$ (if $f'(\alpha) \neq 0$), and

$$\varphi''(x) = f(x)\left(\frac{f''(x)}{[f'(x)]^2}\right)' + \frac{f''(x)}{f'(x)}\;;$$

hence $\varphi''(\alpha) = \frac{f''(\alpha)}{f'(\alpha)} \neq 0$, unless $f''(\alpha) = 0$. In the exceptional case $f''(\alpha) = 0$, Newton's method converges cubically (at least).

4.8 Algebraic Equations

There are many iterative methods specifically designed to solve algebraic equations. Here we only describe how Newton's method applies in this context, essentially confining ourselves to a discussion of an efficient way to evaluate simultaneously the value of a polynomial and its first derivative. In the special case where all zeros of the polynomial are known to be real and simple, we describe an improved variant of Newton's method.

4.8.1 Newton's Method Applied to an Algebraic Equation

We consider an algebraic equation of degree d,

$$f(x) = 0, \quad f(x) = x^d + a_{d-1}x^{d-1} + \cdots + a_0, \tag{4.74}$$

where the leading coefficient is assumed (without restricting generality) to be 1 and where we may also assume $a_0 \neq 0$ without loss of generality. For simplicity we assume all coefficients to be real.

To apply Newton's method to (4.74), one needs good methods for evaluating a polynomial and its derivative. Underlying such methods are *division algorithms* for polynomials. Let t be some parameter and suppose we want to divide $f(x)$ by $x - t$.

We write

$$f(x) = (x - t)(x^{d-1} + b_{d-1}x^{d-2} + \cdots + b_1) + b_0 \tag{4.75}$$

and compare coefficients of powers of x on both sides. This leads immediately to the equations

$$\begin{aligned} b_d &= 1, \\ b_k &= tb_{k+1} + a_k, \quad k = d-1, d-2, \ldots, 0. \end{aligned} \tag{4.76}$$

The coefficients b_k so determined of course depend on t; indeed, b_k is a polynomial in t of degree $d - k$. We now make three useful observations:

(a) From (4.75), we note that $b_0 = f(t)$. Thus, in (4.76), we have an algorithm to evaluate $f(t)$ for any given t. It is known as *Horner's scheme*, although Newton already knew it. It requires d multiplications and d additions, which is more efficient than the naive way of computing f, which would be to first form the successive powers of t and then multiply them into their coefficients. This would require twice as many multiplies as Horner's scheme, but the same number of additions. It is an interesting question of complexity whether the number of multiplications can be further reduced. (It is known by a theorem of Ostrowski that the number d of additions is optimal.) This indeed is possible, and schemes using less than d multiplications have been developed by Pan and others. Unfortunately, the reduction in complexity comes with a price of increased numerical instability. Horner's scheme, therefore, is still the most widely used technique for evaluating a polynomial.

(b) Suppose $t = \alpha$, where α is a zero of f. Then $b_0 = 0$, and (4.75) allows division by $x - \alpha$ without remainder:

$$x^{d-1} + b_{d-1}x^{d-2} + \cdots + b_1 = \frac{f(x)}{x - \alpha}. \tag{4.77}$$

This is the *deflated polynomial*, in which the zero α has been "removed" from f. To compute its coefficients, therefore, all we need to do is apply Horner's scheme with $t = \alpha$. This comes in very handy in Newton's method when f is evaluated by Horner's scheme: once the method has converged to a root α, the final evaluation of f at (or very near) α automatically provides us with the coefficients of the deflated polynomial, and we are ready to reapply Newton's method to this deflated polynomial to compute the remaining zeros of f.

(c) By differentiating (4.75) with respect to x and then putting $x = t$, we obtain

$$f'(t) = t^{d-1} + b_{d-1}t^{d-2} + \cdots + b_1. \tag{4.78}$$

Thus, we can apply Horner's scheme again to this polynomial to evaluate $f'(t)$. Both applications of Horner's scheme are conveniently combined into the following *double Horner scheme*:

$$\begin{aligned} & b_d = 1, \;\; c_d = 1 \\ & \left.\begin{aligned} b_k &= t b_{k+1} + a_k \\ c_k &= t c_{k+1} + b_k \end{aligned}\right\} \quad k = d-1, d-2, \ldots, 1, 0. \end{aligned} \tag{4.79}$$

Then

$$f(t) = b_0, \quad f'(t) = c_1. \tag{4.80}$$

The last step in (4.79) for c_0 is actually redundant, but it does not seem worth the complication in programming to eliminate this extra step.

We now have a convenient way to compute $f(x_n)$ and $f'(x_n)$ in each Newton step $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$: apply the algorithm (4.79) with $t = x_n$ and use (4.80). Once x_n has converged to α, the b generated in (4.79) give us the coefficients of the deflated polynomial (4.77) and we are ready to reapply Newton's method to the deflated polynomial.

It is clear that any real initial approximation x_0 generates a sequence of *real* iterates x_n and, therefore, can only be applied to compute real zeros of f (if any). For complex zeros one must start with a complex x_0, and the whole computation proceeds in complex arithmetic. It is possible, however, to use division algorithms with quadratic divisors to compute quadratic factors of f entirely in real arithmetic (Bairstow's method). See Ex. 44 for details.

One word of caution is in order when one tries to compute all zeros of f successively by Newton's method. It is true that Newton's method combined with Horner's scheme has a built-in mechanism of deflation, but this mechanism is valid only if one assumes convergence to the exact roots. This of course is impossible, partly because of rounding errors and partly because of the stopping criterion used to terminate the iteration prematurely. Thus, there is a build-up of errors in the successively deflated polynomials, which may well have a significant effect on the accuracy of the respective roots (cf. Chap. 1, Sect. 1.3.2(2)). It is therefore imperative, once all roots have been computed, to "purify" them by applying Newton's method one more time to the *original* polynomial f in (4.74), using the computed roots as initial approximations.

4.8.2 An Accelerated Newton Method for Equations with Real Roots

If (4.74) has only real distinct roots,

$$f(\alpha_\nu) = 0, \quad \alpha_d < \alpha_{d-1} < \cdots < \alpha_2 < \alpha_1, \tag{4.81}$$

one can try to speed up Newton's method by approaching each root from the right with double the Newton steps until it is overshot, at which time one switches back to the ordinary Newton iteration to finish off the root.

Underlying this method is the following interesting theorem.

Theorem 4.8.1. *Let f be a polynomial having only real zeros as in (4.81), and let α_1' be the largest zero of f'. Then for every $z > \alpha_1$, defining*

$$z' := z - \frac{f(z)}{f'(z)}, \quad y := z - 2\,\frac{f(z)}{f'(z)}, \quad y' := y - \frac{f(y)}{f'(y)}, \tag{4.82}$$

one has

$$\alpha_1' < y, \quad \alpha_1 \le y' \le z'. \tag{4.83}$$

The theorem suggests the following algorithm: start with some $x_0 > \alpha_1$ and apply

$$x_{k+1} = x_k - 2\,\frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \ldots. \tag{4.84}$$

Then there are the following possibilities.

(i) We have $x_0 > x_1 > x_2 > \cdots > \alpha_1$ and $x_k \downarrow \alpha_1$ as $k \to \infty$. Since we use double Newton steps in (4.84), convergence in this case is faster than for the ordinary Newton iteration. Note also that $f(x_k) > 0$ for all k.

(ii) There exists a first index $k = k_0$ such that

$$f(x_0)f(x_k) > 0 \ \text{ for } \ 0 \le k < k_0, \quad f(x_0)f(x_{k_0}) < 0.$$

Then $y := x_{k_0}$ is to the left of α_1 (we overshot) but, by (4.83), to the right of α_1'. Using now y as the starting value in the ordinary Newton iteration,

$$y_0 = y, \ \ y_{k+1} = y_k - \frac{f(y_k)}{f'(y_k)}, \quad k = 0, 1, 2 \ldots,$$

brings us back to the right of α_1 in the first step and then monotonically down to α_1.

In either case, having obtained α_1, we apply the same procedure to the deflated polynomial $f_1(x) = \frac{f(x)}{x-\alpha_1}$ to compute the next smaller zero. As starting value we can take α_1, or better, if case (ii) has occurred, $y = x_{k_0}$. The procedure can obviously be continued until all roots have been computed.

4.9 Systems of Nonlinear Equations

Although most of the methods discussed earlier allow extensions to systems of nonlinear equations,

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}, \quad \boldsymbol{f} : \mathbb{R}^d \to \mathbb{R}^d, \tag{4.85}$$

we consider here only two such methods: the fixed point iteration and Newton's method.

4.9.1 Contraction Mapping Principle

We write (4.85) in fixed point form, $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{\varphi}(\boldsymbol{x})$, and consider the fixed point iteration

$$\boldsymbol{x}_{n+1} = \boldsymbol{\varphi}(\boldsymbol{x}_n), \quad n = 0, 1, 2, \dots . \tag{4.86}$$

We say that $\boldsymbol{\varphi} : \mathbb{R}^d \to \mathbb{R}^d$ is a *contraction map* (or is *contractive*) on a set $\mathcal{D} \subseteq \mathbb{R}^d$ if there exists a constant γ with $0 < \gamma < 1$ such that, in some appropriate vector norm,

$$\|\boldsymbol{\varphi}(\boldsymbol{x}) - \boldsymbol{\varphi}(\boldsymbol{x}^*)\| \le \gamma \|\boldsymbol{x} - \boldsymbol{x}^*\| \quad \text{for all } \boldsymbol{x} \in \mathcal{D},\ \boldsymbol{x}^* \in \mathcal{D}. \tag{4.87}$$

Theorem 4.9.1. (Contraction Mapping Principle). *Let $\mathcal{D} \subseteq \mathbb{R}^d$ be a complete subset of $\mathbb{R}^d$ (i.e., either bounded and closed, or all of $\mathbb{R}^d$). If $\boldsymbol{\varphi} : \mathbb{R}^d \to \mathbb{R}^d$ is contractive in the sense of (4.87) and maps $\mathcal{D}$ into $\mathcal{D}$, then*

(i) *iteration (4.86) is well defined for any $\boldsymbol{x}_0 \in \mathcal{D}$ and converges to a unique fixed point $\boldsymbol{\alpha} \in \mathcal{D}$,*

$$\lim_{n\to\infty} \boldsymbol{x}_n = \boldsymbol{\alpha}; \tag{4.88}$$

(ii) *for $n = 1, 2, 3, \dots$ there holds*

$$\|\boldsymbol{x}_n - \boldsymbol{\alpha}\| \le \frac{\gamma^n}{1-\gamma} \|\boldsymbol{x}_1 - \boldsymbol{x}_0\| \tag{4.89}$$

and

$$\|\boldsymbol{x}_n - \boldsymbol{\alpha}\| \le \gamma^n \|\boldsymbol{x}_0 - \boldsymbol{\alpha}\|. \tag{4.90}$$

Proof. (i) Since $\boldsymbol{\varphi}(\mathcal{D}) \subseteq \mathcal{D}$, iteration (4.86) is well defined. We have, for $n = 1, 2, 3, \dots$,

$$\|\boldsymbol{x}_{n+1} - \boldsymbol{x}_n\| = \|\boldsymbol{\varphi}(\boldsymbol{x}_n) - \boldsymbol{\varphi}(\boldsymbol{x}_{n-1})\| \le \gamma \|\boldsymbol{x}_n - \boldsymbol{x}_{n-1}\|.$$

Repeated application of this yields

$$\|\boldsymbol{x}_{n+1}-\boldsymbol{x}_n\| \le \gamma^n \|\boldsymbol{x}_1-\boldsymbol{x}_0\|,$$

and hence, since

$$\boldsymbol{x}_{n+p}-\boldsymbol{x}_n = (\boldsymbol{x}_{n+p}-\boldsymbol{x}_{n+p-1}) + (\boldsymbol{x}_{n+p-1}-\boldsymbol{x}_{n+p-2}) + \cdots + (\boldsymbol{x}_{n+1}-\boldsymbol{x}_n),$$

more generally,

$$\begin{aligned}\|\boldsymbol{x}_{n+p}-\boldsymbol{x}_n\| &\le \sum_{k=1}^{p} \|\boldsymbol{x}_{n+k}-\boldsymbol{x}_{n+k-1}\| \le \sum_{k=1}^{p} \gamma^{n+k-1}\|\boldsymbol{x}_1-\boldsymbol{x}_0\| \\ &\le \gamma^n \sum_{k=1}^{\infty} \gamma^{k-1}\|\boldsymbol{x}_1-\boldsymbol{x}_0\| = \frac{\gamma^n}{1-\gamma}\,\|\boldsymbol{x}_1-\boldsymbol{x}_0\|. \qquad (4.91)\end{aligned}$$

Since $\gamma^n \to 0$, it follows that $\{\boldsymbol{x}_n\}$ is a Cauchy sequence in $\mathcal{D}$, and hence, since $\mathcal{D}$ is complete, converges to some $\boldsymbol{\alpha} \in \mathcal{D}$,

$$\lim_{n\to\infty} \boldsymbol{x}_n \to \boldsymbol{\alpha}.$$

The limit $\boldsymbol{\alpha}$ must be a fixed point of $\boldsymbol{\varphi}$ since

$$\|\boldsymbol{x}_n-\boldsymbol{\varphi}(\boldsymbol{\alpha})\| = \|\boldsymbol{\varphi}(\boldsymbol{x}_{n-1})-\boldsymbol{\varphi}(\boldsymbol{\alpha})\| \le \gamma\|\boldsymbol{x}_{n-1}-\boldsymbol{\alpha}\|, \qquad (4.92)$$

hence $\boldsymbol{\alpha} = \lim_{n\to\infty}\boldsymbol{x}_n = \boldsymbol{\varphi}(\boldsymbol{\alpha})$. Moreover, there can be only one fixed point in $\mathcal{D}$, since $\boldsymbol{\alpha} = \boldsymbol{\varphi}(\boldsymbol{\alpha})$, $\boldsymbol{\alpha}^* = \boldsymbol{\varphi}(\boldsymbol{\alpha}^*)$, and $\boldsymbol{\alpha} \in \mathcal{D}$, $\boldsymbol{\alpha}^* \in \mathcal{D}$ imply $\|\boldsymbol{\alpha}-\boldsymbol{\alpha}^*\| = \|\boldsymbol{\varphi}(\boldsymbol{\alpha})-\boldsymbol{\varphi}(\boldsymbol{\alpha}^*)\| \le \gamma\|\boldsymbol{\alpha}-\boldsymbol{\alpha}^*\|$, that is, $(1-\gamma)\|\boldsymbol{\alpha}-\boldsymbol{\alpha}^*\| \le 0$, and hence $\boldsymbol{\alpha} = \boldsymbol{\alpha}^*$, since $1-\gamma > 0$.

(ii) Letting $p \to \infty$ in (4.91) yields the first inequality in Theorem 4.9.1(ii). The second follows by a repeated application of (4.92), since $\boldsymbol{\varphi}(\boldsymbol{\alpha}) = \boldsymbol{\alpha}$. □

Inequality (4.90) shows that the fixed point iteration converges (at least) linearly, with an error bound having asymptotic error constant equal to γ.

4.9.2 Newton's Method for Systems of Equations

As we mentioned earlier, Newton's method can be easily adapted to deal with systems of nonlinear equations, reducing the nonlinear problem to an infinite sequence of linear problems, that is, systems of linear algebraic equations. The tool is linearization at the current approximation.

Thus, given the equation (4.85), written more explicitly as

$$f^i(x^1, x^2, \ldots, x^d) = 0, \quad i = 1, 2, \ldots, d \tag{4.93}$$

(where components are indexed by superscripts), and given an approximation $\boldsymbol{x}_0$ to a solution $\boldsymbol{\alpha} \in \mathbb{R}^d$, the ith equation in (4.93) is linearized at $\boldsymbol{x} = \boldsymbol{x}_0$ by truncating the Taylor expansion of f^i at $\boldsymbol{x}_0$ after the linear terms. This gives

$$f^i(\boldsymbol{x}_0) + \sum_{j=1}^{d} \frac{\partial f^i}{\partial x^j}(\boldsymbol{x}_0)(x^j - x_0^j) = 0, \quad i = 1, 2, \ldots, d,$$

or, written in vector form,

$$\boldsymbol{f}(\boldsymbol{x}_0) + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(\boldsymbol{x}_0)(\boldsymbol{x} - \boldsymbol{x}_0) = \boldsymbol{0}, \tag{4.94}$$

where

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(\boldsymbol{x}) := \left[\frac{\partial f^i}{\partial x^j}(\boldsymbol{x})\right]_{i,j=1}^{d} \tag{4.95}$$

is the *Jacobian matrix* of $\boldsymbol{f}$. This is the natural generalization of the first derivative of a single function to systems of functions. The solution $\boldsymbol{x}$ of (4.94) – a system of linear algebraic equations – will be taken to be the next approximation. Thus, in general, starting with an initial approximation $\boldsymbol{x}_0$, Newton's method will generate a sequence of approximations $\boldsymbol{x}_n \in \mathbb{R}^d$ by means of

$$\begin{aligned} \tfrac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(\boldsymbol{x}_n)\boldsymbol{\Delta}_n &= -\boldsymbol{f}(\boldsymbol{x}_n), \\ \boldsymbol{x}_{n+1} &= \boldsymbol{x}_n + \boldsymbol{\Delta}_n, \end{aligned} \qquad n = 0, 1, 2, \ldots, \tag{4.96}$$

where we assume that the matrix $(\partial \boldsymbol{f}/\partial \boldsymbol{x})(\boldsymbol{x}_n)$ in the first equation is nonsingular for each n. This will be the case if $(\partial \boldsymbol{f}/\partial \boldsymbol{x})(\boldsymbol{\alpha})$ is nonsingular and $\boldsymbol{x}_0$ is sufficiently close to $\boldsymbol{\alpha}$, in which case one can prove as in the one-dimensional case $d = 1$ that Newton's method converges quadratically to $\boldsymbol{\alpha}$, that is, $\|\boldsymbol{x}_{n+1} - \boldsymbol{\alpha}\| = O(\|\boldsymbol{x}_n - \boldsymbol{\alpha}\|^2)$ as $n \to \infty$.

Writing (4.96) in the form

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \left[\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(\boldsymbol{x}_n)\right]^{-1} \boldsymbol{f}(\boldsymbol{x}_n), \quad n = 0, 1, 2, \ldots, \tag{4.97}$$

brings out the formal analogy with Newton's method (4.55) for a single equation. However, it is not necessary to compute the inverse of the Jacobian at each step; it is more efficient to solve the linear system directly as in (4.96).

There are many ways to modify the initial stages of Newton's method (for systems of nonlinear equations) to force the iteration to make good progress toward approaching a solution. These usually go under the name *quasi-Newton methods* and they all share the idea of employing a suitable approximate inverse of the Jacobian rather than the exact one. For these, and also for generalizations of the secant method and the method of false position to systems of nonlinear equations, we refer to specialized texts (cf. the Notes to Chap. 4).

4.10 Notes to Chapter 4

Texts dealing largely with single nonlinear equations are Traub [1964], Householder [1970], and Brent [2002]. Traub's book provides a systematic treatment of iterative methods, both old and new. Although it deals also with questions of convergence, the emphasis is on the derivation, classification, and cataloguing of iterative methods. Householder's book is strong on algebraic and analytic tools (often rooted in nineteenth-century mathematics) underlying numerical methods and less so on the methods themselves. Brent's book, although mainly devoted to optimization, gives a detailed account, and a computer program, of an algorithm essentially due to Dekker, combining the secant method with bisection. Other well-tested algorithms available in software are, for example, the IMSL routine `zreal`, implementing Muller's method (Muller [1956]), and `zporc` for (real) algebraic equations, implementing the Jenkins–Traub three-stage algorithm (Jenkins and Traub [1970]). The routine `zplrc`, based on Laguerre's method (cf., e.g., Fröberg [1985, Sect. 11.5]), also finds complex roots. The computation of zeros of analytic functions is treated in Kravanja and Van Barel [2000].

For the solution of systems of nonlinear equations, the books by Ortega and Rheinboldt [2000] and Rheinboldt [1998] give well-organized, and perhaps the most comprehensive, descriptions and mathematical analyses of nonlinear iterative methods. A less exhaustive, but mathematically penetrating, text, also dealing with operator equations in Banach space, is Ostrowski [1973]. At a lower level of mathematical sophistication, but richer in algorithmic details, is the book by Dennis and Schnabel [1996], which gives equal treatment to nonlinear equations and optimization. The books by Kelley [1995, 1999] dealing with iterative methods for linear and nonlinear systems of equations and for optimization, as well as Kelley [2003] specifically for Newton-type methods, provide e-mail and Web site addresses for respective Matlab codes. Complexity issues are discussed in Sikorski [2001] and iterative regularization methods for nonlinear inverse problems in Kaltenbacher et al. [2008]. A useful guide on available software for problems in optimization (including nonlinear equations) is Moré and Wright [1993]. For older precomputer literature, the two-volume treatise by Durand [1960, 1961] is still a valuable source.

Section 4.1. A sampling of typical systems of nonlinear equations occurring in applied analysis is given in Ortega and Rheinboldt [2000, Chap. 1]. Many nonlinear equation problems arise from minimization problems, where one tries to find a minimum of a function of several variables. If it occurs at an interior point, then indeed the gradient of the function must vanish.

Section 4.1.2. For more on shooting methods, see Chap. 7, Sect. 7.2.

Section 4.1.3. Another, even simpler, way of approximating the solution y of (4.9) by y_n is to apply an n-point quadrature rule to the integral on the left, thus writing $y_n(x) = a(x) + \sum_{k=1}^{n} w_k K(x, t_k) f(t_k, y_n(t_k))$, $0 \leq x \leq 1$, and determining $\{y_n(t_k)\}_{k=1}^{n}$ by putting $x = t_i$, $i = 1, 2, \ldots, n$, in this equation. Again, we are led to a system of nonlinear equations.

Section 4.1.4. This example is taken from Gautschi and Milovanović [1997], where one also finds a discussion on how to compute the Turán-type quadrature rule associated with an s-orthogonal polynomial π_n. This is the quadrature rule of maximum degree of exactness that involves a function and its derivatives up to order $2s$ evaluated at the zeros of π_n.

Section 4.2. More refined measures for the speed of convergence are defined in Brent [2002, p. 21] and, especially, in Ortega and Rheinboldt [2000, Chap. 9], where the concepts of Q-order and R-order of convergence are introduced, the former relating to quotients of errors, as in (4.23), the latter to nth roots $\varepsilon_n^{1/n}$. Also see Potra [1989]. These are quantities that characterize the asymptotic behavior of the error as $n \to \infty$; they say nothing about the initial behavior of the error. If one wants to describe the overall behavior of the iteration, one needs a function, not a number, to define the rate of convergence, for example, a function (if one exists) that relates $\|x_{n+1} - x_n\|$ to $\|x_n - x_{n-1}\|$ (cf. Ex. 25(b)). This is the approach taken in Potra and Pták [1984] to describe the convergence behavior of iterative processes in complete metric spaces. For similar ideas in connection with the convergence behavior of continued fractions, also see Gautschi [1983].

The efficiency index was introduced by Ostrowski [1973, Chap. 3, Sect. 11], who also coined the word "horner" for a unit of work.

Aitken's Δ^2-process is a special case of a more general nonlinear acceleration method, the Shanks transformation, or, in Wynn's recursive implementation, the epsilon algorithm, see Brezinski and Redivo-Zaglia (1991, pp. 78–95).

Section 4.3.2. For Sturm sequences and Sturm's theorem see, for example, Henrici [1988, p. 444ff], and for Gershgorin's theorem, Golub and Van Loan [1996, p. 320]. The bisection method based on Sturm sequences, in the context of eigenvalues of a symmetric tridiagonal matrix, is implemented in the Eispack routine `BISECT` (Smith et al. [1976, p. 211]).

Section 4.4. The method of false position is very old, originating in medieval Arabic mathematics, and even earlier in fifth-century Indian texts (Plofker [1996, p. 254]). Leonardo Pisano (better known as "Fibonacci"), in the thirteenth century, calls it “*regula duarum falsarum positionum*,” which, in the sixteenth and seventeenth centuries, became abbreviated to “*regula positionum*” or also “*regula falsi*.” Peter Bienewitz (1527), obviously having a linear equation in mind, explains the method in these (old German) words: “*Vnd heisst nit darum falsi dass sie falsch vnd unrecht wehr, sunder, dass sie auss zweyen falschen vnd vnwahrhaftigen zalen, vnd zweyen lügen die wahrhaftige vnd begehrte zal finden lernt*.” (Cf. Maas [1985]).

In the form (4.42), the regula falsi can be thought of as a discretized Newton's method, if in the latter, (4.55), one replaces the derivative $f'(x_n)$ by the difference quotient $(f(x_n) - f(b))/(x_n - b)$. This suggests one (of many) possible extension to systems of equations in $\mathbb{R}^d$ (cf. Ortega and Rheinboldt [2000, p. 205]): define vector difference quotients $\boldsymbol{\Delta}_{n,i} = (x_n^i - b^i)^{-1}[\boldsymbol{f}(\boldsymbol{x}_n) - \boldsymbol{f}(\boldsymbol{x}_n + (b^i - x_n^i)\boldsymbol{e}_i)]$, $i = 1, 2, \dots, d$, where $\boldsymbol{e}_i$ is the ith coordinate vector, $\boldsymbol{x}_n^{\mathrm{T}} = [x_n^1, \dots, x_n^d]$ and $\boldsymbol{b}^{\mathrm{T}} = [b^1, \dots, b^d]$ a fixed vector. (Note that the arguments in the two $\boldsymbol{f}$-vectors are the same except for the ith one, which is x_n^i in the first, and b^i in the second.) If we let $\boldsymbol{\Delta}_n = [\boldsymbol{\Delta}_{n,1}, \dots, \boldsymbol{\Delta}_{n,d}]$, a “difference quotient matrix” of order d, the regula falsi becomes $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \boldsymbol{\Delta}_n^{-1}\boldsymbol{f}(\boldsymbol{x}_n)$. As in the one-dimensional case, the method converges no faster than linearly, if at all (Ortega and Rheinboldt [2000, p. 366]).

Section 4.5. The secant method is also rather old; it has been used, for example, in fifteenth-century Indian texts to compute the sine function (Plofker [1996]).

The heuristic motivation of Theorem 4.5.1 uses some simple facts from the theory of linear difference equations with constant coefficients. A reader not familiar with these may wish to consult Henrici [1988, Sect. 7.4 and Notes to Sect. 7.4 on p. 663].

Like the method of false position, the secant method, too, can be extended to $\mathbb{R}^d$ in many different ways (see, e.g., Ortega and Rheinboldt [2000, Sect. 7.2], Dennis and Schnabel [1996, Chap. 8]), one of which is to replace the vector $\boldsymbol{b}$ (in the Notes to Sect. 4.4) by the vector $\boldsymbol{x}_{n-1}$. Theorem 4.5.2 then remains in force (Ortega and Rheinboldt [2000, Sect. 11.2.9]).

Examples of combined methods mentioned at the end of this section are Dekker's method (Dekker [1969]) and its modification by Brent (Brent [2002]). The latter is implemented in the Matlab function `fzero`.

Section 4.6. The history of Newton's method is somewhat shrouded in obscurity. Newton's original ideas on the subject, around 1669, were considerably more complicated and not even remotely similar to what is now conceived to be his method. Raphson in approximately 1690 gave a simplified version of Newton's algorithm, possibly without knowledge of Newton's work. In the English literature, the method is therefore often called the Newton–Raphson method. According to Kollerstrom [1992] and Ypma [1995], Newton's and Raphson's procedures are both purely algebraic without mention of derivatives (or fluxions, as it were). They credit Simpson with being the first to give a calculus description of Newton's method in 1740, without referring either to Newton or to Raphson. As noted by Ypma [loc. cit.], Simpson applied Newton's method even to a 2 × 2 system of nonlinear equations. The modern version of Newton's iteration seems to appear first in a paper by Fourier published posthumously in 1831. See also Alexander [1996] for further historical comments.

Global convergence results for Newton's method analogous to the one in Example 4.6.4 exist also in higher dimension; see, for example, Ortega and Rheinboldt [2000, Sects. 13.3.4 and 13.3.7]. Local convergence, including its quadratic order, is now well established; see Ortega and Rheinboldt [2000, Sect. 12.6 and NR 12.6-1] for precise statements and for a brief but informative history. Crucial in this development was the work of Kantorovich, who studied Newton's method not only in finite-dimensional, but also in infinite-dimensional spaces. A good reference for the latter is Kantorovich and Akilov [1982, Chap. 18]. For a modification of Newton's method which is cubically convergent but requires an extra evaluation of f, see Ortega and Rheinboldt [2000, p. 315].

Section 4.7. Ostrowski [1973, Chap. 4, Sect. 2] calls α a point of attraction for iteration (4.69) if for any x_0 in a sufficiently close neighborhood of α one has $x_n \to \alpha$, and a point of repulsion otherwise. Theorem 4.7.1 tells us that α is a point of attraction if $|\varphi'(\alpha)| < 1$; it is clearly a point of repulsion if $|\varphi'(\alpha)| > 1$. An analogous situation holds in $\mathbb{R}^d$ (Ostrowski [loc. cit., Chap. 22]): if the Jacobian matrix $\partial\boldsymbol{\varphi}/\partial\boldsymbol{x}$ at $\boldsymbol{x} = \boldsymbol{\alpha}$ has spectral radius < 1, then $\boldsymbol{\alpha}$ is a point of attraction and hence the fixed point iteration is locally convergent. If the spectral radius is >1, then $\boldsymbol{\alpha}$ is a point of repulsion.

Section 4.8. An unusually detailed treatment of algebraic equations and their numerical solution, especially by older methods, is given in the French work by Durand [1960]. More recent texts are McNamee [2007] and Kyurkchiev [1998], the latter focusing more specifically on iterative methods for computing all roots of an algebraic equation simultaneously and in particular studying "critical" regions in the complex plane that give rise to divergence if the initial approximations are contained therein. Also focusing on simultaneous computation of all the roots is the book by Petković [2008]. Algebraic equations are special enough that detailed information can be had about the location of their roots, and a number of methods can be devised specifically tailored to them. Good accounts of localization theorems can be found in Householder [1970, Chap. 2] and Marden [1966]. Among the classical methods appropriate for algebraic equations, the best known is Graeffe's method, which basically attempts to separate the moduli of the roots by successive squaring. Another is Cauchy's method – a quadratic extension of Newton's method – which requires second derivatives and is thus applied more readily to polynomial equations than to general nonlinear equations. Combined with the more recent method of Muller [1956] – a quadratic extension of the secant method – it can be made the basis of a reliable rootfinder (Young and Gregory [1988, Vol. 1, Sect. 5.4]). Among contemporary methods, mention should be made of the Lehmer–Schur method (Lehmer [1961], Henrici [1988, Sect. 6.10]), which constructs a sequence of shrinking circular disks in the complex plane eventually capturing a root, and of Rutishauser's QD algorithm (Rutishauser [1957], Henrici [1988, Sect. 7.6]), which under appropriate separation assumptions allows all zeros of a polynomial to be computed simultaneously. The same global character is shared by the Durand–Kerner method, which is basically Newton's method applied to the system of equations $a_i = \sigma_{d-i}(\alpha_1, \dots, \alpha_d)$, $i = 0, 1, \dots, d-1$, expressing the coefficients of the polynomial in terms of the elementary symmetric functions in the zeros. (The same method, incidentally, was already used by Weierstrass [1891] to prove the fundamental theorem of algebra.) For other global methods, see Werner [1982]. Iterative methods carried out in complex interval arithmetic (based on circular disks or rectangles) are studied in Petković [1989]. From a set of initial complex intervals, each containing a zero of the polynomial, they generate a sequence of complex intervals encapsulating the respective zeros ever more closely. The midpoints of these intervals, taken to approximate the zeros, then come equipped with ready-made error estimates.

Section 4.8.1 (a). For Ostrowski's theorem, see Ostrowski [1954], and for literature on the numerical properties of Horner's scheme and more efficient schemes, Gautschi [1975a, Sect. 1.5.1(vi)].

The adverse accumulation of error in polynomial deflation can be mitigated somewhat by a more careful deflation algorithm, which depends on the relative magnitude of the zero being removed; see Peters and Wilkinson [1971] and Cohen [1994].

Section 4.8.2. The accelerated Newton method is due independently to Kahan and Maehly (cf. Wilkinson [1988, p. 480]). A proof of Theorem 4.8.1 can be found in Stoer and Bulirsch [2002, Sect. 5.5].

Section 4.9. Among the many other iterative methods for solving systems of nonlinear equations, we mention the nonlinear analogues of the well-known iterative methods for solving linear systems. Thus, for example, the nonlinear Gauss–Seidel method consists of solving the d single equations $f^i(x_{n+1}^1, \dots, x_{n+1}^{i-1}, t, x_n^{i+1}, \dots, x_n^d) = 0, i = 1, 2, \dots, d$, for t and letting $x_{n+1}^i = t$. The solution of each of these equations will in turn involve some one-dimensional iterative method, for example, Newton's method, which would constitute "inner iterations" in the "outer iteration" defined by Gauss–Seidel. Evidently, any pair of iterative methods can be so combined. Indeed, the roles can also be reversed, for example, by combining Newton's method for systems of nonlinear equations with the Gauss–Seidel method for solving the linear systems in Newton's method. Newton's method then becomes the outer iteration and Gauss–Seidel the inner iteration. Still other methods involve homotopies (or continuation), embedding the given system of nonlinear equations in a one-parameter family of equations and approaching the desired solution via a sequence of intermediate solutions corresponding to appropriately chosen values of the parameter. Each of these intermediate solutions serves as an initial approximation to the next solution. Although the basic idea of such methods is simple, many implementational details must be worked out to make it successful; for a discussion of these, see Allgower and Georg [2003]. The application of continuation methods to polynomial systems arising in engineering and scientific problems is considered in Morgan [1987]. Both texts contain computer programs. Also see Watson et al. [1987] for a software implementation.

Section 4.9.1. If one is interested in individual components, rather than just in norms, one can refine the contraction property by defining a map $\boldsymbol{\varphi}$ to be Γ-contractive on $\mathcal{D}$ if there exists a nonnegative matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{d\times d}$, with spectral radius <1, such that $|\boldsymbol{\varphi}(\boldsymbol{x}) - \boldsymbol{\varphi}(\boldsymbol{x}^*)| \leq \boldsymbol{\Gamma}|\boldsymbol{x} - \boldsymbol{x}^*|$ (componentwise) for all $\boldsymbol{x}, \boldsymbol{x}^* \in \mathcal{D}$. The contraction mapping principle then extends naturally to Γ-contractive maps (cf. Ortega and Rheinboldt [2000, Chap. 13]). Other generalizations of the contraction mapping principle involve perturbations of the map $\boldsymbol{\varphi}$ (Ortega and Rheinboldt [2000, Chap. 12, Sect. 2]).

Section 4.9.2. For quasi-Newton methods (also called modification, or update, methods), see, for example, Ortega and Rheinboldt [2000, Chap. 7, Sect. 3], Dennis and Schnabel [1996, Chap. 6]. As with any iterative method, the increment vector (e.g., the modified Newton increment) may be multiplied at each step by an appropriate scalar to ensure that $\|\boldsymbol{f}(\boldsymbol{x}_{n+1})\| < \|\boldsymbol{f}(\boldsymbol{x}_n)\|$. This is particularly advisable during the initial stages of the iteration.

Exercises and Machine Assignments to Chapter 4

Exercises

1. The following sequences all converge to zero as $n \to \infty$:

   $$v_n = n^{-10}, \quad w_n = 10^{-n}, \quad x_n = 10^{-n^2}, \quad y_n = n^{10}\cdot 3^{-n}, \quad z_n = 10^{-3\cdot 2^n}.$$

   Indicate the type of convergence by placing a check mark in the appropriate position in the following table.

Type of Convergence	v	w	x	y	z
Sublinear					
Linear					
Superlinear					
Quadratic					
Cubic					
None of the above					

2. Suppose a positive sequence $\{\varepsilon_n\}$ converges to zero with order $p > 0$. Does it then also converge to zero with order p' for any $0 < p' < p$?
3. The sequence $\varepsilon_n = \mathrm{e}^{-e^n}$, $n = 0, 1, \dots$, clearly converges to 0 as $n \to \infty$. What is the order of convergence?
4. Give an example of a positive sequence $\{\varepsilon_n\}$ converging to zero in such a way that $\lim_{n\to\infty} \frac{\varepsilon_{n+1}}{\varepsilon_n^p} = 0$ for some $p > 1$, but not converging (to zero) with any order $p' > p$.

5. Suppose $\{x_n\}$ converges linearly to α in the sense that $\lim_{n\to\infty} \frac{x_{n+1}-\alpha}{x_n-\alpha} = c,\ 0 < |c| < 1$.

   (a) Define $x_n^* = \frac{1}{2}(x_n + x_{n-1}),\ n = 1, 2, 3, \ldots$. Clearly, $x_n^* \to \alpha$. Does $\{x_n^*\}$ converge appreciably faster than $\{x_n\}$? Explain by determining the asymptotic error constant.
   (b) Do the same for $x_n^* = \sqrt{x_n x_{n-1}}$, assuming $x_n > 0$ for all n and $\alpha > 0$.

6. Let $\{x_n\}$ be a sequence converging to α linearly with asymptotic error constant c,

$$\lim_{n\to\infty} \frac{x_{n+1}-\alpha}{x_n-\alpha} = c, \quad |c| < 1,$$

   and assume that $x_n \neq \alpha$ for all n.

   (a) Derive Aitken's Δ^2-process (4.21) by assuming two consecutive ratios in the above limit relation (say, for n and $n+1$) to be equal to c.
   (b) Show that the sequence $\{x_n'\}$ in Aitken's Δ^2-process is well defined for n sufficiently large.
   (c) Prove (4.22).

7. Given an iterative method of order p and asymptotic error constant $c \neq 0$, define a new iterative method consisting of m consecutive steps of the given method. Determine the order of this new iterative method and its asymptotic error constant. Hence justify the definition of the efficiency index given near the end of Sect. 4.2.
8. Consider the equation

$$\frac{1}{x-1} + \frac{2}{x+3} + \frac{4}{x-5} - 1 = 0.$$

   (a) Discuss graphically the number of real roots and their approximate location.
   (b) Are there any complex roots?

9. Consider the equation $x \tan x = 1$.

   (a) Discuss the real roots of this equation: their number, approximate location, and symmetry properties. Use appropriate graphs.
   (b) How many bisections would be required to find the smallest positive root to within an error of $\frac{1}{2} \times 10^{-8}$? (Indicate the initial approximations.) Is your answer valid for all roots?
   (c) Are there any complex roots? Explain.

10. Consider the quadratic equation $x^2 - p = 0,\ p > 0$. Suppose its positive root $\alpha = \sqrt{p}$ is computed by the method of false position starting with two numbers a, b satisfying $0 < a < \alpha < b$. Determine the asymptotic error constant c as a

function of b and α. What are the conditions on b for $0 < c < \frac{1}{2}$ to hold, that is, for the method of false position to be (asymptotically) faster than the bisection method?

11. The equation $x^2 - a = 0$ (for the square root $\alpha = \sqrt{a}$) can be written equivalently in the form

$$x = \varphi(x)$$

in many different ways, for example,

$$\varphi(x) = \frac{1}{2}\left(x + \frac{a}{x}\right), \quad \varphi(x) = \frac{a}{x}, \quad \varphi(x) = 2x - \frac{a}{x}.$$

Discuss the convergence (or nonconvergence) behavior of the iteration $x_{n+1} = \varphi(x_n)$, $n = 0, 1, 2, \ldots$, for each of these three iteration functions. In case of convergence, determine the order of convergence.

12. Under the assumptions of Theorem 4.5.1, show that the secant method cannot converge faster than with order $p = \frac{1}{2}(1 + \sqrt{5})$ if the (simple) root α of $f(x) = 0$ satisfies $f''(\alpha) \neq 0$.
13. Let $\{x_n\}$ be a sequence converging to α. Suppose the errors $e_n = |x_n - \alpha|$ satisfy $e_{n+1} \leq M e_n^2 e_{n-1}$ for some constant $M > 0$. What can be said about the order of convergence?
14. Suppose the equation $f(x) = 0$ has a simple root α, and $f''(\alpha) = 0$, $f'''(\alpha) \neq 0$. Provide heuristics in the manner of the text preceding Theorem 4.5.1 showing that the secant method in this case converges quadratically.
15. (a) Consider the iteration $x_{n+1} = x_n^3$. Give a detailed discussion of the behavior of the sequence $\{x_n\}$ in dependence of x_0.
    (b) Do the same as (a), but for $x_{n+1} = x_n^{1/3}$, $x_0 > 0$.
16. Consider the iteration

$$x_{n+1} = \varphi(x_n), \quad \varphi(x) = \sqrt{2 + x}.$$

(a) Show that for any positive x_0 the iterates x_n remain on the same side of $\alpha = 2$ as x_0 and converge monotonically to α.
(b) Show that the iteration converges globally, that is, for any $x_0 > 0$, and not faster than linearly (unless $x_0 = 2$).
(c) If $0 < x_0 < 2$, how many iteration steps are required to obtain α with an error less than 10^{-10}?

17. Consider the equation $x = \cos x$.

(a) Show graphically that there exists a unique positive root α. Indicate, approximately, where it is located.
(b) Prove local convergence of the iteration $x_{n+1} = \cos x_n$.

(c) For the iteration in (b) prove: if $x_n \in [0, \frac{\pi}{2}]$, then

$$|x_{n+1} - \alpha| < \left(\sin \frac{\alpha + \pi/2}{2}\right) |x_n - \alpha|.$$

In particular, one has global convergence on $[0, \frac{\pi}{2}]$.

(d) Show that Newton's method applied to $f(x) = 0$, $f(x) = x - \cos x$, also converges globally on $[0, \frac{\pi}{2}]$.

18. Consider the equation

$$x = \mathrm{e}^{-x}.$$

(a) Show that there is a unique real root α and determine an interval containing it.

(b) Show that the fixed point iteration $x_{n+1} = \mathrm{e}^{-x_n}$, $n = 0, 1, 2, \ldots,$ converges locally to α and determine the asymptotic error constant.

(c) Illustrate graphically that the iteration in (b) actually converges globally, that is, for arbitrary $x_0 > 0$. Then prove it.

(d) An equivalent equation is

$$x = \ln \tfrac{1}{x}.$$

Does the iteration $x_{n+1} = \ln \frac{1}{x_n}$ also converge locally? Explain.

19. Consider the equation

$$\tan x + \lambda x = 0, \quad 0 < \lambda < 1.$$

(a) Show graphically, as simply as possible, that in the interval $[\frac{1}{2}\pi, \pi]$ there is exactly one root α.

(b) Does Newton's method converge to the root $\alpha \in [\frac{1}{2}\pi, \pi]$ if the initial approximation is taken to be $x_0 = \pi$? Justify your answer.

20. Consider the equation

$$f(x) = 0, \; f(x) = \ln^2 x - x - 1, \quad x > 0.$$

(a) Graphical considerations suggest that there is exactly one positive root α, and that $0 < \alpha < 1$. Prove this.

(b) What is the largest positive $b \leq 1$ such that Newton's method, started with $x_0 = b$, converges to α?

21. Consider "Kepler's equation"

$$f(x) = 0, \quad f(x) = x - \varepsilon \sin x - \eta, \; 0 < |\varepsilon| < 1, \; \eta \in \mathbb{R},$$

where ε, η are parameters constrained as indicated.

(a) Show that for each ε, η there is exactly one real root $\alpha = \alpha(\varepsilon, \eta)$. Furthermore, $\eta - |\varepsilon| \le \alpha(\varepsilon, \eta) \le \eta + |\varepsilon|$.

(b) Writing the equation in fixed point form

$$x = \varphi(x), \quad \varphi(x) = \varepsilon \sin x + \eta,$$

show that the fixed point iteration $x_{n+1} = \varphi(x_n)$ converges for arbitrary starting value x_0.

(c) Let m be an integer such that $m\pi < \eta < (m+1)\pi$. Show that Newton's method with starting value

$$x_0 = \begin{cases} (m+1)\pi & \text{if } (-1)^m \varepsilon > 0, \\ m\pi & \text{otherwise} \end{cases}$$

is guaranteed to converge (monotonically) to $\alpha(\varepsilon, \eta)$.

(d) Estimate the asymptotic error constant c of Newton's method.

22. (a) Devise an iterative scheme, using only addition and multiplication, for computing the reciprocal $\frac{1}{a}$ of some positive number a. {*Hint*: use Newton's method. For a cubically convergent scheme, see Ex. 40.}

(b) For what positive starting values x_0 does the algorithm in (a) converge? What happens if $x_0 < 0$?

(c) Since in (binary) floating-point arithmetic it suffices to find the reciprocal of the mantissa, assume $\frac{1}{2} \le a < 1$. Show, in this case, that the iterates x_n satisfy

$$\left| x_{n+1} - \frac{1}{a} \right| < \left| x_n - \frac{1}{a} \right|^2, \quad \text{all } n \ge 0.$$

(d) Using the result of (c), estimate how many iterations are required, at most, to obtain $1/a$ with an error less than 2^{-48}, if one takes $x_0 = \frac{3}{2}$.

23. (a) If $A > 0$, then $\alpha = \sqrt{A}$ is a root of either equation

$$x^2 - A = 0, \quad \frac{A}{x^2} - 1 = 0.$$

Explain why Newton's method applied to the first equation converges for arbitrary starting value $x_0 > 0$, whereas the same method applied to the second equation produces positive iterates x_n converging to α only if x_0 is in some interval $0 < x_0 < b$. Determine b.

(b) Do the same as (a), but for the cube root $\sqrt[3]{A}$ and the equations

$$x^3 - A = 0, \quad \frac{A}{x^3} - 1 = 0.$$

24. (a) Show that Newton's iteration

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right), \quad a > 0,$$

for computing the square root $\alpha = \sqrt{a}$ satisfies

$$\frac{x_{n+1} - \alpha}{(x_n - \alpha)^2} = \frac{1}{2x_n}.$$

Hence, directly obtain the asymptotic error constant.

(b) What is the analogous formula for the cube root?

25. Consider Newton's method

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right), \quad a > 0,$$

for computing the square root $\alpha = \sqrt{a}$. Let $d_n = x_{n+1} - x_n$.

(a) Show that

$$x_n = \frac{a}{d_n + \sqrt{d_n^2 + a}}.$$

(b) Use (a) to show that

$$|d_n| = \frac{d_{n-1}^2}{2\sqrt{d_{n-1}^2 + a}}, \quad n = 1, 2, \ldots.$$

Discuss the significance of this result with regard to the overall behavior of Newton's iteration.

26. (a) Derive the iteration that results by applying Newton's method to $f(x) := x^3 - a = 0$ to compute the cube root $\alpha = a^{\frac{1}{3}}$ of $a > 0$.

(b) Consider the equivalent equation $f_\lambda(x) = 0$, where $f_\lambda(x) = x^{3-\lambda} - ax^{-\lambda}$, and determine λ so that Newton's method converges cubically. Write down the resulting iteration in its simplest form.

27. Consider the two (equivalent) equations

$$\text{(A)} \quad x \ln x - 1 = 0, \qquad \text{(B)} \quad \ln x - \frac{1}{x} = 0.$$

(a) Show that there is exactly one positive root and find a rough interval containing it.

(b) For both (A) and (B), determine the largest interval on which Newton's method converges. {*Hint*: investigate the convexity of the functions involved.}

(c) Which of the two Newton iterations converges asymptotically faster?

28. Prove Theorem 4.6.1.

29. Consider the equation

$$f(x) = 0, \quad \text{where } f(x) = \tan x - cx,\ 0 < c < 1.$$

(a) Show that the smallest positive root α is in the interval $(\pi, \frac{3}{2}\pi)$.

(b) Show that Newton's method started at $x_0 = \pi$ is guaranteed to converge to α if c is small enough. Exactly how small does c have to be?

30. We saw in Sect. 4.1.1 that the equation

$$\cos x \cosh x - 1 = 0$$

has exactly two roots $\alpha_n < \beta_n$ in each interval $[-\frac{\pi}{2} + 2n\pi, \frac{\pi}{2} + 2n\pi]$, $n = 1, 2, 3 \ldots$. Show that Newton's method applied to this equation converges to α_n when initialized by $x_0 = -\frac{\pi}{2} + 2n\pi$ and to β_n when initialized by $x_0 = \frac{\pi}{2} + 2n\pi$.

31. In the engineering of circular shafts the following equation is important for determining critical angular velocities:

$$f(x) = 0, \quad f(x) = \tan x + \tanh x,\ x > 0.$$

(a) Show that there are infinitely many positive roots, exactly one, α_n, in each interval $[(n - \frac{1}{2})\pi, n\pi]$, $n = 1, 2, 3, \ldots$.

(b) Determine $\lim_{n\to\infty}(n\pi - \alpha_n)$.

(c) Discuss the convergence of Newton's method when started at $x_0 = n\pi$.

32. The equation

$$f(x) := x \tan x - 1 = 0,$$

if written as $\tan x = 1/x$ and each side plotted separately, can be seen to have infinitely many positive roots, one, α_n, in each interval $[n\pi, (n + \frac{1}{2})\pi]$, $n = 0, 1, 2, \ldots$.

(a) Show that the smallest positive root α_0 can be obtained by Newton's method started at $x_0 = \frac{\pi}{4}$.

(b) Show that Newton's method started with $x_0 = (n + \frac{1}{4})\pi$ converges monotonically decreasing to α_n if $n \geq 1$.

(c) Expanding α_n (formally) in inverse powers of πn,

$$\alpha_n = \pi n + c_0 + c_1(\pi n)^{-1} + c_2(\pi n)^{-2} + c_3(\pi n)^{-3} + \cdots,$$

determine $c_0, c_1, c_2, \ldots, c_9$. {*Hint*: use the Maple `series` command.}

(d) Use the Matlab function `fzero` to compute α_n for $n = 1 : 10$ and compare the results with the approximation furnished by the expansion in (c).

33. Consider the equation

$$f(x) = 0, \quad f(x) = x \sin x - 1, \; 0 \leq x \leq \pi.$$

 (a) Show graphically (as simply as possible) that there are exactly two roots in the interval $[0, \pi]$ and determine their approximate locations.
 (b) What happens with Newton's method when it is started with $x_0 = \frac{1}{2}\pi$? Does it converge, and if so, to which root? Where do you need to start Newton's method to get the other root?

34. (Gregory, 1672) For an integer $n \geq 1$, consider the equation

$$f(x) = 0, \quad f(x) = x^{n+1} - b^n x + ab^n, \quad a > 0, \; b > 0.$$

 (a) Prove that the equation has exactly two distinct positive roots if and only if

$$a < \frac{n}{(n+1)^{1+\frac{1}{n}}}\, b.$$

 {*Hint*: analyze the convexity of f.}
 (b) Assuming that the condition in (a) holds, show that Newton's method converges to the smaller positive root, when started at $x_0 = a$, and to the larger one, when started at $x_0 = b$.

35. Suppose the equation $f(x) = 0$ has the root α with exact multiplicity $m \geq 2$, and Newton's method converges to this root. Show that convergence is linear, and determine the asymptotic error constant.

36. (a) Let α be a double root of the equation $f(x) = 0$, where f is sufficiently smooth near α. Show that the "doubly relaxed" Newton's method

$$x_{n+1} = x_n - 2\,\frac{f(x_n)}{f'(x_n)},$$

 if it converges to α, does so at least quadratically. Obtain the condition under which the order of convergence is exactly 2, and determine the asymptotic error constant c in this case.
 (b) What are the analogous statements in the case of an m-fold root?

37. Consider the equation $x \ln x = a$.

 (a) Show that for each $a > 0$ the equation has a unique positive root, $x = x(a)$.
 (b) Prove that

$$x(a) \sim \frac{a}{\ln a} \quad \text{as } a \to \infty$$

 (i.e., $\lim_{a\to\infty} \frac{x(a)\ln a}{a} = 1$). {*Hint*: use the rule of Bernoulli–L'Hospital.}
 (c) For large a improve the approximation given in (b) by applying one step of Newton's method.

38. The equation $x^2 - 2 = 0$ can be written as a fixed point problem in different ways, for example,

$$\text{(a) } x = \frac{2}{x}, \quad \text{(b) } x = x^2 + x - 2, \quad \text{(c) } x = \frac{x+2}{x+1}.$$

How does the fixed point iteration perform in each of these three cases? Be as specific as you can.

39. Show that

$$x_{n+1} = \frac{x_n(x_n^2 + 3a)}{3x_n^2 + a}, \quad n = 0, 1, 2, \dots,$$

is a method for computing $\alpha = \sqrt{a}$, $a > 0$, which converges cubically to α (for suitable x_0). Determine the asymptotic error constant. (Cf. also Ex. 43(e) with $\lambda = 2$.)

40. Consider the fixed point iteration

$$x_{n+1} = \varphi(x_n), \quad n = 0, 1, 2, \dots,$$

where

$$\varphi(x) = Ax + Bx^2 + Cx^3.$$

(a) Given a positive number α, determine the constants A, B, C such that the iteration converges locally to $1/\alpha$ with order $p = 3$. {This will give a cubically convergent method for computing the reciprocal $1/\alpha$ of α which uses only addition, subtraction, and multiplication.}

(b) Determine the precise condition on the initial error $\varepsilon_0 = x_0 - \frac{1}{\alpha}$ for the iteration to converge.

41. The equation $f(x) := x^2 - 3x + 2 = 0$ has the roots 1 and 2. Written in fixed point form $x = \frac{1}{\omega}(x^2 - (3-\omega)x + 2)$, $\omega \neq 0$, it suggests the iteration

$$x_{n+1} = \frac{1}{\omega}(x_n^2 - (3-\omega)x_n + 2), \quad n = 0, 1, 2, \dots \quad (\omega \neq 0).$$

(a) Identify as large an ω-interval as possible such that for any ω in this interval the iteration converges to 1 (when $x_0 \neq 1$ is suitably chosen).

(b) Do the same as (a), but for the root 2 (and $x_0 \neq 2$).

(c) For what value(s) of ω does the iteration converge quadratically to 1?

(d) Interpret the algorithm produced in (c) as a Newton iteration for some equation $F(x) = 0$, and exhibit F. Hence discuss for what initial values x_0 the method converges.

42. Let α be a simple zero of f and $f \in C^p$ near α, where $p \geq 3$. Show: if $f''(\alpha) = \cdots = f^{(p-1)}(\alpha) = 0$, $f^{(p)}(\alpha) \neq 0$, then Newton's method applied to $f(x) = 0$ converges to α locally with order p. Determine the asymptotic error constant.

43. The iteration

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n) - \frac{1}{2} f''(x_n) \frac{f(x_n)}{f'(x_n)}}, \quad n = 0, 1, 2, \dots,$$

for solving the equation $f(x) = 0$ is known as *Halley's method.*

(a) Interpret Halley's method geometrically as the intersection with the x-axis of a hyperbola with asymptotes parallel to the x- and y-axes that is osculatory to the curve $y = f(x)$ at $x = x_n$ (i.e., is tangent to the curve at this point and has the same curvature there).

(b) Show that the method can, alternatively, be interpreted as applying Newton's method to the equation $g(x) = 0$, $g(x) := f(x)/\sqrt{f'(x)}$.

(c) Assuming α is a simple root of the equation, and $x_n \to \alpha$ as $n \to \infty$, show that convergence is exactly cubic, unless the "Schwarzian derivative"

$$(Sf)(x) := \frac{f'''(x)}{f'(x)} - \frac{3}{2}\left(\frac{f''(x)}{f'(x)}\right)^2$$

vanishes at $x = \alpha$, in which case the order of convergence is larger than 3.

(d) Is Halley's method more efficient than Newton's method as measured in terms of the efficiency index?

(e) How does Halley's method look in the case $f(x) = x^\lambda - a$, $a > 0$? (Compare with Ex. 39.)

44. Let $f(x) = x^d + a_{d-1}x^{d-1} + \cdots + a_0$ be a polynomial of degree $d \geq 2$ with real coefficients a_i.

(a) In analogy to (4.75), let

$$f(x) = (x^2 - tx - s)(x^{d-2} + b_{d-1}x^{d-3} + \cdots + b_2) + b_1(x - t) + b_0.$$

Derive a recursive algorithm for computing $b_{d-1}, b_{d-2}, \dots, b_1, b_0$ in this order.

(b) Suppose α is a complex zero of f. How can f be deflated to remove the pair of zeros $\alpha, \overline{\alpha}$?

(c) (Bairstow's method) Devise a method based on the division algorithm of (a) to compute a quadratic factor of f. Use Newton's method for a system of two equations in the two unknowns t and s and exhibit recurrence formulae for computing the elements of the 2×2 Jacobian matrix of the system.

45. Let $p(t)$ be a monic polynomial of degree n. Let $\boldsymbol{x} \in \mathbb{C}^n$ and define

$$f_\nu(\boldsymbol{x}) = [x_1, x_2, \dots, x_\nu]\, p, \quad \nu = 1, 2, \dots, n,$$

to be the divided differences of p relative to the coordinates x_μ of $\boldsymbol{x}$. Consider the system of equations

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}, \quad [\boldsymbol{f}(\boldsymbol{x})]^{\mathrm{T}} = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \dots, f_n(\boldsymbol{x})].$$

(a) Let $\alpha^{\mathrm{T}} = [\alpha_1, \alpha_2, \dots, \alpha_n]$ be the zeros of p, assumed mutually distinct. Show that α is, except for a permutation of the components, the unique solution of $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$. {*Hint*: use Newton's formula of interpolation.}

(b) Describe the application of Newton's iterative method to the preceding system of nonlinear equations, $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$. {*Hint*: use Chap. 2, Ex. 58.}

(c) Discuss to what extent the procedure in (a) and (b) is valid for nonpolynomial functions p.

46. For the equation $f(x) = 0$ define

$$\begin{aligned} y^{[0]}(x) &= x, \\ y^{[1]}(x) &= \frac{1}{f'(x)}, \\ &\dots\dots\dots\dots\dots\dots\dots\dots \\ y^{[m]}(x) &= \frac{1}{f'(x)} \frac{\mathrm{d}}{\mathrm{d}x} y^{[m-1]}(x), \quad m = 2, 3, \dots. \end{aligned}$$

Consider the iteration function

$$\varphi_r(x) := \sum_{m=0}^{r} (-1)^m \frac{y^{[m]}(x)}{m!} [f(x)]^m.$$

(When $r = 1$ this is the iteration function for Newton's method.) Show that $\varphi_r(x)$ defines an iteration $x_{n+1} = \varphi_r(x_n)$, $n = 0, 1, 2, \dots$, converging locally with exact order $p = r + 1$ to a root α of the equation if $y^{[r+1]}(\alpha) f'(\alpha) \neq 0$.

Machine Assignments

1.(a) Write a Matlab program that computes (in Matlab double precision) the expanded form $p(x) = x^5 - 5x^4 + 10x^3 - 10x^2 + 5x - 1$ of the polynomial $(x - 1)^5$. Run the program to print $p(x)/\text{prec}$ for 200 equally spaced x-values in a small neighborhood of $x = 1$ (say, $0.9986 \le x \le 1.0014$), where prec = eps is the Matlab (double-precision) machine precision. Prepare a piecewise linear plot of the results. Explain what you observe. What is the "uncertainty interval" for the numerical root corresponding to the mathematical root $x = 1$?

(b) Do the same as (a), but for the polynomial $p(x) = x^5 - 100x^4 + 3995x^3 - 79700x^2 + 794004x - 3160080$, the expanded form of $(x-18)(x-19)(x-20)(x-21)(x-22)$. Do the computation in Matlab single precision and take for prec the respective machine precision. Examine a small interval around $x = 22$ (say, $21.675 \le x \le 22.2$).

2. Consider the equation

$$\frac{1}{2}x - \sin x = 0.$$

(a) Show that the only positive root is located in the interval $[\frac{1}{2}\pi, \pi]$.

(b) Compute the root to 7, 15, and 33 decimal places

(b1) by the method of bisection, using the Matlab function `sbisec` of Sect. 4.3.1 with starting values $a = \frac{1}{2}\pi$, $b = \pi$;

(b2) by the method of false position, using the Matlab function `sfalsepos` of Sect. 4.4 with the same starting values as in (b1);

(b3) by the secant method, using the Matlab function `ssecant` of Sect. 4.5 with the same starting values as in (b1);

(b4) by Newton's method, using the Matlab function `snewton` of Sect. 4.6 with an appropriate starting value a.

In all cases print the number of iterations required.

3. For an integer $n \ge 2$, consider the equation

$$\frac{x + x^{-1}}{x^n + x^{-n}} = \frac{1}{n}.$$

(a) Write the equation equivalently as a polynomial equation, $p_n(x) = 0$.

(b) Use Descartes' rule of sign[4] (applied to $p_n(x) = 0$) to show that there are exactly two positive roots, one in (0,1), the other in (1,∞). How are they related? Denote the larger of the two roots by α_n (>1). It is known (you do not have to prove this) that

$$1 < \alpha_{n+1} < \alpha_n < 3, \quad n = 2, 3, 4, \ldots.$$

(c) Write and run a program applying the bisection method to compute α_n, $n = 2, 3, \ldots, 20$, to six correct decimal places after the decimal point, using $[1, 3]$ as initial interval for α_2, and $[1, \alpha_n]$ as initial interval for α_{n+1} ($n \ge 2$). For each n, count the number of iterations required. Similarly, apply Newton's method (to the equation $p_n(x) = 0$) to compute α_n to the same accuracy, using the initial value 3 for α_2 and the initial value α_n for α_{n+1} ($n \ge 2$).

[4] Descartes' rule of sign says that if a real polynomial has s sign changes in the sequence of its nonzero coefficients, then it has s positive zeros or a (nonnegative) even number less.

(Justify these choices.) Again, for each n, count the number of iterations required. In both cases, print n, α_n, and the number of iterations. Use a `do while` loop to program either method.

4. Consider the equation
$$x = \mathrm{e}^{-x}.$$

   (a) Implement the fixed-point iteration $x_{n+1} = \mathrm{e}^{-x_n}$ on the computer, starting with $x_0 = 1$ and stopping at the first n for which x_{n+1} agrees with x_n to within the machine precision. Print this value of n and the corresponding x_{n+1}.

   (b) If the equation is multiplied by ω ($\neq 0$ and $\neq -1$) and x is added on both sides, one gets the equivalent equation
$$x = \frac{\omega \mathrm{e}^{-x} + x}{1+\omega}.$$
   Under what condition on ω does the fixed-point iteration for this equation converge faster (ultimately) than the iteration in (a)? {This condition involves the root α of the equation.}

   (c) What is the optimal choice of ω? Verify it by a machine computation in a manner analogous to (a).

5. Consider the boundary value problem
$$y'' + \sin y = 0, \quad 0 \le x \le \frac{1}{4}\pi,$$
$$y(0) = 0, \quad y\left(\frac{1}{4}\pi\right) = 1,$$
which describes the angular motion of a pendulum.

   (a) Use the Matlab integrator `ode45.m` to compute and plot the solution $u(x;s)$ of the associated initial value problem
$$u'' + \sin u = 0, \quad 0 \le x \le \frac{1}{4}\pi,$$
$$u(0) = 0, \quad u'(0) = s$$
   for $s = .2(.2)2$.

   (b) Write and run a Matlab program that applies the method of bisection, with tolerance 0.5e−12, to the equation $f(s) = 0$, $f(s) = u(1;s) - 1$. Use the plots of (a) to choose starting values s_0, s_1 such that $f(s_0) < 0$, $f(s_1) > 0$. Print the number of bisections and the value of s so obtained. {*Suggestion*: use a nonsymbolic version `bisec.m` of the program `sbisec.m` of Sect. 4.3.1.}

   (c) Plot the solution curve $y(x) = u(x;s)$ of the boundary value problem, with s as obtained in (b).

6. The boundary value problem

$$y'' = g(x, y, y'), \quad 0 \le x \le 1,$$
$$y(0) = y_0, \quad y(1) = y_1$$

may be discretized by replacing the first and second derivatives by centered difference quotients relative to a grid of equally spaced points $x_k = \frac{k}{n+1}$, $k = 0, 1, \dots, n, n+1$.

(a) Interpret the resulting equations as a fixed point problem in $\mathbb{R}^n$ and formulate the respective fixed point iteration.

(b) Write a Matlab program that applies the fixed point iteration of (a) to the problem

$$y'' - y = 0, \quad y(0) = 0, \ y(1) = 1$$

(cf. the first Example of Chap. 7, Sect. 7.1.1). Run the program for $n = 10, 100, 1000, 10000$, stopping the iteration the first time two successive iterates differ by less than .5e$-$14 in the ∞-norm. Print the number of iterations required and the maximum error of the final iterate as an approximation to the exact solution vector. Assuming this error is $O(h^p)$, determine numerically the values of p and of the constant implied in the order term. {*Suggestion*: for solving tridiagonal systems of equations, use the Matlab routine `tridiag.m` of Chap. 2, MA 8(a).}

(c) Apply the fixed point iteration of (a) to the boundary value problem of MA 5. Show that the iteration function is contractive. {*Hint*: use the fact that the symmetric $n \times n$ tridiagonal matrix $\boldsymbol{A}$ with elements -2 on the diagonal and 1 on the two side diagonals has an inverse satisfying $\|\boldsymbol{A}^{-1}\|_\infty \le (n+1)^2/8$.}

7. (a) Solve the finite difference equations obtained in MA 6(c) by Newton's method, using $n = 10, 100, 1000$, and an error tolerance of 0.5e$-$14. Print the number of iterations in each case and plot the respective solution curves. {*Suggestion*: same as in MA 6(b).}

(b) Do the same as in (a) for the boundary value problem

$$y'' = yy', \quad y(0) = 0, \ y(1) = 1,$$

but with $n = 10, 50, 100$, and error tolerance 0.5e$-$6. How would you check your program for correctness?

(c) Show that the fixed point iteration applied to the finite difference equations for the boundary value problem of (b) does not converge. {*Hint*: use $n^2/8 \le \|\boldsymbol{A}^{-1}\|_\infty \le (n+1)^2/8$ for the $n \times n$ tridiagonal matrix $\boldsymbol{A}$ of MA 6(c).}

8. (a) The *Littlewood–Salem–Izumi constant* α_0, defined as the unique solution in $0 < \alpha < 1$ of

$$\int_0^{3\pi/2} \frac{\cos t}{t^\alpha}\, \mathrm{d}t = 0,$$

is of interest in the theory of positive trigonometric sums (cf., e.g., Koumandos [2011, Theorem 2]). Use Newton's method in conjunction with Gaussian quadrature to compute α_0. {*Hint*: you need the Matlab routine `gauss.m` along with the routines `r_jacobi01.m`, `r_jacobi.m`, `r_jaclog.m`, and `mm_log.m` to do the integrations required. All these routines are available on the Web at `http://www.cs.purdue.edu/archives/2002/wxg/codes/SOPQ.html`.}

(b) Do the same as in (a) for the constant α_1, the unique solution in $0 < \alpha < 1$ of

$$\int_0^{5\pi/4} \frac{\cos(t + \pi/4)}{t^\alpha}\, \mathrm{d}t = 0$$

(cf. *ibid*, Theorem 4).

9. (a) Discuss how to simplify the system of nonlinear equations (4.17) for the recurrence coefficients of the polynomials $\{\pi_{k,n}\}_{k=0}^n$, generating the s-orthogonal polynomials $\pi_n = \pi_{n,n}$, when the measure $d\lambda(t) = w(t)\mathrm{d}t$ is symmetric, i.e., the support of $\mathrm{d}\lambda$ is an interval $[-a, a]$, $0 < a \le \infty$, and $w(-t) = w(t)$ for all t with $0 < t \le a$. {*Hint*: first show that the respective monic s-orthogonal polynomial π_n satisfies $\pi_n(-t) = (-1)^n \pi_n(t)$, $t \in [-a, a]$ and similarly $\pi_{k,n}(-t) = (-1)^k \pi_{k,n}(t)$.}

(b) For $n = 2$, $s = 1$, and $s = 2$, explain how the recurrence coefficients β_0, β_1 can be obtained analytically in terms of the moments $\mu_k = \int_{\mathbb{R}} t^k \mathrm{d}\lambda(t)$, $k = 0, 1, 2, \ldots,$ of the measure. Provide numerical answers in the case of the Legendre measure $\mathrm{d}\lambda(t) = \mathrm{d}t$, $t \in [-1, 1]$.

(c) Write a Matlab program for solving the system of nonlinear equations in (a), using the program `fsolve` of the Matlab Optimization Toolbox. Run the program for the Legendre measure and for $n = 2 : 10$ and $s = 1$ and $s = 2$ for each n. Choose initial approximations as deemed useful and apply appropriate $(s + 1)n$-point Gaussian quadrature rules to do the necessary integrations. Print $\beta_0, \beta_1, \ldots, \beta_{n-1}$ and the zeros of π_n.

Selected Solutions to Exercises

6. (a) From

$$x_{n+1} - \alpha = c(x_n - \alpha),$$
$$x_{n+2} - \alpha = c(x_{n+1} - \alpha),$$

solving the first equation for c and substituting the result into the second equation gives

$$(x_n - \alpha)(x_{n+2} - \alpha) = (x_{n+1} - \alpha)^2.$$

Solving for α, one finds

$$\alpha = \frac{x_n x_{n+2} - x_{n+1}^2}{x_{n+2} - 2x_{n+1} + x_n} = x_n - \frac{(x_{n+1} - x_n)^2}{x_{n+2} - 2x_{n+1} + x_n}.$$

Raplacing α by x_n' yields Aitken's Δ^2-process.

(b) We need to show that $\Delta^2 x_n \neq 0$ for n sufficiently large. Let $\varepsilon_n = x_n - \alpha$. We have

$$\frac{\Delta^2 x_n}{\varepsilon_n} = \frac{\Delta^2 \varepsilon_n}{\varepsilon_n} = \frac{\varepsilon_{n+2}}{\varepsilon_{n+1}} \frac{\varepsilon_{n+1}}{\varepsilon_n} - 2\frac{\varepsilon_{n+1}}{\varepsilon_n} + 1 \to (c-1)^2 \text{ as } n \to \infty,$$

from which the assertion follows.

(c) Let $\varepsilon_n' = x_n' - \alpha$. Subtracting α from both sides of

$$x_n' = x_n - \frac{(\Delta x_n)^2}{\Delta^2 x_n}$$

and dividing by ε_n, we get

$$\frac{\varepsilon_n'}{\varepsilon_n} = 1 - \frac{(\Delta \varepsilon_n)^2}{\varepsilon_n \Delta^2 x_n} = 1 - \left(\frac{\Delta \varepsilon_n}{\varepsilon_n}\right)^2 \frac{1}{\Delta^2 x_n / \varepsilon_n}.$$

Thus, by assumption and the result of (b),

$$\lim_{n\to\infty} \frac{\varepsilon_n'}{\varepsilon_n} = 1 - (c-1)^2 \frac{1}{(c-1)^2} = 0,$$

as claimed.

9. (a) Clearly, with α, also $-\alpha$ is a root of the equation. It suffices, therefore, to look at positive roots. Writing the equation in the form $\tan x = 1/x$ and plotting both sides as functions of x for $x > 0$ (see figure on the next page), one sees that there is exactly one root α_k in each of the intervals $I_k = [k\pi, (k+\frac{1}{2})\pi]$, $k = 0, 1, 2, \ldots$. Moreover, as $k \to \infty$, the root α_k approaches $k\pi$ from the right. (Cf. also Ex. 32.)

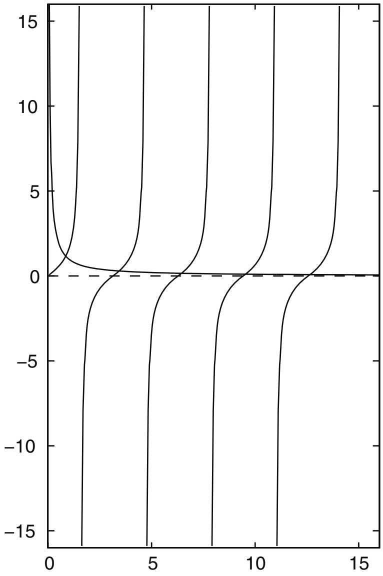
(b) Let $f(x) = x \tan x - 1$. As initial interval, we may take (for example) the interval $[0, \frac{3\pi}{8}]$, since $f(0) = -1$ and $f(\frac{3\pi}{8}) = \frac{3\pi}{8} \tan \frac{3\pi}{8} - 1 = 1.8441\ldots > 0$. We then want n, the number of bisections, such that (cf. (4.33))

$$\frac{3\pi}{8 \cdot 2^n} \leq \frac{1}{2} \cdot 10^{-8},$$

that is,

$$n \geq \left\lceil \frac{8 + \log \frac{3\pi}{4}}{\log\ 2} \right\rceil = 28.$$

This holds for all roots, since $f(k\pi) = -1 < 0$ and $f(k\pi + \frac{3\pi}{8}) = (k\pi + \frac{3\pi}{8}) \tan \frac{3\pi}{8} - 1 > 0, k = 1, 2, 3, \ldots$.



(c) We first apply the addition theorem for the tangent function to write

$$\tan(x + \mathrm{i}y) = \frac{\tan x + \tan(\mathrm{i}y)}{1 - \tan x \tan(\mathrm{i}y)} = \frac{\tan x + \mathrm{i} \tanh y}{1 - \mathrm{i} \tan x \tanh y},$$

or, using the definition of the trigonometric and hyperbolic tangents,

$$\tan(x + \mathrm{i}y) = \frac{\sin x \cosh y + \mathrm{i} \cos x \sinh y}{\cos x \cosh y - \mathrm{i} \sin x \sinh y}.$$

Multiplying numerator and denominator by $\cos x \cosh y + \mathrm{i} \sin x \sinh y$ gives

$$\tan(x + \mathrm{i}y) = \frac{\sin 2x + \mathrm{i} \sinh 2y}{2(\cos^2 x \cosh^2 y + \sin^2 x \sinh^2 y)}.$$

Use the identities $2\cos^2 x = \cos 2x + 1$, $2\sin^2 x = 1 - \cos 2x$ in the denominator, along with $\cosh^2 y - \sinh^2 y = 1$, $\cosh^2 y + \sinh^2 y = \cosh 2y$, to obtain the pretty formula

$$\tan(x + \mathrm{i}y) = \frac{\sin 2x + \mathrm{i}\sinh 2y}{\cos 2x + \cosh 2y}.$$

Now suppose that

$$(x + \mathrm{i}y)\tan(x + \mathrm{i}y) - 1 = 0.$$

Then both the real and imaginary part must vanish, which gives the two equations

$$\begin{aligned} x\sin 2x - y\sinh 2y - \cos 2x - \cosh 2y &= 0, \\ x\sinh 2y + y\sin 2x &= 0. \end{aligned}$$

The second can be written as

$$xy\left(\frac{\sin 2x}{x} + \frac{\sinh 2y}{y}\right) = 0.$$

Since the function in parentheses is always positive, this implies either $x = 0$ or $y = 0$. The latter yields real roots, and the former is impossible, since by the first equation above it would imply $y\sinh 2y + 1 + \cosh 2y = 0$, which is clearly impossible, the left-hand side being >2. Thus, there are no complex roots.

32. (a) We have

$$\begin{aligned} f'(x) &= \tan x + x(1 + \tan^2 x), \\ f''(x) &= 2(1 + \tan^2 x) + 2x\tan x(1 + \tan^2 x) \\ &= 2(1 + \tan^2 x)(1 + x\tan x). \end{aligned}$$

On the interval $[0, \frac{\pi}{2}]$, the function f increases from -1 to $+\infty$ and is convex. Furthermore, $f(\frac{\pi}{4}) = \frac{\pi}{4} - 1 < 0$. We thus need to show that one step of Newton's method with $x_0 = \frac{\pi}{4}$ produces x_1 such that $x_1 < \frac{\pi}{2}$. Since, by convexity, $x_1 > \alpha_0$, from then on, Newton's method will converge monotonically decreasing to α_0. Now,

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = \frac{\pi}{4} - \frac{\frac{\pi}{4} - 1}{1 + 2\cdot\frac{\pi}{4}} = \frac{1 + \frac{\pi^2}{8}}{1 + \frac{\pi}{2}},$$

which is indeed less than $\frac{\pi}{2}$, since

$$1+\frac{\pi^2}{8}<\frac{\pi}{2}\left(1+\frac{\pi}{2}\right).$$

(b) Again, f increases from -1 to $+\infty$ on $[n\pi,(n+\frac{1}{2})\pi]$ and is convex. Since

$$f\left(\left(n+\frac{1}{4}\right)\pi\right)=\left(n+\frac{1}{4}\right)\pi-1>0 \quad \text{for } n\geq 1,$$

Newton's method started at $(n+\frac{1}{4})\pi$ converges monotonically decreasing to α_n.

(c) We have

$$\alpha_n\tan\alpha_n-1=0,$$

and therefore, inserting the expansion for α_n and noting that $\tan(\cdot+\pi n)=\tan(\cdot)$,

$$\begin{aligned}&(\pi n+c_0+c_1(\pi n)^{-1}+c_2(\pi n)^{-2}+\cdots)\\&\quad\times\tan(c_0+c_1(\pi n)^{-1}+c_2(\pi n)^{-2}+\cdots)-1=0.\end{aligned}$$

Evidently, $c_0=0$. Letting $x=(\pi n)^{-1}$ and multiplying through by x gives

$$(1+c_1x^2+c_2x^3+\cdots)\tan(c_1x+c_2x^2+\cdots)-x=0.$$

Using Maple's `series` command, we can find the power series expansion of the left-hand side up to terms of order x^9. Maple produces the coefficients explicitly as functions of $c_1,c_2,\ldots,c_9$. Setting these functions equal to zero and solving successively for the unknowns $c_1,c_2,\ldots,c_9$ yields, after a little bit of algebra, that $c_2=c_4=c_6=\cdots=0$ and

$$\begin{aligned}&c_1=1,\quad c_3=-\frac{4}{3},\quad c_5=\frac{53}{15},\\&c_7=-\frac{1226}{105},\quad c_9=\frac{13597}{315}.\end{aligned}$$

Thus,

$$\begin{aligned}\alpha_n&=\pi n+(\pi n)^{-1}-\frac{4}{3}(\pi n)^{-3}+\frac{53}{15}(\pi n)^{-5}-\frac{1226}{105}(\pi n)^{-7}\\&\quad+\frac{13597}{315}(\pi n)^{-9}+O\left((\pi n)^{-11}\right).\end{aligned}$$

(d) PROGRAMS

```
%EXIV_32D
%
f0='%5.0f %19.15f %19.15f %10.2e\n';
disp(['    n        alpha_n          ' ...
'alpha_n_approx      error'])
for n=1:10
  a0=(n+1/4)*pi;
  a=fzero('equ32',a0);
  x=1/(pi*n);
  an=1/x+x-4*x^3/3+53*x^5/15-1226*x^7/105 ...
    +13597*x^9/315;
  err=an-a;
  fprintf(f0,n,a,an,err)
end

%EQU32  The equation of EXIV_32
%
function y=equ32(x)
y=x*tan(x)-1;
```

OUTPUT

```
>> EXIV_32D
    n        alpha_n          alpha_n_approx      error
    1   3.425618459481728   3.426028729631524   4.10e-04
    2   6.437298179171947   6.437298435880711   2.57e-07
    3   9.529334405361963   9.529334408494419   3.13e-09
    4  12.645287223856643  12.645287223991568   1.35e-10
    5  15.771284874815885  15.771284874827579   1.17e-11
    6  18.902409956860023  18.902409956861607   1.58e-12
    7  22.036496727938566  22.036496727938854   2.88e-13
    8  25.172446326646664  25.172446326646728   6.39e-14
    9  28.309642854452012  28.309642854452026   1.42e-14
   10  31.447714637546234  31.447714637546238   3.55e-15
>>
```

34. (a) We have $f(0) = ab^n$, $f(\infty) = \infty$, and

$$f'(x) = (n+1)x^n - b^n, \quad f''(x) = n(n+1)x^{n-1} > 0 \quad \text{for } x > 0.$$

Thus, $f'(0) < 0$ and f is convex on $[0, \infty]$, so that f has a unique minimum, say at $x = \xi$. Then there are two distinct positive roots precisely when $f(\xi) < 0$. Since

$$\xi = \frac{b}{(n+1)^{\frac{1}{n}}}$$

and

$$\begin{aligned}
f(\xi) &= \frac{b^{n+1}}{(n+1)^{\frac{n+1}{n}}} - b^n \frac{b}{(n+1)^{\frac{1}{n}}} + ab^n \\
&= b^n \left\{ \frac{b}{(n+1)^{1+\frac{1}{n}}} [1-(n+1)] + a \right\} \\
&= b^n \left\{ \frac{-nb}{(n+1)^{1+\frac{1}{n}}} + a \right\},
\end{aligned}$$

the condition amounts to

$$a < \frac{nb}{(n+1)^{1+\frac{1}{n}}}.$$

(b) At the point $x = a$, we have

$$\begin{aligned}
f(a) &= a^{n+1} - b^n a + ab^n = a^{n+1} > 0, \\
f'(a) &= (n+1)a^n - b^n < (n+1)\, \frac{n^n}{(n+1)^{n+1}}\, b^n - b^n \\
&= \left[\left(\frac{n}{n+1} \right)^n - 1 \right] b^n < 0,
\end{aligned}$$

where on the second line the condition in (a) has been used. Since by assumption $a < \frac{n}{n+1}\xi < \xi$, this means that a must be to the left of the smaller root, α_1. By convexity of f, Newton's method started at $x = a$ therefore converges monotonically increasing to α_1. Similarly,

$$\begin{aligned}
f(b) &= b^{n+1} - b^{n+1} + ab^n = ab^n > 0, \\
f'(b) &= (n+1)b^n - b^n = nb^n > 0,
\end{aligned}$$

and $b = (n+1)^{\frac{1}{n}}\xi > \xi$, implying that $b > \alpha_2$, where α_2 is the larger root. By convexity of f, Newton's method started at $x_0 = b$ then converges monotonically decreasing to α_2.

40. (a) Convergence to $\frac{1}{\alpha}$ requires that

$$\frac{1}{\alpha} = \varphi\left(\frac{1}{\alpha}\right) = \frac{A}{\alpha} + \frac{B}{\alpha^2} + \frac{C}{\alpha^3},$$

that is,

$$\alpha^2 A + \alpha B + C = \alpha^2.$$

Cubic convergence requires, in addition, $\varphi'(\frac{1}{\alpha}) = \varphi''(\frac{1}{\alpha}) = 0$, that is,

$$\begin{aligned} \alpha^2 A + 2\alpha B + 3C &= 0, \\ 2\alpha B + 6C &= 0. \end{aligned}$$

We thus have three linear equations in the three unknowns A, B, C. Solving them yields

$$A = 3, \quad B = -3\alpha, \quad C = \alpha^2,$$

giving

$$\varphi(x) = 3x - 3\alpha x^2 + \alpha^2 x^3.$$

(b) Letting

$$\varepsilon_n = x_n - \frac{1}{\alpha},$$

we have from the general theory of fixed point iterations (cf. (4.71)) that

$$\frac{\varepsilon_{n+1}}{\varepsilon_n^3} = \frac{1}{6}\,\varphi'''\left(\frac{1}{\alpha}\right)$$

(since φ''' is constant), that is,

$$\varepsilon_{n+1} = \alpha^2 \varepsilon_n^3, \quad n = 0, 1, 2, \ldots.$$

An easy inductive argument shows that

$$\varepsilon_n = \frac{1}{\alpha}(\alpha\varepsilon_0)^{3^n}.$$

Thus, we have convergence precisely if $\alpha|\varepsilon_0| < 1$.

43. (a) A hyperbola with the lines $x = X_0$ and $y = Y_0$ as asymptotes has the equation

$$(x - X_0)(y - Y_0) - \tfrac{1}{2}a^2 = 0.$$

It intersects the x-axis at

$$x_{n+1} = X_0 - \frac{\frac{1}{2}a^2}{Y_0}.$$

The osculatory requirement of the hyperbola at the point (x_n, f_n) (where $f_n = f(x_n)$, etc.) is expressed by the following three equations:

$$\begin{aligned}(x_n - X_0)(f_n - Y_0) - \frac{1}{2}a^2 &= 0,\\ f_n - Y_0 + \ (x_n - X_0)f_n' &= 0,\\ 2f_n' + \ (x_n - X_0)f_n'' &= 0.\end{aligned}$$

The unknowns are X_0, Y_0 and a; the first is obtained immediately from the last equation,

$$X_0 = x_n + 2\,\frac{f_n'}{f_n''}.$$

The second equation then gives

$$Y_0 = f_n - 2\,\frac{f_n'^2}{f_n''}$$

and the first

$$\frac{1}{2}\,a^2 = -4\,\frac{f_n'^3}{f_n''^2}.$$

Therefore,

$$\begin{aligned}x_{n+1} = X_0 - \frac{\frac{1}{2}a^2}{Y_0} &= x_n + 2\,\frac{f_n'}{f_n''} + \frac{4\,f_n'^3}{f_n''^2\left(f_n - 2\,\frac{f_n'^2}{f_n''}\right)}\\ &= x_n + 2\,\frac{f_n'}{f_n''} - \frac{2\,\frac{f_n'^2}{f_n''}}{f_n' - \frac{1}{2}f_n''\frac{f_n}{f_n'}}\\ &= x_n - \frac{-2\,\frac{f_n'}{f_n''}\left(f_n' - \frac{1}{2}f_n''\frac{f_n}{f_n'}\right) + 2\,\frac{f_n'^2}{f_n''}}{f_n' - \frac{1}{2}f_n''\frac{f_n}{f_n'}}\\ &= x_n - \frac{f_n}{f_n' - \frac{1}{2}f_n''\frac{f_n}{f_n'}},\end{aligned}$$

as claimed.

(b) Newton's method applied to $g = 0$ is (in obvious notations)

$$x_{n+1} = x_n - \frac{g_n}{g_n'} = x_n - \frac{f_n}{\sqrt{f_n'}} \, \frac{f_n'}{\sqrt{f_n'}\,f_n' - \frac{1}{2} f_n \frac{1}{\sqrt{f_n'}} f_n''}$$

$$= x_n - \frac{f_n f_n'}{f_n'^2 - \frac{1}{2} f_n'' f_n}$$

$$= x_n - \frac{f_n}{f_n' - \frac{1}{2} f_n'' \frac{f_n}{f_n'}}.$$

(c) We have

$$x_{n+1} = \varphi(x_n), \quad \text{where } \varphi(x) = x - \frac{f(x)}{f'(x) - \frac{1}{2} f''(x) \frac{f(x)}{f'(x)}}.$$

Thus (dropping the argument x throughout),

$$(f'^2 - \frac{1}{2} f'' f)\varphi = x(f'^2 - \frac{1}{2} f'' f) - f f'.$$

Differentiating with respect to x gives

$$\left(\frac{3}{2} f' f'' - \frac{1}{2} f''' f\right)\varphi + \left(f'^2 - \frac{1}{2} f'' f\right)\varphi'$$

$$= f'^2 - \frac{1}{2} f'' f + x\left(\frac{3}{2} f' f'' - \frac{1}{2} f''' f\right) - f'^2 - f f''$$

$$= -\frac{3}{2} f'' f + x\left(\frac{3}{2} f' f'' - \frac{1}{2} f''' f\right).$$

Since $\varphi(\alpha) = \alpha$, putting $x = \alpha$, one sees that the first term on the left and the last term on the far right cancel each other. What remains simplifies, in view of $f(\alpha) = 0$, to

$$f'^2(\alpha)\varphi'(\alpha) = 0,$$

hence, since $f'(\alpha) \neq 0$, to $\varphi'(\alpha) = 0$. Differentiating again, we get

$$\left(\frac{3}{2} f''^2 + f''' f' - \frac{1}{2} f^{(4)} f\right)\varphi + (3 f' f'' - f''' f)\varphi' + \left(f'^2 - \frac{1}{2} f'' f\right)\varphi''$$

$$= -2 f''' f + x\left(\frac{3}{2} f''^2 + f' f''' - \frac{1}{2} f^{(4)} f\right).$$

Again putting $x = \alpha$ and using $\varphi(\alpha) = \alpha$, $\varphi'(\alpha) = 0$, $f(\alpha) = 0$, we obtain

$$\begin{aligned}&\left(\frac{3}{2}[f''(\alpha)]^2 + f'''(\alpha)f'(\alpha)\right)\alpha + [f'(\alpha)]^2\varphi''(\alpha)\\ &= \left(\frac{3}{2}[f''(\alpha)]^2 + f'(\alpha)f'''(\alpha)\right)\alpha,\end{aligned}$$

that is, $[f'(\alpha)]^2\varphi''(\alpha) = 0$, hence $\varphi''(\alpha) = 0$. Thus, the order of convergence is at least $p = 3$. To obtain $\varphi'''(\alpha)$, we must differentiate once more and get, when $x = \alpha$,

$$\begin{aligned}&\left(4f''f''' + \frac{1}{2}f^{(4)}f'\right)_{x=\alpha}\cdot\alpha + [f'(\alpha)]^2\varphi'''(\alpha)\\ &= \left(-f'''f' + \frac{3}{2}f''^2\right)_{x=\alpha} + \left(4f''f''' + \frac{1}{2}f^{(4)}f'\right)_{x=\alpha}\cdot\alpha,\end{aligned}$$

hence

$$\varphi'''(\alpha) = -\left[\frac{f'''}{f'} - \frac{3}{2}\left(\frac{f''}{f'}\right)^2\right]_{x=\alpha} = -(Sf)(\alpha).$$

Thus, if $(Sf)(\alpha) \neq 0$, the order of convergence is exactly equal to 3.

(d) Yes, slightly. Recall from Sect. 4.2 that the efficiency index is $p^{1/m}$, where p is the order of the method and m the number of "units of work." For Newton's method this is $\sqrt{2} = 1.4142\ldots$, whereas for Halley's method it is $3^{\frac{1}{3}} = 1.4422\ldots$.

(e) In the case $f(x) = x^\lambda - a$, we get

$$\begin{aligned}x_{n+1} &= x_n - \frac{x_n^\lambda - a}{\lambda x_n^{\lambda-1} - \frac{1}{2}\lambda(\lambda-1)x_n^{\lambda-2}\,\frac{x_n^\lambda - a}{\lambda x_n^{\lambda-1}}}\\ &= \frac{\lambda x_n^\lambda - \frac{1}{2}(\lambda-1)x_n^\lambda + \frac{1}{2}(\lambda-1)a - x_n^\lambda + a}{\lambda x_n^\lambda - \frac{1}{2}(\lambda-1)(x_n^\lambda - a)}\,x_n\\ &= \frac{(\lambda-1)x_n^\lambda + (\lambda+1)a}{(\lambda+1)x_n^\lambda + (\lambda-1)a}\,x_n.\end{aligned}$$

45. (a) By Newton's interpolation formula with remainder term, interpolating the nth-degree polynomial p by a polynomial of degree $n-1$, we can write

$$p(t) = \sum_{\nu=1}^{n} f_\nu(\boldsymbol{x})\prod_{\mu=1}^{\nu-1}(t - x_\mu) + \frac{p^{(n)}(\tau)}{n!}\prod_{\mu=1}^{n}(t - x_\mu).$$

Since p is monic of degree n, we have $p^{(n)}(\tau) = n!$, and so

$$p(t) = \sum_{\nu=1}^{n} f_\nu(\boldsymbol{x}) \prod_{\mu=1}^{\nu-1}(t - x_\mu) + \prod_{\mu=1}^{n}(t - x_\mu).$$

Suppose $\boldsymbol{\alpha}^{\mathrm{T}} = [\alpha_1, \alpha_2, \ldots, \alpha_n]$ is a solution of $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$. Then $f_\nu(\boldsymbol{\alpha}) = 0$ for $\nu = 1, 2, \ldots, n$, hence

$$p(t) = \prod_{\mu=1}^{n}(t - \alpha_\mu),$$

showing that α_μ are the zeros of p, which are unique up to permutation. Conversely, let $\boldsymbol{\alpha}^{\mathrm{T}} = [\alpha_1, \alpha_2, \ldots, \alpha_n]$ be the zeros of p in some order. Then, identically in t,

$$p(t) = \sum_{\nu=1}^{n} f_\nu(\boldsymbol{\alpha}) \prod_{\mu=1}^{\nu-1}(t - \alpha_\mu) + \prod_{\mu=1}^{n}(t - \alpha_\mu),$$

that is, since $\prod_{\mu=1}^{n}(t - \alpha_\mu) = p(t)$,

$$0 = \sum_{\nu=1}^{n} f_\nu(\boldsymbol{\alpha}) \prod_{\mu=1}^{\nu-1}(t - \alpha_\mu).$$

Letting $t \to \alpha_1$ gives $f_1(\boldsymbol{\alpha}) = 0$. Dividing the remaining equation by $t - \alpha_1$ and letting $t \to \alpha_2$ gives $f_2(\boldsymbol{\alpha}) = 0$. Continuing in this manner yields $f_3(\boldsymbol{\alpha}) = 0, \ldots, f_n(\boldsymbol{\alpha}) = 0$ in this order.

(b) The Jacobian of $\boldsymbol{f}$ is a lower triangular matrix, namely

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(\boldsymbol{x}) = \begin{bmatrix} [x_1, x_1]p & 0 & \cdots & 0 \\ [x_1, x_1, x_2]p & [x_1, x_2, x_2]p & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ [x_1, x_1, x_2, \ldots, x_n]p & [x_1, x_2, x_2, \ldots, x_n]p & \cdots & [x_1, x_2, \ldots, x_n, x_n]p \end{bmatrix}$$

(cf. Ch. 2, Ex. 58). Given an approximation $\boldsymbol{\alpha}^{[i]}$ to the root vector $\boldsymbol{\alpha}$, Newton's method requires the solution by forward substitution of the lower triangular system

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(\boldsymbol{\alpha}^{[i]})\boldsymbol{\Delta}^{[i]} = -\boldsymbol{f}(\boldsymbol{\alpha}^{[i]})$$

to get the next approximation vector

$$\boldsymbol{\alpha}^{[i+1]} = \boldsymbol{\alpha}^{[i]} + \boldsymbol{\Delta}^{[i]}.$$

(c) The arguments in (a), (b) extend to functions p that are sufficiently smooth, for example, $p \in C^n[\mathbb{R}]$. Then, if $f_\nu(\boldsymbol{\alpha}) = 0$ for some $\boldsymbol{\alpha}^{\mathrm{T}} = [\alpha_1, \alpha_2, \ldots, \alpha_n]$, the $\alpha_1, \alpha_2, \ldots, \alpha_n$ are zeros of p and vice versa. The first statement is an immediate consequence of the first identity above in (a). The converse follows similarly as before by taking limits $t \to \alpha_\mu$ followed by division of both sides of the identity by $t - \alpha_\mu$.

Selected Solutions to Machine Assignments

6.(a) The discretization of the boundary value problem amounts to solving the system of nonlinear equations

$$\begin{aligned} u_{k+1} - 2u_k + u_{k-1} &= h^2 g\left(x_k, u_k, \frac{u_{k+1} - u_{k-1}}{2h}\right), \\ k &= 1, 2, \ldots, n, \\ u_0 &= y_0, \quad u_{n+1} = y_1, \end{aligned}$$

where $h = 1/(n+1)$. With

$$\boldsymbol{A} = \begin{bmatrix} -2 & 1 & & & \boldsymbol{0} \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ \boldsymbol{0} & & & 1 & -2 \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix},$$

$$\boldsymbol{F}(\boldsymbol{u}) = \begin{bmatrix} g\left(x_1, u_1, \frac{u_2 - y_0}{2h}\right) \\ g\left(x_2, u_2, \frac{u_3 - u_1}{2h}\right) \\ \vdots \\ g\left(x_n, u_n, \frac{y_1 - u_{n-1}}{2h}\right) \end{bmatrix},$$

this can be written as a fixed point problem

$$\boldsymbol{A}\boldsymbol{u} = h^2 \boldsymbol{F}(\boldsymbol{u}) - \boldsymbol{b} \quad \text{or} \quad \boldsymbol{u} = \boldsymbol{A}^{-1}(h^2 \boldsymbol{F}(\boldsymbol{u}) - \boldsymbol{b}),$$

where

$$\boldsymbol{b} = y_0 \boldsymbol{e}_1 + y_1 \boldsymbol{e}_n, \;\; \boldsymbol{e}_1 = [1, 0, \ldots, 0]^{\mathrm{T}} \in \mathbb{R}^n, \;\; \boldsymbol{e}_n = [0, 0, \ldots, 1]^{\mathrm{T}} \in \mathbb{R}^n.$$

The corresponding fixed point iteration is

$$\boldsymbol{A}\boldsymbol{u}^{[i+1]} = h^2\boldsymbol{F}(u^{[i]}) - \boldsymbol{b}, \quad i = 0, 1, 2, \ldots;$$
$$\boldsymbol{u}^{[0]} = \text{initial approximation.}$$

(b) PROGRAM

```
%MAIV_6B
%
f0='%8.0f %4.0f %12.4e %12.4e\n';
disp('     n     it        err          order')
eps0=.5e-14;
for n=[10 100 1000 10000]
  h=1/(n+1); h2=h^2;
  a=-2*ones(n,1); b=ones(n-1,1); c=b;
  en=eye(n); en=en(:,n);
  x=linspace(h,1-h,n)'; y=sinh(x)/sinh(1);
  it=0; u0=zeros(n,1); u1=ones(n,1);
  while max(abs(u1-u0))>eps0
    it=it+1;
    u0=u1;
    u1=tridiag(n,a,b,c,h2*u0-en);
  end
  err=max(abs(u1-y)); ord=err/h2;
  fprintf(f0,n,k,err,ord)
end
```

OUTPUT

```
>> MAIV_6B
     n     it        err          order
      10   16   3.6185e-05   4.3784e-03
     100   16   4.3345e-07   4.4217e-03
    1000   16   4.4131e-09   4.4219e-03
   10000   16   5.0103e-11   5.0113e-03
>>
```

Since the central difference approximations of the derivatives have errors of $O(h^2)$, the same can be expected for the errors in the solution. This is confirmed in the last column of the OUTPUT, suggesting also that the constant involved is about 5×10^{-3}.

(c) The system of nonlinear equations is

$$\boldsymbol{A}\boldsymbol{u} = -h^2 \sin \boldsymbol{u} - \boldsymbol{e}_n,\ h = \frac{\pi}{4(n+1)},$$

where $\sin \boldsymbol{u} = [\sin u_1, \sin u_2, \ldots, \sin u_n]^{\mathrm{T}}$. The iteration function is

$$\boldsymbol{\varphi}(\boldsymbol{u}) = -\boldsymbol{A}^{-1}(h^2 \sin \boldsymbol{u} + \boldsymbol{e}_n).$$

We have, using the Mean Value Theorem applied to the sine function,

$$\boldsymbol{\varphi}(\boldsymbol{u}) - \boldsymbol{\varphi}(\boldsymbol{u}^*) = -h^2\boldsymbol{A}^{-1}(\sin \boldsymbol{u} - \sin \boldsymbol{u}^*) = -h^2\boldsymbol{A}^{-1}\mathrm{diag}(\cos(\boldsymbol{\eta}))(\boldsymbol{u} - \boldsymbol{u}^*),$$

where the elements of $\boldsymbol{\eta}$ are on the line segment between $\boldsymbol{u}$ and $\boldsymbol{u}^*$. Therefore,

$$\|\boldsymbol{\varphi}(\boldsymbol{u}) - \boldsymbol{\varphi}(\boldsymbol{u}^*)\|_\infty \le h^2\|\boldsymbol{A}^{-1}\|_\infty \|\boldsymbol{u} - \boldsymbol{u}^*\|_\infty.$$

Here we have, using the *Hint*,

$$h^2\|\boldsymbol{A}^{-1}\|_\infty \le h^2 \cdot \frac{1}{8}(n+1)^2 = \frac{\pi^2}{128} = .077106\ldots < 1,$$

so that $\boldsymbol{\varphi}$ is strongly contractive and we have rapid convergence of the fixed point iteration (cf. Theorem 4.9.1).

8. (a) Let

$$f(\alpha) = \int_0^{3\pi/2} \frac{\cos t}{t^\alpha}\,\mathrm{d}t, \quad 0 < \alpha < 1.$$

Clearly, $f(0) = \sin(3\pi/2) = -1$ and $f(1) = \infty$. A graph of the function reveals that f is monotonically increasing and convex, and thus has a unique zero in $(0, 1)$; it is located near 0.3. Any initial approximation to the right of this zero will guarantee convergence of Newton's method, owing to convexity. We choose 0.4 as initial approximation.

To transform to a standard interval, we make the change of variables $t = \frac{3\pi}{2}x$ and have

$$f(\alpha) = \left(\frac{3\pi}{2}\right)^{1-\alpha} \int_0^1 \cos\left(\frac{3\pi}{2}x\right) x^{-\alpha}\mathrm{d}x.$$

Differentiating, we get

$$f'(\alpha) = \left(\frac{3\pi}{2}\right)^{1-\alpha} \left\{ \int_0^1 \cos\left(\frac{3\pi}{2}x\right) x^{-\alpha} \ln(1/x)\mathrm{d}x \right.$$
$$\left. - \ln\frac{3\pi}{2} \int_0^1 \cos\left(\frac{3\pi}{2}x\right) x^{-\alpha}\mathrm{d}x \right\},$$

and Newton's method becomes

$$\alpha^{[i+1]} = \alpha^{[i]} - \frac{f(\alpha^{[i]})}{f'(\alpha^{[i]})}, \quad i = 0, 1, 2, \ldots; \ \alpha^{[0]} = .4.$$

The integral defining f, and the second integral in the formula for f', call for Gauss–Jacobi quadrature with Jacobi parameters 0 and $-\alpha$. The first integral in f' requires Gauss quadrature relative to the weight function $x^{-\alpha}\ln(1/x)$ on $[0, 1]$. The `OPQ` routines providing the recurrence coefficients for the respective orthogonal polynomials are `r_jacobi01.m` and `r_jaclog.m`, respectively, while the routine `gauss.m` generates the Gaussian quadrature formulae from the recurrence coefficients. The program below also determines, and prints, the respective numbers n and n_0 of quadrature points required for 14 decimal digit accuracy:

PROGRAM

```
%MAIV_8A
%
nmax=50; eps0=.5e-14;
a1=.4; a=0;
while abs(a1-a)> eps0
  a=a1;
  ab=r_jacobi01(nmax,0,-a);
  ab0=r_jaclog(nmax,-a);
  n=1; sg1=0; sg=1;
  while abs(sg1-sg)>eps0
    n=n+1; sg=sg1;
    xw=gauss(n,ab);
    sg1=sum(xw(:,2).*cos(3*pi*xw(:,1)/2));
  end
  f=(3*pi/2)^(1-a)*sg1;
  n0=1; sg01=0; sg0=1;
  while abs(sg01-sg0)>eps0
    n0=n0+1;
    sg0=sg01;
    xw0=gauss(n0,ab0);
    sg01=sum(xw0(:,2).*cos(3*pi*xw0(:,1)/2));
  end
  fd=(3*pi/2)^(1-a)*(sg01-log(3*pi/2)*sg1);
  a1=a-f/fd;
end
fprintf('    n=%2.0f,  n0=%2.0f,
   alpha0=%17.15f\n',n,n0,a1)
```

OUTPUT

```
>> MAIV_8A
    n=10,  n0=10,   alpha0=0.308443779561986
>>
```

A high-precision value of α_0 (to $1,120$ digits) can be found in Arias de Reyna and Van de Lune [2009, Sect. 5].

(b) Letting

$$g(\alpha) = \int_0^{5\pi/4} \frac{\cos(t + \pi/4)}{t^\alpha}\,\mathrm{d}t, \quad 0 < \alpha < 1,$$

a similar calculation as in (a) yields

$$\begin{aligned} g(\alpha) &= \left(\frac{5\pi}{4}\right)^{1-\alpha} \int_0^1 \frac{\cos((5x+1)\pi/4)}{x^\alpha}\,\mathrm{d}x, \\ g'(\alpha) &= \left(\frac{5\pi}{4}\right)^{1-\alpha} \left\{ \int_0^1 \cos((5x+1)\pi/4) x^{-\alpha} \ln(1/x) \mathrm{d}x \right. \\ &\qquad\qquad \left. - \ln\frac{5\pi}{4} \int_0^1 \frac{\cos((5x+1)\pi/4)}{x^\alpha}\,\mathrm{d}x \right\}. \end{aligned}$$

PROGRAM

```
%MAIV_8B
%
nmax=50; eps0=.5e-14;
a1=.7; a=0;
while abs(a1-a)> eps0
  a=a1;
  ab=r_jacobi01(nmax,0,-a);
  ab0=r_jaclog(nmax,-a);
  n=1; sg1=0; sg=1;
  while abs(sg1-sg)>eps0
    n=n+1; sg=sg1;
    xw=gauss(n,ab);
    sg1=sum(xw(:,2).*cos(pi*(5*xw(:,1)+1)/4));
  end
  g=(5*pi/4)^(1-a)*sg1;
  n0=1; sg01=0; sg0=1;
  while abs(sg01-sg0)>eps0
    n0=n0+1;
    sg0=sg01;
    xw0=gauss(n0,ab0);
    sg01=sum(xw0(:,2).*cos(pi*(5*xw0
      (:,1)+1)/4));
  end
  gd=(5*pi/4)^(1-a)*(sg01-log(5*pi/4)*sg1);
  a1=a-g/gd;
end
```

```
fprintf('    n=%2.0f,  n0=%2.0f,
   alpha1=%17.15f\n',n,n0,a1)
```

OUTPUT

```
>> MAIV_8B
    n=10,  n0=10,   alpha1=0.614433447526100
>>
```

Chapter 5
Initial Value Problems for ODEs: One-Step Methods

Initial value problems for ordinary differential equations (ODEs) occur in almost all the sciences, notably in mechanics (including celestial mechanics), where the motion of particles (resp., planets) is governed by Newton's second law – a system of second-order differential equations. It is no wonder, therefore, that astronomers such as Adams, Moulton, and Cowell were instrumental in developing numerical techniques for their integration.[1] Also in quantum mechanics – the analogue of celestial mechanics in the realm of atomic dimensions – differential equations are fundamental; this time it is Schrödinger's equation that reigns, actually a linear *partial* differential equation involving the Laplace operator. Still, when separated in polar coordinates, it reduces to an ordinary second-order linear differential equation. Such equations are at the heart of the theory of special functions of mathematical physics. Coulomb wave functions, for example, are solutions of Schrödinger's equation when the potential is a Coulomb field.

Within mathematics, ordinary differential equations play an important role in the calculus of variations, where optimal trajectories must satisfy the Euler equations, or in optimal control problems, where they satisfy the Pontryagin maximum principle. In both cases, one is led to boundary value problems for ordinary differential equations. This type of problem is discussed in Chap. 7. In the present and next chapter, we concentrate on initial value problems.

We begin with some examples of ordinary differential equations as they arise in the context of numerical analysis.

[1] In fact, it was by means of computational methods that Le Verrier in 1846 predicted the existence of the eighth planet Neptune, based on observed (and unaccounted for) irregularities in the orbit of the next inner planet. Soon thereafter, the planet was indeed discovered at precisely the predicted location. (Some calculations were done previously by Adams, then an undergraduate at Cambridge, but were not published in time.)

W. Gautschi, *Numerical Analysis*, DOI 10.1007/978-0-8176-8259-0_5,

5.1 Examples

Our first example is rather trivial: suppose we want to compute the integral

$$I = \int_a^b f(t)\mathrm{d}t \tag{5.1}$$

for some given function f on $[a, b]$. Letting $y(x) = \int_a^x f(t)\mathrm{d}t$, we get immediately

$$\frac{\mathrm{d}y}{\mathrm{d}x} = f(x), \quad y(a) = 0. \tag{5.2}$$

This is an initial value problem for a first-order differential equation, but a very special one in which y does not appear on the right-hand side. By solving (5.2) on $[a, b]$, one obtains $I = y(b)$. This example, as elementary as it seems to be, is not entirely without interest, because when we integrate (5.2) by modern techniques of solving ODEs, we can take advantage of step control mechanisms, taking smaller steps where f changes more rapidly, and larger ones elsewhere. This gives rise to what may be called *adaptive integration*.

A more interesting extension of this idea is exemplified by the following integral,

$$I = \int_0^\infty J_0(t^2)\mathrm{e}^{-t}\mathrm{d}t, \tag{5.3}$$

containing the Bessel function $J_0(x^2)$, one of the special functions of mathematical physics. It satisfies the linear second-order differential equation

$$\frac{\mathrm{d}^2y}{\mathrm{d}x^2} + \left(2 - \frac{1}{x}\right)\frac{\mathrm{d}y}{\mathrm{d}x} + 4x^2y = 0, \quad x > 0, \tag{5.4}$$

with initial conditions

$$y(0) = 1, \quad y'(0) = 0. \tag{5.5}$$

As is often the case with special functions arising in physics, the associated differential equation has a singularity at the origin $x = 0$, even though the solution $y(x) = J_0(x^2)$ is perfectly regular at $x = 0$. (The singular term in (5.4) has limit 0 at $x = 0$.) Here, we let

$$y_1(x) = \int_0^x J_0(t^2)\mathrm{e}^{-t}\mathrm{d}t, \quad y_2(x) = J_0(x^2), \quad y_3(x) = y_2'(x) \tag{5.6}$$

and obtain

$$\begin{aligned}
\frac{\mathrm{d}y_1}{\mathrm{d}x} &= \mathrm{e}^{-x}y_2, && y_1(0) = 0,\\
\frac{\mathrm{d}y_2}{\mathrm{d}x} &= y_3, && y_2(0) = 1,\\
\frac{\mathrm{d}y_3}{\mathrm{d}x} &= -\left(2-\frac{1}{x}\right)y_3 - 4x^2y_2, && y_3(0) = 0,
\end{aligned} \tag{5.7}$$

an initial value problem for a system of three (linear) first-order differential equations. We need to integrate this on $(0,\infty)$ to find $I = y_1(\infty)$. An advantage of this approach is that the Bessel function need not be calculated explicitly; it is calculated implicitly (as component y_2) through the integration of the differential equation that it satisfies. Another advantage over straightforward numerical integration is again the possibility of automatic error control through appropriate step change techniques. We could equally well get rid of the exponential e^{-x} in (5.7) by calling it y_4 and adding the differential equation $\mathrm{d}y_4/\mathrm{d}x = -y_4$ and initial condition $y_4(0) = 1$. (The difficulty with the singularity at $x = 0$ can be handled, e.g., by introducing initially, say, for $0 \le x \le \frac{1}{3}$, a new dependent variable $\tilde{y}_3$, setting $\tilde{y}_3 = (2-\frac{1}{x})y_2' = (2-\frac{1}{x})y_3$. Then $\tilde{y}_3(0) = 0$, and the second and third equations in (5.7) can be replaced by

$$\begin{aligned}
\frac{\mathrm{d}y_2}{\mathrm{d}x} &= -\frac{x}{1-2x}\tilde{y}_3,\\
\frac{\mathrm{d}\tilde{y}_3}{\mathrm{d}x} &= -\frac{4(1-x)}{1-2x}\tilde{y}_3 + 4x(1-2x)y_2.
\end{aligned}$$

Here, the coefficients are now well-behaved functions near $x = 0$. Once $x = \frac{1}{3}$ has been reached, one can switch back to (5.7), using for y_3 the initial condition $y_3(\frac{1}{3}) = -\tilde{y}_3(\frac{1}{3})$.)

Another example is the *method of lines* in partial differential equations, where one discretizes partial derivatives with respect to all variables but one, thereby obtaining a system of ordinary differential equations. This may be illustrated by the heat equation on a rectangular domain,

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad 0 \le x \le 1, \quad 0 \le t \le T, \tag{5.8}$$

where the temperature $u = u(x,t)$ is to satisfy the initial condition

$$u(x,0) = \varphi(x), \quad 0 \le x \le 1, \tag{5.9}$$

and the boundary conditions

$$u(0,t) = \lambda(t), \quad u(1,t) = \rho(t), \quad 0 \le t \le T. \tag{5.10}$$

Here, φ is a given function of x, and λ, ρ given functions of t. We discretize in the x-variable by placing a grid $\{x_n\}_{n=0}^{N+1}$ with $x_n = nh$, $h = \frac{1}{N+1}$, on the interval $0 \le x \le 1$ and approximating the second derivative by the second divided difference (cf. Chap. 2, (2.117) for $n = 2$, putting $\xi \approx x_1$),

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{x=x_n} \approx \frac{u_{n+1} - 2u_n + u_{n-1}}{h^2}, \quad n = 1, 2, \ldots, N, \tag{5.11}$$

where

$$u_n = u_n(t) := u(x_n, t), \quad n = 0, 1, \ldots, N+1. \tag{5.12}$$

Writing down (5.8) for $x = x_n$, $n = 1, 2, \ldots, N$, and using (5.11), we get approximately

$$\begin{aligned} \frac{\mathrm{d}u_n}{\mathrm{d}t} &= \frac{1}{h^2}(u_{n+1} - 2u_n + u_{n-1}), \\ u_n(0) &= \varphi(x_n), \end{aligned} \qquad n = 1, 2, \ldots, N, \tag{5.13}$$

an initial value problem for a system of N differential equations in the N unknown functions $u_1, u_2, \ldots, u_N$. The boundary functions $\lambda(t)$, $\rho(t)$ enter into (5.13) when reference is made to u_0 or u_{N+1} on the right-hand side of the differential equation. By making the grid finer and finer, hence h smaller, one expects to obtain better and better approximations for $u(x_n, t)$. Unfortunately, this comes at a price: the system (5.13) becomes more and more "stiff" as h decreases, calling for special methods designed especially for stiff equations (cf. Sect. 5.9; Chap. 6, Sect. 6.5).

5.2 Types of Differential Equations

The standard initial value problem involves a system of first-order differential equations

$$\frac{\mathrm{d}y^i}{\mathrm{d}x} = f^i(x, y^1, y^2, \ldots, y^d), \quad i = 1, 2, \ldots, d, \tag{5.14}$$

which is to be solved on an interval $[a, b]$, given the initial values

$$y^i(a) = y_0^i, \quad i = 1, 2, \ldots, d. \tag{5.15}$$

Here, the component functions are indexed by superscripts, and subscripts are reserved to indicate step numbers, the initial step having index 0. We use vector notation throughout by letting

$$\boldsymbol{y}^{\mathrm{T}} = [y^1, y^2, \ldots, y^d], \quad \boldsymbol{f}^{\mathrm{T}} = [f^1, f^2, \ldots, f^d], \quad \boldsymbol{y}_0^{\mathrm{T}} = [y_0^1, y_0^2, \ldots, y_0^d]$$

and writing (5.14), (5.15) in the form

$$\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}x} = \boldsymbol{f}(x, \boldsymbol{y}), \ \ a \le x \le b; \quad \boldsymbol{y}(a) = \boldsymbol{y}_0. \tag{5.16}$$

We are thus seeking a vector-valued function $\boldsymbol{y}(x) \in C^1[a,b]$ that satisfies (5.16) identically on $[a,b]$ and has the starting value $\boldsymbol{y}_0$ at $x = a$. About $\boldsymbol{f}(x,\boldsymbol{y})$, we assume that it is defined for $x \in [a,b]$ and all $\boldsymbol{y} \in \mathbb{R}^d$.[2]

We note some important special cases of (5.16).

1. $d = 1$, $y' = f(x,y)$: a single first-order differential equation.
2. $d > 1$, $u^{(d)} = g(x,u,u',\dots,u^{(d-1)})$: a single dth-order differential equation. The initial conditions here take the form $u^{(i)}(a) = u_0^i$, $i = 0,1,\dots,d-1$. This problem is easily brought into the form (5.16) by defining

$$y^i = u^{(i-1)}, \quad i = 1,2,\dots,d.$$

Then

$$\begin{aligned}
&\frac{\mathrm{d}y^1}{\mathrm{d}x} = y^2, && y^1(a) = u_0^0,\\
&\frac{\mathrm{d}y^2}{\mathrm{d}x} = y^3, && y^2(a) = u_0^1,\\
&\dots\dots\dots\dots\dots\dots\dots && \dots\dots\dots\dots\\
&\frac{\mathrm{d}y^{d-1}}{\mathrm{d}x} = y^d, && y^{d-1}(a) = u_0^{d-2},\\
&\frac{\mathrm{d}y^d}{\mathrm{d}x} = g(x,y^1,y^2,\dots,y^d), && y^d(a) = u_0^{d-1},
\end{aligned} \tag{5.17}$$

which has the form (5.16) with very special (linear) functions $f^1, f^2, \dots, f^{d-1}$, and $f^d(x,\boldsymbol{y}) = g(x,\boldsymbol{y})$.

Although this is the canonical way of transforming a single equation of order d into a system of first-order equations, there are other ways of doing this, which are sometimes more natural. Consider, for example, the *Sturm–Liouville equation*

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(p(x)\frac{\mathrm{d}u}{\mathrm{d}x}\right) + q(x)u = 0, \quad a \le x \le b, \tag{5.18}$$

[2] That is, each component function $f^i(x,\boldsymbol{y})$ is defined on $[a,b] \times \mathbb{R}^d$. In some problems, $\boldsymbol{f}(x,\boldsymbol{y})$ is defined only on $[a,b] \times \mathcal{D}$, where $\mathcal{D} \subset \mathbb{R}^d$ is a compact domain. In such cases, the solution $\boldsymbol{y}(x)$ must be required to remain in $\mathcal{D}$ as x varies in $[a,b]$. This causes complications, which we avoid by assuming $\mathcal{D} = \mathbb{R}^d$.

where $p(x) \neq 0$ on $[a, b]$. Here, the substitution

$$y^1 = u, \quad y^2 = p(x)\frac{\mathrm{d}u}{\mathrm{d}x} \tag{5.19}$$

is more appropriate (and also physically more meaningful), leading to the system

$$\begin{aligned} \frac{\mathrm{d}y^1}{\mathrm{d}x} &= \frac{1}{p(x)}y^2, \\ \frac{\mathrm{d}y^2}{\mathrm{d}x} &= -q(x)y^1. \end{aligned} \tag{5.20}$$

Mathematically, (5.20) has the advantage over (5.18) of not requiring that p be differentiable on $[a, b]$.

3. $\mathrm{d}\boldsymbol{y}/\mathrm{d}x = \boldsymbol{f}(\boldsymbol{y})$, $\boldsymbol{y} \in \mathbb{R}^d$: an *autonomous* system of differential equations. Here, $\boldsymbol{f}$ does not depend explicitly on x. This can always be trivially achieved by introducing, if necessary, $y^0(x) = x$ and writing (5.16) as

$$\frac{\mathrm{d}}{\mathrm{d}x}\begin{bmatrix} \boldsymbol{y} \\ y^0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}(y^0, \boldsymbol{y}) \\ 1 \end{bmatrix}, \quad a \leq x \leq b; \begin{bmatrix} \boldsymbol{y} \\ y^0 \end{bmatrix}(a) = \begin{bmatrix} \boldsymbol{y}_0 \\ a \end{bmatrix}. \tag{5.21}$$

   Many ODE software packages indeed assume that the system is autonomous.
4. Second-order system

$$\frac{\mathrm{d}^2u^i}{\mathrm{d}x^2} = g^i\left(x, u^1, \ldots, u^d, \frac{\mathrm{d}u^1}{\mathrm{d}x}, \ldots, \frac{\mathrm{d}u^d}{\mathrm{d}x}\right), \quad i = 1, 2, \ldots, d. \tag{5.22}$$

   Newton's law in mechanics is of this form, with $d = 3$. The canonical transformation here introduces

$$y^1 = u^1, \ldots, y^d = u^d; y^{d+1} = \frac{\mathrm{d}u^1}{\mathrm{d}x}, \ldots, y^{2d} = \frac{\mathrm{d}u^d}{\mathrm{d}x}$$

   and yields a system of $2d$ first-order equations,

$$\begin{aligned} &\frac{\mathrm{d}y^1}{\mathrm{d}x} = y^{d+1}, \\ &\cdots\cdots\cdots\cdots \\ &\frac{\mathrm{d}y^d}{\mathrm{d}x} = y^{2d}, \\ &\frac{\mathrm{d}y^{d+1}}{\mathrm{d}x} = g^1(x, y^1, \ldots, y^d, y^{d+1}, \ldots, y^{2d}), \\ &\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ &\frac{\mathrm{d}y^{2d}}{\mathrm{d}x} = g^d(x, y^1, \ldots, y^d, y^{d+1}, \ldots, y^{2d}). \end{aligned} \tag{5.23}$$

5. *Implicit system* of first-order differential equations:

$$F^i\left(x, \boldsymbol{u}, \frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}x}\right) = 0, \quad i = 1, 2, \ldots, d; \boldsymbol{u} \in \mathbb{R}^d. \tag{5.24}$$

Here, $F^i = F^i(x, \boldsymbol{u}, \boldsymbol{v})$ are given functions of $2d + 1$ variables, which again we combine into a vector $\boldsymbol{F} = [F^i]$. We denote by $\boldsymbol{F}_x$ the partial derivative of $\boldsymbol{F}$ with respect to x, and by $\boldsymbol{F}_{\boldsymbol{u}}$, $\boldsymbol{F}_{\boldsymbol{v}}$ the Jacobian matrices

$$\boldsymbol{F}_{\boldsymbol{u}}(x, \boldsymbol{u}, \boldsymbol{v}) = \left[\frac{\partial F^i}{\partial u^j}\right], \quad \boldsymbol{F}_{\boldsymbol{v}}(x, \boldsymbol{u}, \boldsymbol{v}) = \left[\frac{\partial F^i}{\partial v^j}\right].$$

Assuming that these Jacobians exist and that $\boldsymbol{F}_{\boldsymbol{v}}$ is nonsingular on $[a, b] \times \mathbb{R}^d \times \mathbb{R}^d$, the implicit system (5.24) may be dealt with by differentiating it totally with respect to x. This yields (in vector notation)

$$\boldsymbol{F}_x + \boldsymbol{F}_{\boldsymbol{u}} \frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}x} + \boldsymbol{F}_{\boldsymbol{v}} \frac{\mathrm{d}^2\boldsymbol{u}}{\mathrm{d}x^2} = \boldsymbol{0}, \tag{5.25}$$

where the arguments in $\boldsymbol{F}_x$, $\boldsymbol{F}_{\boldsymbol{u}}$, $\boldsymbol{F}_{\boldsymbol{v}}$ are x, $\boldsymbol{u}$, $\boldsymbol{u}' = \mathrm{d}\boldsymbol{u}/\mathrm{d}x$. By assumption, this can be solved for the second derivative,

$$\frac{\mathrm{d}^2\boldsymbol{u}}{\mathrm{d}x^2} = \boldsymbol{F}_{\boldsymbol{v}}^{-1}(x, \boldsymbol{u}, \boldsymbol{u}')\left\{-\boldsymbol{F}_x(x, \boldsymbol{u}, \boldsymbol{u}') - \boldsymbol{F}_{\boldsymbol{u}}(x, \boldsymbol{u}, \boldsymbol{u}')\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}x}\right\}, \tag{5.26}$$

yielding an (explicit) system of second-order differential equations (cf. (iv)). If we are to solve the initial value problem for (5.24) on $[a, b]$, with $\boldsymbol{u}(a) = \boldsymbol{u}_0$ prescribed, we need for (5.26) the additional initial data $(\mathrm{d}\boldsymbol{u}/\mathrm{d}x)(a) = \boldsymbol{u}_0'$. This must be obtained by solving $\boldsymbol{F}(a, \boldsymbol{u}_0, \boldsymbol{u}_0') = \boldsymbol{0}$ for $\boldsymbol{u}_0'$, which in general is a nonlinear system of equations (unless $\boldsymbol{F}(x, \boldsymbol{u}, \boldsymbol{v})$ depends linearly on $\boldsymbol{v}$). But from then on, when integrating (5.26) (or the equivalent first-order system), only linear systems (5.25) need to be solved at each step to compute $\mathrm{d}^2\boldsymbol{u}/\mathrm{d}x^2$ for given $x, \boldsymbol{u}, \boldsymbol{u}'$, since the numerical method automatically updates $x, \boldsymbol{u}, \boldsymbol{u}'$ from step to step.

5.3 Existence and Uniqueness

We recall from the theory of differential equations the following basic existence and uniqueness theorem.

Theorem 5.3.1. *Assume that $\boldsymbol{f}(x, \boldsymbol{y})$ is continuous in the first variable for $x \in [a, b]$ and with respect to the second satisfies a uniform Lipschitz condition*

$$\|\boldsymbol{f}(x, \boldsymbol{y}) - \boldsymbol{f}(x, \boldsymbol{y}^*)\| \le L\|\boldsymbol{y} - \boldsymbol{y}^*\|, \ \ x \in [a, b], \boldsymbol{y}, \boldsymbol{y}^* \in \mathbb{R}^d, \tag{5.27}$$

where $\|\cdot\|$ is some vector norm. Then the initial value problem (5.16) has a unique solution $\boldsymbol{y}(x)$, $a \leq x \leq b$, for arbitrary $\boldsymbol{y}_0 \in \mathbb{R}^d$. Moreover, $\boldsymbol{y}(x)$ depends continuously on x_0 and $\boldsymbol{y}_0$.

The Lipschitz condition (5.27) certainly holds if all functions $\frac{\partial f^i}{\partial y^j}(x, \boldsymbol{y})$, $i, j = 1, 2, \ldots, d$, are continuous in the y-variables and bounded on $[a, b] \times \mathbb{R}^d$. This is the case for *linear systems of differential equations*, where

$$f^i(x, \boldsymbol{y}) = \sum_{j=1}^{d} a_{ij}(x) y^j + b_i(x), \quad i = 1, 2, \ldots, d, \tag{5.28}$$

and $a_{ij}(x)$, $b_i(x)$ are continuous functions on $[a, b]$. In nonlinear problems, it is rarely the case, however, that a Lipschitz condition is valid uniformly in all of $\mathbb{R}^d$; it more often holds in some compact and convex domain $\mathcal{D} \subset \mathbb{R}^d$. In this case, one can assert the existence of a unique solution only in some neighborhood of x_0 in which $\boldsymbol{y}(x)$ remains in $\mathcal{D}$. To avoid this complication, we assume in this chapter that $\mathcal{D}$ is so large that $\boldsymbol{y}(x)$ exists on the whole interval $[a, b]$ and that all numerical approximations are also contained in $\mathcal{D}$. Bounds on partial derivatives of $\boldsymbol{f}(x, \boldsymbol{y})$ are assumed to hold uniformly in $[a, b] \times \mathcal{D}$, if not in $[a, b] \times \mathbb{R}^d$, and it is tacitly understood that the bounds may depend on $\mathcal{D}$ but not on x and $\boldsymbol{y}$.

5.4 Numerical Methods

One can distinguish between *analytic approximation methods* and *discrete-variable methods*. In the former, one tries to find approximations $\boldsymbol{y}_a(x) \approx \boldsymbol{y}(x)$ to the exact solution, valid for all $x \in [a, b]$. These usually take the form of a truncated series expansion, either in powers of x, in Chebyshev polynomials, or in some other system of basis functions. In discrete-variable methods, on the other hand, one attempts to find approximations $\boldsymbol{u}_n \in \mathbb{R}^d$ of $\boldsymbol{y}(x_n)$ only at discrete points $x_n \in [a, b]$. The abscissas x_n may be predetermined (e.g., equally spaced on $[a, b]$) or, more likely, are generated dynamically as part of the integration process. If desired, one can from these discrete approximations $\{\boldsymbol{u}_n\}$ again obtain an approximation $\boldsymbol{y}_a(x)$ defined for all $x \in [a, b]$ either by interpolation or, more naturally, by a continuation mechanism built into the approximation method itself. We are concerned here only with discrete-variable methods.

Depending on how the discrete approximations are generated, one distinguishes between *one-step methods* and *multistep methods*. In the former, $\boldsymbol{u}_{n+1}$ is determined solely from a knowledge of x_n, $\boldsymbol{u}_n$ and the step h to proceed from x_n to $x_{n+1} = x_n + h$. In a k-step method ($k > 1$), knowledge of $k - 1$ additional points $(x_{n-\kappa}, \boldsymbol{u}_{n-\kappa})$, $\kappa = 1, 2, \ldots, k - 1$, is required to advance the solution. This chapter is devoted to one-step methods; multistep methods are discussed in Chap. 6.

When describing a single step of a one-step method, it suffices to show how one proceeds from a generic point $(x, \boldsymbol{y})$, $x \in [a,b]$, $\boldsymbol{y} \in \mathbb{R}^d$, to the "next" point $(x+h, \boldsymbol{y}_{\text{next}})$. We refer to this as the *local description* of the one-step method. This also includes a discussion of the local accuracy, that is, how closely $\boldsymbol{y}_{\text{next}}$ agrees at $x+h$ with the solution (passing through the point $(x, \boldsymbol{y})$) of the differential equation. A one-step method for solving the initial value problem (5.16) effectively generates a grid function $\{\boldsymbol{u}_n\}$, $\boldsymbol{u}_n \in \mathbb{R}^d$, on a grid $a = x_0 < x_1 < x_2 < \cdots < x_{N-1} < x_N = b$ covering the interval $[a,b]$, whereby $\boldsymbol{u}_n$ is intended to approximate the exact solution $\boldsymbol{y}(x)$ at $x = x_n$. The point $(x_{n+1}, \boldsymbol{u}_{n+1})$ is obtained from the point $(x_n, \boldsymbol{u}_n)$ by applying a one-step method with an appropriate step $h_n = x_{n+1} - x_n$. This is referred to as the *global* description of a one-step method. Questions of interest here are the behavior of the global error $\boldsymbol{u}_n - \boldsymbol{y}(x_n)$, in particular stability and convergence, and the choice of h_n to proceed from one grid point x_n to the next, $x_{n+1} = x_n + h_n$. Finally, we address special difficulties arising from the stiffness of the given differential equation problem.

5.5 Local Description of One-Step Methods

Given a generic point $x \in [a,b]$, $\boldsymbol{y} \in \mathbb{R}^d$, we define a single step of the one-step method by

$$\boldsymbol{y}_{\text{next}} = \boldsymbol{y} + h\boldsymbol{\Phi}(x, \boldsymbol{y}; h), \quad h > 0. \tag{5.29}$$

The function $\boldsymbol{\Phi}$: $[a,b] \times \mathbb{R}^d \times \mathbb{R}_+ \to \mathbb{R}^d$ may be thought of as the approximate increment per unit step, or the approximate difference quotient, and it defines the method. Along with (5.29), we consider the solution $\boldsymbol{u}(t)$ of the differential equation (5.16) passing through the point $(x, \boldsymbol{y})$, that is, the local initial value problem

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = \boldsymbol{f}(t, \boldsymbol{u}),\ x \le t \le x+h;\ \boldsymbol{u}(x) = \boldsymbol{y}. \tag{5.30}$$

We call $\boldsymbol{u}(t)$ the *reference solution*. The vector $\boldsymbol{y}_{\text{next}}$ in (5.29) is intended to approximate $\boldsymbol{u}(x+h)$. How successfully this is done is measured by the truncation error defined as follows.

Definition 5.5.1. The *truncation error* of the method $\boldsymbol{\Phi}$ at the point $(x, \boldsymbol{y})$ is defined by

$$\boldsymbol{T}(x, \boldsymbol{y}; h) = \frac{1}{h}\,[\boldsymbol{y}_{\text{next}} - \boldsymbol{u}(x+h)]. \tag{5.31}$$

The truncation error thus is a vector-valued function of $d+2$ variables. Using (5.29) and (5.30), we can write for it, alternatively,

$$\boldsymbol{T}(x, \boldsymbol{y}; h) = \boldsymbol{\Phi}(x, \boldsymbol{y}; h) - \frac{1}{h}\,[\boldsymbol{u}(x+h) - \boldsymbol{u}(x)], \tag{5.32}$$

showing that $\boldsymbol{T}$ is the difference between the approximate and exact increment per unit step.

An increasingly finer description of local accuracy is provided by the following definitions, all based on the concept of truncation error.

Definition 5.5.2. The method $\boldsymbol{\Phi}$ is called *consistent* if

$$\boldsymbol{T}(x, \boldsymbol{y}; h) \to \boldsymbol{0} \text{ as } h \to 0, \tag{5.33}$$

uniformly for $(x, \boldsymbol{y}) \in [a, b] \times \mathbb{R}^d$.[3]

By (5.32) and (5.30) we have consistency if and only if

$$\boldsymbol{\Phi}(x, \boldsymbol{y}; 0) = \boldsymbol{f}(x, \boldsymbol{y}), \quad x \in [a, b], \ \boldsymbol{y} \in \mathbb{R}^d. \tag{5.34}$$

Definition 5.5.3. The method $\boldsymbol{\Phi}$ is said to have *order* p if, for some vector norm $\| \cdot \|$,

$$\|\boldsymbol{T}(x, \boldsymbol{y}; h)\| \le C h^p, \tag{5.35}$$

uniformly on $[a, b] \times \mathbb{R}^d$, with a constant C not depending on x, $\boldsymbol{y}$ and h.[4]

We express this property briefly as

$$\boldsymbol{T}(x, \boldsymbol{y}; h) = O(h^p), \quad h \to 0. \tag{5.36}$$

Note that $p > 0$ implies consistency. Usually, p is an integer ≥ 1. It is called the *exact order*, if (5.35) does not hold for any larger p.

Definition 5.5.4. A function $\boldsymbol{\tau}\colon [a, b] \times \mathbb{R}^d \to \mathbb{R}^d$ that satisfies $\boldsymbol{\tau}(x, \boldsymbol{y}) \not\equiv \boldsymbol{0}$ and

$$\boldsymbol{T}(x, \boldsymbol{y}; h) = \boldsymbol{\tau}(x, \boldsymbol{y}) h^p + O(h^{p+1}), \quad h \to 0, \tag{5.37}$$

is called the *principal error function*.

The principal error function determines the leading term in the truncation error. The number p in (5.37) is the exact order of the method since $\boldsymbol{\tau} \not\equiv \boldsymbol{0}$.

All the preceding definitions are made with the idea in mind that $h > 0$ is a small number. Then the larger p, the more accurate the method. This can (and should) always be arranged by a proper scaling of the independent variable x; we tacitly assume that such a scaling has already been made.

[3] More realistically, one should require uniformity on $[a, b] \times \mathcal{D}$, where $\mathcal{D} \subset \mathbb{R}^d$ is a sufficiently large compact domain; cf. Sect. 5.3.

[4] If uniformity of (5.35) is required only on $[a, b] \times \mathcal{D}$, $\mathcal{D} \subset \mathbb{R}^d$ compact, then C may depend on $\mathcal{D}$; cf. Sect. 5.3.

5.6 Examples of One-Step Methods

Some of the oldest methods are motivated by simple geometric considerations based on the slope field defined by the right-hand side of the differential equation. These include the Euler and modified Euler methods. More accurate and sophisticated methods are based on Taylor expansion.

5.6.1 Euler's Method

Euler proposed his method in 1768, in the early days of calculus. It consists of simply following the slope at the generic point $(x, \boldsymbol{y})$ over an interval of length h:

$$\boldsymbol{y}_{\text{next}} = \boldsymbol{y} + h\boldsymbol{f}(x, \boldsymbol{y}). \tag{5.38}$$

Thus, $\boldsymbol{\Phi}(x, \boldsymbol{y}; h) = \boldsymbol{f}(x, \boldsymbol{y})$ does not depend on h, and by (5.34) the method is evidently consistent. For the truncation error, we have by (5.32)

$$\boldsymbol{T}(x, \boldsymbol{y}; h) = \boldsymbol{f}(x, \boldsymbol{y}) - \frac{1}{h}[\boldsymbol{u}(x+h) - \boldsymbol{u}(x)], \tag{5.39}$$

where $\boldsymbol{u}(t)$ is the reference solution defined in (5.30). Since $\boldsymbol{u}'(x) = \boldsymbol{f}(x, \boldsymbol{u}(x)) = \boldsymbol{f}(x, \boldsymbol{y})$, we can write, using Taylor's theorem,

$$\begin{aligned}\boldsymbol{T}(x, \boldsymbol{y}; h) &= \boldsymbol{u}'(x) - \frac{1}{h}[\boldsymbol{u}(x+h) - \boldsymbol{u}(x)] \\ &= \boldsymbol{u}'(x) - \frac{1}{h}\left[\boldsymbol{u}(x) + h\boldsymbol{u}'(x) + \frac{1}{2}h^2\boldsymbol{u}''(\xi) - \boldsymbol{u}(x)\right] \\ &= -\frac{1}{2}h\boldsymbol{u}''(\xi), \quad x < \xi < x + h, \end{aligned} \tag{5.40}$$

assuming $\boldsymbol{u} \in C^2[x, x+h]$. This is certainly true if $\boldsymbol{f} \in C^1$ on $[a, b] \times \mathbb{R}^d$, as we assume. Note the slight abuse of notation in the last two equations, where ξ is to be understood to differ from component to component but to be always in the interval shown. We freely use this notation later on without further comment.

Now differentiating (5.30) totally with respect to t and then setting $t = \xi$ yields

$$\boldsymbol{T}(x, \boldsymbol{y}; h) = -\frac{1}{2}h[\boldsymbol{f}_x + \boldsymbol{f}_y\boldsymbol{f}](\xi, \boldsymbol{u}(\xi)), \tag{5.41}$$

where $\boldsymbol{f}_x$ is the partial derivative of $\boldsymbol{f}$ with respect to x and $\boldsymbol{f}_{\boldsymbol{y}}$ the Jacobian of $\boldsymbol{f}$ with respect to the y-variables. If, in the spirit of Theorem 5.3.1, we assume that $\boldsymbol{f}$ and all its first partial derivatives are uniformly bounded in $[a,b]\times\mathbb{R}^d$, there exists a constant C independent of x, $\boldsymbol{y}$, and h such that

$$\|\boldsymbol{T}(x,\boldsymbol{y};h)\| \le C\cdot h. \tag{5.42}$$

Thus, Euler's method has order $p=1$. If we make the same assumption about all second-order partial derivatives of $\boldsymbol{f}$, we have $\boldsymbol{u}''(\xi)=\boldsymbol{u}''(x)+O(h)$ and, therefore, from (5.40),

$$\boldsymbol{T}(x,\boldsymbol{y};h)=-\tfrac{1}{2}h\,[\boldsymbol{f}_x+\boldsymbol{f}_{\boldsymbol{y}}\boldsymbol{f}](x,\boldsymbol{y})+O(h^2),\quad h\to 0, \tag{5.43}$$

showing that the principal error function is given by

$$\boldsymbol{\tau}(x,\boldsymbol{y})=-\tfrac{1}{2}\,[\boldsymbol{f}_x+\boldsymbol{f}_{\boldsymbol{y}}\boldsymbol{f}](x,\boldsymbol{y}). \tag{5.44}$$

Unless $\boldsymbol{f}_x+\boldsymbol{f}_{\boldsymbol{y}}\boldsymbol{f}\equiv\boldsymbol{0}$, the order of Euler's method is exactly $p=1$.

5.6.2 Method of Taylor Expansion

We have seen that Euler's method basically amounts to truncating the Taylor expansion of the reference solution after its second term. It is a natural idea, already proposed by Euler, to use more terms of the Taylor expansion. This requires the computation of successive "total derivatives" of $\boldsymbol{f}$,

$$\begin{aligned}&\boldsymbol{f}^{[0]}(x,\boldsymbol{y})=\boldsymbol{f}(x,\boldsymbol{y}),\\&\boldsymbol{f}^{[k+1]}(x,\boldsymbol{y})=\boldsymbol{f}_x^{[k]}(x,\boldsymbol{y})+\boldsymbol{f}_{\boldsymbol{y}}^{[k]}(x,\boldsymbol{y})\boldsymbol{f}(x,\boldsymbol{y}),\quad k=0,1,2,\ldots,\end{aligned} \tag{5.45}$$

which determine (see Ex. 2) the successive derivatives of the reference solution $\boldsymbol{u}(t)$ of (5.30) by virtue of

$$\boldsymbol{u}^{(k+1)}(t)=\boldsymbol{f}^{[k]}(t,\boldsymbol{u}(t)),\quad k=0,1,2,\ldots. \tag{5.46}$$

These, for $t=x$, become

$$\boldsymbol{u}^{(k+1)}(x)=\boldsymbol{f}^{[k]}(x,\boldsymbol{y}),\quad k=0,1,2,\ldots, \tag{5.47}$$

and are used to form the Taylor series approximation according to

$$\boldsymbol{y}_{\text{next}}=\boldsymbol{y}+h\left[\boldsymbol{f}^{[0]}(x,\boldsymbol{y})+\frac{1}{2}h\boldsymbol{f}^{[1]}(x,\boldsymbol{y})+\cdots+\frac{1}{p!}h^{p-1}\boldsymbol{f}^{[p-1]}(x,\boldsymbol{y})\right]; \tag{5.48}$$

that is,

$$\boldsymbol{\Phi}(x,\boldsymbol{y};h) = \boldsymbol{f}^{[0]}(x,\boldsymbol{y}) + \frac{1}{2}h\boldsymbol{f}^{[1]}(x,\boldsymbol{y}) + \cdots + \frac{1}{p!}h^{p-1}\boldsymbol{f}^{[p-1]}(x,\boldsymbol{y}). \tag{5.49}$$

For the truncation error, assuming $\boldsymbol{f} \in C^p$ on $[a,b] \times \mathbb{R}^d$ and using (5.47) and (5.49), we obtain from Taylor's theorem

$$\begin{aligned}\boldsymbol{T}(x,\boldsymbol{y};h) &= \boldsymbol{\Phi}(x,\boldsymbol{y};h) - \frac{1}{h}[\boldsymbol{u}(x+h) - \boldsymbol{u}(x)] \\ &= \boldsymbol{\Phi}(x,\boldsymbol{y};h) - \sum_{k=0}^{p-1}\boldsymbol{u}^{(k+1)}(x)\frac{h^k}{(k+1)!} - \boldsymbol{u}^{(p+1)}(\xi)\frac{h^p}{(p+1)!} \\ &= -\boldsymbol{u}^{(p+1)}(\xi)\frac{h^p}{(p+1)!}, \quad x < \xi < x+h,\end{aligned}$$

so that

$$\|\boldsymbol{T}(x,\boldsymbol{y};h)\| \le \frac{C_p}{(p+1)!}h^p, \tag{5.50}$$

where C_p is a bound on the pth total derivative of $\boldsymbol{f}$. Thus, the method has exact order p (unless $\boldsymbol{f}^{[p]}(x,\boldsymbol{y}) \equiv \boldsymbol{0}$), and the principal error function is

$$\boldsymbol{\tau}(x,\boldsymbol{y}) = -\frac{1}{(p+1)!}\boldsymbol{f}^{[p]}(x,\boldsymbol{y}). \tag{5.51}$$

The necessity of computing many partial derivatives in (5.45) was a discouraging factor in the past, when this had to be done by hand. But nowadays, this labor can be delegated to computers, so that the method has become again a viable option.

5.6.3 Improved Euler Methods

There is too much inertia in Euler's method: one should not follow the same initial slope over the whole interval of length h, since along this line segment the slope defined by the slope field of the differential equation changes. This suggests several alternatives. For example, we may wish to reevaluate the slope halfway through the line segment – retake the pulse of the differential equation, as it were – and then follow this revised slope over the whole interval (cf. Fig. 5.1). In formula,

$$\boldsymbol{y}_{\text{next}} = \boldsymbol{y} + h\boldsymbol{f}\left(x + \frac{1}{2}h, \boldsymbol{y} + \frac{1}{2}h\boldsymbol{f}(x,\boldsymbol{y})\right), \tag{5.52}$$

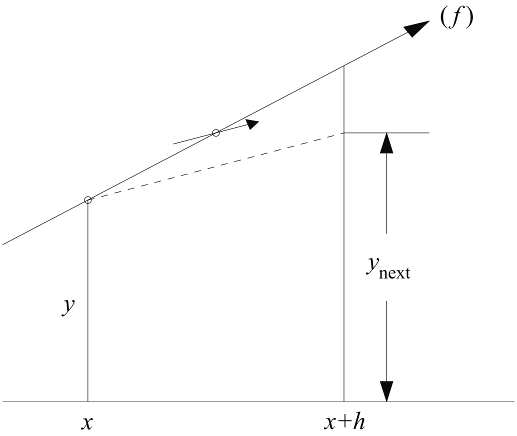


Fig. 5.1 Modified Euler method

or

$$\boldsymbol{\Phi}(x, \boldsymbol{y}; h) = \boldsymbol{f}\left(x + \frac{1}{2}h, \boldsymbol{y} + \frac{1}{2}h\boldsymbol{f}(x, \boldsymbol{y})\right). \tag{5.53}$$

Note the characteristic "nesting" of $\boldsymbol{f}$ that is required here. For programming purposes, it may be desirable to undo the nesting and write

$$\begin{aligned} \boldsymbol{k}_1(x, \boldsymbol{y}) &= \boldsymbol{f}(x, \boldsymbol{y}), \\ \boldsymbol{k}_2(x, \boldsymbol{y}; h) &= \boldsymbol{f}\left(x + \frac{1}{2}h, \boldsymbol{y} + \frac{1}{2}h\boldsymbol{k}_1\right), \\ \boldsymbol{y}_{\text{next}} &= \boldsymbol{y} + h\boldsymbol{k}_2. \end{aligned} \tag{5.54}$$

In other words, we are taking two trial slopes, $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$, one at the initial point and the other nearby, and then taking the latter as the final slope.

We could equally well take the second trial slope at $(x + h, \boldsymbol{y} + h\boldsymbol{f}(x, \boldsymbol{y}))$, but then, having waited too long before reevaluating the slope, take now as the final slope the average of the two slopes:

$$\begin{aligned} \boldsymbol{k}_1(x, \boldsymbol{y}) &= \boldsymbol{f}(x, \boldsymbol{y}), \\ \boldsymbol{k}_2(x, \boldsymbol{y}; h) &= \boldsymbol{f}(x + h, \boldsymbol{y} + h\boldsymbol{k}_1), \\ \boldsymbol{y}_{\text{next}} &= \boldsymbol{y} + \frac{1}{2}h(\boldsymbol{k}_1 + \boldsymbol{k}_2). \end{aligned} \tag{5.55}$$

This is sometimes referred to as *Heun's method* or the *trapezoidal rule*.

The effect of both modifications is to raise the order by 1, as is shown in the next section.

5.6.4 Second-Order Two-Stage Methods

We may take a more systematic approach toward modifying Euler's method, by letting

$$\boldsymbol{\Phi}(x, \boldsymbol{y}; h) = \alpha_1 \boldsymbol{k}_1 + \alpha_2 \boldsymbol{k}_2, \tag{5.56}$$

where

$$\begin{aligned} \boldsymbol{k}_1(x, \boldsymbol{y}) &= \boldsymbol{f}(x, \boldsymbol{y}), \\ \boldsymbol{k}_2(x, \boldsymbol{y}; h) &= \boldsymbol{f}(x + \mu h, \boldsymbol{y} + \mu h \boldsymbol{k}_1). \end{aligned} \tag{5.57}$$

We have now three parameters, α_1, α_2, and μ, at our disposal, and we can try to choose them so as to maximize the order p. A systematic way of determining the maximum order p is to expand both $\boldsymbol{\Phi}(x, \boldsymbol{y}; h)$ and $h^{-1}[\boldsymbol{u}(x+h) - \boldsymbol{u}(x)]$ in powers of h and to match as many terms as we can.

To expand $\boldsymbol{\Phi}$, we need Taylor's expansion for (vector-valued) functions of several variables,

$$\begin{aligned} \boldsymbol{f}(x + \Delta x, \boldsymbol{y} + \Delta \boldsymbol{y}) =& \boldsymbol{f} + \boldsymbol{f}_x \Delta x + \boldsymbol{f}_{\boldsymbol{y}} \Delta \boldsymbol{y} + \frac{1}{2}\big[\boldsymbol{f}_{xx}(\Delta x)^2 + 2\boldsymbol{f}_{x\boldsymbol{y}} \Delta x \Delta \boldsymbol{y} \\ &+ (\Delta \boldsymbol{y})^T \boldsymbol{f}_{\boldsymbol{y}\boldsymbol{y}}(\Delta \boldsymbol{y})\big] + \cdots, \end{aligned} \tag{5.58}$$

where $\boldsymbol{f}_{\boldsymbol{y}}$ denotes the Jacobian of $\boldsymbol{f}$ and $\boldsymbol{f}_{\boldsymbol{y}\boldsymbol{y}} = [f^i_{\boldsymbol{y}\boldsymbol{y}}]$ the vector of Hessian matrices of $\boldsymbol{f}$. In (5.58), all functions and partial derivatives are understood to be evaluated at $(x, \boldsymbol{y})$. Letting $\Delta x = \mu h$, $\Delta \boldsymbol{y} = \mu h \boldsymbol{f}$ then gives

$$\begin{aligned} \boldsymbol{k}_2(x, \boldsymbol{y}; h) =& \ \boldsymbol{f} + \mu h(\boldsymbol{f}_x + \boldsymbol{f}_{\boldsymbol{y}} \boldsymbol{f}) + \frac{1}{2}\mu^2 h^2(\boldsymbol{f}_{xx} + 2\boldsymbol{f}_{x\boldsymbol{y}}\boldsymbol{f} + \boldsymbol{f}^{\mathrm{T}} \boldsymbol{f}_{\boldsymbol{y}\boldsymbol{y}} \boldsymbol{f}) \\ &+ O(h^3). \end{aligned} \tag{5.59}$$

Similarly (cf. (5.47)),

$$\frac{1}{h}[\boldsymbol{u}(x+h) - \boldsymbol{u}(x)] = \boldsymbol{u}'(x) + \frac{1}{2}h\boldsymbol{u}''(x) + \frac{1}{6}h^2\boldsymbol{u}'''(x) + O(h^3), \tag{5.60}$$

where

$$\begin{aligned} \boldsymbol{u}'(x) &= \boldsymbol{f}, \\ \boldsymbol{u}''(x) &= \boldsymbol{f}^{[1]} = \boldsymbol{f}_x + \boldsymbol{f}_{\boldsymbol{y}} \boldsymbol{f}, \\ \boldsymbol{u}'''(x) &= \boldsymbol{f}^{[2]} = \boldsymbol{f}^{[1]}_x + \boldsymbol{f}^{[1]}_{\boldsymbol{y}} \boldsymbol{f} \\ &= \boldsymbol{f}_{xx} + \boldsymbol{f}_{x\boldsymbol{y}}\boldsymbol{f} + \boldsymbol{f}_{\boldsymbol{y}}\boldsymbol{f}_x + (\boldsymbol{f}_{x\boldsymbol{y}} + (\boldsymbol{f}_{\boldsymbol{y}}\boldsymbol{f})_{\boldsymbol{y}})\boldsymbol{f} \\ &= \boldsymbol{f}_{xx} + 2\boldsymbol{f}_{x\boldsymbol{y}}\boldsymbol{f} + \boldsymbol{f}^{\mathrm{T}}\boldsymbol{f}_{\boldsymbol{y}\boldsymbol{y}}\boldsymbol{f} + \boldsymbol{f}_{\boldsymbol{y}}(\boldsymbol{f}_x + \boldsymbol{f}_{\boldsymbol{y}}\boldsymbol{f}), \end{aligned}$$

and where in the last equation we have used (see Ex. 3)

$$(\boldsymbol{f}_y\boldsymbol{f})_y\boldsymbol{f} = \boldsymbol{f}^{\mathrm{T}}\boldsymbol{f}_{yy}\boldsymbol{f} + \boldsymbol{f}_y^2\boldsymbol{f}.$$

Now,

$$\boldsymbol{T}(x,\boldsymbol{y};h) = \alpha_1\boldsymbol{k}_1 + \alpha_2\boldsymbol{k}_2 - \frac{1}{h}[\boldsymbol{u}(x+h) - \boldsymbol{u}(x)],$$

wherein we substitute the expansions (5.59) and (5.60). We find

$$\begin{aligned}\boldsymbol{T}(x,\boldsymbol{y};h) = (\alpha_1+\alpha_2-1)\boldsymbol{f} + \left(\alpha_2\mu - \frac{1}{2}\right)h(\boldsymbol{f}_x + \boldsymbol{f}_y\boldsymbol{f}) + \frac{1}{2}h^2\Bigg[\left(\alpha_2\mu^2 - \frac{1}{3}\right)\\ \times(\boldsymbol{f}_{xx} + 2\boldsymbol{f}_{xy}\boldsymbol{f} + \boldsymbol{f}^{\mathrm{T}}\boldsymbol{f}_{yy}\boldsymbol{f}) - \frac{1}{3}\boldsymbol{f}_y(\boldsymbol{f}_x + \boldsymbol{f}_y\boldsymbol{f})\Bigg] + O(h^3).\end{aligned} \tag{5.61}$$

We can see now that, however we choose the parameters α_1, α_2, μ, we cannot make the coefficient of h^2 equal to zero unless severe restrictions are placed on $\boldsymbol{f}$ (cf. Ex. 6(c)). Thus, the maximum possible order is $p = 2$, and we achieve it by satisfying

$$\begin{aligned}\alpha_1 + \alpha_2 &= 1,\\ \alpha_2\mu &= \frac{1}{2}.\end{aligned} \tag{5.62}$$

This has a one-parameter family of solutions,

$$\begin{aligned}\alpha_1 &= 1 - \alpha_2,\\ \mu &= \frac{1}{2\alpha_2},\end{aligned} \quad (\alpha_2 \neq 0 \text{ arbitrary}). \tag{5.63}$$

We recognize the improved Euler method contained therein with $\alpha_2 = 1$, and Heun's method with $\alpha_2 = \frac{1}{2}$. There are other natural choices; one such would be to look at the principal error function

$$\boldsymbol{\tau}(x,\boldsymbol{y}) = \frac{1}{2}\left[\left(\frac{1}{4\alpha_2} - \frac{1}{3}\right)(\boldsymbol{f}_{xx} + 2\boldsymbol{f}_{xy}\boldsymbol{f} + \boldsymbol{f}^{\mathrm{T}}\boldsymbol{f}_{yy}\boldsymbol{f}) - \frac{1}{3}\boldsymbol{f}_y(\boldsymbol{f}_x + \boldsymbol{f}_y\boldsymbol{f})\right] \tag{5.64}$$

and to note that it consists of a linear combination of two aggregates of partial derivatives. We may wish to minimize some norm of the coefficients, say, the sum of their absolute values. In (5.64), this gives trivially $(4\alpha_2)^{-1} - \frac{1}{3} = 0$, that is, $\alpha_2 = \frac{3}{4}$, and hence suggests a method with

$$\alpha_1 = \tfrac{1}{4},\ \alpha_2 = \tfrac{3}{4},\ \mu = \tfrac{2}{3}. \tag{5.65}$$

5.6.5 Runge–Kutta Methods

Runge–Kutta methods are a straightforward extension of two-stage methods to r-stage methods:

$$\begin{aligned}
\boldsymbol{\Phi}(x,\boldsymbol{y};h) &= \sum_{s=1}^{r} \alpha_s \boldsymbol{k}_s,\\
\boldsymbol{k}_1(x,\boldsymbol{y}) &= \boldsymbol{f}(x,\boldsymbol{y}),\\
\boldsymbol{k}_s(x,\boldsymbol{y};h) &= \boldsymbol{f}\left(x+\mu_s h, \boldsymbol{y}+h\sum_{j=1}^{s-1}\lambda_{sj}\boldsymbol{k}_j\right),\quad s=2,3,\dots,r.
\end{aligned} \tag{5.66}$$

It is natural in (5.66) to impose the conditions (cf. Ex. 10 for the first)

$$\mu_s = \sum_{j=1}^{s-1}\lambda_{sj}, s=2,3,\dots,r;\ \sum_{s=1}^{r}\alpha_s = 1, \tag{5.67}$$

where the last one is nothing but the consistency condition (cf. (5.34)). We call (5.66) an *explicit r-stage Runge–Kutta method*; it requires r evaluations of the right-hand side $\boldsymbol{f}$ of the differential equation. More generally, we can consider *implicit r-stage Runge–Kutta methods*

$$\begin{aligned}
\boldsymbol{\Phi}(x,\boldsymbol{y};h) &= \sum_{s=1}^{r} \alpha_s \boldsymbol{k}_s(x,\boldsymbol{y};h),\\
\boldsymbol{k}_s &= \boldsymbol{f}\left(x+\mu_s h, \boldsymbol{y}+h\sum_{j=1}^{r}\lambda_{sj}\boldsymbol{k}_j\right),\quad s=1,2,\dots,r,
\end{aligned} \tag{5.68}$$

in which the last r equations form a system of (in general nonlinear) equations in the unknowns $\boldsymbol{k}_1, \boldsymbol{k}_2, \dots, \boldsymbol{k}_r$. Since each of these is a vector in $\mathbb{R}^d$, we have a system of rd equations in rd unknowns that must be solved before we can form the approximate increment $\boldsymbol{\Phi}$. Less work is required in *semi-implicit* Runge–Kutta methods, where the summation in the formula for $\boldsymbol{k}_s$ extends from $j=1$ to $j=s$ only. This yields r systems of equations, each having only d unknowns, the components of $\boldsymbol{k}_s$. The considerable computational expense involved in implicit and semi-implicit methods can only be justified in special circumstances, for example, stiff problems. The reason is that implicit methods not only can be made to have higher order than explicit methods, but also have better stability properties (cf. Sect. 5.9).

Already in the case of *explicit* Runge–Kutta methods, and even more so in implicit methods, we have at our disposal a large number of parameters which we can choose to achieve the maximum possible order for all sufficiently smooth $\boldsymbol{f}$.

The approach is analogous to the one taken in Sect. 5.2.4, only technically much more involved, as the partial-derivative aggregates that are going to appear in the principal error function are becoming much more complicated and numerous as r increases. In fact, a satisfactory solution of this problem has become possible only through the employment of graph-theoretical tools, specifically, the theory of rooted trees. This was systematically developed by J.C. Butcher, the principal researcher in this area. A description of these techniques is beyond the scope of this book. We briefly summarize, however, some of the results that can be obtained.

Denote by $p^*(r)$ the maximum attainable order (for arbitrary sufficiently smooth $\boldsymbol{f}$) of an explicit r-stage Runge–Kutta method. Then Kutta[5] has already shown in 1901 that

$$p^*(r) = r \quad \text{for} \quad 1 \le r \le 4, \tag{5.69}$$

and has derived many concrete examples of methods having these orders. Butcher, in the 1960s, established that

$$\begin{aligned} p^*(r) &= r-1 \quad \text{for} \quad 5 \le r \le 7, \\ p^*(r) &= r-2 \quad \text{for} \quad 8 \le r \le 9, \\ p^*(r) &\le r-2 \quad \text{for} \quad r \ge 10. \end{aligned} \tag{5.70}$$

Specific examples of higher-order Runge–Kutta formulae are used later in connection with error monitoring procedures. Here, we mention only the *classical Runge–Kutta formula*[6] of order $p = 4$:

$$\begin{aligned} \boldsymbol{\Phi}(x, \boldsymbol{y}; h) &= \frac{1}{6}(\boldsymbol{k}_1 + 2\boldsymbol{k}_2 + 2\boldsymbol{k}_3 + \boldsymbol{k}_4), \\ \boldsymbol{k}_1(x, \boldsymbol{y}) &= \boldsymbol{f}(x, \boldsymbol{y}), \\ \boldsymbol{k}_2(x, \boldsymbol{y}; h) &= \boldsymbol{f}\left(x + \frac{1}{2}h, \boldsymbol{y} + \frac{1}{2}h\boldsymbol{k}_1\right), \\ \boldsymbol{k}_3(x, \boldsymbol{y}; h) &= \boldsymbol{f}\left(x + \frac{1}{2}h, \boldsymbol{y} + \frac{1}{2}h\boldsymbol{k}_2\right), \\ \boldsymbol{k}_4(x, \boldsymbol{y}; h) &= \boldsymbol{f}(x + h, \boldsymbol{y} + h\boldsymbol{k}_3). \end{aligned} \tag{5.71}$$

[5] Wilhelm Martin Kutta (1867–1944) was a German applied mathematician, active at the Technical University of Stuttgart from 1911 until his retirement. In addition to his work on the numerical solution of ODEs, he did important work on the application of conformal mapping to hydro- and aerodynamical problems. Best known is his formula for the lift exerted on an airfoil, now known as the Kutta–Joukowski formula. For Runge, see footnote 7 in Chap. 2, Sect. 2.2.3

[6] Runge's idea, in 1895, was to generalize Simpson's quadrature formula (cf. Chap. 3, Sect. 3.2.1) to ordinary differential equations. He succeeded only partially in that the generalization he gave had stage number $r = 4$ but only order $p = 3$. The method (5.71) of order $p = 4$ was discovered in 1901 by Kutta through a systematic search.

When $\boldsymbol{f}$ does not depend on $\boldsymbol{y}$, and thus we are in the case of a numerical quadrature problem (cf. (5.2)), then (5.71) reduces to Simpson's formula.

5.7 Global Description of One-Step Methods

We now turn to the numerical solution of the initial value problem (5.16) with the help of one-step formulae such as those developed in Sect. 5.6. The description of such one-step methods is best done in terms of grids and grid functions. A *grid* on the interval $[a, b]$ is a set of points $\{x_n\}_{n=0}^N$ such that

$$a = x_0 < x_1 < x_2 < \cdots < x_{N-1} < x_N = b, \tag{5.72}$$

with *grid lengths* h_n defined by

$$h_n = x_{n+1} - x_n, \quad n = 0, 1, \ldots, N-1. \tag{5.73}$$

The *fineness* of the grid is measured by

$$|h| = \max_{0 \le n \le N-1} h_n. \tag{5.74}$$

We often use the letter h to designate the collection of lengths $h = \{h_n\}$. If $h_1 = h_2 = \cdots = h_{N-1} = (b-a)/N$, we call (5.72) a *uniform grid*, otherwise a *nonuniform grid*. For uniform grids, we use the letter h also to designate the common grid length $h = (b-a)/N$. A vector-valued function $\boldsymbol{v} = \{\boldsymbol{v}_n\}$, $\boldsymbol{v}_n \in \mathbb{R}^d$, defined on the grid (5.72) is called a *grid function*. Thus, $\boldsymbol{v}_n$ is the value of $\boldsymbol{v}$ at the gridpoint x_n. Every function $\boldsymbol{v}(x)$ defined on $[a, b]$ induces a grid function by restriction. We denote the set of grid functions on $[a, b]$ by $\Gamma_h[a, b]$, and for each grid function $\boldsymbol{v} = \{\boldsymbol{v}_n\}$ define its norm by

$$\|\boldsymbol{v}\|_\infty = \max_{0 \le n \le N} \|\boldsymbol{v}_n\|, \quad \boldsymbol{v} \in \Gamma_h[a, b]. \tag{5.75}$$

A one-step method – indeed, any discrete-variable method – is a method producing a grid function $\boldsymbol{u} = \{\boldsymbol{u}_n\}$ such that $\boldsymbol{u} \approx \boldsymbol{y}$, where $\boldsymbol{y} = \{\boldsymbol{y}_n\}$ is the grid function induced by the exact solution $\boldsymbol{y}(x)$ of the initial value problem (5.16). The grid (5.72) may be predetermined, for example, a uniform grid, or, as is more often the case in practice, be produced dynamically as part of the method (cf. Sect. 5.8.3). The most general scheme involving one-step formulae is a *variable-method variable-step method*. Given a sequence $\{\boldsymbol{\Phi}_n\}$ of one-step formulae, the method proceeds as follows,

$$\left.\begin{aligned} x_{n+1} &= x_n + h_n, \\ \boldsymbol{u}_{n+1} &= \boldsymbol{u}_n + h_n \boldsymbol{\Phi}_n(x_n, \boldsymbol{u}_n; h_n), \end{aligned}\right. \quad n = 0, 1, \ldots, N-1, \tag{5.76}$$

where $x_0 = a$, $\boldsymbol{u}_0 = \boldsymbol{y}_0$. For simplicity, we only consider *single-method* schemes involving a single method $\boldsymbol{\Phi}$, although the extension to variable-method schemes would not cause any essential difficulties (just more writing).

To bring out the analogy between (5.16) and (5.76), we introduce operators R and R_h acting on $C^1[a,b]$ and $\Gamma_h[a,b]$, respectively. These are the *residual operators*

$$(R\boldsymbol{v})(x) := \boldsymbol{v}'(x) - \boldsymbol{f}(x, \boldsymbol{v}(x)), \quad \boldsymbol{v} \in C^1[a,b], \tag{5.77}$$

$$\begin{aligned}(R_h\boldsymbol{v})_n &:= \frac{1}{h_n}(\boldsymbol{v}_{n+1} - \boldsymbol{v}_n) - \boldsymbol{\Phi}(x_n, \boldsymbol{v}_n; h_n), n = 0, 1, \ldots, N-1;\\ \boldsymbol{v} &= \{\boldsymbol{v}_n\} \in \Gamma_h[a,b].\end{aligned} \tag{5.78}$$

(The grid function $\{(R_h\boldsymbol{v})_n\}$ is not defined for $n = N$, but we may arbitrarily set $(R_h\boldsymbol{v})_N = (R_h\boldsymbol{v})_{N-1}$.) Then the problem (5.16) and its discrete analogue (5.76) can be written transparently as

$$R\boldsymbol{y} = \boldsymbol{0} \quad \text{on } [a,b], \quad \boldsymbol{y}(a) = \boldsymbol{y}_0, \tag{5.79}$$

$$R_h\boldsymbol{u} = \boldsymbol{0} \quad \text{on } [a,b], \quad \boldsymbol{u}_0 = \boldsymbol{y}_0. \tag{5.80}$$

Note that the discrete residual operator (5.78) is closely related to the truncation error (5.32) when we apply the operator at a point $(x_n, \boldsymbol{y}(x_n))$ on the exact solution trajectory. Then indeed the reference solution $\boldsymbol{u}(t)$ coincides with the solution $\boldsymbol{y}(t)$, and

$$(R_h\boldsymbol{y})_n = \frac{1}{h_n}[\boldsymbol{y}(x_{n+1}) - \boldsymbol{y}(x_n)] - \boldsymbol{\Phi}(x_n, \boldsymbol{y}(x_n); h_n) = -\boldsymbol{T}(x_n, \boldsymbol{y}(x_n); h_n). \tag{5.81}$$

5.7.1 Stability

Stability is a property of the numerical scheme (5.76) alone and *a priori* has nothing to do with its approximation power. It characterizes the robustness of the scheme with respect to small perturbations. Nevertheless, stability combined with consistency yields convergence of the numerical solution to the true solution.

We define stability in terms of the discrete residual operators R_h in (5.78). As usual, we assume $\boldsymbol{\Phi}(x, \boldsymbol{y}; h)$ to be defined on $[a,b] \times \mathbb{R}^d \times [0, h_0]$, where $h_0 > 0$ is some suitable positive number.

Definition 5.7.1. The method (5.76) is called *stable* on $[a,b]$ if there exists a constant $K > 0$ not depending on h such that for an arbitrary grid h on $[a,b]$, and for arbitrary two grid functions $\boldsymbol{v}, \boldsymbol{w} \in \Gamma_h[a,b]$, there holds

$$\|\boldsymbol{v} - \boldsymbol{w}\|_\infty \le K(\|\boldsymbol{v}_0 - \boldsymbol{w}_0\| + \|R_h\boldsymbol{v} - R_h\boldsymbol{w}\|_\infty), \; \boldsymbol{v}, \boldsymbol{w} \in \Gamma_h[a,b], \tag{5.82}$$

for all h with $|h|$ sufficiently small. In (5.82), the infinity norm for grid functions is the norm defined in (5.75).

We refer to (5.82) as the *stability inequality*. The motivation for it is as follows. Suppose we have two grid functions $\boldsymbol{u}, \boldsymbol{w}$ satisfying

$$R_h \boldsymbol{u} = \boldsymbol{0}, \quad \boldsymbol{u}_0 = \boldsymbol{y}_0, \tag{5.83}$$

$$R_h \boldsymbol{w} = \boldsymbol{\varepsilon}, \quad \boldsymbol{w}_0 = \boldsymbol{y}_0 + \boldsymbol{\eta}_0, \tag{5.84}$$

where $\boldsymbol{\varepsilon} = \{\boldsymbol{\varepsilon}_n\} \in \Gamma_h[a,b]$ is a grid function with small $\|\boldsymbol{\varepsilon}_n\|$, and $\|\boldsymbol{\eta}_0\|$ is also small. We may interpret $\boldsymbol{u} \in \Gamma_h[a,b]$ as the result of applying the numerical scheme (5.76) in infinite precision, whereas $\boldsymbol{w} \in \Gamma_h[a,b]$ could be the solution of (5.76) in floating-point arithmetic. Then if stability holds, we have

$$\|\boldsymbol{u} - \boldsymbol{w}\|_\infty \le K(\|\boldsymbol{\eta}_0\| + \|\boldsymbol{\varepsilon}\|_\infty); \tag{5.85}$$

that is, the global change in $\boldsymbol{u}$ is of the same order of magnitude as the local residual errors $\{\boldsymbol{\varepsilon}_n\}$ and initial error $\boldsymbol{\eta}_0$. It should be appreciated, however, that the first equation in (5.84) says $\boldsymbol{w}_{n+1} - \boldsymbol{w}_n - h_n \boldsymbol{\Phi}(x_n, \boldsymbol{w}_n; h_n) = h_n \boldsymbol{\varepsilon}_n$, meaning that "rounding errors" must go to zero as $|h| \to 0$.

Interestingly enough, a Lipschitz condition on $\boldsymbol{\Phi}$ is all that is required for stability.

Theorem 5.7.1. *If $\boldsymbol{\Phi}(x, \boldsymbol{y}; h)$ satisfies a Lipschitz condition with respect to the y-variables,*

$$\|\boldsymbol{\Phi}(x, \boldsymbol{y}; h) - \boldsymbol{\Phi}(x, \boldsymbol{y}^*; h)\| \le M \|\boldsymbol{y} - \boldsymbol{y}^*\| \ \text{ on } [a,b] \times \mathbb{R}^d \times [0, h_0], \tag{5.86}$$

then the method (5.76) is stable.

We precede the proof with the following useful lemma.

Lemma 5.7.1. *Let $\{e_n\}$ be a sequence of numbers $e_n \in \mathbb{R}$ satisfying*

$$e_{n+1} \le a_n e_n + b_n, \quad n = 0, 1, \ldots, N-1, \tag{5.87}$$

where $a_n > 0$ and $b_n \in \mathbb{R}$. Then

$$e_n \le E_n, \ E_n = \left(\prod_{k=0}^{n-1} a_k\right) e_0 + \sum_{k=0}^{n-1} \left(\prod_{\ell=k+1}^{n-1} a_\ell\right) b_k, \ \ n = 0, 1, \ldots, N. \tag{5.88}$$

We adopt here the usual convention that an empty product has the value 1, and an empty sum the value 0.

Proof of Lemma 5.7.1. It is readily verified that

$$E_{n+1} = a_n E_n + b_n, \ \ n = 0, 1, \ldots, N-1; \quad E_0 = e_0.$$

Subtracting this from the inequality in (5.87), we get

$$e_{n+1} - E_{n+1} \le a_n(e_n - E_n), \quad n = 0, 1, \dots, N-1.$$

Now, $e_0 - E_0 = 0$, so that $e_1 - E_1 \le 0$. Therefore, $e_2 - E_2 \le a_1(e_1 - E_1)$, implying $e_2 - E_2 \le 0$ since $a_1 > 0$. In the same way, by induction, $e_n - E_n \le 0$. □

Proof of Theorem 5.7.1. Let $h = \{h_n\}$ be an arbitrary grid on $[a, b]$, and $\boldsymbol{v}, \boldsymbol{w} \in \Gamma_h[a, b]$ two arbitrary (vector-valued) grid functions. By definition of R_h, we can write

$$\boldsymbol{v}_{n+1} = \boldsymbol{v}_n + h_n \boldsymbol{\Phi}(x_n, \boldsymbol{v}_n; h_n) + h_n(R_h\boldsymbol{v})_n, \quad n = 0, 1, \dots, N-1,$$

and similarly for $\boldsymbol{w}_{n+1}$. Subtraction then gives

$$\begin{aligned} \boldsymbol{v}_{n+1} - \boldsymbol{w}_{n+1} =& \boldsymbol{v}_n - \boldsymbol{w}_n + h_n[\boldsymbol{\Phi}(x_n, \boldsymbol{v}_n; h_n) - \boldsymbol{\Phi}(x_n, \boldsymbol{w}_n; h_n)] \\ &+ h_n[(R_h\boldsymbol{v})_n - (R_h\boldsymbol{w})_n], n = 0, 1, \dots, N-1. \end{aligned} \tag{5.89}$$

Now define

$$e_n = \|\boldsymbol{v}_n - \boldsymbol{w}_n\|, \;\; d_n = \|(R_h\boldsymbol{v})_n - (R_h\boldsymbol{w})_n\|, \;\; \delta = \max_n d_n. \tag{5.90}$$

Then using the triangle inequality in (5.89) and the Lipschitz condition (5.86) for $\boldsymbol{\Phi}$, we obtain

$$e_{n+1} \le (1 + h_n M)e_n + h_n\delta, \quad n = 0, 1, \dots, N-1. \tag{5.91}$$

This is inequality (5.87) with $a_n = 1 + h_n M, b_n = h_n\delta$. Since for $k = 0, 1, \dots, n-1$ and $n \le N$, we have

$$\begin{aligned} \prod_{\ell=k+1}^{n-1} a_\ell \le \prod_{\ell=0}^{N-1} a_\ell = \prod_{\ell=0}^{N-1}(1 + h_\ell M) \le \prod_{\ell=0}^{N-1} \mathrm{e}^{h_\ell M} \\ = \mathrm{e}^{(h_0 + h_1 + \dots + h_{N-1})M} = \mathrm{e}^{(b-a)M}, \end{aligned}$$

where $1 + x \le \mathrm{e}^x$ has been used in the second inequality, we obtain from Lemma 5.7.1 that

$$\begin{aligned} e_n &\le \mathrm{e}^{(b-a)M} e_0 + \mathrm{e}^{(b-a)M} \sum_{k=0}^{n-1} h_k\delta \\ &\le \mathrm{e}^{(b-a)M}(e_0 + (b-a)\delta), \quad n = 0, 1, \dots, N-1. \end{aligned}$$

Taking the maximum over n and recalling the definition of e_n and δ, we get

$$\|\boldsymbol{v} - \boldsymbol{w}\|_\infty \le \mathrm{e}^{(b-a)M}(\|\boldsymbol{v}_0 - \boldsymbol{w}_0\| + (b-a)\|R_h\boldsymbol{v} - R_h\boldsymbol{w}\|_\infty),$$

which is (5.82) with $K = \mathrm{e}^{(b-a)M} \max\{1, b-a\}$. □

We have actually proved stability for *all* $|h| \le h_0$, not only for $|h|$ sufficiently small. The proof is virtually the same for a variable-method algorithm involving a family of one-step formulae $\{\boldsymbol{\Phi}_n\}$, if we assume a Lipschitz condition for each $\boldsymbol{\Phi}_n$ with a constant M independent of n.

All one-step methods used in practice satisfy a Lipschitz condition if $\boldsymbol{f}$ does, and the constant M for $\boldsymbol{\Phi}$ can be expressed in terms of the Lipschitz constant L for $\boldsymbol{f}$. This is obvious for Euler's method, and not difficult to prove for others (see Ex. 13). It is useful to note that $\boldsymbol{\Phi}$ need *not* be continuous in x; piecewise continuity suffices, as long as (5.86) holds for all $x \in [a, b]$, taking one-sided limits at points of discontinuity.

For later use, we state another application of Lemma 5.7.1, relative to a grid function $\boldsymbol{v} \in \Gamma_h[a, b]$ satisfying

$$\boldsymbol{v}_{n+1} = \boldsymbol{v}_n + h_n(\boldsymbol{A}_n\boldsymbol{v}_n + \boldsymbol{b}_n), \quad n = 0, 1, \dots, N-1, \tag{5.92}$$

where $\boldsymbol{A}_n \in \mathbb{R}^{d\times d}$, $\boldsymbol{b}_n \in \mathbb{R}^d$, and $h = \{h_n\}$ is an arbitrary grid on $[a, b]$.

Lemma 5.7.2. *Suppose in (5.92) that*

$$\|\boldsymbol{A}_n\| \le M, \ \|\boldsymbol{b}_n\| \le \delta, \quad n = 0, 1, \dots, N-1, \tag{5.93}$$

where the constants M, δ do not depend on h. Then there exists a constant $K > 0$ independent of h, but depending on $\|\boldsymbol{v}_0\|$, such that

$$\|\boldsymbol{v}\|_\infty \le K. \tag{5.94}$$

Proof. The lemma follows at once by observing that

$$\|\boldsymbol{v}_{n+1}\| \le (1 + h_n M)\|\boldsymbol{v}_n\| + h_n\delta, \quad n = 0, 1, \dots, N-1,$$

which is precisely the inequality (5.91) in the proof of Theorem 5.7.1, hence

$$\|\boldsymbol{v}_n\| \le \mathrm{e}^{(b-a)M}\{\|\boldsymbol{v}_0\| + (b-a)\delta\}. \tag{5.95}$$

□

5.7.2 Convergence

Stability is a rather powerful concept; it implies almost immediately convergence, and is also instrumental in deriving asymptotic global error estimates. We begin by defining precisely what we mean by convergence.

Definition 5.7.2. Let $a = x_0 < x_1 < x_2 < \cdots < x_N = b$ be a grid on $[a, b]$ with grid length $|h| = \max\limits_{1\le n\le N}(x_n - x_{n-1})$. Let $\boldsymbol{u} = \{\boldsymbol{u}_n\}$ be the grid function defined by applying the method (5.76) (with $\boldsymbol{\Phi}_n = \boldsymbol{\Phi}$) on $[a, b]$, and $\boldsymbol{y} = \{\boldsymbol{y}_n\}$ the grid function induced by the exact solution of the initial value problem (5.16). The method (5.76) is said to *converge* on $[a, b]$ if there holds

$$\|\boldsymbol{u} - \boldsymbol{y}\|_\infty \to 0 \quad \text{as } |h| \to 0. \tag{5.96}$$

Theorem 5.7.2. *If the method (5.76) is consistent and stable on* $[a, b]$*, then it converges. Moreover, if* $\boldsymbol{\Phi}$ *has order p, then*

$$\|\boldsymbol{u} - \boldsymbol{y}\|_\infty = O(|h|^p) \quad \textit{as } |h| \to 0. \tag{5.97}$$

Proof. By the stability inequality (5.82) applied to the grid functions $\boldsymbol{v} = \boldsymbol{u}$ and $\boldsymbol{w} = \boldsymbol{y}$ of Definition 5.7.2, we have, for $|h|$ sufficiently small,

$$\begin{aligned} \|\boldsymbol{u} - \boldsymbol{y}\|_\infty &\le K(\|\boldsymbol{u}_0 - \boldsymbol{y}(x_0)\| + \|R_h\boldsymbol{u} - R_h\boldsymbol{y}\|_\infty) \\ &= K\|R_h\boldsymbol{y}\|_\infty, \end{aligned} \tag{5.98}$$

since $\boldsymbol{u}_0 = \boldsymbol{y}(x_0)$ and $R_h\boldsymbol{u} = \boldsymbol{0}$ by (5.76). But, by (5.81),

$$\|R_h\boldsymbol{y}\|_\infty = \|\boldsymbol{T}(\cdot, \boldsymbol{y}; h)\|_\infty, \tag{5.99}$$

where $\boldsymbol{T}$ is the truncation error of the method $\boldsymbol{\Phi}$. By definition of consistency,

$$\|\boldsymbol{T}(\cdot, \boldsymbol{y}; h)\|_\infty \to 0 \quad \text{as } |h| \to 0,$$

which proves the first part of the theorem. The second part follows immediately from (5.98) and (5.99), since order p means, by definition, that

$$\|\boldsymbol{T}(\cdot, \boldsymbol{y}; h)\|_\infty = O(|h|^p) \quad \text{as } |h| \to 0. \tag{5.100}$$

□

Since, as we already observed, practically all one-step methods are stable and of order $p \ge 1$ (under reasonable smoothness assumptions on $\boldsymbol{f}$), it follows that they are all convergent as well.

5.7.3 Asymptotics of Global Error

Just as the principal error function describes the leading contribution to the local truncation error, it is of interest to identify the leading term in the global error $\boldsymbol{u}_n - \boldsymbol{y}(x_n)$. To simplify matters, we assume a constant grid length h, although it would not be difficult to deal with variable grid lengths of the form $h_n = \vartheta(x_n)h$, where $\vartheta(x)$ is piecewise continuous and $0 < \vartheta(x) \le \Theta$ for $a \le x \le b$. Thus, we consider our one-step method to have the form

$$\begin{aligned}
&x_{n+1} = x_n + h, \\
&\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + h\boldsymbol{\Phi}(x_n, \boldsymbol{u}_n; h),\ n = 0, 1, \dots, N-1, \\
&x_0 = a,\ \ \boldsymbol{u}_0 = \boldsymbol{y}_0,
\end{aligned} \tag{5.101}$$

defining a grid function $\boldsymbol{u} = \{\boldsymbol{u}_n\}$ on a uniform grid over $[a, b]$. We are interested in the asymptotic behavior of $\boldsymbol{u}_n - \boldsymbol{y}(x_n)$ as $h \to 0$, where $\boldsymbol{y}(x)$ is the exact solution of the initial value problem

$$\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}x} = \boldsymbol{f}(x, \boldsymbol{y}),\ a \le x \le b;\ \boldsymbol{y}(a) = \boldsymbol{y}_0. \tag{5.102}$$

Theorem 5.7.3. *Assume that*

(1) $\boldsymbol{\Phi}(x, \boldsymbol{y}; h) \in C^2$ *on* $[a, b] \times \mathbb{R}^d \times [0, h_0]$*;*
(2) $\boldsymbol{\Phi}$ *is a method of order* $p \ge 1$ *admitting a principal error function* $\boldsymbol{\tau}(x, \boldsymbol{y}) \in C$ *on* $[a, b] \times \mathbb{R}^d$*;*
(3) $\boldsymbol{e}(x)$ *is the solution of the linear initial value problem*

$$\begin{aligned}
&\frac{\mathrm{d}\boldsymbol{e}}{\mathrm{d}x} = \boldsymbol{f}_{\boldsymbol{y}}(x, \boldsymbol{y}(x))\boldsymbol{e} + \boldsymbol{\tau}(x, \boldsymbol{y}(x)), \quad a \le x \le b, \\
&\boldsymbol{e}(a) = \boldsymbol{0}.
\end{aligned} \tag{5.103}$$

Then, for $n = 0, 1, \dots, N$*,*

$$\boldsymbol{u}_n - \boldsymbol{y}(x_n) = \boldsymbol{e}(x_n)h^p + O(h^{p+1})\ \ \textit{as}\ \ h \to 0. \tag{5.104}$$

Before we prove the theorem, we make the following remarks.

1. The precise meaning of (5.104) is
$$\|\boldsymbol{u} - \boldsymbol{y} - h^p\boldsymbol{e}\|_\infty = O(h^{p+1})\ \text{ as }\ h \to 0, \tag{5.105}$$
where $\boldsymbol{u}$, $\boldsymbol{y}$, $\boldsymbol{e}$ are the grid functions $\boldsymbol{u} = \{\boldsymbol{u}_n\}$, $\boldsymbol{y} = \{\boldsymbol{y}(x_n)\}$, $\boldsymbol{e} = \{\boldsymbol{e}(x_n)\}$ and $\|\cdot\|_\infty$ is the norm defined in (5.75).
2. Since by consistency $\boldsymbol{\Phi}(x, \boldsymbol{y}; 0) = \boldsymbol{f}(x, \boldsymbol{y})$, assumption (1) implies $\boldsymbol{f} \in C^2$ on $[a, b] \times \mathbb{R}^d$, which is more than enough to guarantee the existence and uniqueness of the solution $\boldsymbol{e}(x)$ of (5.103) on the whole interval $[a, b]$.
3. The fact that some, but not all, components of $\boldsymbol{\tau}(x, \boldsymbol{y})$ may vanish identically does *not* imply that the corresponding components of $\boldsymbol{e}(x)$ also vanish, since (5.103) is a *coupled* system of differential equations.

Proof of Theorem 5.7.3. We begin with an auxiliary computation, an estimate for

$$\boldsymbol{\Phi}(x_n, \boldsymbol{u}_n; h) - \boldsymbol{\Phi}(x_n, \boldsymbol{y}(x_n); h). \tag{5.106}$$

By Taylor's theorem (for functions of several variables), applied to the ith component of (5.106), we have

$$\Phi^i(x_n,\boldsymbol{u}_n;h)-\Phi^i(x_n,\boldsymbol{y}(x_n);h)=\sum_{j=1}^{d}\Phi^i_{y^j}(x_n,\boldsymbol{y}(x_n);h)[u_n^j-y^j(x_n)]$$
$$+\frac{1}{2}\sum_{j,k=1}^{d}\Phi^i_{y^j y^k}(x_n,\overline{\boldsymbol{u}}_n;h)[u_n^j-y^j(x_n)][u_n^k-y^k(x_n)], \tag{5.107}$$

where $\overline{\boldsymbol{u}}_n$ is on the line segment connecting $\boldsymbol{u}_n$ and $\boldsymbol{y}(x_n)$. Using Taylor's theorem once more, in the variable h, we can write

$$\Phi^i_{y^j}(x_n,\boldsymbol{y}(x_n);h)=\Phi^i_{y^j}(x_n,\boldsymbol{y}(x_n);0)+h\Phi^i_{y^j h}(x_n,\boldsymbol{y}(x_n);\overline{h}),$$

where $0<\overline{h}<h$. Since by consistency $\boldsymbol{\Phi}(x,\boldsymbol{y};0)\equiv\boldsymbol{f}(x,\boldsymbol{y})$ on $[a,b]\times\mathbb{R}^d$, we have

$$\Phi^i_{y^j}(x,\boldsymbol{y};0)=f^i_{y^j}(x,\boldsymbol{y}),\quad x\in[a,b],\ \boldsymbol{y}\in\mathbb{R}^d,$$

and assumption (1) allows us to write

$$\Phi^i_{y^j}(x_n,\boldsymbol{y}(x_n);h)=f^i_{y^j}(x_n,\boldsymbol{y}(x_n))+O(h),\quad h\to 0. \tag{5.108}$$

Now observing that $\boldsymbol{u}_n-\boldsymbol{y}(x_n)=O(h^p)$ by virtue of Theorem 5.7.2, and using (5.108) in (5.107), we get, again by assumption (1),

$$\Phi^i(x_n,\boldsymbol{u}_n;h)-\Phi^i(x_n,\boldsymbol{y}(x_n);h)$$
$$=\sum_{j=1}^{d}f^i_{y^j}(x_n,\boldsymbol{y}(x_n))[u_n^j-y^j(x_n)]+O(h^{p+1})+O(h^{2p}).$$

But $O(h^{2p})$ is also of order $O(h^{p+1})$, since $p\geq 1$. Thus, in vector notation,

$$\boldsymbol{\Phi}(x_n,\boldsymbol{u}_n;h)-\boldsymbol{\Phi}(x_n,\boldsymbol{y}(x_n);h)=\boldsymbol{f}_{\boldsymbol{y}}(x_n,\boldsymbol{y}(x_n))[\boldsymbol{u}_n-\boldsymbol{y}(x_n)]+O(h^{p+1}). \tag{5.109}$$

Now, to highlight the leading term in the global error, we define the grid function $\boldsymbol{r}=\{\boldsymbol{r}_n\}$ by

$$\boldsymbol{r}=h^{-p}(\boldsymbol{u}-\boldsymbol{y}). \tag{5.110}$$

Then

$$\begin{aligned}\frac{1}{h}(\boldsymbol{r}_{n+1}-\boldsymbol{r}_n) &= \frac{1}{h}[h^{-p}(\boldsymbol{u}_{n+1}-\boldsymbol{y}(x_{n+1}))-h^{-p}(\boldsymbol{u}_n-\boldsymbol{y}(x_n))]\\ &= h^{-p}\left[\frac{1}{h}(\boldsymbol{u}_{n+1}-\boldsymbol{u}_n)-\frac{1}{h}(\boldsymbol{y}(x_{n+1})-\boldsymbol{y}(x_n))\right]\\ &= h^{-p}[\boldsymbol{\Phi}(x_n,\boldsymbol{u}_n;h)-\{\boldsymbol{\Phi}(x_n,\boldsymbol{y}(x_n);h)-\boldsymbol{T}(x_n,\boldsymbol{y}(x_n);h)\}],\end{aligned}$$

where we have used (5.101) and the relation (5.81) for the truncation error $\boldsymbol{T}$. Therefore, expressing $\boldsymbol{T}$ in terms of the principal error function $\boldsymbol{\tau}$, we get

$$\begin{aligned}\frac{1}{h}(\boldsymbol{r}_{n+1}-\boldsymbol{r}_n) = h^{-p}[&\boldsymbol{\Phi}(x_n,\boldsymbol{u}_n;h)-\boldsymbol{\Phi}(x_n,\boldsymbol{y}(x_n);h)\\ &+\boldsymbol{\tau}(x_n,\boldsymbol{y}(x_n))h^p+O(h^{p+1})].\end{aligned}$$

For the first two terms in brackets, we use (5.109) and the definition of $\boldsymbol{r}$ in (5.110) to obtain

$$\begin{aligned}\frac{1}{h}(\boldsymbol{r}_{n+1}-\boldsymbol{r}_n) &= \boldsymbol{f}_{\boldsymbol{y}}(x_n,\boldsymbol{y}(x_n))\boldsymbol{r}_n+\boldsymbol{\tau}(x_n,\boldsymbol{y}(x_n))+O(h),\\ &\qquad n=0,1,\dots,N-1,\\ \boldsymbol{r}_0 &= \boldsymbol{0}.\end{aligned}\tag{5.111}$$

Now letting

$$\boldsymbol{g}(x,\boldsymbol{y}) := \boldsymbol{f}_{\boldsymbol{y}}(x,\boldsymbol{y}(x))\boldsymbol{y}+\boldsymbol{\tau}(x,\boldsymbol{y}(x)),\tag{5.112}$$

we can interpret (5.111) by writing

$$\left(R_h^{\text{Euler},\boldsymbol{g}}\boldsymbol{r}\right)_n = \boldsymbol{\varepsilon}_n \ \ (n=0,1,\dots,N-1),\ \ \boldsymbol{\varepsilon}_n = O(h),$$

where $R_h^{\text{Euler},\boldsymbol{g}}$ is the discrete residual operator (5.78) that goes with Euler's method applied to $\boldsymbol{e}'=\boldsymbol{g}(x,\boldsymbol{e})$, $\boldsymbol{e}(a)=\boldsymbol{0}$. Since Euler's method is stable on $[a,b]$ and $\boldsymbol{g}$ (being linear in $\boldsymbol{y}$) certainly satisfies a uniform Lipschitz condition, we have by the stability inequality (5.82)

$$\|\boldsymbol{r}-\boldsymbol{e}\|_\infty = O(h),$$

and hence, by (5.110),

$$\|\boldsymbol{u}-\boldsymbol{y}-h^p\boldsymbol{e}\|_\infty = O(h^{p+1}),$$

as was to be shown. □

5.8 Error Monitoring and Step Control

Most production codes currently available for solving ODEs monitor local truncation errors and control the step length on the basis of estimates for these errors. Here, we attempt to monitor the global error, at least asymptotically, by implementing the asymptotic result of Theorem 5.7.3. This necessitates the evaluation of the Jacobian matrix $\boldsymbol{f}_{\boldsymbol{y}}(x, \boldsymbol{y})$ along or near the solution trajectory; but this is only natural, since $\boldsymbol{f}_{\boldsymbol{y}}$, in a first approximation, governs the effect of perturbations via the variational differential equation (5.103). This equation is driven by the principal error function evaluated along the trajectory, so that estimates of local truncation errors (more precisely, of the principal error function) are needed also in this approach. For simplicity, we again assume constant grid length.

5.8.1 Estimation of Global Error

The idea of our estimation is to integrate the "variational equation" (5.103) along with the main equation (5.102). Since we need $\boldsymbol{e}(x_n)$ in (5.104) only to within an accuracy of $O(h)$ (any $O(h)$ error term in $\boldsymbol{e}(x_n)$, multiplied by h^p, being absorbed by the $O(h^{p+1})$ term), we can use Euler's method for that purpose, which will provide the desired approximation $\boldsymbol{v}_n \approx \boldsymbol{e}(x_n)$.

Theorem 5.8.1. *Assume that*

(1) $\boldsymbol{\Phi}(x, \boldsymbol{y}; h) \in C^2$ *on* $[a, b] \times \mathbb{R}^d \times [0, h_0]$*;*
(2) $\boldsymbol{\Phi}$ *is a method of order* $p \geq 1$ *admitting a principal error function* $\boldsymbol{\tau}(x, \boldsymbol{y}) \in C^1$ *on* $[a, b] \times \mathbb{R}^d$*;*
(3) *an estimate* $\boldsymbol{r}(x, \boldsymbol{y}; h)$ *is available for the principal error function that satisfies*

$$\boldsymbol{r}(x, \boldsymbol{y}; h) = \boldsymbol{\tau}(x, \boldsymbol{y}) + O(h), \quad h \to 0, \tag{5.113}$$

uniformly on $[a, b] \times \mathbb{R}^d$*;*
(4) *along with the grid function* $\boldsymbol{u} = \{\boldsymbol{u}_n\}$ *we generate the grid function* $\boldsymbol{v} = \{\boldsymbol{v}_n\}$ *in the following manner,*

$$\begin{aligned} x_{n+1} &= x_n + h, \\ \boldsymbol{u}_{n+1} &= \boldsymbol{u}_n + h\boldsymbol{\Phi}(x_n, \boldsymbol{u}_n; h), \\ \boldsymbol{v}_{n+1} &= \boldsymbol{v}_n + h[\boldsymbol{f}_{\boldsymbol{y}}(x_n, \boldsymbol{u}_n)\boldsymbol{v}_n + \boldsymbol{r}(x_n, \boldsymbol{u}_n; h)], \\ x_0 &= a, \boldsymbol{u}_0 = \boldsymbol{y}_0, \boldsymbol{v}_0 = \boldsymbol{0}. \end{aligned} \tag{5.114}$$

Then, for $n = 0, 1, \ldots, N$*,*

$$\boldsymbol{u}_n - \boldsymbol{y}(x_n) = \boldsymbol{v}_n h^p + O(h^{p+1}) \textit{ as } h \to 0. \tag{5.115}$$

Proof. The proof consists of establishing the following estimates,

$$\boldsymbol{f}_{\boldsymbol{y}}(x_n, \boldsymbol{u}_n) = \boldsymbol{f}_{\boldsymbol{y}}(x_n, \boldsymbol{y}(x_n)) + O(h), \tag{5.116}$$

$$\boldsymbol{r}(x_n, \boldsymbol{u}_n; h) = \boldsymbol{\tau}(x_n, \boldsymbol{y}(x_n)) + O(h). \tag{5.117}$$

Once this has been done, we can argue as follows. Let (cf. (5.114))

$$\boldsymbol{g}(x, \boldsymbol{y}) := \boldsymbol{f}_{\boldsymbol{y}}(x, \boldsymbol{y}(x))\boldsymbol{y} + \boldsymbol{\tau}(x, \boldsymbol{y}(x)). \tag{5.118}$$

The equation for $\boldsymbol{v}_{n+1}$ in (5.114) has the form

$$\boldsymbol{v}_{n+1} = \boldsymbol{v}_n + h(\boldsymbol{A}_n \boldsymbol{v}_n + \boldsymbol{b}_n),$$

where $\boldsymbol{A}_n$ are bounded matrices and $\boldsymbol{b}_n$ bounded vectors. By Lemma 5.7.2, we have boundedness of $\boldsymbol{v}_n$,

$$\boldsymbol{v}_n = O(1), \quad h \to 0. \tag{5.119}$$

Substituting (5.116) and (5.117) into the equation for $\boldsymbol{v}_{n+1}$, and noting (5.119), we obtain

$$\begin{aligned}\boldsymbol{v}_{n+1} &= \boldsymbol{v}_n + h[\boldsymbol{f}_{\boldsymbol{y}}(x_n, \boldsymbol{y}(x_n))\boldsymbol{v}_n + \boldsymbol{\tau}(x_n, \boldsymbol{y}(x_n)) + O(h)] \\ &= \boldsymbol{v}_n + h\boldsymbol{g}(x_n, \boldsymbol{v}_n) + O(h^2).\end{aligned}$$

Thus, in the notation used in the proof of Theorem 5.7.3,

$$(R_h^{\text{Euler},\boldsymbol{g}} \boldsymbol{v})_n = O(h), \;\; \boldsymbol{v}_0 = \boldsymbol{0}.$$

Since Euler's method is stable, we conclude

$$\boldsymbol{v}_n - \boldsymbol{e}(x_n) = O(h),$$

where $\boldsymbol{e}(x)$ is, as before, the solution of $\boldsymbol{e}' = \boldsymbol{g}(x, \boldsymbol{e})$, $\boldsymbol{e}(a) = \boldsymbol{0}$. Therefore, by (5.104),

$$\boldsymbol{u}_n - \boldsymbol{y}(x_n) = \boldsymbol{e}(x_n)h^p + O(h^{p+1}) = \boldsymbol{v}_n h^p + O(h^{p+1}),$$

as was to be shown.

It remains to prove (5.116) and (5.117). From assumption (1) we note, first of all, that $\boldsymbol{f}(x, \boldsymbol{y}) \in C^2$ on $[a, b] \times \mathbb{R}^d$, since by consistency $\boldsymbol{f}(x, \boldsymbol{y}) = \boldsymbol{\Phi}(x, \boldsymbol{y}; 0)$. By virtue of $\boldsymbol{u}_n = \boldsymbol{y}(x_n) + O(h^p)$ (cf. Theorem 5.7.2), we therefore have

$$\boldsymbol{f}_{\boldsymbol{y}}(x_n, \boldsymbol{u}_n) = \boldsymbol{f}_{\boldsymbol{y}}(x_n, \boldsymbol{y}(x_n)) + O(h^p),$$

which implies (5.116), since $p \geq 1$.

Next, since $\boldsymbol{\tau}(x, \boldsymbol{y}) \in C^1$ by assumption (2), we have

$$\begin{aligned}\boldsymbol{\tau}(x_n, \boldsymbol{u}_n) &= \boldsymbol{\tau}(x_n, \boldsymbol{y}(x_n)) + \boldsymbol{\tau}_{\boldsymbol{y}}(x_n, \overline{\boldsymbol{u}}_n)(\boldsymbol{u}_n - \boldsymbol{y}(x_n)) \\ &= \boldsymbol{\tau}(x_n, \boldsymbol{y}(x_n)) + O(h^p),\end{aligned}$$

so that by assumption (3),

$$\boldsymbol{r}(x_n, \boldsymbol{u}_n; h) = \boldsymbol{\tau}(x_n, \boldsymbol{u}_n) + O(h) = \boldsymbol{\tau}(x_n, \boldsymbol{y}(x_n)) + O(h^p) + O(h),$$

from which (5.117) follows at once. This completes the proof of Theorem 5.8.1. □

5.8.2 Truncation Error Estimates

In order to apply Theorem 5.8.1, we need estimates $\boldsymbol{r}(x, \boldsymbol{y}; h)$ of the principal error function $\boldsymbol{\tau}(x, \boldsymbol{y})$ which are $O(h)$ accurate. A number of them, in increasing order of efficiency, are now described.

1. *Local Richardson extrapolation to zero*: This works for any one-step method $\boldsymbol{\Phi}$, but is usually considered to be too expensive. If $\boldsymbol{\Phi}$ has order p, the procedure is as follows.

$$\begin{aligned}\boldsymbol{y}_h &= \boldsymbol{y} + h\boldsymbol{\Phi}(x, \boldsymbol{y}; h), \\ \boldsymbol{y}_{h/2} &= \boldsymbol{y} + \frac{1}{2} h \boldsymbol{\Phi}\left(x, \boldsymbol{y}; \frac{1}{2} h\right), \\ \boldsymbol{y}_h^* &= \boldsymbol{y}_{h/2} + \frac{1}{2} h \boldsymbol{\Phi}\left(x + \frac{1}{2} h, \boldsymbol{y}_{h/2}; \frac{1}{2} h\right), \\ \boldsymbol{r}(x, \boldsymbol{y}; h) &= \frac{1}{1 - 2^{-p}} \frac{1}{h^{p+1}} (\boldsymbol{y}_h - \boldsymbol{y}_h^*).\end{aligned} \tag{5.120}$$

Note that $\boldsymbol{y}_h^*$ is the result of applying $\boldsymbol{\Phi}$ over two consecutive steps of length $\frac{1}{2} h$ each, whereas $\boldsymbol{y}_h$ is the result of one application over the whole step of length h.

We now verify that $\boldsymbol{r}(x, \boldsymbol{y}; h)$ in (5.120) is an acceptable estimator. To do this, we need to assume that $\boldsymbol{\tau}(x, \boldsymbol{y}) \in C^1$ on $[a, b] \times \mathbb{R}^d$. In terms of the reference solution $\boldsymbol{u}(t)$ through $(x, \boldsymbol{y})$, we have (cf. (5.32) and (5.37))

$$\boldsymbol{\Phi}(x, \boldsymbol{y}; h) = \frac{1}{h}[\boldsymbol{u}(x + h) - \boldsymbol{u}(x)] + \boldsymbol{\tau}(x, \boldsymbol{y}) h^p + O(h^{p+1}). \tag{5.121}$$

Furthermore,

$$\begin{aligned}\frac{1}{h}(\boldsymbol{y}_h - \boldsymbol{y}_h^*) &= \frac{1}{h}(\boldsymbol{y} - \boldsymbol{y}_{h/2}) + \boldsymbol{\Phi}(x, \boldsymbol{y}; h) - \frac{1}{2}\boldsymbol{\Phi}\left(x + \frac{1}{2} h, \boldsymbol{y}_{h/2}; \frac{1}{2} h\right) \\ &= \boldsymbol{\Phi}(x, \boldsymbol{y}; h) - \frac{1}{2}\boldsymbol{\Phi}\left(x, \boldsymbol{y}; \frac{1}{2} h\right) - \frac{1}{2}\boldsymbol{\Phi}\left(x + \frac{1}{2} h, \boldsymbol{y}_{h/2}; \frac{1}{2} h\right).\end{aligned}$$

Applying (5.121) to each of the three terms on the right, we find

$$\begin{aligned}\frac{1}{h}(\boldsymbol{y}_h-\boldsymbol{y}_h^*)=\frac{1}{h}\,[\boldsymbol{u}(x+h)-\boldsymbol{u}(x)]+\boldsymbol{\tau}(x,\boldsymbol{y})h^p+O(h^{p+1})\\ -\frac{1}{2}\frac{1}{h/2}\left[\boldsymbol{u}\left(x+\frac{1}{2}h\right)-\boldsymbol{u}(x)\right]-\frac{1}{2}\boldsymbol{\tau}(x,\boldsymbol{y})\left(\frac{1}{2}h\right)^p+O(h^{p+1})\\ -\frac{1}{2}\frac{1}{h/2}\left[\boldsymbol{u}(x+h)-\boldsymbol{u}\left(x+\frac{1}{2}h\right)\right]\\ -\frac{1}{2}\boldsymbol{\tau}\left(x+\frac{1}{2}h,\boldsymbol{y}+O(h)\right)\left(\frac{1}{2}h\right)^p\\ +O(h^{p+1})=\boldsymbol{\tau}(x,\boldsymbol{y})(1-2^{-p})h^p+O(h^{p+1}).\end{aligned}$$

Consequently,

$$\frac{1}{1-2^{-p}}\,\frac{1}{h}\,(\boldsymbol{y}_h-\boldsymbol{y}_h^*)=\boldsymbol{\tau}(x,\boldsymbol{y})h^p+O(h^{p+1}) \tag{5.122}$$

as required.

Subtracting (5.122) from (5.121) shows, incidentally, that

$$\boldsymbol{\Phi}^*(x,\boldsymbol{y};h):=\boldsymbol{\Phi}(x,\boldsymbol{y};h)-\frac{1}{1-2^{-p}}\,\frac{1}{h}\,(\boldsymbol{y}_h-\boldsymbol{y}_h^*) \tag{5.123}$$

defines a one-step method of order $p+1$.

The procedure in (5.120) is rather expensive. For a fourth-order Runge–Kutta process, it requires a total of 11 evaluations of $\boldsymbol{f}$ per step, almost three times the effort for a single Runge–Kutta step. Therefore, Richardson extrapolation is normally used only after every two steps of $\boldsymbol{\Phi}$; that is, one proceeds according to

$$\begin{aligned}\boldsymbol{y}_h&=\boldsymbol{y}+h\boldsymbol{\Phi}(x,\boldsymbol{y};h),\\ \boldsymbol{y}_{2h}^*&=\boldsymbol{y}_h+h\boldsymbol{\Phi}(x+h,\boldsymbol{y}_h;h),\\ \boldsymbol{y}_{2h}&=\boldsymbol{y}+2h\boldsymbol{\Phi}(x,\boldsymbol{y};2h).\end{aligned} \tag{5.124}$$

Then (5.122) gives

$$\frac{1}{2(2^p-1)}\,\frac{1}{h^{p+1}}\,(\boldsymbol{y}_{2h}-\boldsymbol{y}_{2h}^*)=\boldsymbol{\tau}(x,\boldsymbol{y})+O(h), \tag{5.125}$$

so that the expression on the left is an acceptable estimator $\boldsymbol{r}(x,\boldsymbol{y};h)$. If the two steps in (5.124) yield acceptable accuracy (cf. Sect. 5.8.3), then, again for a fourth-order Runge–Kutta process, the procedure requires only three additional evaluations of $\boldsymbol{f}$, since $\boldsymbol{y}_h$ and $\boldsymbol{y}_{2h}^*$ would have to be computed anyhow. We show, however, that there are still more efficient schemes.

2. *Embedded methods*: The basic idea of this approach is very simple: if the given method $\boldsymbol{\Phi}$ has order p, take any one-step method $\boldsymbol{\Phi}^*$ of order $p^* = p + 1$ and define

$$\boldsymbol{r}(x, \boldsymbol{y}; h) = \frac{1}{h^p}\,[\boldsymbol{\Phi}(x, \boldsymbol{y}; h) - \boldsymbol{\Phi}^*(x, \boldsymbol{y}; h)]. \tag{5.126}$$

This is indeed an acceptable estimator, as follows by subtracting the two relations

$$\boldsymbol{\Phi}(x, \boldsymbol{y}; h) - \frac{1}{h}\,[\boldsymbol{u}(x + h) - \boldsymbol{u}(x)] = \boldsymbol{\tau}(x, \boldsymbol{y})h^p + O(h^{p+1}),$$
$$\boldsymbol{\Phi}^*(x, \boldsymbol{y}; h) - \frac{1}{h}\,[\boldsymbol{u}(x + h) - \boldsymbol{u}(x)] = O(h^{p+1})$$

and dividing the result by h^p.

The tricky part is making this procedure efficient. Following an idea of Fehlberg, one can try to do this by embedding one Runge–Kutta process (of order p) into another (of order $p + 1$). Specifically, let $\boldsymbol{\Phi}$ be some explicit r-stage Runge–Kutta method,

$$\boldsymbol{k}_1(x, \boldsymbol{y}) = \boldsymbol{f}(x, \boldsymbol{y}),$$
$$\boldsymbol{k}_s(x, \boldsymbol{y}; h) = \boldsymbol{f}\left(x + \mu_s h, \boldsymbol{y} + h\sum_{j=1}^{s-1} \lambda_{sj}\boldsymbol{k}_j\right), s = 2, 3, \ldots, r,$$
$$\boldsymbol{\Phi}(x, \boldsymbol{y}; h) = \sum_{s=1}^{r} \alpha_s \boldsymbol{k}_s.$$

Then for $\boldsymbol{\Phi}^*$ choose a similar r^*-stage process, with $r^* > r$, in such a way that

$$\mu_s^* = \mu_s,\ \ \lambda_{sj}^* = \lambda_{sj} \quad \text{for } s = 2, 3, \ldots, r.$$

The estimate (5.126) then costs only $r^* - r$ extra evaluations of $\boldsymbol{f}$. If $r^* = r + 1$, one might even attempt to save the additional evaluation by selecting (if possible)

$$\mu_{r^*} = 1, \lambda_{r^* j} = \alpha_j \quad \text{for } j = 1, 2, \ldots, r^* - 1 \ \ (r^* = r + 1). \tag{5.127}$$

Then indeed, $\boldsymbol{k}_{r^*}$ will be identical with $\boldsymbol{k}_1$ for the next step.

Pairs of such embedded $(p, p + 1)$ Runge–Kutta formulae have been developed in the late 1960s by E. Fehlberg. There is a considerable degree of freedom in choosing the parameters. Fehlberg's choices were guided by an attempt to reduce the magnitude of the coefficients of all the partial derivative aggregates that enter into the principal error function $\boldsymbol{\tau}(x, \boldsymbol{y})$ of $\boldsymbol{\Phi}$ (cf. the end of Sect. 5.6.4 for an elementary example of this technique). He succeeded in obtaining pairs with the following values of parameters p, r, r^* (Table 5.1).

Table 5.1 Embedded Runge–Kutta formulae

p	r	r^*
3	4	5
4	5	6
5	6	8
6	8	10
7	11	13
8	15	17

For the third-order process (and only for that one), one can also arrange for (5.127) to hold (cf. Ex. 15 for a second-order process).

5.8.3 Step Control

Any estimate $\boldsymbol{r}(x, \boldsymbol{y}; h)$ of the principal error function $\boldsymbol{\tau}(x, \boldsymbol{y})$ implies an estimate

$$h^p \boldsymbol{r}(x, \boldsymbol{y}; h) = \boldsymbol{T}(x, \boldsymbol{y}; h) + O(h^{p+1}) \tag{5.128}$$

for the truncation error, which can be used to monitor the local truncation error during the integration process. However, one has to keep in mind that the local truncation error is quite different from the global error, the error that one really wants to control. To get more insight into the relationship between these two errors, we recall the following theorem, which quantifies the continuity of the solution of an initial value problem with respect to initial values.

Theorem 5.8.2. *Let $\boldsymbol{f}(x, \boldsymbol{y})$ be continuous in x for $a \le x \le b$ and satisfy a Lipschitz condition uniformly on $[a, b] \times \mathbb{R}^d$ with Lipschitz constant L* (cf. (5.27)). *Then the initial value problem*

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}x} &= \boldsymbol{f}(x, \boldsymbol{y}), \quad a \le x \le b, \\ \boldsymbol{y}(c) &= \boldsymbol{y}_c, \end{aligned} \tag{5.129}$$

has a unique solution on $[a, b]$ for any c with $a \le c \le b$ and for any $\boldsymbol{y}_c \in \mathbb{R}^d$. Let $\boldsymbol{y}(x; \boldsymbol{s})$ and $\boldsymbol{y}(x; \boldsymbol{s}^*)$ be the solutions of (5.129) corresponding to $\boldsymbol{y}_c = \boldsymbol{s}$ and $\boldsymbol{y}_c = \boldsymbol{s}^*$, respectively. Then, for any vector norm $\| \cdot \|$,

$$\|\boldsymbol{y}(x; \boldsymbol{s}) - \boldsymbol{y}(x; \boldsymbol{s}^*)\| \le \mathrm{e}^{L|x-c|} \|\boldsymbol{s} - \boldsymbol{s}^*\|. \tag{5.130}$$

Solving the given initial value problem (5.102) numerically by a one-step method (not necessarily with constant step) in reality means that one follows a sequence of “solution tracks,” whereby at each grid point x_n one jumps from one track to the next by an amount determined by the truncation error at x_n (cf. Fig. 5.2). This is so by the very definition of truncation error, the reference solution being one of the

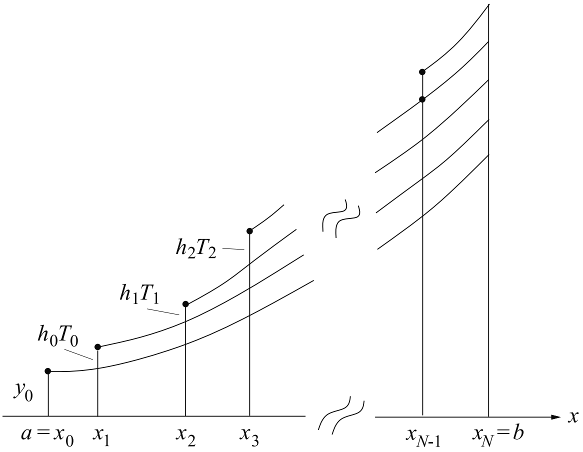


Fig. 5.2 Error accumulation in a one-step method

solution tracks. Specifically, the nth track, $n = 0, 1, \dots, N$, is given by the solution of the initial value problem

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{v}_n}{\mathrm{d}x} &= \boldsymbol{f}(x, \boldsymbol{v}_n), \quad x_n \le x \le b, \\ \boldsymbol{v}_n(x_n) &= \boldsymbol{u}_n, \end{aligned} \tag{5.131}$$

and

$$\boldsymbol{u}_{n+1} = \boldsymbol{v}_n(x_{n+1}) + h_n \boldsymbol{T}(x_n, \boldsymbol{u}_n; h_n), \quad n = 0, 1, \dots, N-1. \tag{5.132}$$

Since by (5.131) we have $\boldsymbol{u}_{n+1} = \boldsymbol{v}_{n+1}(x_{n+1})$, we can apply Theorem 5.8.2 to the solutions $\boldsymbol{v}_{n+1}$ and $\boldsymbol{v}_n$, letting $c = x_{n+1}$, $\boldsymbol{s} = \boldsymbol{u}_{n+1}$, $\boldsymbol{s}^* = \boldsymbol{u}_{n+1} - h_n \boldsymbol{T}(x_n, \boldsymbol{u}_n; h_n)$ (by (5.132)), and thus obtain

$$\|\boldsymbol{v}_{n+1}(x) - \boldsymbol{v}_n(x)\| \le h_n \mathrm{e}^{L|x-x_{n+1}|} \|\boldsymbol{T}(x_n, \boldsymbol{u}_n; h_n)\|, n = 0, 1, \dots, N-1. \tag{5.133}$$

Now

$$\sum_{n=0}^{N-1} [\boldsymbol{v}_{n+1}(x) - \boldsymbol{v}_n(x)] = \boldsymbol{v}_N(x) - \boldsymbol{v}_0(x) = \boldsymbol{v}_N(x) - \boldsymbol{y}(x), \tag{5.134}$$

and since $\boldsymbol{v}_N(x_N) = \boldsymbol{u}_N$, letting $x = x_N$, we get from (5.133) and (5.134) that

$$\begin{aligned} \|\boldsymbol{u}_N - \boldsymbol{y}(x_N)\| &\le \sum_{n=0}^{N-1} \|\boldsymbol{v}_{n+1}(x_N) - \boldsymbol{v}_n(x_N)\| \\ &\le \sum_{n=0}^{N-1} h_n \mathrm{e}^{L|x_N - x_{n+1}|} \|\boldsymbol{T}(x_n, \boldsymbol{u}_n; h_n)\|. \end{aligned}$$

Therefore, if we make sure that

$$\|\boldsymbol{T}(x_n, \boldsymbol{u}_n; h_n)\| \leq \varepsilon_T, \quad n = 0, 1, 2, \ldots, N-1, \tag{5.135}$$

then

$$\|\boldsymbol{u}_N - \boldsymbol{y}(x_N)\| \leq \varepsilon_T \sum_{n=0}^{N-1} (x_{n+1} - x_n) \mathrm{e}^{L|x_N - x_{n+1}|}.$$

Interpreting the sum on the right as a Riemann sum for a definite integral, we finally obtain, approximately,

$$\|\boldsymbol{u}_N - \boldsymbol{y}(x_N)\| \leq \varepsilon_T \int_a^b \mathrm{e}^{L(b-x)} \mathrm{d}x = \frac{\varepsilon_T}{L} (\mathrm{e}^{L(b-a)} - 1).$$

Thus, knowing an estimate for L would allow us to set an appropriate ε_T, namely,

$$\varepsilon_T = \frac{L}{\mathrm{e}^{L(b-a)} - 1} \varepsilon, \tag{5.136}$$

to guarantee an error $\|\boldsymbol{u}_N - \boldsymbol{y}(x_N)\| \leq \varepsilon$. What holds for the whole grid on $[a, b]$, of course, holds for any grid on a subinterval $[a, x]$, $a < x \leq b$. So, in principle, given the desired accuracy ε for the solution $\boldsymbol{y}(x)$, we can determine a "local tolerance level" ε_T by (5.136) and achieve the desired accuracy by keeping the local truncation error below ε_T (cf. (5.135)). Note that as $L \to 0$ we have $\varepsilon_T \to \varepsilon/(b-a)$. This limit value of ε_T would be appropriate for a quadrature problem but definitely not for a true differential equation problem, where ε_T, in general, has to be chosen considerably smaller than the target error tolerance ε.

Considerations such as these motivate the following *step control* mechanism: each integration step (from x_n to $x_{n+1} = x_n + h_n$) consists of these parts:

1. Estimate h_n.
2. Compute $\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + h_n \boldsymbol{\Phi}(x_n, \boldsymbol{u}_n; h_n)$ and $\boldsymbol{r}(x_n, \boldsymbol{u}_n; h_n)$.
3. Test $h_n^p \|\boldsymbol{r}(x_n, \boldsymbol{u}_n; h_n)\| \leq \varepsilon_T$ (cf. (5.128) and (5.135)). If the test passes, proceed with the next step; if not, repeat the step with a smaller h_n, say, half as large, until the test passes.

To estimate h_n, assume first that $n \geq 1$, so that the estimator from the previous step, $\boldsymbol{r}(x_{n-1}, \boldsymbol{u}_{n-1}; h_{n-1})$ (or at least its norm), is available. Then, neglecting terms of $O(h)$,

$$\|\boldsymbol{\tau}(x_{n-1}, \boldsymbol{u}_{n-1})\| \approx \|\boldsymbol{r}(x_{n-1}, \boldsymbol{u}_{n-1}; h_{n-1})\|,$$

and since $\boldsymbol{\tau}(x_n, \boldsymbol{u}_n) \approx \boldsymbol{\tau}(x_{n-1}, \boldsymbol{u}_{n-1})$, likewise

$$\|\boldsymbol{\tau}(x_n, \boldsymbol{u}_n)\| \approx \|\boldsymbol{r}(x_{n-1}, \boldsymbol{u}_{n-1}; h_{n-1})\|.$$

What we want is

$$\|\boldsymbol{\tau}(x_n, \boldsymbol{u}_n)\| h_n^p \approx \theta \varepsilon_T,$$

where θ is "safety factor," say, $\theta = 0.8$. Eliminating $\boldsymbol{\tau}(x_n, \boldsymbol{u}_n)$, we find

$$h_n \approx \left\{ \frac{\theta \varepsilon_T}{\|\boldsymbol{r}(x_{n-1}, \boldsymbol{u}_{n-1}; h_{n-1})\|} \right\}^{1/p}.$$

Note that from the previous step we have $h_{n-1}^p \|\boldsymbol{r}(x_{n-1}, \boldsymbol{u}_{n-1}; h_{n-1})\| \leq \varepsilon_T$, so that

$$h_n \geq \theta^{1/p} h_{n-1},$$

and the tendency is toward increasing the step.

If $n = 0$, we proceed similarly, using some initial guess $h_0^{(0)}$ of h_0 and associated $\boldsymbol{r}(x_0, \boldsymbol{y}_0; h_0^{(0)})$ to obtain

$$h_0^{(1)} = \left\{ \frac{\theta \varepsilon_T}{\|\boldsymbol{r}(x_0, y_0; h_0^{(0)})\|} \right\}^{1/p}.$$

The process may be repeated once or twice to get the final estimate of h_0 and $\|\boldsymbol{r}(x_0, \boldsymbol{y}_0; h_0\|$.

5.9 Stiff Problems

Although there is no generally accepted definition of stiffness[7] of differential equations, a characteristic feature of stiffness is the presence of rapidly changing transients. This manifests itself mathematically in the Jacobian matrix $\boldsymbol{f}_{\boldsymbol{y}}$ having eigenvalues with very large negative real parts along with others of normal magnitude. Standard (in particular explicit) numerical ODE methods are unable to cope with such solutions unless they use unrealistically small step lengths. What is called for are methods enjoying a special stability property called A-stability. We introduce this concept in the context of linear homogeneous systems of differential equations with constant coefficient matrix. Padé approximants to the exponential function turn out to be instrumental in constructing A-stable one-step methods.

[7] The word "stiffness" comes from the differential equation governing the oscillation of a "stiff" spring, that is, a spring with a large spring constant.

5.9.1 A-Stability

A model problem exhibiting stiffness is the linear initial value problem

$$\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}x} = \boldsymbol{A}\boldsymbol{y},\ 0 \le x < \infty;\ \boldsymbol{y}(a) = \boldsymbol{y}_0, \tag{5.137}$$

where $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ is a constant matrix of order d having all its eigenvalues in the left half-plane,

$$\operatorname{Re}\lambda_i(\boldsymbol{A}) < 0,\ i = 1, 2, \dots, d. \tag{5.138}$$

It is well known that all solutions of the differential system in (5.137) then decay exponentially as $x \to \infty$. Those corresponding to eigenvalues with very large negative parts do so particularly fast, giving rise to the phenomenon of stiffness. In particular, for the solution $\boldsymbol{y}(x)$ of (5.137), we have

$$\boldsymbol{y}(x) \to \boldsymbol{0} \quad \text{as } x \to \infty. \tag{5.139}$$

How does a one-step method $\boldsymbol{\Phi}$ behave when applied to (5.137)? First of all, a generic step of the one-step method will now have the form

$$\boldsymbol{y}_{\text{next}} = \boldsymbol{y} + h\boldsymbol{\Phi}(x, \boldsymbol{y}; h) = \varphi(h\boldsymbol{A})\boldsymbol{y}, \tag{5.140}$$

where φ is some function, called the *stability function* of the method. In what follows we assume that the matrix function $\varphi(h\boldsymbol{A})$ is well defined; minimally, we require that φ: $\mathbb{C} \to \mathbb{C}$ is analytic in a neighborhood of the origin. Since the reference solution through the point $(x, \boldsymbol{y})$ is given by $\boldsymbol{u}(t) = \mathrm{e}^{\boldsymbol{A}(t-x)}\boldsymbol{y}$, we have for the truncation error of $\boldsymbol{\Phi}$ at $(x, \boldsymbol{y})$ (cf. (5.31))

$$\boldsymbol{T}(x, \boldsymbol{y}; h) = \frac{1}{h}[\boldsymbol{y}_{\text{next}} - \boldsymbol{u}(x + h)] = \frac{1}{h}[\varphi(h\boldsymbol{A}) - \mathrm{e}^{h\boldsymbol{A}}]\boldsymbol{y}. \tag{5.141}$$

In particular, the method $\boldsymbol{\Phi}$ in this case has order p if and only if

$$\mathrm{e}^z = \varphi(z) + O(z^{p+1}), \quad z \to 0. \tag{5.142}$$

This shows the relevance of approximations to the exponential function in the context of one-step methods applied to the model problem (5.137).

The approximate solution $\boldsymbol{u} = \{\boldsymbol{u}_n\}$ to the initial value problem (5.137), assuming for simplicity a constant grid length h, is given by

$$\boldsymbol{u}_{n+1} = \varphi(h\boldsymbol{A})\boldsymbol{u}_n,\ n = 0, 1, 2, \dots;\ \boldsymbol{u}_0 = \boldsymbol{y}_0;$$

hence

$$\boldsymbol{u}_n = [\varphi(h\boldsymbol{A})]^n \boldsymbol{y}_0, \; n = 0, 1, 2, \dots . \tag{5.143}$$

This will simulate the behavior (5.139) of the exact solution if and only if

$$\lim_{n\to\infty} [\varphi(h\boldsymbol{A})]^n = \mathbf{0}. \tag{5.144}$$

A necessary and sufficient condition for (5.144) to hold is that all eigenvalues of the matrix $\varphi(h\boldsymbol{A})$ be strictly within the unit circle. This in turn is equivalent to

$$|\varphi(h\lambda_i(\boldsymbol{A}))| < 1 \text{ for } i = 1, 2, \dots, d, \tag{5.145}$$

where $\lambda_i(\boldsymbol{A})$ are the eigenvalues of $\boldsymbol{A}$. In view of (5.138), this gives rise to the following definition.

Definition 5.9.1. A one-step method $\boldsymbol{\Phi}$ is called *A-stable* if the function φ associated with $\boldsymbol{\Phi}$ according to (5.140) is defined in the left half of the complex plane and satisfies

$$|\varphi(z)| < 1 \;\; \text{for all } z \text{ with } \operatorname{Re} z < 0. \tag{5.146}$$

We are led to the problem of constructing a function φ (and with it, a one-step method $\boldsymbol{\Phi}$), which is analytic in the left-half plane, approximates well the exponential function near the origin (cf. (5.142)), and satisfies (5.146). An important tool for this is Padé approximation to the exponential function.

5.9.2 Padé[8] Approximation

For any function $g(z)$ analytic in a neighborhood of $z = 0$, one defines its Padé approximants as follows.

Definition 5.9.2. The *Padé approximant* $R[n,m](z)$ to the function $g(z)$ is the rational function

$$R[n,m](z) = \frac{P(z)}{Q(z)}, \;\; P \in \mathbb{P}_m, \;\; Q \in \mathbb{P}_n, \tag{5.147}$$

satisfying

$$g(z)Q(z) - P(z) = O(z^{n+m+1}) \;\; \text{as} \;\; z \to 0. \tag{5.148}$$

[8] Henri Eugène Padé (1863–1953), a French mathematician, was educated partly in Germany and partly in France, where he wrote his thesis under Hermite's supervision. Although much of his time was consumed by high administrative duties, he managed to write many papers on continued fractions and rational approximation. His thesis and related papers became widely known after Borel referred to them in his 1901 book on divergent series.

Thus, when expanding the left-hand side of (5.148) in powers of z, all initial terms should drop out up to (and including) the one with power z^{n+m}. It is known that the rational function $R[n,m]$ is uniquely determined by this definition, even though in exceptional cases P and Q may have common factors. If this is not the case, that is, P and Q are irreducible over the complex numbers, we assume without loss of generality that $Q(0) = 1$.

Our interest here is in the function $g(z) = \mathrm{e}^z$. In this case, $P = P[n,m]$ and $Q = Q[n,m]$ in (5.147) and (5.148) can be explicitly determined.

Theorem 5.9.1. *The Padé approximant $R[n,m]$ to the exponential function $g(z) = \mathrm{e}^z$ is given by*

$$P[n,m](z) = \sum_{k=0}^{m} \frac{m!(n+m-k)!}{(m-k)!(n+m)!}\,\frac{z^k}{k!}, \tag{5.149}$$

$$Q[n,m](z) = \sum_{k=0}^{n} (-1)^k \frac{n!(n+m-k)!}{(n-k)!(n+m)!}\,\frac{z^k}{k!}. \tag{5.150}$$

Moreover,

$$\mathrm{e}^z - \frac{P[n,m](z)}{Q[n,m](z)} = C_{n,m} z^{n+m+1} + \cdots,$$

where

$$C_{n,m} = (-1)^n \frac{n!m!}{(n+m)!(n+m+1)!}. \tag{5.151}$$

Proof. Let

$$v(t) := t^n(1-t)^m.$$

By Leibniz's rule, one finds

$$\begin{aligned} v^{(r)}(t) &= \sum_{k=0}^{r} \binom{r}{k} [t^n]^{(k)} [(1-t)^m]^{(r-k)} \\ &= \sum_{k=0}^{r} \binom{r}{k}\binom{n}{k} k!\,t^{n-k} \binom{m}{r-k} (r-k)!(1-t)^{m-r+k}(-1)^{r-k}; \end{aligned}$$

hence, in particular,

$$v^{(r)}(0) = (-1)^{r-n}\binom{m}{r-n} r! \quad \text{if } r \ge n;\ v^{(r)}(0) = 0 \quad \text{if } r < n;$$

$$v^{(r)}(1) = (-1)^{m}\binom{n}{r-m} r! \quad \text{if } r \ge m;\ v^{(r)}(1) = 0 \quad \text{if } r < m. \tag{5.152}$$

Given any integer $q \geq 0$, repeated integration by parts yields

$$\int_0^1 \mathrm{e}^{tz} v(t)\mathrm{d}t = \mathrm{e}^z \sum_{r=0}^{q} (-1)^r \frac{v^{(r)}(1)}{z^{r+1}} - \sum_{r=0}^{q} (-1)^r \frac{v^{(r)}(0)}{z^{r+1}} + \frac{(-1)^{q+1}}{z^{q+1}} \int_0^1 \mathrm{e}^{tz} v^{(q+1)}(t)\mathrm{d}t. \tag{5.153}$$

Putting here $q = n + m$, so that $v^{(q+1)}(t) \equiv 0$, and multiplying by $(-1)^q z^{q+1}$ gives

$$(-1)^q \mathrm{e}^z \sum_{r=0}^{q} (-1)^r v^{(r)}(1) z^{q-r} - (-1)^q \sum_{r=0}^{q} (-1)^r v^{(r)}(0) z^{q-r} + O(z^{n+m+1}), \quad z \to 0,$$

where the O-term comes from multiplying the integral on the left of (5.153) (which is $O(1)$ as $z \to 0$) by $z^{q+1} = z^{n+m+1}$. In the first sum it suffices, by (5.152), to sum over $r \geq m$, and in the second, to sum over $r \geq n$. Using the new variable of summation k defined by $q - r = k$, we find

$$\mathrm{e}^z \sum_{k=0}^{n} (-1)^k v^{(n+m-k)}(1) z^k - \sum_{k=0}^{m} (-1)^k v^{(n+m-k)}(0) z^k = O(z^{n+m+1}),$$

which clearly is (5.148) for $g(z) = \mathrm{e}^z$. It now suffices to substitute the values (5.152) for the derivatives of v at 0 and 1 to obtain (5.149) and (5.150), after multiplication of numerator and denominator by $(-1)^m/(n+m)!$ Tracing the constants, one readily checks (5.151). □

The Padé approximants to the exponential function have some very useful and important properties. Here are those of interest in connection with A-stability:

1. $P[n,m](z) = Q[m,n](-z)$: *The numerator polynomial is the denominator polynomial with indices interchanged and z replaced by $-z$.* This reflects the property $1/\mathrm{e}^z = \mathrm{e}^{-z}$ of the exponential function. The proof follows immediately from (5.149) and (5.150).
2. *For each $n = 0, 1, 2, \ldots$, all zeros of $Q[n,n]$ have positive real parts* (hence, by (1), all zeros of $P[n,n]$ have negative real parts). A proof can be given by applying the Routh–Hurwitz criterion[9] for stable polynomials.

[9] The Routh–Hurwitz criterion states that a real polynomial $a_0 x^n + a_1 x^{n-1} + \cdots + a_n$, $a_0 > 0$, has all its zeros in the left half of the complex plane if and only if all leading principal minors of the nth-order Hurwitz matrix H are positive. Here the elements in the ith row of H are $a_{2-i}, a_{4-i}, \ldots, a_{2n-i}$ (where $a_k = 0$ if $k < 0$ or $k > n$).

3. *For real* $t \in \mathbb{R}$, *and* $n = 0, 1, 2, \ldots$, *there holds*

$$\left| \frac{P[n,n](it)}{Q[n,n](it)} \right| = 1.$$

Indeed, by property (1), one has $P[n,n](it) = \overline{Q[n,n](it)}$.

4. *There holds*

$$\left| \frac{P[n+1,n](it)}{Q[n+1,n](it)} \right| < 1 \; for \; t \in \mathbb{R}, \; t \neq 0, \; n = 0, 1, 2, \ldots.$$

The proof follows from the basic property (5.148) of the Padé approximant:

$$\mathrm{e}^{it} Q[n+1,n](it) - P[n+1,n](it) = O(|t|^{2n+2}), \; t \to 0.$$

Taking absolute values, and using the triangle inequality, gives

$$\begin{aligned} \big|\, |Q[n+1,n](it)| - |P[n+1,n](it)| \,\big| &\leq |\mathrm{e}^{it} Q[n+1,n](it) \\ &\quad - P[n+1,n](it)| = O(|t|^{2n+2}); \end{aligned}$$

that is,

$$|Q[n+1,n](it)| - |P[n+1,n](it)| = O(|t|^{2n+2}).$$

Multiply this by $|Q[n+1,n](it)| + |P[n+1,n](it)|$ to obtain

$$|Q[n+1,n](it)|^2 - |P[n+1,n](it)|^2 = O(|t|^{2n+2}), \; t \to 0, \qquad (5.154)$$

where the order term is unaffected since both Q and P have the value 1 at $t = 0$ (cf. (5.149) and (5.150)). But $|P[n+1,n](it)|^2 = P[n+1,n](it) \cdot P[n+1,n](-it)$ is a polynomial of degree n in t^2, and similarly, $|Q[n+1,n](it)|^2$ a polynomial of degree $n+1$ in t^2. Thus, (5.154) can hold only if

$$|Q[n+1,n](it)|^2 - |P[n+1,n](it)|^2 = at^{2n+2},$$

where at^{2n+2} is the leading term in $|Q[n+1,n](it)|^2$, hence $a > 0$. From this, the assertion follows immediately.

5. *For each* $n = 0, 1, 2, \ldots$, *all zeros of* $Q[n+1,n]$ *have positive real parts*. We sketch the proof. From (5.149) and (5.150), one notes that

$$Q[n+1,n](z) + P[n+1,n](-z) = 2Q[n+1,n+1](z).$$

We now use Rouché's theorem to show that $Q[n+1,n]$ and $Q[n+1,n+1]$ have the same number of zeros with negative real part (namely, none according to

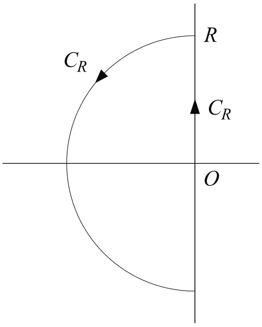


Fig. 5.3 The contour C_R

property (2)). For this, one must show that on the contour C_R: $\{z = it, |t| \le R, t \ne 0\} \cup \{|z| = R, \operatorname{Re} z < 0\}$ (see Fig. 5.3) one has, when R is sufficiently large,

$$|P[n+1,n](-z)| < |Q[n+1,n](z)|, \quad Q[n+1,n](0) \ne 0.$$

For $z = it$, since $P[n+1,n](-it) = \overline{P[n+1,n](it)}$, the inequality follows from property (4). For $|z| = R$, the inequality holds for $R \to \infty$, since deg $P[n+1,n] <$ deg $Q[n+1,n]$. Finally $Q[n+1,n](0) = 1$.

Note also from (4) that $Q[n+1,n]$ cannot have purely imaginary zeros, since $|Q[n+1,n](it)| > |P[n+1,n](it)| \ge 0$ for $t \ne 0$.

6. *A rational function R satisfies* $|R(z)| < 1$ *for* $\operatorname{Re} z < 0$ *if and only if R is analytic in* $\operatorname{Re} z < 0$ *and* $|R(z)| \le 1$ *for* $\operatorname{Re} z = 0$.

   Necessity. If $|R(z)| < 1$ in $\operatorname{Re} z < 0$, there can be no pole in $\operatorname{Re} z \le 0$ or at $z = \infty$. By continuity, therefore, $|R(z)| \le 1$ on $\operatorname{Re} z = 0$.

   Sufficiency. R must be analytic in $\operatorname{Re} z \le 0$ and at $z = \infty$. Clearly, $\lim_{z\to\infty} |R(z)| \le 1$ and $|R(z)| \le 1$ on the imaginary axis. Then, by the maximum principle, $|R(z)| < 1$ for $\operatorname{Re} z < 0$.
7. As a corollary of property (6), we state the following important properties. *For each* $n = 0, 1, 2, \dots,$ *there holds*

$$\left| \frac{P[n,n](z)}{Q[n,n](z)} \right| < 1 \textit{ for } \operatorname{Re} z < 0, \tag{5.155}$$

$$\left| \frac{P[n+1,n](z)}{Q[n+1,n](z)} \right| < 1 \textit{ for } \operatorname{Re} z \le 0,\ z \ne 0. \tag{5.156}$$

The first of these inequalities follows from properties (1) through (3), the second from properties (4) and (5).

Property (7) immediately yields the following theorem.

Theorem 5.9.2. *If the function φ associated with the one-step method $\boldsymbol{\Phi}$ according to (5.140) is either the Padé approximant $\varphi(z) = R[n,n](z)$ of e^z, or the Padé approximant $\varphi(z) = R[n+1,n](z)$ of $\mathrm{e}^z, n = 0,1,2,\ldots$, then the method $\boldsymbol{\Phi}$ is A-stable.*

5.9.3 Examples of A-Stable One-Step Methods

1. *Implicit Euler method*: Also called the *backward* Euler method, this is the one-step method defined by

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + h\boldsymbol{f}(x_{n+1}, \boldsymbol{u}_{n+1}). \tag{5.157}$$

It requires, at each step, the solution of a system of (in general) nonlinear equations for $\boldsymbol{u}_{n+1} \in \mathbb{R}^d$. In the case of the model problem (5.137), this becomes $\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + h\boldsymbol{A}\boldsymbol{u}_{n+1}$ and can be solved explicitly: $\boldsymbol{u}_{n+1} = (\boldsymbol{I} - h\boldsymbol{A})^{-1}\boldsymbol{u}_n$. Thus, the associated function φ here is

$$\varphi(z) = \frac{1}{1-z} = 1 + z + z^2 \ldots, \tag{5.158}$$

the Padé approximant $R[1,0](z)$ of e^z. Since $\varphi(z) - \mathrm{e}^z = O(z^2)$ as $z \to 0$, the method has order $p = 1$ (cf. (5.142)), and by Theorem 5.9.2 is A-stable. (This could easily be confirmed directly.)

2. *Trapezoidal rule*: Here

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + \frac{1}{2}h\,[\boldsymbol{f}(x_n, \boldsymbol{u}_n) + \boldsymbol{f}(x_{n+1}, \boldsymbol{u}_{n+1})], \tag{5.159}$$

again a nonlinear equation in $\boldsymbol{u}_{n+1}$. For the model problem (5.137), this becomes $\boldsymbol{u}_{n+1} = \left(\boldsymbol{I} + \frac{1}{2}h\boldsymbol{A}\right)\boldsymbol{u}_n + \frac{1}{2}h\boldsymbol{A}\boldsymbol{u}_{n+1}$, hence $\boldsymbol{u}_{n+1} = \left(\boldsymbol{I} - \frac{1}{2}h\boldsymbol{A}\right)^{-1}\left(\boldsymbol{I} + \frac{1}{2}h\boldsymbol{A}\right)\boldsymbol{u}_n$, and

$$\varphi(z) = \frac{1+\frac{1}{2}z}{1-\frac{1}{2}z} = 1 + z + \frac{1}{2}z^2 + \frac{1}{4}z^3 + \cdots. \tag{5.160}$$

This is the Padé approximant $\varphi(z) = R[1,1](z)$ of e^z and $\varphi(z) - \mathrm{e}^z = O(z^3)$, so that again the method is A-stable, but now of order $p = 2$.

3. *Implicit Runge–Kutta formulae*: As mentioned in Sect. 5.6.5, an r-stage implicit Runge–Kutta formula has the form (cf. (5.68))

$$\boldsymbol{\Phi}(x, \boldsymbol{y}; h) = \sum_{s=1}^{r} \alpha_s \boldsymbol{k}_s(x, \boldsymbol{y}; h),$$

$$\boldsymbol{k}_s = \boldsymbol{f}\left(x + \mu_s h, \boldsymbol{y} + h\sum_{j=1}^{r} \lambda_{sj}\boldsymbol{k}_j\right),\quad s = 1,2,\ldots,r. \tag{5.161}$$

It is possible to show (cf. Notes to Sect. 5.6.5) that (5.161) is a method of order $p, r \le p \le 2r$, if $f \in C^p$ on $[a,b] \times \mathbb{R}^d$ and

$$\sum_{j=1}^{r} \lambda_{sj} \mu_j^k = \frac{\mu_s^{k+1}}{k+1}, \quad k = 0, 1, \dots, r-1; \; s = 1, 2, \dots, r, \tag{5.162}$$

$$\sum_{s=1}^{r} \alpha_s \mu_s^k = \frac{1}{k+1}, \quad k = 0, 1, \dots, p-1. \tag{5.163}$$

For any set of *distinct* μ_j, and for each $s = 1, 2, \dots, r$, the equations (5.162) represent a system of linear equations for $\{\lambda_{sj}\}_{j=1}^r$, whose coefficient matrix is a Vandermonde matrix, hence nonsingular. It thus can be solved uniquely for the $\{\lambda_{sj}\}$. Both conditions (5.162) and (5.163) can be viewed more naturally in terms of quadrature formulae. Indeed, (5.162) is equivalent to

$$\int_0^{\mu_s} p(t)\mathrm{d}t = \sum_{j=1}^{r} \lambda_{sj}\, p(\mu_j), \text{ all } p \in \mathbb{P}_{r-1}, \tag{5.164}$$

whereas (5.163) means

$$\int_0^1 q(t)\mathrm{d}t = \sum_{s=1}^{r} \alpha_s q(\mu_s), \text{ all } q \in \mathbb{P}_{p-1}. \tag{5.165}$$

We know from Chap. 3, Sect. 3.2.2, that in (5.165) we can indeed have $r \le p \le 2r$, the extreme values corresponding to Newton–Cotes formulae (with prescribed μ_s) and to the Gauss–Legendre formula on [0,1], where the μ_s are the zeros of the (shifted) Legendre polynomial of degree r. In the latter case, we obtain a unique *r-stage Runge–Kutta formula of order $p = 2r$*. We now show that this Runge–Kutta method of maximum order $2r$ is also A-stable.

Instead of the system (5.137), we may as well consider a scalar equation

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \lambda y, \tag{5.166}$$

to which (5.137) can be reduced by spectral decomposition (i.e., λ represents one of the eigenvalues of $\boldsymbol{A}$). Applied to (5.166), the k_s corresponding to (5.161) must satisfy the linear system

$$k_s = \lambda \left(y + h \sum_{j=1}^{r} \lambda_{sj} k_j \right);$$

that is (with $z = \lambda h$),

$$k_s - z \sum_{j=1}^{r} \lambda_{sj} k_j = \lambda y, \quad s = 1, 2, \dots, r.$$

Let

$$d_r(z) = \begin{vmatrix} 1 - z\lambda_{11} & -z\lambda_{12} & \cdots & -z\lambda_{1r} \\ -z\lambda_{21} & 1 - z\lambda_{22} & \cdots & -z\lambda_{2r} \\ \dots\dots\dots & \dots\dots\dots & \dots & \dots\dots\dots \\ -z\lambda_{r1} & -z\lambda_{r2} & \cdots & 1 - z\lambda_{rr} \end{vmatrix},$$

$$d_{r,s}(z) = \begin{vmatrix} 1 - z\lambda_{11} & \cdots & 1 & \cdots & -z\lambda_{1r} \\ -z\lambda_{21} & \cdots & 1 & \cdots & -z\lambda_{2r} \\ \dots\dots\dots & \dots & . & \dots & \dots\dots\dots \\ -z\lambda_{r1} & \cdots & 1 & \cdots & 1 - z\lambda_{rr} \end{vmatrix}, \quad s = 1, 2, \dots, r,$$

where the column of ones is the sth column of the determinant. Clearly, d_r and $d_{r,s}$ are polynomials of degree r and $r - 1$, respectively. By Cramer's rule,

$$k_s = \frac{d_{r,s}(z)}{d_r(z)} \lambda y, \quad s = 1, 2, \dots, r,$$

so that

$$y_{\text{next}} = y + h \sum_{s=1}^{r} \alpha_s k_s = \left\{ 1 + z \sum_{s=1}^{r} \alpha_s \frac{d_{r,s}(z)}{d_r(z)} \right\} y.$$

Thus, the function φ associated with the method $\boldsymbol{\Phi}$ corresponding to (5.161) is

$$\varphi(z) = \frac{d_r(z) + z \sum_{s=1}^{r} \alpha_s d_{r,s}(z)}{d_r(z)}. \tag{5.167}$$

We see that φ is a rational function of type $[r, r]$ and, the method $\boldsymbol{\Phi}$ having order $p = 2r$, we have (cf. (5.142))

$$\mathrm{e}^z = \varphi(z) + O(z^{2r+1}), \quad z \to 0.$$

It follows that φ in (5.167) is the Padé approximant $R[r, r]$ to the exponential function, and hence $\boldsymbol{\Phi}$ is A-stable by Theorem 5.9.2.

4. *Ehle's method* This is a method involving total derivatives of $\boldsymbol{f}$ (cf. (5.45)),

$$\boldsymbol{\Phi}(x, \boldsymbol{y}; h) = \boldsymbol{k}(x, \boldsymbol{y}; h),$$

$$\boldsymbol{k} = \sum_{s=1}^{r} h^{s-1} [\alpha_s \boldsymbol{f}^{[s-1]}(x, \boldsymbol{y}) - \beta_s \boldsymbol{f}^{[s-1]}(x + h, \boldsymbol{y} + h\boldsymbol{k})]. \tag{5.168}$$

It is also implicit, as it requires the solution of the second equation in (5.168) for the vector $\boldsymbol{k} \in \mathbb{R}^d$. A little computation (see Ex. 21) will show that the function φ associated with $\boldsymbol{\Phi}$ in (5.168) is given by

$$\varphi(z) = \frac{1 + \sum_{s=1}^{r} \alpha_s z^s}{1 + \sum_{s=1}^{r} \beta_s z^s}. \tag{5.169}$$

By choosing this φ to be a Padé approximant to e^z, either $R[r, r]$ or $R[r, r-1]$ (by letting $\alpha_r = 0$), we again obtain two A-stable methods. The latter has the additional property of being *strongly* A-stable (or *L-stable*), in the sense that

$$\varphi(z) \to 0 \text{ as } \operatorname{Re} z \to -\infty. \tag{5.170}$$

This means, in view of (5.143), that convergence $\boldsymbol{u}_n \to \boldsymbol{0}$ as $n \to \infty$ is faster for components corresponding to eigenvalues further to the left in the complex plane.

5.9.4 Regions of Absolute Stability

For methods $\boldsymbol{\Phi}$ that are *not* A-stable, it is important to know the *region of absolute stability*,

$$\mathcal{D}_A = \{z \in \mathbb{C} : |\varphi(z)| < 1\}. \tag{5.171}$$

If the method $\boldsymbol{\Phi}$ applied to the model problem (5.137) is to produce an approximate solution $\boldsymbol{u} = \{\boldsymbol{u}_n\}$ with $\lim_{n\to\infty} \boldsymbol{u}_n = \boldsymbol{0}$, it is necessary that $h\lambda_i(\boldsymbol{A}) \in \mathcal{D}_A$ for all eigenvalues $\lambda_i(\boldsymbol{A})$ of $\boldsymbol{A}$. If some of these have very large negative real parts, then this condition imposes a severe restriction on the step length h, unless $\mathcal{D}_A$ contains a large portion of the left-hand plane. For many classical methods, unfortunately, this is not the case. For Euler's method:, for example, we have $\varphi(z) = 1 + z$, hence

$$\mathcal{D}_A = \{z \in \mathbb{C} : |1 + z| < 1\} \quad \text{(Euler)}, \tag{5.172}$$

and the region of absolute stability is the unit disk in $\mathbb{C}$ centered at -1. More generally, for the Taylor expansion method of order $p \geq 1$, and also for any p-stage explicit Runge–Kutta method of order p, $1 \leq p \leq 4$, one has (see Ex. 18)

$$\varphi(z) = 1 + \frac{1}{1!}z + \frac{1}{2!}z^2 + \cdots + \frac{1}{p!}z^p. \tag{5.173}$$

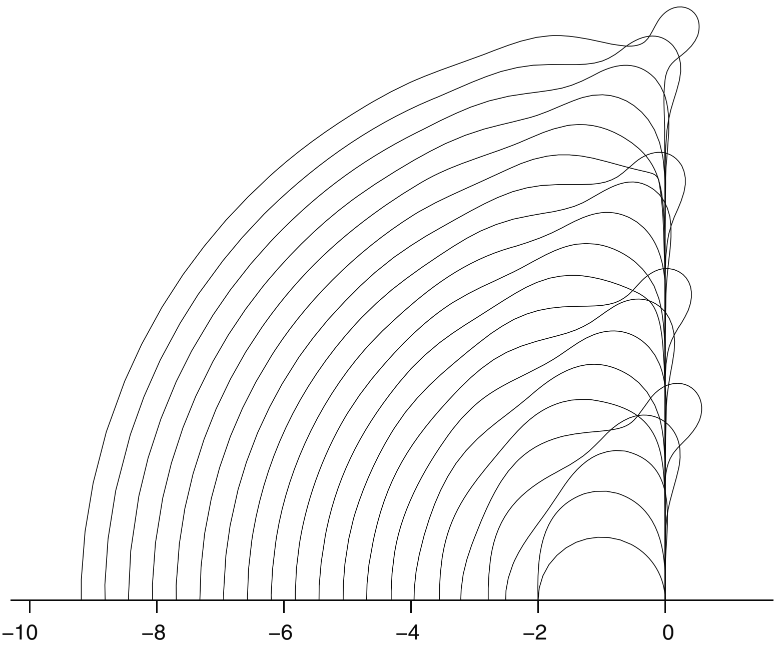


Fig. 5.4 Regions of absolute stability for pth-order methods with φ as in (5.173), $p = 1, 2, \dots, 21$

To compute the contour line $|\varphi(z)| = 1$, which delineates the region $\mathcal{D}_A$, one can find a differential equation for this line and use a one-step method to solve it (see MA 4). That is, we can use a method $\boldsymbol{\Phi}$ to analyze its own stability region – a case of self-analysis, as it were. The results for φ in (5.173) and $p = 1, 2, \dots, 21$ are plotted in Fig. 5.4[10]. Because of symmetry, only the parts of the regions in the upper half-plane are shown.

5.10 Notes to Chapter 5

The classic text on the numerical solution of nonstiff ordinary differential equations is Henrici [1962]. It owes much to the pioneering work of Dahlquist [1956] a few years earlier. Numerous texts have since been written with varying areas of emphasis, but all paying attention to stiff problems. We mention only a few of the more recent ones: the balanced exposition in Lambert [1991], the two-volume work of Hairer et al. [1993] and Hairer and Wanner [2010], which is especially

[10] Figure 5.4 is reproduced, with permission, from W. Gautschi and J. Waldvogel, *Contour plots of analytic functions*, in *Solving problems in scientific computing using Maple and MATLAB*, Walter Gander and Jiří Hřebíček, eds., 4th ed., Springer, Berlin, 2004.

rich in historical references, interesting examples, and numerical experimentation, Shampine [1994] emphasizing software issues, Iserles [2009] including also topics in numerical partial differential equations, Ascher and Petzold [1998] treating also differential algebraic equations, Bellen and Zennaro [2003] focusing on delay differential equations, Shampine et al. [2003] using Matlab as a problem solving environment, Butcher [2008], and Atkinson et al. [2009], which also provides access to Matlab codes.

An authoritative work on Runge–Kutta methods is Butcher [1987]. Error and stability analyses of Runge–Kutta methods in the context of stiff nonlinear differential equations are the subject of the monograph by Dekker and Verwer [1984].

Section 5.1. The example (5.3) is from Klopfenstein [1965]. The technique described therein is widely used in codes calculating trajectories of space vehicles. For a detailed presentation of the method of lines, including Fortran programs, see Schiesser [1991].

Section 5.2. There are still other important types of differential equations, for example, singularly perturbed equations and related differential algebraic equations (DAEs), and differential equations with delayed arguments. For the former, we refer to Griepentrog and März [1986], Brenan et al. [1996], Hairer et al. [1989], and Hairer and Wanner [2010, Chaps. 6–7], for the latter to Pinney [1958], Bellman and Cooke [1963], Cryer [1972], Driver [1977], and Kuang [1993].

Section 5.3. For a proof of the existence and uniqueness part of Theorem 5.3.1, see Henrici [1962, Sect. 1.2]. The proof is given for a scalar initial value problem, but it extends readily to systems. Continuity with respect to initial data is proved, for example, in Coddington and Levinson [1955, Chap. 1, Sect. 7]. Also see Butcher [1987, Sect. 112].

A strengthened version of Theorem 5.3.1 involves a one-sided Lipschitz condition,

$$[\boldsymbol{f}(x,\boldsymbol{y}) - \boldsymbol{f}(x,\boldsymbol{y}^*)]^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{y}^*) \le \lambda\,\|\boldsymbol{y}-\boldsymbol{y}^*\|^2, \text{ all } x \ge a, \text{ all } \boldsymbol{y},\boldsymbol{y}^* \in \mathbb{R}^d,$$

where λ is some constant – the one-sided Lipschitz constant. If this holds, and $\boldsymbol{f}$ is continuous in x, then the initial value problem (5.16) has a unique solution on any interval $[a,b]$, $b > a$ (cf. Butcher [1987, Sect. 112]).

Section 5.4. The numerical solution of differential equations can sometimes benefit from a preliminary transformation of variables. Many examples are given in Daniel and Moore [1970, Part 3]; an important example used in celestial mechanics to regularize and linearize the Newtonian equations of motion are the transformations of Levi–Civita, and of Kustaanheimo and Stiefel, for which we refer to Stiefel and Scheifele [1971] for an extensive treatment. The reader may wish to also consult Zwillinger [1992b] for a large number of other analytical tools that may be helpful in the numerical solution of differential equations.

The distinction between one-step and multistep methods may be artificial, as there are theories that allow treating them both in a unified manner; see, for example, Stetter [1973, Chap. 5], Butcher [1987, Chap. 4], or Hairer et al. [1993, Chap. 3, Sect. 8]. We chose, however, to cover them separately for didactical reasons.

There are many numerical methods in use that are not discussed in our text. Among the more important ones are the extrapolation methods of Gragg and of Gragg, Bulirsch, and Stoer, which extend the ideas in Chap. 3, Sect. 3.2.7, to differential equations. For these, we refer to Stetter [1973, Sect. 6.3] and Hairer et al. [1993, Chap. 2, Sects. 8, 9]. (Extrapolation methods for stiff problems are discussed in Hairer and Wanner [2010, Chap. 4, Sect. 9].) Both these texts also contain accounts of multistep methods involving derivatives, and of Nordsieck-type methods, which carry along not only function values but also derivative values from one step to the next. There are also methods tailored to higher-order systems of differential equations; for second-order systems, for example, see Hairer et al. [1993, Chap. 2, Sect. 14]. Very recently, so-called symplectic methods have created a great deal of interest, especially in connection with Hamiltonian systems. These are numerical methods that preserve invariants of the given differential system; cf. Sanz-Serna and Calvo [1994], Hairer et al. [2006].

Section 5.5. A nonstiff initial value problem (5.16) on $[a, b]$, where b is very large, is likely one that also has a very small Lipschitz constant. The problem, in this case, is not properly scaled, and one should transform the independent variable x, for example by letting $x = (1-t)a + tb$, to get an initial value problem on [0, 1], namely, $\mathrm{d}z/\mathrm{d}t = \boldsymbol{g}(t, z)$, $0 \le t \le 1$, $z(0) = \boldsymbol{y}_0$, where $z(t) = \boldsymbol{y}((1-t)a + tb)$ and $\boldsymbol{g}(t, z) := (b-a)\boldsymbol{f}((1-t)a + tb, z)$. If $\boldsymbol{f}$ has a (very small) Lipschitz constant L, then $\boldsymbol{g}$ has the Lipschitz constant $(b-a)L$, which may well be of more reasonable size.

Section 5.6.1. In the spirit of Laplace's exhortation "*Lisez Euler, lisez Euler, c'est notre maître à tous*!," the reader is encouraged to look at Euler's original account of his method in Euler [1768, Sect. 650]. Even though it is written in Latin, Euler's use of this language is plain and simple.

Section 5.6.2. The method of Taylor expansion was also proposed by Euler [op.cit., Sect. 656]. It has long been perceived as being too cumbersome in practice, but recent advances in automatic differentiation helped to revive interest in this method. Codes have been written that carry out the necessary differentiations systematically by recursion (Gibbons [1960] and Barton et al. [1971]). Combining these techniques with interval arithmetic, as in Moore [1979, Sect. 3.4], also provides rigorous error bounds.

Section 5.6.5. For the results in (5.70), see Butcher [1965]. Actually, $p^*(10) = 7$, as was shown more recently in Butcher [1985]. The highest orders of an explicit Runge–Kutta method ever constructed are $p = 12$ with 17 stages, and $p = 14$ with 35 stages (Feagin [2011]).

It has become customary to associate with the general r-stage Runge–Kutta method (5.68) the array

$$\begin{array}{c|cccc} \mu_1 & \lambda_{11} & \lambda_{12} & \cdots & \lambda_{1r} \\ \mu_2 & \lambda_{21} & \lambda_{22} & \cdots & \lambda_{2r} \\ \vdots & \vdots & \vdots & & \vdots \\ \mu_r & \lambda_{r1} & \lambda_{r2} & \cdots & \lambda_{rr} \\ \hline & \alpha_1 & \alpha_2 & \cdots & \alpha_r \end{array} \quad \left(\text{in matrix form: } \begin{array}{c|c} \mu & \boldsymbol{\Lambda} \\ \hline & \boldsymbol{\alpha}^{\mathrm{T}} \end{array}\right),$$

called the Butcher array. For an explicit method, $\mu_1 = 0$ and $\boldsymbol{\Lambda}$ is lower triangular with zeros on the diagonal. With the first r rows of the Butcher array, we may associate the quadrature rules $\int_0^{\mu_s} u(t)\mathrm{d}t \approx \sum_{j=1}^r \lambda_{sj} u(\mu_j)$, $s = 1, 2, \dots, r$, and with the last row the rule $\int_0^1 u(t)\mathrm{d}t \approx \sum_{s=1}^r \alpha_s u(\mu_s)$. If the respective degrees of exactness are $d_s = q_s - 1$, $1 \le s \le r + 1$ ($d_s = \infty$ if $\mu_s = 0$ and all $\lambda_{sj} = 0$), then by the Peano representation of error functionals (cf. Chap. 3, (3.79)) the remainder terms involve derivatives of u of order q_s, and hence, setting $u(t) = y'(x + th)$, one gets

$$\frac{y(x + \mu_s h) - y(x)}{h} - \sum_{j=1}^r \lambda_{sj} y'(x + \mu_j h) = O(h^{q_s}), \quad s = 1, 2, \dots, r,$$

and

$$\frac{y(x + h) - y(x)}{h} - \sum_{s=1}^r \alpha_s y'(x + \mu_s h) = O(h^{q_{r+1}}).$$

The quantity $q = \min(q_1, q_2, \dots, q_r)$ is called the stage order of the Runge–Kutta formula, and q_{r+1} the quadrature order.

High-order r-stage implicit Runge–Kutta methods have the property that, when $\boldsymbol{f}(x, \boldsymbol{y}) = \boldsymbol{f}(x)$, they reduce to r-point Gauss-type quadrature formulae, either the Gauss formula proper, or the Gauss–Radau ($\mu_1 = 0$ or $\mu_r = 1$) or Gauss–Lobatto ($\mu_1 = 0$ and $\mu_r = 1$) formula; see, for example, Dekker and Verwer [1984, Sect. 3.3], Butcher [1987, Sect. 34], Lambert [1991, Sect. 5.11], and Hairer and Wanner [2010, Chap. 4, Sect. 5]. They can be constructed to have order $2r$, and (in a variety of ways) orders $2r-1$ and $2r-2$, respectively; cf. also Sect. 5.9.3(3). Another interesting way ofconstructing implicit Runge–Kutta methods is by collocation: define $p \in \mathbb{P}_r$ to be such that $p(x) = y$, $p'(x + \mu_s h) = f(x + \mu_s h, p(x + \mu_s h))$, $s = 1, 2, \dots, r$ (cf. Chap. 2, Ex. 66), and let $y_{\text{next}} = p(x + h)$. It has been shown by Wright [1970] (also cf. Butcher [1987, Sect. 346]) that this indeed is an implicit r-stage method – a collocation method, as it is called. For such methods, the stage orders are at least r. This property characterizes collocation methods of orders $\ge r$ (with distinct μ_s); see Hairer et al. [1993, Theorem 7.8, p. 212]. The order of the method is $p = r + k$, $k \ge 0$, if the quadrature order is p (cf. [loc.cit., Chap. 2, Theorem 7.9] and Chap. 3, Theorem 3.2.1). Some (but not all) of the Gauss-type methods previously mentioned are collocation methods. If the polynomial $p(x + th)$ is determined explicitly (not just $p(x + h)$), it provides a means of computing intermediate approximations for arbitrary t with $0 < t < 1$, giving rise to a "continuous" implicit Runge–Kutta method.

Semi-implicit Runge–Kutta methods with all diagonal elements of Λ in the Butcher array being the same nonzero real number are called DIRK methods (Diagonally Implicit Runge-Kutta); see Nørsett [1974], Crouzeix [1976], and Alexander [1977]. SIRK methods (Singly-Implicit Runge-Kutta) are fully implicit methods which share with DIRK methods the property that the matrix Λ (though not triangular) has one single real eigenvalue of multiplicity r. These were derived by Nørsett [1976] and Burrage [1978a], [1978b], [1982]. DIRK methods with r stages have maximum order $r + 1$, but these are difficult to derive for large r, in contrast to SIRK methods (see Dekker and Verwer [1984, Sects. 3.5 and 3.6]).

The best source for Butcher's theory of Runge–Kutta methods and their attainable orders is Butcher [1987, Chap. 3, Sects. 30–34]. A simplified version of this theory can be found in Lambert [1991, Chap. 5] and an alternative approach in Albrecht [1987, 1996]. It may be worth noting that the order conditions for a system of differential equations are not necessarily identical with those for a single differential equation. Indeed, a Runge–Kutta method for a scalar equation may have order $p > 4$, whereas the same method applied to a system has order $< p$. (For $p \leq 4$ this phenomenon does not occur.) Examples of explicit Runge–Kutta formulae of orders 5–8 are given in Butcher [1987, Sect. 33].

An informative cross-section of contemporary work on the Runge–Kutta method, as well as historical essays, celebrating the centenary of Runge's 1895 paper, can be found in Butcher [1996].

Section 5.7.1. The concept of stability as defined in this section is from Keller [1992, Sect. 1.3]. It is also known as zero-stability (relating to $h \to 0$) to distinguish it from other stability concepts used in the context of stiff differential equations; for the latter, see the Notes to Sect. 5.9.1.

Section 5.7.2. Theorem 5.7.2 admits a converse if one assumes $\mathbf{\Phi}$ continuous and satisfying a Lipschitz condition (5.86); that is, consistency is then also necessary for convergence (cf. Henrici [1962, Theorem 3.2]).

Section 5.7.3. Theorem 5.7.3 is due independently to Henrici [1962, Theorem 3.4] and Tihonov and Gorbunov [1963, 1964]. Henrici deals also with variable steps in the form alluded to at the beginning of this section, whereas Tihonov and Gorbunov [1964] deal with arbitrary nonuniform grids.

Section 5.8.1. Although the idea of getting global error estimates by integrating the variational equation along with the main differential equation has already been expressed by Henrici [1962, p.81], its precise implementation as in Theorem 5.8.1 is carried out in Gautschi [1975b].

Section 5.8.2(2). Fehlberg's embedded 4(5) method with $r^* = 6$ (cf. Fehlberg [1969, 1970]) appears to be a popular method. (It is one of two options provided in

Matlab, the other being a 2(3) pair.) A similar method due to England [1969/1970] has the advantage of possessing coefficients $\alpha_5 = \alpha_6 = 0$, which makes occasional error monitoring more efficient. All of Fehlberg's methods of orders $p \geq 5$ have the (somewhat disturbing) peculiarity of yielding zero error estimates in cases, where $\boldsymbol{f}$ does not depend on $\boldsymbol{y}$. High-order pairs of methods not suffering from this defect have been derived in Verner [1978]. Variable-method codes developed, for example, in Shampine and Wisniewski [1978], use pairs ranging from 3(4)–7(8). Instead of optimizing the truncation error in the lower-order method of a pair, as was done by Fehlberg, one can do the same with the higher-order method and use the other only for step control. This is the approach taken by Dormand and Prince [1980] and Prince and Dormand [1981]. Their 4(5) and 7(8) pairs appear to be among current state-of-the-art choices (cf. Hairer et al. [1993, Chap. 2, Sect. 10] and the appendix of this reference for codes).

Section 5.8.3. For a proof of Theorem 5.8.2, see, for example, Butcher [1987, Theorem 112J].

Section 5.9. The major text on stiff problems is Hairer and Wanner [2010]. For Runge–Kutta methods, also see Dekker and Verwer [1984].

Section 5.9.1. The function $\varphi(z)$ for a general Runge–Kutta method can be expressed in terms of the associated Butcher array as $\varphi(z) = 1 + z\boldsymbol{\alpha}^{\mathrm{T}}(\boldsymbol{I} - z\boldsymbol{\Lambda})^{-1}\boldsymbol{e}$, where $\boldsymbol{e}^{\mathrm{T}} = [1, 1, \ldots, 1]$ or, alternatively, as $\det(\boldsymbol{I} - z\boldsymbol{\Lambda} + z\boldsymbol{e}\boldsymbol{\alpha}^{\mathrm{T}})/\det(\boldsymbol{I} - z\boldsymbol{\Lambda})$; see, for example, Dekker and Verwer [1984, Sect. 3.4], Lambert [1991, Sect. 5.12], and Hairer and Wanner [2010, Chap. 4, Sect. 3]. Thus, φ is a rational function if the method is (semi-) implicit, and a polynomial otherwise.

The concept of A-stability, which was introduced by Dahlquist [1963], can be relaxed by requiring $|\varphi(z)| \leq 1$ to hold only in an unbounded subregion S of the left half-plane. Widlund [1967], in the context of multistep methods (cf. Chap. 6, Sect. 6.5.2), for example, takes for S an angular region $|\arg(-z)| \leq \alpha$, where $\alpha < \frac{1}{2}\pi$, and speaks of A(α)-stability, whereas Gear [1971a, Sect. 11.1] takes the union of some half-plane $\operatorname{Re} z \leq \rho < 0$ and a rectangle $\rho \leq \operatorname{Re} z \leq 0$, $|\operatorname{Im} z| \leq \sigma$, and speaks of stiff stability. Other stability concepts relate to more general test problems, some linear and some nonlinear. Of particular interest among the latter are initial value problems (5.16) with $\boldsymbol{f}$ satisfying a one-sided Lipschitz condition with constant $\lambda = 0$. These systems are dissipative in the sense that $\|\boldsymbol{y}(x) - \boldsymbol{z}(x)\|$ is nonincreasing for $x > a$ for any two solutions $\boldsymbol{y}, \boldsymbol{z}$ of the differential equation. Requiring the same to hold for any two numerical solutions $\boldsymbol{u}$, $\boldsymbol{v}$ generated by the one-step method, that is, requiring that $\|\boldsymbol{u}_{n+1} - \boldsymbol{v}_{n+1}\| \leq \|\boldsymbol{u}_n - \boldsymbol{v}_n\|$ for all $n \geq 0$, gives rise to the concept of B-stability (or BN-stability with the "N" standing for "nonautonomous"). The implicit r-stage Runge–Kutta method of order $p = 2r$ (cf. Sect. 5.9.3(3)), for example, is B-stable,and so are some of the other Gauss-type Runge–Kutta methods; see Dekker and Verwer [1984, Sect. 4.1], Butcher [1987, Sect. 356], and Hairer and Wanner [2010, Chap. 4, Sects. 12 and 13]. Another family of Runge–Kutta methods that are B-stable are the so-called algebraically stable methods, that is, methods which satisfy $\boldsymbol{D} = \operatorname{diag}(\alpha_1, \alpha_2, \ldots, \alpha_r) \geq \boldsymbol{0}$ and

$\boldsymbol{D}\boldsymbol{\Lambda} + \boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{D} - \boldsymbol{\alpha}\boldsymbol{\alpha}^{\mathrm{T}}$ nonnegative definite (Dekker and Verwer [1984, Sect. 4.2], Butcher [1987, Sect. 356], and Hairer and Wanner [2010, Chap. 4, Sects. 12 and 13]). Similar stability concepts can also be associated with test equations satisfying one-sided Lipschitz conditions with constants $\lambda \neq 0$ (Dekker and Verwer [1984, Sect. 5.11], Butcher [1987, Sect. 357], and Hairer and Wanner [2010, Chap. 4, pp. 193ff]).

Section 5.9.2. Standard texts on Padé approximation are Baker [1975] and Baker and Graves-Morris [1996]. The proof of Theorem 5.9.1 follows Perron [1957, Sect. 42], who in turn took it from Padé. For a derivation of the Routh–Hurwitz criterion mentioned in (2), see, for example, Marden [1966, Corollary 40,2], and for Rouché's theorem, Henrici [1988, p.280]. The elegant argument used in the proof of Property (4) is due to Axelsson [1969].

Section 5.9.3. For the study of A-stability, it is easier to work with the "relative stability function" $\varphi(z)e^{-z}$ than with $\varphi(z)$ directly. This gives rise to the "order star" theory of Wanner et al. [1978], which, among other things, made it possible to prove that the only Padé approximants to e^z that yield A-stable methods are $\varphi(z) = R[n+k,n](z)$, $n = 0, 1, 2, \ldots$, with $0 \leq k \leq 2$. All Gauss-type Runge–Kutta methods in current use have stability functions given by such Padé approximants and are thus A-stable (Dekker and Verwer [1984, Sect. 3.4]). Also, see Iserles and Nørsett [1991], and Hairer and Wanner [2010, Chap. 4, Sect. 4 for further applications of order stars.

In addition to the implicit Gauss-type Runge–Kutta methods mentioned in (3), there are also A-stable DIRK and SIRK methods; for their construction, see Butcher [1987, Sect. 353] and Hairer and Wanner [2010, Chap. 4, Sect. 6]. Another class of methods that are A-stable, or nearly so, are basically explicit Runge–Kutta methods that make use of the Jacobian matrix and inverse matrices involving it. They are collectively called Runge–Kutta–Rosenbrock methods; see, for example, Dekker and Verwer [1984, Chap. 9] and Hairer and Wanner [2010, Chap. 4, Sect. 7]. A criterion for A-stability of methods belonging to the function φ in (5.169) can be found in Crouzeix and Ruamps (1977).

In all the results previously described, it was tacitly assumed that the nonlinear systems of equations to be solved in an implicit Runge–Kutta method have a unique solution. This is not necessarily the case, not even for linear differential equations with constant coefficient matrix. Such questions of existence and uniqueness are considered in Dekker and Verwer [1984, Chap. 5], where one also finds a discussion of, and references to, the efficient implementation of implicit Runge–Kutta methods. Also see Hairer and Wanner [2010, Chap. 4, Sects. 14, 8].

The theory of consistency and convergence developed in Sect. 5.7 for nonstiff problems must be modified when dealing with stiff sytems of nonlinear differential equations. One-sided Lipschitz conditions are then the natural vehicles, and B-consistency and B-convergence the relevant concepts; see Dekker and Verwer [1984, Chap. 7] and Hairer and Wanner [2010, Chap. 4, Sect. 15].

Section 5.9.4. Regions of absolute stability for the embedded Runge–Kutta pairs 4(5) and the 7(8) pairs of Dormand and Prince (cf. Notes to Sect. 5.8.2(2)), and for the Gragg, Bulirsch, and Stoer extrapolation method (cf. Notes to Sect. 5.4), are displayed in Hairer and Wanner [2010, Chap. 4, Sect. 2]. Attempts to construct explicit Runge–Kutta formulae whose regions of absolute stability, along the negative real axis, extend to the left as far as possible lead to interesting applications of Chebyshev polynomials; see Hairer and Wanner [2010, pp. 31–36].

Exercises and Machine Assignments to Chapter 5

Exercises

1. Consider the initial value problem

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \kappa(y + y^3),\ 0 \le x \le 1;\ y(0) = s,$$

   where $\kappa > 0$ (in fact, $\kappa \gg 1$) and $s > 0$. Under what conditions on s does the solution $y(x) = y(x; s)$ exist on the whole interval $[0, 1]$? {*Hint*: find y explicitly.}
2. Prove (5.46).
3. Prove

$$(\boldsymbol{f}_y \boldsymbol{f})_y \boldsymbol{f} = \boldsymbol{f}^{\mathrm{T}} \boldsymbol{f}_{yy} \boldsymbol{f} + \boldsymbol{f}_y^2 \boldsymbol{f}.$$

4. Let

$$\boldsymbol{f}(x, \boldsymbol{y}) = \begin{bmatrix} f^1(x, \boldsymbol{y}) \\ f^2(x, \boldsymbol{y}) \\ \vdots \\ f^d(x, \boldsymbol{y}) \end{bmatrix}$$

   be a C^1 map from $[a, b] \times \mathbb{R}^d$ to $\mathbb{R}^d$. Assume that

$$\left| \frac{\partial f^i(x, \boldsymbol{y})}{\partial y^j} \right| \le M_{ij} \text{ on } [a, b] \times \mathbb{R}^d,\ i, j = 1, 2, \ldots, d,$$

   where M_{ij} are constants independent of x and $\boldsymbol{y}$, and let $\boldsymbol{M} = [M_{ij}] \in \mathbb{R}_+^{d \times d}$. Determine a Lipschitz constant L of $\boldsymbol{f}$:

   (a) in the ℓ_1 vector norm;
   (b) in the ℓ_2 vector norm;
   (c) in the ℓ_∞ vector norm.

   Express L, if possible, in terms of a matrix norm of $\boldsymbol{M}$.

5. (a) Write the system of differential equations

$$u''' = x^2 uu'' - uv',$$

$$v'' = xvv' + 4u'$$

as a first-order system of differential equations, $\boldsymbol{y}' = \boldsymbol{f}(x, \boldsymbol{y})$.

(b) Determine the Jacobian matrix $\boldsymbol{f}_{\boldsymbol{y}}(x, \boldsymbol{y})$ for the system in (a).

(c) Determine a Lipschitz constant L for $\boldsymbol{f}$ on $[0, 1] \times \mathcal{D}$, where $\mathcal{D} = \{\boldsymbol{y} \in \mathbb{R}^d : \|\boldsymbol{y}\|_1 \leq 1\}$, using, respectively, the ℓ_1, ℓ_2, and ℓ_∞ norm (cf. Ex. 4).

6. For the (scalar) differential equation

$$\frac{\mathrm{d}y}{\mathrm{d}x} = y^\lambda, \quad \lambda > 0,$$

(a) determine the principal error function of the general explicit two-stage Runge–Kutta method (5.56), (5.57);

(b) compare the local accuracy of the modified Euler method with that of Heun's method;

(c) determine a λ-interval such that for each λ in this interval, there is a two-stage explicit Runge–Kutta method of order $p = 3$ having parameters $0 < \alpha_1 < 1, 0 < \alpha_2 < 1$, and $0 < \mu < 1$.

7. For the implicit Euler method

$$\boldsymbol{y}_{\text{next}} = \boldsymbol{y} + h\boldsymbol{f}(x + h, \boldsymbol{y}_{\text{next}}),$$

(a) state a condition under which $\boldsymbol{y}_{\text{next}}$ is uniquely defined;

(b) determine the order and principal error function.

8. Show that any explicit two-stage Runge–Kutta method of order $p = 2$ integrates the special scalar differential equation $\mathrm{d}y/\mathrm{d}x = f(x)$, $f \in \mathbb{P}_1$, exactly.

9. The (scalar) second-order differential equation

$$\frac{\mathrm{d}^2 z}{\mathrm{d}x^2} = g(x, z),$$

in which g does not depend on $\mathrm{d}z/\mathrm{d}x$, can be written as a first-order system

$$\frac{\mathrm{d}}{\mathrm{d}x}\begin{bmatrix} y^1 \\ y^2 \end{bmatrix} = \begin{bmatrix} y^2 \\ g(x, y^1) \end{bmatrix}$$

by letting, as usual, $y^1 = z$, $y^2 = \mathrm{d}z/\mathrm{d}x$. For this system, consider a one-step method $\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + h\boldsymbol{\Phi}(x_n, \boldsymbol{u}_n; h)$ with

$$\boldsymbol{\Phi}(x, \boldsymbol{y}; h) = \begin{bmatrix} y^2 + \frac{1}{2}hk(x, \boldsymbol{y}; h) \\ k(x, \boldsymbol{y}; h) \end{bmatrix}, \; k = g(x+\mu h, y^1+\mu h y^2), \; \boldsymbol{y} = \begin{bmatrix} y^1 \\ y^2 \end{bmatrix}.$$

(Note that this method requires only one evaluation of g per step.)

(a) Can the method be made to have order $p = 2$, and if so, for what value(s) of μ?
(b) Determine the principal error function of any method obtained in (a).

10. Show that the first condition in (5.67) is equivalent to the condition that

$$\boldsymbol{k}_s(x, \boldsymbol{y}; h) = \boldsymbol{u}'(x + \mu_s h) + O(h^2), \quad s \geq 2,$$

where $\boldsymbol{u}(t)$ is the reference solution through the point $(x, \boldsymbol{y})$.

11. Suppose that

$$\int_x^{x+h} z(t)\mathrm{d}t = h\sum_{k=1}^{\nu} w_k z(x + \vartheta_k h) + ch^{\mu+1}z^{(\mu)}(\xi)$$

is a quadrature formula with $w_k \in \mathbb{R}$, $\vartheta_k \in [0, 1]$, $c \neq 0$, and $\xi \in (x, x+h)$, for z sufficiently smooth. Given increment functions $\overline{\boldsymbol{\Phi}}_k(x, \boldsymbol{y}; h)$ defining methods of order $\overline{p}_k$, $k = 1, 2, \ldots, \nu$, show that the one-step method defined by

$$\boldsymbol{\Phi}(x, \boldsymbol{y}; h) = \sum_{k=1}^{\nu} w_k \boldsymbol{f}(x + \vartheta_k h, \boldsymbol{y} + \vartheta_k h\overline{\boldsymbol{\Phi}}_k(x, \boldsymbol{y}; \vartheta_k h))$$

has order p at least equal to $\min(\mu, \overline{p} + 1)$, where $\overline{p} = \min \overline{p}_k$.

12. Let $\boldsymbol{g}(x, \boldsymbol{y}) = (\boldsymbol{f}_x + \boldsymbol{f}_y \boldsymbol{f})(x, \boldsymbol{y})$. Show that the one-step method defined by the increment function

$$\boldsymbol{\Phi}(x, \boldsymbol{y}; h) = \boldsymbol{f}(x, \boldsymbol{y}) + \tfrac{1}{2}h\boldsymbol{g}(x + \tfrac{1}{3}h, \boldsymbol{y} + \tfrac{1}{3}h\boldsymbol{f}(x, \boldsymbol{y}))$$

has order $p = 3$. Express the principal error function in terms of $\boldsymbol{g}$ and its derivatives.

13. Let $\boldsymbol{f}(x, \boldsymbol{y})$ satisfy a Lipschitz condition in $\boldsymbol{y}$ on $[a, b] \times \mathbb{R}^d$, with Lipschitz constant L.

(a) Show that the increment function $\boldsymbol{\Phi}$ of the second-order Runge–Kutta method

$$\begin{aligned} \boldsymbol{k}_1 &= \boldsymbol{f}(x, \boldsymbol{y}), \\ \boldsymbol{k}_2 &= \boldsymbol{f}(x + h, \boldsymbol{y} + h\boldsymbol{k}_1), \\ \boldsymbol{\Phi}(x, \boldsymbol{y}; h) &= \frac{1}{2}(\boldsymbol{k}_1 + \boldsymbol{k}_2) \end{aligned}$$

also satisfies a Lipschitz condition whenever $x + h \in [a, b]$, and determine a respective Lipschitz constant M.

(b) What would the result be for the classical Runge–Kutta method?

(c) What would it be for the general implicit Runge–Kutta method?

14. Describe the application of Newton's method to implement the implicit Runge–Kutta method.

15. Consider the following scheme of constructing an estimator $\boldsymbol{r}(x, \boldsymbol{y}; h)$ for the principal error function $\boldsymbol{\tau}(x, \boldsymbol{y})$ of Heun's method:

$$\begin{aligned}
\boldsymbol{k}_1 &= \boldsymbol{f}(x, \boldsymbol{y}),\\
\boldsymbol{k}_2 &= \boldsymbol{f}(x + h, \boldsymbol{y} + h\boldsymbol{k}_1),\\
\boldsymbol{y}_h &= \boldsymbol{y} + \frac{1}{2}h(\boldsymbol{k}_1 + \boldsymbol{k}_2),\\
\boldsymbol{k}_3 &= \boldsymbol{f}(x + h, \boldsymbol{y}_h),\\
\boldsymbol{k}_4 &= \boldsymbol{f}(x + h + \mu h, \boldsymbol{y}_h + \mu h\boldsymbol{k}_3),\\
\boldsymbol{r}(x, \boldsymbol{y}; h) &= h^{-2}(\beta_1\boldsymbol{k}_1 + \beta_2\boldsymbol{k}_2 + \beta_3\boldsymbol{k}_3 + \beta_4\boldsymbol{k}_4).
\end{aligned}$$

(Note that this scheme requires one additional function evaluation, $\boldsymbol{k}_4$, beyond what would be required anyhow to carry out Heun's method.) Obtain the conditions on the parameters μ, β_1, β_2, β_3, β_4 in order that

$$\boldsymbol{r}(x, \boldsymbol{y}; h) = \boldsymbol{\tau}(x, \boldsymbol{y}) + O(h).$$

Show, in particular, that there is a unique set of βs for any μ with $\mu(\mu+1) \neq 0$. What is a good choice of the parameters, and why?

16. Apply the asymptotic error formula (5.104) to the (scalar) initial value problem $\mathrm{d}y/\mathrm{d}x = \lambda y$, $y(0) = 1$, on $[0, 1]$, when solved by the classical fourth-order Runge–Kutta method. In particular, determine

$$\lim_{h\to 0} h^{-4}\frac{u_N - y(1)}{y(1)},$$

where u_N is the Runge–Kutta approximation to $y(1)$ obtained with step $h = 1/N$.

17. Consider $y' = \lambda y$ on $[0, \infty)$ for complex λ with $\operatorname{Re}\lambda < 0$. Let $\{u_n\}$ be the approximations to $\{y(x_n)\}$ obtained by the classical fourth-order Runge–Kutta method with the step h held fixed. (That is, $x_n = nh$, $h > 0$, and $n = 0, 1, 2, \ldots$.)

(a) Show that $y(x) \to 0$ as $x \to \infty$, for any initial value y_0.

(b) Under what condition on h can we assert that $u_n \to 0$ as $n \to \infty$? In particular, what is the condition if λ is real (negative)?

(c) What is the analogous result for Euler's method?

(d) Generalize to systems $\boldsymbol{y}' = \boldsymbol{A}\boldsymbol{y}$, where $\boldsymbol{A}$ is a constant matrix all of whose eigenvalues have negative real parts.

18. Show that any one-step method of order p, which, when applied to the model problem $\boldsymbol{y}' = \boldsymbol{A}\boldsymbol{y}$, yields

$$\boldsymbol{y}_{\text{next}} = \varphi(h\boldsymbol{A})\boldsymbol{y}, \quad \varphi \text{ a polynomial of degree } q \geq p,$$

must have
$$\varphi(z) = 1 + z + \frac{1}{2!}z^2 + \cdots + \frac{1}{p!}z^p + z^{p+1}\chi(z),$$
where χ is identically 0 if $q = p$, and a polynomial of degree $q - p - 1$ otherwise. In particular, show that $\chi \equiv 0$ for a p-stage explicit Runge–Kutta method of order p, $1 \leq p \leq 4$, and for the Taylor expansion method of order $p \geq 1$.

19. Consider the linear homogeneous system

$$(*) \qquad \boldsymbol{y}' = \boldsymbol{A}\boldsymbol{y}, \quad \boldsymbol{y} \in \mathbb{R}^d,$$

with constant coefficient matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d}$.

(a) For Euler's method applied to (*), determine $\varphi(z)$ (cf. (5.140)) and the principal error function.
(b) Do the same for the classical fourth-order Runge–Kutta method.

20. Consider the model equation

$$\frac{\mathrm{d}y}{\mathrm{d}x} = a(x)[y - b(x)], \quad 0 \leq x < \infty,$$

where $a(x)$, $b(x)$ are continuous and bounded on $\mathbb{R}_+$, and $a(x)$ negative with $|a(x)|$ large, say,

$$a \leq |a(x)| \leq A \text{ on } \mathbb{R}_+, a \gg 1.$$

For the explicit and implicit Euler methods, derive a condition (if any) on the step length h that ensures boundedness of the respective approximations u_n as $x_n = nh \to \infty$ for $h > 0$ fixed. (Assume, in the case of the explicit Euler method, that a is so large that $ah > 1$.)

21. Consider the implicit one-step method

$$\boldsymbol{\Phi}(x, \boldsymbol{y}; h) = \boldsymbol{k}(x, \boldsymbol{y}; h),$$

where $\boldsymbol{k} : [a, b] \times \mathbb{R}^d \times (0, h_0] \to \mathbb{R}^d$ is implicitly defined, in terms of total derivatives of $\boldsymbol{f}$, by

$$\boldsymbol{k} = \sum_{s=1}^{r} h^{s-1}[\alpha_s \boldsymbol{f}^{[s-1]}(x, \boldsymbol{y}) - \beta_s \boldsymbol{f}^{[s-1]}(x + h, \boldsymbol{y} + h\boldsymbol{k})],$$

with suitable constants α_s and β_s (Ehle's method; cf. Sect. 5.9.3(4)).

(a) Show how the method works on the model problem $\mathrm{d}y/\mathrm{d}x = \lambda y$. What is the maximum possible order in this case? Is the resulting method (of maximal order) A-stable?

(b) We may associate with the one-step method the quadrature rule

$$\int_x^{x+h} g(t)\mathrm{d}t = \sum_{s=1}^{r} h^s[\alpha_s g^{(s-1)}(x) - \beta_s g^{(s-1)}(x+h)] + E(g).$$

Given any p with $r \le p \le 2r$, show that α_s, β_s can be chosen so as to have $E(g) = O(h^{p+1})$ when $g(t) = \mathrm{e}^{t-x}$.

(c) With α_s, β_s chosen as in (b), prove that $E(g) = O(h^{p+1})$ for any $g \in C^p$ (not just for $g(t) = \mathrm{e}^{t-x}$). {*Hint*: expand $E(g)$ in powers of h through h^p inclusive; then specialize to $g(t) = \mathrm{e}^{t-x}$ and draw appropriate conclusions.}

(d) With α_s, β_s chosen as in (b), show that the implicit one-step method has order p if $\boldsymbol{f} \in C^p$. {*Hint*: use the definition of truncation error and Lipschitz conditions on the total derivatives $\boldsymbol{f}^{[s-1]}$.}

(e) Work out the optimal one-step method with $r = 2$ and order $p = 4$.

(f) How can you make the method L-stable (cf. (5.170)) and have maximum possible order? Illustrate with $r = 2$.

Machine Assignments

1. (a) Write Matlab routines implementing the basic step $(x, \boldsymbol{y}) \mapsto (x+h, \boldsymbol{y}_{\text{next}})$ in the case of Euler's method and the classical fourth-order Runge–Kutta method, entering the function $\boldsymbol{f}$ of the differential equation $\boldsymbol{y}' = \boldsymbol{f}(x, \boldsymbol{y})$ as an input function.

   (b) Consider the initial value problem

$$\boldsymbol{y}' = \boldsymbol{A}\boldsymbol{y},\ 0 \le x \le 1,\ \boldsymbol{y}(0) = \mathbf{1},$$

where

$$\boldsymbol{A} = \frac{1}{2}\begin{bmatrix} \lambda_2+\lambda_3 & \lambda_3-\lambda_1 & \lambda_2-\lambda_1 \\ \lambda_3-\lambda_2 & \lambda_1+\lambda_3 & \lambda_1-\lambda_2 \\ \lambda_2-\lambda_3 & \lambda_1-\lambda_3 & \lambda_1+\lambda_2 \end{bmatrix},\ \mathbf{1} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

The exact solution is

$$\boldsymbol{y}(x) = \begin{bmatrix} y^1 \\ y^2 \\ y^3 \end{bmatrix},\quad \begin{aligned} y^1 &= -\mathrm{e}^{\lambda_1 x} + \mathrm{e}^{\lambda_2 x} + \mathrm{e}^{\lambda_3 x}, \\ y^2 &= \mathrm{e}^{\lambda_1 x} - \mathrm{e}^{\lambda_2 x} + \mathrm{e}^{\lambda_3 x}, \\ y^3 &= \mathrm{e}^{\lambda_1 x} + \mathrm{e}^{\lambda_2 x} - \mathrm{e}^{\lambda_3 x}. \end{aligned}$$

Integrate the initial value problem with constant step length $h = 1/N$ by

(i) Euler's method (order $p = 1$);
(ii) the classical Runge–Kutta method (order $p = 4$),

using the programs written in (a). In each case, along with the approximation vectors $\boldsymbol{u}_n \in \mathbb{R}^3$, $n = 1, 2, \ldots, N$, generate vectors $\boldsymbol{v}_n \in \mathbb{R}^3$, $n = 1, 2, \ldots, N$, that approximate the solution of the variational equation according to Theorem 5.8.1. (For the estimate $\boldsymbol{r}(x, \boldsymbol{y}; h)$ of the principal error function take the true value $\boldsymbol{r}(x, \boldsymbol{y}; h) = \boldsymbol{\tau}(x, \boldsymbol{y})$ according to Ex. 19.) In this way obtain estimates $\tilde{\boldsymbol{e}}_n = h^p \boldsymbol{v}_n$ (p = order of the method) of the global errors $\boldsymbol{e}_n = \boldsymbol{u}_n - \boldsymbol{y}(x_n)$. Use $N = 5, 10, 20, 40, 80$, and print x_n, $\|\boldsymbol{e}_n\|_\infty$, and $\|\tilde{\boldsymbol{e}}_n\|_\infty$ for $x_n = .2 : .2 : 1$.

Suggested λ-values are

(i) $\lambda_1 = -1, \lambda_2 = 0, \lambda_3 = 1$;
(ii) $\lambda_1 = 0, \lambda_2 = -1, \lambda_3 = -10$;
(iii) $\lambda_1 = 0, \lambda_2 = -1, \lambda_3 = -40$;
(iv) $\lambda_1 = 0, \lambda_2 = -1, \lambda_3 = -160$.

Summarize what you learn from these examples and from others that you may wish to run.

2. Consider the initial value problem

$$y'' = \cos(xy),\ y(0) = 1,\ y'(0) = 0,\ 0 \le x \le 1.$$

(a) Does the solution $y(x)$ exist on the whole interval $0 \le x \le 1$? Explain.
(b) Use a computer algebra system, for example Maple, to determine the Maclaurin expansion of the solution $y(x)$ up to, and including, the term with x^{50}. Evaluate the expansion to 15 decimal digits for $x = 0.25 : 0.25 : 1.0$.
(c) Describe in detail the generic step of the classical fourth-order Runge–Kutta method applied to this problem.
(d) Use the fourth-order Runge–Kutta routine `RK4.m` of MA1(a), in conjunction with a function `fMAV_2.m` appropriate for this assignment, to produce approximations $\boldsymbol{u}_n \approx \boldsymbol{y}(x_n)$ at $x_n = n/N$, $n = 0, 1, 2, \ldots, N$, for $N = [4, 16, 64, 256]$. Print the results $y(x_n), y'(x_n)$ to 12 decimal places for $x_n = .25 : .25 : 1.0$, including the errors $e_n = |u_n^1 - y(x_n)|$. (Use the Taylor expansion of (b) to compute $y(x_n)$.) Plot the solution y, y' obtained with $N = 256$.

3. On the interval $[2\sqrt{q}, 2\sqrt{q} + 1]$, $q \ge 0$ an integer, consider the initial value problem (Fehlberg, 1968)

$$\frac{\mathrm{d}^2 c}{\mathrm{d}x^2} = -\pi^2 x^2 c - \pi \frac{s}{\sqrt{c^2 + s^2}},$$

$$\frac{\mathrm{d}^2 s}{\mathrm{d}x^2} = -\pi^2 x^2 s + \pi \frac{c}{\sqrt{c^2 + s^2}},$$

with initial conditions at $x = 2\sqrt{q}$ given by

$$c = 1, \frac{\mathrm{d}c}{\mathrm{d}x} = 0, \ s = 0, \ \frac{\mathrm{d}s}{\mathrm{d}x} = 2\pi\sqrt{q}.$$

(a) Show that the exact solution is

$$c(x) = \cos\left(\frac{\pi}{2}x^2\right), \ s(x) = \sin\left(\frac{\pi}{2}x^2\right).$$

(b) Write the problem as an initial value problem for a system of first-order differential equations.

(c) Consider the Runge–Kutta–Fehlberg (3, 4) pair $\boldsymbol{\Phi}$, $\boldsymbol{\Phi}^*$ given by

$$\begin{aligned}
\boldsymbol{k}_1 &= \boldsymbol{f}(x, \boldsymbol{y}),\\
\boldsymbol{k}_2 &= \boldsymbol{f}\left(x + \frac{2}{7}h, \boldsymbol{y} + \frac{2}{7}h\boldsymbol{k}_1\right),\\
\boldsymbol{k}_3 &= \boldsymbol{f}\left(x + \frac{7}{15}h, \boldsymbol{y} + \frac{77}{900}h\boldsymbol{k}_1 + \frac{343}{900}h\boldsymbol{k}_2\right),\\
\boldsymbol{k}_4 &= \boldsymbol{f}\left(x + \frac{35}{38}h, \boldsymbol{y} + \frac{805}{1444}h\boldsymbol{k}_1 - \frac{77175}{54872}h\boldsymbol{k}_2 + \frac{97125}{54872}h\boldsymbol{k}_3\right),\\
\boldsymbol{\Phi}(x, \boldsymbol{y}; h) &= \frac{79}{490}\boldsymbol{k}_1 + \frac{2175}{3626}\boldsymbol{k}_3 + \frac{2166}{9065}\boldsymbol{k}_4,
\end{aligned}$$

respectively

$$\begin{aligned}
&\boldsymbol{k}_1, \boldsymbol{k}_2, \boldsymbol{k}_3, \boldsymbol{k}_4 \text{ as previously},\\
&\boldsymbol{k}_5 = \boldsymbol{f}(x + h, \boldsymbol{y} + h\boldsymbol{\Phi}(x, \boldsymbol{y}; h)),\\
&\boldsymbol{\Phi}^*(x, \boldsymbol{y}; h) = \frac{229}{1470}\boldsymbol{k}_1 + \frac{1125}{1813}\boldsymbol{k}_3 + \frac{13718}{81585}\boldsymbol{k}_4 + \frac{1}{18}\boldsymbol{k}_5.
\end{aligned}$$

Solve the initial value problem in (b) for $q = 0(1)3$ by the method $\boldsymbol{\Phi}$, using constant step length $h = 0.2$. Repeat the integration with half the step length, and keep repeating (and halving the step) until $\max_n \|\boldsymbol{u}_n - \boldsymbol{y}(x_n)\|_\infty \le .5 \times 10^{-6}$, where $\boldsymbol{u}_n$, $\boldsymbol{y}(x_n)$ are the approximate resp. exact solution vectors at $x_n = 2\sqrt{q} + nh$. For each run print

$$q, \quad h, \quad \max_n \|\boldsymbol{u}_n - \boldsymbol{y}(x_n)\|_\infty, \qquad \max_n |c_n^2 + s_n^2 - 1|,$$

where c_n, s_n are the approximate values obtained for $c(x_n)$ resp. $s(x_n)$, and the maxima are taken over $n = 1, 2, \ldots, N$ with N such that $Nh = 1$.

(d) For the same values of q as in (c) and $h = 0.2, 0.1, 0.05, 0.025, 0.0125$, print the global and (estimated) local errors,

$$q, h, \|\boldsymbol{u}_{n+1} - \boldsymbol{y}(x_{n+1})\|_\infty,\ h\|\boldsymbol{\Phi}(x_n, \boldsymbol{u}_n; h) - \boldsymbol{\Phi}^*(x_n, \boldsymbol{u}_n; h)\|_\infty,$$

for $x_n = 0(.2).8$.

(e) Implement Theorem 5.8.1 on global error estimation, using the Runge–Kutta–Fehlberg $(3, 4)$ method $\boldsymbol{\Phi}$, $\boldsymbol{\Phi}^*$ of (c) and the estimator $\boldsymbol{r}(x, \boldsymbol{y}; h) = h^{-3}[\boldsymbol{\Phi}(x, \boldsymbol{y}; h) - \boldsymbol{\Phi}^*(x, \boldsymbol{y}; h)]$ of the principal error function of $\boldsymbol{\Phi}$. For the same values of q as in (d), and for $h = 0.05, 0.025, 0.0125$, print the exact and estimated global errors,

$$q, \quad h, \quad \|\boldsymbol{u}_{n+1} - \boldsymbol{y}(x_{n+1})\|_\infty, \quad h^3\|\boldsymbol{v}_{n+1}\|_\infty \quad \text{for} \quad x_n = 0 : .2 : .8.$$

4. (a) Let $f(z) = 1 + \frac{1}{1!}z + \frac{1}{2!}z^2 + \cdots + \frac{1}{p!}z^p$. For $p = 1(1)4$ write a Matlab program, using the `contour` command, to plot the lines along which $|f(z)| = r$, $r = 0.1(0.1)1$ (level lines of f) and the lines along which $\arg f(z) = \theta$, $\theta = 0(\frac{1}{8}\pi)2\pi - \frac{1}{8}\pi$ (phase lines of f).

(b) For any analytic function f, derive differential equations for the level and phase lines of f. {*Hint*: write $f(z) = r\exp(\mathrm{i}\theta)$ and use θ as the independent variable for the level lines, and r as the independent variable for the phase lines. In each case, introduce arc length as the final independent variable.}

(c) Use the Matlab function `ode45` to compute from the differential equation of (b) the level lines $|f(z)| = 1$ of the function f given in (a), for $p = 1(1)21$; these determine the regions of absolute stability of the Taylor expansion method (cf. Ex. 18). {*Hint*: use initial conditions at the origin. Produce only those parts of the curves that lie in the upper half-plane (why?). To do so in Matlab, let `ode45` run sufficiently long, interpolate between the first pair of points lying on opposite sides of the real axis to get a point on the axis, and then delete the rest of the data before plotting.}

5. Newton's equations for the motion of a particle on a planar orbit (with eccentricity ε, $0 < \varepsilon < 1$) are

$$\begin{aligned} x'' &= -\frac{x}{r^3},\ x(0) = 1 - \varepsilon,\ x'(0) = 0, \\ & \qquad\qquad\qquad\qquad\qquad\qquad t \ge 0, \\ y'' &= -\frac{y}{r^3},\ y(0) = 0,\ y'(0) = \sqrt{\frac{1+\varepsilon}{1-\varepsilon}}, \end{aligned}$$

where

$$r^2 = x^2 + y^2.$$

(a) Verify that the solution can be written in the form $x(t) = \cos u - \varepsilon$, $y(t) = \sqrt{1-\varepsilon^2}\sin u$, where $u = u(t)$ is the solution of Kepler's equation $u - \varepsilon\sin u - t = 0$.

(b) Reformulate the problem as an initial value problem for a system of first-order differential equations.

(c) Write a Matlab program for solving the initial value problem in (b) on the interval $[0, 20]$ by the classical Runge–Kutta method, for $\varepsilon = 0.3, 0.5$, and 0.7. Use step lengths $h = 1/N$, $N = [40, 80, 120]$ and, along with the approximate solution $u(t; h)$, $v(t; h)$, compute and plot the approximate principal error functions $r(t; h) = h^{-4}[u(t; h) - x(t)]$, $s(t; h) = h^{-4}[v(t; h) - y(t)]$ when $N = 120$ (i.e., $h = .008333\ldots$). Compute the exact solution from the formula given in (a), using Newton's method to solve Kepler's equation.

Selected Solutions to Exercises

15. We know from (5.64) (where $\alpha_2 = 1/2$) that

$$\boldsymbol{\tau}(x, \boldsymbol{y}) = \frac{1}{2}\left[\frac{1}{6}(\boldsymbol{f}_{xx} + 2\boldsymbol{f}_{xy}\boldsymbol{f} + \boldsymbol{f}^{\mathrm{T}}\boldsymbol{f}_{yy}\boldsymbol{f}) - \frac{1}{3}\boldsymbol{f}_y(\boldsymbol{f}_x + \boldsymbol{f}_y\boldsymbol{f})\right],$$

and from (5.59) (where $\mu = 1$) that

$$\boldsymbol{k}_2 = \boldsymbol{f} + h(\boldsymbol{f}_x + \boldsymbol{f}_y\boldsymbol{f}) + \frac{1}{2}h^2(\boldsymbol{f}_{xx} + 2\boldsymbol{f}_{xy}\boldsymbol{f} + \boldsymbol{f}^{\mathrm{T}}\boldsymbol{f}_{yy}\boldsymbol{f}) + O(h^3).$$

For $\boldsymbol{k}_3$ and $\boldsymbol{k}_4$, one finds similarly that

$$\begin{aligned}\boldsymbol{k}_3 &= \boldsymbol{f} + h(\boldsymbol{f}_x + \boldsymbol{f}_y\boldsymbol{f}) + \frac{1}{2}h^2[\boldsymbol{f}_{xx} + 2\boldsymbol{f}_{xy}\boldsymbol{f} + \boldsymbol{f}^{\mathrm{T}}\boldsymbol{f}_{yy}\boldsymbol{f} \\ &\quad + \boldsymbol{f}_y(\boldsymbol{f}_x + \boldsymbol{f}_y\boldsymbol{f})] + O(h^3), \\ \boldsymbol{k}_4 &= \boldsymbol{f} + (\mu+1)h(\boldsymbol{f}_x + \boldsymbol{f}_y\boldsymbol{f}) + \frac{1}{2}h^2[(\mu+1)^2(\boldsymbol{f}_{xx} + 2\boldsymbol{f}_{xy}\boldsymbol{f} + \boldsymbol{f}^{\mathrm{T}}\boldsymbol{f}_{yy}\boldsymbol{f}) \\ &\quad + (2\mu+1)\boldsymbol{f}_y(\boldsymbol{f}_x + \boldsymbol{f}_y\boldsymbol{f})] + O(h^3).\end{aligned}$$

For $h^{-2}(\beta_1\boldsymbol{k}_1 + \beta_2\boldsymbol{k}_2 + \beta_3\boldsymbol{k}_3 + \beta_4\boldsymbol{k}_4)$ to approximate $\boldsymbol{\tau}(x, \boldsymbol{y})$ to within $O(h)$, that is,

$$\begin{aligned}&h^{-2}(\beta_1 + \beta_2 + \beta_3 + \beta_4)\boldsymbol{f} + h^{-1}[\beta_2 + \beta_3 + (\mu+1)\beta_4](\boldsymbol{f}_x + \boldsymbol{f}_y\boldsymbol{f}) \\ &\quad + \frac{1}{2}[(\beta_2 + \beta_3 + (\mu+1)^2\beta_4)(\boldsymbol{f}_{xx} + 2\boldsymbol{f}_{xy}\boldsymbol{f} + \boldsymbol{f}^T\boldsymbol{f}_{yy}\boldsymbol{f}) \\ &\quad + (\beta_3 + (2\mu+1)\beta_4)\boldsymbol{f}_y(\boldsymbol{f}_x + \boldsymbol{f}_y\boldsymbol{f})] + O(h) = \boldsymbol{\tau}(x, \boldsymbol{y}) + O(h),\end{aligned}$$

requires that

$$\begin{aligned} \beta_1 + \beta_2 + \beta_3 + \beta_4 &= 0, \\ \beta_2 + \beta_3 + (\mu + 1)\beta_4 &= 0, \\ \beta_2 + \beta_3 + (\mu + 1)^2\beta_4 &= \frac{1}{6}, \\ \beta_3 + (2\mu + 1)\beta_4 &= -\frac{1}{3}. \end{aligned}$$

The determinant of this system is

$$\begin{vmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & \mu+1 \\ 0 & 1 & 1 & (\mu+1)^2 \\ 0 & 0 & 1 & 2\mu+1 \end{vmatrix} = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & \mu+1 \\ 0 & 0 & 0 & \mu(\mu+1) \\ 0 & 0 & 1 & 2\mu+1 \end{vmatrix} = -\mu(\mu + 1),$$

so that the condition $\mu(\mu+1) \neq 0$ ensures a unique solution. A good solution is the one corresponding to $\mu = 1$, since then no extra evaluation of $\boldsymbol{f}$ is needed, $\boldsymbol{k}_3$ and $\boldsymbol{k}_4$ being the evaluations needed anyhow to execute the second step of Heun's method. In this case, we get

$$\mu = 1, \beta_1 = \frac{1}{12}, \beta_2 = \frac{5}{12}, \beta_3 = -\frac{7}{12}, \beta_4 = \frac{1}{12}.$$

(For literature, see L.F. Shampine and H.A. Watts, Computing error estimates for Runge–Kutta methods, *Math. Comp. 25* (1971), 445–455.)

21. (a) If $y' = \lambda y$, then $f(x, y) = \lambda y$, $f^{[s-1]}(x, y) = \lambda^s y$, and

$$\begin{aligned} k &= \sum_{s=1}^{r} h^{s-1}[\alpha_s \lambda^s y - \beta_s \lambda^s (y + hk)] \\ &= \frac{1}{h}\sum_{s=1}^{r}(\alpha_s - \beta_s)(\lambda h)^s y - \sum_{s=1}^{r}\beta_s(\lambda h)^s k. \end{aligned}$$

Thus,

$$\left(1 + \sum_{s=1}^{r}\beta_s(\lambda h)^s\right) k = \frac{1}{h}\sum_{s=1}^{r}(\alpha_s - \beta_s)(\lambda h)^s y.$$

Letting $z = \lambda h$, we get

$$k(x, y; h) = \frac{1}{h}\,\frac{\sum_{s=1}^{r}(\alpha_s - \beta_s)z^s}{1 + \sum_{s=1}^{r}\beta_s z^s}\, y.$$

There follows

$$y_{\text{next}} = y + hk(x, y; h) = \left\{ 1 + \frac{\sum_{s=1}^{r} (\alpha_s - \beta_s) z^s}{1 + \sum_{s=1}^{r} \beta_s z^s} \right\} y$$
$$= \frac{1 + \sum_{s=1}^{r} \alpha_s z^s}{1 + \sum_{s=1}^{r} \beta_s z^s} y,$$

that is,

$$\varphi(z) = \frac{1 + \sum_{s=1}^{r} \alpha_s z^s}{1 + \sum_{s=1}^{r} \beta_s z^s}.$$

The method has order p if and only if

$$\mathrm{e}^z = \frac{1 + \sum_{s=1}^{r} \alpha_s z^s}{1 + \sum_{s=1}^{r} \beta_s z^s} + O(z^{p+1}), \quad z \to 0.$$

The maximum possible order is $p = 2r$ and is obtained by choosing the α_s and β_s so that φ is the Padé approximant $R[r, r]$ to the exponential function. The resulting method is A-stable by Theorem 5.9.2.

(b) For $g(t) = \mathrm{e}^{t-x}$, we have $g^{(s-1)}(t) = \mathrm{e}^{t-x}$, so that the quadrature formula becomes

$$\int_x^{x+h} \mathrm{e}^{t-x} \mathrm{d}t = \mathrm{e}^h - 1 = \sum_{s=1}^{r} h^s (\alpha_s - \beta_s \mathrm{e}^h) + E(g).$$

Therefore,

$$E(g) = \mathrm{e}^h - 1 - \sum_{s=1}^{r} \alpha_s h^s + \mathrm{e}^h \sum_{s=1}^{r} \beta_s h^s$$
$$= \left(1 + \sum_{s=1}^{r} \beta_s h^s\right) \mathrm{e}^h - \left(1 + \sum_{s=1}^{r} \alpha_s h^s\right)$$
$$= \left(1 + \sum_{s=1}^{r} \beta_s h^s\right) \left\{ \mathrm{e}^h - \frac{1 + \sum_{s=1}^{r} \alpha_s h^s}{1 + \sum_{s=1}^{r} \beta_s h^s} \right\}.$$

Given any p with $r \le p \le 2r$, we can always choose nonnegative integers $m \le r$ and $n \le r$ such that $p = n + m$. Setting $\alpha_{m+1} = \cdots = \alpha_r = 0$, $\beta_{n+1} = \cdots = \beta_r = 0$, and choosing the remaining parameters so that the rational function in braces becomes the $R[n, m]$-Padé approximant to the exponential function makes the expression in braces of order $O(h^{n+m+1}) = O(h^{p+1})$, that is, $E(g) = O(h^{p+1})$.

Clearly, $p = 2r$ is optimal, in which case $m = n = r$, and we get uniquely the $R[r, r]$-Padé approximant. If $p < 2r$, there are more than one choice for m and n.

(c) Assuming $g \in C^p$, we have the expansions

$$\int_x^{x+h} g(t)\mathrm{d}t = hg(x) + \frac{h^2}{2!}g'(x) + \cdots + \frac{h^p}{p!}g^{(p-1)}(x) + O(h^{p+1}),$$

$$h^s g^{(s-1)}(x+h) = h^s g^{(s-1)}(x) + \frac{h^{s+1}}{1!}g^{(s)}(x) + \cdots + \frac{h^p}{(p-s)!}g^{(p-1)}(x) + O(h^{p+1}).$$

Substitution into the quadrature formula yields an expansion of the type

$$E(g) = e_1 hg(x) + e_2 h^2 g'(x) + \cdots + e_p h^p g^{(p-1)}(x) + O(h^{p+1}),$$

with certain constants e_i independent of h and g. Now put $g(t) = \mathrm{e}^{t-x}$. Since $g^{(i)}(x) = 1$ for all $i \geq 0$, we get

$$E(\mathrm{e}^{t-x}) = e_1 h + e_2 h^2 + \cdots + e_p h^p + O(h^{p+1}).$$

Since the quadrature rule has been chosen such that $E(\mathrm{e}^{t-x}) = O(h^{p+1})$ (cf. (b)), it follows that, necessarily, $e_1 = e_2 = \cdots = e_p = 0$. Hence, $E(g) = O(h^{p+1})$ for any $g \in C^p[x, x+h]$.

(d) We have for the truncation error of the method,

$$\boldsymbol{T}(x, \boldsymbol{y}; h) = \boldsymbol{k}(x, \boldsymbol{y}; h) - \frac{1}{h}\left[\boldsymbol{u}(x+h) - \boldsymbol{u}(x)\right],$$

with $\boldsymbol{u}(t)$ the reference solution through $(x, \boldsymbol{y})$. Since by assumption $\boldsymbol{f} \in C^p$, we have $\boldsymbol{u} \in C^{p+1}$. Put $g(t) = \boldsymbol{u}'(t)$ in the quadrature formula, divide by h, and use (c) to get

$$\frac{1}{h}\left[\boldsymbol{u}(x+h) - \boldsymbol{u}(x)\right] = \sum_{s=1}^{r} h^{s-1}\big[\alpha_s \boldsymbol{u}^{(s)}(x) - \beta_s \boldsymbol{u}^{(s)}(x+h)\big] + O(h^p).$$

Therefore, since $\boldsymbol{f}^{[s-1]}(x, \boldsymbol{y}) = \boldsymbol{u}^{(s)}(x)$ (cf. (5.47)), we obtain

$$\boldsymbol{T}(x, \boldsymbol{y}; h) = \sum_{s=1}^{r} h^{s-1}[\alpha_s \boldsymbol{u}^{(s)}(x) - \beta_s \boldsymbol{f}^{[s-1]}(x+h, \boldsymbol{y} + h\boldsymbol{k})] - \sum_{s=1}^{r} h^{s-1}[\alpha_s \boldsymbol{u}^{(s)}(x) - \beta_s \boldsymbol{u}^{(s)}(x+h)] + O(h^p)$$

$$= -\sum_{s=1}^{r} h^{s-1}\beta_s[\boldsymbol{f}^{[s-1]}(x+h, \boldsymbol{y}+h\boldsymbol{k})$$
$$- \boldsymbol{f}^{[s-1]}(x+h, \boldsymbol{u}(x+h))] + O(h^p),$$

where we have used $\boldsymbol{u}^{(s)}(t) = \boldsymbol{f}^{[s-1]}(t, \boldsymbol{u}(t))$ (cf. (5.46)). With L_{s-1} denoting a Lipschitz constant for $\boldsymbol{f}^{[s-1]}$, the expression in brackets is bounded in norm by

$$L_{s-1}\|\boldsymbol{y} + h\boldsymbol{k}(x,\boldsymbol{y};h) - \boldsymbol{u}(x+h)\|$$
$$= hL_{s-1}\left\|\boldsymbol{k}(x,\boldsymbol{y};h) - \frac{1}{h}[\boldsymbol{u}(x+h) - \boldsymbol{u}(x)]\right\| = hL_{s-1}\|\boldsymbol{T}(x,\boldsymbol{y};h)\|$$

since $\boldsymbol{u}(x) = \boldsymbol{y}$. There follows

$$\|\boldsymbol{T}(x,\boldsymbol{y};h)\| \le \left(\sum_{s=1}^{r} h^s|\beta_s|L_{s-1}\right)\|\boldsymbol{T}(x,\boldsymbol{y};h)\| + O(h^p),$$

that is,

$$[1 - O(h)]\|\boldsymbol{T}(x,\boldsymbol{y};h)\| = O(h^p), \;\; \|\boldsymbol{T}(x,\boldsymbol{y};h)\| = O(h^p).$$

(e) The [2,2]-Padé approximant to the exponential function is (cf. (5.149), (5.150))

$$R[2,2](z) = \frac{1 + \frac{1}{2}z + \frac{1}{12}z^2}{1 - \frac{1}{2}z + \frac{1}{12}z^2}.$$

Therefore, according to (b),

$$\alpha_1 = \frac{1}{2},\;\; \alpha_2 = \frac{1}{12};\;\;\; \beta_1 = -\frac{1}{2},\;\; \beta_2 = \frac{1}{12}.$$

(f) Choose for $\varphi(z)$ the Padé approximant $R[r, r-1](z)$ by setting $\alpha_r = 0$. The (maximum) order is then $p = 2r - 1$. For $r = 2$, this becomes

$$R[2,1] = \frac{1 + \frac{1}{3}z}{1 - \frac{2}{3}z + \frac{1}{6}z^2},$$

so that

$$\alpha_1 = \frac{1}{3},\;\; \alpha_2 = 0;\;\;\; \beta_1 = -\frac{2}{3},\;\; \beta_2 = \frac{1}{6}.$$

Selected Solutions to Machine Assignments

2. (a) Write $y' = z$, so that the initial value problem can be written in vector form as

$$\frac{\mathrm{d}}{\mathrm{d}x}\begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} z \\ \cos(xy) \end{bmatrix}, \quad \begin{bmatrix} y \\ z \end{bmatrix}(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad 0 \le x \le 1.$$

We claim that the right-hand side of the system satisfies a uniform Lipschitz condition on $[0,1] \times \mathbb{R}^2$ in the L_1-norm, with Lipschitz constant $L = 1$. Indeed,

$$\left\| \begin{matrix} z - z^* \\ \cos(xy) - \cos(xy^*) \end{matrix} \right\|_1 = \left\| \begin{matrix} z - z^* \\ -2\sin(x\frac{y+y^*}{2})\sin(x\frac{y-y^*}{2}) \end{matrix} \right\|_1$$

$$\le |z - z^*| + 2|x|\frac{|y - y^*|}{2} \le |y - y^*| + |z - z^*| = \left\| \begin{matrix} y - y^* \\ z - z^* \end{matrix} \right\|_1,$$

as claimed. By Theorem 5.3.1, therefore, the initial value problem has a unique solution, which exists on all of [0.1].

(b) The following Maple program produces the Taylor expansion of the solution at $x = 0$ to 50 terms and evaluates it for $x = .25 : .25 : 1$ to 15 digits.

```
eq:=diff(y(x),x,x)-cos(x*y(x))=0;
ini:=y(0)=1, D(y)(0)=0;
Order:=50;
sol:=dsolve({eq,ini},{y(x)},type=series);
p:=convert(sol,polynom);
Digits:=15;
for x from .25 to 1 by .25 do p(x) od;
```

The result

$$y(.25) = 1.03108351021039,$$

$$y(.50) = 1.12215798572506,$$

$$y(.75) = 1.26540089160147,$$

$$y(1.0) = 1.44401698100709,$$

confirmed by another Maple run with 60 terms, is used to determine the error in the Matlab routine `MAV_2D` below.

(c) The classical fourth-order Runge–Kutta method, for the first-order system, emanating at the generic point $\left(x, \begin{bmatrix} y \\ z \end{bmatrix}\right)$, proceeds as follows:

$$\begin{bmatrix} k_1 \\ \ell_1 \end{bmatrix} = \begin{bmatrix} z \\ \cos(xy) \end{bmatrix},$$

$$\begin{bmatrix} k_2 \\ \ell_2 \end{bmatrix} = \begin{bmatrix} z + \frac{1}{2}h\ell_1 \\ \cos((x + \frac{1}{2}h)(y + \frac{1}{2}hk_1)) \end{bmatrix},$$

$$\begin{bmatrix} k_3 \\ \ell_3 \end{bmatrix} = \begin{bmatrix} z + \frac{1}{2}h\ell_2 \\ \cos((x + \frac{1}{2}h)(y + \frac{1}{2}hk_2)) \end{bmatrix},$$

$$\begin{bmatrix} k_4 \\ \ell_4 \end{bmatrix} = \begin{bmatrix} z + h\ell_3 \\ \cos((x + h)(y + hk_3)) \end{bmatrix},$$

$$\begin{bmatrix} y \\ z \end{bmatrix}_{\text{next}} = \begin{bmatrix} y \\ z \end{bmatrix} + \frac{1}{6}h\left(\begin{bmatrix} k_1 \\ \ell_1 \end{bmatrix} + 2\begin{bmatrix} k_2 \\ \ell_2 \end{bmatrix} + 2\begin{bmatrix} k_3 \\ \ell_3 \end{bmatrix} + \begin{bmatrix} k_4 \\ \ell_4 \end{bmatrix}\right).$$

(d) PROGRAM

For the routine RK4.m, see the answer to MA 1(a).

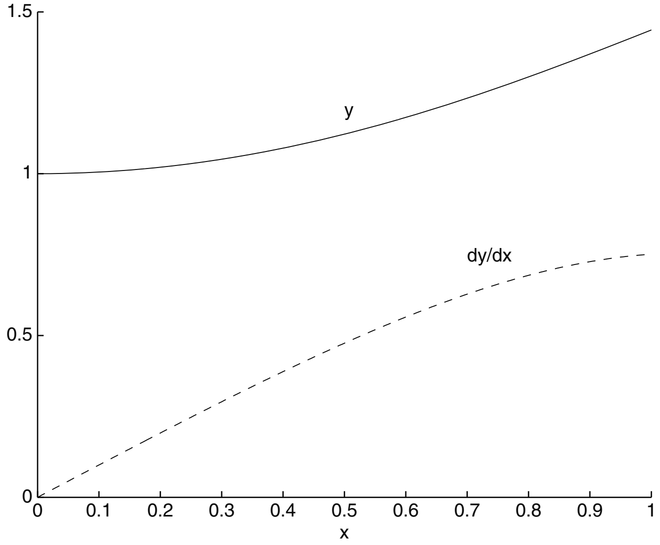
```
%MAV_2D
%
f0='%8.2f %16.12f %15.12f %12.4e  N=%3.0f\n';
f1='%8.2f %16.12f %15.12f %12.4e\n';
exact=[1.03108351021039;1.12215798572506; ...
1.26540089160147;1.44401698100709];
disp(['     x          y           dy/dx' ...
'          error'])
for N=[4 16 64 256]
  h=1/N;
  y=zeros(N+1,1); y1=zeros(N+1,1);
  y(1)=1; y1(1)=0; u=[1;0];
  ip=0;
  for n=1:N
    x=(n-1)/N;
    u=RK4(@fMAV_2,x,u,h);
    y(n+1)=u(1); y1(n+1)=u(2);
    if 4*n/N-fix(4*n/N)==0
      ip=ip+1;
      yexact=exact(ip); err=abs(u(1)-yexact);
      if n==N/4
```

```
        fprintf(f0,n/N,u(1),u(2),err,N)
      else
        fprintf(f1,n/N,u(1),u(2),err)
      end
    end
  end
  fprintf('\n')
  if N==256
    x=(0:N)'/N;
    hold on
    plot(x,y); plot(x,y1,'--');
    xlabel('x')
    text(.5,1.2,'y','FontSize',14)
    text(.7,.75,'dy/dx','FontSize',14)
    hold off
  end
end

%FMAV_2  Differential equation for MAV_2
%
function yprime=fMAV_2(x,y)
yprime=[y(2);cos(x*y(1))];
```

OUTPUT

```
>> MAV_2D
   x          y             dy/dx         error
  0.25  1.031084895175  0.247302726779  1.3850e-06  N=  4
  0.50  1.122162986649  0.476315286524  5.0009e-06
  0.75  1.265408064509  0.658662664483  7.1729e-06
  1.00  1.444014056895  0.751268242944  2.9241e-06

  0.25  1.031083515581  0.247306326743  5.3703e-09  N= 16
  0.50  1.122158005046  0.476319854529  1.9320e-08
  0.75  1.265400918802  0.658656442176  2.7201e-08
  1.00  1.444016971240  0.751230496912  9.7675e-09

  0.25  1.031083510231  0.247306340242  2.1031e-11  N= 64
  0.50  1.122157985801  0.476319869139  7.5974e-11
  0.75  1.265400891710  0.658656406501  1.0847e-10
  1.00  1.444016980977  0.751230324680  3.0409e-11

  0.25  1.031083510210  0.247306340295  8.1268e-14  N=256
  0.50  1.122157985725  0.476319869194  2.9399e-13
  0.75  1.265400891602  0.658656406353  4.2588e-13
  1.00  1.444016981007  0.751230323984  1.1346e-13

>>
```

PLOT



5. (a) First differentiate

$$u(t) - \varepsilon \sin u(t) - t = 0$$

to obtain

$$u'(t) = \frac{1}{1 - \varepsilon \cos u}.$$

Letting $x(t) = \cos u - \varepsilon$, $y(t) = \sqrt{1-\varepsilon^2}\sin u$, we then have

$$x'(t) = \frac{-\sin u}{1 - \varepsilon \cos u}, \quad x''(t) = \frac{-\cos u + \varepsilon}{(1 - \varepsilon \cos u)^3} = -\frac{x(t)}{(1 - \varepsilon \cos u)^3},$$
$$y'(t) = \sqrt{1-\varepsilon^2}\frac{\cos u}{1 - \varepsilon \cos u}, \quad y''(t) = -\frac{\sqrt{1-\varepsilon^2}\sin u}{(1 - \varepsilon \cos u)^3} = -\frac{y(t)}{1 - \varepsilon \cos u)^3}.$$

Since, on the other hand,

$$x^2(t) + y^2(t) = (\cos u - \varepsilon)^2 + (1 - \varepsilon^2)\sin^2 u = (1 - \varepsilon \cos u)^2,$$

we have $r = 1 - \varepsilon \cos u$, and thus

$$x'' + \frac{x}{r^3} = -\frac{x}{r^3} + \frac{x}{r^3} = 0,$$

$$y'' + \frac{y}{r^3} = -\frac{y}{r^3} + \frac{y}{r^3} = 0,$$

showing that the differential equations are satisfied. So are the initial conditions, since $u(0) = 0$, and thus

$$x(0) = 1 - \varepsilon, x'(0) = 0; \;\; y(0) = 0, y'(0) = \frac{\sqrt{1-\varepsilon^2}}{1-\varepsilon} = \sqrt{\frac{1+\varepsilon}{1-\varepsilon}}.$$

(b) Let $u = x, v = x', w = y, z = y'$. Then the first-order system is

$$\frac{\mathrm{d}u}{\mathrm{d}t} = v, \quad u(0) = 1 - \varepsilon,$$

$$\frac{\mathrm{d}v}{\mathrm{d}t} = -\frac{u}{r^3}, \quad v(0) = 0,$$

$$\frac{\mathrm{d}w}{\mathrm{d}t} = z, \quad w(0) = 0,$$

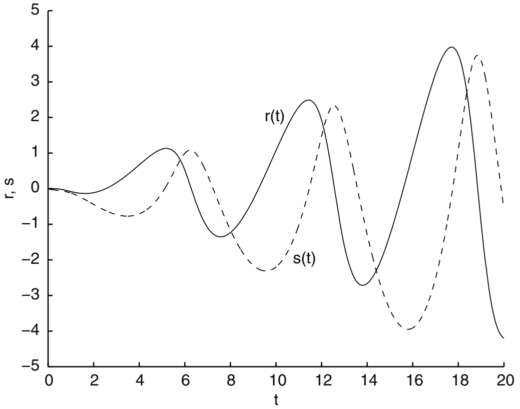
$$\frac{\mathrm{d}z}{\mathrm{d}t} = -\frac{w}{r^3}, \quad z(0) = \sqrt{\frac{1+\varepsilon}{1-\varepsilon}},$$

where $r^2 = u^2 + w^2$.

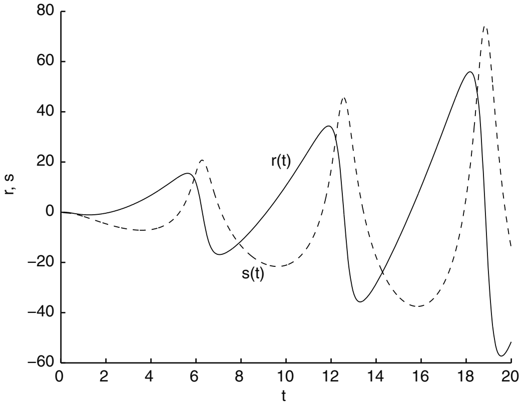
(c) PROGRAM

```
%MAV_5C
%
e=.3;
%e=.5;
%e=.7;
eps0=.5e-12;
for N=[40 80 120]
  h=1/N; u1=[1-e;0;0;sqrt((1+e)/(1-e))]; v1=0;
  r=zeros(20*N,1); s=zeros(20*N,1);
  for n=0:20*N-1
    u=u1; v=v1; t=n*h; t1=t+h;
    u1=RK4(@fMAV_5C,t,u,h);
    v1=kepler(t1,v,e,eps0);
    y=[cos(v1)-e;sqrt(1-e^2)*sin(v1)];
    pr=h^(-4)*(u1(1:2)-y);
    x(n+1)=t1; r(n+1)=pr(1); s(n+1)=pr(2);
  end
  if N==120
    hold on
    plot(x,r);
    plot(x,s,'--');
    axis([0 20 -5 5])     % for e=.3
%    axis([0 20 -60 80])   % for e=.5
%    axis([0 20 -4000 7000]) % for e=.7
    xlabel('t')
    ylabel('r, s')
    text(9.5,2,'r(t)')    % for e=.3
%    text(10.75,-2,'s(t)') % for e=.5
%    text(10.5,1500,'r(t)')  % for e=.7
    text(9.5,20,'r(t)')   % for e=.3
%    text(8.2,-25,'s(t)')  % for e=.5
%    text(9,-1000,'s(t)')  % for e=.7
    hold off
  end
end
```

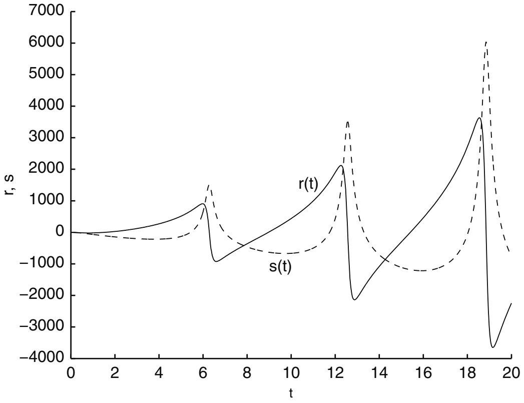
PLOTS



ε=.3



$\varepsilon = .5$



$\varepsilon = .7$

Chapter 6
Initial Value Problems for ODEs: Multistep Methods

We saw in Chap. 5 that (explicit) one-step methods are increasingly difficult to construct as one upgrades the order requirement. This is no longer true for multistep methods, where an increase in order is straightforward but comes with a price: a potential danger of instability. In addition, there are other complications such as the need for an initialization procedure and considerably more complicated procedures for changing the grid length. Yet, in terms of work involved, multistep methods are still among the most attractive methods. We discuss them along lines similar to one-step methods, beginning with a local description and examples and proceeding to the global description and problems of stiffness. By the very nature of multistep methods, the discussion of stability is now more extensive.

6.1 Local Description of Multistep Methods

6.1.1 Explicit and Implicit Methods

We consider as before the initial value problem for a first-order system of differential equations

$$\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}x} = \boldsymbol{f}(x, \boldsymbol{y}), \quad a \le x \le b; \quad \boldsymbol{y}(a) = \boldsymbol{y}_0 \tag{6.1}$$

(cf. Chap. 5, (5.14)–(5.16)). Our task is again to determine a vector-valued grid function $\boldsymbol{u} \in \Gamma_h[a, b]$ (cf. Chap. 5, Sect. 5.7) such that $\boldsymbol{u}_n \approx \boldsymbol{y}(x_n)$ at the nth grid point x_n.

A *k-step method* ($k > 1$) obtains $\boldsymbol{u}_{n+k}$ in terms of k preceding approximations $\boldsymbol{u}_{n+k-1}, \boldsymbol{u}_{n+k-2}, \dots, \boldsymbol{u}_n$. We call k the *step number* (or *index*) of the method.

W. Gautschi, *Numerical Analysis*, DOI 10.1007/978-0-8176-8259-0_6,

We consider only *linear* k-step methods, which, in their most general form, but assuming a constant grid length h, can be written as

$$\begin{aligned}&\boldsymbol{u}_{n+k} + \alpha_{k-1}\boldsymbol{u}_{n+k-1} + \cdots + \alpha_0\boldsymbol{u}_n\\ &\quad = h[\beta_k \boldsymbol{f}_{n+k} + \beta_{k-1}\boldsymbol{f}_{n+k-1} + \cdots + \beta_0\boldsymbol{f}_n],\ \ n = 0, 1, 2, \ldots, N-k, \end{aligned} \tag{6.2}$$

where

$$x_r = a + rh,\ \ \boldsymbol{f}_r = \boldsymbol{f}(x_r, \boldsymbol{u}_r), \quad r = 0, 1, \ldots, N, \tag{6.3}$$

and the α and β are given (scalar) coefficients. The relation (6.2) is linear in the function values $\boldsymbol{f}_r$ (in contrast to Runge–Kutta methods); nevertheless, we are still dealing with a nonlinear difference equation for the grid function $\boldsymbol{u}$.

The definition (6.2) must be supplemented by a *starting procedure* for obtaining the approximations to $\boldsymbol{y}(x_s)$,

$$\boldsymbol{u}_s = \boldsymbol{u}_s(h), \quad s = 0, 1, \ldots, k-1. \tag{6.4}$$

These normally depend on the grid length h, so may also the coefficients α_s, β_s in (6.2). The method (6.2) is called *explicit* if $\beta_k = 0$ and *implicit* otherwise.

Implicit methods require the solution of a system of nonlinear equations,

$$\boldsymbol{u}_{n+k} = h\beta_k \boldsymbol{f}(x_{n+k}, \boldsymbol{u}_{n+k}) + \boldsymbol{g}_n, \tag{6.5}$$

where

$$\boldsymbol{g}_n = h\sum_{s=0}^{k-1}\beta_s \boldsymbol{f}_{n+s} - \sum_{s=0}^{k-1}\alpha_s \boldsymbol{u}_{n+s} \tag{6.6}$$

is a known vector. Fortunately, the nonlinearity in (6.5) is rather weak and in fact disappears in the limit as $h \to 0$. This suggests the use of successive iteration on (6.5),

$$\boldsymbol{u}_{n+k}^{[\nu]} = h\beta_k \boldsymbol{f}\left(x_{n+k}, \boldsymbol{u}_{n+k}^{[\nu-1]}\right) + \boldsymbol{g}_n, \quad \nu = 1, 2, \ldots, \tag{6.7}$$

where $\boldsymbol{u}_{n+k}^{[0]}$ is a suitable initial approximation for $\boldsymbol{u}_{n+k}$. By a simple application of the contraction mapping principle (cf. Chap. 4, Sect. 4.9.1), one shows that (6.7) indeed converges as $\nu \to \infty$, for arbitrary initial approximation, provided h is small enough.

Theorem 6.1.1. *Suppose $\boldsymbol{f}$ satisfies a uniform Lipschitz condition on $[a, b] \times \mathbb{R}^d$* (cf. Chap. 5, Sect. 5.3),

$$\|\boldsymbol{f}(x, \boldsymbol{y}) - \boldsymbol{f}(x, \boldsymbol{y}^*)\| \le L\|\boldsymbol{y} - \boldsymbol{y}^*\|,\ \ x \in [a, b],\ \ \boldsymbol{y}, \boldsymbol{y}^* \in \mathbb{R}^d, \tag{6.8}$$

and assume that

$$\lambda := h|\beta_k|L < 1. \tag{6.9}$$

Then (6.5) has a unique solution $\boldsymbol{u}_{n+k}$. Moreover, for arbitrary $\boldsymbol{u}_{n+k}^{[0]}$,

$$\boldsymbol{u}_{n+k} = \lim_{\nu\to\infty} \boldsymbol{u}_{n+k}^{[\nu]}, \tag{6.10}$$

and

$$\left\| \boldsymbol{u}_{n+k}^{[\nu]} - \boldsymbol{u}_{n+k} \right\| \le \frac{\lambda^\nu}{1-\lambda} \left\| \boldsymbol{u}_{n+k}^{(1)} - \boldsymbol{u}_{n+k}^{(0)} \right\|, \quad \nu = 1, 2, 3, \dots \tag{6.11}$$

Proof. We define the map $\boldsymbol{\varphi}\colon \mathbb{R}^d \to \mathbb{R}^d$ by

$$\boldsymbol{\varphi}(\boldsymbol{y}) := h\beta_k \boldsymbol{f}(x_{n+k}, \boldsymbol{y}) + \boldsymbol{g}_n. \tag{6.12}$$

Then, for any $\boldsymbol{y}, \boldsymbol{y}^* \in \mathbb{R}^d$, we have

$$\begin{aligned}\|\boldsymbol{\varphi}(\boldsymbol{y}) - \boldsymbol{\varphi}(\boldsymbol{y}^*)\| &= h|\beta_k| \|\boldsymbol{f}(x_{n+k}, \boldsymbol{y}) - \boldsymbol{f}(x_{n+k}, \boldsymbol{y}^*)\| \\ &\le h|\beta_k| L \|\boldsymbol{y} - \boldsymbol{y}^*\|,\end{aligned}$$

showing, in view of (6.9), that $\boldsymbol{\varphi}$ is a contraction operator on $\mathbb{R}^d$. By the contraction mapping principle, there is a unique fixed point of $\boldsymbol{\varphi}$, that is, a vector $\boldsymbol{y} = \boldsymbol{u}_{n+k}$ satisfying $\boldsymbol{\varphi}(\boldsymbol{y}) = \boldsymbol{y}$. This proves the first part of the theorem. The second part is also a consequence of the contraction mapping principle if one notes that (6.7) is just the fixed point iteration $\boldsymbol{u}_{n+k}^{[\nu]} = \boldsymbol{\varphi}(\boldsymbol{u}_{n+k}^{[\nu-1]})$. □

Strictly speaking, an implicit multistep method requires "iteration to convergence" in (6.7), that is, iteration until the required fixed point is obtained to machine accuracy. This may well entail too many iterations to make the method competitive, since each iteration step costs one evaluation of $\boldsymbol{f}$. In practice, one often terminates the iteration after the first or second step, having selected the starting value $\boldsymbol{u}_{n+k}^{[0]}$ judiciously (cf. Sect. 6.2.3). It should also be noted that stiff systems, for which L may be quite large, would require unrealistically small steps h to satisfy (6.9). In such cases, Newton's method (see Ex. 1), rather than fixed point iteration, would be preferable.

6.1.2 Local Accuracy

In analogy with one-step methods (cf. Chap. 5, (5.77) and (5.78)), we define *residual operators* by

$$(R\boldsymbol{v})(x) := \boldsymbol{v}'(x) - \boldsymbol{f}(x, \boldsymbol{v}(x)), \quad \boldsymbol{v} \in C^1[a, b], \tag{6.13}$$

$$(R_h\boldsymbol{v})_n := \frac{1}{h}\sum_{s=0}^{k} \alpha_s \boldsymbol{v}_{n+s} - \sum_{s=0}^{k} \beta_s \boldsymbol{f}(x_{n+s}, \boldsymbol{v}_{n+s}), \quad \boldsymbol{v} \in \Gamma_h[a, b],$$

$$n = 0, 1, \dots, N-k. \tag{6.14}$$

(We may arbitrarily define $(R_h\boldsymbol{v})_N = \cdots = (R_h\boldsymbol{v})_{N-k+1} = (R_h\boldsymbol{v})_{N-k}$ to obtain a grid function $R_h\boldsymbol{v}$ defined on the entire grid.) In (6.14) and throughout this chapter, we adopt the convention $\alpha_k = 1$. Since there is no longer a natural "generic" point $(x, \boldsymbol{y})$ in which to define our method, we take the analogue of Chap. 5, (5.81) (except for sign) as our definition of truncation error:

$$(\boldsymbol{T}_h)_n = (R_h\boldsymbol{y})_n, \quad n = 0, 1, \ldots, N, \tag{6.15}$$

where $\boldsymbol{y}(x)$ is the exact solution of (6.1). This defines a grid function $\boldsymbol{T}_h$ on a uniform grid on $[a, b]$. We define consistency, order, and principal error function as before in Chap. 5, that is, the method (6.2) is *consistent* if

$$\|\boldsymbol{T}_h\|_\infty \to \boldsymbol{0} \text{ as } h \to 0, \tag{6.16}$$

has *order* p if

$$\|\boldsymbol{T}_h\|_\infty = O(h^p) \text{ as } h \to 0, \tag{6.17}$$

and admits a *principal error function* $\boldsymbol{\tau} \in C[a, b]$ if

$$\boldsymbol{\tau}(x) \not\equiv \boldsymbol{0} \text{ and } (\boldsymbol{T}_h)_n = \boldsymbol{\tau}(x_n)h^p + O(h^{p+1}) \text{ as } h \to 0 \tag{6.18}$$

in the usual sense that $\|\boldsymbol{T}_h - h^p\boldsymbol{\tau}\|_\infty = O(h^{p+1})$. The infinity norm of grid functions in (6.16)–(6.18) is as defined in Chap. 5, (5.75).

Note that (6.15) can be written in the simple form

$$\sum_{s=0}^{k} \alpha_s \boldsymbol{y}(x_{n+s}) - h\sum_{s=0}^{k} \beta_s \boldsymbol{y}'(x_{n+s}) = h(\boldsymbol{T}_h)_n, \tag{6.19}$$

since $\boldsymbol{f}(x_{n+s}, \boldsymbol{y}(x_{n+s})) = \boldsymbol{y}'(x_{n+s})$ by virtue of the differential equation (6.1). Although (6.19) is a relation for vector-valued functions, the relationship is exactly the same for each component. This suggests defining a linear operator $L_h\colon C^1[\mathbb{R}] \to C[\mathbb{R}]$ on scalar functions by letting

$$(L_h z)(x) := \sum_{s=0}^{k} [\alpha_s z(x + sh) - h\beta_s z'(x + sh)], \ \ z \in C^1[\mathbb{R}]. \tag{6.20}$$

If $L_h z$ were identically 0 for all $z \in C^1[\mathbb{R}]$, then so would be the truncation error, and our method would produce exact answers if started with exact initial values. This is unrealistic, however; nevertheless, we would like L_h to annihiliate as many functions as possible. This motivates the following concept of degree. Given a set of linearly independent "gauge functions" $\{\omega_r(x)\}_{r=0}^{\infty}$ (usually complete on compact intervals), we say that the method (6.2) has Ω-*degree* p if its associated linear operator L_h satisfies

$$L_h\omega \equiv 0 \text{ for all } \omega \in \Omega_p, \text{ all } h > 0. \tag{6.21}$$

Here, Ω_p is the set of functions spanned by the first $p+1$ gauge functions $\omega_0, \omega_1, \ldots, \omega_p$, hence

$$\Omega_0 \subset \Omega_1 \subset \Omega_2 \subset \cdots, \quad \dim \Omega_m = m+1. \tag{6.22}$$

We say that Ω_m is *closed under translation* if $\omega(x) \in \Omega_m$ implies $\omega(x+c) \in \Omega_m$ for arbitrary real c. Similarly, Ω_m is said to be *closed under scaling* if $\omega(x) \in \Omega_m$ implies $\omega(cx) \in \Omega_m$ for arbitrary real c. For example, algebraic polynomials $\Omega_m = \mathbb{P}_m$ are closed under translation as well as scaling, trigonometric polynomials $\Omega_m = \mathbb{T}_m[0, 2\pi]$ and exponential sums $\Omega_m = \mathbb{E}_m$ (cf. Chap. 2, Examples to (2.2)) only under translation, and spline functions $\mathbb{S}_m^k(\Delta)$ (for fixed partition) neither under scaling nor under translation.

The following theorem is no more than a simple observation.

Theorem 6.1.2. (a) *If Ω_p is closed under translation, then the method (6.2) has Ω-degree p if and only if*

$$(L_h\omega)(0) = 0 \textit{ for all } \omega \in \Omega_p, \textit{ all } h > 0. \tag{6.23}$$

(b) *If Ω_p is closed under translation and scaling, then the method (6.2) has Ω-degree p if and only if*

$$(L_1\omega)(0) = 0 \textit{ for all } \omega \in \Omega_p, \tag{6.24}$$

where $L_1 = L_h$ for $h = 1$.

Proof. (a) The necessity of (6.23) is trivial. To prove its sufficiency, it is enough to show that for any $x_0 \in \mathbb{R}$,

$$(L_h\omega)(x_0) = 0, \text{ all } \omega \in \Omega_p, \text{ all } h > 0.$$

Take any $\omega \in \Omega_p$ and define $\omega_0(x) = \omega(x + x_0)$. Then, by assumption, $\omega_0 \in \Omega_p$; hence, for all $h > 0$,

$$0 = (L_h\omega_0)(0) = \sum_{s=0}^{k} [\alpha_s\omega(x_0 + sh) - h\beta_s\omega'(x_0 + sh)] = (L_h\omega)(x_0).$$

(b) The necessity of (6.24) is trivial. For the sufficiency, let $\omega_0(x) = \omega(x_0 + xh)$ for any given $\omega \in \Omega_p$. Then, by assumption, $\omega_0 \in \Omega_p$ and

$$0 = (L_1\omega_0)(0) = \sum_{s=0}^{k} [\alpha_s\omega_0(s) - \beta_s\omega_0'(s)]$$

$$= \sum_{s=0}^{k} [\alpha_s\omega(x_0 + sh) - h\beta_s\omega'(x_0 + sh)] = (L_h\omega)(x_0).$$

Since $x_0 \in \mathbb{R}$ and $h > 0$ are arbitrary, the assertion follows. □

Case (b) of Theorem 6.1.2 suggests the introduction of the *linear functional* $L\colon C^1[\mathbb{R}] \to \mathbb{R}$ associated with the method (6.2),

$$Lu := \sum_{s=0}^{k} [\alpha_s u(s) - \beta_s u'(s)], \quad u \in C^1[\mathbb{R}]. \tag{6.25}$$

For $\Omega_m = \mathbb{P}_m$ we refer to Ω-degree as the *algebraic* (or *polynomial*) *degree*. Thus, (6.2) has algebraic degree p if $Lu = 0$ for all $u \in \mathbb{P}_p$. By linearity, this is equivalent to

$$Lt^r = 0, \quad r = 0, 1, \dots, p. \tag{6.26}$$

Example. Determine all explicit two-step methods

$$\boldsymbol{u}_{n+2} + \alpha_1 \boldsymbol{u}_{n+1} + \alpha_0 \boldsymbol{u}_n = h(\beta_1 \boldsymbol{f}_{n+1} + \beta_0 \boldsymbol{f}_n) \tag{6.27}$$

having polynomial degree $p = 0, 1, 2, 3$.

Here

$$Lu = u(2) + \alpha_1 u(1) + \alpha_0 u(0) - \beta_1 u'(1) - \beta_0 u'(0).$$

The first four equations in (6.26) are

$$\begin{aligned} 1 + \alpha_1 + \alpha_0 &= 0, \\ 2 + \alpha_1 - \beta_1 - \beta_0 &= 0, \\ 4 + \alpha_1 - 2\beta_1 &= 0, \\ 8 + \alpha_1 - 3\beta_1 &= 0. \end{aligned} \tag{6.28}$$

We have algebraic degree 0, 1, 2, 3 if, respectively, the first, the first two, the first three, and all four equations in (6.28) are satisfied. Thus,

$$\begin{aligned} \alpha_1 &= -\alpha_0 - 1, \quad \beta_0, \beta_1 \ \text{arbitrary} \quad (p = 0), \\ \alpha_1 &= -\alpha_0 - 1,\ \beta_1 = -\alpha_0 - \beta_0 + 1 \quad (p = 1), \\ \alpha_1 &= -\alpha_0 - 1,\ \beta_1 = -\frac{1}{2}\alpha_0 + \frac{3}{2},\ \beta_0 = -\frac{1}{2}\alpha_0 - \frac{1}{2} \quad (p = 2), \\ \alpha_1 &= 4,\ \alpha_0 = -5,\ \beta_1 = 4,\ \beta_0 = 2 \quad (p = 3) \end{aligned} \tag{6.29}$$

yield $(3 - p)$-parameter families of methods of degree p. Since $Lt^4 = 16 + \alpha_1 - 4\beta_1 = 4 \neq 0$, degree $p = 4$ is impossible. This means that the last method in (6.29),

$$\boldsymbol{u}_{n+2} + 4\boldsymbol{u}_{n+1} - 5\boldsymbol{u}_n = 2h(2\boldsymbol{f}_{n+1} + \boldsymbol{f}_n), \tag{6.30}$$

is optimal as far as algebraic degree is concerned. Other special cases include the *midpoint rule*

$$\boldsymbol{u}_{n+2} = \boldsymbol{u}_n + 2h\boldsymbol{f}_{n+1} \quad (\alpha_0 = -1;\ p = 2)$$

and the *Adams–Bashforth* second-order *method* (cf. Sect. 6.2.1),

$$\boldsymbol{u}_{n+2} = \boldsymbol{u}_{n+1} + \frac{1}{2}h(3\boldsymbol{f}_{n+1} - \boldsymbol{f}_n) \quad (\alpha_0 = 0;\ p = 2).$$

The "optimal" method (6.30) is nameless – and for good reason. Suppose, indeed, that we apply it to the trivial (scalar) initial value problem

$$y' = 0, \ \ y(0) = 0 \ \text{ on } \ 0 \le x \le 1,$$

which has the exact solution $y(x) \equiv 0$. Assume $u_0 = 0$ but $u_1 = \varepsilon$ (to account for a small rounding error in the second starting value). Even if (6.30) is applied with infinite precision, we then get

$$u_n = \frac{1}{6}\,\varepsilon[1 + (-1)^{n+1}5^n], \ \ n = 0, 1, \ldots, N.$$

Assuming further that $\varepsilon = h^{p+1}$ (pth-order one-step method), we will have, at the end of the interval [0,1],

$$u_N = \frac{1}{6}\,h^{p+1}[1 + (-1)^{N+1}5^N] \sim \frac{1}{6}\,(-1)^{N+1}\,N^{-p-1}5^N \ \text{ as } \ N \to \infty.$$

Thus, $|u_N| \to \infty$ exponentially fast and highly oscillating on top of it. We have here an example of "strong instability"; this is analyzed later in more detail (cf. Sect. 6.3).

6.1.3 Polynomial Degree vs. Order

We recall from (6.19) and (6.20) that for the truncation error $\boldsymbol{T}_h$ of (6.2), we have

$$h(\boldsymbol{T}_h)_n = \sum_{s=0}^{k}[\alpha_s \boldsymbol{y}(x_n + sh) - h\beta_s \boldsymbol{y}'(x_n + sh)] = (L_h \boldsymbol{y})(x_n),$$
$$n = 0, 1, 2, \ldots, N - k. \quad (6.31)$$

Let

$$\boldsymbol{u}(t) := \boldsymbol{y}(x_n + th), \ \ 0 \le t \le k. \quad (6.32)$$

(More precisely, we should write $\boldsymbol{u}_{n,h}(t)$.) Then

$$h(\boldsymbol{T}_h)_n = \sum_{s=0}^{k}[\alpha_s \boldsymbol{u}(s) - \beta_s \boldsymbol{u}'(s)] = L\boldsymbol{u}, \quad (6.33)$$

where L on the right is to be applied componentwise to each component of the vector $\boldsymbol{u}$. If $\boldsymbol{T}_h$ is the truncation error of a method of algebraic degree p, then the linear functional L in (6.33) annihilates all polynomials of degree p and thus, if $\boldsymbol{u} \in C^{p+1}[0,k]$, can be represented in terms of the Peano kernel

$$\lambda_p(\sigma) = L_{(t)}(t-\sigma)_+^p \tag{6.34}$$

by

$$L\boldsymbol{u} = \frac{1}{p!}\int_0^k \lambda_p(\sigma)\boldsymbol{u}^{(p+1)}(\sigma)\mathrm{d}\sigma \tag{6.35}$$

(cf. Chap. 3, Sect. 3.2.6). From the explicit formula

$$\lambda_p(\sigma) = \sum_{s=0}^{k} [\alpha_s(s-\sigma)_+^p - \beta_s p(s-\sigma)_+^{p-1}], \quad p \geq 1, \tag{6.36}$$

it is easily seen that $\lambda_p \in \mathbb{S}_p^{p-2}(\Delta)$, where Δ is the subdivision of $[0,k]$ into subintervals of length 1. Outside the interval $[0,k]$ we have $\lambda_p \equiv 0$. Moreover, if L is definite of order p, then, and only then (cf. Chap. 3, (3.80) and Ex. 51),

$$L\boldsymbol{u} = \ell_{p+1}\boldsymbol{u}^{(p+1)}(\overline{\sigma}), \;\; 0 < \overline{\sigma} < k; \;\; \ell_{p+1} = L\,\frac{t^{p+1}}{(p+1)!}. \tag{6.37}$$

The Peano representation (6.35) of L in combination with (6.33) allows us to identify the polynomial degree of a multistep method with its order as defined in Sect. 6.1.2.

Theorem 6.1.3. *A multistep method (6.2) of polynomial degree p has order p whenever the exact solution $\boldsymbol{y}(x)$ of (6.1) is in the smoothness class $\boldsymbol{C}^{p+1}[a,b]$. If the associated functional L is definite, then*

$$(\boldsymbol{T}_h)_n = \ell_{p+1}\boldsymbol{y}^{(p+1)}(\overline{x}_n)h^p, \quad x_n < \overline{x}_n < x_{n+k}, \tag{6.38}$$

where ℓ_{p+1} is as given in (6.37). Moreover, for the principal error function $\boldsymbol{\tau}$ of the method, whether definite or not, we have, if $\boldsymbol{y} \in C^{p+2}[a,b]$,

$$\boldsymbol{\tau}(x) = \ell_{p+1}\boldsymbol{y}^{(p+1)}(x). \tag{6.39}$$

Proof. By (6.32) and (6.33), we have

$$h(\boldsymbol{T}_h)_n = L\boldsymbol{u}, \;\; \boldsymbol{u}(t) = \boldsymbol{y}(x_n + th), \;\; n = 0,1,2,\ldots,N-k;$$

hence, by (6.35),

$$h(\boldsymbol{T}_h)_n = \frac{1}{p!}\int_0^k \lambda_p(\sigma)\boldsymbol{u}^{(p+1)}(\sigma)\mathrm{d}\sigma = \frac{h^{p+1}}{p!}\int_0^k \lambda_p(\sigma)\boldsymbol{y}^{(p+1)}(x_n+\sigma h)\mathrm{d}\sigma. \tag{6.40}$$

Therefore,

$$\begin{aligned}\|(\boldsymbol{T}_h)_n\| &= \frac{h^p}{p!}\left\|\int_0^k \lambda_p(\sigma)\boldsymbol{y}^{(p+1)}(x_n+\sigma h)\mathrm{d}\sigma\right\| \\ &\le \frac{h^p}{p!}\int_0^k |\lambda_p(\sigma)|\|\boldsymbol{y}^{(p+1)}(x_n+\sigma h)\|\mathrm{d}\sigma \\ &\le \frac{h^p}{p!}\|\boldsymbol{y}^{(p+1)}\|_\infty \int_0^k |\lambda_p(\sigma)|\mathrm{d}\sigma,\end{aligned}$$

and we see that

$$\|\boldsymbol{T}_h\|_\infty \le Ch^p, \quad C = \frac{1}{p!}\|\boldsymbol{y}^{(p+1)}\|_\infty \int_0^k |\lambda_p(\sigma)|\mathrm{d}\sigma. \tag{6.41}$$

This proves the first part of the theorem.

If L is definite, then (6.38) follows directly from (6.33) and from (6.37) applied to the vector-valued function $\boldsymbol{u}$ in (6.32). Finally, (6.39) follows from (6.40) by noting that $\boldsymbol{y}^{(p+1)}(x_n+\sigma h) = \boldsymbol{y}^{(p+1)}(x_n)+O(h)$ and the fact that $\frac{1}{p!}\int_0^k \lambda_p(\sigma)\mathrm{d}\sigma = \ell_{p+1}$ (cf. Chap. 3, (3.80)). □

The proof of Theorem 6.1.3 also exhibits, in (6.41), an explicit bound on the local truncation error. For methods with definite functional L, there is the even simpler bound (6.41) with

$$C = |\ell_{p+1}|\|\boldsymbol{y}^{(p+1)}\|_\infty \quad (L \text{ definite}), \tag{6.42}$$

which follows from (6.38). It is seen later in Sect. 6.3.4 that for the global discretization error it is not ℓ_{p+1} which is relevant, but rather

$$C_{k,p} = \frac{\ell_{p+1}}{\sum_{s=0}^k \beta_s}, \tag{6.43}$$

which is called the *error constant* of the k-step method (6.2) (of order p). The denominator in (6.43) is positive if the method is stable and consistent (cf. Sect. 6.4.3).

As a simple example, consider the midpoint rule

$$\boldsymbol{u}_{n+2} = \boldsymbol{u}_n + 2h\boldsymbol{f}_n$$

for which

$$L\boldsymbol{u} = \boldsymbol{u}(2) - \boldsymbol{u}(0) - 2\boldsymbol{u}'(1).$$

We already know that it has order $p = 2$. The Peano kernel is

$$\lambda_2(\sigma) = (2-\sigma)_+^2 - 4(1-\sigma)_+,$$

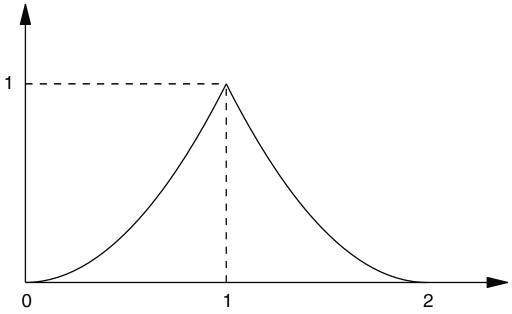


Fig. 6.1 Peano kernel of the midpoint rule

so that

$$\lambda_2(\sigma) = \begin{cases} (2-\sigma)^2 - 4(1-\sigma) = \sigma^2 & \text{if } 0 < \sigma < 1, \\ (2-\sigma)^2 & \text{if } 1 < \sigma < 2. \end{cases}$$

It consists of two parabolic arcs as shown in Fig. 6.1. Evidently, L is positive definite; hence by (6.38) and (6.37),

$$(\boldsymbol{T}_h)_n = \ell_3 \boldsymbol{y}^{(3)}(\overline{x}_n)h^2, \quad \ell_3 = L\,\frac{t^3}{6} = \frac{1}{3}.$$

6.2 Examples of Multistep Methods

An alternative way of deriving multistep formulae, which does not require the solution of linear algebraic systems (as does the method of Sect. 6.1.2), starts from the fundamental theorem of calculus,

$$\boldsymbol{y}(x_{n+k}) = \boldsymbol{y}(x_{n+k-1}) + \int_{x_{n+k-1}}^{x_{n+k}} \boldsymbol{y}'(x)\mathrm{d}x. \tag{6.44}$$

(Instead of x_{n+k-1} on the right, we could take any other grid point $x_{n+\kappa}$, $0 \le \kappa < k-1$, but we limit ourselves to the case shown in (6.44).) A multistep formula results from (6.44) if the integral is expressed (approximately) by a linear combination of derivative values at some grid points selected from the set $\{x_{n+\kappa}\colon \kappa = 0, 1, \ldots, k\}$. Those selected may be called the "active" grid points. A simple way to do this is to approximate $\boldsymbol{y}'$ by the unique polynomial interpolating $\boldsymbol{y}'$ at the active grid points. If we carry along the remainder term, we will also get an expression for the truncation error.

We implement this in the two most important cases where the active grid points are $x_n, x_{n+1}, \ldots, x_{n+k-1}$ and $x_{n+1}, x_{n+2}, \ldots, x_{n+k}$, respectively, giving rise to the family of Adams-type multistep methods.

6.2.1 Adams[1]–Bashforth Method

Replace $\boldsymbol{y}'$ in (6.44) by the interpolation polynomial $\boldsymbol{p}_{k-1}(\boldsymbol{y}'; x_n, x_{n+1}, \dots, x_{n+k-1}; x)$ of degree $\leq k-1$ interpolating $\boldsymbol{y}'$ at the grid points $x_n, x_{n+1}, \dots, x_{n+k-1}$ (cf. Chap. 2, Sect. 2.2.1). If we include the remainder term (cf. Chap. 2, Sect. 2.2.2), assuming that $\boldsymbol{y} \in C^{k+1}$, and make the change of variable $x = x_{n+k-1} + th$ in the integral of (6.44), we obtain

$$\boldsymbol{y}(x_{n+k}) = \boldsymbol{y}(x_{n+k-1}) + h \sum_{s=0}^{k-1} \beta_{k,s} \boldsymbol{y}'(x_{n+s}) + h\boldsymbol{r}_n, \tag{6.45}$$

where

$$\beta_{k,s} = \int_0^1 \prod_{\substack{r=0 \\ r \neq s}}^{k-1} \left(\frac{t+k-1-r}{s-r} \right) \mathrm{d}t, \quad s = 0, 1, \dots, k-1, \tag{6.46}$$

and

$$\boldsymbol{r}_n = \gamma_k h^k \boldsymbol{y}^{(k+1)}(\overline{x}_n), \;\; x_n < \overline{x}_n < x_{n+k-1}; \;\; \gamma_k = \int_0^1 \binom{t+k-1}{k} \mathrm{d}t. \tag{6.47}$$

The formula (6.45) suggests the multistep method

$$\boldsymbol{u}_{n+k} = \boldsymbol{u}_{n+k-1} + h \sum_{s=0}^{k-1} \beta_{k,s} \boldsymbol{f}(x_{n+s}, \boldsymbol{u}_{n+s}), \tag{6.48}$$

which is called the *kth-order Adams–Bashforth method*. It is called kth-order, since comparison of (6.45) with (6.19) shows that in fact $\boldsymbol{r}_n$ is the truncation error of (6.48),

$$\boldsymbol{r}_n = (\boldsymbol{T}_h)_n, \tag{6.49}$$

and (6.47) shows that $\boldsymbol{T}_h = O(h^k)$. In view of the form (6.47) of the truncation error, we can infer, as mentioned in Sect. 6.1.3, that the linear functional L associated with

[1] John Couch Adams (1819–1892), son of a tenant farmer, studied at Cambridge University where, in 1859, he became professor of astronomy and geometry and, from 1861 on, director of the observatory. His calculations, while still a student at Cambridge, predicted the existence of the then unknown planet Neptune, based on irregularities in the orbit of the next inner planet Uranus. Unfortunately, publication of his findings was delayed and it was Le Verrier, who did similar calculations (and managed to publish them), and young astronomers in Berlin who succeeded in locating the planet at the position predicted, who received credit for this historical discovery. Understandably, this led to a prolonged dispute of priority. Adams is also known for his work on lunar theory and magnetism. Later in his life, he turned to computational problems in number theory. An ardent admirer of Newton, he took it upon himself to catalogue a large body of scientific papers left behind by Newton after his death.

(6.48) is definite. We prove later in Sect. 6.4 (cf. Ex. 11(b)) that there is no stable explicit k-step method that has order $p > k$. In this sense, (6.48) with $\beta_{k,s}$ given by (6.46) is optimal.

If $\boldsymbol{y} \in C^{k+2}$, it also follows from (6.47) and (6.49) that

$$(\boldsymbol{T}_h)_n = \gamma_k h^k \boldsymbol{y}^{(k+1)}(x_n) + O(h^{k+1}), \quad h \to 0,$$

that is, the Adams–Bashforth method (6.48) has the principal error function

$$\boldsymbol{\tau}(x) = \gamma_k \boldsymbol{y}^{(k+1)}(x). \tag{6.50}$$

The constant γ_k (defined in (6.47) is therefore the same as the constant ℓ_{k+1} defined earlier in (6.37).

Had we used Newton's form of the interpolation polynomial (cf. Chap. 2, Sect. 2.2.6) and observed that the divided differences required for equally spaced points are (cf. Chap. 2, Ex. 54)

$$[x_{n+k-1}, x_{n+k-2}, \ldots, x_{n+k-s-1}]\boldsymbol{f} = \frac{1}{s!h^s} \nabla^s \boldsymbol{f}_{n+k-1}, \tag{6.51}$$

where $\nabla \boldsymbol{f}_{n+k-1} = \boldsymbol{f}_{n+k-1} - \boldsymbol{f}_{n+k-2}$, $\nabla^2 \boldsymbol{f}_{n+k-1} = \nabla(\nabla \boldsymbol{f}_{n+k-1}), \ldots$ are ordinary (backward) differences, we would have obtained (6.48) in the form

$$\boldsymbol{u}_{n+k} = \boldsymbol{u}_{n+k-1} + h \sum_{s=0}^{k-1} \gamma_s \nabla^s \boldsymbol{f}_{n+k-1}, \tag{6.52}$$

where

$$\gamma_s = \int_0^1 \binom{t+s-1}{s} \mathrm{d}t, \quad s = 0, 1, 2, \ldots. \tag{6.53}$$

The difference form (6.52) of the Adams–Bashforth method has important practical advantages over the Lagrange form (6.48). For one thing, the coefficients in (6.52) do not depend on the step number k. Adding more terms in the sum of (6.52) thus increases the order (and step number) of the method. Related to this is the fact that the first omitted term in the summation of (6.52) is a good approximation of the truncation error. Indeed, by Chap. 2, (2.117), we know that

$$\boldsymbol{y}^{(k+1)}(\overline{x}_n) \approx k![x_{n+k-1}, x_{n+k-2}, \ldots, x_n, x_{n-1}]\boldsymbol{y}';$$

hence, by (6.47), (6.49), and (6.51),

$$(\boldsymbol{T}_h)_n = \gamma_k h^k \boldsymbol{y}^{(k+1)}(\overline{x}_n) \approx \gamma_k h^k k! \frac{\nabla^k \boldsymbol{f}_{n+k-1}}{k!h^k} = \gamma_k \nabla^k \boldsymbol{f}_{n+k-1}.$$

When implementing the method (6.52), one needs to set up a table of (backward) differences for each component of $\boldsymbol{f}$. By adding an extra column of differences at the end of the table (the kth differences), we are thus able to monitor the local truncation errors by simply multiplying these differences by γ_k. No such easy procedure is available for the Lagrange form of the method.

It is, therefore, of some importance to have an effective method for calculating the coefficients γ_s, $s = 0, 1, 2, \ldots$. Such a method can be derived from the generating function $\gamma(z) = \sum_{s=0}^{\infty} \gamma_s z^s$ of the coefficients. We have

$$\begin{aligned}\gamma(z) &= \sum_{s=0}^{\infty} z^s \int_0^1 \binom{t+s-1}{s} \mathrm{d}t = \sum_{s=0}^{\infty} z^s (-1)^s \int_0^1 \binom{-t}{s} \mathrm{d}t \\ &= \int_0^1 \sum_{s=0}^{\infty} \binom{-t}{s} (-z)^s \mathrm{d}t = \int_0^1 (1-z)^{-t}\,\mathrm{d}t \\ &= \int_0^1 \mathrm{e}^{-t\ln(1-z)}\mathrm{d}t = -\left.\frac{\mathrm{e}^{-t\ln(1-z)}}{\ln(1-z)}\right|_{t=0}^{t=1} \\ &= -\frac{z}{(1-z)\ln(1-z)}.\end{aligned}$$

Thus, the γ_s are the coefficients in the Maclaurin expansion of

$$\gamma(z) = -\frac{z}{(1-z)\ln(1-z)}. \tag{6.54}$$

In particular,

$$\frac{z}{1-z} = -\ln(1-z) \sum_{s=0}^{\infty} \gamma_s z^s;$$

that is,

$$z + z^2 + z^3 + \cdots \; = \left(z + \frac{1}{2}z^2 + \frac{1}{3}z^3 + \cdots\right)(\gamma_0 + \gamma_1 z + \gamma_2 z^2 + \cdots),$$

which, on comparing coefficients of like powers on the left and right, yields

$$\begin{aligned}\gamma_0 &= 1, \\ \gamma_s &= 1 - \frac{1}{2}\gamma_{s-1} - \frac{1}{3}\gamma_{s-2} - \cdots - \frac{1}{s+1}\gamma_0, \\ &\qquad\qquad s = 1, 2, 3, \ldots.\end{aligned} \tag{6.55}$$

It is, therefore, easy to compute as many of the coefficients γ_s as desired, for example,

$$\gamma_0 = 1,\ \ \gamma_1 = \frac{1}{2},\ \ \gamma_2 = \frac{5}{12},\ \ \gamma_3 = \frac{3}{8}, \ldots$$

Note that they are all positive, as they must be in view of (6.53).

6.2.2 *Adams–Moulton*[2] *Method*

This is the implicit analogue of the Adams–Bashforth method, that is, the point x_{n+k} is included among the active grid points. To obtain again a method of order k, we need, as before, k active points and therefore select them to be $x_{n+k}, x_{n+k-1}, \dots, x_{n+1}$. The derivation of the method is then entirely analogous to the one in Sect. 6.2.1, and we limit ourselves to simply stating the results.

The Lagrange form of the method is now

$$\boldsymbol{u}_{n+k} = \boldsymbol{u}_{n+k-1} + h \sum_{s=1}^{k} \beta_{k,s}^{*} \boldsymbol{f}\,(x_{n+s}, \boldsymbol{u}_{n+s}) \tag{6.56}$$

with

$$\beta_{k,s}^{*} = \int_0^1 \prod_{\substack{r=1 \\ r\neq s}}^{k} \left(\frac{t+k-1-r}{s-r} \right) \mathrm{d}t, \quad s = 1, 2, \dots, k, \tag{6.57}$$

whereas Newton's form becomes

$$\boldsymbol{u}_{n+k} = \boldsymbol{u}_{n+k-1} + h \sum_{s=0}^{k-1} \gamma_s^{*}\, \nabla^s \boldsymbol{f}_{n+k} \tag{6.58}$$

with

$$\gamma_s^{*} = \int_{-1}^{0} \binom{t+s-1}{s} \mathrm{d}t, \quad s = 0, 1, 2, \dots . \tag{6.59}$$

The truncation error and principal error function are

$$(\boldsymbol{T}_h^{*})_n = \gamma_k^{*} h^k \boldsymbol{y}^{(k+1)}(\overline{x}_n^{*}), \;\; x_{n+1} < \overline{x}_n^{*} < x_{n+k}; \;\; \boldsymbol{\tau}^{*}(x) = \gamma_k^{*} \boldsymbol{y}^{(k+1)}(x), \tag{6.60}$$

and the generating function for the γ_s^{*} is

$$\gamma^{*}(z) = \sum_{s=0}^{\infty} \gamma_s^{*} z^s = -\frac{z}{\ln(1-z)}. \tag{6.61}$$

From this, one finds as before,

$$\begin{aligned} \gamma_0^{*} &= 1, \\ \gamma_s^{*} &= -\frac{1}{2}\,\gamma_{s-1}^{*} - \frac{1}{3}\,\gamma_{s-2}^{*} - \cdots - \frac{1}{s+1}\,\gamma_0^{*}, \\ & \qquad\qquad\qquad\qquad s = 1, 2, 3, \dots . \end{aligned} \tag{6.62}$$

[2] Forest Ray Moulton (1872–1952) was professor of Astronomy at the University of Chicago and, from 1927 to 1936, director of the Utilities Power and Light Corp. of Chicago. He used his method during World War I and thereafter to integrate the equations of exterior ballistics. He made also contributions to celestial mechanics.

So, for example,

$$\gamma_0^* = 1,\ \ \gamma_1^* = -\frac{1}{2},\ \ \gamma_2^* = -\frac{1}{12},\ \ \gamma_3^* = -\frac{1}{24}, \ldots .$$

It follows again from (6.60) that the truncation error is approximately the first omitted term in the sum of (6.58),

$$(\boldsymbol{T}_h^*)_n \approx \gamma_k^* \, \nabla^k \boldsymbol{f}_{n+k}. \tag{6.63}$$

Since the formula (6.58) is implicit, however, it has to be solved by iteration. A common procedure is to get a first approximation by assuming the last, $(k-1)$th, difference retained to be constant over the step from x_{n+k-1} to x_{n+k}, which allows one to generate the lower-order differences backward until one obtains (an approximation to) $\boldsymbol{f}_{n+k}$. We then compute $\boldsymbol{u}_{n+k}$ from (6.58) and a new $\boldsymbol{f}_{n+k}$ in terms of it. Then we revise all the differences required in (6.58) and reevaluate $\boldsymbol{u}_{n+k}$. The process can be repeated until it converges to the desired accuracy. In effect, this is the fixed point iteration of Sect. 6.1.1, with a special choice of the initial approximation.

6.2.3 Predictor–Corrector Methods

These are pairs of an explicit and an implicit multistep method, usually of the same order, where the explicit formula is used to predict the next approximation and the implicit formula to correct it. Suppose we use an explicit k-step method of order k, with coefficients α_s, β_s, for the predictor, and an implicit $(k-1)$-step method of order k with coefficients α_s^*, β_s^*, for the corrector. Assume further, for simplicity, that both methods are definite (in the sense of (6.37)). Then, in Lagrange form, if $\overset{\circ}{\boldsymbol{u}}_{n+k}$ is the predicted approximation, one proceeds as follows:

$$\begin{cases} \overset{\circ}{\boldsymbol{u}}_{n+k} = -\displaystyle\sum_{s=0}^{k-1} \alpha_s \boldsymbol{u}_{n+s} + h \sum_{s=0}^{k-1} \beta_s \boldsymbol{f}_{n+s}, \\[2ex] \boldsymbol{u}_{n+k} = -\displaystyle\sum_{s=1}^{k-1} \alpha_s^* \boldsymbol{u}_{n+s} + h \left\{ \beta_k^* \boldsymbol{f}(x_{n+k}, \overset{\circ}{\boldsymbol{u}}_{n+k}) + \sum_{s=1}^{k-1} \beta_s^* \boldsymbol{f}_{n+s} \right\}, \\[2ex] \boldsymbol{f}_{n+k} = \boldsymbol{f}(x_{n+k}, \boldsymbol{u}_{n+k}). \end{cases} \tag{6.64}$$

This requires exactly two evaluations of $\boldsymbol{f}$ per step and is often referred to as a PECE method, where "P" stands for "predict," "E" for "evaluate," and "C" for "correct." One could of course correct once more, and then either quit, or reevaluate $\boldsymbol{f}$, and so on. Thus there are methods of type P(EC)^2, P(EC)^{2}E, and the like. Each additional reevaluation costs another function evaluation and, therefore, the most economic methods are those of type PECE.

Let us analyze the truncation error of the PECE method (6.64). It is natural to define it by

$$(\boldsymbol{T}_h^{\mathrm{PECE}})_n = \frac{1}{h}\sum_{s=1}^{k}\alpha_s^*\boldsymbol{y}(x_{n+s}) - \left\{\beta_k^*\boldsymbol{f}(x_{n+k},\overset{\circ}{\boldsymbol{y}}_{n+k}) + \sum_{s=1}^{k-1}\beta_s^*\boldsymbol{y}'(x_{n+s})\right\},$$

where $\alpha_k^* = 1$ and

$$\overset{\circ}{\boldsymbol{y}}_{n+k} = -\sum_{s=0}^{k-1}\alpha_s\boldsymbol{y}(x_{n+s}) + h\sum_{s=0}^{k-1}\beta_s\boldsymbol{y}'(x_{n+s});$$

that is, we apply (6.64) on exact values $\boldsymbol{u}_{n+s} = \boldsymbol{y}(x_{n+s})$, $s = 0, 1, \ldots, k-1$. We can write

$$\begin{aligned}\left(\boldsymbol{T}_h^{\mathrm{PECE}}\right)_n &= \frac{1}{h}\sum_{s=1}^{k}\alpha_s^*\boldsymbol{y}(x_{n+s}) - \sum_{s=1}^{k}\beta_s^*\boldsymbol{y}'(x_{n+s})\\ &\quad + \beta_k^*[\boldsymbol{y}'(x_{n+k}) - \boldsymbol{f}(x_{n+k},\overset{\circ}{\boldsymbol{y}}_{n+k})]\\ &= \ell_{k+1}^*h^k\boldsymbol{y}^{(k+1)}(\overline{x}_n^*) + \beta_k^*[\boldsymbol{f}(x_{n+k},\boldsymbol{y}(x_{n+k})) - \boldsymbol{f}(x_{n+k},\overset{\circ}{\boldsymbol{y}}_{n+k})], \quad (6.65)\end{aligned}$$

having used the truncation error $(\boldsymbol{T}_h^*)_n$ (in (6.38)) of the corrector formula. Since

$$\boldsymbol{y}(x_{n+k}) - \overset{\circ}{\boldsymbol{y}}_{n+k} = \ell_{k+1}h^{k+1}\boldsymbol{y}^{(k+1)}(\overline{x}_n) \qquad (6.66)$$

is h times the truncation error of the predictor formula, the Lipschitz condition on $\boldsymbol{f}$ yields

$$\|\boldsymbol{f}(x_{n+k},\boldsymbol{y}(x_{n+k})) - \boldsymbol{f}(x_{n+k},\overset{\circ}{\boldsymbol{y}}_{n+k})\| \le L|\ell_{k+1}|h^{k+1}\|\boldsymbol{y}^{(k+1)}\|_\infty, \qquad (6.67)$$

and hence, from (6.65), we obtain

$$\|\boldsymbol{T}_h^{\mathrm{PECE}}\|_\infty \le (|\ell_{k+1}^*| + hL|\ell_{k+1}\beta_k^*|)\|\boldsymbol{y}^{(k+1)}\|_\infty h^k \le Ch^k,$$

where

$$C = (|\ell_{k+1}^*| + (b-a)L|\ell_{k+1}\beta_k^*|)\|\boldsymbol{y}^{(k+1)}\|_\infty.$$

Thus, the PECE method also has order k, and its principal error function is identical with that of the corrector formula, as follows immediately from (6.65) and (6.67).

The local truncation error $(\boldsymbol{T}_h^{\text{PECE}})_n$ can be estimated in terms of the difference between the (locally) predicted approximation $\overset{\circ}{\boldsymbol{y}}_{n+k}$ and the (locally) corrected approximation $\boldsymbol{u}_{n+k}$. One has, indeed, by definition of the truncation error,

$$\begin{aligned}\boldsymbol{u}_{n+k} &= -\sum_{s=1}^{k-1} \alpha_s^* \boldsymbol{y}(x_{n+s}) + h \left\{ \beta_k^* \boldsymbol{f}(x_{n+k}, \overset{\circ}{\boldsymbol{y}}_{n+k}) + \sum_{s=1}^{k-1} \beta_s^* \boldsymbol{y}'(x_{n+s}) \right\} \\ &= \boldsymbol{y}(x_{n+k}) - h \left(\boldsymbol{T}_h^{\text{PECE}}\right)_n ,\end{aligned}$$

that is,

$$\boldsymbol{u}_{n+k} - \boldsymbol{y}(x_{n+k}) = -\ell_{k+1}^* h^{k+1} \boldsymbol{y}^{(k+1)}(x_n) + O(h^{k+2}),$$

whereas, from (6.66),

$$\overset{\circ}{\boldsymbol{y}}_{n+k} - \boldsymbol{y}(x_{n+k}) = -\ell_{k+1} h^{k+1} \boldsymbol{y}^{(k+1)}(x_n) + O(h^{k+2}),$$

assuming that $\boldsymbol{y}(x) \in C^{k+2}[a,b]$. Upon subtraction, one gets

$$\boldsymbol{u}_{n+k} - \overset{\circ}{\boldsymbol{y}}_{n+k} = -(\ell_{k+1}^* - \ell_{k+1}) h^{k+1} \boldsymbol{y}^{(k+1)}(x_n) + O(h^{k+2}),$$

and thus,

$$\boldsymbol{y}^{(k+1)}(x_n) = -\frac{1}{\ell_{k+1}^* - \ell_{k+1}} \frac{1}{h^{k+1}} (\boldsymbol{u}_{n+k} - \overset{\circ}{\boldsymbol{y}}_{n+k}) + O(h).$$

Since, by (6.65) and (6.67),

$$\left(\boldsymbol{T}_h^{\text{PECE}}\right)_n = \ell_{k+1}^* h^k \boldsymbol{y}^{(k+1)}(x_n) + O(h^{k+1}),$$

we obtain

$$\left(\boldsymbol{T}_h^{\text{PECE}}\right)_n = -\frac{\ell_{k+1}^*}{\ell_{k+1}^* - \ell_{k+1}} \frac{1}{h} (\boldsymbol{u}_{n+k} - \overset{\circ}{\boldsymbol{y}}_{n+k}) + O(h^{k+1}). \tag{6.68}$$

The first term on the right of (6.68) is called the *Milne estimator* of the PECE truncation error.

The most popular choice for the predictor is a kth-order Adams–Bashforth formula, and for the corrector, the corresponding Adams–Moulton formula. Here,

$$\begin{cases} \overset{\circ}{\boldsymbol{u}}_{n+k} = \boldsymbol{u}_{n+k-1} + h \displaystyle\sum_{s=0}^{k-1} \beta_{k,s} \boldsymbol{f}_{n+s}, \\ \boldsymbol{u}_{n+k} = \boldsymbol{u}_{n+k-1} + h \left\{ \beta_{k,k}^* \boldsymbol{f}(x_{n+k}, \overset{\circ}{\boldsymbol{u}}_{n+k}) + \displaystyle\sum_{s=1}^{k-1} \beta_{k,s}^* \boldsymbol{f}_{n+s} \right\}, \\ \boldsymbol{f}_{n+k} = \boldsymbol{f}(x_{n+k}, \boldsymbol{u}_{n+k}), \end{cases} \tag{6.69}$$

with coefficients $\beta_{k,s}$, $\beta_{k,s}^*$ as defined in (6.46) and (6.57), respectively.

A predictor–corrector scheme is meaningful only if the corrector formula is more accurate than the predictor formula, not necessarily in terms of order but in terms of error coefficients. Since the principal error functions of the kth-order predictor and corrector are identical except for a multiplicative constant, ℓ_{k+1} in the case of the predictor, and ℓ^*_{k+1} in the case of the corrector, we want $|C^*_{k,k}| < |C_{k,k}|$, which, assuming Sect. 6.3.5, (6.117), is the same as

$$|\ell^*_{k+1}| < |\ell_{k+1}|. \tag{6.70}$$

For the pair of Adams formulae, we have $\ell_{k+1} = \gamma_k$ and $\ell^*_{k+1} = \gamma^*_k$ (cf. (6.50), (6.60)), and it is easy to show (see Ex. 4) that

$$|\gamma^*_k| < \frac{1}{k-1}\,\gamma_k, \quad k \geq 2, \tag{6.71}$$

so that indeed the corrector has a smaller error constant than the predictor. More precisely, it can be shown (see Ex. 5) that

$$\gamma_k \sim \frac{1}{\ln k}\,, \quad \gamma^*_k \sim -\frac{1}{k\,\ln^2 k} \;\text{ as }\; k \to \infty, \tag{6.72}$$

although both approximations are not very accurate (unless k is extremely large), the relative errors being of order $O(1/\ln k)$.

6.3 Global Description of Multistep Methods

We already commented in Sect. 6.1.1 on the fact that a linear multistep method such as (6.2) represents a system of nonlinear difference equations. To study its properties, one inevitably has to deal with the theory of difference equations. Since nonlinearities are hidden behind a (small) factor h, it turns out that the theory of *linear* difference equations with *constant* coefficients will suffice to carry through the analysis. We therefore begin with recalling the basic facts of this theory. We then define stability in a manner similar to that of Chap. 5, Sect. 5.7.1, and identify a root condition for the characteristic equation of the difference equation as the true source of stability. This, together with consistency, then immediately implies convergence, as for one-step methods.

6.3.1 Linear Difference Equations

With notations close to those adopted in Sect. 6.1 we consider a (scalar) linear difference equation of order k,

$$v_{n+k} + \alpha_{k-1}v_{n+k-1} + \cdots + \alpha_0 v_n = \varphi_{n+k}, \quad n = 0, 1, 2, \ldots, \tag{6.73}$$

where α_s are given real numbers not depending on n and not necessarily with $\alpha_0 \neq 0$, and $\{\varphi_{n+k}\}_{n=0}^{\infty}$ is a given sequence. Any sequence $\{v_n\}_{n=0}^{\infty}$ satisfying (6.73) is called a *solution* of the difference equation. It is uniquely determined by the *starting values* $v_0, v_1, \dots, v_{k-1}$. (If $\alpha_0 = \alpha_1 = \cdots = \alpha_{\ell-1} = 0$, $1 \leq \ell < k$, then $\{v_n\}_{n \geq k}$ is not affected by $v_0, v_1, \dots, v_{\ell-1}$.) Equation (6.73) is called *homogeneous* if $\varphi_{n+k} = 0$ for all $n \geq 0$ and *inhomogeneous* otherwise. It has exact order k if $\alpha_0 \neq 0$.

6.3.1.1 Homogeneous Equation

We begin with the homogeneous equation

$$v_{n+k} + \alpha_{k-1} v_{n+k-1} + \cdots + \alpha_0 v_n = 0, \quad n = 0, 1, 2, \dots. \tag{6.74}$$

We call

$$\alpha(t) = \sum_{s=0}^{k} \alpha_s t^s \quad (\alpha_k = 1) \tag{6.75}$$

the *characteristic polynomial* of (6.74) and

$$\alpha(t) = 0 \tag{6.76}$$

its *characteristic equation*. If t_s, $s = 1, 2, \dots, k'$ $(k' \leq k)$, denote the distinct roots of (6.76) and m_s their multiplicities, then the general solution of (6.74) is given by

$$v_n = \sum_{s=1}^{k'} \left(\sum_{r=0}^{m_s - 1} c_{rs} n^r \right) t_s^n, \quad n = 0, 1, 2, \dots, \tag{6.77}$$

where c_{rs} are arbitrary (real or complex) constants. There is a one-to-one correspondence between these k constants and the k starting values $v_0, v_1, \dots, v_{k-1}$.

We remark that if $\alpha_0 = 0$, then one of the roots t_s is zero, which contributes an identically vanishing solution (except for $n = 0$, where by convention $t_s^0 = 0^0 = 1$). If additional coefficients $\alpha_1 = \cdots = \alpha_{\ell-1}$, $\ell < k$, are zero, this further restricts the solution manifold. Note also that a complex root $t_s = \rho e^{i\theta}$ contributes a complex solution component in (6.77). However, since the α_s are assumed real, then with t_s also $\overline{t}_s = \rho e^{-i\theta}$ is a root of (6.76), and we can combine the two complex solutions $n^r t_s^n$ and $n^r \overline{t}_s^n$ to form a pair of *real* solutions,

$$\frac{1}{2}\, n^r (t_s^n + \overline{t}_s^n) = n^r \rho^n \cos n\theta, \quad \frac{1}{2i}\, n^r (t_s^n - \overline{t}_s^n) = n^r \rho^n \sin n\theta.$$

If we do this for each complex root t_s and select all coefficients in (6.77) real, we obtain the general solution in real form.

The following is a simple but important observation.

Theorem 6.3.1. *We have* $|v_n| \leq M$*, all* $n \geq 0$*, for every solution* $\{v_n\}$ *of the homogeneous equation (6.74), with* M *depending only on the starting values* $v_0, v_1, \dots, v_{k-1}$ *(but not on* n*) if and only if*

$$\alpha(t_s) = 0 \ \ \textit{implies} \ \begin{cases} \textit{either} \ \ |t_s| < 1 \\ \textit{or} \ \ |t_s| = 1, \ m_s = 1. \end{cases} \tag{6.78}$$

Proof. `Sufficiency of` (6.78). If (6.78) holds for every root t_s of (6.76), then every term $n^r t_s^n$ is bounded for all $n \geq 0$ (going to zero as $n \to \infty$ in the first case of (6.78) and being equal to 1 in absolute value in the second case). Since the constants c_{rs} in (6.77) are uniquely determined by the starting values $v_0, v_1, \dots, v_{k-1}$, the assertion $|v_n| \leq M$ follows.

`Necessity of` (6.78). If $|v_n| \leq M$, we cannot have $|t_s| > 1$, since we can always arrange to have $c_{rs} \neq 0$ for a corresponding term in (6.77) and select all other constants to be 0. This singles out an unbounded solution of (6.74). Nor can we have, for the same reason, $|t_s| = 1$ and $m_s > 1$. □

Condition (6.78) is referred to as the *root condition* for the difference equation (6.74) (and also for (6.73)).

Representation (6.77) of the general solution is inconvenient insofar as it does not explicitly exhibit its dependence on the starting values. A representation that does can be obtained by defining k special solutions $\{h_{n,s}\}$, $s = 0, 1, \dots, k-1$, of (6.74), having as starting values those of the unit matrix, that is,

$$h_{n,s} = \delta_{n,s} \ \text{ for } \ n = 0, 1, \dots, k-1, \tag{6.79}$$

with $\delta_{n,s}$ the Kronecker delta. Then indeed, the general solution of (6.74) is

$$v_n = \sum_{s=0}^{k-1} v_s h_{n,s}. \tag{6.80}$$

(Note that if $\alpha_0 = 0$, then $h_{0,0} = 1$ and $h_{n,0} = 0$ for all $n \geq 1$; similarly, if $\alpha_0 = \alpha_1 = 0$, and so on.)

6.3.1.2 Inhomogeneous Equation

To deal with the general inhomogeneous equation (6.73), we define for each $m = k, k+1, k+2, \dots$ the solution $\{g_{n,m}\}_{n=0}^{\infty}$ of the "initial value problem"

$$\sum_{s=0}^{k} \alpha_s g_{n+s,m} = \delta_{n,m-k}, \ \ n = 0, 1, 2, \dots,$$
$$g_{0,m} = g_{1,m} = \cdots = g_{k-1,m} = 0. \tag{6.81}$$

Here the difference equation is a very special case of (6.73), namely, with $\varphi_{n+k} = \delta_{n,m-k} = \begin{cases} 0 & \text{if } n \neq m-k \\ 1 & \text{if } n = m-k \end{cases}$ an "impulse function." We can then superimpose these special solutions to form the solution

$$v_n = \sum_{m=k}^{n} g_{n,m}\varphi_m \tag{6.82}$$

of the initial value problem $v_0 = v_1 = \cdots = v_{k-1} = 0$ for (6.73) (*Duhamel's Principle*). This is easily verified by observing first that $g_{n,m} = 0$ for $n < m$, so that (6.82) can be written in the form

$$v_n = \sum_{m=k}^{\infty} g_{n,m}\varphi_m. \tag{6.83}$$

We then have

$$\begin{aligned}\sum_{s=0}^{k} \alpha_s v_{n+s} &= \sum_{s=0}^{k} \alpha_s \sum_{m=k}^{\infty} g_{n+s,m}\varphi_m \\ &= \sum_{m=k}^{\infty} \varphi_m \sum_{s=0}^{k} \alpha_s g_{n+s,m} = \varphi_{n+k},\end{aligned}$$

where the last equation follows from (6.81).

Since effectively (6.81) is a "delayed" initial value problem for a homogeneous difference equation with k starting values $0, 0, \ldots, 0, 1$, we can express $g_{n,m}$ in (6.83) alternatively as $h_{n-m+k-1,k-1}$ (cf. (6.79)). The general solution of the inhomogeneous equation (6.73) is the general solution of the homogeneous equation plus the special solution (6.82) of the inhomogeneous equation; thus, in view of (6.80),

$$v_n = \sum_{s=0}^{k-1} v_s h_{n,s} + \sum_{m=k}^{n} h_{n-m+k-1,k-1}\varphi_m. \tag{6.84}$$

Theorem 6.3.2. *There exists a constant $M > 0$, independent of n, such that*

$$|v_n| \leq M \left\{ \max_{0\leq s\leq k-1} |v_s| + \sum_{m=k}^{n} |\varphi_m| \right\}, \quad n = 0, 1, 2, \ldots, \tag{6.85}$$

for every solution $\{v_n\}$ of (6.73) and for every $\{\varphi_{n+k}\}$, if and only if the characteristic polynomial $\alpha(t)$ of (6.73) satisfies the root condition (6.78).

Proof. `Sufficiency of` (6.78). By Theorem 6.3.1 the inequality (6.85) follows immediately from (6.84), with M the constant of Theorem 6.3.1.

`Necessity of` (6.78). Take $\varphi_m = 0$, all $m \geq k$, in which case $\{v_n\}$ is a bounded solution of the homogeneous equation, and hence, by Theorem 6.3.1, the root condition must hold. □

As an application of Theorems 6.3.1 and 6.3.2, we take $v_n = h_{n,s}$ (cf. (6.79)) and $v_n = g_{n,m}$ (cf. (6.81)). The former is bounded by M, by Theorem 6.3.1, and so is the latter, by (6.85), since $g_{n,s} = 0$ for $0 \leq s \leq k-1$ and all φ_m are 0 except one, which is 1. Thus, if the root condition is satisfied, then

$$|h_{n,s}| \leq M, \quad |g_{n,m}| \leq M, \quad \text{all } n \geq 0. \tag{6.86}$$

Note that, since $h_{s,s} = 1$, we must have $M \geq 1$.

6.3.2 Stability and Root Condition

We now return to the general multistep method (6.2) for solving the initial value problem (6.1). In terms of the residual operator R_h of (6.14) we define stability, similarly as for one-step methods (cf. Chap. 5, Sect. 5.7.1), as follows.

Definition 6.3.1. Method (6.2) is called *stable* on $[a,b]$ if there exists a constant $K > 0$ not depending on h such that for an arbitrary (uniform) grid h on $[a,b]$ and for arbitrary two grid functions $\boldsymbol{v}, \boldsymbol{w} \in \Gamma_h[a,b]$, there holds

$$\|\boldsymbol{v} - \boldsymbol{w}\|_\infty \leq K \left(\max_{0 \leq s \leq k-1} \|\boldsymbol{v}_s - \boldsymbol{w}_s\| + \|R_h \boldsymbol{v} - R_h \boldsymbol{w}\|_\infty \right), \quad \boldsymbol{v}, \boldsymbol{w} \in \Gamma_h[a,b], \tag{6.87}$$

for all h sufficiently small.

The motivation for this "stability inequality" is much the same as for one-step methods (Chap. 5, (5.83)–(5.85)), and we do not repeat it here.

Let $\mathcal{F}$ be the family of functions $\boldsymbol{f}$ satisfying a uniform Lipschitz condition

$$\|\boldsymbol{f}(x,\boldsymbol{y}) - \boldsymbol{f}(x,\boldsymbol{y}^*)\| \leq L\|\boldsymbol{y} - \boldsymbol{y}^*\|, \quad x \in [a,b], \quad \boldsymbol{y}, \boldsymbol{y}^* \in \mathbb{R}^d, \tag{6.88}$$

with Lipschitz constant $L = L_f$ depending on $\boldsymbol{f}$.

Theorem 6.3.3. *The multistep method (6.2) is stable for every $\boldsymbol{f} \in \mathcal{F}$ if and only if its characteristic polynomial (6.75) satisfies the root condition (6.78).*

Proof. `Necessity of` (6.78). Consider $\boldsymbol{f}(x,\boldsymbol{y}) \equiv \boldsymbol{0}$, which is certainly in $\mathcal{F}$, and for which

$$(R_h \boldsymbol{v})_n = \frac{1}{h} \sum_{s=0}^{k} \alpha_s \boldsymbol{v}_{n+s}.$$

Take $\boldsymbol{v} = \boldsymbol{u}$ and $\boldsymbol{w} = \boldsymbol{0}$ in (6.87), where $\boldsymbol{u}$ is a grid function satisfying

$$\sum_{s=0}^{k} \alpha_s \boldsymbol{u}_{n+s} = \boldsymbol{0}, \quad n = 0, 1, 2, \ldots . \tag{6.89}$$

Since (6.87) is to hold for arbitrarily fine grids, the integer n in (6.89) can assume arbitrarily large values, and it follows from (6.87) and $R_h \boldsymbol{u} = \boldsymbol{0}$ that $\boldsymbol{u}$ is uniformly bounded,

$$\|\boldsymbol{u}\|_\infty \le K \max_{0 \le s \le k-1} \|\boldsymbol{u}_s\|, \tag{6.90}$$

the bound depending only on the starting values. Since $\boldsymbol{u}$ is a solution of the homogeneous difference equation (6.74), its characteristic polynomial must satisfy the root condition by Theorem 6.3.1.

Sufficiency of (6.78). Let $\boldsymbol{f} \in \mathcal{F}$ and $\boldsymbol{v}, \boldsymbol{w} \in \Gamma_h[a, b]$ be arbitrary grid functions. By definition of R_h, we have

$$\sum_{s=0}^{k} \alpha_s \boldsymbol{v}_{n+s} = h \sum_{s=0}^{k} \beta_s \boldsymbol{f}(x_{n+s}, \boldsymbol{v}_{n+s}) + h(R_h \boldsymbol{v})_n, \quad n = 0, 1, 2, \ldots, N - k,$$

and similarly for $\boldsymbol{w}$. Subtraction then gives

$$\sum_{s=0}^{k} \alpha_s (\boldsymbol{v}_{n+s} - \boldsymbol{w}_{n+s}) = \boldsymbol{\varphi}_{n+k}, \quad n = 0, 1, 2, \ldots, N - k,$$

where

$$\boldsymbol{\varphi}_{n+k} = h \sum_{s=0}^{h} \beta_s [\boldsymbol{f}(x_{n+s}, \boldsymbol{v}_{n+s}) - \boldsymbol{f}(x_{n+s}, \boldsymbol{w}_{n+s})] + h[(R_h \boldsymbol{v})_n - (R_h \boldsymbol{w})_n]. \tag{6.91}$$

Therefore, $\boldsymbol{v} - \boldsymbol{w}$ is formally a solution of the inhomogeneous difference equation (6.73) (the forcing function $\boldsymbol{\varphi}_{n+k}$, though, depending also on $\boldsymbol{v}$ and $\boldsymbol{w}$), so that by (6.80) and (6.83) we can write

$$\boldsymbol{v}_n - \boldsymbol{w}_n = \sum_{s=0}^{k-1} h_{n,s} (\boldsymbol{v}_s - \boldsymbol{w}_s) + \sum_{m=k}^{n} g_{n,m} \boldsymbol{\varphi}_m.$$

Since the root condition is satisfied, we have by (6.86)

$$|h_{n,s}| \le M, \quad |g_{n,m}| \le M$$

for some constant $M \ge 1$, uniformly in n and m. Therefore,

$$\|\boldsymbol{v}_n - \boldsymbol{w}_n\| \le M \left\{ k \max_{0 \le s \le k-1} \|\boldsymbol{v}_s - \boldsymbol{w}_s\| + \sum_{m=k}^{n} \|\boldsymbol{\varphi}_m\| \right\}, \\ n = 0, 1, 2, \ldots, N. \tag{6.92}$$

By (6.91) we can estimate

$$\begin{aligned}\|\boldsymbol{\varphi}_m\| &= \left\| h\sum_{s=0}^{k}\beta_s[\boldsymbol{f}(x_{m-k+s},\boldsymbol{v}_{m-k+s})-\boldsymbol{f}(x_{m-k+s},\boldsymbol{w}_{m-k+s})]\right.\\ &\qquad \left. + h[(R_h\boldsymbol{v})_{m-k}-(R_h\boldsymbol{w})_{m-k}]\right\|\\ &\le h\beta L\sum_{s=0}^{k}\|\boldsymbol{v}_{m-k+s}-\boldsymbol{w}_{m-k+s}\| + h\|R_h\boldsymbol{v}-R_h\boldsymbol{w}\|_\infty, \end{aligned} \tag{6.93}$$

where

$$\beta = \max_{0\le s\le k}|\beta_s|.$$

Letting $\boldsymbol{e}=\boldsymbol{v}-\boldsymbol{w}$ and $\boldsymbol{r}_h = R_h\boldsymbol{v}-R_h\boldsymbol{w}$, we obtain from (6.92) and (6.93)

$$\|\boldsymbol{e}_n\| \le M\left\{k\max_{0\le s\le k-1}\|\boldsymbol{e}_s\| + h\beta L\sum_{m=k}^{n}\sum_{s=0}^{k}\|\boldsymbol{e}_{m-k+s}\| + Nh\|\boldsymbol{r}_h\|_\infty\right\}.$$

Noting that

$$\begin{aligned}\sum_{m=k}^{n}\sum_{s=0}^{k}\|\boldsymbol{e}_{m-k+s}\| &= \sum_{s=0}^{k}\sum_{m=k}^{n}\|\boldsymbol{e}_{m-k+s}\| \le \sum_{s=0}^{k}\sum_{m=0}^{n}\|\boldsymbol{e}_m\|\\ &= (k+1)\sum_{m=0}^{n}\|\boldsymbol{e}_m\|\end{aligned}$$

and using $Nh = b-a$, we get

$$\|\boldsymbol{e}_n\| \le M\left\{k\max_{0\le s\le k-1}\|\boldsymbol{e}_s\| + h(k+1)\beta L\sum_{m=0}^{n}\|\boldsymbol{e}_m\| + (b-a)\|\boldsymbol{r}_h\|_\infty\right\}. \tag{6.94}$$

Now let h be so small that

$$1-h(k+1)\beta LM \ge \frac{1}{2}.$$

Then, splitting off the term with $\|\boldsymbol{e}_n\|$ on the right of (6.94) and moving it to the left, we obtain

$$\begin{aligned}&(1-h(k+1)\beta LM)\|\boldsymbol{e}_n\|\\ &\quad\le M\left\{k\max_{0\le s\le k-1}\|\boldsymbol{e}_s\| + h(k+1)\beta L\sum_{m=0}^{n-1}\|\boldsymbol{e}_m\| + (b-a)\|\boldsymbol{r}_h\|_\infty\right\},\end{aligned}$$

or

$$\|\boldsymbol{e}_n\| \leq 2M \left\{ h(k+1)\beta L \sum_{m=0}^{n-1} \|\boldsymbol{e}_m\| + k \max_{0 \leq s \leq k-1} \|\boldsymbol{e}_s\| + (b-a)\|\boldsymbol{r}_h\|_\infty \right\}.$$

Thus,

$$\|\boldsymbol{e}_n\| \leq hA \sum_{m=0}^{n-1} \|\boldsymbol{e}_m\| + B, \tag{6.95}$$

where

$$A = 2(k+1)\beta LM, \quad B = 2M \left(k \max_{0 \leq s \leq k-1} \|\boldsymbol{e}_s\| + (b-a)\|\boldsymbol{r}_h\|_\infty \right). \tag{6.96}$$

Consider, along with (6.95), the difference equation

$$E_n = hA \sum_{m=0}^{n-1} E_m + B, \quad E_0 = B. \tag{6.97}$$

It is easily seen by induction that

$$E_n = B(1+hA)^n, \quad n = 0, 1, 2, \dots . \tag{6.98}$$

Subtracting (6.97) from (6.95), we get

$$\|\boldsymbol{e}_n\| - E_n \leq hA \sum_{m=0}^{n-1} (\|\boldsymbol{e}_m\| - E_m). \tag{6.99}$$

Clearly, $\|\boldsymbol{e}_0\| \leq B = E_0$. Thus, by (6.99), $\|\boldsymbol{e}_1\| - E_1 \leq 0$, and by induction on n,

$$\|\boldsymbol{e}_n\| \leq E_n, \quad n = 0, 1, 2, \dots .$$

Thus, by (6.98),

$$\|\boldsymbol{e}_n\| \leq B(1+hA)^n \leq B\mathrm{e}^{nhA} \leq B\mathrm{e}^{(b-a)A}.$$

Recalling the definition of B in (6.96), we find

$$\|\boldsymbol{e}_n\| \leq 2M\mathrm{e}^{(b-a)A} \left\{ k \max_{0 \leq s \leq k-1} \|\boldsymbol{e}_s\| + (b-a)\|R_h\boldsymbol{v} - R_h\boldsymbol{w}\|_\infty \right\},$$

which is the stability inequality (6.87) with

$$K = 2M\mathrm{e}^{(b-a)A}\max\{k, b-a\}. \qquad \square$$

Theorem 6.3.3 in particular shows that all Adams methods are stable, since for them

$$\alpha(t) = t^k - t^{k-1} = t^{k-1}(t-1),$$

and the root condition is trivially satisfied.

Theorem 6.3.3 also holds for predictor–corrector methods of the type considered in (6.64), if one defines the residual operator R_h^{PECE} in the definition of stability in the obvious way:

$$(R_h^{\mathrm{PECE}}\boldsymbol{v})_n = \frac{1}{h}\sum_{s=1}^{k}\alpha_s^*\boldsymbol{v}_{n+s} - \left\{\beta_k^*\boldsymbol{f}(x_{n+k}, \overset{\circ}{\boldsymbol{v}}_{n+k}) + \sum_{s=1}^{k-1}\beta_s^*\boldsymbol{f}(x_{n+s}, \boldsymbol{v}_{n+s})\right\}, \tag{6.100}$$

with

$$\overset{\circ}{\boldsymbol{v}}_{n+k} = -\sum_{s=0}^{k-1}\alpha_s\boldsymbol{v}_{n+s} + h\sum_{s=0}^{k-1}\beta_s\boldsymbol{f}(x_{n+s}, \boldsymbol{v}_{n+s}), \tag{6.101}$$

and considers the characteristic polynomial to be that of the corrector formula (see Ex. 7).

The problem of constructing stable multistep methods of maximum order is considered later in Sect. 6.4.

6.3.3 Convergence

With the powerful property of stability at hand, the convergence of multistep methods follows almost immediately as a corollary. We first define what we mean by convergence.

Definition 6.3.2. Consider a uniform grid on $[a, b]$ with grid length h. Let $\boldsymbol{u} = \{\boldsymbol{u}_n\}$ be the grid function obtained by applying the multistep method (6.2) on $[a, b]$, with starting approximations $\boldsymbol{u}_s(h)$ as in (6.4). Let $\boldsymbol{y} = \{\boldsymbol{y}_n\}$ be the grid function induced by the exact solution of the initial value problem on $[a, b]$. Method (6.2) is said to converge on $[a, b]$ if there holds

$$\|\boldsymbol{u} - \boldsymbol{y}\|_\infty \to 0 \text{ as } h \to 0 \tag{6.102}$$

whenever

$$\boldsymbol{u}_s(h) \to \boldsymbol{y}_0 \text{ as } h \to 0,\ \ s = 0, 1, \ldots, k-1. \tag{6.103}$$

Theorem 6.3.4. *The multistep method (6.2) converges for all $\boldsymbol{f} \in \mathcal{F}$* (cf. (6.88)) *if and only if it is consistent and stable. If, in addition, (6.2) has order p and $\boldsymbol{u}_s(h) - \boldsymbol{y}(x_s) = O(h^p)$, $s = 0, 1, \dots, k-1$, then*

$$\|\boldsymbol{u} - \boldsymbol{y}\|_\infty = O(h^p) \;\; as \;\; h \to 0. \tag{6.104}$$

Proof. `Necessity`. Let $\boldsymbol{f} \equiv \boldsymbol{0}$ (which is certainly in $\mathcal{F}$) and $\boldsymbol{y}_0 = \boldsymbol{0}$. Then $\boldsymbol{y}(x) \equiv \boldsymbol{0}$ and (6.2) reduces to (6.89). Since the same relations hold for each component of $\boldsymbol{y}$ and $\boldsymbol{u}$, we may as well consider a scalar problem. (This holds for the rest of the necessity part of the proof.) Assume first, by way of contradiction, that (6.2) is not stable. Then, by Theorem 6.3.3, there is a root t_s of the characteristic equation $\alpha(t) = 0$ for which either $|t_s| > 1$, or $|t_s| = 1$ and $m_s > 1$. In the first case, (6.89) has a solution $u_n = ht_s^n$ for which $|u_n| = h|t_s|^n$. Clearly, the starting values $u_0, u_1, \dots, u_{k-1}$ all tend to $y_0 = 0$ as $h \to 0$, but $|u_n| \to \infty$ as $n \to \infty$. Since for h sufficiently small we can have n arbitrarily large, this contradicts convergence of $\{u_n\}$ to the solution $\{y_n\}$, $y_n \equiv 0$. The same argument applies in the second case if we consider (say) $u_n = h^{\frac{1}{2}} n^{m_s - 1} t_s^n$, where now $|t_s| = 1$, $m_s > 1$. This proves the necessity of stability.

To prove consistency, we must show that $\alpha(1) = 0$ and $\alpha'(1) = \beta(1)$, where (anticipating (6.122)) we define $\beta(t) = \sum_{s=0}^k \beta_s t^s$. For the former, we consider $f \equiv 0$, $y_0 = 1$, which has the exact solution $y(x) \equiv 1$ and the numerical solution still satisfying (6.89). If we take $u_s = 1$, $s = 0, 1, \dots, k-1$, the assumed convergence implies that $u_{n+s} \to 1$ as $h \to 0$, hence $0 = \sum_{s=0}^k \alpha_s u_{n+s} \to \alpha(1)$. For the latter, we consider $f \equiv 1$ and $y_0 = 0$, that is, $y(x) \equiv x - a$. The multistep method now generates a grid function $u = \{u_n\}$ satisfying

$$\sum_{s=0}^k \alpha_s u_{n+s} - \beta(1)h = 0.$$

A particular solution is given by $u_n = \frac{\beta(1)}{\alpha'(1)} nh$. Indeed, since $\alpha(1) = 0$, as already shown, we have

$$\begin{aligned}\sum_{s=0}^k \alpha_s u_{n+s} - \beta(1)h &= \frac{\beta(1)}{\alpha'(1)} h \sum_{s=0}^k \alpha_s (n+s) - \beta(1)h \\ &= \frac{\beta(1)}{\alpha'(1)} h[n\alpha(1) + \alpha'(1)] - \beta(1)h \\ &= \frac{\beta(1)}{\alpha'(1)} h\alpha'(1) - \beta(1)h = 0.\end{aligned}$$

Since also $u_s \to y_0 = 0$ for $h \to 0$, $s = 0, 1, \dots, k-1$, and since by assumption $\{u_n\}$ converges to $\{y_n\}$, $y_n = nh$, we must have $\frac{\beta(1)}{\alpha'(1)} = 1$, that is, $\alpha'(1) = \beta(1)$.

Sufficiency. Letting $\boldsymbol{v} = \boldsymbol{u}$ and $\boldsymbol{w} = \boldsymbol{y}$ in the stability inequality (6.87), and noting that $R_h\boldsymbol{u} = 0$ and $R_h\boldsymbol{y} = \boldsymbol{T}_h$, the grid function $\{(\boldsymbol{T}_h)_n\}$ of the truncation errors (cf. (6.15)), we get

$$\|\boldsymbol{u} - \boldsymbol{y}\|_\infty \le K\left\{\max_{0\le s\le k-1}\|\boldsymbol{u}_s - \boldsymbol{y}(x_s)\| + \|\boldsymbol{T}_h\|_\infty\right\}. \tag{6.105}$$

If method (6.2) is consistent, the second term in braces tends to zero and, therefore, also the term on the left, if (6.103) holds, that is, if $\boldsymbol{u}_s - \boldsymbol{y}(x_s) \to 0$ as $h \to 0$. This completes the proof of the first part of the theorem. The second part follows likewise, since by assumption both terms between braces in (6.105) are of $O(h^p)$. □

In view of the remark near the end of Sect. 6.3.2, Theorem 6.3.4 holds also for the kth-order predictor–corrector method (6.64) with $p = k$. The proof is the same, since for the truncation error, $\|\boldsymbol{T}_h^{\text{PECE}}\|_\infty = O(h^k)$, as was shown in Sect. 6.2.3.

6.3.4 Asymptotics of Global Error

A refinement of Theorem 6.3.4, (6.104), exhibiting the leading term in the global discretization error, is given by the following theorem.

Theorem 6.3.5. *Assume that*

(1) $\boldsymbol{f}(x, \boldsymbol{y}) \in C^2$ *on* $[a, b] \times \mathbb{R}^d$*;*
(2) *the multistep method (6.2) is stable (i.e., satisfies the root condition) and has order* $p \ge 1$*;*
(3) *the exact solution* $\boldsymbol{y}(x)$ *of (6.1) is of class* $C^{p+2}[a, b]$*;*
(4) *the starting approximations (6.4) satisfy*

$$\boldsymbol{u}_s - \boldsymbol{y}(x_s) = O(h^{p+1}) \text{ as } h \to 0, \quad s = 0, 1, \dots, k-1;$$

(5) $\boldsymbol{e}(x)$ *is the solution of the linear initial value problem*

$$\frac{\mathrm{d}\boldsymbol{e}}{\mathrm{d}x} = \boldsymbol{f}_y(x, \boldsymbol{y}(x))\boldsymbol{e} - \boldsymbol{y}^{(p+1)}(x), \quad \boldsymbol{e}(a) = 0. \tag{6.106}$$

Then, for $n = 0, 1, 2, \dots, N$*,*

$$\boldsymbol{u}_n - \boldsymbol{y}(x_n) = C_{k,p}h^p\boldsymbol{e}(x_n) + O(h^{p+1}) \text{ as } h \to 0, \tag{6.107}$$

where $C_{k,p}$ *is the error constant of (6.2), that is,*

$$C_{k,p} = \frac{\ell_{p+1}}{\sum_{s=0}^k \beta_s}, \quad \ell_{p+1} = L\,\frac{t^{p+1}}{(p+1)!}. \tag{6.108}$$

Before proving the theorem, we make the following remarks:

1. Under the assumptions made in (1) and (3), the solution $\boldsymbol{e}$ of (6.106) exists uniquely on $[a, b]$. It is the same for all multistep methods of order p.
2. The error constant $C_{k,p}$ depends only (through the coefficients α_s, β_s) on the multistep method and not on the initial value problem to be solved.
3. For a given differential system (6.1), the asymptotically best k-step method of order p would be one for which $|C_{k,p}|$ is smallest. Unfortunately, as we see later in Sect. 6.4.3, the minimum of $|C_{k,p}|$ over all *stable* k-step methods of order p cannot be generally attained.
4. Stability and $p \geq 1$ implies $\sum_{s=0}^{k} \beta_s \neq 0$. In fact, $\alpha(1) = \sum_{s=0}^{k} \alpha_s = 0$, since $L1 = 0$, and $Lt = \sum_{s=0}^{k} s\alpha_s - \sum_{k=0}^{k} \beta_s = 0$ since $p \geq 1$. Consequently,

$$\sum_{s=0}^{k} \beta_s = \sum_{s=0}^{k} s\alpha_s = \alpha'(1) \neq 0 \tag{6.109}$$

by the root condition. Actually, $\sum_{s=0}^{k} \beta_s > 0$, as we show later in Sect. 6.4.3.

Proof of Theorem 6.3.5. Define the grid function $\boldsymbol{r} = \{\boldsymbol{r}_n\}$ by

$$\boldsymbol{r} = h^{-p}(\boldsymbol{u} - \boldsymbol{y}). \tag{6.110}$$

Then

$$\begin{aligned}\frac{1}{h}\sum_{s=0}^{k} \alpha_s \boldsymbol{r}_{n+s} &= h^{-p}\left[\frac{1}{h}\sum_{s=0}^{k} \alpha_s \boldsymbol{u}_{n+s} - \frac{1}{h}\sum_{s=0}^{k} \alpha_s \boldsymbol{y}(x_{n+s})\right] \\ &= h^{-p}\left[\sum_{s=0}^{k} \beta_s \boldsymbol{f}(x_{n+s}, \boldsymbol{u}_{n+s}) - \sum_{s=0}^{k} \beta_s \boldsymbol{y}'(x_{n+s}) - (\boldsymbol{T}_h)_n\right],\end{aligned}$$

where $\boldsymbol{T}_h$ is the truncation error defined in (6.19). Expanding $\boldsymbol{f}$ about the exact solution trajectory and noting the form of the principal error function given in Theorem 6.1.3, we obtain

$$\begin{aligned}\frac{1}{h}\sum_{s=0}^{k} \alpha_s \boldsymbol{r}_{n+s} = h^{-p}\Bigg[&\sum_{s=0}^{k} \beta_s \{\boldsymbol{f}(x_{n+s}, \boldsymbol{y}(x_{n+s})) \\ &+ \boldsymbol{f}_{\boldsymbol{y}}(x_{n+s}, \boldsymbol{y}(x_{n+s}))(\boldsymbol{u}_{n+s} - \boldsymbol{y}(x_{n+s})) + O(h^{2p})\} \\ &-\sum_{s=0}^{k} \beta_s \boldsymbol{y}'(x_{n+s}) - \ell_{p+1}\boldsymbol{y}^{(p+1)}(x_n)h^p + O(h^{p+1})\Bigg],\end{aligned}$$

having used Assumption (1) and the fact that $\boldsymbol{u}_{n+s} - \boldsymbol{y}(x_{n+s}) = O(h^p)$ by Theorem 6.3.4. Now the sums over $\boldsymbol{f}$ and $\boldsymbol{y}'$ cancel, since $\boldsymbol{y}$ is the solution of the differential system (6.1). Furthermore, $O(h^{2p})$ is of $O(h^{p+1})$ since $p \geq 1$, and making use of the definition (6.110) of $\boldsymbol{r}$, we can simplify the preceding to

$$\begin{aligned}\frac{1}{h}\sum_{s=0}^{k}\alpha_s \boldsymbol{r}_{n+s} &= h^{-p}\left[\sum_{s=0}^{k}\beta_s \boldsymbol{f}_{\boldsymbol{y}}(x_{n+s}, \boldsymbol{y}(x_{n+s}))h^p \boldsymbol{r}_{n+s}\right.\\ &\qquad \left. - \ell_{p+1}\boldsymbol{y}^{(p+1)}(x_n)h^p + O(h^{p+1})\right]\\ &= \sum_{s=0}^{k}\beta_s \boldsymbol{f}_{\boldsymbol{y}}(x_{n+s}, \boldsymbol{y}(x_{n+s}))\boldsymbol{r}_{n+s} - \ell_{p+1}\boldsymbol{y}^{(p+1)}(x_n) + O(h).\end{aligned}$$

Now

$$\begin{aligned}\sum_{s=0}^{k}\beta_s \boldsymbol{y}^{(p+1)}(x_{n+s}) &= \sum_{s=0}^{k}\beta_s\left[\boldsymbol{y}^{(p+1)}(x_n) + O(h)\right]\\ &= \left(\sum_{s=0}^{k}\beta_s\right)\boldsymbol{y}^{(p+1)}(x_n) + O(h),\end{aligned}$$

so that

$$\begin{aligned}\ell_{p+1}\boldsymbol{y}^{(p+1)}(x_n) &= \frac{\ell_{p+1}}{\sum_{s=0}^{k}\beta_s}\sum_{s=0}^{k}\beta_s \boldsymbol{y}^{(p+1)}(x_{n+s}) + O(h)\\ &= C_{k,p}\sum_{s=0}^{k}\beta_s \boldsymbol{y}^{(p+1)}(x_{n+s}) + O(h),\end{aligned}$$

by the definition (6.108) of the error constant $C_{k,p}$. Thus,

$$\frac{1}{h}\sum_{s=0}^{k}\alpha_s \boldsymbol{r}_{n+s} - \sum_{s=0}^{k}\beta_s[\boldsymbol{f}_{\boldsymbol{y}}(x_{n+s}, \boldsymbol{y}(x_{n+s}))\boldsymbol{r}_{n+s} - C_{k,p}\boldsymbol{y}^{(p+1)}(x_{n+s})] = O(h).$$

Defining

$$\overset{\circ}{\boldsymbol{r}} = \frac{1}{C_{k,p}}\boldsymbol{r}, \tag{6.111}$$

we finally get

$$\frac{1}{h}\sum_{s=0}^{k}\alpha_s \overset{\circ}{\boldsymbol{r}}_{n+s} - \sum_{s=0}^{k}\beta_s[\boldsymbol{f}_{\boldsymbol{y}}(x_{n+s}, \boldsymbol{y}(x_{n+s}))\overset{\circ}{\boldsymbol{r}}_{n+s} - \boldsymbol{y}^{(p+1)}(x_{n+s})] = O(h). \tag{6.112}$$

The left-hand side can now be viewed as the residual operator R_h^g of the multistep method (6.2) applied to the grid function $\overset{\circ}{\boldsymbol{r}}$, not for the original differential system (6.1), however, but for the linear system (6.106) with right-hand side

$$\boldsymbol{g}(x,\boldsymbol{e}) := \boldsymbol{f}_y(x,\boldsymbol{y}(x))\boldsymbol{e} - \boldsymbol{y}^{(p+1)}(x), \;\; a \le x \le b, \;\; \boldsymbol{e} \in \mathbb{R}^d. \tag{6.113}$$

That is,

$$\|R_h^g \overset{\circ}{\boldsymbol{r}}\|_\infty = O(h). \tag{6.114}$$

For the exact solution $\boldsymbol{e}(x)$ of (6.106), we have likewise

$$\|R_h^g \boldsymbol{e}\|_\infty = O(h), \tag{6.115}$$

since by (6.15) (applied to the linear system $\boldsymbol{e}' = \boldsymbol{g}(x,\boldsymbol{e})$) R_h^g is the truncation error of the multistep method (6.2), and its order is $p \ge 1$. Since by Assumption (2) the multistep method (6.2) is stable, we can apply the stability inequality (6.87) (for the system $\boldsymbol{e}' = \boldsymbol{g}(x,\boldsymbol{e})$) to the two grid functions $\boldsymbol{v} = \overset{\circ}{\boldsymbol{r}}$ and $\boldsymbol{w} = \boldsymbol{e}$, giving

$$\begin{aligned}\|\overset{\circ}{\boldsymbol{r}} - \boldsymbol{e}\|_\infty &\le K\left(\max_{0\le s\le k-1} \|\overset{\circ}{\boldsymbol{r}}_s - \boldsymbol{e}(x_s)\| + \|R_h^g \overset{\circ}{\boldsymbol{r}} - R_h^g \boldsymbol{e}\|_\infty\right) \\ &= K\left(\max_{0\le s\le k-1} \|\overset{\circ}{\boldsymbol{r}}_s - \boldsymbol{e}(x_s)\| + O(h)\right).\end{aligned} \tag{6.116}$$

It remains to observe that, for $0 \le s \le k-1$,

$$\overset{\circ}{\boldsymbol{r}}_s - \boldsymbol{e}(x_s) = \frac{1}{C_{k,p}}\boldsymbol{r}_s - \boldsymbol{e}(x_s) = \frac{1}{C_{k,p}} h^{-p}[\boldsymbol{u}_s - \boldsymbol{y}(x_s)] - \boldsymbol{e}(x_s),$$

and hence, by Assumption (4) and $\boldsymbol{e}(a) = \boldsymbol{0}$,

$$\overset{\circ}{\boldsymbol{r}}_s - \boldsymbol{e}(x_s) = O(h) - [\boldsymbol{e}(a) + sh\boldsymbol{e}'(\xi_s)] = O(h),$$

to conclude

$$\max_{0\le s\le k-1} \|\overset{\circ}{\boldsymbol{r}}_s - \boldsymbol{e}(x_s)\| = O(h)$$

and, therefore, by (6.116), (6.111), and (6.110),

$$\|\boldsymbol{u} - \boldsymbol{y} - C_{k,p}h^p\boldsymbol{e}\|_\infty = O(h^{p+1}),$$

as was to be shown. □

The proof of Theorem 6.3.5 applies to the predictor–corrector method (6.64) with $C_{k,p}$ replaced by $C_{k,k}^* = \ell_{k+1}^* / \sum_{s=0}^k \beta_s^*$, the error constant for the corrector formula, once one has shown that $\overset{\circ}{\boldsymbol{u}}_{n+k} - \boldsymbol{u}_{n+k} = O(h^{k+1})$ in (6.64), and

$\overset{\circ}{\boldsymbol{y}}_{n+k} - \boldsymbol{y}(x_{n+k}) = O(h^{k+1})$. The first relation is true for special predictor–corrector schemes (cf. (6.17), (6.120)), whereas the second says that h times the truncation error of the predictor formula has the order shown (cf. Sect. 6.2.3).

6.3.5 Estimation of Global Error

It is natural to try, similarly as with one-step methods (cf. Chap. 5, Sect. 5.8.1), to estimate the leading term in the asymptotic formula (6.107) of the global error by integrating with Euler's method the "variational equation" (6.106) along with the multistep integration of the principal equation (6.1). The main technical difficulty is to correctly estimate the driving function $\boldsymbol{y}^{(p+1)}$ in the linear system (6.106). It turns out, however, that Milne's procedure (cf. (6.68)) for estimating the local truncation error in predictor–corrector schemes can be extended to estimate the global error as well, provided the predictor and corrector formulae have the same characteristic polynomial, more precisely, if in the predictor–corrector scheme (6.64) there holds

$$\alpha_s^* = \alpha_s \;\text{ for }\; s = 1, 2, \ldots, k; \;\; \alpha_0 = 0. \tag{6.117}$$

This is true, in particular, for the Adams predictor–corrector scheme (6.69). We formulate the procedure in the following theorem.

Theorem 6.3.6. *Assume that*

(1) $\boldsymbol{f}(x, \boldsymbol{y}) \in C^2$ *on* $[a, b] \times \mathbb{R}^d$*;*
(2) *the predictor–corrector scheme (6.64) is based on a pair of kth-order formulae* $(k \geq 1)$ *satisfying (6.117) and having local error constants* ℓ_{k+1}, ℓ_{k+1}^* *for the predictor and corrector, respectively;*
(3) *the exact solution* $\boldsymbol{y}(x)$ *of (6.1) is of class* $C^{k+2}[a, b]$*;*
(4) *the starting approximations (6.4) satisfy*

$$\boldsymbol{u}_s - \boldsymbol{y}(x_s) = O(h^{k+1}) \;\; as \;\; h \to 0, \;\; s = 0, 1, \ldots, k-1;$$

(5) *along with the grid function* $\boldsymbol{u} = \{\boldsymbol{u}_n\}$ *constructed by the predictor–corrector scheme, we generate the grid function* $\boldsymbol{v} = \{\boldsymbol{v}_n\}$ *in the following manner* (where $\overset{\circ}{\boldsymbol{u}}_{n+k}$ is defined as in (6.64)):

$$\boldsymbol{v}_s = 0, \quad s = 0, 1, \ldots, k-1;$$

$$\boldsymbol{v}_{n+k} = \boldsymbol{v}_{n+k-1} + h\left\{\boldsymbol{f}_y(x_n, \boldsymbol{u}_n)\boldsymbol{v}_n + \frac{h^{-(k+1)}}{\ell_{k+1}^* - \ell_{k+1}}(\overset{\circ}{\boldsymbol{u}}_{n+k} - \boldsymbol{u}_{n+k})\right\}, \quad n = 0, 1, 2, \ldots, N-k. \tag{6.118}$$

Then, for $n = 0, 1, \ldots, N$*,*

$$\boldsymbol{u}_n - \boldsymbol{y}(x_n) = C_{k,k}^* h^k \boldsymbol{v}_n + O(h^{k+1}) \;\; as \;\; h \to 0, \tag{6.119}$$

where $C_{k,k}^*$ *is the (global) error constant for the corrector formula.*

Proof. The proof is the same as that of Theorem 5.8.1, once it has been shown, in place of Chap. 5, (5.117), that

$$\frac{h^{-(k+1)}}{\ell_{k+1}^{*}-\ell_{k+1}}(\overset{\circ}{\boldsymbol{u}}_{n+k}-\boldsymbol{u}_{n+k})=\boldsymbol{y}^{(k+1)}(x_n)+O(h). \tag{6.120}$$

We now proceed to establish (6.120).

By (6.64), we have

$$\begin{aligned}\overset{\circ}{\boldsymbol{u}}_{n+k}-\boldsymbol{u}_{n+k}&=\sum_{s=1}^{k-1}\alpha_s^{*}\boldsymbol{u}_{n+s}-\sum_{s=0}^{k-1}\alpha_s\boldsymbol{u}_{n+s}\\&\quad+h\left\{\sum_{s=0}^{k-1}\beta_s\boldsymbol{f}(x_{n+s},\boldsymbol{u}_{n+s})-\beta_k^{*}\boldsymbol{f}(x_{n+k},\overset{\circ}{\boldsymbol{u}}_{n+k})-\sum_{s=1}^{k-1}\beta_s^{*}\boldsymbol{f}(x_{n+s},\boldsymbol{u}_{n+s})\right\}.\end{aligned}$$

The first two sums on the right cancel because of (6.117). (It is here where $\alpha_0=0$ is used.) In the expression between braces, we expand each $\boldsymbol{f}$ about the exact solution trajectory to obtain

$$\begin{aligned}\{\cdots\}&=\sum_{s=0}^{k-1}\beta_s\left[\boldsymbol{f}(x_{n+s},\boldsymbol{y}(x_{n+s}))+\boldsymbol{f}_{\boldsymbol{y}}(x_{n+s},\boldsymbol{y}(x_{n+s}))(\boldsymbol{u}_{n+s}-\boldsymbol{y}(x_{n+s}))\right.\\&\qquad\left.+\ O(h^{2k})\right]\\&\quad-\beta_k^{*}\left[\boldsymbol{f}(x_{n+k},\boldsymbol{y}(x_{n+k}))+\boldsymbol{f}_{\boldsymbol{y}}(x_{n+k},\boldsymbol{y}(x_{n+k}))(\overset{\circ}{\boldsymbol{u}}_{n+k}-\boldsymbol{y}(x_{n+k}))+\ O(h^{2k})\right]\\&\quad-\sum_{s=1}^{k-1}\beta_s^{*}\left[\boldsymbol{f}(x_{n+s},\boldsymbol{y}(x_{n+s}))+\boldsymbol{f}_{\boldsymbol{y}}(x_{n+s},\boldsymbol{y}(x_{n+s}))(\boldsymbol{u}_{n+s}-\boldsymbol{y}(x_{n+s}))+O(h^{2k})\right]\\&=\sum_{s=0}^{k-1}\beta_s\boldsymbol{y}'(x_{n+s})-\sum_{s=1}^{k}\beta_s^{*}\boldsymbol{y}'(x_{n+s})\\&\qquad+\sum_{s=0}^{k-1}\beta_s\boldsymbol{f}_{\boldsymbol{y}}(x_{n+s},\boldsymbol{y}(x_{n+s}))(\boldsymbol{u}_{n+s}-\boldsymbol{y}(x_{n+s}))\\&\qquad-\beta_k^{*}\boldsymbol{f}_{\boldsymbol{y}}(x_{n+k},\boldsymbol{y}(x_{n+k}))(\overset{\circ}{\boldsymbol{u}}_{n+k}-\boldsymbol{y}(x_{n+k}))\\&\qquad-\sum_{s=1}^{k-1}\beta_s^{*}\boldsymbol{f}_{\boldsymbol{y}}(x_{n+s},\boldsymbol{y}(x_{n+s}))(\boldsymbol{u}_{n+s}-\boldsymbol{y}(x_{n+s}))+O(h^{k+1}),\end{aligned} \tag{6.121}$$

since $O(h^{2k})$ is of $O(h^{k+1})$ when $k\geq 1$.

Now from the definition of truncation error, we have for the predictor and corrector formulae

$$\frac{1}{h}\left[\boldsymbol{y}(x_{n+k}) + \sum_{s=0}^{k-1}\alpha_s \boldsymbol{y}(x_{n+s})\right] - \sum_{s=0}^{k-1}\beta_s \boldsymbol{y}'(x_{n+s}) = (\boldsymbol{T}_h)_n,$$

$$\frac{1}{h}\left[\boldsymbol{y}(x_{n+k}) + \sum_{s=1}^{k-1}\alpha_s^* \boldsymbol{y}(x_{n+s})\right] - \sum_{s=1}^{k}\beta_s^* \boldsymbol{y}'(x_{n+s}) = (\boldsymbol{T}_h^*)_n.$$

Upon subtraction, and using (6.117), we find

$$\begin{aligned}\sum_{s=0}^{k-1}\beta_s \boldsymbol{y}'(x_{n+s}) - \sum_{s=1}^{k}\beta_s^* \boldsymbol{y}'(x_{n+s}) &= (\boldsymbol{T}_h^*)_n - (\boldsymbol{T}_h)_n \\ &= (\ell_{k+1}^* - \ell_{k+1})h^k \boldsymbol{y}^{(k+1)}(x_n) + O(h^{k+1}).\end{aligned}$$

It suffices to show that the remaining terms in (6.121) together are $O(h^{k+1})$. This we do by expanding $\boldsymbol{f}_y$ about the point $(x_n, \boldsymbol{y}(x_n))$ and by using, from (6.107) and (6.108),

$$\boldsymbol{u}_{n+s} - \boldsymbol{y}(x_{n+s}) = C_{k,k}^* h^k \boldsymbol{e}(x_n) + O(h^{k+1})$$

as well as

$$\begin{aligned}\overset{\circ}{\boldsymbol{u}}_{n+k} - \boldsymbol{y}(x_{n+k}) &= -\sum_{s=0}^{k-1}\alpha_s(\boldsymbol{u}_{n+s} - \boldsymbol{y}(x_{n+s})) + h\sum_{s=0}^{k-1}\beta_s[\boldsymbol{f}(x_{n+s}, \boldsymbol{u}_{n+s}) \\ &\qquad - \boldsymbol{f}(x_{n+s}, \boldsymbol{y}(x_{n+s}))] - h(\boldsymbol{T}_h)_n \\ &= -\sum_{s=0}^{k-1}\alpha_s[C_{k,k}^* h^k \boldsymbol{e}(x_{n+s}) + O(h^{k+1})] + O(h^{k+1}) \\ &= -\left(\sum_{s=0}^{k-1}\alpha_s\right) C_{k,k}^* h^k \boldsymbol{e}(x_n) + O(h^{k+1}) = C_{k,k}^* h^k \boldsymbol{e}(x_n) + O(h^{k+1}),\end{aligned}$$

where in the last equation we have used that $\sum_{s=0}^{k}\alpha_s = 1 + \sum_{s=0}^{k-1}\alpha_s = 0$. We see that the terms in question add up to

$$h^k C_{k,k}^* \boldsymbol{f}_y(x_n, \boldsymbol{y}(x_n))\boldsymbol{e}(x_n)\left\{\sum_{s=0}^{k-1}\beta_s - \beta_k^* - \sum_{s=1}^{k-1}\beta_s^*\right\} + O(h^{k+1}).$$

Since $k \geq 1$, we have $Lt = L^*t = 0$ for the functionals L and L^* associated with the predictor and corrector formula so that the expression in braces is

$$\sum_{s=0}^{k-1}\beta_s - \sum_{s=1}^{k}\beta_s^* = \sum_{s=0}^{k}s\alpha_s - \sum_{s=1}^{k}s\alpha_s^* = \sum_{s=1}^{k}s(\alpha_s - \alpha_s^*) = 0,$$

again by (6.117). This completes the proof of Theorem 6.3.6. □

The formula (6.120) can also be used to estimate the local truncation error in connection with step control procedures such as those discussed for one-step methods in Chap. 5, Sect. 5.8.3. Changing the grid length in multistep methods, however, is more complicated than in one-step methods, and for this we refer to the specialized literature.

6.4 Analytic Theory of Order and Stability

We now turn our attention to the following problems.

(1) Construct a multistep formula of maximum algebraic degree, given its characteristic polynomial $\alpha(t)$. Normally, the latter is chosen to satisfy the root condition.
(2) Determine the maximum algebraic degree among all k-step methods whose characteristic polynomials $\alpha(t)$ satisfy the root condition.

Once we have solved problem (1), it is in principle straightforward to solve (2). We let $\alpha(t)$ vary over all polynomials of degree k satisfying the root condition and for each α construct the multistep formula of maximum degree. We then simply observe the maximum order so attainable.

To deal with problem (1), it is useful to begin with an analytic characterization of algebraic degree – or order – of a multistep formula.

6.4.1 *Analytic Characterization of Order*

With the k-step method (6.2) we associate two polynomials,

$$\alpha(t) = \sum_{s=0}^{k} \alpha_s t^s, \quad \beta(t) = \sum_{s=0}^{k} \beta_s t^s \quad (\alpha_k = 1), \tag{6.122}$$

the first being the characteristic polynomial already introduced in Sect. 6.3.1, (6.75). We define

$$\delta(\zeta) = \frac{\alpha(\zeta)}{\ln \zeta} - \beta(\zeta), \quad \zeta \in \mathbb{C}, \tag{6.123}$$

which, since $\alpha(1) = 0$, is a function holomorphic in the disk $|\zeta - 1| < 1$.

Theorem 6.4.1. *The multistep method (6.2) has (exact) polynomial degree p if and only if $\delta(\zeta)$ has a zero of (exact) multiplicity p at $\zeta = 1$.*

Proof. In terms of the linear functional

$$Lu = \sum_{s=0}^{k} [\alpha_s u(s) - \beta_s u'(s)], \tag{6.124}$$

method (6.2) has exact polynomial degree p if and only if (cf. Sect. 6.1.3, (6.35), where, without restriction of generality, we may assume scalar functions u)

$$Lu = \frac{1}{p!}\int_0^k \lambda_p(\sigma)u^{(p+1)}(\sigma)\mathrm{d}\sigma, \quad \frac{1}{p!}\int_0^k \lambda_p(\sigma)\mathrm{d}\sigma = \ell_{p+1} \neq 0, \tag{6.125}$$

for every $u \in C^{p+1}[0, k]$. Choose $u(t) = \mathrm{e}^{tz}$, where z is a complex parameter. Then (6.125) implies

$$\sum_{s=0}^{k} [\alpha_s \mathrm{e}^{sz} - \beta_s z \mathrm{e}^{sz}] = \frac{z^{p+1}}{p!}\int_0^k \lambda_p(\sigma)\mathrm{e}^{\sigma z}\mathrm{d}\sigma,$$

that is,

$$\frac{\alpha(\mathrm{e}^z)}{z} - \beta(\mathrm{e}^z) = \frac{z^p}{p!}\int_0^k \lambda_p(\sigma)\mathrm{e}^{\sigma z}\mathrm{d}\sigma. \tag{6.126}$$

Since the coefficient of z^p on the right, when $z = 0$, equals $\ell_{p+1} \neq 0$, the function on the left – an entire function – has a zero of exact multiplicity p at $z = 0$. Thus, exact polynomial degree p of L implies that the function $\delta(\mathrm{e}^z)$ has a 0 of exact multiplicity p at $z = 0$. The converse is also true, since otherwise, (6.125) and (6.126) would hold with a different value of p. The theorem now follows readily by applying the conformal map

$$\zeta = \mathrm{e}^z, \quad z = \ln \zeta, \tag{6.127}$$

and by observing that the multiplicity of a zero remains unchanged under such a map. □

Based on Theorem 6.4.1, the first problem mentioned now allows an easy solution. Suppose we are given the characteristic polynomial $\alpha(t)$ of degree k and we want to find $\beta(t)$ of degree k' ($\leq k$) such that the method (6.2) has maximum order. (Typically, $k' = k-1$ for an explicit method, and $k' = k$ for an implicit one.) We simply expand in (6.123) the first term of $\delta(\zeta)$ in a power series about $\zeta = 1$,

$$\frac{\alpha(\zeta)}{\ln \zeta} = c_0 + c_1(\zeta - 1) + c_2(\zeta - 1)^2 + \cdots, \tag{6.128}$$

and then have no other choice for β than to take

$$\beta(\zeta) = c_0 + c_1(\zeta - 1) + \cdots + c_{k'}(\zeta - 1)^{k'}. \tag{6.129}$$

In this way, $\delta(\zeta)$ has a zero of maximum multiplicity at $\zeta = 1$, given $\alpha(t)$ and the degree k' of β. In fact,

$$\delta(\zeta) = c_{k'+1}(\zeta - 1)^{k'+1} + \cdots,$$

and we get order $p \geq k' + 1$. The order could be larger than $k' + 1$ if by chance $c_{k'+1} = 0$. If p is the exact order so attained, then

$$\delta(\zeta) = c_p(\zeta - 1)^p + \cdots, \quad c_p \neq 0, \quad p \geq k' + 1. \tag{6.130}$$

It is interesting to compare this with (6.126):

$$\begin{aligned}\delta(\zeta) &= \frac{(\ln \zeta)^p}{p!} \int_0^k \lambda_p(\sigma)\zeta^\sigma \mathrm{d}\sigma \\ &= \frac{[\zeta - 1 - \frac{1}{2}(\zeta - 1)^2 + \cdots]^p}{p!} \int_0^k \lambda_p(\sigma) \left[1 + \binom{\sigma}{1}(\zeta - 1) + \cdots \right] \mathrm{d}\sigma \\ &= \left(\frac{1}{p!}\int_0^k \lambda_p(\sigma)\mathrm{d}\sigma\right)(\zeta - 1)^p + \cdots \\ &= \ell_{p+1}(\zeta - 1)^p + \cdots .\end{aligned}$$

Thus,

$$\ell_{p+1} = c_p. \tag{6.131}$$

Similarly, if the method is stable, then

$$C_{k,p} = \frac{\ell_{p+1}}{\sum_{s=0}^k \beta_s} = \frac{\ell_{p+1}}{\beta(1)} = \frac{c_p}{c_0}. \tag{6.132}$$

We see that the local and global error constants can be found directly from expansion (6.128). It must be observed, however, that (6.131) and (6.132) hold only for the k-step methods of *maximum* degree. If the degree p is not maximal, then

$$\ell_{p+1} = d_p \tag{6.133}$$

and

$$C_{k,p} = \frac{d_p}{c_0}, \tag{6.134}$$

where

$$\delta(\zeta) = d_p(\zeta - 1)^p + \cdots, \quad d_p \neq 0. \tag{6.135}$$

It seems appropriate, at this point, to observe that if $\alpha(\zeta)$ and $\beta(\zeta)$ have a common factor $\omega(\zeta)$,

$$\alpha(\zeta) = \omega(\zeta)\alpha_0(\zeta), \quad \beta(\zeta) = \omega(\zeta)\beta_0(\zeta),$$

and $\omega(1) \neq 0$, then

$$\delta(\zeta) = \omega(\zeta)\delta_0(\zeta), \quad \delta_0(\zeta) = \frac{\alpha_0(\zeta)}{\ln \zeta} - \beta_0(\zeta),$$

and $\delta_0(\zeta)$ vanishes at $\zeta = 1$ with the same order as $\delta(\zeta)$. The multistep method $\{\alpha, \beta\}$ and the "reduced" multistep method $\{\alpha_0, \beta_0\}$ therefore have the same order and indeed the same error constants (6.133) and (6.134). On the other hand, $\omega(1) = 0$ would imply $\beta(1) = 0$, and the method $\{\alpha, \beta\}$ would not be stable (cf. Sect. 6.3.4, (6.109)). Since, on top of that, a solution of the difference equation (6.2) corresponding to $\{\alpha_0, \beta_0\}$ is also a solution of (6.2) for $\{\alpha, \beta\}$ (cf. Ex. 8), it would be pointless to consider the method $\{\alpha, \beta\}$. For these reasons it is no restriction to assume that the polynomials $\alpha(\zeta)$, $\beta(\zeta)$ are irreducible.

Example. Construct all stable implicit two-step methods of maximum order.

Here,

$$\alpha(\zeta) = (\zeta - 1)(\zeta - \lambda), \quad -1 \le \lambda < 1,$$

since 1 is always a zero of $\alpha(\zeta)$ and the second zero λ must satisfy the root condition. For the expansion (6.128) we have

$$\frac{\alpha(\zeta)}{\ln \zeta} = \frac{(\zeta-1)^2 + (1-\lambda)(\zeta-1)}{\zeta - 1 - \frac{1}{2}(\zeta-1)^2 + \cdots} = \frac{1 - \lambda + (\zeta - 1)}{1 - \frac{1}{2}(\zeta-1) + \frac{1}{3}(\zeta-1)^2 - \cdots}$$
$$= c_0 + c_1(\zeta - 1) + c_2(\zeta-1)^2 + \cdots .$$

An easy calculation gives

$$c_0 = 1 - \lambda, \;\; c_1 = \frac{1}{2}(3 - \lambda), \;\; c_2 = \frac{1}{12}(5 + \lambda),$$
$$c_3 = -\frac{1}{24}(1 + \lambda), \;\; c_4 = \frac{1}{720}(11 + 19\lambda) .$$

Thus,

$$\beta(\zeta) = c_0 + c_1(\zeta - 1) + c_2(\zeta-1)^2 = \frac{5+\lambda}{12}\,\zeta^2 + \frac{2 - 2\lambda}{3}\,\zeta - \frac{1 + 5\lambda}{12},$$

giving the desired method

$$\boldsymbol{u}_{n+2} - (1+\lambda)\boldsymbol{u}_{n+1} + \lambda \boldsymbol{u}_n = h\left\{\frac{5+\lambda}{12}\,\boldsymbol{f}_{n+2} + \frac{2-2\lambda}{3}\,\boldsymbol{f}_{n+1} - \frac{1+5\lambda}{12}\,\boldsymbol{f}_n\right\}.$$

If $c_3 \ne 0$ (i.e., $\lambda \ne -1$), the order is exactly $p = 3$, and the error constant is

$$C_{2,3} = \frac{c_3}{c_0} = -\frac{1}{24}\,\frac{1+\lambda}{1-\lambda} \quad (\lambda \ne -1).$$

The case $\lambda = -1$ is exceptional, giving exact order $p = 4$ (since $c_4 = -\frac{1}{90} \ne 0$); in fact,

$$\boldsymbol{u}_{n+2} - \boldsymbol{u}_n = \frac{h}{3}\,(\boldsymbol{f}_{n+2} + 4\boldsymbol{f}_{n+1} + \boldsymbol{f}_n) \quad (\lambda = -1)$$

is precisely Simpson's rule. This is an example of an "optimal" method – a stable k-step method of order $k + 2$ for k even (cf. Sect. 6.4.2, Theorem 6.4.2(b)).

Example. Construct a pair of two-step methods, one explicit, the other implicit, both having $\alpha(\zeta) = \zeta^2 - \zeta$ and order $p = 2$, but global error constants that are equal in modulus and opposite in sign.

Let $\beta(\zeta)$ and $\beta^*(\zeta)$ be the β-polynomials for the explicit and implicit formula, respectively, and $C_{2,2}$, $C^*_{2,2}$ the corresponding error constants. We have

$$\frac{\alpha(\zeta)}{\ln \zeta} = \frac{\zeta(\zeta-1)}{\zeta - 1 - \frac{1}{2}(\zeta-1)^2 + \cdots} = 1 + \frac{3}{2}(\zeta-1) + \frac{5}{12}(\zeta-1)^2 + \cdots .$$

Thus,

$$\beta(\zeta) = 1 + \frac{3}{2}(\zeta-1) = \frac{3}{2}\zeta - \frac{1}{2},$$

giving

$$C_{2,2} = \frac{\frac{5}{12}}{1} = \frac{5}{12}.$$

For β^*, we try

$$\beta^*(\zeta) = 1 + \frac{3}{2}(\zeta-1) + b(\zeta-1)^2.$$

As we are not aiming for optimal degree, we must use (6.134) and (6.135) and find

$$C^*_{2,2} = \frac{\frac{5}{12} - b}{1} = \frac{5}{12} - b.$$

Since we want $C^*_{2,2} = -C_{2,2}$, we get

$$\frac{5}{12} - b = -\frac{5}{12}, \quad b = \frac{5}{6},$$

and so,

$$\beta^*(\zeta) = 1 + \frac{3}{2}(\zeta-1) + \frac{5}{6}(\zeta-1)^2 = \frac{5}{6}\zeta^2 - \frac{1}{6}\zeta + \frac{1}{3}.$$

The desired pair of methods, therefore, is

$$\begin{cases} \boldsymbol{u}_{n+2} = \boldsymbol{u}_{n+1} + \dfrac{h}{2}(3\boldsymbol{f}_{n+1} - \boldsymbol{f}_n), \\ \boldsymbol{u}^*_{n+2} = \boldsymbol{u}^*_{n+1} + \dfrac{h}{6}(5\boldsymbol{f}^*_{n+2} - \boldsymbol{f}^*_{n+1} + 2\boldsymbol{f}^*_n). \end{cases} \tag{6.136}$$

The interest in such pairs of formulae is rather evident: if both formulae are used independently (i.e., *not* in a predictor–corrector mode, but the corrector formula being iterated to convergence), then by Theorem 6.3.5, (6.107), we have

$$\begin{aligned} \boldsymbol{u}_n - \boldsymbol{y}(x_n) &= C_{2,2}h^2\boldsymbol{e}(x_n) + O(h^3), \\ \boldsymbol{u}^*_n - \boldsymbol{y}(x_n) &= -C_{2,2}h^2\boldsymbol{e}(x_n) + O(h^3); \end{aligned} \tag{6.137}$$

that is, asymptotically for $h \to 0$, the exact solution is halfway between $\boldsymbol{u}_n$ and $\boldsymbol{u}^*_n$. This generates upper and lower bounds for each solution component and built-in error bounds $\frac{1}{2}|\boldsymbol{u}^*_n - \boldsymbol{u}_n|$ (absolute value taken componentwise). A break-down occurs in the ith component if $e^i(x_n) \approx 0$; if $e^i(x)$ changes sign across x_n, the bounds switch from an upper to a lower one and vice versa.

Naturally, it would not be difficult to generate such "equilibrated" pairs of formulae having orders much larger than 2 (cf. Ex. 9).

Example. Given $\alpha(t)$, construct an explicit k-step method of maximum order in Newton's form (involving backward differences).

Here we want L in the form

$$Lu = \sum_{s=0}^{k} \alpha_s u(s) - \sum_{s=0}^{k-1} \gamma_s \nabla^s u'(k-1).$$

The mapping $\zeta = \mathrm{e}^z$ used previously in (6.127) is no longer appropriate here, since we do not want the coefficients of $\beta(\zeta)$. The principle used in the proof of Theorem 6.4.1, however, remains the same: we want $\frac{1}{z} L_{(t)} \mathrm{e}^{tz}$ to vanish at $z = 0$ with multiplicity as large as possible. Now

$$\frac{1}{z} L_{(t)} \mathrm{e}^{tz} = \frac{1}{z} \sum_{s=0}^{k} \alpha_s \mathrm{e}^{sz} - \sum_{s=0}^{k-1} \gamma_s [\nabla^s_{(t)} \mathrm{e}^{tz}]_{t=k-1}$$

and

$$\nabla_{(t)} \mathrm{e}^{tz} = \mathrm{e}^{tz} - \mathrm{e}^{(t-1)z} = \mathrm{e}^{tz} \left(\frac{\mathrm{e}^z - 1}{\mathrm{e}^z} \right),$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$\nabla^s_{(t)} \mathrm{e}^{tz} = \mathrm{e}^{tz} \left(\frac{\mathrm{e}^z - 1}{\mathrm{e}^z} \right)^s.$$

Therefore,

$$\frac{1}{z} L_{(t)} \mathrm{e}^{tz} = \frac{\alpha(\mathrm{e}^z)}{z} - \mathrm{e}^{(k-1)z} \sum_{s=0}^{k-1} \gamma_s \left(\frac{\mathrm{e}^z - 1}{\mathrm{e}^z} \right)^s. \tag{6.138}$$

This suggests the mapping

$$\zeta = \frac{\mathrm{e}^z - 1}{\mathrm{e}^z}, \quad z = -\ln(1 - \zeta),$$

which maps a neighborhood of $z = 0$ conformally onto a neighborhood of $\zeta = 0$. Thus, (6.138) has a zero at $z = 0$ of maximal multiplicity if and only if

$$\begin{aligned} &\frac{\alpha\left(\dfrac{1}{1-\zeta}\right)}{-\ln(1-\zeta)} - \frac{1}{(1-\zeta)^{k-1}} \sum_{s=0}^{k-1} \gamma_s \zeta^s \\ &= \frac{1}{(1-\zeta)^{k-1}} \left\{ \frac{(1-\zeta)^{k-1} \alpha\left(\dfrac{1}{1-\zeta}\right)}{-\ln(1-\zeta)} - \sum_{s=0}^{k-1} \gamma_s \zeta^s \right\} \end{aligned}$$

has a zero at $\zeta = 0$ of maximal multiplicity. Thus, we have to expand

$$\frac{(1-\zeta)^{k-1}\alpha\left(\dfrac{1}{1-\zeta}\right)}{-\ln(1-\zeta)} = \sum_{s=0}^{\infty} \gamma_{ks}\zeta^s \tag{6.139}$$

and take

$$\gamma_s = \gamma_{ks}, \quad s = 0, 1, \ldots, k-1. \tag{6.140}$$

To illustrate, for Adams–Bashforth methods we have

$$\alpha(t) = t^k - t^{k-1},$$

hence

$$\frac{(1-\zeta)^{k-1}\alpha\left(\dfrac{1}{1-\zeta}\right)}{-\ln(1-\zeta)} = \frac{\zeta}{-(1-\zeta)\ \ln(1-\zeta)} =: \gamma(\zeta),$$

which is the generating function

$$\gamma(\zeta) = \sum_{s=0}^{\infty} \gamma_s\zeta^s$$

obtained earlier in Sect. 6.2.1, (6.54). We now see more clearly why the coefficients γ_s are independent of k.

Example. Given $\alpha(t)$, construct an implicit k-step method of maximum order in Newton's form.

Here,

$$Lu = \sum_{s=0}^{k} [\alpha_s u(s) - \gamma_s^* \nabla^s u'(k)],$$

and a calculation similar to the one in the previous Example yields

$$\gamma_s^* = \gamma_{ks}^*, \quad s = 0, 1, \ldots, k, \tag{6.141}$$

where

$$\frac{(1-\zeta)^{k}\alpha\left(\dfrac{1}{1-\zeta}\right)}{-\ln(1-\zeta)} = \sum_{s=0}^{\infty} \gamma_{ks}^*\zeta^s. \tag{6.142}$$

Again, for Adams–Moulton methods,

$$\frac{(1-\zeta)^{k}\alpha\left(\dfrac{1}{1-\zeta}\right)}{-\ln(1-\zeta)} = \frac{\zeta}{-\ln(1-\zeta)} =: \gamma^*(\zeta),$$

with

$$\gamma^*(\zeta) = \sum_{s=0}^{\infty} \gamma_s^* \zeta^s$$

the generating function found earlier.

Example. Given $\beta(t)$, construct a k-step method of maximum order in Newton's form.

In all previous examples we were given $\alpha(t)$ and could thus choose it to satisfy the root condition. There is no intrinsic reason why we should not start with $\beta(t)$ and determine $\alpha(t)$ so as to maximize the order. It then needs to be checked, of course, whether the $\alpha(t)$ thus found satisfies the root condition. Thus, in the example at hand,

$$Lu = \sum_{s=0}^{k} \gamma_s \nabla^s u(k) - \sum_{s=0}^{k} \beta_s u'(s). \tag{6.143}$$

Following the procedure used previously, we get

$$\begin{aligned} \frac{1}{z} L_{(t)} \mathrm{e}^{tz} &= \frac{\mathrm{e}^{kz}}{z} \sum_{s=0}^{k} \gamma_s \left(\frac{\mathrm{e}^z - 1}{\mathrm{e}^z} \right)^s - \beta(\mathrm{e}^z) \\ &= \frac{1}{-(1-\zeta)^k \, \ln(1-\zeta)} \sum_{s=0}^{k} \gamma_s \zeta^s - \beta\left(\frac{1}{1-\zeta} \right), \end{aligned}$$

and we want this to vanish at $\zeta = 0$ with maximum order. Clearly, $\gamma_0 = 0$, and if

$$-(1-\zeta)^k \, \ln(1-\zeta)\, \beta\left(\frac{1}{1-\zeta} \right) = \sum_{s=1}^{\infty} d_{ks} \zeta^s, \tag{6.144}$$

we must take the remaining coefficients to be

$$\gamma_s = d_{ks}, \quad s = 1, 2, \ldots, k, \tag{6.145}$$

to achieve order $p \geq k$.

A particularly simple example obtains if $\beta(t) = t^k$, in which case

$$-(1-\zeta)^k \, \ln(1-\zeta)\, \beta\left(\frac{1}{1-\zeta} \right) = -\ln(1-\zeta) = \zeta + \frac{1}{2}\zeta^2 + \frac{1}{3}\zeta^3 + \cdots ,$$

so that

$$\gamma_0 = 0, \quad \gamma_s = \frac{1}{s}, \quad s = 1, 2, \ldots, k. \tag{6.146}$$

The method

$$\nabla \boldsymbol{u}_{n+k} + \frac{1}{2} \nabla^2 \boldsymbol{u}_{n+k} + \cdots + \frac{1}{k} \nabla^k \boldsymbol{u}_{n+k} = h \boldsymbol{f}_{n+k} \tag{6.147}$$

of order k so obtained is called the *backward differentiation method* and is of some interest in connection with stiff problems (cf. Sect. 6.5.2). Its characteristic polynomial $\alpha(t)$ is easily obtained from (6.143) by noting that $\nabla^s u(k) = \sum_{r=0}^{s}(-1)^r\binom{s}{r}u(k-r)$. One finds

$$\alpha(t) = \sum_{s=0}^{k}\alpha_s t^s, \quad \beta(t) = t^k, \quad \alpha_k = \sum_{r=1}^{k}\frac{1}{r},$$
$$\alpha_s = (-1)^{k-s}\sum_{r=0}^{s}\binom{k-s+r}{k-s}\frac{1}{k-s+r}, \quad s = 0, 1, \dots, k-1. \tag{6.148}$$

Note here that $\alpha_k \neq 1$, but we can normalize (6.143) by dividing both sides by α_k. It turns out that $\alpha(t)$ in (6.148) satisfies the root condition for $k = 1, 2, \dots, 6$ but not for $k = 7$, in which case one root of α lies outside the unit circle (cf. Ex. 10(a)).

We remark that computer algebra systems such as Maple are very useful to implement series expansions of the type considered in this section; see, e.g., MA 3.

6.4.2 Stable Methods of Maximum Order

We now give an answer to problem (2) stated at the beginning of Sect. 6.4.

Theorem 6.4.2. (a) *If k is odd, then every stable k-step method has order $p \leq k+1$.*

(b) *If k is even, then every stable k-step method has order $p \leq k+2$, the order being $k+2$ if and only if $\alpha(t)$ has all its zeros on the circumference of the unit circle.*

Before we prove this theorem, we make the following remarks.

1. In case (a), we can attain order $p = k+1$ for any given $\alpha(t)$ satisfying the root condition; cf. Sect. 6.4.1, (6.130) with $k' = k$.
2. Since $\alpha(t)$ is a real polynomial, all complex zeros of α occur in conjugate pairs. It follows from part (b) of Theorem 6.4.2 that $p = k+2$ if and only if $\alpha(t)$ has zeros at $t = 1$ and $t = -1$, and all other zeros (if any) are located on $|t| = 1$ in conjugate complex pairs.
3. The maximum order among *all* k-step methods (stable or not) is known to be $p = 2k$. The stability requirement thus reduces this maximum possible order to roughly one-half.

Proof of Theorem 6.4.2. We want to determine $\alpha(t)$, subject to the root condition, such that

$$\delta(\zeta) = \frac{\alpha(\zeta)}{\ln\zeta} - \beta(\zeta) \tag{6.149}$$

has a zero at $\zeta = 1$ of maximum multiplicity (cf. Theorem 6.4.1). We map the unit disk $|\zeta| \leq 1$ conformally onto the left half-plane $\operatorname{Re} z \leq 0$ (which is easier to deal with) by means of

$$\zeta = \frac{1+z}{1-z}, \quad z = \frac{\zeta - 1}{\zeta + 1}. \tag{6.150}$$

This maps the point $\zeta = 1$ to the origin $z = 0$ and preserves multiplicities of zeros at these two points. The function $\delta(\zeta)$ in (6.149) is transformed to

$$d(z) = \left\{ \frac{\alpha\left(\frac{1+z}{1-z}\right)}{\ln \frac{1+z}{1-z}} - \beta\left(\frac{1+z}{1-z}\right) \right\} \left(\frac{1-z}{2}\right)^k,$$

except for the factor $[(1-z)/2]^k$ which, however, does not vanish at $z = 0$. We write

$$d(z) = \frac{a(z)}{\ln \frac{1+z}{1-z}} - b(z), \tag{6.151}$$

where

$$a(z) = \left(\frac{1-z}{2}\right)^k \alpha\left(\frac{1+z}{1-z}\right), \quad b(z) = \left(\frac{1-z}{2}\right)^k \beta\left(\frac{1+z}{1-z}\right) \tag{6.152}$$

are both polynomials of degree $\leq k$.

Our problem thus reduces to the following purely analytical problem: *How many initial terms of the Maclaurin expansion of* $d(z)$ *in* (6.151) *can be made to vanish if* $a(z)$ *is to have all its zeros in* $\operatorname{Re} z \leq 0$*, and those with* $\operatorname{Re} z = 0$ *are to be simple*?

To solve this problem, we need some preliminary facts:

(i) We have $a(1) = 1$ and $a(0) = 0$. This follows trivially from (6.152), since $\alpha(t)$ has leading coefficient 1 and $\alpha(1) = 0$.

(ii) The polynomial $a(z)$ has exact degree k, unless $\zeta = -1$ is a zero of $\alpha(\zeta)$, in which case $a(z)$ has degree $k - \mu$, where μ is the multiplicity of the zero $\zeta = -1$. (If the root condition is satisfied, then, of course, $\mu = 1$.) This also follows straightforwardly from (6.152).

(iii) Let

$$a(z) = a_1 z + a_2 z^2 + \cdots + a_k z^k,$$

where $a_1 \neq 0$ by the root condition. If $a(z)$ has all its zeros in $\operatorname{Re} z \leq 0$, then $a_s \geq 0$ for $s = 1, 2, \ldots, k$. (The converse is *not* true.) This is easily seen if we factor $a(z)$ with respect to its real and complex zeros:

$$a(z) = a_\ell z \prod_\rho (z - r_\rho) \prod_\sigma [z - (x_\sigma + \mathrm{i} y_\sigma)][z - (x_\sigma - \mathrm{i} y_\sigma)], \quad a_\ell \neq 0.$$

Simplifying, we can write

$$a(z) = a_\ell z \prod_\rho (z - r_\rho) \prod_\sigma [z - x_\sigma)^2 + y_\sigma^2].$$

Since by assumption, $r_\rho \leq 0$, $x_\sigma \leq 0$, all nonzero coefficients of $a(z)$ have the sign of a_ℓ, and $a_\ell > 0$ since $a(1) = 1$.

(iv) Let

$$\frac{z}{\ln \dfrac{1+z}{1-z}} = \lambda_0 + \lambda_2 z^2 + \lambda_4 z^4 + \cdots .$$

Then

$$\lambda_0 = \tfrac{1}{2}, \quad \lambda_{2\nu} < 0 \text{ for } \nu = 1, 2, 3, \ldots .$$

The proof of this is deferred to the end of this section.

Suppose now that

$$\frac{z}{\ln \dfrac{1+z}{1-z}} \, \frac{a(z)}{z} = b_0 + b_1 z + b_2 z^2 + \cdots . \tag{6.153}$$

For maximum order p, we must take

$$b(z) = b_0 + b_1 z + \cdots + b_k z^k \tag{6.154}$$

in (6.151). The problem at hand then amounts to determining how many of the coefficients $b_{k+1}, b_{k+2}, \ldots$ in (6.153) can vanish simultaneously if $a(z)$ is restricted to have all its zeros in $\operatorname{Re} z \leq 0$ and only simple zeros on $\operatorname{Re} z = 0$.

Expansion (6.153), in view of (iv), is equivalent to

$$\begin{aligned} &(\lambda_0 + \lambda_2 z^2 + \lambda_4 z^4 + \cdots)(a_1 + a_2 z + a_3 z^2 + \cdots + a_k z^{k-1}) \\ &\quad = b_0 + b_1 z + b_2 z^2 + \cdots . \end{aligned}$$

Comparing coefficients of like power on the right and left, we get

$$\begin{aligned} b_0 &= \lambda_0 a_1, \\ b_1 &= \lambda_0 a_2, \\ \left.\begin{aligned} b_{2\nu} &= \lambda_0 a_{2\nu+1} + \lambda_2 a_{2\nu-1} + \cdots + \lambda_{2\nu} a_1 \\ b_{2\nu+1} &= \lambda_0 a_{2\nu+2} + \lambda_2 a_{2\nu} + \cdots + \lambda_{2\nu} a_2 \end{aligned}\right\} &\quad \nu = 1, 2, 3, \ldots, \end{aligned} \tag{6.155}$$

where for convenience we assume $a_\mu = 0$ if $\mu > k$. We distinguish two cases.

Case 1. k is odd. Then by (6.155),

$$b_{k+1} = \lambda_0 a_{k+2} + \lambda_2 a_k + \lambda_4 a_{k-2} + \cdots + \lambda_{k+1} a_1.$$

Since $a_{k+2} = 0$ by convention, $\lambda_{2\nu} < 0$ by (iv), and $a_s \geq 0$, $a_1 > 0$ by (iii), it follows that $b_{k+1} < 0$. Thus, $p = k + 1$ is the maximum possible order.

Case 2. k is even. Here (6.155) gives

$$b_{k+1} = \lambda_0 a_{k+2} + \lambda_2 a_k + \lambda_4 a_{k-2} + \cdots + \lambda_k a_2.$$

Again, $a_{k+2} = 0$, $\lambda_{2\nu} < 0$, and $a_s \geq 0$, so that $b_{k+1} \leq 0$. We have $b_{k+1} = 0$ if and only if $a_2 = a_4 = \cdots = a_k = 0$, that is,

$$a(-z) = -a(z). \tag{6.156}$$

Since $a_k = 0$, we conclude from (ii) that $\alpha(\zeta)$ has a zero at $\zeta = -1$. A trivial zero is $\zeta = 1$. In view of (6.156), the polynomial $a(z)$ cannot have zeros off the imaginary axis without violating the root condition. Therefore, $a(z)$ has all its zeros on $\operatorname{Re} z = 0$, hence $\alpha(\zeta)$ all its zeros on $|\zeta| = 1$. Conversely, if $\alpha(\zeta)$ has zeros at $\zeta = \pm 1$ and all other zeros on $|\zeta| = 1$, then $a_k = 0$, and

$$a(z) = a_{k-1} z \prod_{\gamma} [(z - \mathrm{i} y_\gamma)(z + \mathrm{i} y_\gamma)] = a_{k-1} z \prod_{\gamma} (z^2 + y_\gamma^2);$$

that is, $a(z)$ is an odd polynomial and, therefore, $b_{k+1} = 0$.

This proves the second half of part (b) of Theorem 6.4.2. To complete the proof, we have to show that $b_{k+2} = 0$ is impossible. This follows again from (6.155), if we note that (for k even)

$$\begin{aligned} b_{k+2} &= \lambda_0 a_{k+3} + \lambda_2 a_{k+1} + \lambda_4 a_{k-1} + \cdots + \lambda_{k+2} a_1 \\ &= \lambda_4 a_{k-1} + \cdots + \lambda_{k+2} a_1 < 0 \end{aligned}$$

for the same reason as in Case 1. □

It remains to prove the crucial property (iv). Let

$$f(z) := \frac{z}{\ln \dfrac{1+z}{1-z}} = \lambda_0 + \lambda_2 z^2 + \lambda_4 z^4 + \cdots. \tag{6.157}$$

The fact that $\lambda_0 = \frac{1}{2}$ follows easily by taking the limit of $f(z)$ as $z \to 0$. By Cauchy's formula,

$$\lambda_{2\nu} = \frac{1}{2\pi \mathrm{i}} \oint_C \frac{f(z)\mathrm{d}z}{z^{2\nu+1}} = \frac{1}{2\pi \mathrm{i}} \oint_C \frac{\mathrm{d}z}{z^{2\nu} \ln \dfrac{1+z}{1-z}}, \quad \nu > 1,$$

where C is a contour encircling the origin in the positive sense of direction anywhere in the complex plane cut along $(-\infty, -1)$ and $(1, \infty)$ (where f in (6.157) is one-valued analytic). To get a negativity result for the $\lambda_{2\nu}$ one would like to push the contour as close to the cuts as possible. This is much easier to do after a change of variables according to

$$u = \frac{1}{z}.$$

Then

$$\lambda_{2\nu} = -\frac{1}{2\pi \mathrm{i}} \oint_{\Gamma} u^{2\nu-2} \left[\ln \frac{u+1}{u-1} \right]^{-1} \mathrm{d}u, \tag{6.158}$$

where the cut in the u-plane now runs from -1 to 1, and Γ is a contour encircling this cut in the negative sense of direction. By letting Γ shrink onto the cut and noting that

$$\ln \frac{u+1}{u-1} \to \ln \frac{x+1}{1-x} - \mathrm{i}\pi \quad \text{as} \quad u \to x + \mathrm{i}0, \quad -1 < x < 1,$$

whereas

$$\ln \frac{u+1}{u-1} \to \ln \frac{x+1}{1-x} + \mathrm{i}\pi \quad \text{as} \quad u \to x - \mathrm{i}0, \quad -1 < x < 1,$$

we find in the limit

$$\begin{aligned} \lambda_{2\nu} &= -\frac{1}{2\pi \mathrm{i}} \left\{ \int_{-1}^{1} x^{2\nu-2} \frac{\mathrm{d}x}{\ln \frac{x+1}{1-x} - \mathrm{i}\pi} + \int_{1}^{-1} x^{2\nu-2} \frac{\mathrm{d}x}{\ln \frac{x+1}{1-x} + \mathrm{i}\pi} \right\} \\ &= -\int_{-1}^{1} \frac{x^{2\nu-2}}{\pi^2 + \ln^2 \frac{x+1}{1-x}} \, \mathrm{d}x < 0, \quad \nu \geq 1, \end{aligned} \tag{6.159}$$

as was to be shown.

We also note from (6.159) that

$$|\lambda_{2\nu}| < \frac{1}{\pi^2} \int_{-1}^{1} x^{2\nu-2} \mathrm{d}x = \frac{2}{\pi^2} \frac{1}{2\nu - 1}, \quad \nu = 1, 2, 3, \ldots,$$

that is,

$$\lambda_{2\nu} = O\left(\frac{1}{\nu}\right) \quad \text{as} \quad \nu \to \infty.$$

We now recall a theorem of Littlewood, which says that if

$$f(z) = \sum_{n=0}^{\infty} \lambda_n z^n$$

is convergent in $|z| < 1$ and satisfies

$$f(x) \to s \quad \text{as} \quad x \uparrow 1, \quad \lambda_n = O\left(\frac{1}{n}\right) \quad \text{as} \quad n \to \infty,$$

then

$$\sum_{n=0}^{\infty} \lambda_n = s.$$

In our case,

$$f(x) = \frac{x}{\ln \dfrac{1+x}{1-x}} \to 0 \text{ as } x \uparrow 1$$

and so

$$\sum_{\nu=0}^{\infty} \lambda_{2\nu} = 0, \text{ that is, } \sum_{\nu=1}^{\infty} \lambda_{2\nu} = -\frac{1}{2}. \tag{6.160}$$

For an application of (6.160), see Ex. 11(b).

6.4.3 Applications

Theorem 6.4.2 and its proof technique have a number of interesting consequences, which we now discuss.

Theorem 6.4.3. *For every stable k-step method of maximum order p ($= k+1$ or $k+2$), we have that*

$$\ell_{p+1} < 0, \quad C_{k,p} = \frac{\ell_{p+1}}{\sum_{s=0}^{k} \beta_s} < 0. \tag{6.161}$$

Proof. We recall from (6.131) and (6.132) that

$$\ell_{p+1} = c_p, \quad C_{k,p} = \frac{c_p}{c_0},$$

where $c_0 = \beta(1)$ and

$$\delta(\zeta) = \frac{\alpha(\zeta)}{\ln \zeta} - \beta(\zeta) = c_p(\zeta - 1)^p + \cdots, \quad c_p \neq 0.$$

With the transformation

$$\zeta = \frac{1+z}{1-z}, \quad \zeta - 1 = \frac{2z}{1-z} = 2z + \cdots$$

used in the proof of Theorem 6.4.2, we get for the function $d(z)$ in (6.151)

$$\left(\frac{1-z}{2}\right)^{-k} d(z) = 2^p c_p z^p + \cdots,$$

or

$$d(z) = 2^{p-k} c_p z^p + \cdots .$$

On the other hand, by (6.153) and (6.154),

$$d(z) = b_p z^p + \cdots .$$

Therefore, $c_p = 2^{k-p} b_p$, and

$$\ell_{p+1} = 2^{k-p} b_p, \quad C_{k,p} = 2^{k-p} \frac{b_p}{\beta(1)}. \tag{6.162}$$

From the proof of Theorem 6.4.2, we know that $b_p < 0$. This proves the first relation in (6.161).

To prove the second, we must show that

$$\beta(1) > 0. \tag{6.163}$$

Since $p \geq 1$, we have

$$\beta(1) = \sum_{s=0}^{k} \beta_s = \sum_{s=0}^{k} s\alpha_s = \alpha'(1).$$

By the root condition, $\alpha'(1) \neq 0$. If we had $\alpha'(1) < 0$, then $\alpha(1) = 0$ in conjunction with $\alpha(t) \sim t^k$ as $t \to \infty$ would imply that $\alpha(t)$ vanishes for some $t > 1$, contradicting the root condition. Thus, $\alpha'(1) > 0$, proving (6.163). □

We note, incidentally, from (6.162), since $\beta(1) = 2^k b(0) = 2^k b_0$ (cf. (6.152) and (6.154)), that

$$C_{k,p} = 2^{-p} \frac{b_p}{b_0}. \tag{6.164}$$

Theorem 6.4.4. (a) *Let $k \geq 3$ be odd, and*

$$c_k = \inf |C_{k,k+1}|, \tag{6.165}$$

where the infimum is taken over all stable k-step methods of (maximum) order $p = k + 1$. If $k \geq 5$, there is no such method for which the infimum in (6.165) is attained. If $k = 3$, all three-step methods of order $p = 4$ with $\alpha(\zeta) = (\zeta - 1)(\zeta + 1)(\zeta - \lambda)$, $-1 < \lambda < 1$, have $|C_{3,4}| = c_3 = \frac{1}{180}$.

(b) *Let $k \geq 2$ be even, and*

$$c_k^* = \inf |C_{k,k+2}|, \tag{6.166}$$

where the infimum is taken over all stable k-step methods of (maximum) order $p = k + 2$. If $k \geq 4$, there is no such method for which the infimum in (6.166) is attained. If $k = 2$, Simpson's rule is the only two-step method of order $p = 4$ with $|C_{2,4}| = c_2^* = \frac{1}{180}$.

Proof. (a) By (6.164) we have

$$C_{k,k+1} = 2^{-(k+1)} \frac{b_{k+1}}{b_0}.$$

From the proof of Theorem 6.4.2 (cf. (6.155)), we have

$$b_0 = \lambda_0 a_1 = \tfrac{1}{2} a_1, \;\; b_{k+1} = \lambda_2 a_k + \lambda_4 a_{k-2} + \cdots + \lambda_{k+1} a_1 < 0.$$

So,

$$|C_{k,k+1}| = \frac{1}{2^k a_1} \left(|\lambda_2| a_k + |\lambda_4| a_{k-2} + \cdots + |\lambda_{k+1}| a_1\right) \geq \frac{|\lambda_{k+1}|}{2^k}. \tag{6.167}$$

We claim that

$$c_k = \inf |C_{k,k+1}| = \frac{|\lambda_{k+1}|}{2^k}.$$

Indeed, take

$$a(z) = z \frac{(z - x_1)(z - x_2) \cdots (z - x_{k-1})}{(1 - x_1)(1 - x_2) \cdots (1 - x_{k-1})},$$

where the x_i are distinct negative numbers. Then $a(1) = 1$ (as must be), and $a(z)$ satisfies the root condition. Now let $x_i \to -\infty$ (all i), then $a(z) \to z$, that is,

$$a_1 \to 1, \;\; a_s \to 0 \;\text{ for }\; s = 2, 3, \ldots, k.$$

Therefore, by (6.167),

$$|C_{k,k+1}| \to \frac{|\lambda_{k+1}|}{2^k}.$$

Now suppose that $|C_{k,k+1}| = c_k$ for some stable method. Then, necessarily,

$$a_k = a_{k-2} = \cdots = a_3 = 0,$$

that is,

$$a(z) = a_1 z + a_2 z^2 + a_4 z^4 + \cdots + a_{k-1} z^{k-1}. \tag{6.168}$$

In particular (cf. Sect. 6.4.2, (ii)), $\zeta = -1$ is a zero of $\alpha(\zeta)$ and $a_{k-1} \neq 0$ by the stability requirement. We distinguish two cases:

Case 1. $k = 3$. Here,

$$a(z) = a_1 z + a_2 z^2 = z(a_1 + a_2 z).$$

By stability, $a(z)$ has a zero anywhere on the negative real axis and a zero at $z = 0$. Transforming back to ζ in the usual way (cf. Sect. 6.4.2, (6.150)), this means that

$$\alpha(\zeta) = (\zeta - 1)(\zeta + 1)(\zeta - \lambda), \quad -1 < \lambda < 1.$$

All these methods (of order $p = 4$) by construction have

$$|C_{3,4}| = c_3 = \frac{|\lambda_4|}{8} = \frac{1}{180}.$$

Case 2. $k \geq 5$. If z_i are the zeros of $a(z)$ in (6.168), then by Vieta's rule,

$$\sum_{i=1}^{k-1} z_i = -\frac{a_{k-2}}{a_{k-1}} = 0;$$

hence

$$\sum_{i=1}^{k-1} \operatorname{Re} z_i = 0.$$

Since by stability, $\operatorname{Re} z_i \leq 0$ for all i, we must have $\operatorname{Re} z_i = 0$ for all i. This means that

$$a(z) = \text{const} \cdot z \prod_j (z^2 + y_j^2)$$

is an odd polynomial, so $a_2 = a_4 = \cdots = a_{k-1} = 0$, contradicting stability ($\zeta = -1$ is a multiple zero).

(b) The proof is similar to the one in case (a), and we leave it as an exercise for the reader (see Ex. 13). □

It is interesting to observe that if in the infimum (6.165) of Theorem 6.4.4 we admit only methods whose characteristic polynomials $\alpha(\zeta)$ have the zero $\zeta_1 = 1$ and all other zeros bounded in absolute value by $\gamma < 1$ (and hence are stable), then it can be shown that the infimum is attained precisely when

$$\alpha(\zeta) = (\zeta - 1)(\zeta + \gamma)^{k-1}, \tag{6.169}$$

that is, all zeros other than $\zeta = 1$ are placed at the point $\zeta = -\gamma$ farthest away from $\zeta = 1$. Moreover, there are explicit expressions for the minimum error constant. For example, if k is odd, then (cf. Ex. 12)

$$\min |C_{k,k+1}| = 2^{-k}\left\{ |\lambda_{k+1}| + \binom{k-1}{2}|\lambda_{k-1}|\omega^2 + \binom{k-1}{4}|\lambda_{k-3}|\omega^4 + \cdots + |\lambda_2|\omega^{k-1} \right\}, \tag{6.170}$$

where

$$\omega = \frac{1-\gamma}{1+\gamma} \tag{6.171}$$

and $\lambda_2, \lambda_4, \ldots$ are the expansion coefficients in (iv) of the proof of Theorem 6.4.2. If $\gamma = 0$, we of course recover the Adams–Moulton formulae.

6.5 Stiff Problems

In Sect. 6.4 we were concerned with multistep methods applied to problems on a finite interval; however, the presence of "stiffness" (i.e., of rapidly decaying solutions) requires consideration of infinite intervals and related stability concepts. These again, as in the case of one-step methods, are developed in connection with a simple model problem exhibiting exponentially decaying solutions. The relevant concepts of stability then describe to what extent multistep methods are able to simulate such solutions, especially those decaying at a rapid rate. It turns out that multistep methods are much more limited in their ability to effectively deal with such solutions than are one-step methods. This is particularly so if one requires A-stability, as defined in the next section, and to a lesser extent if one weakens the stability requirement in a manner described briefly in Sect. 6.5.2.

6.5.1 A-Stability

For simplicity we consider the scalar model problem (cf. Chap. 5, (5.166))

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \lambda y, \quad 0 \le x < \infty, \quad \operatorname{Re}\lambda < 0, \tag{6.172}$$

all of whose solutions decay exponentially at infinity. In particular,

$$y(x) \to 0 \text{ as } x \to \infty \tag{6.173}$$

for every solution of (6.172).

Definition 6.5.1. A multistep method (6.2) is called *A-stable* if, when applied to (6.172), it produces a grid function $\{u_n\}_{n=0}^{\infty}$ satisfying

$$u_n \to 0 \text{ as } n \to \infty, \tag{6.174}$$

regardless of the choice of starting values (6.4). (It is assumed that the method is applied with constant grid length $h > 0$.)

We may assume (cf. Sect. 6.4.1) that the multistep method is irreducible; that is, the polynomials $\alpha(t)$ and $\beta(t)$ defined in (6.122) have no common zeros.

Application of (6.2) to (6.172) yields

$$\sum_{s=0}^{k} \alpha_s u_{n+s} - h\lambda \sum_{s=0}^{k} \beta_s u_{n+s} = 0, \tag{6.175}$$

a constant-coefficient difference equation of order k whose characteristic polynomial is (cf. Sect. 6.3.1)

$$\tilde{\alpha}(t) = \alpha(t) - \tilde{h}\beta(t), \quad \tilde{h} = h\lambda \in \mathbb{C}. \tag{6.176}$$

All solutions of (6.175) will tend to 0 as $n \to \infty$ if the zeros of $\tilde{\alpha}$ are all strictly less than 1 in absolute value. The multistep method, therefore, is A-stable if and only if

$$\{\tilde{\alpha}(\zeta) = 0,\ \operatorname{Re}\lambda < 0\} \text{ implies } |\zeta| < 1.$$

This is the same as saying that

$$\{\tilde{\alpha}(\zeta) = 0,\ |\zeta| \geq 1\} \text{ implies } \operatorname{Re}\lambda \geq 0.$$

But $\tilde{\alpha}(\zeta) = 0$ implies $\beta(\zeta) \neq 0$ (since otherwise $\alpha(\zeta) = \beta(\zeta) = 0$, contrary to the assumed irreducibility of the method). Thus $\tilde{\alpha}(\zeta) = 0$ implies

$$\tilde{h} = h\lambda = \frac{\alpha(\zeta)}{\beta(\zeta)},$$

and A-stability is characterized by the condition

$$\operatorname{Re}\frac{\alpha(\zeta)}{\beta(\zeta)} \geq 0 \text{ if } |\zeta| \geq 1. \tag{6.177}$$

Theorem 6.5.1. *If the multistep method (6.2) is A-stable, then it has order $p = 2$ and error constant $C_{k,p} \leq -\frac{1}{12}$. The trapezoidal rule is the only A-stable method for which $p = 2$ and $C_{k,p} = -\frac{1}{12}$.*

Proof. From (6.135) and (6.134) we have for any k-step method of order p

$$\frac{\alpha(\zeta)}{\ln\zeta} - \beta(\zeta) = c_0 C_{k,p}(\zeta - 1)^p + \cdots,$$

which, after division by $\alpha(\zeta) = \alpha'(1)(\zeta - 1) + \cdots = \beta(1)(\zeta - 1) + \cdots = c_0(\zeta - 1) + \cdots$ gives

$$\frac{1}{\ln\zeta} - \frac{\beta(\zeta)}{\alpha(\zeta)} = C_{k,p}(\zeta - 1)^{p-1} + \cdots. \tag{6.178}$$

For the trapezoidal rule, having $\alpha_T(\zeta) = \zeta - 1$ and $\beta_T(\zeta) = \frac{1}{2}(\zeta + 1)$, one easily finds

$$\frac{1}{\ln\zeta} - \frac{\beta_T(\zeta)}{\alpha_T(\zeta)} = -\frac{1}{12}(\zeta - 1) + \cdots. \tag{6.179}$$

Letting

$$\Delta(\zeta) = \frac{\beta(\zeta)}{\alpha(\zeta)} - \frac{\beta_T(\zeta)}{\alpha_T(\zeta)}, \tag{6.180}$$

one obtains from (6.178) and (6.179) by subtraction

$$\Delta(\zeta) = -\left(c + \frac{1}{12}\right)(\zeta - 1 + \cdots), \tag{6.181}$$

where

$$c = \begin{cases} C_{k,p} & \text{if} \quad p = 2, \\ 0 & \text{if} \quad p > 2. \end{cases} \tag{6.182}$$

Given that our method is A-stable, we have from (6.177) that $\operatorname{Re}[\alpha(\zeta)/\beta(\zeta)] \geq 0$ if $|\zeta| \geq 1$, or equivalently,

$$\operatorname{Re} \frac{\beta(\zeta)}{\alpha(\zeta)} \geq 0 \ \text{ if } \ |\zeta| \geq 1.$$

On the other hand,

$$\operatorname{Re} \frac{\beta_T(\zeta)}{\alpha_T(\zeta)} = 0 \ \text{ if } \ |\zeta| = 1.$$

It follows from (6.180) that $\operatorname{Re} \Delta(\zeta) \geq 0$ on $|\zeta| = 1$, and by the maximum principle applied to the real part of $\Delta(\zeta)$, since $\Delta(\zeta)$ is analytic in $|\zeta| > 1$ (there are no zeros of $\alpha(\zeta)$ outside the unit circle), that

$$\operatorname{Re} \Delta(\zeta) \geq 0 \ \text{ for } \ |\zeta| > 1. \tag{6.183}$$

Now putting $\zeta = 1 + \varepsilon$, $\operatorname{Re} \varepsilon > 0$, we have that $|\zeta| > 1$, and therefore, by (6.181) and (6.183), for $|\varepsilon|$ sufficiently small, that

$$c + \tfrac{1}{12} \leq 0.$$

If $p > 2$, this is clearly impossible in view of (6.182), whereas for $p = 2$ we must have $C_{k,p} \leq -\frac{1}{12}$. This proves the first part of Theorem 6.5.1. To prove the second part, we note that $p = 2$ and $C_{k,p} = -\frac{1}{12}$ imply $\Delta(\zeta) = O((\zeta - 1)^2)$, and taking $\zeta = 1 + \varepsilon$ as previously, the real part of $(\zeta - 1)^2$, and in fact of any power $(\zeta - 1)^q$, $q \geq 2$, can take on either sign if $\operatorname{Re} \varepsilon > 0$. Consequently, by (6.183), $\Delta(\zeta) \equiv 0$ and, therefore, the method being irreducible, it follows that $\alpha(\zeta) = \alpha_T(\zeta)$ and $\beta(\zeta) = \beta_T(\zeta)$. □

6.5.2 A(α)-Stability

According to Theorem 6.5.1, asking for A-stability puts multistep methods into a straitjacket. One can loosen it by weakening the demands on the *region of absolute stability*, that is, the region

$$\mathcal{D}_A = \{\tilde{h} \in \mathbb{C} : \ \tilde{\alpha}(\zeta) = 0 \ \text{ implies } \ |\zeta| < 1\}. \tag{6.184}$$

A-stability requires the left half-plane $\operatorname{Re} \tilde{h} < 0$ to be contained in $\mathcal{D}_A$. In many applications, however, it is sufficient that only part of the left half-plane be contained in $\mathcal{D}_A$, for example, the wedge-like region

$$W_\alpha = \{\tilde{h} \in \mathbb{C} : \ |\arg(-\tilde{h})| < \alpha, \ \ \tilde{h} \neq 0\}, \ \ 0 < \alpha < \frac{1}{2}\pi. \tag{6.185}$$

This gives rise to the following definition.

Table 6.1 $A(\alpha)$-stable kth-order backward differentiation methods

k	1	2	3	4	5	6
α	90°	90°	86.03°	73.35°	51.84°	17.84°

Definition 6.5.2. The multistep method (6.2) is said to be $A(\alpha)$-*stable*, $0 < \alpha < \frac{1}{2}\pi$, if $W_\alpha \subset \mathcal{D}_A$.

There are now multistep methods that have order $p > 2$ and are $A(\alpha)$-stable for suitable α, the best known being the kth-order backward differentiation method (6.147). This one is known (cf. Ex. 10) to be stable in the sense of Sect. 6.3.2 for $1 \le k \le 6$ (but not for $k > 6$) and for these values of k turn out to be also $A(\alpha)$-stable with angles α as shown in Table 6.1. They can therefore be effectively employed for problems whose stiffness is such that the responsible eigenvalues are relatively close to the negative real axis.

In principle, for any given $\alpha < \frac{1}{2}\pi$, there exist $A(\alpha)$-stable k-step methods of order k for every $k = 1, 2, 3, \dots$, but their error constants may be so large as to make them practically useless.

6.6 Notes to Chapter 6

Most texts mentioned in the Notes to Chap. 5 also contain discussions of multistep methods. A book specifically devoted to the convergence and stability theory of multistep methods is Henrici [1977]. It also applies the statistical theory of roundoff errors to such methods, something that is rarely found elsewhere in this context. A detailed treatment of variable-step/variable-order Adams-type methods is given in the book by Shampine and Gordon [1975].

Section 6.1.2. The choice of exponential sums as gauge functions in Ω_p is made in Brock and Murray [1952]; trigonometric polynomials are considered in Quade [1951] and Gautschi [1961], and products, respectively sums, of ordinary and trigonometric polynomials in Stiefel and Bettis [1969] and Bettis [1969/1970] (also see Stiefel and Scheifele [1971], Chap. 7, Sect. 24].

Section 6.2.1. Adams described his method in a chapter of the book Bashforth and Adams [1883] on capillary action. He not only derived both the explicit and implicit formulae (6.52) and (6.58) but also proposed a scheme of prediction and correction. He did not predict $\overset{\circ}{\boldsymbol{u}}_{n+k}$ by the first formula in (6.64) but rather by the backtracking scheme described at the end of Sect. 6.2.2. He then solved the implicit equation by what amounts to Newton's method, the preferred method nowadays of dealing with (mildly) stiff problems.

Section 6.2.2. Moulton describes his predictor–corrector method in Moulton [1926]. He predicts exactly as Adams did, and then iterates on the corrector formula as described in the text, preferably only once, by taking the step sufficiently small. No reference is made to Adams, but then, according to the preface, the history of the subject is not one of the concerns of the book.

Section 6.2.3. Milne's suggestion of estimating the truncation error in terms of the difference of the corrected and predicted value is made in Milne [1926].

The first asymptotic formula in (6.72) is due to Spital and Trench, the second to Steffensen; for literature and an elementary derivation, see Gautschi [1976b].

Section 6.3.1. A basic and elegant text on linear difference equations is Miller [1968]. Scalar difference equations such as those encountered in this paragraph are viewed there as special cases of first-order systems of difference equations. A more extensive account of difference equations is Agarwal [2000]. Readers who wish to learn about practical aspects of difference equations, including applications to numerical analysis and the sciences, are referred to Wimp [1984], Goldberg [1986], Lakshmikantham and Trigiante [2002], and Kelley and Peterson [2001].

Section 6.3.2. As in the case of one-step methods, the stability concept of Definition 6.3.1, taken from Keller [1992, Sect. 1.3], is also referred to as zero stability (i.e., for $h \to 0$), to distinguish it from other stability concepts used in connection with stiff problems (cf. Sect. 6.5). The phenomenon of (strong) instability (i.e., lack of zero stability) was first noted by Todd [1950]. Simple roots $t_s \neq 1$ of the characteristic equation (6.78) with $|t_s| = 1$, although not violating the root condition, may give rise to "weak" stability (cf. Henrici [1962, p. 242]). This was first observed by Rutishauser [1952] in connection with Milne's method – the implicit fourth-order method with $\alpha(t) = t^2 - 1$ (Simpson's rule) – although Dahlquist was aware of it independently; see the interesting historical account of Dahlquist [1985] on the evolution of concepts of numerical (in)stability.

Theorem 6.3.3 is due to Dahlquist [1956]. The proof given here follows, with minor deviations, the presentation in Hull and Luxemburg [1960].

Section 6.3.3. There is a stability and convergence theory also for linear multistep methods on nonuniform grids. The coefficients $\alpha_s = \alpha_{s,n}$, $\beta_s = \beta_{s,n}$ in the k-step method (6.2) then depend on the grid size ratios $\omega_i = h_i/h_{i-1}$, $i = n+k-1, n+k-2, \dots, n+1$. If the coefficients $\alpha_{s,n}$, $\beta_{s,n}$ are uniformly bounded, the basic stability and convergence results of Sects. 6.3.2 and 6.3.3 carry over to grids that are quasiuniform in the sense of having ratios h_n/h_{n-1} uniformly bounded and bounded away from zero; see, for example, Hairer et al. [1993, Chap. 3, Sect. 5].

Section 6.3.4. Theorem 6.3.5 is due to Salihov [1962] and, independently, with a refined form of the $O(h^{p+1})$ term in (6.107), to Henrici [1962, Theorem 5.12], [1977, Theorem 4.2].

Result (6.107) may be interpreted as providing the first term in an asymptotic expansion of the error in powers of h. The existence of a full expansion has been investigated by Gragg [1964]. In Gragg [1965], a modified midpoint rule is defined which has an expansion in even powers of h and serves as a basis of Gragg's extrapolation method.

Section 6.3.5. The global validity of the Milne estimator, under the assumptions stated, is proved by Henrici [1962, Theorem 5.13]. The idea of Theorem 6.3.6(5) is also suggested there, but not carried out.

Practical codes based on Adams predictor–corrector type methods use variable steps and variable order in an attempt to obtain the solution to a prescribed accuracy with a minimum amount of computational work. In addition to sound strategies of when and how to change the step and the order of the method, this requires either interpolation to provide the necessary values of f relative to the new step, or extensions of Adams multistep formulae to nonuniform grids. The former was already suggested by both Adams and Moulton. It is the latter approach that is usually adopted nowadays. A detailed discussion of the many technical and practical issues involved in it can be found in Chaps. 5–7, and well-documented Fortran programs in Chap. 10, of Shampine and Gordon [1975]. Such codes are "self-starting" in the sense that they start off with Euler's method and a very small step, and from then on let the control mechanisms take over to arrive at a proper order and proper step size. A more recent version of Shampine and Gordon's code is described in Hairer et al. [1993, Chap. 3, Sect. 7]. So are two other codes, available on Netlib, both based on Nordsieck's formulation (cf. Notes to Chap. 5, Sect. 5.4) of the Adams method, one originally due to Brown et al. [1989] and the other to Gear [1971c].

Section 6.4.1. Pairs of multistep methods, such as those considered in the second Example of this section, having the same global error constants except for sign, are called polar pairs in Rakitskiĭ [1961, Sect. 2]. They are studied further in Salihov [1962]. The backward differentiation formula (6.147), with $k = 1$ or $k = 2$, was proposed by Curtiss and Hirschfelder [1952] to integrate stiff equations. The method with values of k up to 6, for which it is stiffly stable (Gear [1969]), is implemented as part of the code DIFSUB in Gear [1971b,c] and subsequently in the "Livermore solver" LSODE in Hindmarsh [1980]. For a variable-step version of the backward differentiation formulae, see Hairer et al. [1993, p. 400].

Section 6.4.2. Theorem 6.4.2 is a celebrated result of Dahlquist proved in Dahlquist [1956] and extended to higher-order systems of differential equations in his thesis, Dahlquist [1959]. The k-step method of maximum order $p = 2k$ is derived in Dahlquist [1956, Sect. 2.4]. The proof of (iv) based on Cauchy's formula is from Dahlquist [1956, p. 51]. A more algebraic proof is given in Henrici [1962, p. 233]. For the theorem of Littlewood, see Titchmarsh [1939, Sect. 7.66].

Section 6.4.3. The result of Theorem 6.4.4(a) was announced in Gautschi [1963]; case (b) follows from Problem 37 in Henrici [1962, p. 286]. For the remark at the end of this paragraph, see Gautschi and Montrone [1980].

Section 6.5. The standard text on multistep methods for stiff problems is again Hairer and Wanner [2010, Chap. 5]. It contains, in Chap. 5, Sect. 5, references to, and numerical experiments with, a number of multistep codes.

Section 6.5.1. Theorem 6.5.1 is due to Dahlquist [1963]. The proof follows Hairer and Wanner [2010, Chap. 5, Sect. 1]. There are basically two ways to get around the severe limitations imposed by Theorem 6.5.1. One is to weaken the stability requirement, the other to strengthen the method. An example of the former is A(α)-stability defined in Sect. 6.5.2; various possibilities for the latter are discussed in Hairer and Wanner [2010, Chap. 5, Sect. 3]. Order star theory [*ibid.,* Sect. 4], this time on Riemann surfaces, is again an indispensable tool for studying the attainable order, subject to A-stability.

Section 6.5.2. The concept of A(α)-stability was introduced by Widlund [1967]. For Table 6.1 as well as for the remark in the final paragraph, see Hairer and Wanner [2010, Chap. 5, Sect. 2].

The regions $\mathcal{D}_A$ of absolute stability for explicit and implicit k-step Adams methods, as well as for the respective predictor–corrector methods, are depicted for $k = 1, 2, \dots, 6$ in Hairer and Wanner [2010, Chap. 5, Sect. 1]. As one would expect, they become rapidly small, more so for the explicit method than for the others. More favorable are the stability domains for the backward differentiation methods, which are also shown in the cited reference and nicely illustrate the results of Table 6.1.

As in the case of one-step methods, there is a theory of nonlinear stability and convergence also for multistep methods and their "one-legged" companions $\sum_{s=0}^{k} \alpha_s \boldsymbol{u}_{n+s} = h\boldsymbol{f}\left(\sum_{s=0}^{k} \beta_s x_{n+s}, \sum_{s=0}^{k} \beta_s \boldsymbol{u}_{n+s}\right)$. The theory is extremely rich in technical results, involving yet another concept of stability, G-stability, for which we refer again to Hairer and Wanner [2010, Chap. 5, Sects. 6–9].

Exercises and Machine Assignments to Chapter 6

Exercises

1. Describe how Newton's method is applied to solve the system of nonlinear equations

$$\boldsymbol{u}_{n+k} = h\beta_k \boldsymbol{f}(x_{n+k}, \boldsymbol{u}_{n+k}) + \boldsymbol{g}_n, \quad \boldsymbol{g}_n = h\sum_{s=0}^{k-1} \beta_s \boldsymbol{f}_{n+s} - \sum_{s=0}^{k-1} \alpha_s \boldsymbol{u}_{n+s}$$

for the next approximation, $\boldsymbol{u}_{n+k}$.

2. The system of nonlinear equations

$$\boldsymbol{u}_{n+k} = h\beta_k \boldsymbol{f}(x_{n+k}, \boldsymbol{u}_{n+k}) + \boldsymbol{g}_n, \quad \beta_k \neq 0,$$

arising in each step of an implicit multistep method (cf. (6.5)) may be solved by

- Newton's method;
- the modified Newton method (with the Jacobian held fixed at its value at the initial approximation);
- the method of successive approximations (fixed point iteration).

Assume that $\boldsymbol{f}$ has continuous second partial derivatives with respect to the $\boldsymbol{u}$-variables and the initial approximation $\boldsymbol{u}_{n+k}^{[0]}$ satisfies $\boldsymbol{u}_{n+k}^{[0]} = \boldsymbol{u}_{n+k} + O(h^g)$ for some $g > 0$.

(a) Show that the νth iterate $\boldsymbol{u}_{n+k}^{[\nu]}$ in Newton's method has the property that $\boldsymbol{u}_{n+k}^{[\nu]} - \boldsymbol{u}_{n+k} = O(h^{r_\nu})$, where $r_\nu = 2^\nu(g+1) - 1$. Derive analogous statements for the other two methods.
(b) Show that $g = 1$, if one takes $\boldsymbol{u}_{n+k}^{[0]} = \boldsymbol{u}_{n+k-1}$.
(c) Suppose one adopts the following stopping criterion: quit the iteration after μ iterations, where μ is the smallest integer ν such that $r_\nu > p + 1$, where p is the order of the method. For each of the three iterations, determine μ for $p = 2, 3, \ldots, 10$. (Assume $g = 1$ for simplicity.)
(d) If $g = p$, what would μ be in (c)?

3. (a) Consider an explicit multistep method of the form

$$\boldsymbol{u}_{n+2} - \boldsymbol{u}_{n-2} + \alpha(\boldsymbol{u}_{n+1} - \boldsymbol{u}_{n-1}) = h[\beta(\boldsymbol{f}_{n+1} + \boldsymbol{f}_{n-1}) + \gamma \boldsymbol{f}_n].$$

Show that the parameters α, β, γ can be chosen uniquely so that the method has order $p = 6$. {*Hint*: to preserve symmetry, and thus algebraic simplicity, define the associated linear functional on the interval $[-2, 2]$ rather than $[0, 4]$ as in Sect. 6.1.2. Why is this permissible?}
(b) Discuss the stability properties of the method obtained in (a).

4. For the local error constants γ_k, γ_k^* of, respectively, the Adams–Bashforth and Adams–Moulton method, prove that

$$|\gamma_k^*| < \frac{1}{k-1}\gamma_k \quad \text{for } k \geq 2.$$

5. For the local error constants γ_k, γ_k^* of, respectively, the Adams–Bashforth and Adams–Moulton method, show that, as $k \to \infty$,

$$\gamma_k = \frac{1}{\ln k}\left[1 + O\left(\frac{1}{\ln k}\right)\right], \quad \gamma_k^* = -\frac{1}{k \ln^2 k}\left[1 + O\left(\frac{1}{\ln k}\right)\right].$$

{*Hint*: express the constants in terms of the gamma function, use

$$\frac{\Gamma(k+t)}{\Gamma(k+1)} = k^{t-1}\left[1 + O\left(\frac{1}{k}\right)\right], \; k \to \infty,$$

and integrate by parts.}

6. Consider the predictor–corrector method using the Adams–Bashforth formula as predictor and the Adams–Moulton formula (once) as corrector, both in difference form:

$$\begin{aligned}\overset{\circ}{\boldsymbol{u}}_{n+k} &= \boldsymbol{u}_{n+k-1} + h\sum_{s=0}^{k-1}\gamma_s\nabla^s\boldsymbol{f}_{n+k-1},\\ \boldsymbol{u}_{n+k} &= \boldsymbol{u}_{n+k-1} + h\sum_{s=0}^{k-1}\gamma_s^*\nabla^s\overset{\circ}{\boldsymbol{f}}_{n+k},\\ \boldsymbol{f}_{n+k} &= \boldsymbol{f}(x_{n+k},\boldsymbol{u}_{n+k}),\end{aligned}$$

where $\overset{\circ}{\boldsymbol{f}}_{n+k} = \boldsymbol{f}(x_{n+k},\overset{\circ}{\boldsymbol{u}}_{n+k})$, $\nabla\overset{\circ}{\boldsymbol{f}}_{n+k} = \overset{\circ}{\boldsymbol{f}}_{n+k} - \boldsymbol{f}_{n+k-1}$, and so on.

(a) Show that

$$\boldsymbol{u}_{n+k} = \overset{\circ}{\boldsymbol{u}}_{n+k} + h\gamma_{k-1}\nabla^k\overset{\circ}{\boldsymbol{f}}_{n+k}.$$

{*Hint*: first show that $\gamma_s^* = \gamma_s - \gamma_{s-1}$ for $s = 0, 1, 2, \dots$, where γ_{-1} is defined to be zero.}

(b) Show that

$$\nabla^k\overset{\circ}{\boldsymbol{f}}_{n+k} = \overset{\circ}{\boldsymbol{f}}_{n+k} - \sum_{s=0}^{k-1}\nabla^s\boldsymbol{f}_{n+k-1}.$$

{*Hint*: use the binomial identity $\sum_{\sigma=0}^{m}\binom{\sigma+j}{\sigma} = \binom{m+j+1}{j+1}$.}

7. Prove that the predictor–corrector method

$$\begin{cases}\overset{\circ}{\boldsymbol{u}}_{n+k} = -\sum_{s=0}^{k-1}\alpha_s\boldsymbol{u}_{n+s} + h\sum_{s=0}^{k-1}\beta_s\boldsymbol{f}_{n+s},\\ \boldsymbol{u}_{n+k} = -\sum_{s=1}^{k-1}\alpha_s^*\boldsymbol{u}_{n+s} + h\left\{\beta_k^*\boldsymbol{f}(x_{n+k},\overset{\circ}{\boldsymbol{u}}_{n+k}) + \sum_{s=1}^{k-1}\beta_s^*\boldsymbol{f}_{n+s}\right\},\\ \boldsymbol{f}_{n+k} = \boldsymbol{f}(x_{n+k},\boldsymbol{u}_{n+k})\end{cases}$$

is stable for every $\boldsymbol{f} \in \mathcal{F}$ (cf. (6.87), (6.88)), if and only if its characteristic polynomial $\alpha^*(t) = \sum_{s=0}^{k}\alpha_s^*t^s$, $\alpha_0^* = 0$, satisfies the root condition.

8. Let $\alpha(\zeta) = \omega(\zeta)\alpha_0(\zeta)$, $\beta(\zeta) = \omega(\zeta)\beta_0(\zeta)$, and suppose $\{\boldsymbol{u}_n\}$ is a solution of the difference equation (6.2) corresponding to $\{\alpha_0, \beta_0\}$. Show that $\{\boldsymbol{u}_n\}$ also satisfies the difference equation (6.2) corresponding to $\{\alpha, \beta\}$.

9. Construct a pair of four-step methods, one explicit, the other implicit, both having $\alpha(\zeta) = \zeta^4 - \zeta^3$ and order $p = 4$, but global error constants that are equal in modulus and opposite in sign.

10. (a) Compute the zeros of the characteristic polynomial $\alpha(t)$ of the k-step backward differentiation method (6.147) for $k = 1(1)7$ and the modulus of the absolutely largest zero other than 1. Hence, confirm the statement made at the end of Sect. 6.4.1.

    (b) Compare the error constant of the k-step backward differentiation method with that of the k-step Adams–Moulton method for $k = 1(1)7$.

11. (a) Show that the polynomial $b(z)$ in (6.152), for an explicit k-step method, must satisfy $b(1) = 0$.

(b) Use the proof techniques of Theorem 6.4.2 to show that every stable explicit k-step method has order $p \le k$. {*Hint*: make use of (6.160).}

12. Determine min $|C_{k,k+1}|$, where the minimum is taken over all k-step methods of order $k+1$ whose characteristic polynomials have all their zeros ζ_i (except $\zeta_1 = 1$) in the disk $\Gamma_\gamma = \{z \in \mathbb{C} : \ |z| \le \gamma\}$, where γ is a prescribed number with $0 \le \gamma < 1$. {*Hint*: use the theory developed in Sects. 6.4.2 and 6.4.3.}
13. Prove Theorem 6.4.4(b).

Machine Assignments

1. This assignment pertains to an initial value problem for a scalar first-order differential equation.

   (a) A kth-order Adams–Bashforth predictor step amounts to adding h times a linear combination $\sum_{s=0}^{k-1} \gamma_s \nabla^s f_{n+k-1}$ of k backward differences to the last approximation $u_{\text{last}} = u_{n+k-1}$. Write a Matlab routine `AB.m` implementing this step for $k = 1{:}10$. Use Maple to generate the required coefficients γ_s. Take as input variables the number u_{last}, the k-vector $\boldsymbol{F} = [f_n, f_{n+1}, \ldots, f_{n+k-1}]$ of k successive function values, k, and h.

   (b) Do the same as in (a) for the kth-order Adams–Moulton corrector step (in Newton's form), writing a routine `AM.m` whose input is $u_{\text{last}} = u_{n+k-1}$, the vector $\boldsymbol{F0} = [f_{n+1}, f_{n+2}, \ldots, f_{n+k-1}, \overset{\circ}{f}_{n+k}]$, k, and h.

   (c) Use the routines in (a) and (b) to write a routine `PECE.m` implementing the PECE predictor/corrector scheme (6.64) based on the pair of Adams predictor and corrector formulae:

$$\begin{aligned}
\text{P}: &\quad \overset{\circ}{u}_{n+k} = u_{n+k-1} + h \sum_{s=0}^{k-1} \gamma_s \nabla^s f_{n+k-1},\\
\text{E}: &\quad \overset{\circ}{f} = f(x_{n+k}, \overset{\circ}{u}_{n+k}),\\
\text{C}: &\quad u_{n+k} = u_{n+k-1} + h \sum_{s=0}^{k-1} \gamma_s^* \nabla^s \overset{\circ}{f}_{n+k},\\
\text{E}: &\quad f_{n+k} = f(x_{n+k}, u_{n+k}),
\end{aligned}$$

   where $\nabla \overset{\circ}{f}_{n+k} = \overset{\circ}{f}_{n+k} - f_{n+k-1}$, $\nabla^2 \overset{\circ}{f}_{n+k} = \nabla(\nabla \overset{\circ}{f}_{n+k}) = \overset{\circ}{f}_{n+k} - 2f_{n+k-1} + f_{n+k-2}$, etc. As input parameters include the function f, the initial and final values of x, the k initial approximations, the order k, the number N of (equally spaced) grid intervals, and the values of $n+k$ at which printout is to occur.

2. (a) Consider the initial value problem

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{1}{1 - \varepsilon \cos y}, \quad y(0) = 0, \quad 0 \le x \le 2\pi, \quad 0 < \varepsilon < 1.$$

Show that the exact solution $y = y(x)$ is the solution of Kepler's equation $y - \varepsilon \sin y - x = 0$ (cf. Chap. 4, Ex. 21 and Chap. 5, MA 5(a) and (c)). What is $y(\pi)$? What is $y(2\pi)$?

(b) Use the routine `AB.m` of MA 1(a) to solve the initial value problem by the Adams–Bashforth methods of orders $k = 2, 4, 6$, with $N = 40, 160, 640$ integration steps of length $h = 2\pi/N$. At $x = \frac{1}{2}\pi, \pi, \frac{3}{2}\pi, 2\pi$, i.e., for $n + k = \frac{1}{4}N, \frac{1}{2}N, \frac{3}{4}N, N$, print the approximations u_{n+k} obtained, along with the errors err = $u_{n+k} - y(x)$ and the scaled errors err/h^k. Compute $y(x)$ by applying Newton's method to solve Kepler's equation.

(c) Do the same as in (b) with the `PECE.m` routine of MA 1(c).

In both programs start with "exact" initial values. According to (6.107) and the remarks at the end of Sect. 6.3.4, the scaled errors in the printout should be approximately equal to $C_{k,k}e(x)$ resp. $C^*_{k,k}e(x)$, where $C_{k,k}$, $C^*_{k,k}$ are the global error constants of the kth-order Adams–Bashforth resp. the kth-order Adams predictor/corrector scheme, and $x = \frac{1}{2}\pi, \pi, \frac{3}{2}\pi, 2\pi$. Thus, the errors of the predictor/corrector scheme should be approximately equal to $(C^*_{k,k}/C_{k,k})$ times the errors in the Adams–Bashforth method. Examine to what extent this is confirmed by your numerical results.

3. Use the analytic characterization of order given in Theorem 6.4.1, in conjunction with Maple's series expansion capabilities, to:

   (a) determine the coefficients $\{\beta_{k,s}\}_{s=0}^{k-1}$ of the kth-order Adams–Bashforth method (6.48) for $k = 1 : 10$;

   (b) determine the coefficients $\{\beta^*_{k,s}\}_{s=1}^{k}$ of the kth-order Adams–Moulton method (6.56) for $k = 1 : 10$.

4. (a) Write a Matlab routine for plotting the regions $\mathcal{D}_A$ of absolute stability for the kth-order Adams–Moulton methods, $k = 3 : 10$. {*Hint*: seek the boundaries of the regions $\mathcal{D}_A$ in polar coordinates.} In particular, compute the abscissae of absolute stability on the negative real axis.

   (b) Do the same for the Adams (PECE) predictor/corrector method. Compare the stability properties of this predictor/corrector method with those of the corrector alone.

5. Consider the (slightly modified) model problem

$$\frac{\mathrm{d}y}{\mathrm{d}x} = -\omega[y - a(x)], \quad 0 \le x \le 1; \quad y(0) = y_0,$$

where $\omega > 0$ and (i) $a(x) = x^2$, $y_0 = 0$; (ii) $a(x) = \mathrm{e}^{-x}$, $y_0 = 1$; (iii) $a(x) = \mathrm{e}^{x}$, $y_0 = 1$.

   (a) In each of the cases (i)–(iii), obtain the exact solution $y(x)$.

   (b) In each of the cases (i)–(iii), apply the kth-order Adams predictor/corrector method, for $k = 2 : 5$, using exact starting values and step lengths $h = \frac{1}{20}, \frac{1}{40}, \frac{1}{80}, \frac{1}{160}$. Print the exact values y_n and the errors $u_n - y_n$ for $x_n = 0.25, 0.5, 0.75, 1$. Try $\omega = 1$, $\omega = 10$, and $\omega = 50$. Summarize your results.

(c) Repeat (b), but using the k-step backward differentiation method (6.147) (in Lagrangian form).

6. Consider the nonlinear system

$$\left.\begin{aligned}\frac{\mathrm{d}y_1}{\mathrm{d}x} &= 2y_1(1-y_2), \quad y_1(0)=1,\\ \frac{\mathrm{d}y_2}{\mathrm{d}x} &= -y_2(1-y_1), \quad y_2(0)=3,\end{aligned}\right. \quad 0 \le x \le 10,$$

of interest in population dynamics.

(a) Use Matlab's `ode45` routine to plot the solution $\boldsymbol{y} = [y_1, y_2]^T$ of the system to get an idea of its behavior. Also plot the norm of the Jacobian matrix $\boldsymbol{f_y}$ along the solution curve to check on the stiffness of the system.

(b) Determine a step length h, or the corresponding number N of steps, in the classical Runge–Kutta method that would produce about eight correct decimal digits. {*Hint*: for $N = 10, 20, 40, 80, \ldots$ compute the solution with N steps and $2N$ steps and stop as soon as the two solutions agree to within eight decimal places at all grid points common to both solutions. For the basic Runge–Kutta step, use the routine `RK4` from Chap. 5, MA 1(a).}

(c) Apply $N = 640$ steps of the pair of fourth-order methods constructed in Ex. 9 to obtain asymptotically upper and lower bounds to the solution. Plot suitably scaled errors $\boldsymbol{u}_n - \boldsymbol{y}_n$, $\boldsymbol{u}_n^* - \boldsymbol{y}_n$, $n = 1(1)N$, where $\boldsymbol{y}_n$ is the solution computed in (b) by the Runge–Kutta method. For the required initial approximations, use the classical Runge–Kutta method. Use Newton's method to solve the implicit equation for $\boldsymbol{u}_n^*$.

Selected Solutions to Exercises

2. Write the system of equations more simply as

$$\boldsymbol{u} = \boldsymbol{\varphi}(\boldsymbol{u}), \quad \boldsymbol{\varphi}(\boldsymbol{u}) = h\beta_k \boldsymbol{f}(x_{n+k}, \boldsymbol{u}) + \boldsymbol{g}_n,$$

where $\boldsymbol{g}_n$ is a constant vector not depending on $\boldsymbol{u}$, and denote the solution by $\boldsymbol{u}^*$.

(a) *Newton's method* applied to $\boldsymbol{u} - \boldsymbol{\varphi}(\boldsymbol{u}) = \boldsymbol{0}$ can be written as

$$\boldsymbol{u}^{[\nu+1]} = \boldsymbol{u}^{[\nu]} - \left(\boldsymbol{I} - \boldsymbol{\varphi}_{\boldsymbol{u}}(\boldsymbol{u}^{[\nu]})\right)^{-1}\left(\boldsymbol{u}^{[\nu]} - \boldsymbol{\varphi}(\boldsymbol{u}^{[\nu]})\right), \quad \nu = 0, 1, 2, \ldots,$$

where $\boldsymbol{\varphi}_{\boldsymbol{u}}$ is the Jacobian of $\boldsymbol{\varphi}$. Multiplying through by $\boldsymbol{I} - \boldsymbol{\varphi}_{\boldsymbol{u}}(\boldsymbol{u}^{[\nu]})$ and rearranging gives

$$\boldsymbol{u}^{[\nu+1]} = \boldsymbol{\varphi}\big(\boldsymbol{u}^{[\nu]}\big) + \boldsymbol{\varphi}_{\boldsymbol{u}}\big(\boldsymbol{u}^{[\nu]}\big)\big(\boldsymbol{u}^{[\nu+1]} - \boldsymbol{u}^{[\nu]}\big) \quad \text{(Newton)}.$$

Likewise, for modified Newton,

$$\boldsymbol{u}^{[\nu+1]} = \boldsymbol{\varphi}\big(\boldsymbol{u}^{[\nu]}\big) + \boldsymbol{\varphi}_{\boldsymbol{u}}\big(\boldsymbol{u}^{[0]}\big)\big(\boldsymbol{u}^{[\nu+1]} - \boldsymbol{u}^{[\nu]}\big) \quad \text{(modified Newton)}.$$

Successive approximations, finally, gives simply

$$\boldsymbol{u}^{[\nu+1]} = \boldsymbol{\varphi}(\boldsymbol{u}^{[\nu]}) \quad \text{(successive approximations)}.$$

Now, by Taylor expansion at u^* (cf. Chap. 5, Sect. 5.6.4, (5.58)), we have

$$\boldsymbol{\varphi}(\boldsymbol{u}^{[\nu]}) = \boldsymbol{\varphi}(\boldsymbol{u}^*) + \boldsymbol{\varphi}_{\boldsymbol{u}}(\boldsymbol{u}^*)\boldsymbol{\varepsilon}_\nu + \tfrac{1}{2}\,\boldsymbol{\varepsilon}_\nu^{\mathrm{T}}\boldsymbol{\varphi}_{\boldsymbol{uu}}(\overline{\boldsymbol{u}})\boldsymbol{\varepsilon}_\nu,$$

$$\boldsymbol{\varphi}_{\boldsymbol{u}}(\boldsymbol{u}^{[\nu]}) = \boldsymbol{\varphi}_{\boldsymbol{u}}(\boldsymbol{u}^*) + \sum_k \frac{\partial}{\partial u^k}[\boldsymbol{\varphi}_{\boldsymbol{u}}](\boldsymbol{u}^*)\varepsilon_\nu^k,$$

where

$$\text{(3)} \qquad \boldsymbol{\varepsilon}_\nu = \boldsymbol{u}^{[\nu]} - \boldsymbol{u}^*$$

and ε_ν^k is the kth component of $\boldsymbol{\varepsilon}_\nu$.

Consider first Newton's method. Write it in terms of the $\boldsymbol{\varepsilon}$ as

$$\begin{aligned}\boldsymbol{u}^* + \boldsymbol{\varepsilon}_{\nu+1} = \boldsymbol{\varphi}(\boldsymbol{u}^*) &+ \boldsymbol{\varphi}_{\boldsymbol{u}}(\boldsymbol{u}^*)\boldsymbol{\varepsilon}_\nu + \frac{1}{2}\,\boldsymbol{\varepsilon}_\nu^{\mathrm{T}}\boldsymbol{\varphi}_{\boldsymbol{uu}}(\overline{\boldsymbol{u}})\boldsymbol{\varepsilon}_\nu \\ &+ \left(\boldsymbol{\varphi}_{\boldsymbol{u}}(\boldsymbol{u}^*) + \sum_k \frac{\partial}{\partial u^k}[\boldsymbol{\varphi}_{\boldsymbol{u}}](\boldsymbol{u}^*)\varepsilon_\nu^k\right)(\boldsymbol{\varepsilon}_{\nu+1} - \boldsymbol{\varepsilon}_\nu),\end{aligned}$$

or, simplifying, noting that $\boldsymbol{u}^* = \boldsymbol{\varphi}(u^*)$,

$$\begin{aligned}\left(\boldsymbol{I} - \boldsymbol{\varphi}_{\boldsymbol{u}}(\boldsymbol{u}^*) - \sum_k \frac{\partial}{\partial u^k}[\boldsymbol{\varphi}_{\boldsymbol{u}}](\boldsymbol{u}^*)\varepsilon_\nu^k\right)\boldsymbol{\varepsilon}_{\nu+1} &= \frac{1}{2}\,\boldsymbol{\varepsilon}_\nu^T\boldsymbol{\varphi}_{\boldsymbol{uu}}(\overline{\boldsymbol{u}})\boldsymbol{\varepsilon}_\nu \\ &- \sum_k \frac{\partial}{\partial u^k}[\boldsymbol{\varphi}_{\boldsymbol{u}}](\boldsymbol{u}^*)\varepsilon_\nu^k \cdot \boldsymbol{\varepsilon}_\nu.\end{aligned}$$

Now recall from the definition of $\boldsymbol{\varphi}$, and our smoothness assumptions, that $\boldsymbol{\varphi}_{\boldsymbol{u}}(\boldsymbol{u}) = O(h)$ and also $\boldsymbol{\varphi}_{\boldsymbol{uu}}(\overline{\boldsymbol{u}}) = O(h)$. Suppose $\boldsymbol{\varepsilon}_\nu = O(h^{r_\nu})$. By assumption, $r_0 = g > 0$, and from the relation above we see that the left-hand side is $O(h^{r_{\nu+1}})$, while the right-hand side is $O(h^{2r_\nu+1})$, so that

$$r_{\nu+1} = 2r_\nu + 1.$$

The solution of this difference equation, with starting value $r_0 = g$, is readily seen to be

$$r_\nu = 2^\nu(g+1) - 1 \quad \text{(Newton)}.$$

For the modified Newton's method, one gets similarly

$$\big(I - \boldsymbol{\varphi}_{\boldsymbol{u}}(\boldsymbol{u}^{[0]})\big)\,\boldsymbol{\varepsilon}_{\nu+1} = \frac{1}{2}\,\boldsymbol{\varepsilon}_\nu^T \boldsymbol{\varphi}_{\boldsymbol{uu}}(\overline{\boldsymbol{u}})\boldsymbol{\varepsilon}_\nu - \sum_k \frac{\partial}{\partial u^k}[\boldsymbol{\varphi}_{\boldsymbol{u}}](\boldsymbol{u}^{[0]})\varepsilon_0^k \cdot \boldsymbol{\varepsilon}_\nu,$$

so that

$$r_{\nu+1} = \min\,(2r_\nu + 1, g + r_\nu + 1), \quad r_0 = g.$$

For $\nu = 0$ this gives $r_1 = 2g + 1$. Using induction on ν, suppose $r_\nu = \nu(g+1) + g$ for some $\nu \geq 1$ (true when $\nu = 1$). Then, since

$$\begin{aligned}\min(2r_\nu{+}1, g{+}r_\nu + 1) &= \min((2\nu{+}1)(g+1) + g, (\nu+1)(g+1) + g) \\ &= (\nu+1)(g+1) + g,\end{aligned}$$

we obtain $r_{\nu+1} = (\nu+1)(g+1) + g$, which is the induction hypothesis with ν replaced by $\nu + 1$. Therefore,

$$r_\nu = \nu(g+1) + g \quad \text{(modified Newton)}.$$

Finally, in the case of fixed point iteration, from

$$\boldsymbol{\varepsilon}_{\nu+1} = \boldsymbol{\varphi}_{\boldsymbol{u}}(\boldsymbol{u}^*)\boldsymbol{\varepsilon}_\nu + \frac{1}{2}\boldsymbol{\varepsilon}_\nu^{\mathrm{T}}\boldsymbol{\varphi}_{\boldsymbol{uu}}(\overline{\boldsymbol{u}})\boldsymbol{\varepsilon}_\nu,$$

one gets

$$r_{\nu+1} = r_\nu + 1, \quad r_0 = g,$$

hence

$$r_\nu = \nu + g \quad \text{(successive approximations)}.$$

(b) If $\boldsymbol{u}^{[0]} = \boldsymbol{u}_{n+k-1}$, then, since $\boldsymbol{u}^* = \boldsymbol{u}_{n+k}$,

$$\boldsymbol{\varepsilon}_0 = \boldsymbol{u}^{[0]} - \boldsymbol{u}^* = \boldsymbol{u}_{n+k-1} - \boldsymbol{u}_{n+k}.$$

From the multistep formula

$$\boldsymbol{u}_{n+k} + \sum_{s=0}^{k-1}\alpha_s \boldsymbol{u}_{n+s} = h\sum_{s=0}^{k}\beta_s \boldsymbol{f}_{n+s} = O(h),$$

one obtains

$$\boldsymbol{\varepsilon}_0 = \boldsymbol{u}_{n+k-1} + \sum_{s=0}^{k-1}\alpha_s \boldsymbol{u}_{n+s} + O(h),$$

which, in view of $\boldsymbol{u}_{n+s} = \boldsymbol{u}_n + O(h)$ and $\alpha_k = 1$, yields

$$\boldsymbol{\varepsilon}_0 = \boldsymbol{u}_n \sum_{s=0}^{k}\alpha_s + O(h) = O(h),$$

by consistency. Thus, $g = 1$.

(c) Solve $r_\lambda = p + 1$ and take $\mu = \lceil \lambda \rceil$ if λ is not an integer, and $\mu = \lambda + 1$ otherwise. For Newton's method, this gives (when $g = 1$)

$$\lambda = \frac{\log(p+2)}{\log 2} - 1, \ \ \mu = 2 \text{ for } 2 \le p \le 5, \ \ \mu = 3 \text{ for } 6 \le p < 14.$$

For modified Newton,

$$\lambda = \frac{p}{2}, \ \ \mu = \begin{cases} \dfrac{p+2}{2}, & p \text{ even,} \\ \dfrac{p+1}{2}, & p \text{ odd.} \end{cases}$$

Finally, for successive approximation,

$$\lambda = p, \ \ \mu = p + 1.$$

(d) If $g = p$, then, from the results in (a), for Newton's method,

$$\lambda = \frac{\log\left(1 + \frac{1}{p+1}\right)}{\log 2} < 1, \ \ \mu = 1;$$

for modified Newton,

$$\lambda = \frac{1}{p+1} < 1, \ \ \mu = 1;$$

and for successive approximations,

$$\lambda = 1, \ \ \mu = 2.$$

10. (a) From (6.148), after some calculation or with the help of Maple, one finds the following characteristic polynomials:

$$\begin{aligned}
\alpha(t) &= t - 1 \ \text{ for } k = 1, \\
&= t^2 - \tfrac{4}{3}t + \tfrac{1}{3} \ \text{ for } k = 2, \\
&= t^3 - \tfrac{18}{11}t^2 + \tfrac{9}{11}t - \tfrac{2}{11} \ \text{ for } k = 3, \\
&= t^4 - \tfrac{48}{25}t^3 + \tfrac{36}{25}t^2 - \tfrac{16}{25}t + \tfrac{3}{25} \ \text{ for } k = 4, \\
&= t^5 - \tfrac{300}{137}t^4 + \tfrac{300}{137}t^3 - \tfrac{200}{137}t^2 + \tfrac{75}{137}t - \tfrac{12}{137} \ \text{ for } k = 5, \\
&= t^6 - \tfrac{120}{49}t^5 + \tfrac{150}{49}t^4 - \tfrac{400}{147}t^3 + \tfrac{75}{49}t^2 - \tfrac{24}{49}t + \tfrac{10}{147} \ \text{ for } k = 6, \\
&= t^7 - \tfrac{980}{363}t^6 + \tfrac{490}{121}t^5 - \tfrac{4900}{1089}t^4 + \tfrac{1225}{363}t^3 - \tfrac{196}{121}t^2 + \tfrac{490}{1089}t - \tfrac{20}{363} \ \text{ for } k = 7.
\end{aligned}$$

Matlab gives for the roots:

k	roots ρ_i, $i = 1, 2, \dots, k$	$\max_{i\neq 1} \lvert\rho_i\rvert$
1	1	$---$
2	1, .3333	.3333
3	1, .3182 ± .2839i	.4264
4	1, .2693 ± .4920i, .3815	.5609
5	1, .2100 ± .6769i, .3848 ± .1621i	.7087
6	1, .1453 ± .8511i, .3762 ± .2885i, .4061	.8634
7	1, .0768 ± 1.0193i, .3628 ± .3988i, .4103 ± .1143i	1.0222

It is seen that the root condition is satisfied for $1 \le k \le 6$, but not for $k = 7$.

(b) From the discussion following (6.143), the local error constant ℓ_{k+1} is the coefficient of ζ^k in the expansion of $\frac{1}{\alpha_k} z^{-1} L_{(t)} \mathrm{e}^{tz}$ ($\mathrm{e}^z = (1-\zeta)^{-1}$), hence $\ell_{k+1} = -\frac{1}{\alpha_k} d_{k,k+1}$. The global error constant, therefore, is

$$C_{k,k+1} = -\frac{d_{k,k+1}}{\alpha_k^2}.$$

In the particular case of (6.146), one gets

$$C_{k,k+1} = -\frac{\gamma_{k+1}}{\alpha_k^2} = -\frac{1}{(k+1)\alpha_k^2}, \quad \alpha_k = \sum_{r=1}^{k} \frac{1}{r}.$$

For Adams–Moulton (cf. (6.60)), one has

$$C^*_{k,k+1} = \gamma^*_k,$$

since $\sum_{s=0}^{k} \beta^*_k = 1$ by consistency. Below is a table of α_k, $C_{k,k+1}$, and $C^*_{k,k+1}$ for $k = 1:7$.

k	α_k	$C_{k,k+1}$	$C^*_{k,k+1}$
1	1	$-\frac{1}{2} = -.500$	$-\frac{1}{2} = -.500$
2	$\frac{3}{2}$	$-\frac{4}{27} = -.148$	$-\frac{1}{12} = -.0833$
3	$\frac{11}{6}$	$-\frac{9}{121} = -.0744$	$-\frac{1}{24} = -.0417$
4	$\frac{25}{12}$	$-\frac{144}{3125} = -.0461$	$-\frac{19}{720} = -.0264$
5	$\frac{137}{60}$	$-\frac{600}{18769} = -.0320$	$-\frac{3}{160} = -.0188$
6	$\frac{49}{20}$	$-\frac{400}{16807} = -.0238$	$-\frac{863}{60480} = -.0143$
7	$\frac{363}{140}$	$-\frac{2450}{131769} = -.0186$	$-\frac{275}{24192} = -.0114$

Both error constants are negative for all k, those for Adams–Moulton consistently smaller in absolute value (for $2 \le k \le 7$) than those for backward differentiation, by a factor of about 0.6.

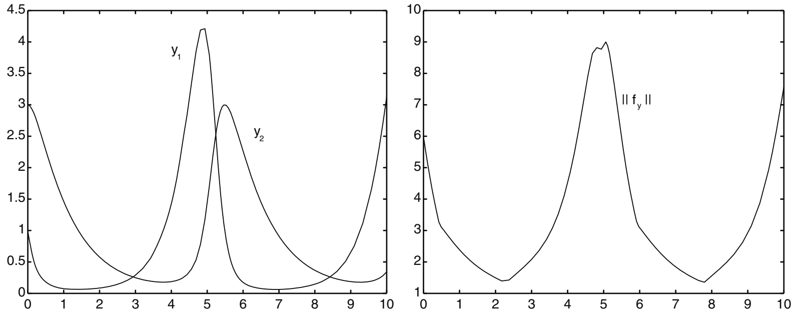
Selected Solutions to Machine Assignments

6. (a)

PROGRAMS

```
%MAVI_6A
%
y0=[1;3]; xspan=[0 10];
[x,y]=ode45(@fMAVI_6,xspan,y0);
figure
plot(x,y)
sy=size(y,1);
normJ=zeros(sy,1);
for i=1:sy
  z=[y(i,1);y(i,2)];
  normJ(i)=norm(JfMAVI_6(x(i),z),inf);
end;
figure
plot(x,normJ)

%FMAVI_6  Differential equation for MAVI_6
%
function yprime=fMAVI_6(x,y)
yprime=[2*y(1)*(1-y(2));-y(2)*(1-y(1))];

%JfMAVI_6   Jacobian matrix of fMAVI_6
%
function J=JfMAVI_6(x,y)
J=[2*(1-y(2)) -2*y(1);y(2) -(1-y(1))];
```

PLOTS



The graph on the right shows that the differential equation is definitely nonstiff.

(b) For the Matlab program pertaining to this part (b), see the beginning of the program shown in part (c).

```
      OUTPUT

   >> MAVI_6BC
      h=0.00390625   N=2560
   >>
```

(c) The pair of fourth-order methods is (cf. Ex. 9)

$$\boldsymbol{u}_{n+4} = \boldsymbol{u}_{n+3} + h \sum_{s=0}^{3} \beta_s \boldsymbol{f}_{n+s},$$
$$\boldsymbol{u}^*_{n+4} = \boldsymbol{u}^*_{n+3} + h \sum_{s=0}^{4} \beta^*_s \boldsymbol{f}^*_{n+s},$$

where $\boldsymbol{f}_{n+s} = \boldsymbol{f}(x_{n+s}, \boldsymbol{u}_{n+s})$, $\boldsymbol{f}^*_{n+s} = \boldsymbol{f}(x_{n+s}, \boldsymbol{u}^*_{n+s})$, and the coefficients β_s, β^*_s are given by

s	$24\beta_s$	$360\beta^*_s$
0	−9	116
1	37	−449
2	−59	621
3	55	−179
4		251

To compute $\boldsymbol{u}^*_{n+4}$ in the implicit method, we write the latter in the form

$$\boldsymbol{F}(\boldsymbol{u}^*_{n+4}) := \boldsymbol{u}^*_{n+4} - h\beta^*_4 \boldsymbol{f}(x_{n+4}, \boldsymbol{u}^*_{n+4}) - \boldsymbol{g} = \boldsymbol{0}, \quad \boldsymbol{g} = \boldsymbol{u}^*_{n+3} + h \sum_{s=0}^{3} \beta^*_s \boldsymbol{f}^*_{n+s}$$

and apply to it Newton's method,

$$\begin{aligned} \boldsymbol{J}_{\boldsymbol{F}}(\boldsymbol{u}^{[i]})\boldsymbol{\Delta}_i &= \boldsymbol{u}^{[i]} - h\beta^*_4 \boldsymbol{f}(x_{n+4}, \boldsymbol{u}^{[i]}) - \boldsymbol{g}, \\ \boldsymbol{u}^{[i+1]} &= \boldsymbol{u}^{[i]} - \boldsymbol{\Delta}_i, \end{aligned}$$

where

$$\boldsymbol{J}_{\boldsymbol{F}}(\boldsymbol{u}) = \boldsymbol{I} - h\beta^*_4 \boldsymbol{f}_{\boldsymbol{y}}(x_{n+4}, \boldsymbol{u}), \quad \boldsymbol{f}_{\boldsymbol{y}}(\boldsymbol{y}) = \begin{bmatrix} 2(1-y_2) & -2y_1 \\ y_2 & -(1-y_1) \end{bmatrix}.$$

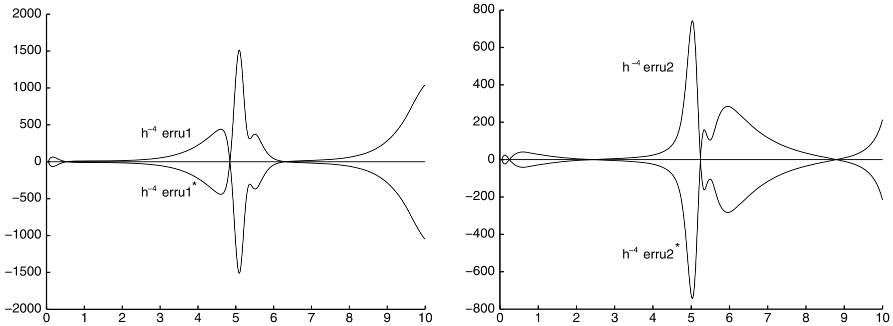
If we take $\boldsymbol{u}^{[0]} = \boldsymbol{u}_{n+4}$–the approximation obtained from the explicit formula–we find that never more than three iterations (mostly 2) are required to iterate to an accuracy of about eight decimal digits.

PROGRAM

```
%MAVI_6BC
%
% Part (b)
%
N=10; Y2=zeros(2,N); maxerr=1;
while maxerr>.5e-8
  N=2*N; h=10/N; Y=Y2;
  err=zeros(1,N/2);
  y1=[1;3];
  for n=1:N
    x=(n-1)*h;
    y=y1;
    y1=RK4(@fMAVI_6,x,y,h);
    Y2(:,n)=y1;
    if floor(n/2)==n/2
      err(n/2)=norm(Y(:,n/2)-Y2(:,n),inf);
    end
  end
  maxerr=max(err);
end
fprintf(' h=%10.8f N=%3.0f\n',2*h,N/2)
%
% Part (c)
%
g=zeros(2,1); err=zeros(2,1);
N=640;
h=10/N; fac=5120/N; U=zeros(2,N); ...
  Ustar=zeros(2,N);
F=zeros(2,N); Fstar=zeros(2,N); u0=[1;3];
U(:,1)=RK4(@fMAVI_6,0,u0,h); F(:,1) ...
  =fMAVI_6(h,U(:,1));
Ustar(:,1)=U(:,1); Fstar(:,1)=F(:,1);
for s=2:3
  U(:,s)=RK4(@fMAVI_6,h,U(:,s-1),h); ...
     F(:,s)=fMAVI_6(s*h,U(:,s));
  Ustar(:,s)=U(:,s); Fstar(:,s)=F(:,s);
end
U(:,4)=U(:,3)+(h/24)*(55*F(:,3)-59 ...
  *F(:,2)+37*F(:,1)-9*fMAVI_6(0,u0));
F(:,4)=fMAVI_6(4*h,U(:,4));
g=Ustar(:,3)+(h/360)*(-179*Fstar(:,3) ...
  +621*Fstar(:,2)-449*Fstar(:,1) ...
  +116*fMAVI_6(0,u0));
```

```
u4new=U(:,4); d=[1;1];
while norm(d,inf)>.5e-8
  u4=u4new;
  J=eye(2)-(251/360)*h*JfMAVI_6(4*h,u4);
  d=J\(u4-(251/360)*h*fMAVI_6(4*h,u4)-g);
  u4new=u4-d;
end
Ustar(:,4)=u4new; Fstar(:,4) ...
  =fMAVI_6(4*h,Ustar(:,4));
for n=5:N
  U(:,n)=U(:,n-1)+(h/24)*(55*F(:,n-1) ...
    -59*F(:,n-2)+37*F(:,n-3)-9*F(:,n-4));
  F(:,n)=fMAVI_6(n*h,U(:,n));
  g=Ustar(:,n-1)+(h/360)*(-179 ...
    *Fstar(:,n-1)+621*Fstar(:,n-2) ...
    -449*Fstar(:,n-3)+116*Fstar(:,n-4));
  unnew=U(:,n); d=[1;1];
  while norm(d,inf)>.5e-8
    un=unnew;
    J=eye(2)-(251/360)*h*JfMAVI_6(n*h,un);
    d=J\(un-(251/360)*h*fMAVI_6(n*h,un)-g);
    unnew=un-d;
  end
  Ustar(:,n)=unnew; Fstar(:,n) ...
    =fMAVI_6(n*h,Ustar(:,n));
end
f1N=fac*(1:N); xp=h*(1:N)';
erru1=h^(-4)*(U(1,:)-Y2(1,f1N))';
erru1star=h^(-4)*(Ustar(1,:)-Y2(1,f1N))';
erru2=h^(-4)*(U(2,:)-Y2(2,f1N))';
erru2star=h^(-4)*(Ustar(2,:)-Y2(2,f1N))';
figure
hold on
plot(xp,erru1)
plot(xp,erru1star)
plot([0 10],[0 0])
text(2.5,400,'h^{-4} erru1', ...
   'FontSize',14)
text(2.5,-400,'h^{-4} erru1^*', ...
   'FontSize',14)
hold off
figure
hold on
plot(xp,erru2)
plot(xp,erru2star)
plot([0 10],[0 0])
text(3.2,500,'h^{-4} erru2', ...
   'FontSize',14)
text(3.2,-500,'h^{-4} erru2^*', ...
   'FontSize',14)
hold off
```

PLOTS (for N=60)



The graphs on the left show the first components $h^{-4}(u_n^1 - y_n^1)$, $h^{-4}(u_n^{*1} - y_n^1)$ of the scaled errors, those on the right the second components $h^{-4}(u_n^2 - y_n^2)$, $h^{-4}(u_n^{*2} - y_n^2)$. They are approximately proportional to the respective global error constants $C_{4,4}$ and $C_{4,4}^*$ of the explicit and implicit method, where $C_{4,4}^* = -C_{4,4}$ by construction. Thus the graphs are essentially symmetric with respect to the real axis. It can also be seen that the error bounds switch their roles from upper to lower bounds, and vice versa, several times during the integration (three times for the first component, four times for the second component). Evidently, these are the places where the first resp. second component of the solution $\boldsymbol{e}(x)$ of the variational differential equation crosses the real axis (cf. Theorem 6.3.5, (6.107)).

Chapter 7
Two-Point Boundary Value Problems for ODEs

Many problems in applied mathematics require solutions of differential equations specified by conditions at more than one point of the independent variable. These are called *boundary value problems*; they are considerably more difficult to deal with than initial value problems, largely because of their global nature. Unlike (local) existence and uniqueness theorems known for initial value problems (cf. Theorem 5.3.1), there are no comparably general theorems for boundary value problems. Neither existence nor uniqueness is, in general, guaranteed.

We concentrate here on *two-point boundary value problems*, in which the system of differential equations

$$\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}x} = \boldsymbol{f}(x, \boldsymbol{y}), \quad \boldsymbol{f} : [a,b] \times \mathbb{R}^d \to \mathbb{R}^d, \;\; d \geq 2, \tag{7.1}$$

is supplemented by conditions at the two endpoints a and b. In the most general case they take the form

$$\boldsymbol{g}(\boldsymbol{y}(a), \boldsymbol{y}(b)) = \boldsymbol{0}, \tag{7.2}$$

where $\boldsymbol{g}$ is a nonlinear mapping $\boldsymbol{g}\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$. Often, however, they are linear and even of the very special kind in which some components of $\boldsymbol{y}$ are prescribed at one endpoint, and some (other or the same) components at the other endpoint, the total number of conditions being equal to the dimension d of the system.

There are other important problems, such as eigenvalue problems and problems with free boundary, that can be transformed to two-point boundary value problems and, therefore, also solved numerically in this manner.

An *eigenvalue problem* is an overdetermined problem containing a parameter λ, say,

$$\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}x} = \boldsymbol{f}(x, \boldsymbol{y}; \lambda), \quad a \leq x \leq b, \tag{7.3}$$

W. Gautschi, *Numerical Analysis*, DOI 10.1007/978-0-8176-8259-0_7,

with $\boldsymbol{f}$ as in (7.1), but depending on an additional scalar parameter λ, and $d+1$ boundary conditions (instead of d) of the form

$$\boldsymbol{g}(\boldsymbol{y}(a), \boldsymbol{y}(b); \lambda) = \boldsymbol{0}, \quad \boldsymbol{g} : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^{d+1}. \tag{7.4}$$

The best-known example of an eigenvalue problem is the *Sturm–Liouville problem*, where one seeks a nontrivial solution u of

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(p(x)\frac{\mathrm{d}u}{\mathrm{d}x}\right) + q(x)u = \lambda u, \quad a \le x \le b, \tag{7.5}$$

subject to homogeneous boundary conditions

$$u(a) = 0, \quad u(b) = 0. \tag{7.6}$$

Since with any solution of (7.5) and (7.6) every constant multiple of it is also a solution, we may specify, in addition to (7.6), that (for example)

$$u'(a) = 1, \tag{7.7}$$

and in this way also make sure that $u \not\equiv 0$. The problem then becomes of the form (7.3), (7.4), if (7.5) is written as a system of two first-order equations (cf. (5.18)–(5.20) of Chap. 5). In this particular case, $\boldsymbol{f}$ in (7.3) is linear homogeneous, and $\boldsymbol{g}$ in (7.4) is also linear and independent of λ. Normally, (7.5) will have no solution satisfying all three boundary conditions (7.6) and (7.7), except for special values of λ; these are called *eigenvalues* of the problem (7.5)–(7.7). Similarly, there will be exceptional values of λ – again called eigenvalues – for which (7.3) and (7.4) admit a solution.

To write (7.3) and (7.4) as a two-point boundary value problem, we introduce an additional component and associated (trivial) differential equation,

$$y^{d+1} = \lambda, \quad \frac{\mathrm{d}y^{d+1}}{\mathrm{d}x} = 0$$

and simply adjoin this to (7.3). That is, we let

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y} \\ y^{d+1} \end{bmatrix} \in \mathbb{R}^{d+1}$$

and write (7.3) and (7.4) in the form

$$\frac{\mathrm{d}\boldsymbol{Y}}{\mathrm{d}x} = \boldsymbol{F}(x, \boldsymbol{Y}), \quad a \le x \le b; \quad \boldsymbol{G}(\boldsymbol{Y}(a), \boldsymbol{Y}(b)) = \boldsymbol{0}, \tag{7.8}$$

where

$$\boldsymbol{F}(x, \boldsymbol{Y}) = \begin{bmatrix} \boldsymbol{f}(x, \boldsymbol{y}; y^{d+1}) \\ 0 \end{bmatrix}, \quad \boldsymbol{G}(\boldsymbol{Y}(a), \boldsymbol{Y}(b)) = \boldsymbol{g}(\boldsymbol{y}(a), \boldsymbol{y}(b); y^{d+1}(a)). \tag{7.9}$$

Thus, for example, the Sturm–Liouville problem (7.5) and (7.6), with

$$y_1 = u, \quad y_2 = p(x)\,\frac{\mathrm{d}u}{\mathrm{d}x}, \quad y_3 = \lambda,$$

becomes the standard two-point boundary value problem

$$\begin{aligned} \frac{\mathrm{d}y_1}{\mathrm{d}x} &= \frac{1}{p(x)}\,y_2, \\ \frac{\mathrm{d}y_2}{\mathrm{d}x} &= -q(x)y_1 + y_3y_1, \\ \frac{\mathrm{d}y_3}{\mathrm{d}x} &= 0, \end{aligned} \tag{7.10}$$

subject to

$$y_1(a) = 0, \quad y_1(b) = 0, \quad y_2(a) = p(a). \tag{7.11}$$

If one of the boundary points, say, b, is unknown, then the problem (7.1) and (7.2), where now $\boldsymbol{g} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{d+1}$, is a *problem with free boundary*. This too can be reduced to an ordinary two-point boundary value problem if one sets

$$z^{d+1} = b - a,$$

which is a constant as far as dependence on x is concerned, and introduces a new independent variable t by

$$x = a + tz^{d+1}, \quad 0 \le t \le 1.$$

Letting then

$$\boldsymbol{z}(t) = \boldsymbol{y}(a + tz^{d+1}), \quad \boldsymbol{Z}(t) = \begin{bmatrix} \boldsymbol{z}(t) \\ z^{d+1} \end{bmatrix},$$

gives

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{Z}}{\mathrm{d}t} &= \begin{bmatrix} z^{d+1}\boldsymbol{f}(a + tz^{d+1}, \boldsymbol{z}) \\ 0 \end{bmatrix}, \quad 0 \le t \le 1, \\ \boldsymbol{g}(\boldsymbol{z}(0), \boldsymbol{z}(1)) &= \boldsymbol{0}, \end{aligned} \tag{7.12}$$

a two-point boundary value problem on a fixed interval, [0,1]. Once it is solved, one recovers $b = a + z^{d+1}$ and $\boldsymbol{y}(x) = \boldsymbol{z}((x-a)/z^{d+1})$, $a \le x \le b$.

We begin with the problem of existence and uniqueness, both for linear and for nonlinear boundary value problems. We then show how initial value techniques can be employed to solve boundary value problems and discuss some of the practical difficulties associated with that. Approaches relying more on systems of linear or nonlinear equations are those based on finite difference or variational approximations, and we give a brief account of these as well.

7.1 Existence and Uniqueness

Before dealing with general considerations on existence and uniqueness (or nonuniqueness) of solutions to boundary value problems, it may be useful to look at some very simple but instructive examples.

7.1.1 Examples

For linear problems the breakdown of uniqueness or existence is exceptional and occurs, if at all, only for some "critical" intervals, often denumerably infinite in number. For nonlinear problems, the situation can be more complex.

Example. $y'' - y = 0$, $y(0) = 0$, $y(b) = \beta$.

The general solution here is made up of the hyperbolic cosine and sine. Since the hyperbolic cosine is ruled out by the first boundary condition, one obtains from the second boundary condition uniquely

$$y(x) = \beta\,\frac{\sinh x}{\sinh b}\,, \quad 0 \le x \le b. \tag{7.13}$$

There are no exceptional (critical) intervals here.

Example. $y'' + y = 0$, $y(0) = 0$, $y(b) = \beta$.

Although this problem differs only slightly from the one in the previous Example, the structure of the solution is fundamentally different because of the oscillatory nature of the general solution, consisting of the trigonometric cosine and sine. If b is not an integer multiple of π, there is a unique solution as before,

$$y(x) = \beta\,\frac{\sin x}{\sin b}\,, \quad b \ne n\pi \;\; (n = 1, 2, 3, \ldots). \tag{7.14}$$

If, however, $b = n\pi$, then there are infinitely many solutions, or none, accordingly as $\beta = 0$ or $\beta \ne 0$. In the former case, all solutions have the form $y(x) = c \sin x$, with c an arbitrary constant. In the latter case, the second boundary condition cannot be satisfied since every solution candidate must necessarily vanish at b. In either case, $b = n\pi$ is a critical point.

We now minimally modify these examples to make them nonlinear.

Example. $y'' + |y| = 0$, $y(0) = 0$, $y(b) = \beta$.

As a preliminary consideration, suppose $y(x_0) = 0$ for a solution of the differential equation. What can we say about $y(x)$ for $x > x_0$?

We distinguish three cases: (1) $y'(x_0) < 0$. In this case, y becomes negative to the right of x_0, and hence the first Example applies: the solution becomes, and remains, a negative hyperbolic sine, $y(x) = c \sinh(x - x_0)$, $c < 0$. (2) $y'(x_0) = 0$. By the uniqueness of the initial value problem (the differential equation satisfies a uniform Lipschitz condition with Lipschitz constant 1), we get $y(x) \equiv 0$ for $x > x_0$. (3) $y'(x_0) > 0$. Now y is positive on a right neighborhood of x_0, so that by the second Example, $y(x) = c \sin(x - x_0)$, $c > 0$, for $x_0 \le x \le x_0 + \pi$. At $x = x_0 + \pi$, however, $y(x_0 + \pi) = 0$, $y'(x_0 + \pi) = -c < 0$, and by what was said in Case (1), from then on we have $y(x) = -c \ \sinh(x - x_0 - \pi)$ (which ensures continuity of the first derivative of y at $x = x_0 + \pi$). Thus, in this third case, the solution $y(x)$, $x > x_0$, consists of two arcs, a trigonometric sine arc followed by a hyperbolic sine arc.

To discuss the solution of the boundary value problem at hand, we distinguish again three cases. In each case (and subcase) one arrives at the solution by considering all three possibilities $y'(0) < 0$, $y'(0) = 0$, $y'(0) > 0$, and eliminating all but one.

Case I: $b < \pi$. Here we have the unique solution

$$y(x) = \begin{cases} 0 & \text{if} \quad \beta = 0, \\[2ex] \beta \dfrac{\sin x}{\sin b} & \text{if} \quad \beta > 0, \\[2ex] \beta \dfrac{\sinh x}{\sinh b} & \text{if} \quad \beta < 0. \end{cases} \tag{7.15}$$

Case II: $b = \pi$.

(a) $\beta = 0$: *infinitely* many solutions $y(x) = c \ \sin x$, $c \ge 0$ arbitrary.
(b) $\beta > 0$: *no* solution.
(c) $\beta < 0$: *unique* solution $y(x) = \beta \dfrac{\sinh x}{\sinh \pi}$.

Case III: $b > \pi$.

(a) $\beta = 0$: *unique* solution $y(x) \equiv 0$.
(b) $\beta > 0$: *no* solution.

Table 7.1 Number of solutions of the boundary value problem in Example 7.3

b	$\beta > 0$	$\beta = 0$	$\beta < 0$
$< \pi$	1	1	1
$= \pi$	0	∞	1
$> \pi$	0	1	2

(c) $\beta < 0$: *exactly two* solutions,

$$y_1(x) = \beta \, \frac{\sinh x}{\sinh b}, \quad 0 \le x \le b, \tag{7.16}$$

$$y_2(x) = \begin{cases} -\beta \, \dfrac{\sin x}{\sinh(b-\pi)}, \; 0 \le x \le \pi, \\ \beta \, \dfrac{\sinh(x-\pi)}{\sinh(b-\pi)}, \; \pi \le x \le b. \end{cases} \tag{7.17}$$

In summary, we indicate the number of solutions in Table 7.1. It is rather remarkable how the seemingly innocuous modification of changing y to $|y|$ produces such a profound change in the qualitative behavior of the solution.

7.1.2 A Scalar Boundary Value Problem

A problem of some importance is the two-point boundary value problem for a scalar nonlinear second-order differential equation

$$y'' = f(x, y, y'), \quad a \le x \le b, \tag{7.18}$$

with linear boundary conditions

$$\begin{aligned} a_0 y(a) - a_1 y'(a) &= \alpha, \\ b_0 y(b) + b_1 y'(b) &= \beta, \end{aligned} \tag{7.19}$$

where we assume, of course, that not both a_0 and a_1 are zero, and similarly for b_0 and b_1. We further assume that f is continuous on $[a, b] \times \mathbb{R} \times \mathbb{R}$ and satisfies uniform Lipschitz conditions

$$\begin{aligned} |f(x, u_1^*, u_2) - f(x, u_1, u_2)| &\le L_1 |u_1^* - u_1|, \\ |f(x, u_1, u_2^*) - f(x, u_1, u_2)| &\le L_2 |u_2^* - u_2| \end{aligned} \tag{7.20}$$

for all $x \in [a,b]$ and all real u_1, u_2, u_1^* , u_2^*. These assumptions are sufficient to ensure that each initial value problem for (7.18) has a unique solution on the whole interval $[a,b]$ (cf. Theorem 5.3.1).

We associate with (7.18) and (7.19) the initial value problem

$$u'' = f(x,u,u'), \quad a \le x \le b, \tag{7.21}$$

subject to

$$\begin{aligned} a_0u(a) - a_1u'(a) &= \alpha, \\ c_0u(a) - c_1u'(a) &= s. \end{aligned} \tag{7.22}$$

For the two initial conditions in (7.22) to be linearly independent, we must assume that

$$\det\begin{bmatrix} a_0 & -a_1 \\ c_0 & -c_1 \end{bmatrix} \neq 0; \text{ that is, } c_0a_1 - c_1a_0 \neq 0.$$

Since we are otherwise free to choose the constants c_0, c_1 as we please, we may as well take them to satisfy

$$c_0a_1 - c_1a_0 = 1. \tag{7.23}$$

Then the initial conditions become

$$\begin{aligned} u(a) &= a_1s - c_1\alpha, \\ u'(a) &= a_0s - c_0\alpha. \end{aligned} \tag{7.24}$$

We consider c_0, c_1 to be fixed from now on, and s a parameter to be determined.

The solution of the initial value problem (7.21) and (7.24) is denoted by $u(x;s)$. If it is to solve the boundary value problem (7.18) and (7.19), we must have

$$\phi(s) = 0, \quad \phi(s) := b_0u(b;s) + b_1u'(b;s) - \beta. \tag{7.25}$$

Here and in the following, the prime in $u'(x;s)$ always indicates differentiation with respect to the first variable, x. Clearly, (7.25) is a nonlinear equation in the unknown s (cf. Chap. 4).

Theorem 7.1.1. *The boundary value problem (7.18) and (7.19) has as many distinct solutions as $\phi(s)$ has distinct zeros.*

Proof. (a) If $\phi(s_1) = 0$, then clearly $u(x;s_1)$ is a solution of the boundary value problem (7.18) and (7.19). If $s_2 \neq s_1$ is another zero of $\phi(s)$, then by (7.24) either $u(a;s_2) \neq u(a,s_1)$ (if $a_1 \neq 0$) or $u'(a;s_2) \neq u'(a;s_1)$ (if $a_0 \neq 0$); that is, $u(x;s_2) \not\equiv u(x;s_1)$. Thus, to two distinct zeros of $\phi(s)$ there correspond two distinct solutions of (7.18) and (7.19).

(b) If $y(x)$ is a solution of the boundary value problem (7.18) and (7.19), then defining $s := c_0 y(a) - c_1 y'(a)$, we have that $y(x) = u(x; s)$; hence $\phi(s) = 0$. Thus, to every solution of (7.18) and (7.19) there corresponds a zero of ϕ. □

Theorem 7.1.1 is the basis for solving (7.18) and (7.19) numerically. Solve $\phi(s) = 0$ by any of the standard methods for solving nonlinear equations. We discuss this in more detail in Sect. 7.2.

For a certain class of boundary value problems (7.18) and (7.19) one can show that $\phi(s) = 0$ has exactly one solution.

Theorem 7.1.2. *Assume that*

(1) $f(x, u_1, u_2)$ *is continuous on* $[a, b] \times \mathbb{R} \times \mathbb{R}$.
(2) *Both* f_{u_1} *and* f_{u_2} *are continuous and satisfy*

$$0 < f_{u_1}(x, u_1, u_2) \le L_1, \ |f_{u_2}(x, u_1, u_2)| \le L_2 \ on \ [a, b] \times \mathbb{R} \times \mathbb{R}.$$

(3) $a_0 a_1 \ge 0, \ \ b_0 b_1 \ge 0, \ \ |a_0| + |b_0| > 0.$

Then the boundary value problem (7.18) and (7.19) has a unique solution.

Note that Assumption (2) implies (7.20), hence unique solvability on $[a, b]$ of initial value problems for (7.18). Assumption (3) requires that a_0 and a_1 be of the same sign, as well as b_0 and b_1, and that not both a_0 and b_0 vanish. We may assume, by multiplying one or both of the boundary conditions (7.19) by –1 if necessary, that

$$(3') \qquad a_0 \ge 0, a_1 \ge 0; \ \ b_0 \ge 0, b_1 \ge 0; \ \ a_0 + b_0 > 0.$$

Proof of Theorem 7.1.2. The idea of the proof is to show that

$$\phi'(s) \ge c > 0 \ \text{ for all } \ s \in \mathbb{R}. \tag{7.26}$$

The function $\phi(s)$ then increases monotonically from $-\infty$ to $+\infty$, and hence vanishes for exactly one value of s.

We have

$$\phi'(s) = b_0 \frac{\partial}{\partial s} u(b; s) + b_1 \frac{\partial}{\partial s} u'(b; s).$$

It is convenient to denote

$$v(x) = \frac{\partial}{\partial s} u(x; s),$$

where the dependence on s is suppressed in the notation for v. Since differentiation with respect to x and s may be interchanged under the assumptions made, we can write

$$\phi'(s) = b_0 v(b) + b_1 v'(b). \tag{7.27}$$

Furthermore, $u(x;s)$ satisfies, identically in s,

$$u''(x;s) = f(x, u(x;s), u'(x;s)), \quad a \le x \le b,$$
$$u(a;s) = a_1 s - c_1\alpha, \quad u'(a;s) = a_0 s - c_0\alpha,$$

from which, by differentiation with respect to s and interchange with differentiation in x where necessary, one gets

$$v''(x) = f_{u_1}(x, u(x;s), u'(x;s))v(x) + f_{u_2}(x, u(x;s), u'(x;s))v'(x),$$
$$v(a) = a_1, \quad v'(a) = a_0. \tag{7.28}$$

Thus, v is the solution of a "linear" boundary value problem,

$$v'' = p(x)v' + q(x)v, \quad a \le x \le b,$$
$$v(a) = a_1, \quad v'(a) = a_0, \tag{7.29}$$

where

$$|p(x)| \le L_2, \quad 0 < q(x) \le L_1 \quad \text{on } [a,b]. \tag{7.30}$$

We are going to show that, on $a \le x \le b$,

$$v(x) > a_1 + a_0\,\frac{1 - \mathrm{e}^{-L_2(x-a)}}{L_2}\,, \quad v'(x) > a_0\mathrm{e}^{-L_2(x-a)}. \tag{7.31}$$

From this, (7.26) will follow. Indeed, since not both a_0 and a_1 can vanish and by (3′) at least one is positive, it follows from (7.31) that $v(b) > 0$. If $b_0 > 0$, then (7.27) shows, since $b_1 \ge 0$ and $v'(b) > 0$ by (7.31), that $\phi'(s)$ is positive and bounded away from 0 (as a function of s). The same conclusion follows if $b_0 = 0$, since then $b_1 > 0$ and $\phi'(s) = b_1 v'(b) > 0$ in (7.27).

To prove (7.31), we first show that $v(x) > 0$ for $a < x \le b$. This is certainly true in a small right neighborhood of a, since by (7.29) either $v(a) > 0$ or $v(a) = 0$ and $v'(a) > 0$. If the assertion were false, we would therefore have $v(x_0) = 0$ for some x_0 in $(a,b]$. But then v must have a local maximum at some x_1 with $a < x_1 < x_0$. This is clear in the cases where $v(a) = 0$, $v'(a) > 0$, and $v(a) > 0$, $v'(a) > 0$. In the remaining case $v(a) > 0$, $v'(a) = 0$, it follows from the fact that then $v''(a) > 0$ by virtue of the differential equation in (7.29) and the positivity of q (cf. (7.30)). Thus,

$$v(x_1) > 0, \quad v'(x_1) = 0, \quad v''(x_1) < 0.$$

But this contradicts the differential equation (7.29) at $x = x_1$, since $q(x_1) > 0$. This establishes the positivity of v on $(a,b]$. We thus have, using again the positivity of q,

$$v''(x) - p(x)v'(x) > 0 \ \text{ for } \ a < x \le b.$$

Multiplication by the "integrating factor" $\exp\left(-\int_a^x p(t)\mathrm{d}t\right)$ yields

$$\frac{\mathrm{d}}{\mathrm{d}x}\left[\mathrm{e}^{-\int_a^x p(t)\mathrm{d}t} v'(x)\right] > 0,$$

and upon integration from a to x,

$$\mathrm{e}^{-\int_a^x p(t)\mathrm{d}t} v'(x) - v'(a) > 0.$$

This, in turn, by the second initial condition in (7.29), gives

$$v'(x) > a_0 \mathrm{e}^{\int_a^x p(t)\mathrm{d}t},$$

from which the second inequality in (7.31) follows by virtue of $p(t) \geq -L_2$ (cf. (7.30)). The first inequality in (7.31) follows by integrating the second from a to x. □

Theorem 7.1.2 has an immediate application to the Sturm–Liouville problem

$$\mathcal{L}y = r(x), \;\; a \leq x \leq b; \;\; \mathcal{B}_a y = \alpha, \;\; \mathcal{B}_b y = \beta, \tag{7.32}$$

where

$$\begin{aligned} \mathcal{L}y &:= -y'' + p(x)y' + q(x)y, \\ \mathcal{B}_a y &= a_0 y(a) - a_1 y'(a), \quad \mathcal{B}_b y = b_0 y(b) + b_1 y'(b). \end{aligned} \tag{7.33}$$

Corollary 7.1.1. *If p, q, and r are continuous on $[a, b]$ with*

$$q(x) > 0 \;\; for \;\; a \leq x \leq b, \tag{7.34}$$

and if a_0, a_1, b_0, b_1 satisfy the condition (3) of Theorem 7.1.2, then (7.32) has a unique solution.

We remark that the differential equation in (7.32) can be written equivalently in "self-adjoint form" if we multiply both sides by $P(x) = \exp(-\int p\mathrm{d}x)$. This yields

$$-\frac{\mathrm{d}}{\mathrm{d}x}\left(P(x)\frac{\mathrm{d}y}{\mathrm{d}x}\right) + Q(x)y = R(x), \quad a \leq x \leq b, \tag{7.35}$$

with

$$Q(x) = P(x)q(x), \;\; R(x) = P(x)r(x).$$

Note that P in (7.35) is positive and not only continuous, but also continuously differentiable on $[a, b]$. Furthermore, the positivity of q is equivalent to the positivity of Q.

The following is a result of a somewhat different nature, providing an alternative.

Theorem 7.1.3. *The boundary value problem (7.32) has a unique solution for arbitrary α and β if and only if the corresponding homogeneous problem (with $r \equiv 0$, $\alpha = \beta = 0$) has only the trivial solution $y \equiv 0$.*

Proof. Define u_1 and u_2 by

$$\mathcal{L}u_1 = r(x),\ \ a \le x \le b;\ \ u_1(a) = -c_1\alpha,\ u_1'(a) = -c_0\alpha,$$

and

$$\mathcal{L}u_2 = 0,\ \ a \le x \le b;\ \ u_2(a) = a_1,\ u_2'(a) = a_0,$$

with c_0, c_1 as defined in (7.23). Then one easily verifies that $\mathcal{B}_a u_1 = \alpha$, $\mathcal{B}_a u_2 = 0$, so that

$$u(x) = u_1(x) + su_2(x) \tag{7.36}$$

satisfies both the inhomogeneous differential equation and the first boundary condition of (7.32). The inhomogeneous boundary value problem therefore has a unique solution if and only if

$$\mathcal{B}_b u_2 \neq 0, \tag{7.37}$$

so that the second boundary condition $\mathcal{B}_b u = \beta$ can be solved uniquely for s in (7.36). On the other hand, for the homogeneous boundary value problem to have only the trivial solution, we must have (7.37), since otherwise $\mathcal{L}u_2 = 0$, $\mathcal{B}_a u_2 = 0$, and $\mathcal{B}_b u_2 = 0$, whereas one of $u_2(a)$ and $u_2'(a)$ must be different from zero, since not both a_1 and a_0 can vanish. □

7.1.3 General Linear and Nonlinear Systems

The two-point boundary value problem for the general nonlinear system (7.1), with linear boundary conditions, takes the form

$$\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}x} = \boldsymbol{f}(x, \boldsymbol{y}), \quad a \le x \le b,$$

$$\boldsymbol{A}\boldsymbol{y}(a) + \boldsymbol{B}\boldsymbol{y}(b) = \boldsymbol{\gamma}, \tag{7.38}$$

where $\boldsymbol{A}$, $\boldsymbol{B}$ are square matrices of order d with constant elements, and $\boldsymbol{\gamma}$ is a given d-vector. For linear independence and consistency we assume that

$$\operatorname{rank}\,[\boldsymbol{A}, \boldsymbol{B}] = d. \tag{7.39}$$

In this general form, the associated initial value problem is

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}x} &= \boldsymbol{f}(x, \boldsymbol{u}), \quad a \le x \le b, \\ \boldsymbol{u}(a) &= \boldsymbol{s}, \end{aligned} \tag{7.40}$$

where $\boldsymbol{s} \in \mathbb{R}^d$ is a "trial" initial vector. If we denote by $\boldsymbol{u}(x; \boldsymbol{s})$ the solution of (7.40) and assume it to exist on $[a, b]$, then (7.38) is equivalent to a problem of solving a nonlinear system of equations,

$$\boldsymbol{\phi}(\boldsymbol{s}) = \boldsymbol{0}, \quad \boldsymbol{\phi}(\boldsymbol{s}) = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{B}\boldsymbol{u}(b; \boldsymbol{s}) - \boldsymbol{\gamma}. \tag{7.41}$$

Again, the boundary value problem (7.38) has as many distinct solutions as the nonlinear system (7.41) has distinct solution vectors. By imposing sufficiently strong – but often unrealistic – conditions on $\boldsymbol{f}$, $\boldsymbol{A}$, and $\boldsymbol{B}$, it is possible to prove that (7.41), and hence (7.38), has a unique solution, but we do not pursue this any further.

For linear systems, we have

$$\boldsymbol{f}(x, \boldsymbol{y}) = \boldsymbol{C}(x)\boldsymbol{y} + \boldsymbol{d}(x), \quad a \le x \le b, \tag{7.42}$$

in which case the initial value problem (7.40) is known to have the solution

$$\boldsymbol{u}(x) = \boldsymbol{Y}(x)\boldsymbol{s} + \boldsymbol{v}(x), \tag{7.43}$$

where $\boldsymbol{Y}(x) \in \mathbb{R}^{d \times d}$ is a fundamental solution of the homogeneous system $d\boldsymbol{Y}/\mathrm{d}x = \boldsymbol{C}(x)\boldsymbol{Y}$ with initial value $\boldsymbol{Y}(a) = \boldsymbol{I}$, and $\boldsymbol{v}(x)$ a particular solution of the inhomogeneous system $d\boldsymbol{v}/\mathrm{d}x = \boldsymbol{C}(x)\boldsymbol{v} + \boldsymbol{d}(x)$ satisfying $\boldsymbol{v}(a) = \boldsymbol{0}$. The boundary value problem (7.38) is then equivalent to the system of linear algebraic equations

$$[\boldsymbol{A} + \boldsymbol{B}\boldsymbol{Y}(b)]\boldsymbol{s} = \boldsymbol{\gamma} - \boldsymbol{B}\boldsymbol{v}(b) \tag{7.44}$$

and has a unique solution if and only if the matrix of this system is nonsingular.

We remark that if some components of $\boldsymbol{y}(a)$ are prescribed as part of the boundary conditions in (7.38), then of course they are incorporated in the vector $\boldsymbol{s}$, and one obtains a smaller system of nonlinear (resp., linear) equations in the remaining (unknown) components of $\boldsymbol{s}$.

7.2 Initial Value Techniques

The techniques used in Sects. 7.1.2 and 7.1.3 are also of computational interest in that they lend themselves to the application of numerical methods for solving nonlinear equations or systems of equations. We show, for example, how Newton's method can be used in this context, first for a scalar second-order boundary value problem, and then for a general problem involving a first-order system of differential equations.

7.2.1 Shooting Method for a Scalar Boundary Value Problem

We have seen in Sect. 7.1.2 that the boundary value problem (7.18) and (7.19) leads to the nonlinear equation (7.25) via the initial value problem (7.21) and (7.24). Solving the initial value problem is referred to, in this context, as "shooting." One aims by means of trial initial conditions to satisfy the second boundary condition, which is the "target." A mechanism of readjusting the aim, based on the amount by which the target has been missed, is provided by Newton's method. Specifically, one starts with an initial approximation $s^{(0)}$ for s in (7.25), and then iterates according to

$$s^{(\nu+1)} = s^{(\nu)} - \frac{\phi(s^{(\nu)})}{\phi'(s^{(\nu)})}\,, \quad \nu = 0, 1, 2, \dots\,, \tag{7.45}$$

until, it is hoped, $s^{(\nu)} \to s_\infty$ as $\nu \to \infty$. If that occurs, then $y(x) = u(x; s_\infty)$ will be a solution of the boundary value problem. If there is more than one solution, the process (7.45) needs to be repeated, perhaps several times, with different starting values $s^{(0)}$.

For any given s, the values of $\phi(s)$ and $\phi'(s)$ needed in (7.45) are computed simultaneously by "shooting," that is, by solving the initial value problem (7.21) and (7.24) together with the one in (7.28) obtained by differentiation with respect to s. If both are written as first-order systems, by letting

$$y_1(x) = u(x; s), \quad y_2(x) = u'(x; s), \quad y_3(x) = v(x), \quad y_4(x) = v'(x),$$

one solves on $[a, b]$ the initial value problem

$$\begin{aligned}
\frac{\mathrm{d}y_1}{\mathrm{d}x} &= y_2, & y_1(a) &= a_1 s - c_1\alpha,\\
\frac{\mathrm{d}y_2}{\mathrm{d}x} &= f(x, y_1, y_2), & y_2(a) &= a_0 s - c_0\alpha,\\
\frac{\mathrm{d}y_3}{\mathrm{d}x} &= y_4, & y_3(a) &= a_1,\\
\frac{\mathrm{d}y_4}{\mathrm{d}x} &= f_{u_1}(x, y_1, y_2)y_3 + f_{u_2}(x, y_1, y_2)y_4, & y_4(a) &= a_0,
\end{aligned} \tag{7.46}$$

with $c_0,\ c_1$ as chosen in (7.23), and then computes

$$\phi(s) = b_0 y_1(b) + b_1 y_2(b) - \beta, \quad \phi'(s) = b_0 y_3(b) + b_1 y_4(b). \tag{7.47}$$

Thus, each Newton step (7.45) requires the solution on $[a, b]$ of an initial value problem (7.46) with $s = s^{(\nu)}$.

Example. $y'' = -\mathrm{e}^{-y}, 0 \le x \le 1, y(0) = y(1) = 0.$

We first show that this problem has a unique solution. To do this, we "embed" the problem into the following problem:

$$y'' = f(y),\ 0 \le x \le 1;\ \ y(0) = y(1) = 0, \tag{7.48}$$

where

$$f(y) = \begin{cases} -\mathrm{e}^{-y} & \text{if} \quad y \ge 0, \\ \mathrm{e}^{y} - 2 & \text{if} \quad y \le 0. \end{cases} \tag{7.49}$$

Then $f_y(y) = \mathrm{e}^{-y}$ if $y \ge 0$ and $f_y(y) = \mathrm{e}^{y}$ if $y \le 0$, so that $0 < f_y(y) \le 1$ for all real y. Thus, Assumption (2) of Theorem 7.1.2 is satisfied, and so are (trivially) the other two assumptions. It follows that (7.48) has a unique solution, and since clearly this solution cannot become negative (the second derivative being necessarily negative), it also solves the problem in our Example.

Since $a_0 = b_0 = 1, a_1 = b_1 = \alpha = \beta = 0$ in this example, the system (7.46) becomes, for $0 \le x \le 1$,

$$\begin{aligned} \frac{\mathrm{d}y_1}{\mathrm{d}x} &= y_2, & y_1(a) &= 0, \\ \frac{\mathrm{d}y_2}{\mathrm{d}x} &= -\mathrm{e}^{-y_1}, & y_2(0) &= s, \\ \frac{\mathrm{d}y_3}{\mathrm{d}x} &= y_4, & y_3(0) &= 0, \\ \frac{\mathrm{d}y_4}{\mathrm{d}x} &= \mathrm{e}^{-y_1} y_3, & y_4(0) &= 1, \end{aligned}$$

and (7.47) simplifies to

$$\phi(s) = y_1(1), \quad \phi'(s) = y_3(1).$$

Newton's method (7.45), of course, has to be started with a positive initial approximation $s^{(0)}$. If we use $s^{(0)} = 1$ and an error tolerance of $\frac{1}{2}\,10^{-12}$, it produces, with the help of Matlab routine `ode45`, the results shown in Table 7.2 (cf. also Ex. 3).

Example. $y'' = \lambda \sinh(\lambda y), 0 \le x \le 1, y(0) = y(1) = 0.$

If $y'(0) = s$, $s \ne 0$, the solution y of the differential equation has the sign of s in a right neighborhood of $x = 0$, and so does y''. Thus, $|y|$ is monotonically increasing and cannot attain the value zero at $x = 1$. It follows that $y(x) \equiv 0$ is the unique solution of the boundary value problem. Moreover, it can be shown (see Ex. 8(c)) that for $y'(0) = s$ the modulus $|y|$ of the solution tends to infinity at a

Table 7.2 Numerical results for the first Example

ν	$s^{(\nu)}$
0	1.0
1	0.45
2	0.463630
3	0.463632593668

finite value x_∞ of x, where

$$x_\infty \sim \frac{1}{\lambda} \ln \frac{8}{|s|}, \quad s \to 0. \tag{7.50}$$

Thus, if we apply shooting with $y'(0) = s$, we can reach the end point $x = 1$ without the solution blowing up on us only if $x_\infty > 1$, that is, approximately, if

$$|s| < 8\mathrm{e}^{-\lambda}. \tag{7.51}$$

This places a severe restriction on the permissible initial slope; for example, $|s| < 3.63\ldots \times 10^{-4}$ if $\lambda = 10$ and $|s| < 1.64\ldots \times 10^{-8}$ if $\lambda = 20$. It is indeed one of the limitations of “ordinary” shooting that rather accurate initial data must be known in order to succeed.

7.2.2 Linear and Nonlinear Systems

For linear systems, the shooting method amounts to solving the linear system of algebraic equations (7.44), which requires the numerical solution of $d + 1$ initial value problems to obtain $\boldsymbol{Y}(b)$ and $\boldsymbol{v}(b)$, and possibly one more final integration, with the starting vector found, to determine the solution $\boldsymbol{y}(x)$ at the desired values of x. It is strictly a superposition method and no iteration is required, but the procedure often suffers from ill-conditioning of the matrix involved.

For the general nonlinear boundary value problem (7.38), there is no difficulty, formally, in defining a shooting method. One simply has to solve the system of nonlinear equations (7.41), for example by Newton's method,

$$\left.\begin{aligned} \boldsymbol{s}^{(\nu+1)} &= \boldsymbol{s}^{(\nu)} + \boldsymbol{\Delta}_\nu \\ \frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{s}}(\boldsymbol{s}^{(\nu)})\boldsymbol{\Delta}_\nu &= -\boldsymbol{\phi}(\boldsymbol{s}^{(\nu)}) \end{aligned}\right\} \quad \nu = 0, 1, 2, \ldots, \tag{7.52}$$

where $\partial\boldsymbol{\phi}/\partial\boldsymbol{s}$ is the Jacobian of $\boldsymbol{\phi}$,

$$\frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{s}}(\boldsymbol{s}) = \boldsymbol{A} + \boldsymbol{B}\,\frac{\partial \boldsymbol{u}(b; \boldsymbol{s})}{\partial \boldsymbol{s}}\,.$$

With $\boldsymbol{u}(x;\boldsymbol{s})$ denoting, as before, the solution of (7.40), we let $\boldsymbol{V}(x)$ be its Jacobian (with respect to $\boldsymbol{s}$),

$$\boldsymbol{V}(x) = \frac{\partial \boldsymbol{u}(x;\boldsymbol{s})}{\partial \boldsymbol{s}}, \quad a \le x \le b.$$

Then, in order to simultaneously compute $\boldsymbol{\phi}(\boldsymbol{s})$ and $(\partial\boldsymbol{\phi}/\partial\boldsymbol{s})(\boldsymbol{s})$, we can integrate the initial value problem (7.40) adjoined with that for its Jacobian,

$$\left.\begin{aligned} \frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}x} &= \boldsymbol{f}(x,\boldsymbol{u}) \\ \frac{\mathrm{d}\boldsymbol{V}}{\mathrm{d}x} &= \boldsymbol{f}_{\boldsymbol{y}}(x,\boldsymbol{u})\boldsymbol{V} \\ \boldsymbol{u}(a) = \boldsymbol{s},\ \ &\boldsymbol{V}(a) = \boldsymbol{I} \end{aligned}\right\} \quad a \le x \le b, \tag{7.53}$$

and get

$$\boldsymbol{\phi}(\boldsymbol{s}) = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{B}\boldsymbol{u}(b;\boldsymbol{s}) - \boldsymbol{\gamma}, \ \ \frac{\partial\boldsymbol{\phi}}{\partial\boldsymbol{s}}(\boldsymbol{s}) = \boldsymbol{A} + \boldsymbol{B}\boldsymbol{V}(b). \tag{7.54}$$

Although the procedure is formally straightforward, it is difficult to implement. For one thing, we may not be able to integrate (7.53) on the whole interval $[a,b]$; some component may blow up before we reach the endpoint b (cf. the second Example of Sect. 7.2.1). Another problem has to do with the convergence of Newton's method (7.52). Typically, this requires $\boldsymbol{s}$ in (7.53) to be very close to $\boldsymbol{s}_\infty$ – the true initial vector for one of the possible solutions of the boundary value problem. A good illustration of these difficulties is provided by the following example.

Example.

$$\left.\begin{aligned} \frac{\mathrm{d}y_1}{\mathrm{d}x} &= \frac{y_1^2}{y_2} \\ \frac{\mathrm{d}y_2}{\mathrm{d}x} &= \frac{y_2^2}{y_1} \end{aligned}\right\} \quad 0 \le x \le 1, \tag{7.55}$$

$$y_1(0) = 1, \ \ y_1(1) = -\mathrm{e}\ (= 2.718\ldots).$$

This is really a linear system in disguise, namely, the one for the reciprocal functions y_1^{-1} and y_2^{-1}. Hence it is easily seen that an exact solution to (7.55) is (cf. Ex. 4)

$$y_1(x) = y_2(x) = \mathrm{e}^x, \quad 0 \le x \le 1. \tag{7.56}$$

We write the system in this complicated nonlinear form to bring out the difficulties inherent in the shooting method.

Since y_1 is known at $x = 0$, we have only one unknown, $s = y_2(0)$. We thus define $u_1(x; s)$, $u_2(x; s)$ to be the solution of the initial value problem

$$\frac{\mathrm{d}}{\mathrm{d}x}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \dfrac{u_1^2}{u_2} \\ \dfrac{u_2^2}{u_1} \end{bmatrix}, \ 0 \le x \le 1; \quad \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}(0) = \begin{bmatrix} 1 \\ s \end{bmatrix}. \tag{7.57}$$

The equation to be solved then is

$$\phi(s) = 0, \ \ \phi(s) = u_1(1; s) - \mathrm{e}. \tag{7.58}$$

We denote $\frac{\partial u_1(x;s)}{\partial s} = v_1(x)$, $\frac{\partial u_2(x;s)}{\partial s} = v_2(x)$, differentiate (7.57) with respect to s, and note that the Jacobian of (7.57) is

$$\boldsymbol{f_y}(u_1, u_2) = \begin{bmatrix} \dfrac{2u_1}{u_2} & -\dfrac{u_1^2}{u_2^2} \\ -\dfrac{u_2^2}{u_1^2} & \dfrac{2u_2}{u_1} \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}.$$

If we append the differentiated system to the original one, we get

$$\begin{aligned} \frac{\mathrm{d}u_1}{\mathrm{d}x} &= \frac{u_1^2}{u_2}, & u_1(0) &= 1, \\ \frac{\mathrm{d}u_2}{\mathrm{d}x} &= \frac{u_2^2}{u_1}, & u_2(0) &= s, \\ \frac{\mathrm{d}v_1}{\mathrm{d}x} &= \frac{2u_1}{u_2}\, v_1 - \frac{u_1^2}{u_2^2}\, v_2, & v_1(0) &= 0, \\ \frac{\mathrm{d}v_2}{\mathrm{d}x} &= -\frac{u_2^2}{u_1^2}\, v_1 + \frac{2u_2}{u_1}\, v_2, & v_2(0) &= 1. \end{aligned} \tag{7.59}$$

Assuming this can be solved on [0,1], we will have

$$\phi(s) = u_1(1; s) - \mathrm{e}, \ \ \phi'(s) = v_1(1), \tag{7.60}$$

and can thus apply Newton's method,

$$s^{(\nu+1)} = s^{(\nu)} - \frac{\phi(s^{(\nu)})}{\phi'(s^{(\nu)})}\,, \tag{7.61}$$

taking $s = s^{(\nu)}$ in (7.59).

Because of the elementary nature of the system (7.57), we can solve the initial value problem (7.57) in closed form (cf. Ex. 4),

$$u_1(x;s) = \frac{s}{s\cosh x - \sinh x}, \quad u_2(x;s) = \frac{s}{\cosh x - s\sinh x}. \tag{7.62}$$

It is convenient to look at the solutions of (7.57) as curves in the phase plane (u_1, u_2). The desired solution, by (7.56), is then given by $u_1 = u_2$.

Clearly, s has to be positive in (7.57), since otherwise u_1 would initially decrease and, to meet the condition at $x = 1$, would have to turn around and have a vanishing derivative at some point in (0,1). That would cause a problem in (7.57), since either $u_1 = 0$ or $u_2 = \infty$ at that point.

For u_1 to remain bounded on [0,1], it then follows from the first relation in (7.62) that we must have $s > \tanh x$ for $0 \le x \le 1$; that is,

$$s > \tanh 1 = 0.76159\ldots\,.$$

At $s = \tanh 1$, we have

$$\lim_{x\to 1} u_1(x;\tanh 1) = \infty, \quad \lim_{x\to 1} u_2(x;\tanh 1) = \sinh 1 = 1.1752\ldots\,.$$

This solution, in the phase plane, has a horizontal asymptote.

Similarly, for u_2 to remain bounded on [0,1], we must have

$$s < \coth 1 = 1.3130\ldots\,.$$

When $s = \coth 1$, then

$$\lim_{x\to 1} u_1(x;\coth 1) = \cosh 1 = 1.5430\ldots\,, \quad \lim_{x\to 1} u_2(x;\coth 1) = \infty,$$

giving a solution with a vertical asymptote. The locus of points $[u_1(1;s), u_2(1;s)]$ as $\tanh 1 < s < \coth 1$ is easily found to be the hyperbola

$$u_2 = \frac{u_1 \sinh 1}{u_1 - \cosh 1}.$$

From this, we get a complete picture in the phase plane of all solutions of (7.57); see Fig. 7.1. Thus, only in a relatively small s-interval, $0.76159\ldots < s < 1.3130\ldots$, it is possible to shoot from the initial point to the endpoint without one component of the solution blowing up in between.

What about the convergence of Newton's method (7.61)? The equation to be solved is (7.58), that is,

$$\phi(s) = 0, \quad \phi(s) = \frac{s}{s\cosh 1 - \sinh 1} - \mathrm{e}.$$

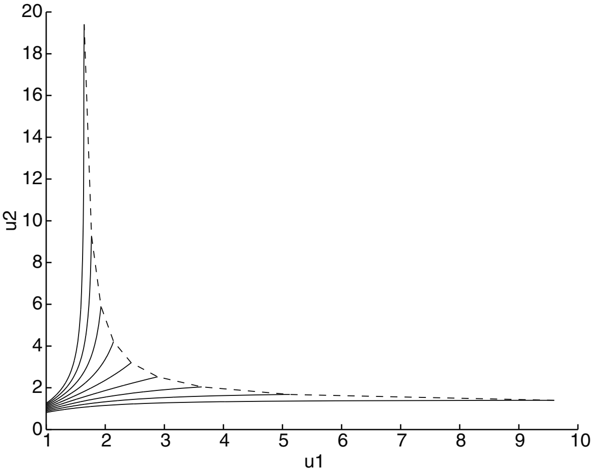


Fig. 7.1 The solutions of (7.57) in the phase plane. The *solid lines* are solutions for fixed s in $\tanh 1 \leq s \leq \coth 1$ and x varying between 0 and 1. The *dashed line* is the locus of points $[u_1(1;s), u_2(1;s)]$, $\tanh 1 < s < \coth 1$

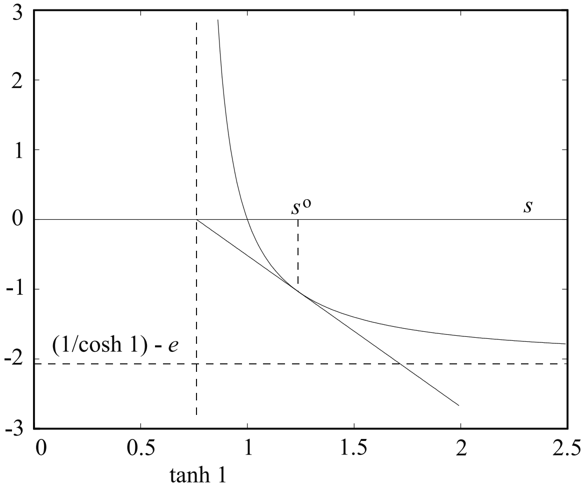


Fig. 7.2 The graph of $\phi(s)$ and convergence of Newton's method

From the graph of $\phi(s)$ (see Fig. 7.2), in particular, the convexity of ϕ, it follows that Newton's method converges precisely if

$$\tanh 1 < s < s^0, \tag{7.63}$$

where s^0 is such that

$$s^0 - \frac{\phi(s^0)}{\phi'(s^0)} = \tanh 1. \tag{7.64}$$

It can be shown (see Ex. 5(a)) that $s^0 < \coth 1$, so that the convergence of Newton's method imposes an additional restriction on the choice of s.

7.2.3 Parallel Shooting

To circumvent the difficulties inherent in ordinary shooting, as indicated in Examples of Sects. 7.2.1 and 7.2.2, one may divide the interval $[a, b]$ into N subintervals,

$$a = x_0 < x_1 < x_2 < \cdots < x_{N-1} < x_N = b, \tag{7.65}$$

and apply shooting concurrently on each subinterval. The hope is that if the intervals are sufficiently small, not only does the appropriate boundary value problem have a unique solution, but also the solution is not given a chance to grow excessively. We say "appropriate" boundary value problem since in addition to the two boundary conditions, there are now also continuity conditions at the interior subdivision points. This, of course, enlarges considerably the problem size. To enhance the prospects of success, it is advisable to generate the subdivision (7.65) dynamically, as described further on, rather than to choose it artificially without regard to the particular features of the problem at hand.

To describe the procedure in more detail, consider the boundary value problem for a general nonlinear system,

$$\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}x} = \boldsymbol{f}(x, \boldsymbol{y}), \;\; a \le x \le b; \quad \boldsymbol{A}\boldsymbol{y}(a) + \boldsymbol{B}\boldsymbol{y}(b) = \boldsymbol{\gamma}. \tag{7.66}$$

Let $h_n = x_n - x_{n-1}$, $n = 1, 2, \ldots, N$, and define

$$\boldsymbol{y}_n(t) = \boldsymbol{y}(x_{n-1} + th_n), \;\; 0 \le t \le 1. \tag{7.67}$$

Clearly,

$$\frac{\mathrm{d}\boldsymbol{y}_n}{\mathrm{d}t} = h_n \boldsymbol{y}'(x_{n-1} + th_n) = h_n \boldsymbol{f}(x_{n-1} + th_n, \boldsymbol{y}_n(t)), \;\; 0 \le t \le 1.$$

Thus, by letting

$$\boldsymbol{f}_n(t, \boldsymbol{z}) = h_n \boldsymbol{f}(x_{n-1} + th_n, \boldsymbol{z}), \tag{7.68}$$

we have for the (vector-valued) functions $\boldsymbol{y}_n$ the following system of, as yet uncoupled, differential equations

$$\frac{\mathrm{d}\boldsymbol{y}_n}{\mathrm{d}t} = \boldsymbol{f}_n(t, \boldsymbol{y}_n), \ \ n = 1, 2, \ldots, N; \ \ 0 \le t \le 1. \tag{7.69}$$

The coupling comes about through the boundary and interface (continuity) conditions

$$\begin{aligned} &\boldsymbol{A}\boldsymbol{y}_1(0) + \boldsymbol{B}\boldsymbol{y}_N(1) = \boldsymbol{\gamma}, \\ &\quad \boldsymbol{y}_{n+1}(0) - \boldsymbol{y}_n(1) = \boldsymbol{0}, \ \ n = 1, 2, \ldots, N-1. \end{aligned} \tag{7.70}$$

Introducing "supervectors" and "supermatrices,"

$$\boldsymbol{Y}(t) = \begin{bmatrix} \boldsymbol{y}_1(t) \\ \vdots \\ \boldsymbol{y}_N(t) \end{bmatrix}, \quad \boldsymbol{F}(t, \boldsymbol{Y}) = \begin{bmatrix} \boldsymbol{f}_1(t, \boldsymbol{y}_1) \\ \vdots \\ \boldsymbol{f}_N(t, \boldsymbol{y}_N) \end{bmatrix}, \quad \Gamma = \begin{bmatrix} \boldsymbol{\gamma} \\ \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \end{bmatrix},$$

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{I} \end{bmatrix}, \quad \boldsymbol{Q} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{B} \\ -\boldsymbol{I} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & -\boldsymbol{I} & \boldsymbol{0} \end{bmatrix}, \tag{7.71}$$

we can write (7.69) and (7.70) compactly as

$$\frac{\mathrm{d}\boldsymbol{Y}}{\mathrm{d}t} = \boldsymbol{F}(t, \boldsymbol{Y}), \ \ 0 \le t \le 1; \ \ \boldsymbol{P}\boldsymbol{Y}(0) + \boldsymbol{Q}\boldsymbol{Y}(1) = \boldsymbol{\Gamma}; \tag{7.72}$$

this has the same form as (7.66) but is much bigger in size. Parallel shooting consists of applying ordinary shooting to the big system (7.72). Thus, we solve on $0 \le t \le 1$

$$\frac{\mathrm{d}\boldsymbol{U}}{\mathrm{d}t} = \boldsymbol{F}(t, \boldsymbol{U}), \ \ \boldsymbol{U}(0) = \boldsymbol{S}, \tag{7.73}$$

to obtain $\boldsymbol{U}(t) = \boldsymbol{U}(t; \boldsymbol{S})$ and try to determine the vector $\boldsymbol{S} \in \mathbb{R}^{Nd}$ such that

$$\boldsymbol{\Phi}(\boldsymbol{S}) = \boldsymbol{P}\boldsymbol{S} + \boldsymbol{Q}\boldsymbol{U}(1; \boldsymbol{S}) - \boldsymbol{\Gamma} = \boldsymbol{0}. \tag{7.74}$$

If we use Newton's method, this is done by the iteration

$$\left.\begin{aligned} \boldsymbol{S}^{(\nu+1)} &= \boldsymbol{S}^{(\nu)} + \boldsymbol{\Delta}_\nu, \\ [\boldsymbol{P} + \boldsymbol{Q}\boldsymbol{V}(1; \boldsymbol{S}^{(\nu)})]\boldsymbol{\Delta}_\nu &= -\boldsymbol{\Phi}(\boldsymbol{S}^{(\nu)}) \end{aligned}\right\} \quad \nu = 0, 1, 2, \ldots, \tag{7.75}$$

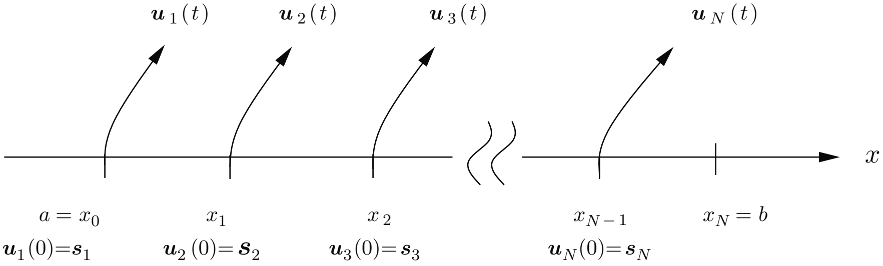


Fig. 7.3 Parallel shooting (one-sided)

where

$$V(t;\boldsymbol{S}) = \frac{\partial \boldsymbol{U}}{\partial \boldsymbol{S}}(t;\boldsymbol{S}), \quad 0 \le t \le 1. \tag{7.76}$$

If we partition $\boldsymbol{U}^{\mathrm{T}} = [\boldsymbol{u}_1^{\mathrm{T}}, \dots, \boldsymbol{u}_N^{\mathrm{T}}]$, $\boldsymbol{S}^{\mathrm{T}} = [\boldsymbol{s}_1^{\mathrm{T}}, \dots, \boldsymbol{s}_N^{\mathrm{T}}]$ in accordance with (7.69), then, since (7.69) by itself is uncoupled, we find that $\boldsymbol{u}_n = \boldsymbol{u}_n(t;\boldsymbol{s}_n)$ depends only on $\boldsymbol{s}_n$. As a consequence, the "big" Jacobian $\boldsymbol{V}$ in (7.76) is block diagonal,

$$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{v}_2 & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \dots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{v}_N \end{bmatrix}, \quad \boldsymbol{v}_n(t;\boldsymbol{s}_n) = \frac{\partial \boldsymbol{u}_n}{\partial \boldsymbol{s}_n}(t;\boldsymbol{s}_n), \; n = 1, 2, \dots, N,$$

and so is the Jacobian $\boldsymbol{F}_{\boldsymbol{Y}}(t, \boldsymbol{U})$ of $\boldsymbol{F}$ in (7.73). This means that $\boldsymbol{U}$ in (7.73) and $\boldsymbol{V}$ in (7.76) can be computed by solving uncoupled systems of initial value problems on $0 \le t \le 1$,

$$\left.\begin{aligned} \frac{\mathrm{d}\boldsymbol{u}_n}{\mathrm{d}t} &= \boldsymbol{f}_n(t, \boldsymbol{u}_n), \;\; \boldsymbol{u}_n(0) = \boldsymbol{s}_n \\ \frac{\mathrm{d}\boldsymbol{v}_n}{\mathrm{d}t} &= \frac{\partial \boldsymbol{f}_n}{\partial \boldsymbol{y}_n}(t, \boldsymbol{u}_n)\boldsymbol{v}_n, \;\; \boldsymbol{v}_n(0) = \boldsymbol{I}_n \end{aligned}\right\} \quad n = 1, 2, \dots, N. \tag{7.77}$$

This can be done in parallel – hence the name "parallel shooting."

The procedure may be summarized schematically as in Fig. 7.3. Alternatively, if N is even, one may shoot both forward and backward, as indicated in Fig. 7.4 for $N = 4$. This reduces the size of the big system by one-half.

Even though multiple shooting can be quite effective, there are many practical problems associated with it. Perhaps the major problems are related to obtaining good initial approximations (recall, we have to choose a reasonable vector $\boldsymbol{S}$ in (7.73)) and to constructing a natural subdivision (7.65). With regard to the latter, suppose we have some rough approximation $\boldsymbol{\eta}(x) \approx \boldsymbol{y}(x)$ on $a \le x \le b$.

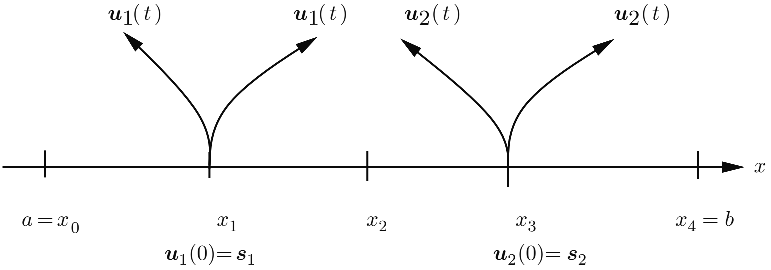


Fig. 7.4 Parallel shooting (two-sided), $N = 4$

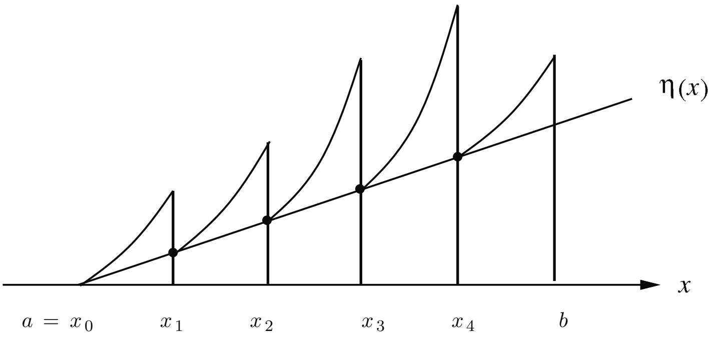


Fig. 7.5 Construction of subdivision

Then, taking $x_0 = a$, we may construct $x_1, x_2, \ldots$ recursively as follows. For $i = 0, 1, 2, \ldots$ solve the initial value problem

$$\frac{\mathrm{d}\boldsymbol{z}_i}{\mathrm{d}x} = \boldsymbol{f}(x, \boldsymbol{z}_i), \quad \boldsymbol{z}_i(x_i) = \boldsymbol{\eta}(x_i), \quad x \geq x_i, \tag{7.78}$$

and take for x_{i+1} the smallest $x > x_i$ such that (say) $\|\boldsymbol{z}_i(x)\| \geq 2\|\boldsymbol{\eta}(x)\|$. In other words, we do not allow the solution of (7.78) to increase more than twice in size; see Fig. 7.5. Thus, (7.78) are strictly auxiliary integrations whose sole purpose is to produce an appropriate subdivision of $[a, b]$.

There are circumstances in which reasonable initial approximations may be readily available, for example, if one solves the given boundary value problem (7.66) by a homotopy method. Basically, this means that the problem is embedded in a family of problems,

$$P_\omega : \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}x} = \boldsymbol{f}_\omega(x, \boldsymbol{y}), \quad a \leq x \leq b; \quad \boldsymbol{A}\boldsymbol{y}(a) + \boldsymbol{B}\boldsymbol{y}(b) = \boldsymbol{\gamma},$$

where ω is a (usually physically meaningful) parameter, say, in the interval $0 \leq \omega \leq 1$. This is done in such a way that for $\omega = 0$ the solution of P_ω is easy, and for $\omega = 1$, we have $\boldsymbol{f}_\omega(x, \boldsymbol{y}) = \boldsymbol{f}(x, \boldsymbol{y})$. One then solves a sequence of boundary value problems P_{ω_μ} corresponding to parameter values $0 = \omega_0 < \omega_1 < \omega_2 < \cdots < \omega_m = 1$ which are chosen sufficiently close to one another so that the solution of P_{ω_μ} differs relatively little from the solution of $P_{\omega_{\mu-1}}$. When $\omega = \omega_1$, one takes for $\boldsymbol{\eta}(x)$ the easy solution of P_{ω_0}, constructs the appropriate subdivision as before, and then solves P_{ω_1} by parallel shooting based on the subdivision generated. One next solves P_{ω_2}, using for $\boldsymbol{\eta}(x)$ the solution of P_{ω_1}, and proceeds as with P_{ω_1}. Continuing in this way, we will eventually have solved $P_{\omega_m} = P_1$, the given problem (7.66). Although the procedure is rather labor-intensive, it has the potential of providing accurate solutions to very difficult problems.

7.3 Finite Difference Methods

A more static approach toward solving boundary value problems is via direct discretization. One puts a grid on the interval of interest, replaces derivatives by finite difference expressions, and requires the discrete version of the problem to hold at all interior grid points. This gives rise to a system of linear or nonlinear equations for the unknown values of the solution at the grid points.

We consider and analyze only the simplest finite difference schemes. We assume throughout a uniform grid, say,

$$a = x_0 < x_1 < x_2 < \cdots < x_N < x_{N+1} = b, x_n = a + nh, h = \frac{b-a}{N+1}, \tag{7.79}$$

and we continue to use the terminology of grid functions introduced in Chap. 5, Sect. 5.7.

7.3.1 *Linear Second-Order Equations*

We consider the Sturm–Liouville problem (cf. (7.32) and (7.33))

$$\mathcal{L}y = r(x), \quad a \le x \le b, \tag{7.80}$$

where

$$\mathcal{L}y := -y'' + p(x)y' + q(x)y, \tag{7.81}$$

with the simplest boundary conditions

$$y(a) = \alpha, \quad y(b) = \beta. \tag{7.82}$$

If p, q, r are continuous and q positive on $[a,b]$, then (7.80) and (7.82) by Corollary 7.1.1 have a unique solution. Under these assumptions, there are positive constants $\overline{p}$, $\underline{q}$, and $\overline{q}$ such that

$$|p(x)| \le \overline{p}, \;\; 0 < \underline{q} \le q(x) \le \overline{q} \;\text{ for }\; a \le x \le b. \tag{7.83}$$

A simple finite difference operator, acting on a grid function $u \in \Gamma_h[a,b]$, which approximates the operator $\mathcal{L}$ in (7.81) is

$$\begin{aligned}(\mathcal{L}_h u)_n &= -\frac{u_{n+1} - 2u_n + u_{n-1}}{h^2} \; + \; p(x_n)\,\frac{u_{n+1} - u_{n-1}}{2h} \; + q(x_n)u_n,\\ n &= 1,2,\ldots,N.\end{aligned} \tag{7.84}$$

For any smooth function v on $[a,b]$, we define the grid function of the *truncation error* T_h by

$$(T_h v)_n = (\mathcal{L}_h v)_n - (\mathcal{L}v)(x_n), \;\; n = 1,2,\ldots,N. \tag{7.85}$$

If $v = y$ is the exact solution of (7.80) and (7.82), this reduces to an earlier definition in Chap. 6, Sect. 6.1.2. By Taylor's formula one easily finds that for $v \in C^4[a,b]$,

$$(T_h v)_n = -\frac{h^2}{12}\,[v^{(4)}(\xi_1) - 2p(x_n)v'''(\xi_2)], \;\; \xi_1, \xi_2 \in [x_n - h, x_n + h], \tag{7.86}$$

and more precisely, if $v \in C^6[a,b]$, since $\mathcal{L}_h$ is an even function of h,

$$(T_h v)_n = -\frac{h^2}{12}\,[v^{(4)}(x_n) - 2p(x_n)v'''(x_n)] + O(h^4), \;\; h \to 0. \tag{7.87}$$

In analogy to terminology introduced in Chap. 5, we call the difference operator $\mathcal{L}_h$ *stable* if there exists a constant M independent of h such that for h sufficiently small, one has for any grid function $v = \{v_n\}$

$$\|v\|_\infty \le M\{\max(|v_0|, |v_{N+1}|) + \|\mathcal{L}_h v\|_\infty\}, \;\; v \in \Gamma_h[a,b], \tag{7.88}$$

where $\|v\|_\infty = \max_{0\le n\le N+1} |v_n|$ and $\|\mathcal{L}_h v\|_\infty = \max_{1\le n\le N} |(\mathcal{L}_h v)_n|$. The following theorem gives a sufficient condition for stability.

Theorem 7.3.1. *If $h\overline{p} \le 2$, then $\mathcal{L}_h$ is stable. Indeed, (7.88) holds for $M = \max(1, 1/\underline{q})$. (Here $\overline{p}, \underline{q}$ are the constants defined in (7.83).)*

Proof. From (7.84) one computes

$$\tfrac{1}{2}h^2\,(\mathcal{L}_h v)_n = a_n v_{n-1} + b_n v_n + c_n v_{n+1}, \tag{7.89}$$

where

$$\begin{aligned} a_n &= -\frac{1}{2}\left[1 + \frac{1}{2}\,hp(x_n)\right], \\ b_n &= 1 + \frac{1}{2}\,h^2 q(x_n), \\ c_n &= -\frac{1}{2}\left[1 - \frac{1}{2}\,hp(x_n)\right]. \end{aligned} \tag{7.90}$$

Since, by assumption, $\frac{1}{2}\,h|p(x_n)| \le \frac{1}{2}\,h\overline{p} \le 1$, we have $a_n \le 0$, $c_n \le 0$, and

$$|a_n| + |c_n| = \frac{1}{2}\left[1 + \frac{1}{2}\,hp(x_n)\right] + \frac{1}{2}\left[1 - \frac{1}{2}\,hp(x_n)\right] = 1. \tag{7.91}$$

Also,

$$b_n \ge 1 + \tfrac{1}{2}\,h^2\underline{q}. \tag{7.92}$$

Now by (7.89), we have

$$b_n v_n = -a_n v_{n-1} - c_n v_{n+1} + \frac{1}{2}\,h^2(\mathcal{L}_h v)_n,$$

which, upon taking absolute values and using (7.91) and (7.92), yields

$$\left(1 + \frac{1}{2}\,h^2\underline{q}\right)|v_n| \le \|v\|_\infty + \frac{1}{2}\,h^2\|\mathcal{L}_h v\|_\infty, \quad n = 1, 2, \ldots, N. \tag{7.93}$$

We distinguish two cases.

Case I: $\|v\|_\infty = |v_{n_0}|$, $1 \le n_0 \le N$. Here (7.93) gives

$$\left(1 + \frac{1}{2}\,h^2\underline{q}\right)|v_{n_0}| \le |v_{n_0}| + \frac{1}{2}\,h^2\|\mathcal{L}_h v\|_\infty;$$

hence

$$|v_{n_0}| \le \frac{1}{\underline{q}}\,\|\mathcal{L}_h v\|_\infty\ ,$$

and (7.88) follows since by assumption $\frac{1}{\underline{q}} \le M$.

Case II: $\|v\|_\infty = |v_{n_0}|$, $n_0 = 0$ or $n_0 = N + 1$. In this case, (7.88) is trivial, since $M \ge 1$. □

The method of finite differences now consists of replacing (7.80) and (7.82) by

$$(\mathcal{L}_h u)_n = r(x_n), \quad n = 1, 2, \ldots, N; \quad u_0 = \alpha, \quad u_{N+1} = \beta. \tag{7.94}$$

In view of (7.89), this is the same as

$$a_n u_{n-1} + b_n u_n + c_n u_{n+1} = \frac{1}{2} h^2 r(x_n), \;\; n = 1, 2, \dots, N, u_0 = \alpha, \quad u_{N+1} = \beta,$$

and gives rise to the system of linear equations

$$\begin{bmatrix} b_1 & c_1 & & & 0 \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{N-1} & b_{N-1} & c_{N-1} \\ 0 & & & a_N & b_N \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \\ u_N \end{bmatrix} = \frac{1}{2} h^2 \begin{bmatrix} r(x_1) \\ r(x_2) \\ \vdots \\ r(x_{N-1}) \\ r(x_N) \end{bmatrix} - \begin{bmatrix} a_1\alpha \\ 0 \\ \vdots \\ 0 \\ c_N\beta \end{bmatrix}. \quad (7.95)$$

The matrix of the system is tridiagonal and strictly diagonally dominant, since $|a_n| + |c_n| = 1$ and $b_n > 1$ by (7.92). In particular, it is nonsingular, so that the system (7.95) has a unique solution. (Uniqueness follows also from the stability of $\mathcal{L}_h$: the homogeneous system with $r(x_n) = \alpha = \beta = 0$ can have only the trivial solution, since $\mathcal{L}_h u = 0$, $u_0 = u_{N+1} = 0$ implies $\|u\|_\infty = 0$ by (7.88).)

Now that we know that the difference method defines a unique approximation, the next question is: how good is it? An answer to that is given in the next two theorems.

Theorem 7.3.2. *If* $h\overline{p} \le 2$, *then*

$$\|u - y\|_\infty \le M \|T_h y\|_\infty, \;\; M = \max(1, 1/\underline{q}), \quad (7.96)$$

where $u = \{u_n\}$ *is the solution of (7.95),* $y = \{y_n\}$ *the grid function induced by the exact solution* $y(x)$ *of (7.80) and (7.82), and* $T_h y$ *the grid function of the truncation errors defined in (7.85), where* $v = y$. *If* $y \in C^4[a, b]$, *then*

$$\|u - y\|_\infty \le \tfrac{1}{12} h^2 M (\|y^{(4)}\|_\infty + 2\,\overline{p}\,\|y^{(3)}\|_\infty), \quad (7.97)$$

where $\|y^{(k)}\|_\infty = \max_{a \le x \le b} |y^{(k)}(x)|$, $k = 3, 4$.

Proof. From

$$\begin{aligned} (\mathcal{L}_h u)_n &= r(x_n), \quad u_0 = \alpha, \;\; u_{N+1} = \beta, \\ (\mathcal{L} y)(x_n) &= r(x_n), \quad y(x_0) = \alpha, \;\; y(x_{N+1}) = \beta, \end{aligned}$$

we obtain, letting $v_n = u_n - y(x_n)$,

$$\begin{aligned} (\mathcal{L}_h v)_n &= (\mathcal{L}_h u)_n - (\mathcal{L}_h y)_n \\ &= r(x_n) - [(\mathcal{L} y)(x_n) + (\mathcal{L}_h y)_n - (\mathcal{L} y)(x_n)] \\ &= r(x_n) - r(x_n) - (T_h y)_n \\ &= -(T_h y)_n, \end{aligned}$$

so that

$$\|\mathcal{L}_h v\|_\infty = \|T_h y\|_\infty. \tag{7.98}$$

By Theorem 7.3.1, $\mathcal{L}_h$ is stable with the stability constant M as defined in (7.96). Since $v_0 = v_{N+1} = 0$, there follows $\|v\|_\infty \le M\|\mathcal{L}_h v\|_\infty$, which in view of (7.98) and the definition of v is (7.96). The second assertion (7.97) follows directly from (7.86). □

In the spirit of Chap. 5, Sect. 5.7.3 and Chap. 6, Sect. 6.3.4, the result (7.97) of Theorem 7.3.2 can be refined as follows.

Theorem 7.3.3. *Let $p, q \in C^2[a,b]$, $y \in C^6[a,b]$, and $h\overline{p} \le 2$. Then*

$$u_n - y(x_n) = h^2 e(x_n) + O(h^4), \quad n = 0, 1, \ldots, N+1, \tag{7.99}$$

where $e(x)$ is the solution of

$$\mathcal{L}e = \theta(x), \quad a \le x \le b; \quad e(a) = 0, \quad e(b) = 0, \tag{7.100}$$

with

$$\theta(x) = \frac{1}{12}\,[y^{(4)}(x) - 2p(x)y'''(x)]. \tag{7.101}$$

Proof. We first note that our assumptions are such that $\theta \in C^2[a,b]$, which, by (7.100), implies that $e \in C^4[a,b]$.

Let

$$\overset{\circ}{v}_n = \frac{1}{h^2}\,(u_n - y(x_n)).$$

We want to show that

$$\overset{\circ}{v}_n = e(x_n) + O(h^2). \tag{7.102}$$

As in the proof of Theorem 7.3.2, we have

$$(\mathcal{L}_h \overset{\circ}{v})_n = -\frac{1}{h^2}\,(T_h y)_n.$$

By (7.87) with $v = y$, this gives

$$(\mathcal{L}_h \overset{\circ}{v})_n = \theta(x_n) + O(h^2). \tag{7.103}$$

Furthermore,

$$(\mathcal{L}_h e)_n = (\mathcal{L}e)(x_n) + (\mathcal{L}_h e)_n - (\mathcal{L}e)(x_n) = \theta(x_n) + (T_h e)_n, \tag{7.104}$$

by (7.100) and the definition (7.85) of truncation error. Since $e \in C^4[a,b]$, we have from (7.86) that

$$(T_h e)_n = O(h^2). \tag{7.105}$$

Subtracting (7.104) from (7.103), therefore, yields

$$(\mathcal{L}_h v)_n = O(h^2), \quad \text{where} \quad v_n = \overset{\circ}{v}_n - e(x_n).$$

Since $v_0 = v_{N+1} = 0$ and $\mathcal{L}_h$ is stable, by assumption, there follows from the stability inequality that $|v_n| \le M\|\mathcal{L}_h v\|_\infty = O(h^2)$, which is (7.102). □

Theorem 7.3.3 could be used as a basis for Richardson extrapolation (cf. Chap. 3, Sect. 3.2.7). Another application is the *method of difference correction* due to L. Fox. A "difference correction" is any quantity E_n such that

$$E_n = e(x_n) + O(h^2), \quad n = 1, 2, \ldots, N. \tag{7.106}$$

It then follows from (7.99) that

$$u_n - h^2 E_n = y(x_n) + O(h^4); \tag{7.107}$$

that is, $\hat{u}_n = u_n - h^2 E_n$ is an improved approximation having order of accuracy $O(h^4)$. Fox's idea is to construct a difference correction E_n by applying the basic difference method to the boundary value problem (7.100) in which $\theta(x_n)$ is replaced by a suitable difference approximation Θ_n:

$$(\mathcal{L}_h E)_n = \Theta_n, \;\; n = 1, 2, \ldots, N; \;\; E_0 = 0, \;\; E_{N+1} = 0. \tag{7.108}$$

Letting $v_n = E_n - e(x_n)$, we then find

$$(\mathcal{L}_h v)_n = (\mathcal{L}_h E)_n - (\mathcal{L}_h e)_n = \Theta_n - \theta(x_n) + O(h^2),$$

by virtue of (7.108), (7.104), and (7.105). Since $v_0 = v_{N+1} = 0$, stability then yields

$$|v_n| = |E_n - e(x_n)| \le M\|\Theta - \theta\|_\infty + O(h^2),$$

so that for (7.106) to hold, all we need is to make sure that

$$\Theta_n - \theta(x_n) = O(h^2), \quad n = 1, 2, \ldots, N. \tag{7.109}$$

This can be achieved by replacing the derivatives on the right of (7.101) by suitable finite difference approximations (see Ex. 10).

7.3.2 Nonlinear Second-Order Equations

A natural nonlinear extension of the linear problem (7.80) and (7.82) is

$$\mathcal{K}y = 0, \quad y(a) = \alpha, \quad y(b) = \beta, \tag{7.110}$$

where

$$\mathcal{K}y := -y'' + f(x, y, y') \tag{7.111}$$

and $f(x, y, z)$ is a given function of class C^1 defined on $[a, b]\times\mathbb{R}\times\mathbb{R}$, now assumed to be nonlinear in y and/or z. In analogy to (7.83) we make the assumption that

$$|f_z| \le \overline{p}, \quad 0 < \underline{q} \le f_y \le \overline{q} \quad \text{on } [a, b] \times \mathbb{R} \times \mathbb{R}. \tag{7.112}$$

Then, by Theorem 7.1.2, the problem (7.110) has a unique solution.

We use again the simplest difference approximation $\mathcal{K}_h$ to $\mathcal{K}$,

$$(\mathcal{K}_h u)_n = -\frac{u_{n+1} - 2u_n + u_{n-1}}{h^2} + f\left(x_n, u_n, \frac{u_{n+1} - u_{n-1}}{2h}\right), \tag{7.113}$$

and define the truncation error as before by

$$(T_h v)_n = (\mathcal{K}_h v)_n - (\mathcal{K}v)(x_n), \quad n = 1, 2, \ldots, N, \tag{7.114}$$

for any smooth function v on $[a, b]$. If $v \in C^4[a, b]$, then by Taylor's theorem, applied at $x = x_n$, $y = v(x_n)$, $z = v'(x_n)$,

$$\begin{aligned}(T_h v)_n &= -\left[\frac{v(x_n + h) - 2v(x_n) + v(x_n - h)}{h^2} - v''(x_n)\right] \\ &\quad + f\left(x_n, v(x_n), \frac{v(x_n + h) - v(x_n - h)}{2h}\right) - f(x_n, v(x_n), v'(x_n)) \\ &= -\frac{h^2}{12}\, v^{(4)}(\xi_1) + f_z(x_n, v(x_n), \overline{z}_n)\left[\frac{v(x_n + h) - v(x_n - h)}{2h} - v'(x_n)\right] \\ &= -\frac{h^2}{12}\, v^{(4)}(\xi_1) + f_z(x_n, v(x_n), \overline{z}_n)\, \frac{h^2}{6}\, v'''(\xi_2),\end{aligned}$$

where $\xi_i \in [x_n - h, x_n + h]$, $i = 1, 2$, and $\overline{z}_n$ is between $v'(x_n)$ and $(2h)^{-1}[v(x_n + h) - v(x_n - h)]$. Thus,

$$(T_h v)_n = -\frac{h^2}{12}\,[v^{(4)}(\xi_1) - 2f_z(x_n, v(x_n), \overline{z}_n)v'''(\xi_2)]. \tag{7.115}$$

Since $\mathcal{K}_h$ is nonlinear, the definition of stability needs to be slightly modified. We call $\mathcal{K}_h$ *stable* if for h sufficiently small, and for any two grid functions $v = \{v_n\}$, $w = \{w_n\}$, there is a constant M such that

$$\|v - w\|_\infty \le M\{\max(|v_0 - w_0|,\ |v_{N+1} - w_{N+1}|) + \|\mathcal{K}_h v - \mathcal{K}_h w\|_\infty\},$$
$$v, w \in \Gamma_h[a, b]. \tag{7.116}$$

If $\mathcal{K}_h$ is linear, this reduces to the previous definition (7.88), since $v - w$, just like v, is an arbitrary grid function.

Theorem 7.3.4. *If $h\overline{p} \le 2$, then $\mathcal{K}_h$ is stable. Indeed, (7.116) holds with $M =$* max *(1,1/$\underline{q}$).* (Here, $\overline{p}$, $\underline{q}$ are the constants defined in (7.112).)

Proof. Much the same as the proof of Theorem 7.3.1 See Ex. 11. □

The method of finite differences now takes on the following form:

$$(\mathcal{K}_h u)_n = 0, \quad n = 1, 2, \ldots, N; \quad u_0 = \alpha, \quad u_{N+1} = \beta. \tag{7.117}$$

This is a system of N nonlinear equations in the N unknowns $u_1, u_2, \ldots, u_N$. We show shortly that under the assumptions made, the system (7.117) has a unique solution. Its error can be estimated exactly as in Theorem 7.3.2 Indeed, by applying the stability inequality (7.116) with $v = u$ and $w = y$, where u is the grid function satisfying (7.117) and y the grid function induced by the exact solution $y(x)$ of (7.110), we get, with M as defined in Theorem 7.3.4,

$$\begin{aligned}\|u - y\|_\infty &\le M\|\mathcal{K}_h u - \mathcal{K}_h y\|_\infty = M\|\mathcal{K}_h y\|_\infty\\ &= M\|\mathcal{K}y + (\mathcal{K}_h y - \mathcal{K}y)\|_\infty\\ &= M\|\mathcal{K}_h y - \mathcal{K}y\|_\infty\\ &= M\|T_h y\|_\infty,\end{aligned}$$

which is (7.96). The same error estimate as in (7.97) then again follows immediately from (7.115) and the first assumption in (7.112).

In order to show that (7.117) has a unique solution, we write the system in fixed point form and apply the contraction mapping principle. It is convenient to introduce a parameter ω in the process – a "relaxation parameter" as it were – by writing (7.117) equivalently in the form

$$u = \boldsymbol{g}(u),\ \ \boldsymbol{g}(u) = u - \frac{1}{1+\omega}\,\frac{1}{2}h^2\,\mathcal{K}_h u \quad (\omega \ne -1),$$
$$u_0 = \alpha,\ \ u_{N+1} = \beta. \tag{7.118}$$

Here we think of $\boldsymbol{g}$ as a mapping $\mathbb{R}^{N+2} \to \mathbb{R}^{N+2}$, by defining $g_0(u) = \alpha$, $g_{N+1}(u) = \beta$. We want to show that $\boldsymbol{g}$ is a contraction map on $\mathbb{R}^{N+2}$ if h satisfies the condition of Theorem 7.3.4 and ω is suitably chosen. This then will prove (cf. Theorem 4.9.1) the existence and uniqueness of the solution of (7.118), and hence of (7.117).

Given any two grid functions $v = \{v_n\}$, $w = \{w_n\}$, we can write by Taylor's theorem, after a simple calculation,

$$g_n(v) - g_n(w) = \frac{1}{1+\omega}\,[a_n(v_{n-1} - w_{n-1}) + (1 + \omega - b_n)(v_n - w_n) \\ + c_n(v_{n+1} - w_{n+1})], \quad 1 \le n \le N, \qquad (7.119)$$

whereas for $n = 0$ or $n = N + 1$ the difference on the left, of course, is zero; here

$$a_n = \frac{1}{2}\left[1 + \frac{1}{2}\,hf_z(z_n, \overline{y}_n, \overline{z}_n)\right],$$

$$b_n = 1 + \frac{1}{2}\,h^2 f_y(x_n, \overline{y}_n, \overline{z}_n),$$

$$c_n = \frac{1}{2}\left[1 - \frac{1}{2}\,hf_z(x_n, \overline{y}_n, \overline{z}_n)\right],$$

with $\overline{y}_n$, $\overline{z}_n$ suitable intermediate values. Since $h\overline{p} \le 2$, we have

$$a_n \ge 0, \quad c_n \ge 0, \quad a_n + c_n = 1. \qquad (7.120)$$

Assuming, furthermore, that

$$\omega \ge \frac{1}{2}\,h^2\overline{q}, \qquad (7.121)$$

we have

$$1 + \omega - b_n \ge 1 + \omega - \left(1 + \frac{1}{2}\,h^2\overline{q}\right) = \omega - \frac{1}{2}\,h^2\overline{q} \ge 0.$$

Hence, all coefficients on the right of (7.119) are nonnegative, and since

$$0 \le 1 + \omega - b_n \le 1 + \omega - \left(1 + \frac{1}{2}\,h^2\underline{q}\right) = \omega - \frac{1}{2}\,h^2\underline{q},$$

we obtain, upon taking norms and noting (7.120),

$$|g_n(v) - g_n(w)| \le \frac{1}{1+\omega}\left(a_n + \omega - \frac{1}{2}\,h^2\underline{q} + c_n\right)\|v - w\|_\infty \\ = \frac{1}{1+\omega}\left(1 + \omega - \frac{1}{2}\,h^2\underline{q}\right)\|v - w\|_\infty;$$

that is,

$$\|\boldsymbol{g}(v) - \boldsymbol{g}(w)\|_\infty \le \gamma(\omega)\, \|v - w\|_\infty, \;\; \gamma(\omega) := 1 - \frac{\frac{1}{2}\, h^2 \underline{q}}{1+\omega}\,. \tag{7.122}$$

Clearly, $\gamma(\omega) < 1$, showing that $\boldsymbol{g}$ is a contraction map on $\mathbb{R}^{N+2}$, as claimed.

In principle, one could apply the method of successive iteration to (7.118), and it would converge for arbitrary initial approximation. Faster convergence can be expected, however, by applying Newton's method directly on (7.117); for details, see Ex. 12.

7.4 Variational Methods

Variational methods take advantage of the fact that the solution of important types of boundary value problems satisfies certain extremal properties. This then suggests solving the respective extremal problems – at least approximately – in place of the boundary value problems. This can be done by classical methods. We illustrate the method for a linear second-order boundary value problem with simplified (Dirichlet) boundary conditions.

7.4.1 Variational Formulation

Without restriction of generality (cf. (7.35)), we can assume that the problem is in self-adjoint form:

$$\mathcal{L}y = r(x), \;\; a \le x \le b; \;\; y(a) = \alpha, \;\; y(b) = \beta, \tag{7.123}$$

where

$$\mathcal{L}y := -\frac{\mathrm{d}}{\mathrm{d}x}\left(p(x)\,\frac{\mathrm{d}y}{\mathrm{d}x}\right) + q(x)y, \;\; a \le x \le b. \tag{7.124}$$

We assume $p \in C^1[a,b]$ and q, r continuous on $[a,b]$, and

$$p(x) \ge \underline{p} > 0, \;\; q(x) > 0 \;\text{ on }\; [a,b]. \tag{7.125}$$

Under these assumptions, the problem (7.123) has a unique solution (cf. Corollary 7.1.1).

If $\ell(x)$ is a linear function having the same boundary values as y in (7.123), then $z(x) = y(x) - \ell(x)$ satisfies $\mathcal{L}z = r(x) - (\mathcal{L}\ell)(x)$, $z(a) = z(b) = 0$, which is a problem of the same type as (7.123), but with *homogeneous* boundary conditions. We may assume, therefore, that $\alpha = \beta = 0$, and thus consider

$$\mathcal{L}y = r(x), \;\; a \leq x \leq b; \;\; y(a) = y(b) = 0. \tag{7.126}$$

Denoting by $C_0^2[a,b]$ the linear space $C_0^2[a,b] = \{u \in C^2[a,b] \colon u(a) = u(b) = 0\}$, we may write (7.126) in operator form as

$$\mathcal{L}y = r, \;\; y \in C_0^2[a,b]. \tag{7.127}$$

Note that $\mathcal{L} : C^2[a,b] \to C[a,b]$ is a linear operator. It is convenient to enlarge the space C_0^2 somewhat and define

$$\begin{aligned} V_0 = \{v \in C[a,b] : \;\; & v' \text{ piecewise continuous} \\ & \text{and bounded on } [a,b], v(a) = v(b) = 0\}. \end{aligned}$$

On V_0 we can define the usual inner product

$$(u,v) := \int_a^b u(x)v(x)\mathrm{d}x, \;\; u,v \in V_0. \tag{7.128}$$

Theorem 7.4.1. *The operator $\mathcal{L}$ in (7.124) is symmetric on $C_0^2[a,b]$ relative to the inner product (7.128); that is,*

$$(\mathcal{L}u, v) = (u, \mathcal{L}v), \;\; \textit{all} \;\; u,v \in C_0^2[a,b]. \tag{7.129}$$

Proof. Use integration by parts to obtain

$$\begin{aligned} (\mathcal{L}u, v) &= \int_a^b [-(p(x)u')' + q(x)u]v(x)\mathrm{d}x \\ &= -(pu'v)\big|_a^b + \int_a^b [p(x)u'(x)v'(x) + q(x)u(x)v(x)]\mathrm{d}x \\ &= \int_a^b [p(x)u'(x)v'(x) + q(x)u(x)v(x)]\mathrm{d}x. \end{aligned}$$

Since the last integral is symmetric in u and v, it is also equal to $(\mathcal{L}v, u)$, which in turn, by the symmetry of $(\cdot\,,\cdot)$, proves the theorem. □

Note that the last integral in the proof of Theorem 7.4.1 is defined not only on $C_0^2[a,b]$, but also on V_0. It suggests an alternative inner product,

$$[u,v] := \int_a^b [p(x)u'(x)v'(x) + q(x)u(x)v(x)]\mathrm{d}x, \;\; u,v \in V_0, \tag{7.130}$$

and the proof of Theorem 7.4.1 shows that

$$(\mathcal{L}u, v) = [u, v] \text{ if } u \in C_0^2[a, b], \quad v \in V_0. \tag{7.131}$$

In particular, if $u = y$ is a solution of (7.126), then

$$[y, v] = (r, v), \text{ all } v \in V_0; \tag{7.132}$$

this is the *variational*, or *weak*, form of (7.126).

Theorem 7.4.2. *Under the assumptions made on* p, q*, and* r (cf. (7.125))*, there exist positive constants* $\underline{c}$ *and* $\overline{c}$ *such that*

$$\underline{c}\, \|u\|_\infty^2 \le [u, u] \le \overline{c}\, \|u'\|_\infty^2, \quad \textit{all } u \in V_0. \tag{7.133}$$

In fact,

$$\underline{c} = \frac{\underline{p}}{b-a}, \quad \overline{c} = (b-a)\|p\|_\infty + (b-a)^3\|q\|_\infty. \tag{7.134}$$

Proof. For any $u \in V_0$, since $u(a) = 0$, we have

$$u(x) = \int_a^x u'(t)\mathrm{d}t, \quad x \in [a, b].$$

By Schwarz's inequality,

$$u^2(x) \le \int_a^x 1\mathrm{d}t \cdot \int_a^x [u'(t)]^2\mathrm{d}t \le (b-a)\int_a^b [u'(t)]^2\mathrm{d}t, \quad x \in [a, b],$$

and, therefore,

$$\|u\|_\infty^2 \le (b-a)\int_a^b [u'(t)]^2\mathrm{d}t \le (b-a)^2\|u'\|_\infty^2. \tag{7.135}$$

Using the assumption (7.125), we get

$$[u, u] = \int_a^b (p(x)[u'(x)]^2 + q(x)u^2(x))\mathrm{d}x \ge \underline{p}\int_a^b [u'(x)]^2\mathrm{d}x \ge \frac{\underline{p}}{b-a}\, \|u\|_\infty^2,$$

where the last inequality follows from the left inequality in (7.135). This proves the lower bound in (7.133). The upper bound is obtained by observing that

$$[u, u] \le (b-a)\|p\|_\infty\, \|u'\|_\infty^2 + (b-a)\|q\|_\infty\|u\|_\infty^2 \le \overline{c}\, \|u'\|_\infty^2,$$

where (7.135) has been used in the last step. □

We remark that (7.133) implies the uniqueness of solutions of (7.126). In fact, if

$$\mathcal{L}y = r, \quad \mathcal{L}y^* = r, \quad y, y^* \in C_0^2[a,b],$$

then $\mathcal{L}(y - y^*) = 0$, hence, by (7.131) and (7.133),

$$0 = (\mathcal{L}(y - y^*), y - y^*) = [y - y^*, y - y^*] \geq \underline{c}\, \|y - y^*\|_\infty^2,$$

and it follows that $y \equiv y^*$.

7.4.2 The Extremal Problem

We define the quadratic functional

$$F(u) := [u,u] - 2(r,u), \quad u \in V_0, \tag{7.136}$$

where r is the right-hand function in (7.126). The extremal property for the solution y of (7.127) is expressed in the following theorem.

Theorem 7.4.3. *Let y be the solution of (7.127). Then*

$$F(u) > F(y), \quad \textit{all} \;\; u \in V_0, \quad u \not\equiv y. \tag{7.137}$$

Proof. By (7.132), $(r,u) = [y,u]$, so that

$$\begin{aligned} F(u) &= [u,u] - 2(r,u) = [u,u] - 2[y,u] + [y,y] - [y,y] \\ &= [y-u, y-u] - [y,y] > -[y,y], \end{aligned}$$

where strict inequality holds in view of (7.133) and $y - u \not\equiv 0$. On the other hand, since $[y,y] = (\mathcal{L}y, y) = (r,y)$, by (7.131), we have

$$F(y) = [y,y] - 2(r,y) = (r,y) - 2(r,y) = -(r,y) = -[y,y],$$

which, combined with the previous inequality, proves the theorem. □

Theorem 7.4.3 thus expresses the following extremal property of the solution of (7.127):

$$F(y) = \min_{u \in V_0} F(u). \tag{7.138}$$

We view (7.138) as an extremal problem for determining y, and in the next section solve it approximately by determining a function u_S from a finite-dimensional subset $S \subset V_0$ that minimizes $F(u)$ on S. In this connection, it is useful to note the identity

$$[y-u, y-u] = F(u) + [y,y], \quad u \in V_0, \tag{7.139}$$

satisfied by the solution y, which was established in the course of the proof of Theorem 7.4.3

7.4.3 Approximate Solution of the Extremal Problem

Let $S \subset V_0$ be a finite-dimensional subspace of V_0 and $\dim S = n$ its dimension. Let $u_1, u_2, \dots, u_n$ be a basis of S, so that

$$u \in S \ \text{ if and only if } \ u = \sum_{\nu=1}^{n} \xi_\nu u_\nu, \quad \xi_\nu \in \mathbb{R}. \tag{7.140}$$

We approximate the solution y of (7.138) by $u_S \in S$, which satisfies

$$F(u_S) = \min_{u \in S} F(u). \tag{7.141}$$

Before we analyze the quality of the approximation $u_S \approx y$, let us explain the mechanics of the method.

We have, for any $u \in S$,

$$\begin{aligned} F(u) &= \left[\sum_{\nu=1}^{n} \xi_\nu u_\nu, \ \sum_{\mu=1}^{n} \xi_\mu u_\mu \right] - 2 \left(r, \sum_{\nu=1}^{n} \xi_\nu u_\nu \right) \\ &= \sum_{\nu,\mu=1}^{n} [u_\nu, u_\mu] \xi_\nu \xi_\mu - 2 \sum_{\nu=1}^{n} (r, u_\nu) \xi_\nu . \end{aligned}$$

Define

$$\boldsymbol{U} = \begin{bmatrix} [u_1, u_1] & [u_1, u_2] & \cdots & [u_1, u_n] \\ [u_2, u_1] & [u_2, u_2] & \cdots & [u_2, u_n] \\ \cdots & \cdots & \cdots & \cdots \\ [u_n, u_1] & [u_n, u_2] & \cdots & [u_n, u_n] \end{bmatrix}, \quad \boldsymbol{\xi} = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{bmatrix}, \quad \boldsymbol{\rho} = \begin{bmatrix} (r, u_1) \\ (r, u_2) \\ \vdots \\ (r, u_n) \end{bmatrix}. \tag{7.142}$$

In early applications of the method to structural mechanics, and ever since, the matrix $\boldsymbol{U}$ is called the *stiffness matrix*, and $\boldsymbol{\rho}$ the *load vector*. In terms of these, the functional F can be written in matrix form as

$$F(u) = \boldsymbol{\xi}^{\mathrm{T}} \boldsymbol{U} \boldsymbol{\xi} - 2\boldsymbol{\rho}^{\mathrm{T}} \boldsymbol{\xi}, \quad \boldsymbol{\xi} \in \mathbb{R}^n. \tag{7.143}$$

The matrix $\boldsymbol{U}$ is not only symmetric, but also positive definite, since $\boldsymbol{\xi}^T \boldsymbol{U} \boldsymbol{\xi} = [u, u] > 0$, unless $u \equiv 0$ (i.e., $\boldsymbol{\xi} = \boldsymbol{0}$).

Our approximate extremal problem (7.141) thus takes the form

$$\begin{aligned} &\phi(\boldsymbol{\xi}) = \min, \\ &\phi(\boldsymbol{\xi}) := \boldsymbol{\xi}^{\mathrm{T}} \boldsymbol{U} \boldsymbol{\xi} - 2\boldsymbol{\rho}^{\mathrm{T}} \boldsymbol{\xi}, \quad \boldsymbol{\xi} \in \mathbb{R}^n, \end{aligned} \tag{7.144}$$

an unconstrained quadratic minimization problem in $\mathbb{R}^n$. Since $\boldsymbol{U}$ is positive definite, the problem (7.144) has a unique solution $\hat{\boldsymbol{\xi}}$ given by the solution of the linear system

$$\boldsymbol{U}\boldsymbol{\xi} = \boldsymbol{\rho}. \tag{7.145}$$

It is easily verified that

$$\phi(\boldsymbol{\xi}) > \phi(\hat{\boldsymbol{\xi}}), \ \text{ all } \ \boldsymbol{\xi} \in \mathbb{R}^n, \ \ \boldsymbol{\xi} \neq \hat{\boldsymbol{\xi}}; \tag{7.146}$$

indeed, since $\boldsymbol{\rho} = \boldsymbol{U}\hat{\boldsymbol{\xi}}$, and thus $\boldsymbol{\rho}^{\mathrm{T}} = \hat{\boldsymbol{\xi}}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}} = \hat{\boldsymbol{\xi}}^{\mathrm{T}}\boldsymbol{U}$, we have

$$\begin{aligned} \phi(\boldsymbol{\xi}) &= \boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{\xi} - 2\boldsymbol{\rho}^{\mathrm{T}}\boldsymbol{\xi} = \boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{\xi} - 2\hat{\boldsymbol{\xi}}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{\xi} \\ &= \boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{\xi} - 2\hat{\boldsymbol{\xi}}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{\xi} + \hat{\boldsymbol{\xi}}^{\mathrm{T}}\boldsymbol{U}\hat{\boldsymbol{\xi}} - \hat{\boldsymbol{\xi}}^{\mathrm{T}}\boldsymbol{U}\hat{\boldsymbol{\xi}} \\ &= (\boldsymbol{\xi} - \hat{\boldsymbol{\xi}})^{\mathrm{T}}\boldsymbol{U}(\boldsymbol{\xi} - \hat{\boldsymbol{\xi}}) + \phi(\hat{\boldsymbol{\xi}}), \end{aligned}$$

where $-\hat{\boldsymbol{\xi}}^{\mathrm{T}}\boldsymbol{U}\hat{\boldsymbol{\xi}} = -\hat{\boldsymbol{\xi}}^{\mathrm{T}}\boldsymbol{\rho} = \hat{\boldsymbol{\xi}}^{\mathrm{T}}\boldsymbol{\rho} - 2\boldsymbol{\rho}^{\mathrm{T}}\hat{\boldsymbol{\xi}} = \hat{\boldsymbol{\xi}}^{\mathrm{T}}\boldsymbol{U}\hat{\boldsymbol{\xi}} - 2\boldsymbol{\rho}^{\mathrm{T}}\hat{\boldsymbol{\xi}} = \phi(\hat{\boldsymbol{\xi}})$ has been used in the last step. From this, (7.146) follows immediately. Thus,

$$u_S = \sum_{\nu=1}^{n} \hat{\xi}_\nu u_\nu, \quad \hat{\boldsymbol{\xi}} = \left[\hat{\xi}_1, \hat{\xi}_2, \ldots, \hat{\xi}_n\right]^{\mathrm{T}} \quad \text{where } \boldsymbol{U}\hat{\boldsymbol{\xi}} = \boldsymbol{\rho}. \tag{7.147}$$

In practice, the basis functions of S are chosen to have small support, which results in a matrix $\boldsymbol{U}$ having a band structure.

It is now straightforward to establish the *optimal approximation property* of u_S in the norm $[\,\cdot\,,\cdot\,]$; that is,

$$[y - u_S, y - u_S] = \min_{u\in S}[y - u, y - u]. \tag{7.148}$$

Indeed, by (7.139) and (7.141), the left-hand side is equal to $F(u_S) + [y, y] = \min_{u\in S}\{F(u) + [y, y]\}$, which, again by (7.139), equals the right-hand side of (7.148).

The approximation property (7.148) gives rise to the following error estimate.

Theorem 7.4.4. *There holds*

$$\|y - u_S\|_\infty \le \sqrt{\overline{c}/\underline{c}}\,\|y' - u'\|_\infty, \ \ \textit{all} \ \ u \in S, \tag{7.149}$$

where $\overline{c}$ and $\underline{c}$ are the constants defined in (7.134). In particular,

$$\|y - u_S\|_\infty \le \sqrt{\overline{c}/\underline{c}}\, \inf_{u\in S} \|y' - u'\|_\infty. \tag{7.150}$$

Proof. By (7.133) and (7.148), we have

$$\underline{c}\,\|y - u_S\|_\infty^2 \le [y - u_S, y - u_S] \le [y - u, y - u] \le \overline{c}\,\|y' - u'\|_\infty^2,$$

from which (7.149) follows. □

The point of Theorem 7.4.4 is that, in order to get a good error bound, we have to use an approximation process $y \approx u$, $u \in S$, which approximates the first derivative of y as well as possible. Note that this approximation process is *independent* of the one yielding u_S; its sole purpose is to provide a good error bound for u_S.

Example. Let Δ be a subdivision of $[a, b]$, say,

$$a = x_1 < x_2 < x_3 < \cdots < x_{n-1} < x_n = b, \tag{7.151}$$

and take (see Chap. 2, Sect. 2.3.4 for notation)

$$S = \{s \in \mathbb{S}_3^2(\Delta) : \;\; s(a) = s(b) = 0\}. \tag{7.152}$$

Here S is a subspace not only of V_0, but even of $C_0^2[a, b]$. Its dimension is easily seen to be n. Given the solution y of (7.127), there is a unique $s_{\mathrm{compl}} \in S$ such that

$$\begin{aligned} s_{\mathrm{compl}}(x_i) &= y(x_i), \quad i = 1, 2, \ldots, n, \\ s'_{\mathrm{compl}}(a) &= y'(a), \quad s'_{\mathrm{compl}}(b) = y'(b), \end{aligned} \tag{7.153}$$

the "complete cubic spline interpolant" to y (cf. Chap. 2, Sect. 2.3.4 (b.1)). From Chap. 2, (2.147), we know that

$$\|s'_{\mathrm{compl}} - y'\|_\infty \le \tfrac{1}{24}\, |\Delta|^3 \|y^{(4)}\|_\infty \;\text{ if }\; y \in C^4[a, b].$$

Combining this with the result of Theorem 7.4.4 (in which $u = s_{\mathrm{compl}}$), we get the error bound

$$\|y - u_S\|_\infty \le \tfrac{1}{24}\, \sqrt{\overline{c}/\underline{c}}\, |\Delta|^3 \|y^{(4)}\|_\infty \; = O(|\Delta|^3), \tag{7.154}$$

which is one order of magnitude better than the one for the ordinary finite difference method (cf. (7.97)). However, there is more work involved in computing the stiffness matrix (many integrals!), and also in solving the linear system (7.145). Even with a basis of S that has small support (extending over at most four consecutive subintervals of Δ), one still has to deal with a banded matrix $\boldsymbol{U}$ having bandwidth 7 (not 3, as in (7.95)).

7.5 Notes to Chapter 7

Background material on the theory of boundary value problems can be found in most textbooks on ordinary differential equations. Specialized texts are Bailey et al. [1968], Bernfeld and Lakshmikantham [1974], and Agarwal [1986]; all three, but especially the first, also contain topics on the numerical solution of boundary value problems and applications. An early book strictly devoted to numerical methods for solving boundary value problems is Fox [1990], an eminently practical account still noteworthy for its consistent use of "difference correction," that is, the incorporation of remainder terms into the solution process. Subsequent books and monographs are Keller [1992], Na [1979], and Ascher et al. [1995]. The books by Keller and Na complement each other in that the former gives a mathematically rigorous treatment of the major methods in use, with some applications provided in the final chapter, and the latter a more informal presentation intended for engineers, and containing a wealth of engineering applications. Na's book also discusses methods less familiar to numerical analysts, such as the МЕТОД ПРОГОНКИ developed by Russian scientists, here translated as the "method of chasing," and interesting methods based on transformation of variables. The book by Ascher et al. is currently the major reference work in this area. One of its special features is an extensive discussion of the numerical condition and associated condition numbers for boundary value problems. In its first chapter, it also contains a large sampling of boundary value problems occurring in real-life applications.

Sturm–Liouville eigenvalue problems – both regular and singular – and their numerical solution are given a thorough treatment in Pryce [1993]. A set of 60 test problems is included in one of the appendices, and references to available software in another.

Section 7.1.1. The third Example is from Bailey et al. [1968, Chap. 1, Sect. 4].

Section 7.1.2. The exposition in this section, in particular, the proof of Theorem 7.1.2, follows Keller [1992, Sect. 1.2].

Section 7.1.3. An example of an existence and uniqueness theorem for the general boundary value problem (7.38) is Theorem 1.2.6 in Keller [1992]. The remark at the end of this section can be generalized to "partially separated" boundary conditions, which give rise to a "reduced" superposition method; see Ascher et al. [1995, Sect. 4.2.4].

Section 7.2. In this section, we give only a bare outline and some of the key ideas involved in shooting methods. To make shooting a viable method, even for linear boundary value problems, requires attention to many practical and technical details. For these, we must refer to the relevant literature, for example, Roberts and Shipman [1972] or, especially, Ascher et al. [1995, Chap. 4]. The latter reference also contains two computer codes, one for linear, the other for nonlinear (nonstiff) boundary value problems.

Shooting basically consists of solving a finite-dimensional system of equations generated by solutions of initial value problems, which is then solved iteratively, for example, by Newton's method. Alternatively, one could apply Newton's method, or more precisely, the Newton–Kantorovich method, directly to the boundary value problem in question, considered as an operator equation in a Banach space of smooth functions on $[a, b]$ satisfying homogeneous boundary conditions. This is the method of quasilinearization originally proposed by Bellman and Kalaba [1965].

Yet another approach is "invariant imbedding," where the endpoint b of the interval $[a, b]$ is made a variable with respect to which one differentiates to obtain an auxiliary nonlinear initial value problem of the Riccati type. See, for example, Ascher et al. [1995, Sect. 4.5]. A different view of invariant imbedding based on a system of first-order partial differential equations and associated characteristics is developed in Meyer [1973].

Section 7.2.1. Instead of Newton's method (7.45) one could, of course, use other iterative methods for solving the equation $\phi(s) = 0$ in (7.25), for example, a fixed point iteration based on the equivalent equation $s = s - m\phi(s)$, $m \neq 0$, which is analyzed in Keller [1992, Sect. 2.2], or Steffensen's method (cf. Chap. 4, Sect. 4.6, (4.66)).

For the origin of the second Example in this section, see Troesch [1976]. The analytic solution of the associated initial value problem is given, for example, in Stoer and Bulirsch [2002, pp. 514–516]; also cf. Ex. 8(c),(d).

Section 7.2.2. It is relatively straightforward to analyze the effect, in superposition methods, of the errors committed in the numerical integration of the initial value problems involved; see, for example, Keller [1992, Sect. 2.1] and Ascher et al. [1995, Sect. 4.2.2]. More important, and potentially more disastrous, are the effects of rounding errors; see, for example, Ascher et al. [1995, Sect. 4.2.3].

In place of (7.52), other iterative methods could be used to solve the equation $\boldsymbol{\phi}(\boldsymbol{s}) = \boldsymbol{0}$ in (7.41), for example, one of the quasi-Newton methods (cf. Chap. 4, Notes to Sect. 4.9.2), or, as in the scalar case, a fixed point iteration based on $\boldsymbol{s} = \boldsymbol{s} - \boldsymbol{M}\boldsymbol{\phi}(\boldsymbol{s})$, with $\boldsymbol{M}$ a nonsingular matrix chosen such that the map $\boldsymbol{s} \mapsto \boldsymbol{s} - \boldsymbol{M}\boldsymbol{\phi}(\boldsymbol{s})$ is contractive.

The Example in this section is from Morrison et al. [1962], where the term "shooting" appears to have been used for the first time in the context of boundary value problems.

Section 7.2.3. Parallel shooting is important also for linear boundary value problems of the type (7.38), (7.42), since without it, numerical linear dependencies may be developing that could render the method of simple shooting useless. There are various versions of parallel shooting, some involving reorthogonalization of solution vectors; see Ascher et al. [1995, Sects. 4.3 and 4.4]. For a discussion of homotopy methods, including numerical examples, see Roberts and Shipman [1972, Chap. 7].

Sections 7.3.1 and 7.3.2. The treatment in these sections closely follows Keller [1992, Sects. 3.1 and 3.2]. Maintaining second-order accuracy on nonuniform grids is not entirely straightforward; see, for example, Ascher et al. [1995, Sect. 5.6.1].

Extensions of the method of finite differences to linear and nonlinear systems of the type (7.38) can be based on the local use of the trapezoidal or midpoint rule. This is discussed in Keller [1992, Sect. 3.3] and Ascher et al. [1995, Sects. 5.1 and 5.2]. Local use of implicit Runge–Kutta methods afford more accuracy, and so do the methods of extrapolation and "deferred corrections"; for these, see Ascher et al. [1995, Sects. 5.3, 5.4, 5.5.2 and 5.5.3].

Boundary value problems for single higher-order differential equations are often solved by collocation methods using spline functions; a discussion of this is given in Ascher et al. [1995, Sects. 5.6.2–5.6.4].

Section 7.4. The treatment of variational methods in this section is along the lines of Stoer and Bulirsch [2002, Sect. 7.5]. There are many related methods, collectively called projection methods, whose application to two-point boundary value problems and convergence analysis is the subject of a survey by Reddien [1980] containing extensive references to the literature.

Exercises and Machine Assignments to Chapter 7

Exercises

1. Consider the nonlinear boundary value problem (Blasius equation)

$$y''' + \tfrac{1}{2} y y'' = 0, \quad 0 \le x < \infty,$$
$$y(0) = y'(0) = 0, \;\; y'(\infty) = 1.$$

   (a) Letting $y''(0) = \lambda$ and $z(t) = \lambda^{-\frac{1}{3}} y(\lambda^{-\frac{1}{3}} t)$ (assuming $\lambda > 0$), derive an initial value problem for z on $0 \le t < \infty$.
   (b) Explain, and illustrate numerically and graphically, how the solution of the initial value problem in (a) can be used to obtain the solution $y(x)$ of the given boundary value problem.

2. The boundary value problem

$$y'' = -\frac{1}{x} y y', \;\; 0 < x \le 1; \quad y(0) = 0, \;\; y(1) = 1,$$

   although it has a singularity at $x = 0$ and certainly does not satisfy (7.112), has the smooth solution $y(x) = 2x/(1 + x)$.

   (a) Determine analytically the s-interval for which the initial value problem

$$u'' = -\frac{1}{x} u u', \;\; 0 < x \le 1; \quad u(0) = 0, \;\; u'(0) = s$$

   has a smooth solution $u(x; s)$ on $0 \le x \le 1$.
   (b) Determine the s-interval for which Newton's method applied to $u(1; s) - 1 = 0$ converges.

3. Use Matlab to reproduce the results in Table 7.2 and to prepare plots of the four solution components.
4. Derive (7.56) and (7.62).
5. Let

$$\phi(s) = \frac{s}{s \cosh 1 - \sinh 1} - \mathrm{e}$$

and s^0 be the solution of

$$s^0 - \frac{\phi(s^0)}{\phi'(s^0)} = \tanh 1$$

(cf. the Example in Sect. 7.2.2, in particular (7.58), (7.62)).

(a) Show that $s^0 < \coth 1$. {*Hint*: consider what $s^0 < t^0$ means in terms of one Newton step at t^0 for the equation $\phi(s) = 0$.}
(b) Use the bisection method to compute s^0 to six decimal places. What are appropriate initial approximations?

6. Generalizing the Example in Sect. 7.2.2, let $\nu > 0$ and consider the boundary value problem

$$\left.\begin{aligned} \frac{dy_1}{dx} &= \frac{y_1^{\nu+1}}{y_2^{\nu}} \\ \frac{dy_2}{dx} &= \frac{y_2^{\nu+1}}{y_1^{\nu}} \end{aligned}\right\} \quad 0 \le x \le 1,$$

$$y_1(0) = 1, \quad y_1(1) = \mathrm{e}.$$

(a) Determine the exact solution.
(b) Solve the initial value problem

$$\frac{d}{dx}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} u_1^{\nu+1}/u_2^{\nu} \\ u_2^{\nu+1}/u_1^{\nu} \end{bmatrix}, \quad 0 \le x \le 1; \quad \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}(0) = \begin{bmatrix} 1 \\ s \end{bmatrix}$$

in closed form.
(c) Find the conditions on $s > 0$ guaranteeing that $u_1(x)$, $u_2(x)$ both remain positive and finite on [0, 1]. In particular, show that, as $\nu \to \infty$, the interval in which s must lie shrinks to the point $s = 1$. What happens when $\nu \to 0$?

7. Suppose the Example in Sect. 7.2.2 is modified by multiplying the right-hand sides of the differential equation by λ, and by replacing the second boundary condition by $y_1(1) = \mathrm{e}^{-\lambda}$, where $\lambda > 0$ is a large parameter.

(a) What is the exact solution?
(b) What are the conditions on s for the associated initial value problem to have positive and bounded solutions? What happens as $\lambda \to \infty$? As $\lambda \to 0$?

8. The Jacobian elliptic functions sn and cn are defined by

$$\mathrm{sn}(u|k) = \sin\varphi, \quad \mathrm{cn}(u|k) = \cos\varphi, \quad 0 < k < 1,$$

where φ is uniquely determined by

$$u = \int_0^{\varphi} \frac{d\theta}{(1 - k^2 \sin^2\theta)^{\frac{1}{2}}}.$$

(a) Show that $K(k) := \int_0^{\frac{1}{2}\pi} (1 - k^2 \sin^2 \theta)^{-\frac{1}{2}} \mathrm{d}\theta$ is the smallest positive zero of cn.

(b) Show that

$$\frac{\mathrm{d}}{\mathrm{d}u}\mathrm{sn}(u|k) = \mathrm{cn}(u|k)\sqrt{1 - k^2[\mathrm{sn}(u|k)]^2}\,,$$

$$\frac{\mathrm{d}}{\mathrm{d}u}\mathrm{cn}(u|k) = -\mathrm{sn}(u|k)\sqrt{1 - k^2[\mathrm{sn}(u|k)]^2}\,.$$

(c) Show that the initial value problem

$$y'' = \lambda \sinh(\lambda y), \;\; y(0) = 0, \; y'(0) = s \quad (|s| < 2)$$

has the exact solution

$$y(x; s) = \frac{2}{\lambda} \sinh^{-1}\left(\frac{s}{2}\,\frac{\mathrm{sn}(\lambda x|k)}{\mathrm{cn}(\lambda x|k)}\right), \;\; k^2 = 1 - \frac{s^2}{4}\,.$$

Hence show that $y(x; s)$, $x > 0$, becomes singular for the first time when $x = x_\infty$, where

$$x_\infty = \frac{K(k)}{\lambda}\,.$$

(d) From the known expansion (see Radon [1950], p. 76 and Sect. 7, (I_b))

$$K(k) = \ln \frac{4}{\sqrt{1 - k^2}} + \frac{1}{4}\left(\ln \frac{4}{\sqrt{1 - k^2}} - 1\right)(1 - k^2) + \cdots\,, \;\; k \to 1,$$

conclude that

$$x_\infty \sim \frac{1}{\lambda} \ln \frac{8}{|s|} \;\; \text{as } s \to 0.$$

9. It has been shown in the first Example of Sect. 7.2.1 that the boundary value problem

$$y'' + \mathrm{e}^{-y} = 0, \quad 0 \le x \le 1, \;\; y(0) = y(1) = 0$$

has a unique solution that is nonnegative on [0, 1].

(a) Set up a finite difference method for solving the problem numerically. (Use a uniform grid $x_n = \frac{n}{N+1}$, $n = 0, 1, \ldots, N + 1$, and the simplest of finite difference approximations to y''.)

(b) Write the equations for the approximate vector $\boldsymbol{u}^{\mathrm{T}} = [u_1, u_2, \ldots, u_N]$ in fixed point form $\boldsymbol{u} = \boldsymbol{\varphi}(\boldsymbol{u})$ and find a compact domain $\mathcal{D} \subset \mathbb{R}^N$ such that $\boldsymbol{\varphi} : \mathbb{R}^N \to \mathbb{R}^N$ maps $\mathcal{D}$ into $\mathcal{D}$ and is contractive in $\mathcal{D}$. {*Hint*: use the fact that the tridiagonal matrix $\boldsymbol{A} = \mathrm{tri}[1, -2, 1]$ has a nonpositive inverse $\boldsymbol{A}^{-1}$ satisfying $\|\boldsymbol{A}^{-1}\|_\infty \le \frac{1}{8}(N + 1)^2$.}

(c) Discuss the convergence of the fixed point iteration applied to the system of finite difference equations.

10. Given the grid $\{x_n\}_{n=0}^{N+1}$ of (7.79), construct finite difference approximations Θ_n for $\theta(x_n)$, where $\theta(x) = \frac{1}{12}[y^{(4)}(x) - 2p(x)y'''(x)]$, such that $\Theta_n - \theta(x_n) = O(h^2)$ (cf. (7.109)). {*Hint*: distinguish the cases $2 \leq n \leq N-1$ and $n = 1$ resp. $n = N$.}
11. Prove Theorem 7.3.4.
12. Describe the application of Newton's method for solving the nonlinear finite difference equations $\mathcal{K}_h u = 0$ of (7.117).
13. Let Δ be the subdivision

$$a = x_0 < x_1 < \cdots < x_n < x_{n+1} = b$$

and $S = \{s \in \mathbb{S}_1^0(\Delta) : \; s(a) = s(b) = 0\}$.

(a) With $[\cdot\,,\cdot]$ the inner product defined in (7.130), find an expression for $[u_\nu, u_\mu]$ in terms of the basis of hat functions (cf. Chap. 2, Ex. 72, but note the difference in notation) and in terms of the integrals involved; do this in a similar manner for $\rho_\nu = (r, u_\nu)$, where $(\cdot\,,\cdot)$ is the inner product defined in (7.128).

(b) Suppose that each integral is split into a sum of integrals over each subinterval of Δ and the trapezoidal rule is employed to approximate the values of the integrals. Obtain the resulting approximations for the stiffness matrix $\boldsymbol{U}$ and the load vector $\boldsymbol{\rho}$. Interpret the linear system (7.145) thus obtained as a finite difference method.

14. Apply the approximate variational method of Sect. 7.4.3 to the boundary value problem

$$-y'' = r(x), \;\; 0 \leq x \leq 1; \quad y(0) = y(1) = 0,$$

using for S a space of continuous piecewise quadratic functions. Specifically, take a uniform subdivision

$$\Delta: \quad 0 = x_0 < x_1 < x_2 < \cdots < x_{n-1} < x_n = 1, \quad x_\nu = \nu h,$$

of $[0, 1]$ into n subintervals of length $h = 1/n$ and let $S = \{s \in \mathbb{S}_2^0 : \; s(0) = s(1) = 0\}$.

(a) How many basis functions is S expected to have? Explain.

(b) Construct a basis for S. {*Hint*: for $\nu = 1, 2, \ldots, n$ take $u_\nu = A_{\nu-1}$ to be the quadratic function on $[x_{\nu-1}, x_\nu]$ having values $u_\nu(x_{\nu-1}) = u_\nu(x_\nu) = 0$, $u_\nu(x_{\nu-\frac{1}{2}}) = 1$ and define $A_{\nu-1}$ to be zero outside of $[x_{\nu-1}, x_\nu]$. Add to these functions the basis of hat functions B_ν for $\mathbb{S}_1^0(\Delta)$.}

(c) Compute the stiffness matrix $\boldsymbol{U}$ (in (7.142)) for the basis constructed in (b).

(d) Interpret the resulting system $\boldsymbol{U}\boldsymbol{\xi} = \boldsymbol{\rho}$ as a finite difference method applied to the given boundary value problem. What are the meanings of the components of $\boldsymbol{\xi}$?

15. (a) Show that the solution u_S of (7.141) is the orthogonal projection of the exact solution y of (7.138) onto the space S relative to the inner product $[\cdot\,,\cdot]$; that is,

$$[y - u_S, v] = 0 \quad \text{for all } v \in S.$$

(b) With $\|u\|_E$ denoting the energy norm of u (i.e., $\|u\|_E^2 = [u, u]$), show that

$$\|y - u_S\|_E^2 = \|y\|_E^2 - \|u_S\|_E^2.$$

16. Consider the boundary value problem (7.127) and (7.124). Define the energy norm by $\|u\|_E^2 = [u, u]$. Let Δ_1 and Δ_2 be two subdivisions of $[a, b]$ and $S_i = \{s \in \mathbb{S}_m^k(\Delta_i),\ s(a) = s(b) = 0\},\ i = 1, 2$, for some integers m, k with $0 \le k < m$.

(a) With y denoting the exact solution of the boundary value problem, and Δ_1 being a refinement of Δ_2, show that

$$\|y - u_{S_1}\|_E \le \|y - u_{S_2}\|_E.$$

(b) Let Δ_2 be an arbitrary subdivision of $[a, b]$ with all grid points (including the endpoints) being rational numbers. Prove that there exists a uniform subdivision Δ_1 of $[a, b]$, with $|\Delta_1| = h$ sufficiently small, such that

$$\|u - u_{S_1}\|_E \le \|y - u_{S_2}\|_E,$$

where S_i are as defined at the beginning of the exercise.

17. Apply the variational method to the boundary value problem

$$\mathcal{L}y := -py'' + qy = r(x), \quad 0 \le x \le 1;$$

$$y(0) = y(1) = 0,$$

where p and q are constants with $p > 0$, $q \ge 0$. Use approximants from the space $S = \operatorname{span}\{u_\nu(x) = \sin(\nu\pi x),\ \nu = 1, 2, \dots, n\}$, and interpret $\mathcal{L}u_S$. Find an explicit form for u_S in the case of constant r.

18. Let y be the exact solution of the boundary value problem (7.123)–(7.125) and u_S the approximate solution of the associated extremal problem with $S = \{s \in \mathbb{S}_1^0(\Delta) :\ s(a) = s(b) = 0\}$ and $\Delta :\ a = x_0 < x_1 < \cdots < x_n < x_{n+1} = b$. Prove that

$$\|y - u_S\|_\infty \le \frac{1}{2}\sqrt{\frac{\overline{c}}{\underline{c}}}\ \max_{0\le\nu\le n} \operatorname{osc}_{[x_\nu, x_{\nu+1}]}(y'),$$

where $\operatorname{osc}_{[c,d]}(f) := \max_{[c,d]} f - \min_{[c,d]} f$ and $\overline{c}$, $\underline{c}$ are the constants defined in (7.134). In particular, show that

$$\|y - u_S\|_\infty \le \tfrac{1}{2}\sqrt{\tfrac{\overline{c}}{\underline{c}}}\ |\Delta|\|y''\|_\infty.$$

{*Hint*: apply Theorem 7.4.4, in particular, (7.150).}

19. Consider the boundary value problem (7.127) and (7.124) with $p(x)$ and $q(x)$ being positive constants,

$$p(x) = p > 0, \quad q(x) = q > 0.$$

Let $S = \{s \in \mathbb{S}_1^0(\Delta) : \ s(a) = s(b) = 0\}$, where the subdivision Δ: $a = x_0 < x_1 < x_2 < \cdots < x_n < x_{n+1} = b$ is assumed to be *quasi uniform*; that is,

$$\Delta x_\nu := x_{\nu+1} - x_\nu \geq \beta|\Delta|, \quad \nu = 0, 1, \ldots, n,$$

for some positive constant β. (Recall that $|\Delta| := \max_{0\leq\nu\leq n} \Delta x_\nu$.) Let $\boldsymbol{U}$ be the stiffness matrix (cf. (7.142)) for the basis $u_\nu = B_\nu$, $\nu = 1, 2, \ldots, n$, of hat functions (cf. Chap. 2, Ex. 72, but note the difference in notation). Write $u(x) = \sum_{\nu=1}^{n} \xi_\nu u_\nu(x)$ for any $u \in S$, and $\boldsymbol{\xi}^{\mathrm{T}} = [\xi_1, \xi_2, \ldots, \xi_n]$.

(a) Show that $\boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{\xi} = [u, u]$.

(b) Show that $\|u'\|_{L_2}^2 = \boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{T}_1\boldsymbol{\xi}$, where $\boldsymbol{T}_1$ is a symmetric tridiagonal matrix with

$$(\boldsymbol{T}_1)_{\nu,\nu} = \frac{1}{\Delta x_{\nu-1}} + \frac{1}{\Delta x_\nu}\,, \quad \nu = 1, 2, \ldots, n;$$

$$(\boldsymbol{T}_1)_{\nu+1,\nu} = (\boldsymbol{T}_1)_{\nu,\nu+1} = -\frac{1}{\Delta x_\nu}\,, \quad \nu = 1, \ldots, n-1.$$

{*Hint*: use integration by parts, being careful to observe that u' is only piecewise continuous.}

(c) Show that $\|u\|_{L_2}^2 = \boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{T}_0\boldsymbol{\xi}$, where $\boldsymbol{T}_0$ is a symmetric tridiagonal matrix with

$$(\boldsymbol{T}_0)_{\nu,\nu} = \frac{1}{3}(\Delta x_{\nu-1} + \Delta x_\nu), \quad \nu = 1, 2, \ldots, n;$$

$$(\boldsymbol{T}_0)_{\nu+1,\nu} = (\boldsymbol{T}_0)_{\nu,\nu+1} = \frac{1}{6}\Delta x_\nu, \quad \nu = 1, \ldots, n-1.$$

(d) Combine (a)–(c) to compute $[u, u]$ and hence to estimate the Euclidean condition number $\mathrm{cond}_2\, \boldsymbol{U}$. {*Hint*: use Gershgorin's theorem to estimate the eigenvalues of $\boldsymbol{U}$.}

(e) The analysis in (d) fails if $q = 0$. Show, however, in the case of a *uniform* grid, that when $q = 0$ then $\mathrm{cond}_2\, \boldsymbol{U} \leq 1/\sin^2 \frac{\pi}{4n}$.

(f) Indicate how the argument in (d) can be extended to variable $p(x)$, $q(x)$ satisfying $0 < p(x) \leq \bar{p}$, $0 < \underline{q} \leq q(x) \leq \bar{q}$ on $[a, b]$.

20. The *method of collocation* for solving a boundary value problem

$$\mathcal{L}y = r(x), \ \ 0 \leq x \leq 1; \quad y(0) = y(1) = 0,$$

consists of selecting an n-dimensional subspace $S \subset V_0$ and determining $u_S \in S$ such that $(\mathcal{L}u_S)(x_\mu) = r(x_\mu)$ for a discrete set of points $0 < x_1 < x_2 <$

$\cdots < x_n < 1$. Apply this method to the problem of Ex. 17, with S as defined there. Discuss the solvability of the system of linear equations involved in the method. {*Hint*: use the known fact that the only trigonometric sine polynomial $\sum_{\nu=1}^{n} \xi_\nu \sin(\nu\pi x)$ of degree n that vanishes at n distinct points in (0, 1) is the one identically zero.}

Machine Assignments

1. The following eigenvalue problem arises in the physics of gas discharges. Determine the smallest positive $\lambda > 0$ such that

$$\varphi'' + \frac{1}{r}\varphi' + \lambda^2\varphi(1-\varphi) = 0, \quad 0 < r \leq 1,$$
$$\varphi(0) = a,\ \ \varphi'(0) = 0,\ \ \varphi(1) = 0,$$

where a is given, $0 < a < 1$.

(a) Explain why $\lambda = 0$ cannot be an eigenvalue.

(b) Reduce the problem to an initial value problem. {*Hint*: make a change of variables, $x = \lambda r$, $y(x) = \varphi(x/\lambda)$.}

(c) Use Maple to determine the Taylor expansion up to the power x^8 of the solution $y(x, a)$ to the initial value problem of (b).

(d) Integrate the initial value problem starting at $x = .1$, using the Taylor expansion of (c) to determine the initial data $y(.1, a)$ and $\frac{dy}{dx}(.1, a)$. Use the classical Runge–Kutta method (for example, the Matlab routine of Chap. 5, MA 1(a)) and integrate until the solution y becomes negative. Then apply interpolation to compute an approximation to λ, the solution of $y(\cdot, a) = 0$, to an accuracy of about five decimal digits. Prepare a table of the λ so obtained for $a = .1 : .1 : .9$, including the values of the integration step h required.

(e) For $a = .1 : .1 : .9$ use Matlab to produce graphs of the solutions $y(x, a)$ on intervals from $x = 0$ to $x = \lambda$, the zero of y. (Determine the endpoints of these intervals from the results of (d).) Use the Matlab routine `ode45` to do the integration from .1 to λ and connect the points $(0, a)$ and $(.1, y(.1, a))$ by a straight line segment. (Compute $y(.1, a)$) by the Taylor expansion of (c).)

2. The shape of an ideal flexible chain of length L, hung from two points (0, 0) and (1, 1), is determined by the solution of the eigenvalue problem

$$y'' = \lambda\sqrt{1 + (y')^2}, \quad y(0) = 0,\ y(1) = 1,\ \int_0^1 \sqrt{1 + (y')^2}\mathrm{d}x = L.$$

Strictly speaking, this is not a problem of the form (7.3), (7.4), but nevertheless can be solved analytically as well as numerically.

(a) On physical grounds, what condition must L satisfy for the problem to have a solution?

(b) Derive three equations in three unknowns: the two constants of integration and the eigenvalue λ. Obtain a transcendental equation for λ by eliminating the other two unknowns. Solve the equation numerically and thus find, and plot, the solution for $L = 2, 4, 8$, and 16.

(c) If one approximately solves the problem by a finite difference method over a uniform grid $x_0 = 0 < x_1 < x_2 < \cdots < x_N < x_{N+1} = 1$, $x_n = \frac{n}{N+1}$, approximating the integral in the third boundary condition by the composite trapezoidal rule, a system of $N+1$ nonlinear equations in $N+1$ unknowns results. Solve the system by a homotopy method, using L as the homotopy parameter. Since for $L = \sqrt{2}$ the solution is trivial, select a sequence of parameter values $L_0 = \sqrt{2} < L_1 < \cdots < L_m$ and solve the finite difference equations for L_i using the solution for L_{i-1} as the initial approximation. Implement this for the values of L given in (b), taking a sequence $\{L_i\}$ which contains these values. Compare the numerical results for the eigenvalues with the analytic ones for $N = 10,\ 20,\ 40$. (Use the routine `fsolve` from the Matlab optimization toolbox to solve the system of nonlinear equations.)

3. Change the boundary value problem of the first Example of Sect. 7.2.1 to

$$y'' = -\mathrm{e}^{y},\ \ 0 \le x \le 1,\ \ y(0) = y(1) = 0.$$

Then Theorem 7.1.2 no longer applies (why not?). In fact, it is known that the problem has two solutions. Use Matlab to compute the respective initial slopes $y'(0)$ to 12 significant digits by Newton's method, as indicated in the text. {*Hint*: use approximations $s^{(0)} = 1$ and $s^{(0)} = 15$ to the initial slopes.}

4. Consider the boundary value problem

$$\text{(BVP)} \qquad y'' = y^2,\ \ 0 \le x \le b;\quad y(0) = 0,\ y(b) = \beta,$$

and the associated initial value problem

$$\text{(IVP)} \qquad u'' = u^2,\ \ u(0) = 0,\ u'(0) = s.$$

Denote the solution of (IVP) by $u(x) = u(x; s)$.

(a) Let $v(x) = u(x; -1)$. Show that

$$v'(x) = -\sqrt{\tfrac{2}{3}v^3(x) + 1},$$

and thus the function v, being convex (i.e., $v'' > 0$), has a minimum at some $x_0 > 0$ with value $v_{\min} = -(3/2)^{1/3} = -1.1447142\ldots$. Show that v is symmetric with respect to the line $x = x_0$.

(b) Compute x_0 numerically in terms of the beta integral. $\{$*Point of information*: the beta integral is $B(p,q) = \int_0^1 t^{p-1}(1-t)^{q-1}\mathrm{d}t$ and has the value $\frac{\Gamma(p)\Gamma(q)}{\Gamma(p+q)}$.$\}$

(c) Use Matlab to compute v by solving the initial value problem

$$v'' = v^2, \quad x_0 \le x \le 3x_0; \quad v(x_0) = v_{\min},\ v'(x_0) = 0.$$

Plot the solution (and its symmetric part) on $-x_0 \le x \le 3x_0$.

(d) In terms of the function v defined in (a), show that

$$u(x; -s^3) = s^2 v(sx), \quad \text{all } s \in \mathbb{R}.$$

As s ranges over all the reals, the solution manifold $\{s^2 v(sx)\}$ thus encompasses all the solutions of (IVP). Prepare a plot of this solution manifold. Note that there exists an envelope of the manifold, located in the lower half-plane. Explain why, in principle, this envelope must be the solution of a first-order differential equation.

(e) Based on the plot obtained in (d), discuss the number of possible solutions to the original boundary value problem (BVP). In particular, determine for what values of b and β there does not exist any solution.

(f) Use the method of finite differences on a uniform grid to compute the two solutions of (BVP) for $b = 3x_0$, $\beta = v_0$, where $v_0 = v(3x_0)$ is a quantity already computed in (c). Solve the systems of nonlinear difference equations by Newton's method. In trying to get the first solution, approximate the solution v of (a) on $0 \le x \le 3x_0$ by a quadratic function $\tilde{v}$ satisfying $\tilde{v}(0) = 0$, $\tilde{v}'(0) = -1$, $\tilde{v}(3x_0) = v_0$, and then use its restriction to the grid as the initial approximation to Newton's method. For the second solution, try the initial approximation obtained from the linear approximation $\overline{v}(x) = v_0 x/(3x_0)$. In both cases, plot initial approximations as well as the solutions to the difference equations. What happens if Newton's method is replaced by the method of successive approximations?

5. The following boundary value problem occurs in soil engineering. Determine $y(r)$, $1 \le r < \infty$, such that

$$\frac{1}{r}\frac{\mathrm{d}}{\mathrm{d}r}\left(ry\frac{\mathrm{d}y}{\mathrm{d}r}\right) + \rho(1-y) = 0, \quad y(1) = \eta,\ y(\infty) = 1,$$

where ρ, η are parameters satisfying $\rho > 0$, $0 < \eta < 1$. The quantity of interest is $\sigma = \left.\frac{\mathrm{d}y}{\mathrm{d}r}\right|_{r=1}$.

(a) Let $z(x) = [y(\mathrm{e}^x)]^2$. Derive the boundary value problem and the quantity of interest in terms of z.

(b) Consider the initial value problem associated with the boundary value problem in (a), having initial conditions

$$z(0) = \eta^2, \quad z'(0) = s.$$

Discuss the qualitative behavior of its solutions for real values of s. {*Hint*: suggested questions may be: admissible domain in the (x, z)-plane, convexity and concavity of the solutions, the role of the line $z = 1$.}

(c) From your analysis in (b), devise an appropriate shooting procedure for solving the boundary value problem numerically and for computing the quantity σ. Run your procedure on the computer for various values of η and ρ. In particular, prepare a five-decimal table showing the values of s and σ for $\eta = 0.1(0.1).9$ and $\rho = 0.5, 1, 2, 5, 10$, and plot σ versus η.

Selected Solutions to Exercises

1. (a) We have $z'(t) = \lambda^{-\frac{2}{3}} y'(\lambda^{-\frac{1}{3}} t)$, $z''(t) = \lambda^{-1} y''(\lambda^{-\frac{1}{3}} t)$, $z'''(t) = \lambda^{-\frac{4}{3}} y'''(\lambda^{-\frac{1}{3}} t)$. Put $x = \lambda^{-\frac{1}{3}} t$ in the given boundary value problem to obtain $\lambda^{\frac{4}{3}} z''' + \frac{1}{2} \lambda^{\frac{1}{3}} z \cdot \lambda z'' = 0$, that is,

$$z''' + \tfrac{1}{2} z z'' = 0, \quad 0 \le t < \infty.$$

The initial conditions for z follow from those for y and the definition of λ:

$$z(0) = z'(0) = 0, \;\; z''(0) = 1.$$

(b) The boundary condition at ∞ for y' transforms to $z'(\infty) = \lambda^{-\frac{2}{3}}$, so that $\lambda = [z'(\infty)]^{-\frac{3}{2}}$. Thus, solving the initial value problem of (a) on $[0, \infty)$, we obtain $z'(\infty)$, hence λ, hence $y(x)$ as the solution of the initial value problem

$$y''' + \tfrac{1}{2} y y'' = 0,$$

$$y(0) = y'(0) = 0, \quad y''(0) = \lambda$$

To explore the convergence of $z'(t)$ as $t \to \infty$, we run the small Matlab program

```
%EXVII_1B1
%
f0='%8.2f %12.8f\n';
disp('     t        zprime')
z0=[0;0;1]; tspan=[0 5 10 15];
options=odeset('AbsTol',.5e-8);
[t,z]=ode45(@fEXVII_1,tspan,z0,options);
```

```
for i=1:4
  fprintf(f0,t(i,1),z(i,2))
end

%FEXVII_1
%
function yprime=fEXVII_1(x,y)
yprime=[y(2);y(3);-y(1)*y(3)/2];
```

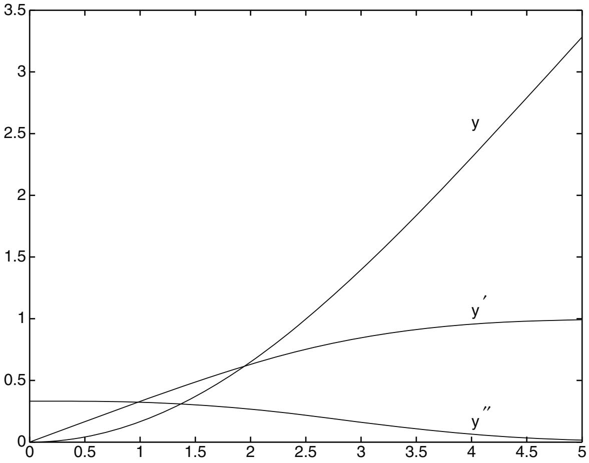
The results

```
>> EXVII_1B1
     t        zprime
    0.00   0.00000000
    5.00   2.08544470
   10.00   2.08553204
   15.00   2.08553204
>>
```

show that, for graphical purposes, $z'(\infty) = 2.0855$ is an acceptable value for the limit. Thus, $\lambda = [z'(\infty)]^{-\frac{3}{2}} = .33204$. We can now solve the initial value problem of interest. The program below plots the three components of the solution $\boldsymbol{y}(x) = [y(x), y'(x), y''(x)]$ on the interval $[0, 5]$.

```
%EXVII_1B2
%
global lambda
lambda=.33204;
y0=[0;0;lambda]; xspan=[0 5];
[x,y]=ode45(@fEXVII_1,xspan,y0);
plot(x,y)
```

PLOT



14. (a) The space $\mathbb{S}_2^0(\Delta)$ has degree of freedom $3n-(n-1)=2n+1$, since there are n quadratics having three degrees of freedom each, and $n-1$ continuity requirements at the $n-1$ interior grid points. Since the boundary conditions reduce the degree of freedom by two, S has $2n-1$ degrees of freedom, and we expect this to be the number of basis functions in S.

(b) Following the *Hint*, let $A_{\nu-1}(x)$, $\nu=1,2,\dots,n$, be supported on $[x_{\nu-1},x_\nu]$ and a quadratic function there, vanishing at the endpoints of the support interval and having the value 1 at the midpoint. Let $B_\nu(x)$, $\nu=1,2,\dots,n-1$, be the interior hat functions of those forming a basis of $\mathbb{S}_1^0(\Delta)$ (cf. Chap. 2, Sect. 2.3.2, but note the difference in notation). We claim that

$$u_\nu(x)=A_{\nu-1}(x),\;\;\nu=1,\dots,n;\;\;\;u_{n+\nu}(x)=B_\nu(x),\nu=1,\dots,n-1,$$

is a basis of S (it has the correct number of functions). To prove this, we must show that $\text{span}(u_1,u_2,\dots,u_{2n-1})=S$ and that the u_ν are linearly independent. Let

$$u(x)=\sum_{\nu=1}^{n}c_\nu A_{\nu-1}(x)+\sum_{\nu=1}^{n-1}c_{n+\nu}B_\nu(x).$$

It is clear that $u\in S$ (note that $A_0(0)=B_1(0)=0$, $A_{n-1}(1)=B_{n-1}(1)=0$), so that $\text{span}(u_1,\dots,u_{2n-1})\subset S$. Conversely, let $s\in S$ be an arbitrary member of S. Then it can be represented in the form above, i.e., $S\subset\text{span}(u_1,\dots,u_{2n-1})$. Indeed, note that for $\nu=1,2,\dots,n$; $\mu=0,1,\dots,n$

$$\begin{aligned}&A_{\nu-1}(x_\mu)=0,\quad A_{\nu-1}(x_{\mu+\frac12})=\delta_{\nu,\mu+1},\\&B_\nu(x_\mu)=\delta_{\nu\mu},\quad B_\nu(x_{\mu+\frac12})=\tfrac12(\delta_{\nu\mu}+\delta_{\nu,\mu+1}).\end{aligned}$$

Thus, putting $x=x_\mu$, we find $s(x_\mu)=c_{n+\mu}$ for $\mu=1,2,\dots,n-1$, and putting $x=x_{\mu+\frac12}$, we get $s(x_{\mu+\frac12})=c_{\mu+1}+\frac12(c_{n+\mu}+c_{n+\mu+1})$ for $\mu=0,1,\dots,n-1$ (where $c_{2n}=0$). The first set of equations determines $c_{n+1},c_{n+2},\dots,c_{2n-1}$, and the second set (written in reverse order) determines $c_n,c_{n-1},\dots,c_1$. This proves $\text{span}(u_1,\dots,u_{2n-1})=S$. The linear independence follows likewise (put $s(x)\equiv0$ in the argument above).

(c) Straightforward interpolation gives

$$A_{\nu-1}(x)=\frac{2}{h}(x-x_{\nu-1})\left[1-\frac{2}{h}(x-x_{\nu-\frac{1}{2}})\right] \text{ on } [x_{\nu-1},x_\nu],$$

$$\nu=1,2,\dots,n,$$

whereas

$$B_\nu(x)=\begin{cases}\dfrac{x-x_{\nu-1}}{h} \text{ on } [x_{\nu-1},x_\nu],\\ \dfrac{x_{\nu+1}-x}{h} \text{ on } [x_\nu,x_{\nu+1}],\end{cases} \quad \nu=1,2,\dots,n-1.$$

Hence,

$$A'_{\nu-1}(x)=\frac{4}{h^2}[(2\nu-1)h-2x] \text{ on } [x_{\nu-1},x_\nu],$$

$$B'_\nu(x)=\begin{cases}\dfrac{1}{h} \text{ on } [x_{\nu-1},x_\nu],\\ -\dfrac{1}{h} \text{ on } [x_\nu,x_{\nu+1}].\end{cases}$$

It is clear that the leading $n\times n$ diagonal block of $\boldsymbol{U}$ is a diagonal matrix, since $\int_0^1 A'_{\nu-1}(x)A'_{\mu-1}(x)\mathrm{d}x=0$ if $\nu\neq\mu$ and $p=1,q=0$ in (7.130). Its diagonal elements are

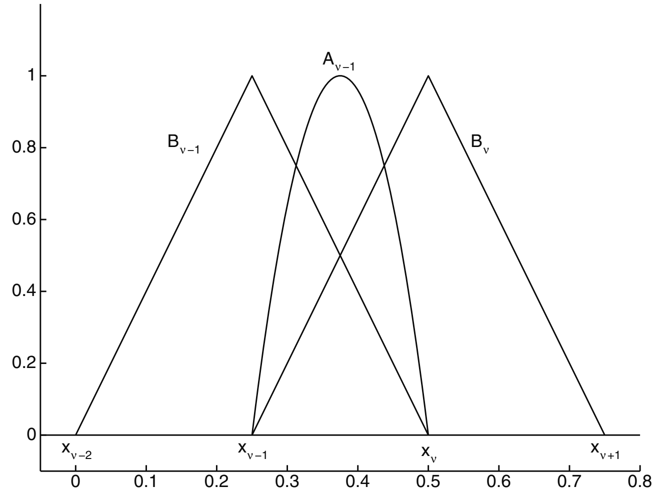
$$\int_0^1[A'_{\nu-1}(x)]^2\mathrm{d}x=\int_{x_{\nu-1}}^{x_\nu}\frac{16}{h^4}[(2\nu-1)h-2x]^2\mathrm{d}x$$

$$=\frac{16}{h^4}\left.\frac{[(2\nu-1)h-2x]^3}{3\cdot(-2)}\right|_{x_{\nu-1}}^{x_\nu}=\frac{16}{3h},\quad \nu=1,2,\dots,n.$$

The $n\times(n-1)$ block of $\boldsymbol{U}$ consisting of the first n rows and last $n-1$ columns is the zero matrix (and hence also the block symmetric to it). This is so because

$$\int_0^1 A'_{\nu-1}(x)B'_\mu(x)\mathrm{d}x=0 \quad \text{if } |\mu-\nu|>1,$$

on the next page the integrand being identically zero, and since by symmetry (see the figure for $\nu=2$ and $h=1/4$),



$$\begin{aligned}\int_0^1 A'_{\nu-1}(x)B'_{\nu-1}(x)\mathrm{d}x &= \int_0^1 A'_{\nu-1}(x)B'_{\nu}(x)\mathrm{d}x\\ &= \int_{x_{\nu-1}}^{x_\nu} \frac{4}{h^2}[(2\nu-1)h-2x]\frac{1}{h}\mathrm{d}x\\ &= \frac{4}{h^3}\ \frac{[(2\nu-1)h-2x]^2}{2\cdot(-2)}\bigg|_{x_{\nu-1}}^{x_\nu} = 0.\end{aligned}$$

Finally, the last $(n-1)\times(n-1)$ diagonal block is tridiagonal, since

$$\int_0^1 B'_\nu(x)B'_\mu(x)\mathrm{d}x = 0 \quad \text{if } |\mu-\nu|>1,$$

and

$$\int_0^1 [B'_\nu(x)]^2\mathrm{d}x = \frac{2}{h}, \quad \int_0^1 B'_\nu(x)B'_{\nu+1}(x)\mathrm{d}x = -\frac{1}{h}.$$

Thus,

$$\boldsymbol{U} = \frac{1}{h}\begin{bmatrix}\boldsymbol{D} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{T}\end{bmatrix}, \quad \boldsymbol{D} = \frac{16}{3}\boldsymbol{I},\ \boldsymbol{T} = \mathrm{tri}(-1,2,-1).$$

(d) The first n components of the load vector $\boldsymbol{\rho}$ (cf. (7.142)) are

$$\rho_\nu = \int_0^1 r(x)A_{\nu-1}(x)\mathrm{d}x = \int_{x_{\nu-1}}^{x_\nu} r(x)\frac{2}{h}(x-x_{\nu-1})\left[1-\frac{2}{h}(x-x_{\nu-\frac{1}{2}})\right]\mathrm{d}x.$$

With the change of variable $x = x_{\nu-1} + th$, this becomes

$$\rho_\nu = 4h \int_0^1 r(x_{\nu-1} + th)t(1-t)\mathrm{d}t, \quad \nu = 1, 2, \ldots, n.$$

The remaining components are

$$\rho_{n+\nu} = \int_0^1 r(x)B_\nu(x)\mathrm{d}x = \int_{x_{\nu-1}}^{x_\nu} r(x)\frac{x - x_{\nu-1}}{h}\mathrm{d}x + \int_{x_\nu}^{x_{\nu+1}} r(x)\frac{x_{\nu+1} - x}{h}\mathrm{d}x,$$

which, by the change of variable $x = x_\nu + th$, becomes

$$\rho_{n+\nu} = h\left\{\int_{-1}^0 r(x_\nu + th)(1+t)\mathrm{d}t + \int_0^1 r(x_\nu + th)(1-t)\mathrm{d}t\right\}, \\ \nu = 1, 2, \ldots, n-1.$$

They can readily be interpreted in terms of weighted averages of r over one or two consecutive subintervals. Indeed, if we introduce the weight functions

$$w_0(t) = t(1-t), \;\; 0 \le t \le 1; \quad w_1(t) = \begin{cases} 1+t, & -1 \le t \le 0, \\ 1-t, & 0 \le t \le 1, \end{cases}$$

then, since $\int_0^1 w_0(t)\mathrm{d}t = \frac{1}{6}$ and $\int_{-1}^1 w_1(t)\mathrm{d}t = 1$, we can write

$$\rho_\nu = \frac{2}{3}h\tilde{r}_\nu, \quad \tilde{r}_\nu = \frac{\displaystyle\int_0^1 r(x_{\nu-1} + th)w_0(t)\mathrm{d}t}{\displaystyle\int_0^1 w_0(t)\mathrm{d}t}, \;\; \nu = 1, 2, \ldots, n,$$

and

$$\rho_{n+\nu} = h\bar{r}_\nu, \quad \bar{r}_\nu = \frac{\displaystyle\int_{-1}^1 r(x_\nu + th)w_1(t)\mathrm{d}t}{\displaystyle\int_{-1}^1 w_1(t)\mathrm{d}t}, \;\; \nu = 1, 2, \ldots, n-1.$$

We can now interpret the system of linear equations $\boldsymbol{U}\boldsymbol{\xi} = \boldsymbol{\rho}$ as follows. First note that in

$$u(x) = \sum_{\nu=1}^{n} \xi_\nu A_{\nu-1}(x) + \sum_{\nu=1}^{n-1} \xi_{n+\nu} B_\nu(x)$$

we have from (b) that

$$u(x_\mu) = \xi_{n+\mu}, \quad \mu = 1, 2, \dots, n-1,$$

and

$$\begin{aligned} u\Big(x_{\mu+\frac{1}{2}}\Big) &= \xi_{\mu+1} + \frac{1}{2}(\xi_{n+\mu} + \xi_{n+\mu+1}) \\ &= \xi_{\mu+1} + \frac{1}{2}(u(x_\mu) + u(x_{\mu+1})), \mu = 0, 1, \dots, n-1. \end{aligned}$$

Thus, writing $u_\lambda = u(x_\lambda)$, the meaning of the $\boldsymbol{\xi}$-components is

$$\begin{aligned} &\xi_\nu = u_{\nu-\frac{1}{2}} - \frac{1}{2}(u_{\nu-1} + u_\nu), \quad \nu = 1, 2, \dots, n; \\ &\xi_{n+\nu} = u_\nu, \quad \nu = 1, 2, \dots, n-1. \end{aligned}$$

By Taylor's formula centered at $x_{\nu-1/2}$,

$$\xi_\nu = -\frac{h^2}{8}u''(x_{\nu-\frac{1}{2}}) + O(h^4).$$

The first n equations of the system $\boldsymbol{U}\boldsymbol{\xi} = \boldsymbol{\rho}$ thus are

$$\frac{16}{3h}\xi_\nu = \frac{2}{3}h\tilde{r}_\nu,$$

that is,

$$-u''(x_{\nu-\frac{1}{2}}) = \tilde{r}_\nu + O(h^2), \quad \nu = 1, 2, \dots, n.$$

The remaining equations are the standard finite difference equations

$$-(u_{\nu-1} - 2u_\nu + u_{\nu+1}) = h^2\bar{r}_\nu + O(h^2), \quad \nu = 1, 2, \dots, n-1,$$

where $u_0 = u_n = 0$.

19. (a) We have

$$[u, u] = \left[\sum_{\nu=1}^{n} \xi_\nu u_\nu, \sum_{\mu=1}^{n} \xi_\mu u_\mu\right] = \sum_{\nu,\mu=1}^{n} [u_\nu, u_\mu]\xi_\nu\xi_\mu = \boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{\xi}.$$

(b) We have

$$\|u'\|_{L_2}^2 = \int_a^b [u'(x)]^2 \mathrm{d}x = \sum_{\nu=0}^{n} \int_{x_\nu}^{x_{\nu+1}} [u'(x)]^2 \mathrm{d}x.$$

To each integral in the summation on the right, we apply integration by parts:

$$\int_{x_\nu}^{x_{\nu+1}} u'(x)u'(x)\mathrm{d}x = u'u\Big|_{x_\nu}^{x_{\nu+1}} - \int_{x_\nu}^{x_{\nu+1}} u''(x)u(x)\mathrm{d}x$$
$$= u'(x_{\nu+1}-0)u(x_{\nu+1}) - u'(x_\nu+0)u(x_\nu),$$

since $u''(x) \equiv 0$ on $(x_\nu, x_{\nu+1})$. On the interval $[x_0, x_1]$ one has $u(x) = \xi_1 B_1(x)$, hence

$$u(x_0) = 0,\ \ u(x_1) = \xi_1;\ \ u'(x_0+0) = u'(x_1-0) = \frac{1}{\Delta x_0}\xi_1.$$

On the interval $[x_\nu, x_{\nu+1}]$ $(1 \le \nu \le n-1)$, one has $u(x) = \xi_\nu B_\nu(x) + \xi_{\nu+1}B_{\nu+1}(x)$, hence

$$u(x_\nu)=\xi_\nu,\ \ u(x_{\nu+1}) = \xi_{\nu+1};\ \ u'(x_\nu+0)=u'(x_{\nu+1}-0) = \frac{1}{\Delta x_\nu}(\xi_{\nu+1}-\xi_\nu).$$

Finally, on $[x_n, x_{n+1}]$, one has $u(x) = \xi_n B_n(x)$, hence

$$u(x_n) = \xi_n,\ \ u(x_{n+1}) = 0;\ \ u'(x_n+0) = u'(x_{n+1}-0) = -\frac{1}{\Delta x_n}\xi_n.$$

There follows

$$\|u'\|_{L_2}^2 = \frac{1}{\Delta x_0}\xi_1^2 + \sum_{\nu=1}^{n-1}\left[\frac{1}{\Delta x_\nu}(\xi_{\nu+1}-\xi_\nu)\xi_{\nu+1} - \frac{1}{\Delta x_\nu}(\xi_{\nu+1}-\xi_\nu)\xi_\nu\right] + \frac{1}{\Delta x_n}\xi_n^2$$
$$= \frac{1}{\Delta x_0}\xi_1^2 + \sum_{\nu=1}^{n-1}\frac{1}{\Delta x_\nu}(\xi_{\nu+1}^2 - 2\xi_\nu\xi_{\nu+1} + \xi_\nu^2) + \frac{1}{\Delta x_n}\xi_n^2$$
$$= \sum_{\nu=1}^{n}\left(\frac{1}{\Delta x_{\nu-1}} + \frac{1}{\Delta x_\nu}\right)\xi_\nu^2 - 2\sum_{\nu=1}^{n-1}\frac{1}{\Delta x_\nu}\xi_\nu\xi_{\nu+1}$$
$$= \boldsymbol{\xi}^{\mathrm{T}}\boldsymbol{T}_1\boldsymbol{\xi}.$$

(c) We have

$$\|u\|_{L_2}^2 = \int_a^b [u(x)]^2 \mathrm{d}x = \int_a^b \sum_{\nu=1}^n \xi_\nu B_\nu(x) \sum_{\mu=1}^n \xi_\mu B_\mu(x) \mathrm{d}x$$
$$= \sum_{\nu,\mu=1}^n \xi_\nu \xi_\mu \int_a^b B_\nu(x) B_\mu(x) \mathrm{d}x.$$

Since $\int_a^b B_\nu(x) B_\mu(x)\mathrm{d}x = 0$ if $|\mu - \nu| > 1$, the matrix of this quadratic form is tridiagonal, and clearly symmetric. One computes

$$\int_a^b B_\nu^2(x)\mathrm{d}x = \int_{x_{\nu-1}}^{x_\nu} \left(\frac{x - x_{\nu-1}}{\Delta x_{\nu-1}}\right)^2 \mathrm{d}x + \int_{x_\nu}^{x_{\nu+1}} \left(\frac{x_{\nu+1} - x}{\Delta x_\nu}\right)^2 \mathrm{d}x$$
$$= \Delta x_{\nu-1} \int_0^1 t^2 \mathrm{d}t + \Delta x_\nu \int_0^1 (1-t)^2 \mathrm{d}t$$
$$= \frac{1}{3}(\Delta x_{\nu-1} + \Delta x_\nu), \quad \nu = 1, 2, \ldots, n,$$

and

$$\int_a^b B_\nu(x) B_{\nu+1}(x)\mathrm{d}x = \int_{x_\nu}^{x_{\nu+1}} \frac{x_{\nu+1} - x}{\Delta x_\nu} \, \frac{x - x_\nu}{\Delta x_\nu} \mathrm{d}x$$
$$= \Delta x_\nu \int_0^1 (1-t)t\mathrm{d}t = \frac{1}{6}\Delta x_\nu, \; \nu = 1, 2, \ldots, n-1.$$

There follows

$$\|u\|_{L_2}^2 = \boldsymbol{\xi}^{\mathrm{T}} \boldsymbol{T}_0 \boldsymbol{\xi},$$

as claimed.

(d) We have by (7.130) and the results in (b) and (c),

$$[u, u] = p\|u'\|_{L_2}^2 + q\|u\|_{L_2}^2 = \boldsymbol{\xi}^{\mathrm{T}}(p\boldsymbol{T}_1 + q\boldsymbol{T}_0)\boldsymbol{\xi}.$$

Comparison with (a) shows that $\boldsymbol{U} = p\boldsymbol{T}_1 + q\boldsymbol{T}_0$. The Gershgorin disks of the tridiagonal matrix $\boldsymbol{U}$ have centers at

$$p\left(\frac{1}{\Delta x_{\nu-1}} + \frac{1}{\Delta x_\nu}\right) + \frac{1}{3}q\,(\Delta x_{\nu-1} + \Delta x_\nu), \quad \nu = 1, 2, \ldots, n,$$

with respective radii

$$p\frac{1}{\Delta x_1}+\frac{1}{6}q\Delta x_1,$$
$$p\left(\frac{1}{\Delta x_{\nu-1}}+\frac{1}{\Delta x_\nu}\right)+\frac{1}{6}q\,(\Delta x_{\nu-1}+\Delta x_\nu),\quad \nu=2,\dots,n-1,$$
$$p\frac{1}{\Delta x_{n-1}}+\frac{1}{6}q\Delta x_{n-1}.$$

Since by Gershgorin's theorem all eigenvalues λ_ν of $\boldsymbol{U}$ are contained in the union of these disks, and are real, we have

$$\begin{aligned}\lambda_{\max}\le\max\Big\{&2p\left(\frac{1}{2}\frac{1}{\Delta x_0}+\frac{1}{\Delta x_1}\right)+\frac{1}{2}q\left(\frac{2}{3}\Delta x_0+\Delta x_1\right);\\ &2p\left(\frac{1}{\Delta x_{\nu-1}}+\frac{1}{\Delta x_\nu}\right)+\frac{1}{2}q\,(\Delta x_{\nu-1}+\Delta x_\nu),\ \ \nu=2,\dots,n-1;\\ &2p\left(\frac{1}{\Delta x_{n-1}}+\frac{1}{2}\frac{1}{\Delta x_n}\right)+\frac{1}{2}q\left(\Delta x_{n-1}+\frac{2}{3}\Delta x_n\right)\Big\}.\end{aligned}$$

By the quasi uniformity of the grid, we have

$$\frac{1}{\Delta x_\nu}\le\frac{1}{\beta}\frac{1}{|\Delta|},\quad \nu=0,1.\dots,n,$$

and therefore

$$\lambda_{\max}\le\frac{4p}{\beta|\Delta|}+q|\Delta|.$$

Similarly,

$$\begin{aligned}\lambda_{\min}\ge\min\Big\{&p\,\frac{1}{\Delta x_0}+\frac{1}{6}q\,(2\Delta x_0+\Delta x_1);\\ &\frac{1}{6}\,q\,(\Delta x_{\nu-1}+\Delta x_\nu),\ \ \nu=2,\dots,n-1;\\ &p\,\frac{1}{\Delta x_n}+\frac{1}{6}q\,(\Delta_{n-1}+2\Delta x_n)\Big\}.\end{aligned}$$

By the quasiuniformity of the grid and $p>0$, we get

$$\lambda_{\min}\ge\frac{1}{3}q\beta|\Delta|.$$

There follows

$$\operatorname{cond}_2\boldsymbol{U}=\frac{\lambda_{\max}}{\lambda_{\min}}\le\frac{12p}{q\beta^2|\Delta|^2}+\frac{3}{\beta}.$$

(e) It is clear from (d) that in the case $q = 0$ one has $\boldsymbol{U} = p\boldsymbol{T}_1$, and for a uniform grid with $|\Delta| = h = \frac{b-a}{n+1}$, by the definition of $\boldsymbol{T}_1$,

$$[u,u] = \boldsymbol{\xi}^{\mathrm{T}} \frac{p}{h} \boldsymbol{T} \boldsymbol{\xi}, \quad \boldsymbol{U} = \frac{p}{h} \boldsymbol{T},$$

where $\boldsymbol{T} = \mathrm{tri}(-1, 2, -1)$ is a tridiagonal matrix of order n with elements 2 on the diagonal and -1 on the two side diagonals. Thus, $\mathrm{cond}_2 \boldsymbol{U} = \mathrm{cond}_2 \boldsymbol{T}$. To find the eigenvalues of $\boldsymbol{T}$, note that the characteristic polynomial $\pi_n(\lambda) = \det(\boldsymbol{T} - \lambda \boldsymbol{I})$ satisfies

$$\pi_{k+1}(\lambda) = (2-\lambda)\pi_k(\lambda) - \pi_{k-1}(\lambda), \quad k = 0, 1, \dots, n-1,$$
$$\pi_{-1}(\lambda) = 0, \ \ \pi_0(\lambda) = 1.$$

Hence, in terms of the Chebyshev polynomial T_n, one has (cf. Chap. 2, Sect. 2.2.4, (2.83))

$$\pi_n(\lambda) = T_n\left(1 - \frac{1}{2}\lambda\right), \ \ n \geq 2; \quad \pi_1(\lambda) = 2T_1\left(1 - \frac{1}{2}\lambda\right).$$

For the eigenvalues λ_ν we therefore have $1 - \frac{1}{2}\lambda_\nu = \cos \frac{2\nu-1}{2n}\pi$, thus,

$$\lambda_\nu = 2\left(1 - \cos\frac{2\nu-1}{2n}\pi\right) = 4\sin^2\frac{2\nu-1}{2n}\frac{\pi}{2}, \quad \nu = 1, 2, \dots, n.$$

It follows that

$$\lambda_{\max} = 4\sin^2\frac{2n-1}{2n}\frac{\pi}{2} = 4\cos^2\frac{\pi}{4n}, \quad \lambda_{\min} = 4\sin^2\frac{\pi}{4n},$$

and

$$\mathrm{cond}_2 \boldsymbol{U} = \cot^2\frac{\pi}{4n}.$$

In particular, $\mathrm{cond}_2 \boldsymbol{U} \leq 1/\sin^2\frac{\pi}{4n}$.

(f) In (7.130) use the mean value theorem of integration to write

$$[u,u] = p(\xi)\|u'\|_{L_2}^2 + q(\eta)\|u\|_{L_2}^2.$$

Hence, p and q, throughout the argument in (d), can be replaced by $p(\xi)$ and $q(\eta)$, respectively. The result is

$$\mathrm{cond}_2 \boldsymbol{U} \leq \frac{12\overline{p}}{\underline{q}\beta^2|\Delta|^2} + \frac{3}{\beta}.$$

Selected Solutions to Machine Assignments

1. (a) If $\lambda = 0$, then

$$\varphi'' + \frac{1}{r}\varphi' = 0,$$

giving

$$\varphi(r) = c_1 \ln r + c_2$$

with constants c_1, c_2. The first condition $\varphi(0) = a$ requires $c_1 = 0$, $c_2 = a$, hence $\varphi(r) = a$. This contradicts the third condition, $\varphi(1) = 0$, since $a > 0$.

(b) With the suggested change of variables $x = \lambda r$, $y(x) = \varphi(x/\lambda)$, we have

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{1}{\lambda}\varphi'(x/\lambda), \quad \frac{\mathrm{d}^2 y}{\mathrm{d}x^2} = \frac{1}{\lambda^2}\varphi''(x/\lambda).$$

Putting $r = x/\lambda$ in the boundary value problem, we get

$$\lambda^2 \frac{\mathrm{d}^2 y}{\mathrm{d}x^2} + \frac{\lambda}{x} \cdot \lambda \frac{\mathrm{d}y}{\mathrm{d}x} + \lambda^2 y(1-y) = 0,$$

$$y(0) = a, \quad \frac{\mathrm{d}y}{\mathrm{d}x}(0) = 0, \quad y(\lambda) = 0,$$

that is,

$$\frac{\mathrm{d}^2 y}{\mathrm{d}x^2} + \frac{1}{x}\frac{\mathrm{d}y}{\mathrm{d}x} + y(1-y) = 0,$$

$$y(0) = a, \quad \frac{\mathrm{d}y}{\mathrm{d}x}(0) = 0.$$

Thus, we need to integrate this initial value problem until the solution y vanishes for the first time. The corresponding value of x is the desired eigenvalue.

(c)

```
 PROGRAM  (Maple)

eq:=diff(y(x),x,x)+(1/x)*diff(y(x),x)+y(x)*(1-y(x))=0;
ini:=y(0)=a, D(y)(0)=0;
Order:=10;
sol:=dsolve({eq,ini},{y(x)},type=series);
p:=convert(sol,polynom);
```

The results produced for the coefficients $c_0, c_1, c_2, \ldots$ in $y(x,a) = c_0 + c_1x + c_2x^2 \ldots$ are

$$\begin{aligned}
c_1 &= c_3 = c_5 = c_7 = 0,\\
c_0 &= a,\\
c_2 &= -\frac{1}{4}\,a(1-a),\\
c_4 &= \frac{1}{64}\,a(1-a)(1-2a),\\
c_6 &= -\frac{1}{2304}\,a(1-a)(1-8a+8a^2),\\
c_8 &= \frac{1}{147456}\,a(1-a)(1-2a)(1-26a+26a^2).
\end{aligned}$$

(d) We use cubic interpolation (which has the same order $O(h^4)$ of error as the Runge–Kutta method) and add one more integration step after the solution has become negative to make the interpolation problem more symmetric.

PROGRAMS

```
%MAVII_1D
%
f0='%6.2f %9.5f %11.4e\n';
disp(' a  lambda  h')
y=zeros(4,1);
for a=.1:.1:.9
  h=.01; lam1=0; errlam=1;
  while errlam>.5e-5
    h=h/2; lam0=lam1; x=.1;
    y(2)=yMAVII_1(x-2*h,a);
    y(3)=yMAVII_1(x-h,a);
    [y(4),y1]=yMAVII_1(x,a);
    u1=[y(4);y1]; u=1;
    while u>0
      u0=u1; y(1:3)=y(2:4);
      u1=RK4(@fMAVII_1,x,u0,h);
      y(4)=u1(1); u=y(4); x=x+h;
    end
    y(1:3)=y(2:4);
    u2=RK4(@fMAVII_1,x,u1,h);
    y(4)=u2(1); p=[(-y(1)+3*y(2) ...
      -3*y(3)+y(4))/6 (2*y(1)-5*y(2) ...
      +4*y(3)-y(4))/2 (-11*y(1) ...
      +18*y(2)-9*y(3)+2*y(4))/6 y(1)];
    r=roots(p); lam1=0;
    for k=1:3
      if 1<r(k) & r(k)<2
        lam1=x+(r(k)-3)*h ;
      end
    end
    if lam1==0
      fprintf(['interpolant not in' ...
         'range for a=%5.2f\n'],a)
      return
    end
    errlam=abs(lam1-lam0);
  end
  fprintf(f0,a,lam1,h)
end
```

```
%FMAVII_1
%
function yprime=fMAVII_1(x,y)
yprime=[y(2);-y(2)/x-y(1)*(1-y(1))];
```

```
%YMAVII_1
%
function [y,y1]=yMAVII_1(x,a)
y=a-(1/4)*a*(1-a)*x^2+(1/64)*a*(1-a)*(1-2*a)*x^4-(1/2304) ...
  *a*(1-a)*(1-8*a+8*a^2)*x^6+(1/147456)*a*(1-a)*(1-2*a) ...
  *(1-26*a+26*a^2)*x^8;
y1=-(1/2)*a*(1-a)*x+(1/16)*a*(1-a)*(1-2*a)*x^3-(1/384) ...
  *a*(1-a)*(1-8*a+8*a^2)*x^5+(1/18432)*a*(1-a)*(1-2*a) ...
  *(1-26*a+26*a^2)*x^7;
```

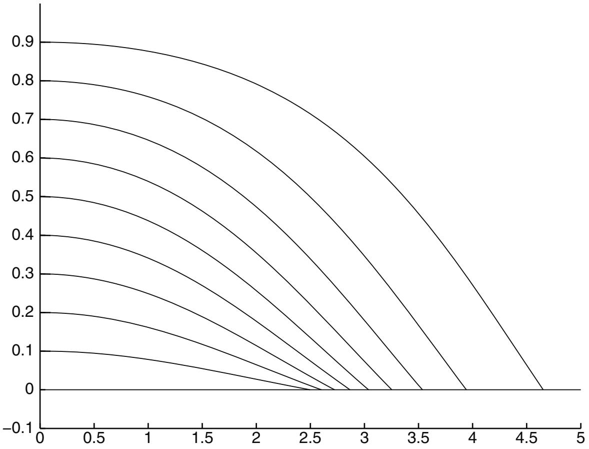
```
 OUTPUT
>> MAVII_1D
   a     lambda       h
  0.10   2.49725  4.8828e-06
  0.20   2.60240  4.8828e-06
  0.30   2.72378  4.8828e-06
  0.40   2.86654  4.8828e-06
  0.50   3.03864  4.8828e-06
  0.60   3.25347  4.8828e-06
  0.70   3.53610  4.8828e-06
  0.80   3.94284  4.8828e-06
  0.90   4.65326  4.8828e-06
>>
```

(e)

PROGRAM

```
%MAVII_1E
%
endpoint=[2.497;2.602;2.724;2.867; ...
   3.039;3.253;3.536;3.943;4.653];
hold on
for ia=1:9
  a=ia*.1;
  [y,y1]=yMAVII_1(.1,a);
  u0=[y;y1]; xspan=[.1 endpoint(ia)];
  [x,u]=ode45(@fMAVII_1,xspan,u0);
  plot(x,u(:,1))
  plot([0 .1],[a y])
end
plot([0 5],[0 0])
axis([0 5 -.1 1])
hold off
```

PLOTS



4. (a) The function $v(x) = u(x; -1)$ satisfies

$$\frac{1}{2}\,\frac{\mathrm{d}}{\mathrm{d}x}(v')^2 = \frac{1}{3}\,\frac{\mathrm{d}}{\mathrm{d}x}v^3,$$

which, when integrated from 0 to x, and using $v(0) = 0$, $v'(0) = -1$, gives

$$[v'(x)]^2 - 1 = \frac{2}{3}\,v^3(x),$$

that is,

$$v'(x) = \pm\sqrt{\tfrac{2}{3}\, v^3(x) + 1}.$$

Since v has a negative slope at $x = 0$, and $v(0) = 0$, it initially decreases and becomes negative. Therefore, in the formula above, we must take the square root with the minus sign. This proves the assertion. Let $v(x_0 \pm t) = z^{\pm}(t)$. Both functions $z^{\pm}$ satisfy the same differential equation $z'' = z^2$ with the same initial conditions $z(0) = v_{\min}$, $z'(0) = 0$. Hence, they are the same, $z^+(t) \equiv z^-(t)$, proving symmetry of v with respect to the line $x = x_0$.

(b) From part (a) we have

$$\frac{\mathrm{d}v}{\mathrm{d}x} = -\sqrt{\tfrac{2}{3}\, v^3 + 1}, \quad \frac{\mathrm{d}x}{\mathrm{d}v} = -\frac{1}{\sqrt{\tfrac{2}{3}v^3 + 1}},$$

from which there follows, since $x = 0$ for $v = 0$,

$$x = x(v) := -\int_0^v \frac{\mathrm{d}t}{\sqrt{\tfrac{2}{3}t^3 + 1}}.$$

Since $x_0 = x(v_{\min})$, we get

$$x_0 = -\int_0^{v_{\min}} \frac{\mathrm{d}t}{\sqrt{\tfrac{2}{3}t^3 + 1}} = \int_0^{|v_{\min}|} \frac{\mathrm{d}\tau}{\sqrt{1 - \tfrac{2}{3}\,\tau^3}}, \quad |v_{\min}| = \left(\frac{3}{2}\right)^{1/3}.$$

The change of variables $\frac{2}{3}\,\tau^3 = t$, $\mathrm{d}\tau = (18)^{-1/3} t^{-2/3}\mathrm{d}t$ yields

$$x_0 = (18)^{-1/3} \int_0^1 t^{-2/3}(1-t)^{-1/2}\mathrm{d}t = (18)^{-1/3} B(\tfrac{1}{3}, \tfrac{1}{2}) = \frac{\sqrt{\pi}}{(18)^{1/3}} \frac{\Gamma(1/3)}{\Gamma(5/6)},$$

since $\Gamma(1/2) = \sqrt{\pi}$. With the Matlab command

```
x0=sqrt(pi)*gamma(1/3)/(18^(1/3)*gamma(5/6))
```

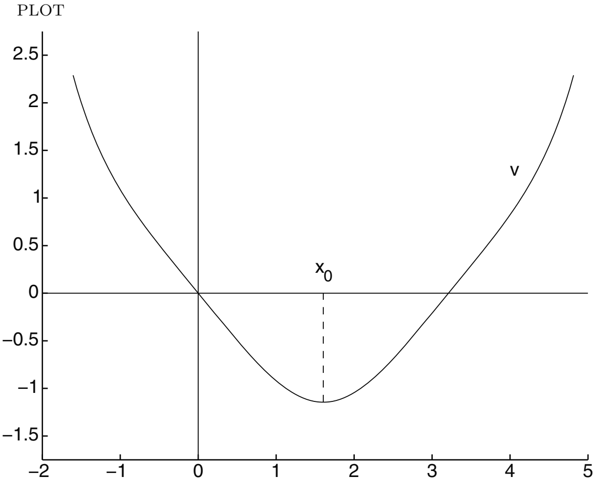
one computes

$$x_0 = 1.605097826619464.$$

(c)

```
 PROGRAM

%MAVII_4C
%
x0=1.6050978; vmin=-(3/2)^(1/3);
vspan=[x0 3*x0]; v0=[vmin;0];
```

```
options=odeset('AbsTol',.5e-8);
[x,v]=ode45(@fMAVII_4C,vspan,v0,options);
v(size(v,1),1)
hold on
plot(x,v(:,1))
plot(2*x0-x,v(:,1))
hold off

%fMAVII_4C
%
function vprime=fMAVII_4C(x,v)
vprime=[v(2);v(1)^2];
```



(d) Let $u(x) = s^2 v(sx)$. Then

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2}\,u = s^4 v''(sx) = s^4 v^2(sx) = u^2,$$

showing that u satisfies the differential equation of (IVP). Furthermore,

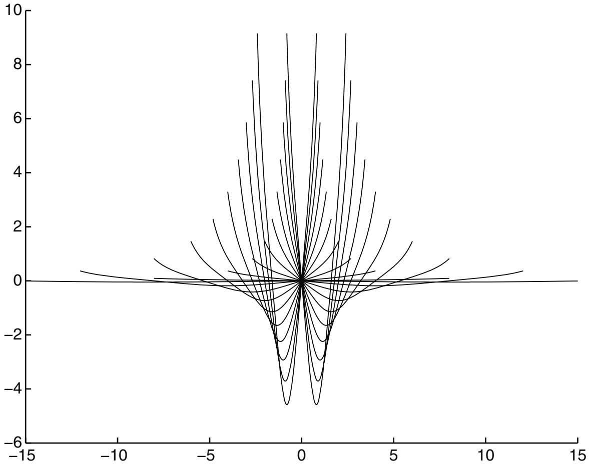
$$u(0) = 0, \quad \left.\frac{\mathrm{d}}{\mathrm{d}x}\,u(x)\right|_{x=0} = s^3\, v'(sx)\big|_{x=0} = -s^3.$$

Therefore, $u(x) = u(x; -s^3)$ as claimed.

PROGRAM

```
% MAVII_4D
%
x0=1.6050978; xf=3*x0;
vspan=[x0 3*x0]; vmin=-(3/2)^(1/3);
v0=[vmin;0]; options=odeset(’AbsTol’,.5e-8);
[x,v]=ode45(@fMAVII_4C,vspan,v0,options);
hold on
for s=-2:.2:-.2
  sx=x0:.01:xf;
  vsx=interp1(x,v(:,1),sx,spline);
  xs=sx/s;
  plot(xs,(s^2)*vsx);
  sx=x0:-.01:-x0;
  xs=sx/s;
  plot(xs,(s^2)*vsx);
  axis([-15 15 -6 10])
end
for s=.2:.2:2
  sx=x0:.01:xf;
  vsx=interp1(x,v(:,1),sx,spline);
  xs=sx/s;
  plot(xs,(s^2)*vsx)
  sx=x0:-.01:-x0;
  xs=sx/s;
  plot(xs,(s^2)*vsx);
  axis([-15 15 -6 10])
end
hold off
```

PLOTS



If $y = e(x)$ is the equation of the envelope, and $v(x;s) := s^2v(sx)$, then for each $x \in \mathbb{R}$ there must exist an s such that

$$e(x) = v(x;s) \quad \text{and} \quad \mathrm{e}'(x) = v'(x;s),$$

where prime means differentiation with respect to x. If the first equation is solved (in principle) for s as a function of x and e, and the result $s = s(x,e)$ substituted into the second equation, there results the (rather complicated) differential equation

$$e' = v'(x;s(x,e)).$$

(e) If the point (b,β) lies in the domain *above* the envelope, there are exactly two solutions of (BVP), one touching the right-hand border of the envelope, the other the left-hand border. If (b,β) lies *on* the envelope, there is exactly one solution, the one touching the envelope at the point (b,β). If (b,β) lies in the domain *below* the envelope, there cannot exist any solution.

(f) Given the uniform grid $\Delta : x_0 = 0 < x_1 < \cdots < x_N < x_{N+1} = 3x_0$, with $|\Delta| = h = 3x_0/(N+1)$, the simplest difference equations are

$$u_{n-1} - 2u_n + u_{n+1} = h^2u_n^2, \quad n = 1,2,\dots,N,$$
$$u_0 = 0, \;\; u_{N+1} = v_0.$$

Writing $\boldsymbol{u} = [u_1,u_2,\dots,u_N]^{\mathrm{T}}$ and letting $\boldsymbol{A} = \mathrm{tri}(1,-2,1)$ be the tridiagonal matrix of order N with elements -2 on the diagonal and 1 on the side diagonals, the system of difference equations can be written as

$$\boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{0}, \quad \boldsymbol{f}(\boldsymbol{u}) := \boldsymbol{A}\boldsymbol{u} - h^2\boldsymbol{u}^2 + v_0\boldsymbol{e}_N,$$

where $\boldsymbol{u}^2 = [u_1^2,u_2^2,\dots,u_N^2]^{\mathrm{T}}$ and $\boldsymbol{e}_N = [0,0,\dots,0,1]^{\mathrm{T}}$. The Jacobian of $\boldsymbol{f}$ is

$$\boldsymbol{f_u}(\boldsymbol{u}) = \boldsymbol{A} - 2h^2\mathrm{diag}(\boldsymbol{u}),$$

where $\mathrm{diag}(\boldsymbol{u})$ is the diagonal matrix with diagonal elements $u_1,u_2,\dots,u_N$. Newton's method therefore takes the form

$$\boldsymbol{u}^{[i+1]} = \boldsymbol{u}^{[i]} - \boldsymbol{d}^{[i]}, \quad \boldsymbol{f_u}(\boldsymbol{u}^{[i]})\boldsymbol{d}^{[i]} = \boldsymbol{f}(\boldsymbol{u}^{[i]}), \;\; i = 0,1,2,\dots,$$

where $\boldsymbol{u}^{[0]} = [u_1^{[0]},u_2^{[0]},\dots,u_N^{[0]}]^{\mathrm{T}}$, $u_n^{[0]} = \tilde{v}(x_n)$ resp. $u_n^{[0]} = \bar{v}(x_n)$. The functions $\tilde{v}$, $\bar{v}$ determining the two initial approximations are

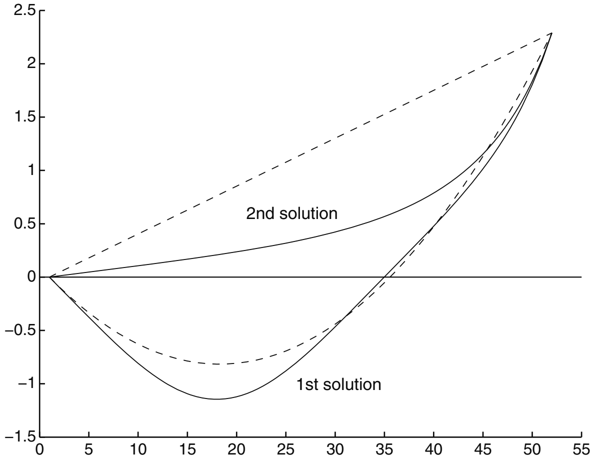
$$\tilde{v}(x) = \frac{v_0 + 3x_0}{(3x_0)^2}\,x^2 - x, \quad \bar{v}(x) = \frac{v_0}{3x_0}\,x.$$

The program implementing this method with $N = 50$ and accuracy tolerance $\varepsilon_0 = .5 \times 10^{-8}$, and with initial approximations as suggested, is shown below.

```
 PROGRAM

% MAVII_4F
%
N=50; eps0=.5e-8;
x0=1.6050978; v0=2.2893929;
h=3*x0/(N+1); h2=h^2;
u0=zeros(N,1);
A=zeros(N);
e=eye(N);
eN=e(:,N);
hold on
for i=1:2
  for n=1:N
    if(i==1)
      u0(n)=n*h*(-1+((n*h)/(3*x0)) ...
        *(1+v0/(3*x0)));
    else
      u0(n)=n*h*v0/(3*x0);
    end
    A(n,n)=-2;
    if(n<N)
    A(n,n+1)=1;
    A(n+1,n)=1;
    end
  end
  u0p=[0;u0;v0];
  plot(u0p,’--’);
  axis([0 55 -1.5 2.5])
  plot([0 55],[0 0])
  u1=u0; u0=zeros(N,1);
  it=0;
  while(norm(u1-u0,inf)>eps0 & it<20)
   it=it+1;
   u0=u1;
   u2=u0.^2;
   f=A*u0-h2*u2+v0*eN;
   J=A-2*h2*diag(u0);
   d=J\f;
   u1=u0-d;
 end
```

```
    u1p=[0;u1;v0];
    plot(u1p);
    text(26,-1,'1st solution','FontSize',14)
    text(21,.6,'2nd solution','FontSize',14)
  end
  hold off
```

When applying this routine, we get rapid convergence (within five to six iterations). Plots of the answers are shown below. The dashed lines are initial approximations.



The method of successive approximations, described by

$$\boldsymbol{u}^{[i+1]} = \boldsymbol{A}^{-1}(h^2(\boldsymbol{u}^{[i]})^2 - v_0\boldsymbol{e}_N), \quad i = 0, 1, 2, \ldots,$$

does not converge, neither for $N = 50$ nor for $N = 20, 10$, or 5.

References

Abramowitz, Milton [1954]. On the practical evaluation of integrals, *J. Soc. Indust. Appl. Math. 2*, 20–35. (Reprinted in Davis and Rabinowitz [2007, Appendix 1].)

Abramowitz, Milton and Stegun, Irene A. [1964]. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, NBS Appl. Math. Ser., v. 55, Government Printing Office, Washington, D.C.

Agarwal, Ravi P. [1986]. *Boundary value problems for higher order differential equations*, World Scientific, Teaneck, NJ.

Agarwal, Ravi P. [2000]. *Difference equations and inequalities: Theory, methods, and applications*, 2d ed., Monographs and Textbooks Pure Appl. Math., v. 155, Dekker, New York.

Ahlberg, J. H., Nilson, E. N., and Walsh, J. L. [1967]. *The theory of splines and their applications*, Math. Sci. Engrg., v. 38, Academic Press, New York.

Akl, Selim G. and Lyons, Kelly A. [1993]. *Parallel computational geometry*, Prentice-Hall, Englewood Cliffs, NJ.

Albrecht, Peter [1987]. A new theoretical approach to Runge–Kutta methods, *SIAM J. Numer. Anal. 24*, 391–406.

Albrecht, Peter [1996]. The Runge–Kutta theory in a nutshell, *SIAM J. Numer. Anal. 33*, 1712–1735.

Alefeld, Götz and Herzberger, Jürgen [1983]. *Introduction to interval computations*, translated from the German by Jon Rokne, Comput. Sci. Appl. Math., Academic Press, New York.

Alexander, Dan [1996]. Newton's method — or is it?, *Focus 16*, Newsletter of the Mathematical Association of America, no. 5, 32–33.

Alexander, Roger [1977]. Diagonally implicit Runge–Kutta methods for stiff o.d.e.'s, *SIAM J. Numer. Anal. 14*, 1006–1021.

Allaire, Grégoire and Kaber, Sidi Mahmoud [2008]. *Numerical linear algebra*, translated from the 2002 French original by Karin Trabelsi, Texts in Appl. Math., v. 55, Springer, New York.

Allenby, R. B. and Redfern, E. J. [1989]. *Introduction to number theory with computing*, Edward Arnold, London.

Allgower, Eugene L. and Georg, Kurt [2003]. *Introduction to numerical continuation methods*, reprint of the 1990 edition, Classics Appl. Math., v. 45, SIAM, Philadelphia, PA.

Ames, William F. [1992]. *Numerical methods for partial differential equations*, 3d ed., Comput. Sci. Sci. Comput., Academic Press, Boston, MA.

Anonymous [1996]. Inquiry board traces Ariane 5 failure to overflow error, *SIAM News 29*, no. 8, pp. 1, 12–13.

Anscombe, Francis John [1981]. *Computing in statistical science through APL*, Springer Ser. Statist., Springer, New York.

W. Gautschi, *Numerical Analysis*, DOI 10.1007/978-0-8176-8259-0,

Arias de Reyna, J. and Van de Lune, J. [2009]. High precision computation of a constant in the theory of trigonometric series, *Math. Comp. 78*, 2187–2191.

Ascher, Uri M. and Petzold, Linda R. [1998]. *Computer methods for ordinary differential equations and differential-algebraic equations*, SIAM, Philadelphia, PA.

Ascher, Uri M., Mattheij, Robert M. M., and Russell, Robert D. [1995]. *Numerical solution of boundary value problems for ordinary differential equations*, corrected reprint of the 1988 original, Classics Appl. Math., v. 13, SIAM, Philadelphia, PA.

Ash, J. Marshall and Jones, Roger L. [1981]. Optimal numerical differentiation using three function evaluations, *Math. Comp. 37*, 159–167.

Ash, J. Marshall, Janson, S., and Jones, R. L. [1984]. Optimal numerical differentiation using N function evaluations, *Calcolo 21*, 151–169.

Ashyralyev, A. and Sobolevskiĭ, P. E. [1994]. *Well-posedness of parabolic difference equations*, translated from the Russian by A. Iacob, Operator Theory: Adv. Appl., v. 69, Birkhäuser, Basel.

Askey, Richard [1972]. Positivity of the Cotes numbers for some Jacobi abscissas, *Numer. Math. 19*, 46–48.

Askey, R. [1979]. Positivity of the Cotes numbers for some Jacobi abscissas II, *J. Inst. Math. Appl. 24*, 95–98.

Askey, Richard and Fitch, James [1968]. Positivity of the Cotes numbers for some ultraspherical abscissas, *SIAM J. Numer. Anal. 5*, 199–201.

Atkinson, Kendall E. [1968]. On the order of convergence of natural cubic spline interpolation, *SIAM J. Numer. Anal. 5*, 89–101.

Atkinson, Kendall E. [1976]. *A survey of numerical methods for the solution of Fredholm integral equations of the second kind*, SIAM, Philadelphia, PA.

Atkinson, Kendall E. [1989]. *An introduction to numerical analysis*, 2d ed., Wiley, New York.

Atkinson, Kendall E. [1997]. *The numerical solution of integral equations of the second kind*, Cambridge Monographs Appl. Comput. Math., v. 4, Cambridge University Press, Cambridge.

Atkinson, Kendall and Han, Weimin [2009]. *Theoretical numerical analysis: A functional analysis framework*, 3d ed., Text in Appl. Math., v. 39, Springer, Dordrecht.

Atkinson, Kendall E., Han, Weimin, and Stewart, David [2009]. *Numerical solution of ordinary differential equations*, Pure Appl. Math., Wiley, Hoboken, NJ.

Atteia, Marc [1992]. *Hilbertian kernels and spline functions*, Stud. Comput. Math., v. 4, North-Holland, Amsterdam.

Axelsson, Owe [1969]. A class of A-stable methods, *BIT 9*, 185–199.

Axelsson, O. and Barker, V. A. [2001]. *Finite element solution of boundary value problems: Theory and computation*, reprint of the 1984 original, Classics Appl. Math., v. 35, SIAM, Philadelphia, PA.

Babuška, Ivo and Strouboulis, Theofanis [2001]. *The finite element method and its reliability*, Numer. Math. Scient. Comput., Oxford University Press, New York.

Babuška, Ivo, Whiteman, John R., and Strouboulis, Theofanis [2011]. *Finite elements: An introduction to the method and error estimation*, Oxford University Press, New York.

Bach, Eric and Shallit, Jeffrey [1996]. *Algorithmic number theory*, v. 1: *Efficient algorithms*, Found. Comput. Ser., MIT Press, Cambridge, MA.

Bailey, Paul B., Shampine, Lawrence F., and Waltman, Paul E. [1968]. *Nonlinear two point boundary value problems*, Math. Sci. Engrg., v. 44, Academic Press, New York.

Baker, Christopher T. H. [1977]. *The numerical treatment of integral equations*, Monographs Numer. Anal., Clarendon Press, Oxford.

Baker, George A., Jr. [1975]. *Essentials of Padé approximants*, Academic Press, New York.

Baker, George A., Jr. and Graves-Morris, Peter [1996]. *Padé approximants*, 2d ed., Encyclopedia Math. Appl., v. 59, Cambridge University Press, Cambridge.

Bartels, Richard H., Beatty, John C., and Barsky, Brian A. [1987]. *An introduction to splines for use in computer graphics and geometric modeling*, with forewords by Pierre Bézier and A. Robin Forrest, Morgan Kaufmann, Palo Alto, CA.

Barton, D., Willers, I. M., and Zahar, R. V. M. [1971]. The automatic solution of systems of ordinary differential equations by the method of Taylor series, *Comput. J. 14*, 243–248.

Bashforth, Francis and Adams, J. C. [1883]. *An attempt to test the theories of capillary action by comparing the theoretical and measured forms of drops of fluid, with an explanation of the method of integration employed in constructing the tables which give the theoretical forms of such drops*, Cambridge University Press, Cambridge.

Bauer, F. L., Rutishauser, H., and Stiefel, E. [1963]. New aspects in numerical quadrature, in *Proc. Sympos. Appl. Math. 15*, 199–218, Amer. Math. Soc., Providence, RI.

Beatson, R. K. [1986]. On the convergence of some cubic spline interpolation schemes, *SIAM J. Numer. Anal. 23*, 903–912.

Beatson, R. K. and Chacko, E. [1989]. A quantitative comparison of end conditions for cubic spline interpolation, in *Approximation theory VI*, v. 1, Charles K. Chui, L. L. Schumaker, and J. D. Ward, eds., 77–79, Academic Press, Boston, MA.

Beatson, R. K. and Chacko, E. [1992]. Which cubic spline should one use?, *SIAM J. Sci. Statist. Comput. 13*, 1009–1024.

Bellen, Alfredo and Zennaro, Marino [2003]. *Numerical methods for delay differential equations*, Numer. Math. Scient. Comput., Oxford University Press, New York.

Bellman, Richard and Cooke, Kenneth L. [1963]. *Differential-difference equations*, Academic Press, New York, 1963.

Bellman, Richard E. and Kalaba, Robert E. [1965]. *Quasilinearization and nonlinear boundary-value problems*, Modern Analyt. Comput. Methods in Sci. and Math., v. 3, American Elsevier, New York.

Bellomo, Nicola and Preziosi, Luigi [1995]. *Modelling mathematical methods and scientific computation*, CRC Math. Modelling Ser., CRC Press, Boca Raton, FL.

Beltzer, A. I. [1990]. *Variational and finite element methods: A symbolic computation approach*, Springer, Berlin.

Bernfeld, Stephen R. and Lakshmikantham, V. [1974]. *An introduction to nonlinear boundary value problems*, Math. Sci. Engrg., v. 109, Academic Press, New York.

Berrut, Jean-Paul [1984]. Baryzentrische Formeln zur trigonometrischen Interpolation I., *Z. Angew. Math. Phys. 35*, 91–105; Baryzentrische Formeln zur trigonometrischen Interpolation II. Stabilität und Anwendung auf die Fourieranalyse bei ungleichabständigen Stützstellen, *ibid.*, 193–205.

Berrut, Jean-Paul [1989]. Barycentric formulae for cardinal (SINC-) interpolants, *Numer. Math. 54*, 703–718. [Erratum: *ibid. 55* (1989), 747.]

Berrut, Jean-Paul and Trefethen, Lloyd N. [2004]. Barycentric Lagrange nterpolation, *SIAM Rev. 46*, 501–517.

Berrut, Jean-Paul, Baltensperger, Richard, and Mittelmann, Hans D. [2005]. Recent developments in barycentric rational interpolation, in *Trends and applications in constructive approximation*, Detlef H. Mache, József Szabados, and Marcel G. de Bruin, eds., 27–51, Internat. Ser. Numer. Math., v. 151, Birkhäuser, Basel.

Bettis, D. G. [1969/1970]. Numerical integration of products of Fourier and ordinary polynomials, *Numer. Math. 14*, 421–434.

Bini, Dario and Pan, Victor Y. [1994]. *Polynomial and matrix computations*, v. 1: *Fundamental algorithms*, Progr. Theoret. Comput. Sci., Birkhäuser, Boston, MA.

Birkhoff, Garrett and Lynch, Robert E. [1984]. *Numerical solution of elliptic problems*, SIAM Stud. Appl. Math., v. 6, SIAM, Philadelphia, PA.

Björck, Åke [1996]. *Numerical methods for least squares problems*, SIAM, Philadelphia, PA.

Blue, James L. [1979]. A Legendre polynomial integral, *Math. Comp. 33*, 739–741.

Bochev, Pavel B. and Gunzburger, Max D. [2009]. *Least-squares finite element methods*, Appl. Math. Sci., v. 166, Springer, New York.

Bojanov, B. D., Hakopian, H. A., and Sahakian, A. A. [1993]. *Spline functions and multivariate interpolations*, Math. Appl., v. 248, Kluwer, Dordrecht.

de Boor, Carl [2001]. *A practical guide to splines*, revised edition, Appl. Math. Sci., v. 27, Springer, New York.

de Boor, C., Höllig, K., and Riemenschneider, S. [1993]. *Box splines*, Appl. Math. Sci., v. 98, Springer, New York.

Borwein, Peter and Erdélyi, Tamás [1995]. *Polynomials and polynomial inequalities*, Graduate Texts in Math., v. 161, Springer, New York.

Braess, Dietrich [1986]. *Nonlinear approximation theory*, Springer Ser. Comput. Math., v. 7, Springer, Berlin.

Braess, Dietrich [2007]. *Finite elements: Theory, fast solvers, and applications in elasticity theory*, translated from the German by Larry L. Schumaker, 3d ed., Cambridge University Press, Cambridge.

Bramble, James H. [1993]. *Multigrid methods*, Pitman Res. Notes in Math. Ser., v. 294, Wiley, New York.

Branham, Richard L., Jr. [1990]. *Scientific data analysis: An introduction to overdetermined systems*, Springer, New York.

Brass, Helmut [1977]. *Quadraturverfahren*, Studia Mathematica, Skript 3, Vandenhoeck & Ruprecht, Göttingen.

Brebbia, C. A. and Dominguez, J. [1992]. *Boundary elements: An introductory course*, 2d ed., McGraw-Hill, New York.

Brenan, K. E., Campbell, S. L., and Petzold, L. R. [1996]. *Numerical solution of initial-value problems in differential-algebraic equations*, revised and corrected reprint of the 1989 original, Classics Appl. Math., v. 14, SIAM, Philadelphia, PA.

Brenner, Susanne C. and Scott, L. Ridgway [2008]. *The mathematical theory of finite element methods*, 3d ed., Texts in Appl. Math., v. 15, Springer, New York.

Brent, Richard P. [2002]. *Algorithms for minimization without derivatives*, reprint of the 1973 original, Dover Publ., Mineola, NY.

Brezinski, Claude [1997]. *Projection methods for systems of equations*, Studies in computational mathematics, v. 7, North-Holland, Amsterdam.

Brezinski, Claude and Redivo-Zaglia, Michela [1991]. *Extrapolation methods: Theory and practice*, Studies in computational mathematics, v. 2, North-Holland, Amsterdam.

Brezzi, Franco and Fortin, Michel [1991]. *Mixed and hybrid finite element methods*, Springer Ser. Comput. Math., v. 15, Springer, New York.

Briggs, William L., Henson, Van Emden, McCormick, Steve F. [2000]. *A multigrid tutorial*, 2d ed., SIAM, Philadelphia, PA.

Brock, P. and Murray, F. J. [1952]. The use of exponential sums in step by step integration I., II., *Math. Tables Aids Comput. 6*, 63–78; *ibid.*, 138–150.

Brown, Peter N., Byrne, George D., and Hindmarsh, Alan C. [1989]. VODE: A variable-coefficient ODE solver, *SIAM J. Sci. Statist. Comput. 10*, 1038–1051.

Broyden, Charles George and Vespucci, Maria Teresa [2004]. *Krylov solvers for linear algebraic systems*, Stud. Comput. Math., v. 11, Elsevier, Amsterdam.

Bruce, J. W., Giblin, P. J., and Rippon, P. J. [1990]. *Microcomputers and mathematics*, Cambridge University Press, Cambridge.

Brunner, Hermann [2004]. *Collocation methods for Volterra integral and related functional differential equations*, Cambridge Monographs Appl. Comput. Math., v. 15, Cambridge University Press, Cambridge.

Brunner, H. and van der Houwen, P. J. [1986]. *The numerical solution of Volterra equations*, CWI Monographs, v. 3, North-Holland, Amsterdam.

Brutman, L. [1997]. Lebesgue functions for polynomial interpolation: A survey, The heritage of P. L. Chebyshev: A Festschrift in honor of the 70th birthday of T. J. Rivlin, *Ann. Numer. Math. 4*, 111–127.

Brutman, L. and Passow, E. [1995]. On the divergence of Lagrange interpolation to $|x|$, *J. Approx. Theory 81*, 127–135.

Bultheel, Adhemar and Cools, Ronald, eds. [2010]. *The birth of numerical analysis*, World Scientific, Hackensack, NJ.

Burrage, Kevin [1978a]. A special family of Runge–Kutta methods for solving stiff differential equations, *BIT 18*, 22–41.

Burrage, Kevin [1978b]. High order algebraically stable Runge–Kutta methods, *BIT 18*, 373–383.

Burrage, Kevin [1982]. Efficiently implementable algebraically stable Runge–Kutta methods, *SIAM J. Numer. Anal. 19*, 245–258.

Burrage, Kevin [1995]. *Parallel and sequential methods for ordinary differential equations*, Numer. Math. Scient. Comput., Oxford Sci. Publ., The Clarendon Press, Oxford University Press, New York.

Burrus, C. Sidney, Gopinath, Ramesh A., and Guo, Haitao [1998]. *Introduction to wavelets and wavelet transforms: A primer*, with additional material and programs by Jan E. Odegard and Ivan W. Selesnick, Prentice Hall, Upper Saddle River, NJ.

Butcher, J. C. [1965]. On the attainable order of Runge–Kutta methods, *Math. Comp. 19*, 408–417.

Butcher, J. C. [1985]. The nonexistence of ten-stage eighth order explicit Runge–Kutta methods, *BIT 25*, 521–540.

Butcher, J. C. [1987]. *The numerical analysis of ordinary differential equations: Runge–Kutta and general linear methods*, Wiley-Interscience, Wiley, Chichester.

Butcher, J. C., guest ed. [1996]. Special issue celebrating the centenary of Runge–Kutta methods, *Appl. Numer. Math. 22*, nos. 1–3.

Butcher, J. C. [2008]. *Numerical methods for ordinary differential equations*, 2d ed., Wiley, Chichester.

Byrne, Charles L. [2008]. *Applied iterative methods*, A K Peters, Wellesley, MA.

Canuto, Claudio, Hussaini, M. Yousuff, Quarteroni, Alfio, and Zang, Thomas A. [1988]. *Spectral methods in fluid dynamics*, Springer Ser. Comput. Phys., Springer, New York.

Canuto, C., Hussaini, M. Y., Quarteroni, A., and Zang, T. A. [2006]. *Spectral methods: Fundamentals in single domains*, Scientific Computation, Springer, Berlin.

Canuto, C., Hussaini, M. Y., Quarteroni, A., and Zang, T. A. [2007]. *Spectral methods: Evolution to complex geometries and applications to fluid dynamics*, Scientific Computation, Springer, Berlin.

Celia, Michael A. and Gray, William G. [1992]. *Numerical methods for differential equations: Fundamental concepts for scientific and engineering applications*, Prentice-Hall, Englewood Cliffs, NJ.

Chaitin-Chatelin, Françoise and Frayssé, Valérie [1996]. *Lectures on finite precision computations*, with a foreword by Iain S. Duff, Software, Environments, and Tools, SIAM, Philadelphia, PA.

Chan, Raymond Hon-Fu and Jin, Xiao-Qing [2007]. *An introduction to iterative Toeplitz solvers*, Fundamentals of Algorithms, v. 5, SIAM, Philadelphia, PA.

Chatelin, Françoise [1983]. *Spectral approximation of linear operators*, with a foreword by P. Henrici and solutions to exercises by Mario Ahués, Comput. Sci. Appl. Math., Academic Press, New York.

Chatelin, Françoise [1993]. *Eigenvalues of matrices*, with exercises by Mario Ahués; translated from the French and with additional material by Walter Ledermann, Wiley, Chichester.

Chebyshev, P. L. [1859]. Sur l'interpolation par la méthode des moindres carrés, *Mém. Acad. Impér. Sci. St. Petersbourg (7) 1*, no. 15, 1–24. [Œuvres, v. 1, 473–498.]

Chen, Goong and Zhou, Jian-Xin [1992]. *Boundary element methods*, Comput. Math. Appl., Academic Press, London.

Chen, Zhangxin [2005]. *Finite element methods and their applications*, Scient. Comput., Springer, Berlin.

Cheney, E. W. [1998]. *Introduction to approximation theory*, reprint of the second (1982) edition, AMS Chelsea Publishing, Providence, RI.

Cheney, E. W. [1986]. *Multivariate approximation theory: Selected topics*, CBMS-NSF Regional Conf. Ser. Appl. Math., no. 51, SIAM, Philadelphia, PA.

Cheney, Ward and Kincaid, David [1994]. *Numerical mathematics and computing*, 3d ed., Brooks/Cole, Pacific Grove, CA.

Chihara, T. S. [1978]. *An introduction to orthogonal polynomials*, Math. Appl., v. 13, Gordon and Breach, New York.

Chu, Moody T. and Golub, Gene H. [2005]. *Inverse eigenvalue problems: Theory, algorithms, and applications*, Numer. Math. Scient. Comput., Oxford University Press, New York.

Chui, Charles K. [1988]. *Multivariate splines*, with an appendix by Harvey Diamond, CBMS-NSF Regional Conf. Ser. Appl. Math., no. 54, SIAM, Philadelphia, PA.

Chui, Charles K. [1992]. *An introduction to wavelets*, Wavelet Analysis and Its Applications, v. 1, Academic Press, Boston, MA.

Ciarlet, Philippe G. [2002]. *The finite element method for elliptic problems*, reprint of the 1978 original, Classics Appl. Math., v. 40, SIAM, Philadelphia, PA.

Ciarlet, Philippe G. [1989]. *Introduction to numerical linear algebra and optimisation*, with the assistance of Bernadette Miara and Jean-Marie Thomas; translated from the French by A. Buttigieg, Cambridge Texts Appl. Math., Cambridge University Press, Cambridge.

Ciarlet, P. G. and Lions, J.-L., eds. [1990–2003]. *Handbook of numerical analysis*, vols. 1–9, North-Holland, Amsterdam.

Coddington, Earl A. and Levinson, Norman [1955]. *Theory of ordinary differential equations*, Internat. Ser. Pure Appl. Math., McGraw-Hill, New York.

Coe, Tim, Mathisen, Terje, Moler, Cleve, and Pratt, Vaughan [1995]. Computational aspects of the Pentium affair, *IEEE Comput. Sci. Engrg. 2*, no. 1, 18–30.

Cohen, A. M. [1994]. Is the polynomial so perfidious?, *Numer. Math. 68*, 225–238.

Cohen, Elaine, Riesenfeld, Richard F., and Elber, Gershon [2001]. *Geometrix modeling with splines: An introduction*, with a foreword by Tom Lyche, A K Peters, Natick, MA.

Cohen, Henri [1993]. *A course in computational algebraic number theory*, Grad. Texts Math., v. 138, Springer, Berlin.

Conte, Samuel D. and de Boor, Carl [1980]. *Elementary numerical analysis: An algorithmic approach*, 3d ed., McGraw-Hill, New York.

Cools, Ronald and Rabinowitz, Philip [1993]. Monomial cubature rules since "Stroud": A compilation, *J. Comput. Appl. Math. 48*, 309–326.

Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L., and Stein, Clifford [2009]. *Introduction to algorithms*, 3d ed., MIT Press, Cambridge, MA.

Cox, David, Little, John, and O'Shea, Donal [2007]. *Ideals, varieties, and algorithms: An introduction to computational algebraic geometry and commutative algebra*, 3d ed., Undergrad. Texts Math., Springer, New York.

Crandall, Richard E. [1994]. *Projects in scientific computation*, Springer, New York.

Crandall, Richard E. [1996]. *Topics in advanced scientific computation*, Springer, New York.

Crouzeix, Michel [1976]. Sur les méthodes de Runge–Kutta pour l'approximation des problèmes d'évolution, in *Computing methods in applied sciences and engineering*, Lecture Notes in Econom. and Math. Systems, no. 134, Springer, Berlin, 206–223.

Crouzeix, Michel and Ruamps, Françoise [1977]. On rational approximations to the exponential, *RAIRO Numer. Anal. 11* (1977), 214–243.

Cryer, Colin W. [1972]. *Numerical methods for functional differential equations*, in *Delay and functional differential equations and their applications*, Klaus Schmitt, ed., 17–101, Academic Press, New York.

Cullum, Jane K. and Willoughby, Ralph A. [2002]. *Lanczos algorithms for large symmetric eigenvalue computations*, v. 1: *Theory*, reprint of the 1985 original, Classics Appl. Math., v. 41, SIAM, Philadelphia, PA.

Cullum, Jane K. and Willoughby, Ralph A. [1985]. *Lanczos algorithms for large symmetric eigenvalue computations*, v. 2: *Programs*, Progress in Scient. Comput., v. 4, Birkhäuser, Boston, MA.

Curtiss, C. F. and Hirschfelder, J. O. [1952]. Integration of stiff equations, *Proc. Nat. Acad. Sci. U.S.A. 38*, 235–243.

Dahlquist, Germund [1956]. Convergence and stability in the numerical integration of ordinary differential equations, *Math. Scand. 4*, 33–53.

Dahlquist, Germund [1959]. *Stability and error bounds in the numerical integration of ordinary differential equations*, Kungl. Tekn. Högsk. Handl. Stockholm, no. 130.

Dahlquist, Germund G. [1963]. A special stability problem for linear multistep methods, *BIT 3*, 27–43.

Dahlquist, Germund [1985]. 33 years of numerical instability. I., *BIT 25*, 188–204.

Dahlquist, Germund and Björck, Åke [2003]. *Numerical methods*, translated from the Swedish by Ned Anderson; reprint of the 1974 English translation, Dover Publ., Mineola, NY.

Dahlquist, Germund and Björck, Åke [2008]. *Numerical methods in scientific computing*, v. 1, SIAM, Philadelphia, PA.

Danaila, Ionut, Joly, Pascal, Kaber, Sidi Mahmoud, and Postel, Marie [2007]. *An introduction to scientific computing: Twelve computational projects solved with MATLAB*, Springer, New York.

Daniel, James W. and Moore, Ramon E. [1970]. *Computation and theory in ordinary differential equations*, Freeman, San Francisco, CA.

Datta, Biswa Nath [2010]. *Numerical linear algebra and applications*, 2d ed., SIAM, Philadelphia, PA.

Daubechies, Ingrid [1992]. *Ten lectures on wavelets*, CBMS-NSF Regional Conf. Ser. Appl. Math., no. 61, SIAM, Philadelphia, PA.

Davenport, J. H., Siret, Y., and Tournier, E. [1993]. *Computer algebra: Systems and algorithms for algebraic computation*, 2d ed., with a preface by Daniel Lazard; translated from the French by A. Davenport and J. H. Davenport; with a foreword by Antony C. Hearn, Academic Press, London.

Davis, Philip J. [1975]. *Interpolation and approximation*, republication with minor corrections of the 1963 original, with a new preface and bibliography, Dover Publ., New York.

Davis, Philip J. and Rabinowitz, Philip [2007]. *Methods of numerical integration*, corrected reprint of the second (1984) edition, Dover Publ., Mineola, NY.

Davis, Timothy A. [2006]. *Direct methods for sparse linear systems*, Fundamentals of Algorithms, v. 2, SIAM, Philadelphia, PA.

Dekker, T. J. [1969]. Finding a zero by means of successive linear interpolation, in *Constructive aspects of the fundamental theorem of algebra*, Bruno Dejon and Peter Henrici, eds., 37–48, Wiley-Interscience, New York.

Dekker, K. and Verwer, J. G. [1984]. *Stability of Runge–Kutta methods for stiff nonlinear differential equations*, CWI Monographs, v. 2, North-Holland, Amsterdam.

Demkowicz, Leszek [2007]. *Computing with hp-adaptive finite elements*, v. 1: *One and two dimensional elliptic and Maxwell problems*, with 1 CD-ROM (UNIX), Chapman & Hall/CRC Appl. Math. Nonlin. Sci. Ser., Chapman & Hall/CRC, Boca Raton, FL.

Demmel, James W. [1997]. *Applied numerical linear algebra*, SIAM, Philadelphia.

Dennis, J. E., Jr. and Schnabel, Robert B. [1996]. *Numerical methods for unconstrained optimization and nonlinear equations*, corrected reprint of the 1983 original, Classics Appl. Math., v. 16, SIAM, Philadelphia, PA.

Deuflhard, Peter and Bornemann, Folkmar [2002]. *Scientific computing with ordinary differential equations*, translated from the 1994 German original by Werner C. Rheinboldt, Texts in Appl. Math., v. 42, Springer, New York.

Deuflhard, Peter and Hohmann, Andreas [2003]. *Numerical analysis in modern scientific computing*, 2d ed., Texts in Appl. Math., v. 43, Springer, New York.

DeVore, Ronald A. and Lorentz, George G. [1993]. *Constructive approximation*, Grundlehren Math. Wiss., v. 303, Springer, Berlin.

Devroye, Luc [1986]. *Nonuniform random variate generation*, Springer, New York.

Dierckx, Paul [1993]. *Curve and surface fitting with splines*, Monographs Numer. Anal., Oxford Sci. Publ., Oxford University Press, New York.

Dongarra, Jack J., Duff, Iain S., Sorensen, Danny C., and van der Vorst, Henk A. [1991]. *Solving linear systems on vector and shared memory computers*, SIAM, Philadelphia, PA.

Dongarra, Jack J., Duff, Iain S., Sorensen, Danny C., and van der Vorst, Henk A. [1998]. *Numerical linear algebra for high-performance computers*, Software, Environments, and Tools, v. 7, SIAM, Philadelphia, PA.

Dormand, J. R. and Prince, P. J. [1980]. A family of embedded Runge–Kutta formulae, *J. Comput. Appl. Math. 6*, 19–26.

Driscoll, Tobin A. [2009]. *Learning MATLAB*, SIAM, Philadelphia, PA.

Driver, R. D. [1977]. *Ordinary and delay differential equations*, Appl. Math. Sci., v. 20, Springer, New York.

Duff, I. S., Erisman, A. M., and Reid, J. K. [1989]. *Direct methods for sparse matrices*, 2d ed., Monographs Numer. Anal., Oxford Sci. Publ., Oxford University Press, New York.

Durand, E. [1960, 1961]. *Solutions numériques des équations algébriques*; v. 1: *Équations du type $F(x) = 0$, racines d'un polynôme*; v. 2: *Systèmes de plusieurs équations, valeurs propres des matrices*, Masson, Paris.

Dutka, Jacques [1984]. Richardson extrapolation and Romberg integration, *Historia Math. 11*, 3–21.

Edelman, Alan [1997]. The mathematics of the Pentium division bug, *SIAM Rev. 39*, 54–67.

Edelsbrunner, Herbert [1987]. *Algorithms in combinatorial geometry*, EATCS Monographs Theoret. Comput. Sci., v. 10, Springer, Berlin.

Edelsbrunner, Herbert [2006]. *Geometry and topology for mesh generation*, reprint of the 2001 original, Cambridge Monographs Appl. Comput. Math., v. 7, Cambridge University Press, Cambridge.

Edelsbrunner, Herbert and Harer, John L. [2010]. *Computational topology: An introduction*, Amer. Math. Soc., Providence, RI.

Efendiev, Yalchin and Hou, Thomas Y. [2009]. *Multiscale finite element methods: Theory and applications*, Surveys Tutor. Appl. Math. Sci., v. 4, Springer, New York.

Ehle, Byron L. [1968]. High order A-stable methods for the numerical solution of systems of D. E.'s, *BIT 8*, 276–278.

Ehle, Byron L. [1973]. A-stable methods and Padé approximations to the exponential, *SIAM J. Math. Anal. 4*, 671–680.

Eijgenraam, P. [1981]. *The solution of initial value problems using interval arithmetic: Formulation and analysis of an algorithm*, Mathematical Centre Tracts, v. 144, Mathematisch Centrum, Amsterdam.

Elaydi, Saber [2005]. *An introduction to difference equations*, 3d ed., Undergrad. Texts Math., Springer, New York.

Elhay, Sylvan and Kautsky, Jaroslav [1987]. Algorithm 655 — IQPACK: FORTRAN subroutines for the weights of interpolatory quadratures, *ACM Trans. Math. Software 13*, 399–415.

Elman, Howard C., Silvester, David J., and Wathen, Andrew J. [2005]. *Finite elements and fast iterative solvers: With applications in incompressible fluid dynamics*, Numer. Math. Scient. Comput., Oxford University Press, New York.

Engels, Hermann [1979]. Zur Geschichte der Richardson-Extrapolation, *Historia Math. 6*, 280–293.

Engels, H. [1980]. *Numerical quadrature and cubature*, Comput. Math. Appl., Academic Press, London.

England, R. [1969/1970]. Error estimates for Runge–Kutta type solutions to systems of ordinary differential equations, *Comput. J. 12*, 166–170.

Epperson, James F. [1987]. On the Runge example, *Amer. Math. Monthly 94*, 329–341.

Erdös, P. and Vértesi, P. [1980]. On the almost everywhere divergence of Lagrange interpolatory polynomials for arbitrary system of nodes, *Acta Math. Acad. Sci. Hungar. 36*, 71–89.

Eriksson, K., Estep, D., Hansbo, P., and Johnson, C. [1996]. *Computational differential equations*, Cambridge University Press, Cambridge.

Ern, Alexandre and Guermonde, Jean-Luc [2004]. *Theory and practice of finite elements*, Appl. Math. Sci., v. 159, Springer, Berlin.

Euler, Leonhard [1768]. *Institutionum calculi integralis*, v. 1, sec. 2, Ch. 7: De integratione aequationum differentialium per approximationem, *Impenfis Academiae Imperialis Scientiarum*, Petropoli. [Opera Omnia, ser. 1, v. 11, 424–434.]

Evans, Gwynne [1993]. *Practical numerical integration*, Wiley, Chichester.

Evans, Michael and Swartz, Tim [2000]. *Approximating integrals via Monte Carlo and deterministic methods*, Oxford Stat. Sci. Ser., Oxford University Press, Oxford.

Evtushenko, Yurij G. [1985]. *Numerical optimization techniques*, translated from the Russian; translation edited and with a foreword by J. Stoer, Transl. Ser. Math. Engrg., Springer, New York.

Fang, K.-T. and Wang, Y. [1994]. *Number-theoretic methods in statistics*, Monographs Statist. Appl. Probab., v. 51, Chapman & Hall, London.

Farin, Gerald [1997]. *Curves and surfaces for computer aided geometric design: A practical guide*, 4th ed., Chapter 1 by P. Bézier; Chapters 11 and 22 by W. Boehm, Comput. Sci. Scient. Comput., Academic Press, San Diego, CA.

Farin, Gerald E. [1999]. *NURBS: From projective geometry to practical use*, 2d ed., A K Peters, Natick, MA.

Farin, Gerald and Hansford, Dianne [2008]. *Mathematical principles for scientific computing and visualization*, A K Peters, Wellesley, MA.

Fassbender, Heike [2000]. *Symplectic methods for the symplectic eigenproblem*, Kluwer Academic/Plenum Publ., New York.

Feagin, Terry [2011]. Personal communication, email, June 28. For coefficients, see http://sce.uhcl.edu/rungekutta.

Fehlberg, E. [1969]. Klassische Runge–Kutta-Formeln fünfter und siebenter Ordnung mit Schrittweiten-Kontrolle, *Computing 4*, 93–106. [Corrigendum: *ibid. 5*, 184.]

Fehlberg, E. [1970]. Klassische Runge–Kutta-Formeln vierter und niedrigerer Ordnung mit Schrittweiten-Kontrolle und ihre Anwendung auf Wärmeleitungs-probleme, *Computing 6*, 61–71.

Feinerman, Robert P. and Newman, Donald J. [1974]. *Polynomial approximation*, Williams & Wikins, Baltimore, MD.

Fejér, L. [1918]. Interpolation und konforme Abbildung, *Göttinger Nachrichten*, 319–331. [Ges. Arbeiten, v. 2, 100–111, Akadémiai Kiadó, Budapest, 1970.]

Fejér, L. [1933]. Mechanische Quadraturen mit positiven Cotesschen Zahlen, *Math. Z. 37*, 287–309. [Ges. Arbeiten, v. 2, Akadémiai Kiadó, Budapest, 1970, 457–478.]

Fischer, Bernd [1996]. *Polynomial based iteration methods for symmetric linear systems*, Wiley-Teubner Ser. Advances Numer. Math., Wiley, Chichester.

Fischer, J.-W. [2002]. Romberg quadrature using the Bulirsch sequence, *Numer. Math. 90*, 509–519.

Fletcher, R. [2001]. *Practical methods of optimization*, paperback reprinting of the 1987 second edition, Wiley-Interscience, New York.

Fornberg, Bengt [1981]. Numerical differentiation of analytic functions, *ACM Trans. Math. Software 7*, 512–526; Algorithm 579 — CPSC: Complex power series coefficients, *ibid.* 542–547.

Fornberg, Bengt [1996]. *A practical guide to pseudospectral methods*, Cambridge Monographs Appl. Comput. Math., v. 1, Cambridge University Press, Cambridge.

Förster, Klaus-Jürgen [1993]. Variance in quadrature: A survey, in *Numerical integration IV*, H. Brass and G. Hämmerlin, eds., 91–110, Internat. Ser. Numer. Math., v. 112, Birkhäuser, Basel.

Forsythe, George E. [1957]. Generation and use of orthogonal polynomials for data-fitting with a digital computer, *J. Soc. Indust. Appl. Math. 5*, 74–88.

Forsythe, George E. [1958]. Singularity and near singularity in numerical analysis, *Amer. Math. Monthly 65*, 229–240.

Forsythe, George E. [1970]. Pitfalls in computation, or why a math book isn't enough, *Amer. Math. Monthly 77*, 931–956.

Forsythe, George E. and Moler, Cleve B. [1967]. *Computer solution of linear algebraic systems*, Prentice-Hall, Englewood Cliffs, NJ.

Fox, L. [1990]. *The numerical solution of two-point boundary problems in ordinary differential equations*, reprint of the 1957 original, Dover Publ., New York.

Freeman, T.L. and Phillips, C. [1992]. *Parallel numerical algorithms*, Prentice-Hall Internat. Ser. Comput. Sci., Prentice-Hall International, New York.

Freud, Géza [1971]. *Orthogonal polynomials*, translated from the German by I. Földes, Pergamon Press, Oxford.

Fröberg, Carl-Erik [1985]. *Numerical mathematics: Theory and computer applications*, Benjamin/Cummings, Menlo Park, CA.

Gaier, Dieter [1987]. *Lectures on complex approximation*, translated from the German by Renate McLaughlin, Birkhäuser, Boston, MA.

Gallier, Jean [2000]. *Curves and surfaces in geometric modeling: Theory and algorithms*, The Morgan Kaufmann Ser. Comput. Graphics Geom. Modeling, Morgan Kaufmann, San Francisco, CA.

Garfunkel, S. and Steen, L. A., eds. [1988]. *For all practical purposes: An introduction to contemporary mathematics*, Freeman, New York.

Gauss, K. F. [1995]. *Theory of the combination of observations least subject to errors, Part* 1, *Part* 2, *Supplement*, translated from the Latin by G. W. Stewart, Classics Appl. Math., v. 11, SIAM, Philadelphia, PA.

Gautschi, Walter [1961]. Numerical integration of ordinary differential equations based on trigonometric polynomials, *Numer. Math. 3*, 381–397.

Gautschi, Walter [1963]. On error reducing multi-step methods (abstract), *Notices Amer. Math. Soc. 10*, no. 1, Part I, 95.

Gautschi, Walter [1971/1972]. Attenuation factors in practical Fourier analysis, *Numer. Math. 18*, 373–400.

Gautschi, Walter [1973]. On the condition of algebraic equations, *Numer. Math. 21*, 405–424.

Gautschi, Walter [1975a]. Computational methods in special functions: A survey, in *Theory and application of special functions*, Richard A. Askey, ed., 1–98, Math. Res. Center, Univ. Wisconsin Publ., no. 35, Academic Press, New York.

Gautschi, Walter [1975b]. Stime dell'errore globale nei metodi "one-step" per equazioni differenziali ordinarie, collection of articles dedicated to Mauro Picone, *Rend. Mat. (6) 8*, 601–617.

Gautschi, Walter [1976a]. Advances in Chebyshev quadrature, in *Numerical analysis Dundee 1975*, G. A. Watson, ed., 100–121, Lecture Notes in Math., v. 506, Springer, Berlin.

Gautschi, Walter [1976b]. Comportement asymptotique des coefficients dans les formules d'intégration d'Adams, de Störmer et de Cowell, *C. R. Acad. Sci. Paris Sér. A-B 283*, A787–A788.

Gautschi, Walter [1981]. A survey of Gauss–Christoffel quadrature formulae, in *E. B. Christoffel: The influence of his work on mathematics and the physical sciences*, P. L. Butzer and F. Fehér, eds., 72–147, Birkhäuser, Basel.

Gautschi, Walter [1982]. On generating orthogonal polynomials, *SIAM J. Sci. Statist. Comput. 3*, 289–317.

Gautschi, Walter [1983]. On the convergence behavior of continued fractions with real elements, *Math. Comp. 40*, 337–342.

Gautschi, Walter [1984]. Questions of numerical condition related to polynomials, in *MAA Stud. Math.*, v. 24: *Studies in numerical analysis*, Gene H. Golub, ed., 140–177, Math. Assoc. America, Washington, DC.

Gautschi, Walter [1988]. Gauss–Kronrod quadrature: A survey, in *Numerical methods and approximation theory III*, G. V. Milovanović, ed., 39–66, University of Niš, Niš.

Gautschi, Walter [1990]. How (un)stable are Vandermonde systems?, in *Asymptotic and computational analysis*, R. Wong, ed., 193–210, Lecture Notes in Pure Appl. Math., v. 124, Dekker, New York.

Gautschi, Walter [1993]. Gauss-type quadrature rules for rational functions, in *Numerical integration IV*, H. Brass and G. Hämmerlin, eds., 111–130, Internat. Ser. Numer. Math., v. 112, Birkhäuser, Basel.

Gautschi, Walter, ed. [1994a]. *Mathematics of Computation 1943–1993: A half-century of computational mathematics*, Proc. Sympos. Appl. Math., v. 48, Amer. Math. Soc., Providence, RI.

Gautschi, Walter [1994b]. Algorithm 726: ORTHPOL — a package of routines for generating orthogonal polynomials and Gauss-type quadrature rules, *ACM Trans. Math. Software 20*, 21–62.

Gautschi, Walter [1996]. Orthogonal polynomials: Applications and computation, in *Acta numerica 1996*, A. Iserles, ed., 45–119, Cambridge University Press, Cambridge.

Gautschi, Walter [1997]. Moments in quadrature problems, Approximation theory and applications, *Comput. Math. Appl. 33*, no. 1–2, 105–118.

Gautschi, Walter [2001]. Remark on "Barycentric formulae for cardinal (SINC-) interpolants" by J.-R. Berrut, *Numer. Math. 87*, 791–792.

Gautschi, Walter [2004]. *Orthogonal polynomials: Computation and approximation*. Numer. Math. Scient. Comput., Oxford Sci. Publ., Oxford University Press, New York.

Gautschi, Walter [2008]. On Euler's attempt to compute logarithms by interpolation: A commentary to his letter of February 16, 1734 to Daniel Bernoulli, *J. Comput. Appl. Math. 219*, 408–415.

Gautschi, Walter [2010]. The spiral of Theodorus, numerical analysis, and special functions, *J. Comput. Appl. Math. 235*, 1042–1052.

Gautschi, Walter [2011a]. Experimental mathematics involving orthogonal polynomials, in *Approximation and Computation: In honor of Gradimir V. Milovanović*, W. Gautschi, G. Mastroianni, and Th. M. Rassias, eds., 115–131, Springer Ser. Optim. Appl., v. 42, Springer, New York.

Gautschi, Walter [2011b]. Optimally scaled and optimally conditioned Vandermonde and Vandermonde-like matrices, *BIT Numer. Math. 51*, 103–125.

Gautschi, Walter and Milovanović, Gradimir V. [1997]. s-orthogonality and construction of Gauss–Turán type quadrature formulae, special issue dedicated to William B. Gragg, *J. Comput. Appl. Math. 86*, 205–218.

Gautschi, W. and Montrone, M. [1980]. Metodi multistep con minimo coefficiente dell'errore globale, *Calcolo 17*, 67–75.

Gear, C. W. [1969]. The automatic integration of stiff ordinary differential equations, in *Information processing 68*, v. 1: *Mathematics, Software*, A. J. H. Morrell, ed., 187–193, North-Holland, Amsterdam.

Gear, C. William [1971a]. *Numerical initial value problems in ordinary differential equations*, Prentice-Hall Ser. Autom. Comput., Prentice-Hall, Englewood Cliffs, NJ.

Gear, C. W. [1971b]. The automatic integration of ordinary differential equations, *Comm. ACM 14*, 176–179.

Gear, C. W. [1971c]. Algorithm 407 — DIFSUB for solution of ordinary differential equations, *Comm. ACM 14*, 185–190.

Geddes, K. O., Czapor, S. R., and Labahn, G. [1992]. *Algorithms for computer algebra*, Kluwer, Boston, MA.

Gentle, James E. [2003]. *Random number generation and Monte Carlo methods*, 2d ed., Stat. Comput., Springer, New York.

Gentle, James E. [2009]. *Computational statistics*, Springer, New York.

Gericke, Helmuth [1970]. *Geschichte des Zahlbegriffs*, Bibliographisches Institut, Mannheim.

Ghizzetti, A. and Ossicini, A. [1970]. *Quadrature formulae*, Academic Press, New York.

Gibbons, A. [1960]. A program for the automatic integration of differential equations using the method of Taylor series, *Comput. J. 3*, 108–111.

Gil, Amparo, Segura, Javier, and Temme, Nico M. [2007]. *Numerical methods for special functions*, SIAM, Philadelphia, PA.

Gill, Philip E., Murray, Walter, and Wright, Margaret H. [1981]. *Practical optimization*, Academic Press, London.

Gill, Philip E., Murray, Walter, and Wright, Margaret H. [1991]. *Numerical linear algebra and optimization*, v. 1, Addison-Wesley, Redwood City, CA.

Godlewski, Edwige and Raviart, Pierre-Arnaud [1996]. *Numerical approximation of hyperbolic systems of conservation laws*, Appl. Math. Sci., v. 118, Springer, New York.

Goldberg, Samuel [1986]. *Introduction to difference equations: With illustrative examples from economics, psychology, and sociology*, 2d ed., Dover Publ., New York.

Goldstine, Herman H. [1977]. *A history of numerical analysis from the 16th through the 19th century*, Stud. Hist. Math. Phys. Sci., v. 2, Springer, New York.

Golub, Gene H. [1973]. Some modified matrix eigenvalue problems, *SIAM Rev. 15*, 318–334.

Golub, Gene H. and Ortega, James M. [1992]. *Scientific computing and differential equations: An introduction to numerical methods*, Academic Press, Boston, MA.

Golub, Gene and Ortega, James M. [1993]. *Scientific computing: An introduction with parallel computing*, Academic Press, Boston, MA.

Golub, Gene H. and Van Loan, Charles F. [1996]. *Matrix computations*, 3d ed., Johns Hopkins Stud. Math. Sci., v. 3, Johns Hopkins University Press, Baltimore, MD.

Golub, Gene H. and Welsch, John H. [1969]. Calculation of Gauss quadrature rules, *Math. Comp. 23*, 221–230. Loose microfiche suppl. A1–A10.

Golub, Gene H. and Meurant, Gérard [2010]. *Matrices, moments, and quadrature with applications*, Princeton Ser. Appl. Math., Princeton University Press, Princeton, NJ.

Gonnet, Gaston H. and Scholl, Ralf [2009]. *Scientific computation*, Cambridge University Press, Cambridge.

Gottlieb, David and Orszag, Steven A. [1977]. *Numerical analysis of spectral methods: Theory and applications*, CBMS-NSF Regional Conf. Ser. Appl. Math., no. 26, SIAM, Philadelphia, PA.

Gragg, William B. [1964]. *Repeated extrapolation to the limit in the numerical solution of ordinary differential equations*, Ph.D. thesis, University of California, Los Angeles, CA.

Gragg, William B. [1965]. On extrapolation algorithms for ordinary initial value problems, *J. Soc. Indust. Appl. Math. Ser. B Numer. Anal. 2*, 384–403.

Greenbaum, Anne [1997]. *Iterative methods for solving linear systems*, Frontiers Appl. Math., v. 17, SIAM, Philadelphia, PA.

Greenspan, Donald and Casulli, Vincenzo [1988]. *Numerical analysis for applied mathematics, science, and engineering*, Addison-Wesley, Advanced Book Program, Redwood City, CA.

Gregory, Robert T. and Karney, David L. [1978]. *A collection of matrices for testing computational algorithms*, corrected reprint of the 1969 edition, Krieger, Huntington, NY.

Griepentrog, Eberhard and März, Roswitha [1986]. *Differential-algebraic equations and their numerical treatment*, Teubner Texts Math., v. 88, Teubner, Leipzig.

Griewank, Andreas and Corliss, George F., eds. [1991]. *Automatic differentiation of algorithms: Theory, implementation, and application*, SIAM Proc. Ser., SIAM, Philadelphia, PA.

Griewank, Andreas and Walther, Andrea [2008]. *Evaluating derivatives: Priciples and techniques of algorithmic differentiation*, 2d ed., SIAM, Philadelphia, PA.

Guo, Ben-Yu [1998]. *Spectral methods and their applications*, World Scientific, River Edge, NJ.

Gustafsson, Bertil [2008]. *High order difference methods for time dependent PDE*, Springer Ser. Comput. Math., v. 38, Springer, Berlin.

Gustafsson, Bertil, Kreiss, Heinz-Otto, and Oliger, Joseph [1995]. *Time dependent problems and difference methods*, Pure Appl. Math., Wiley-Interscience, Wiley, New York.

Hackbusch, Wolfgang [1994]. *Iterative solution of large sparse systems of equations*, translated and revised from the 1991 German original, Appl. Math. Sci., v. 95, Springer, New York.

Hageman, Louis A. and Young, David M. [2004]. *Applied iterative methods*, unabridged republication of the 1981 original, Dover Publ., Mineola, NY. York.

Hairer, E. and Wanner, G. [2010]. *Solving ordinary differential equations. II: Stiff and differential-algebraic problems*, 2d rev. ed., Springer Ser. Comput. Math., v. 14, Springer, Berlin.

Hairer, Ernst, Lubich, Christian, and Roche, Michel [1989]. *The numerical solution of differential-algebraic systems by Runge–Kutta methods*, Lecture Notes in Math., v. 1409, Springer, Berlin.

Hairer, Ernst, Lubich, Christian, and Wanner, Gerhard [2006]. *Geometric numerical integration: Structure-preserving algorithms for ordinary differential equations*, 2d ed., Springer Ser. Comput. Math., v. 31, Springer, Berlin.

Hairer, E., Nørsett, S. P., and Wanner, G. [1993]. *Solving ordinary differential equations. I: Nonstiff problems*, 2d ed., Springer Ser. Comput. Math., v. 8, Springer, Berlin.

Hall, Charles A. and Meyer, W. Weston [1976]. Optimal error bounds for cubic spline interpolation, *J. Approx. Theory 16*, 105–122.

Hall, Charles A. and Porsching, Thomas A. [1990]. *Numerical analysis of partial differential equations*, Prentice-Hall, Englewood Cliffs, NJ.

Hall, W. S. [1994]. *The boundary element method*, Solid Mech. Appl., v. 27, Kluwer, Dordrecht.

Hammer, R., Hocks, M., Kulisch, U., and Ratz, D. [1993]. *Numerical toolbox for verified computing. I: Basic numerical problems, theory, algorithms, and Pascal-XSC programs*, Springer Ser. Comput. Math., v. 21, Springer, Berlin.

Hammer, R., Hocks, M., Kulisch, U., and Ratz, D. [1995]. *C^{++} toolbox for verified computing. I: Basic numerical problems, theory, algorithms, and programs*, Springer, Berlin.

Hämmerlin, Günther and Hoffmann, Karl-Heinz [1991]. *Numerical mathematics*, translated from the German by Larry Schumaker, Undergrad. Texts Math., Readings in Math., Springer, New York.

Hansen, Eldon and Walster, G. William [2004]. *Global optimization using interval analysis*; 2d ed., revised and expanded, with a foreword by Ramon Moore, Monographs and Textbooks Pure Appl. Math., v. 264, Dekker, New York.

Hartmann, Friedel [1989]. *Introduction to boundary elements: Theory and applications*, with an appendix by Peter Schoepp, Springer, Berlin.

Heath, Michael T. [1997]. *Scientific computing: An introductory survey*, McGraw-Hill Ser. Comput. Sci., McGraw-Hill, New York.

Heck, André [2003]. *Introduction to Maple*, 3d ed., Springer, New York.

Heiberger, Richard M. [1989]. *Computation for the analysis of designed experiments*, Wiley Ser. Probab. Math. Stat.: Appl. Probab. Stat., Wiley, New York.

Henrici, Peter [1962]. *Discrete variable methods in ordinary differential equations*, Wiley, New York.

Henrici, Peter [1977]. *Error propagation for difference methods*, reprint of the 1963 edition, Krieger, Huntington, NY.

Henrici, Peter [1979a]. Fast Fourier methods in computational complex analysis, *SIAM Rev. 21*, 481–527.

Henrici, Peter [1979b]. Barycentric formulas for interpolating trigonometric polynomials and their conjugates, *Numer. Math. 33*, 225–234.

Henrici, Peter [1988, 1991, 1986]. *Applied and computational complex analysis*. v. 1: *Power series — integration — conformal mapping — location of zeros*, reprint of the 1974 original, Wiley Classics Lib., Wiley-Interscience, Wiley, New York; v. 2: *Special functions — integral transforms — asymptotics — continued fractions*, reprint of the 1977 original, Wiley Classics Lib., Wiley-Interscience, Wiley, New York; v. 3: *Discrete Fourier analysis — Cauchy integrals — construction of conformal maps — univalent functions*, Pure Appl. Math., Wiley-Interscience, Wiley, New York.

Hermite, Charles and Stieltjes, Thomas Jan [1905]. *Correspondance d'Hermite et de Stieltjes*, published by B. Baillaud and H. Bourget, with a preface by Emile Picard, vols. 1,2, Gauthier-Villars, Paris. [Partie inédite de la correspondance d'Hermite avec Stieltjes, *Proc. Sem. History Math. 4*, Inst. Henri Poincaré, Paris, 1983, 75–87.]

Hernández, Eugenio and Weiss, Guido [1996]. *A first course on wavelets*, Stud. Adv. Math., CRC Press, Boca Raton, FL.

Hestenes, Magnus Rudolf [1980]. *Conjugate direction methods in optimization*, Appl. Math., v. 12, Springer, New York.

Hesthaven, Jan S., Gottlieb, Sigal, and Gottlieb, David [2007]. *Spectral methods for time-dependent methods*, Cambridge Monographs Appl. Comput. Math., v. 21, Cambridge University Press, Cambridge.

Hewitt, Edwin and Stromberg, Karl [1975]. *Real and abstract analysis: A modern treatment of the theory of functions of a real variable*, third printing, Graduate Texts Math., no. 25, Springer, New York.

Higham, Nicholas J. [2002]. *Accuracy and stability of numerical algorithms*, 2d ed., SIAM, Philadelphia, PA.

Higham, Nicholas J. [2008]. *Functions of matrices: Theory and computation*, SIAM, Philadelphia, PA.

Higham, Desmond J. and Higham, Nicholas J. [2005]. *Matlab guide*, 2d ed., SIAM, Philadelphia, PA.

Hindmarsh, Allan C. [1980]. LSODE and LSODI, two new initial value ordinary differential equation solvers, *ACM SIGNUM Newsletter 15*, no. 4, 10–11.

Holladay, John C. [1957]. A smoothest curve approximation, *Math. Tables Aids Comput. 11*, 233–243.

Höllig, Klaus [2003]. *Finite element methods with B-splines*, Frontiers Appl. Math., v. 26, SIAM, Philadelphia, PA.

Hörmann, Wolfgang, Leydold, Josef, and Derflinger, Gerhard [2004]. *Automatic nonuniform random variate generation*, Stat. Comput., Springer, Berlin.

Hoschek, Josef and Lasser, Dieter [1993]. *Fundamentals of computer aided geometric design*, translated from the 1992 German edition by Larry L. Schumaker, A K Peters, Wellesley, MA.

Householder, A. S. [1970]. *The numerical treatment of a single nonlinear equation*, Internat. Ser. Pure Appl. Math., McGraw-Hill, New York.

Householder, Alston S. [2006]. *Principles of numerical analysis*, reprinting of the 1974 corrected edition, Dover Publ., Mineola, NY.

Hu, T. C. and Shing, M. T. [2002]. *Combinatorial algorithms*, enlarged second edition, Dover Publ., Mineola, NY.

Hull, T. E. and Luxemburg, W. A. J. [1960]. Numerical methods and existence theorems for ordinary differential equations, *Numer. Math. 2*, 30–41.

Hunt, Brian R., Lipsman, Ronald L., and Rosenberg, Jonathan M. [2006]. *A guide to MATLAB®*, 2d ed., Cambridge University Press, Cambridge.

IEEE [1985]. *IEEE standard for binary floating-point arithmetic, ANSI/IEEE standard 754-1985*, Institute of Electrical and Electronics Engineers, New York. [Reprinted in *SIGPLAN Notices 22*, no. 2 (1987), 9–25.]

Il'in, V. P. [1992]. *Iterative incomplete factorization methods*, Ser. Soviet East Europ. Math., v. 4, World Scientific, River Edge, NJ.

Imhof, J. P. [1963]. On the method for numerical integration of Clenshaw and Curtis, *Numer. Math. 5*, 138–141.

Isaacson, Eugene and Keller, Herbert Bishop [1994]. *Analysis of numerical methods*, corrected reprint of the 1966 original, Dover Publ., New York.

Iserles, Arieh, ed. [1992–2010]. *Acta numerica*, Cambridge University Press, Cambridge.

Iserles, Arieh [2009]. *A first course in the numerical analysis of differential equations*, 2d ed., Cambridge Texts Appl. Math., Cambridge University Press, Cambridge.

Iserles, A. and Nørsett, S. P. [1991]. *Order stars*, Appl. Math. Math. Comput., v. 2, Chapman & Hall, London.

Jacobi, C. G. J. [1826]. Ueber Gaußs neue Methode, die Werthe der Integrale näherungsweise zu finden, *J. Reine Angew. Math. 1*, 301–308.

Jaulin, Luc, Kleffer, Michel, Didrit, Olivier, and Walter, Éric [2001]. *Applied interval analysis: With examples in parameter and state estimation, robust control and robotics*, Springer, London.

Jenkins, M. A. and Traub, J. F. [1970]. A three-stage algorithm for real polynomials using quadratic iteration, *SIAM J. Numer. Anal. 7*, 545–566.

Jennings, Alan and McKeown, J. J. [1992]. *Matrix computation*, 2d ed., Wiley, Chichester.

Jiang, Bo-nan [1998]. *The least-squares finite element method: Theory and applications in computational fluid dynamics and electromagnetics*, Scient. Comput., Springer, Berlin.

Johnson, Claes [2009]. *Numerical solution of partial differential equations by the finite element method*, reprint of the 1987 edition, Dover Publ., Mineola, NY.

Joyce, D. C. [1971]. Survey of extrapolation processes in numerical analysis, *SIAM Rev. 13*, 435–490.

Kalos, Malvin H. and Whitlock, Paula A. [2008]. *Monte Carlo methods*, 2d ed., Wiley-Blackwell, Weinheim.

Kaltenbacher, Barbara, Neubauer, Andreas, and Scherzer, Otmar [2008]. *Iterative regularization methods for nonlinear ill-posed problems*, Radon Ser. Comput. Appl. Math., v. 6, de Gruyter, Berlin.

Kantorovich, L. V. and Akilov, G. P. [1982]. *Functional analysis*, translated from the Russian by Howard L. Silcock, 2d ed., Pergamon Press, Oxford.

Kaucher, Edgar W. and Miranker, Willard L. [1984]. *Self-validating numerics for function space problems: Computation with guarantees for differential and integral equations*, Notes Rep. Comput. Sci. Appl. Math., v. 9, Academic Press, Orlando, FL.

Kautsky, J. and Elhay, S. [1982]. Calculation of the weights of interpolatory quadratures, *Numer. Math. 40*, 407–422.

Kedem, Gershon [1980]. Automatic differentiation of computer programs, *ACM Trans. Math. Software 6*, 150–165.

Keller, Herbert B. [1992]. *Numerical methods for two-point boundary-value problems*, corrected reprint of the 1968 edition, Dover Publ., New York.

Kelley, C. T. [1995]. *Iterative methods for linear and nonlinear equations*, Frontiers Appl. Math., v. 16, SIAM, Philadelphia, PA.

Kelley, C. T. [1999]. *Iterative methods for optimization*, Frontiers Appl. Math., v. 18, SIAM, Philadelphia, PA.

Kelley, C. T. [2003]. *Solving nonlinear equations with Newton's method*, Fundamentals of Algorithms, SIAM, Philadelphia, PA.

Kelley, Walter G. and Peterson, Allan C. [2001]. *Difference equations: An introduction with applications*, 2d ed., Harcourt/Academic Press, San Diego, CA.

Kennedy, William J., Jr. and Gentle, James E. [1980]. *Statistical computing*, Statistics: Textbooks and Monographs, v. 33, Dekker, New York.

Kerner, Immo O. [1966]. Ein Gesamtschrittverfahren zur Berechnung der Nullstellen von Polynomen, *Numer. Math. 8*, 290–294.

Kershaw, D. [1971]. A note on the convergence of interpolatory cubic splines, *SIAM J. Numer. Anal. 8*, 67–74.

Kincaid, David and Cheney, Ward [1996]. *Numerical analysis: Mathematics of scientific computing*, 2d ed., Brooks/Cole, Pacific Grove, CA.

King, J. Thomas and Murio, Diego A. [1986]. Numerical differentiation by finite-dimensional regularization, *IMA J. Numer. Anal. 6*, 65–85.

Klopfenstein, R. W. [1965]. Applications of differential equations in general problem solving, *Comm. ACM 8*, 575–578.

Knight, Andrew [2000]. *Basics of MATLAB® and beyond*, Chapman & Hall/CRC, Boca Raton, FL.

Knuth, Donald E. [1975, 1981, 1973, 2005–2006]. *The art of computer programming*; v. 1: *Fundamental algorithms*, second printing of the second edition; v. 2: *Seminumerical algorithms*, 2d ed.; v. 3: *Sorting and searching*; v. 4, Fasc. 1–4: *NMIX, a RISC computer for the new millennium*, *generating all tubles and permutations*, *generating all combinations and partitions*, *generating all trees — history of combinatorial generation*; Addison-Wesley Ser. Comput. Sci. Inform. Process., Addison-Wesley, Reading, MA and Addison-Wesley, Upper Saddle River, NJ.

Ko, Ker-I [1991]. *Complexity theory of real functions*, Progr. Theoret. Comput. Sci., Birkhäuser, Boston, MA.

Köckler, Norbert [1994]. *Numerical methods and scientific computing: Using software libraries for problem solving*, Oxford Sci. Publ., Oxford University Press, New York.

Kollerstrom Nick [1992]. Thomas Simpson and "Newton's method of approximation": An enduring myth, *British J. Hist. Sci. 25*, 347–354.

Kopriva, David A. [2009]. *Implementing spectral methods for partial differential equations: Algorithms for scientists and engineers*, Scient. Comput., Springer, Berlin.

Koumandos, Stamatis [2011]. Positive trigonometric sums and starlike functions, in *Approximation and Computation: In honor of Gradimir V. Milovanović*, W. Gautschi, G. Mastroianni, and Th. M. Rassias, eds., 157–182, Springer Ser. Optim. Appl., v. 42, Springer, New York.

Kowalski, Marek A., Sikorski, Krzysztof A., and Stenger, Frank [1995]. *Selected topics in approximation and computation*, Oxford University Press, New York.

Kravanja, Peter and Van Barel, Marc [2000]. *Computing the zeros of analytic functions*, Lecture Notes in Math., v. 1727, Springer, Berlin.

Kressner, Daniel [2005]. *Numerical methods for general and structured eigenvalue problems*, Lecture Notes in Comput. Sci. Engrg., v. 46, Springer, Berlin.

Křížek, M. and Neittaanmäki, P. [1990]. *Finite element approximation of variational problems and applications*, Pitman Monographs Surveys Pure Appl. Math., v. 50, Wiley, New York.

Krommer, Arnold R. and Ueberhuber, Christoph W. [1998]. *Computational integration*, SIAM, Philadelphia, PA.

Kröner, Dietmar [1997]. *Numerical schemes for conservation laws*, Wiley-Teubner Ser. Advances Numer. Math., Wiley, Chichester./Thomas

Kronsjö, Lydia [1987]. *Algorithms: Their complexity and efficiency*, 2d ed., Wiley Ser. Comput., Wiley, Chichester.

Krylov, Vladimir Ivanovich [1962]. *Approximate calculation of integrals*, translated from the Russian by Arthur H. Stroud, ACM Monograph Ser., Macmillan, New York.

Kuang, Yang [1993]. *Delay differential equations with applications in population dynamics*, Math. Sci. Engrg., v. 191, Academic Press, Boston, MA.

Kulisch, Ulrich [2008]. *Computer arithmetic and validity: Theory, implementation, and applications*, de Gruyter Stud. Math., v. 33, de Gruyter, Berlin.

Kulisch, Ulrich W. and Miranker, Willard L. [1981]. *Computer arithmetic in theory and practice*, Comput. Sci. Appl. Math., Academic Press, New York.

Kwon, Young W. and Bang, Hyochoong [2000]. *The finite element method using Matlab*, 2d ed., CRC Mech. Engrg. Ser., Chapman & Hall/CRC, Boca Raton, FL.

Kythe, Prem K. [1998]. *Computational conformal mapping*, Birkhäuser, Boston, MA.

Kythe, Prem K. and Puri, Pratap [2002]. *Computational methods for linear integral equations*, Birkhäuser, Boston, MA.

Kythe, Prem K. and Wei, Dongming [2004]. *An introduction to linear and nonlinear finite element analysis: A computational approach*, Birkhäuser, Boston, MA.

Kyurkchiev, Nikolay V. [1998]. *Initial approximations and root finding methods*, Math. Res., v. 104, Wiley-VCH, Berlin.

Lakshmikantham, V. and Trigiante, D. [2002]. *Theory of difference equations: Numerical methods and applications*, 2d ed., Monographs and Textbooks Pure Appl. Math., v. 251, Dekker, New York.

Lambert, J. D. [1991]. *Numerical methods for ordinary differential systems: The initial value problem*, Wiley, Chichester.

Lange, Kenneth [2010]. *Numerical analysis for statisticians*, 2d ed., Stat. Comput., Springer, New York.

Larsson, Stig and Thomée, Vidar [2003]. *Partial differential equations with numerical methods*, Texts in Appl. Math., v. 45, Springer, Berlin.

Laurie, D. P. [1997]. Calculation of Gauss–Kronrod quadrature rules, *Math. Comp. 66*, 1133–1145.

Lax, Peter D. [1973]. *Hyperbolic systems of conservation laws and the mathematical theory of shock waves*, CBMS Regional Conf. Ser. Appl. Math., no. 11, SIAM, Philadelphia, PA.

Legendre, A. M. [1805]. *Nouvelles méthodes pour la détermination des orbites des comètes*, Courcier, Paris.

Lehmer, D. H. [1961]. A machine method for solving polynomial equations, *J. Assoc. Comput. Mach. 8*, 151–162.

Lemieux, Christiane [2009]. *Monte Carlo and quasi-Monte Carlo sampling*, Springer Ser. Stat., Springer, New York.

LeVeque, Randall J. [1992]. *Numerical methods for conservation laws*, 2d ed., Lectures in Math. ETH Zürich, Birkhäuser, Basel.

LeVeque, Randall J. [2002]. *Finite volume methods for hyperbolic problems*, Cambridge Texts Appl. Math., Cambridge University Press, Cambridge.

LeVeque, Randall J. [2007]. *Finite difference methods for ordinary and partial differential equations: Steady-state and time-dependent problems*, SIAM, Philadelphia, PA.

Levin, Meishe and Girshovich, Jury [1979]. *Optimal quadrature formulas*, Teubner-Texte Math., Teubner, Leipzig.

Liu, Jun S. [2008]. *Monte Carlo strategies in scientific computing*, Springer Ser. Stat., Springer, New York.

Lorentz, George G., Golitschek, Manfred v., and Makovoz, Yuly [1996]. *Constructive approximation: Advanced problems*, Grundlehren Math. Wiss., v. 304, Springer, Berlin.

Lorentz, George G., Jetter, Kurt, and Riemenschneider, Sherman D. [1983]. *Birkhoff interpolation*, Encyclopedia Math. Appl., v. 19, Addison-Wesley, Reading, MA.

Lorentz, Rudolph A. [1992]. *Multivariate Birkhoff interpolation*, Lecture Notes in Math., v. 1516, Springer, Berlin.

Lozier, D. W. and Olver, F. W. J. [1994]. Numerical evaluation of special functions, in Gautschi, ed. [1994a, 79–125].

Luke, Yudell L. [1975]. *Mathematical functions and their approximations*, Academic Press, New York.

Luke, Yudell L. [1977]. *Algorithms for the computation of mathematical functions*, Academic Press, New York.

Lyness, J. N. [1968]. Differentiation formulas for analytic functions, *Math. Comp. 22*, 352–362.

Lyness, J. N. and Moler, C. B. [1967]. Numerical differentiation of analytic functions, *SIAM J. Numer. Anal. 4*, 202–210.

Lyness, J. N. and Sande, G. [1971]. Algorithm 413 — ENTCAF and ENTCRE: Evaluation of normalized Taylor coefficients of an analytic function, *Comm. ACM 14*, 669–675.

Maas, Christoph [1985]. Was ist das Falsche an der Regula Falsi?, *Mitt. Math. Ges. Hamburg 11*, 311–317.

MacLeod, Allan J. [1996]. Algorithm 757: MISCFUN, a software package to compute uncommon special functions, *ACM Trans. Math. Software 22*, 288–301.

Maindonald, J. H. [1984]. *Statistical computation*, Wiley Ser. Probab. Math. Statist.: Appl. Probab. Statist., Wiley, New York.

Manno, István [1999]. *Introduction to the Monte Carlo method*, Akadémiai Kiadó, Budapest.

Mäntylä, Martti [1988]. *An introduction to solid modeling*, Principles Comput. Sci. Ser., v. 13, Computer Science Press, Rockville, MD.

Marden, Morris [1966]. *Geometry of polynomials*, 2d ed., Math. Surveys, no. 3, Amer. Math. Soc., Providence, RI.

Mathew, Tarek P. A. [2008]. *Domain decomposition methods for the numerical solution of partial differential equations*, Lecture Notes in Comput. Sci. Engrg., v. 61, Springer, Berlin.

McCormick, Stephen F. [1989]. *Multilevel adaptive methods for partial differential equations*, Frontiers in Appl. Math., v. 6, SIAM, Philadelphia, PA.

McCormick, Stephen F. [1992]. *Multilevel projection methods for partial differential equations*, CBMS-NSF Regional Conf. Ser. Appl. Math., no. 62, SIAM, Philadelphia, PA.

McLeod, Robin J. Y. and Baart, M. Louisa [1998]. *Geometry and interpolation of curves and surfaces*, Cambrige University Press, Cambridge.

McNamee, J. M. [2007]. *Numerical methods for roots of polynomials. Part I*, Stud. Comput. Math., v. 14, Elsevier, Amsterdam.

Meurant, G. [1999]. *Computer solution of large linear systems*, Stud. Math. Appl., v. 28, North-Holland Publ., Amsterdam.

Meurant, Gérard [2006]. *The Lanczos and conjugate gradient algorithms: From theory to finite precision computations*, Software, Environments, and Tools, v. 19, SIAM, Philadelphia, PA.

Meyer, Arnd [1987]. *Modern algorithms for large sparse eigenvalue problems*, Math. Res., v. 34, Akademie-Verlag, Berlin.

Meyer, Gunter H. [1973]. *Initial value methods for boundary value problems: Theory and application of invariant imbedding*, Math. Sci. Engrg., v. 100, Academic Press, New York.

Micchelli, Charles A. [1980]. Some positive Cotes numbers for the Chebyshev weight function, *Aequationes Math. 21*, 105–109.

Miel, George and Mooney, Rose [1985]. On the condition number of Lagrangian numerical differentiation, *Appl. Math. Comput. 16*, 241–252.

Mignotte, Maurice [1992]. *Mathematics for computer algebra*, translated from the French by Catherine Mignotte, Springer, New York.

Miller, Kenneth S. [1968]. *Linear difference equations*, Benjamin, New York.

Milne, W. E. [1926]. Numerical integration of ordinary differential equations, *Amer. Math. Monthly 33*, 455–460.

Milovanović, G. V., Mitrinović, D. S., and Rassias, Th. M. [1994]. *Topics in polynomials: Extremal problems, inequalities, zeros*, World Scientific, River Edge, NJ.

Mishra, Bhubaneswar [1993]. *Algorithmic algebra*, Texts and Monographs Comput. Sci., Springer, New York.

Moler, Cleve B. [2004]. *Numerical computing with MATLAB*, SIAM, Philadelphia, PA.

Monahan, John F. [2001]. *Numerical methods of Statistics*, Cambridge Ser. Stat. Probab. Math., v. 7, Cambridge University Press, Cambridge.

Monk, Peter [2003]. *Finite element methods for Maxwell's equation*, Numer. Math. Scient. Comput., Oxford University Press, New York.

Moore, Ramon E. [1966]. *Interval analysis*, Prentice-Hall Ser. Automat. Comput., Prentice-Hall, Englewood Cliffs, NJ.

Moore, Ramon E. [1979]. *Methods and applications of interval analysis*, SIAM Stud. Appl. Math., v. 2, SIAM, Philadelphia, PA.

Moore, Ramon E., Kearfott, R. Baker, and Cloud, Michael L. [2009]. *Introduction to interval analysis*, SIAM, Philadelphia, PA.

Moré, Jorge J. and Wright, Stephen J. [1993]. *Optimization software guide*, Frontiers Appl. Math., v. 14, SIAM, Philadelphia, PA.

Morgan, Alexander [1987]. *Solving polynomial systems using continuation for engineering and scientific problems*, Prentice-Hall, Englewood Cliffs, NJ.

Morrison, David D., Riley, James D., and Zancanaro, John F. [1962]. Multiple shooting method for two-point boundary value problems, *Comm. ACM 5*, 613–614.

Morton, K. W. and Mayers, D. F. [2005]. *Numerical solution of partial differential equations: An introduction*, 2d ed., Cambridge University Press, Cambridge.

Moulton, Forest Ray [1926]. *New methods in exterior ballistics*, University of Chicago Press, Chicago, IL.

Muller, David E. [1956]. A method for solving algebraic equations using an automatic computer, *Math. Tables Aids Comp. 10*, 208–215.

Muller, Jean-Michel, Brisebarre, Nicolas, de Dinechin, Florent, Jeannerod, Claude-Pierre, Lefèvre, Vincent, Melquiond, Guillaume, Revol, Nathalie, Stehlé, Damien, and Torres, Serge [2010]. *Handbook of floating-point arithmetic*, Birkhäuser, Boston, MA.

Mysovskikh, I. P. [1981]. *Interpolatory cubature formulas* (Russian), Nauka, Moscow.

Na, Tsung Yen [1979]. *Computational methods in engineering boundary value problems*, Math. Sci. Engrg., v. 145, Academic Press, New York.

Nash, Stephen G., ed. [1990]. *A history of scientific computing*, ACM Press History Ser., Addison-Wesley, Reading, MA.

Natanson, I. P. [1964, 1965, 1965]. *Constructive function theory*. v. 1: *Uniform approximation*, translated from the Russian by Alexis N. Obolensky; v. 2: *Approximation in mean*, translated from the Russian by John R. Schulenberger; v. 3: *Interpolation and approximation quadratures*, translated from the Russian by John R. Schulenberger, Ungar, New York.

Nazareth, J. L. [1987]. *Computer solution of linear programs*, Monographs Numer. Anal., Oxford Sci. Publ., Oxford University Press, New York.

Neidinger, Richard D. [2010]. Introduction to automatic differentiation and MATLAB object-oriented programming, *SIAM Rev. 52*, 545–563.

Németh, Géza [1992]. *Mathematical approximation of special functions: Ten papers on Chebyshev expansions*, Nova Science, Commack, NY.

Neumaier, Arnold [1990]. *Interval methods for systems of equations*, Encyclopedia Math. Appl., v. 37, Cambridge University Press, Cambridge.

Neumaier, Arnold [2001]. *Introduction to numerical analysis*, Cambridge University Press, Cambridge.

Niederreiter, Harald [1992]. *Random number generation and quasi-Monte Carlo methods*, CBMS-NSF Regional Conf. Ser. Appl. Math., no. 63, SIAM, Philadelphia, PA.

Nijenhuis, Albert and Wilf, Herbert S. [1978]. *Combinatorial algorithms: For computers and calculators*, 2d ed., Comput. Sci. Appl. Math., Academic Press, New York.

Nikolov, Geno and Simian, Corina [2011]. Gauss-type quadrature formulae for parabolic splines with equidistant knots, in: *Approximation and Computation: In honor of Gradimir V. Milovanović*, W. Gautschi, G. Mastroianni, and Th. M. Rassias, eds., 209–231, Springer Ser. Optim. Appl., v. 42, Springer, New York.

Nikol'skiĭ, S. M. [1988]. *Quadrature formulas* (Russian), 4th ed., with an appendix by N. P. Korneĭchuk and a preface by Nikol'skiĭ and Korneĭchuk, Nauka, Moscow.

Niven, Ivan, Zuckerman, Herbert S., and Montgomery, Hugh L. [1991]. *An introduction to the theory of numbers*, 5th ed., Wiley, New York.

Nørsett, S. P. [1974]. *Semi explicit Runge–Kutta methods*, Rep. No. 6/74, ISBN 82-7151-009-6, Dept. Math., Univiversity of Trondheim, Norway.

Nørsett, Syvert P. [1976]. Runge–Kutta methods with a multiple real eigenvalue only, *BIT 16*, 388–393.

Notaris, Sotirios E. [1994]. An overview of results on the existence or nonexistence and the error term of Gauss–Kronrod quadrature formulae, in *Approximation and computation: A festschrift in honor of Walter Gautschi*, R. V. M. Zahar, ed., 485–496, Internat. Ser. Numer. Math., v. 119, Birkhäuser, Boston, MA.

Notaris, Sotirios E. [1997]. Interpolatory quadrature formulae with Chebyshev abscissae of the third and fourth kind, *J. Comput. Appl. Math. 81*, 83–99.

Notaris, Sotirios E. [1998]. New interpolatory quadrature formulae with Chebyshev abscissae, *J. Comput. Appl. Math. 94*, 199–214.

Notaris, Sotirios E. [2002]. Positivity of the weights of interpolatory quadrature formulae with Jacobi abscissae, *BIT 42*, 440–446.

Notaris, Sotirios E. [2003]. New interpolatory quadrature formulae with Gegenbauer abscissae, *J. Comput. Appl. Math. 161*, 295–312.

Novak, Erich, Sloan, Jan H., Traub, Josef F., and Woźniakowski, Henryk [2009]. *Essays on the complexity of continuous problems*, Europ. Math. Soc., Zürich.

Novikov, I. Ya., Protasov, V. Yu., and Skopina, M. A. [2010]. *Wavelet theory*, translated by Evgenia Sorokina, Transl. Math. Monographs, v. 239, Amer. Math. Society, Providence, R. I.

Nürnberger, Günther [1989]. *Approximation by spline functions*, Springer, Berlin.

O'Leary, Dianne P. [2009]. *Scientific computing with case studies*, with the collaboration of James G. Nagy, Nargess Memarsadeghi, David A. Schug, Isabel Beichl, Francis Sullivan, and Yalin E. Sagduyu, SIAM, Philadelphia, PA.

Oliver, J. [1980]. Algorithm 017 — An algorithm for numerical differentiation of a function of one real variable, *J. Comp. Appl. Math. 6*, 145–160.

Olver, Frank W. J., Lozier, Daniel W., Boisvert, Ronald F., and Clark, Charles [2010]. *NIST handbook of mathematical functions*, National Institute of Standards and Technology, U.S. Department of Commerce, Gaithersburg, MD and Cambridge University Press, New York.

Ono, Hiroshi [2006]. On the 25 stage 12th order explicit Runge–Kutta method (Japanese), *Trans. Japan Soc. Indust. Appl. Math. 16*, 177–186.

Ortega, James M. [1989]. *Introduction to parallel and vector solution of linear systems*, Frontiers Comput. Sci., Plenum, New York.

Ortega, J. M. and Rheinboldt, W. C. [2000]. *Iterative solution of nonlinear equations in several variables*, reprint of the 1970 original, Classics Appl. Math., v. 30, SIAM, Philadelphia, PA.

Ortega, James M. and Voigt, Robert G. [1985]. *Solution of partial differential equations on vector and parallel computers*, SIAM, Philadelphia, PA.

Østerby, Ole and Zlatev, Zahari [1983]. *Direct methods for sparse matrices*, Lecture Notes in Comput. Sci., v. 157, Springer, Berlin.

Ostrowski, A. [1954]. On two problems in abstract algebra connected with Horner's rule, in *Studies in mathematics and mechanics presented to Richard von Mises*, Garrett Birkhoff, Gustav Kuerti, and Gabor Szegö, eds., 40–48, Academic Press, New York. [Collected Math. Papers, v. 2, Birkhäuser, Basel, 1983, 510–518.]

Ostrowski, A. M. [1973]. *Solution of equations in Euclidean and Banach spaces*, 3d ed. of *Solution of equations and systems of equations*, Pure Appl. Math., v. 9, Academic Press, New York.

Otto, S. R. and Denier, J. P. [2005]. *An introduction to programming and numerical methods in MATLAB*, Springer, London.

Overton, Michael L. [2001]. *Numerical computing with IEEE floating point arithmetic: Including one theorem, one rule of thumb, and one hundred and one exercises*, SIAM, Philadelphia, PA.

Panik, Michael J. [1996]. *Linear programming: Mathematics, theory, and algorithms*, Appl. Optim., v. 2, Kluwer, Dordrecht.

Papamichael, Nicolas and Stylianopoulos, Nikos [2010]. *Numerical conformal mapping: Domain decomposition and the mapping of quadrilaterals*, World Scientific, Hackensack, NJ.

Pardalos, P. M., Phillips, A. T., and Rosen, J. B. [1992]. *Topics in parallel computing in mathematical programming*, Appl. Discr. Math. Theoret. Comput. Sci., v. 2, Science Press, New York.

Parlett, Beresford N. [1998]. *The symmetric eigenvalue problem*, corrected reprint of the 1980 original, Classics Appl. Math., v. 20, SIAM, Philadelphia, PA.

Paszkowski, Stefan [1983]. *Computational applications of Chebyshev polynomials and series* (Russian), translation from the Polish by S. N. Kiro, Nauka, Moscow.

Perron, Oskar [1957]. *Die Lehre von den Kettenbrüchen*, third improved and expanded edition, v. 2: *Analytisch-funktionentheoretische Kettenbrüche*, Teubner, Stuttgart.

Peters, G. and Wilkinson, J. H. [1971]. Practical problems arising in the solution of polynomial equations, *J. Inst. Math. Appl. 8*, 16–35.

Petković, Miodrag [1989]. *Iterative methods for simultaneous inclusion of polynomial zeros*, Lecture Notes in Math., v. 1387, Springer, Berlin.

Petković, Miodrag [2008]. *Point estimation of root finding methods*, Lecture Notes in Math., v. 1933, Springer, Berlin.

Petković, Miodrag and Petković, Ljiliana D. [1998]. *Complex interval arithmetic and its applications*, Math. Res., v. 105, Wiley-VCH Verlag, Berlin.

Petrushev, P. P. and Popov, V. A. [1987]. *Rational approximation of real functions*, Encyclopedia Math. Appl., v. 28, Cambridge University Press, Cambridge.

Peyret, Roger [2002]. *Spectral methods for incompressible viscous flow*, Appl. Math. Sci., v. 148, Springer, New York.

Piessens, Robert, de Doncker-Kapenga, Elise, Überhuber, Christoph W., and Kahaner, David K. [1983]. *QUADPACK: A subroutine package for automatic integration*, Springer Ser. Comput. Math., v. 1, Springer, Berlin.

Pinkus, Allan M. [1989]. *On L^1-approximation*, Cambridge Tracts in Math., v. 93, Cambridge University Press, Cambridge.

Pinney, Edmund [1958]. *Ordinary difference-differential equations*, University of California Press, Berkeley, CA.

Plofker, Kim [1996]. An example of the secant method of iterative approximation in a fifteenth-century Sanskrit text, *Historia Math. 23*, 246–256.

Pohst, Michael E. [1993]. *Computational algebraic number theory*, DMV Seminar, v. 21, Birkhäuser, Basel.

Pomerance, Carl, ed. [1990]. *Cryptology and computational number theory*, Proc. Sympos. Appl. Math., v. 42, Amer. Math. Soc., Providence, RI.

Potra, F. A. [1989]. On Q-order and R-order of convergence, *J. Optim. Theory Appl. 63*, 415–431.

Potra, F.-A. and Pták, V. [1984]. *Nondiscrete induction and iterative processes*, Res. Notes Math., v. 103, Pitman, Boston.

Powell, M. J. D. [1981]. *Approximation theory and methods*, Cambridge University Press, Cambridge.

Prautsch, Hartmut, Boehm, Wolfgang, and Paluzny, Marco [2002]. *Bézier and B-spline techniques*, Math. Visual., Springer, Berlin.

Preparata, Franco P. and Shamos, Michael Ian [1985]. *Computational geometry: An introduction*, Texts and Monographs Comput. Sci., Springer, New York.

Prince, P. J. and Dormand, J. R. [1981]. High order embedded Runge–Kutta formulae, *J. Comput. Appl. Math. 7*, 67–75.

Prössdorf, Siegfried and Silbermann, Bernd [1991]. *Numerical analysis for integral and related operator equations*, Operator Theory: Adv. Appl., v. 52, Birkhäuser, Basel.

Pryce, John D. [1993]. *Numerical solution of Sturm–Liouville problems*, Monographs Numer. Anal., Oxford Sci. Publ., Oxford University Press, New York.

Quade, W. [1951]. Numerische Integration von Differentialgleichungen bei Approximation durch trigonometrische Ausdrücke, *Z. Angew. Math. Mech. 31*, 237–238.

Quarteroni, Alfio [2009]. *Numerical models for differential problems*, translated from the 4th (2008) Italian edition by Silvia Quarteroni, MS&A. Modeling, Simul. Appl., v. 2, Springer-Verlag Italia, Milan.

Quarteroni, Alfio and Valli, Alberto [1994]. *Numerical approximation of partial differential equations*, Springer Ser. Comput. Math., v. 23, Springer, Berlin.

Quarteroni, Alfio and Valli, Alberto [1999]. *Domain decomposition methods for partial differential equations*, Numer. Math. Scient. Comput., Oxford Sci. Publ., Oxford University Press, New York.

Quarteroni, Alfio, Saleri, Fausto, and Gervasio, Paola [2010]. *Scientific computing with MATLAB and Octave*, 3d ed., Texts Comput. Sci. Engrg., v. 2, Springer, Berlin.

Rabinowitz, Philip [1992]. Extrapolation methods in numerical integration, *Numer. Algorithms 3*, 17–28.

Radon, Brigitte (1950). Sviluppi in serie degli integrali ellittici, *Atti Accad. Naz. Lincei. Mem. Cl. Sci. Fis. Mat. Nat. Sez. I (8) 2* (1950), 69–109.

Rakitskiĭ, Ju. V. [1961]. Some properties of solutions of systems of ordinary differential equations by one-step methods of numerical integration (Russian), *Ž. Vyčisl. Mat. i Mat. Fiz. 1*, 947–962. [Engl. transl. in *U.S.S.R. Comput. Math. Math. Phys. 1* (1962), 1113–1128.]

Ratschek, H. and Rokne, J. [1984]. *Computer methods for the range of functions*, Ellis Horwood Ser. Math. Appl., Horwood, Chichester.

Ratschek, H. and Rokne, J. [1988]. *New computer methods for global optimization*, Ellis Horwood Ser. Math. Appl., Horwood, Chichester.

Reddien, G. W. [1980]. Projection methods for two-point boundary value problems, *SIAM Rev. 22*, 156–171.

Reddy, J. N. [2004]. *An introduction to nonlinear finite element analysis*, Oxford University Press, Oxford.

Resnikoff, Howard L. and Wells, Raymond O., Jr. [1998]. *Wavelet analysis. The scalable structure of information*, Springer, New York.

Rheinboldt, Werner C. [1998]. *Methods for solving systems of nonlinear equations*, 2d ed., CBMS-NSF Regional Conf. Ser. Appl. Math., v. 70, SIAM, Philadelphia, PA.

Rice, John R. [1966]. A theory of condition, *SIAM J. Numer. Anal. 3*, 287–310.

Riesel, Hans [1994]. *Prime numbers and computer methods for factorization*, 2d ed., Progr. Math., v. 126, Birkhäuser, Boston, MA.

Ritter, Klaus [2000]. *Average-case analysis of numerical problems*, Lecture Notes in Math., v. 1733, Springer, Berlin.

Rivlin, T. J. [1975]. Optimally stable Lagrangian numerical differentiation, *SIAM J. Numer. Anal. 12*, 712–725.

Rivlin, Theodore J. [1981]. *An introduction to the approximation of functions*, corrected reprint of the 1969 original, Dover Books Adv. Math., Dover, New York.

Rivlin, Theodore J. [1990]. *Chebyshev polynomials: From approximation theory to algebra and number theory*, 2d ed., Pure Appl. Math., Wiley, New York.

Roberts, Sanford M. and Shipman, Jerome S. [1972]. *Two-point boundary value problems: Shooting methods*, Modern Analyt. Comput. Methods Sci. Math., no. 31, American Elsevier, New York.

Rosen, Kenneth H. [2000]. *Elementary number theory and its applications*, 4th ed., Addison-Wesley, Reading, MA. Reading, MA.

Rudin, Walter [1976]. *Principles of mathematical analysis*, 3d ed., Internat. Ser. Pure Appl. Math., McGraw-Hill, New York.

Rump, Siegfried M. [2010]. Verification methods: Rigorous results using floating-point arithmetic, in *Acta Numerica*, v. 19, 287–449.

Rutishauser, Heinz [1952]. Über die Instabilität von Methoden zur Integration gewöhnlicher Differentialgleichungen, *Z. Angew. Math. Physik 3*, 65–74.

Rutishauser, Heinz [1957]. *Der Quotienten-Differenzen-Algorithmus*, Mitt. Inst. Angew. Math. Zürich, no. 7.

Rutishauser, Heinz [1990]. *Lectures on numerical mathematics*, edited by M. Gutknecht with the assistance of Peter Henrici, Peter Läuchli, and Hans-Rudolf Schwarz; with a foreword by Gutknecht and a preface by Henrici, Läuchli, and Schwarz; translated from the German and with a preface by Walter Gautschi, Birkhäuser, Boston, MA.

Saad, Youcef [1992]. *Numerical methods for large eigenvalue problems*, Algorithms Architect. Adv. Sci. Comput., Halsted Press, New York.

Saad, Yousef [2003]. *Iterative methods for sparse linear systems*, 2d ed., SIAM, Philadelphia, PA.

Salihov, N. P. [1962]. Polar difference methods for solving the Cauchy problem for a system of ordinary differential equations (Russian), *Ž. Vyčisl. Mat. i Mat. Fiz. 2*, 515–528. [Engl. transl. in *U.S.S.R. Comput. Math. Math. Phys. 2* (1963), 535–553.]

Salomon, David [2006]. *Curves and surfaces for computer graphics*, Springer, New York.

Salzer, Herbert E. [1949]. Coefficients for facilitating trigonometric interpolation, *J. Math. Physics 27*, 274–278.

Samarskii, Alexander A. [2001]. *The theory of difference schemes*, Monographs and Textbooks Pure Appl. Math., v. 240, Dekker, New York.

Sanz-Serna, J. M. and Calvo, M. P. [1994]. *Numerical Hamiltonian problems*, Appl. Math. Math. Comput., v. 7, Chapman & Hall, London.

Schendel, U. [1984]. *Introduction to numerical methods for parallel computers*, translated from the German by B. W. Conolly, Ellis Horwood Ser. Math. Appl., Halsted Press, New York.

Schiesser, W. E. [1991]. *The numerical method of lines: Integration of partial differential equations*, Academic Press, San Diego, CA.

Schinzinger, Roland and Laura, Patricio A. A. [2003]. *Conformal mapping: Methods and applications*, revised edition of the 1991 original, Dover Publ., Mineola, NY.

Schoenberg, I. J. [1946]. Contributions to the problem of approximation of equidistant data by analytic functions. Part A: On the problem of smoothing or graduation. A first class of analytic approximation formulae, *Quart. Appl. Math. 4*, 45–99; Part B: On the problem of osculatory interpolation. A second class of analytic approximation formulae, *ibid.*, 112–141.

Schoenberg, I. J. [1973]. *Cardinal spline interpolation*, CBMS Regional Conf. Ser. Appl. Math., no. 12, SIAM, Philadelphia, PA.

Schumaker, Larry L. [2007]. *Spline functions: Basic theory*, 3d ed., Cambridge Math. Lib., Cambridge University Press, Cambridge.

Schwab, Ch. [1998]. *p- and hp-finite element method: Theory and applications in solid and fluid mechanics*, Numer. Math. Scient. Comput., Oxford University Press, Oxford, New York.

Schwarz, H.-R. [1989]. *Numerical analysis: A comprehensive introduction*, with a contribution by J. Waldvogel, translated from the German, Wiley, Chichester.

Scott, L. Ridgway, Clark, Terry, and Bagheri, Babak [2005]. *Scientific parallel computing*, Princeton University Press, Princeton, NJ.

Sehmi, N. S. [1989]. *Large order structural eigenanalysis techniques: Algorithms for finite element systems*, Ellis Horwood Ser. Math. Appl., Halsted Press, New York.

Sewell, Granville [2005]. *The numerical solution of ordinary and partial differential equations*, 2d ed., Pure Appl. Math. (New York), Wiley-Interscience, Hoboken, NJ.

Shaĭdurov, V. V. [1995]. *Multigrid methods for finite elements*, translated from the 1989 Russian original by N. B. Urusova and revised by the author, Math. Appl., v. 318, Kluwer, Dordrecht.

Shampine, L. F. [1975]. Discrete least squares polynomial fits, *Comm. ACM 18*, 179–180.

Shampine, Lawrence F. [1994]. *Numerical solution of ordinary differential equations*, Chapman & Hall, New York.

Shampine, L. F. and Gordon, M. K. [1975]. *Computer solution of ordinary differential equations: The initial value problem*, Freeman, San Francisco, CA.

Shampine, L. F. and Wisniewski, J. A. [1978]. *The variable order Runge–Kutta code RKSW and its performance*, Sandia Rep. SAND78-1347, Sandia National Laboratories, Albuquerque, NM.

Shampine, L. F., Gladwell, I., and Thompson, S. [2003]. *Solving ODEs with MATLAB*, Cambridge University Press, Cambridge.

Sheynin, Oscar [1993]. On the history of the principle of least squares, *Arch. Hist. Exact Sci. 46*, 39–54.

Shonkwiler, Ronald W. and Mendivil, Franklin [2009]. *Explorations in Monte Carlo methods*, Undergrad. Texts Math., Springer, New York.

Sidi, Avram [2003]. *Practical extrapolation methods: Theory and applications*, Cambridge Monographs Appl. Comput. Math., v. 10, Cambridge University Press, Cambridge.

Sikorski, Krzystof A. [2001]. *Optimal solution of nonlinear equations*, Oxford University Press, Oxford.

Sima, Vasile [1996]. *Algorithms for linear-quadratic optimization*, Monographs and Textbooks Pure Appl. Math., v. 200, Dekker, New York.

Sloan, I. H. and Joe, S. [1994]. *Lattice methods for multiple integration*, Oxford Sci. Publ., Oxford University Press, New York.

Smale, Steve [1987]. Algorithms for solving equations, in *Proc. Internat. Congress Mathematicians*, Andrew M. Gleason, ed., v. 1, 172–195, Amer. Math. Soc., Providence, RI.

Smith, Barry F., Bjørstad, Petter E., and Gropp, William D. [1996]. *Domain decomposition: Parallel multilevel methods for elliptic partial differential equations*, Cambridge University Press, Cambridge.

Smith, B. T., Boyle, J. M., Dongarra, J. J., Garbow, B. S., Ikebe, Y., Klema, V. C., and Moler, C. B. [1976]. *Matrix eigensystem routines — EISPACK guide*, 2d ed., Lecture Notes in Comput. Sci., v. 6, Springer, Berlin.

Sobol', Ilya M. [1994]. *A primer for the Monte-Carlo method*, CRC Press, Boca Raton, FL.

Sobolev, S. L. and Vaskevich, V. L. [1997]. *The theory of cubature formulas*, translated from the 1996 Russian original and with a foreword by S. S. Kutateladze, revised by Vaskevich, Math. Appl., v. 415, Kluwer, Dordrecht.

Sottas, G. [1982]. On the positivity of quadrature formulas with Jacobi asbscissas, *Computing 29*, 83–88.

Sottas, G. [1988]. Positivity domain of ultraspherical type quadrature formulas with Jacobi asbscissas: Numerical investigations, in *Numerical integration III*, 285–294, Internat. Schriftenreihe Numer. Math., v. 85, Birkhäuser, Basel.

Sottas, G. [1989]. On the positivity of ultraspherical type quadrature formulas with Jacobi abscissas, *Aequationes Math. 38*, 10–27.

Späth, Helmuth [1992]. *Mathematical algorithms for linear regression*, translated and revised from the 1987 German original by the author, Comput. Sci. Scient. Comput., Academic Press, Boston, MA.

Späth, Helmuth [1995]. *One-dimensional spline interpolation algorithms*, with the collaboration of Jörg Meier, translated from the German by Len Bos, A K Peters, Wellesley, MA.

Stanoyevitch, Alexander [2005]. *Introduction to MATLAB® with numerical preliminaries*, Wiley-Interscience, Hoboken, NJ.

Steffensen, J. F. [1950]. *Interpolation*, 2d ed., Chelsea, New York.

Stegun, Irene A. and Abramowitz, Milton [1956]. Pitfalls in computation, *J. Soc. Indust. Appl. Math. 4*, 207–219.

Steinbach, Olaf [2008]. *Numerical approximation methods for elliptic boundary value problems: Finite and boundary elements*, translated from the 2003 German original, Springer, New York.

Stenger, Frank [1993]. *Numerical methods based on sinc and analytic functions*, Springer Ser. Comput. Math., v. 20, Springer, New York.

Stenger, Frank [2000]. Summary of Sinc numerical methods, Numerical analysis in the 20th century, v. 1, Approximation theory, *J. Comput. Appl. Math. 121*, 379-420.

Stenger, Frank [2011]. *Handbook of Sinc numerical methods*, CRC Press, Boca Raton.

Stepleman, R. S. and Winarsky, N. D. [1979]. Adaptive numerical differentiation, *Math. Comp. 33*, 1257–1264.

Sterbenz, Pat H. [1974]. *Floating-point computation*, Prentice-Hall Ser. Automat. Comput., Prentice-Hall, Englewood Cliffs, NJ.

Stetter, Hans J. [1973]. *Analysis of discretization methods for ordinary differential equations*, Springer Tracts Nat. Philos., v. 23, Springer, New York.

Stewart, G. W. [1973]. *Introduction to matrix computations*, Comput. Sci. Appl. Math., Academic Press, New York.

Stewart, G. W. [1998]. *Matrix algorithms*, v. 1: *Basic decompositions*, SIAM, Philadelphia, PA.

Stewart, G. W. [2001]. *Matrix algorithms*, v. 2: *Eigensystems*, SIAM, Philadelphia, PA.

Stewart, William J. [1994]. *Introduction to the numerical solution of Markov chains*, Princeton University Press, Princeton, NJ.

Stiefel, E. and Bettis, D. G. [1969]. Stabilization of Cowell's method, *Numer. Math. 13*, 154–175.

Stiefel, E. L. and Scheifele, G. [1971]. *Linear and regular celestial mechanics: Perturbed two-body motion, numerical methods, canonical theory*, Grundlehren Math. Wiss., v. 174, Springer, New York.

Stoer, J. and Bulirsch, R. [2002]. *Introduction to numerical analysis*, translated from the German by R. Bartels, W. Gautschi, and C. Witzgall, 3d ed., Texts in Appl. Math., v. 12, Springer, New York.

Strikwerda, John C. [2004]. *Finite difference schemes and partial differential equations*, 2d ed., SIAM, Philadelphia, PA.

Stroud, A. H. [1971]. *Approximate calculation of multiple integrals*, Prentice-Hall Ser. Automat. Comput., Prentice-Hall, Englewood Cliffs, NJ.

Stroud, A. H. and Secrest, Don [1966]. *Gaussian quadrature formulas*, Prentice-Hall, Englewood Cliffs, NJ.

Szabados, J. and Vértesi, P. [1990]. *Interpolation of functions*, World Scientific, Teaneck, NJ.

Szegő, Gabriel [1936]. On some Hermitian forms associated with two given curves of the complex plane, *Trans. Amer. Math. Soc. 40*, 450–461. [Collected Papers, Richard Askey, ed., v. 2, 666–683, Birkhäuser, Boston, 1982.]

Szegö, Gabor [1975]. *Orthogonal polynomials*, 4th ed., Amer. Math. Soc. Colloq. Publ., v. 23, Amer. Math. Soc., Providence, RI.

Taylor, Walter F. [1992]. *The geometry of computer graphics*, The Wadsworth & Brooks/Cole Math. Ser., Wadsworth & Brooks/Cole, Pacific Grove, CA.

Thisted, Ronald A. [1988]. *Elements of statistical computing: Numerical computation*, Chapman & Hall, New York.

Thomas, J. W. [1995]. *Numerical partial differential equations: Finite difference methods*, Texts in Appl. Math., v. 22, Springer, New York.

Thomas, J. W. [1999]. *Numerical partial differential equations: Conservation laws and elliptic equations*, Texts in Appl. Math., v. 33, Springer, New York.

Thomée, Vidar [2006]. *Galerkin finite element methods for parabolic problems*, 2d ed., Springer Ser. Comput. Math., v. 25, Springer, Berlin.

Tihonov, A. N. and Gorbunov, A. D. [1963]. Asymptotic error bounds for the Runge–Kutta method (Russian), *Ž. Vyčisl. Mat. i Mat. Fiz. 3*, 195–197. [Engl. transl. in *U.S.S.R. Comput. Math. Math. Phys. 3*, no. 1 (1963), 257–261.]

Tihonov, A. N. and Gorbunov, A. D. [1964]. Error estimates for a Runge–Kutta type method and the choice of optimal meshes (Russian), *Ž. Vyčisl. Mat. i Mat. Fiz. 4*, 232–241. [Engl. transl. in *U.S.S.R. Comput. Math. Math. Phys. 4*, no. 2 (1964), 30–42.]

Titchmarsh, E. C. [1939]. *The theory of functions*, 2d ed., Oxford University Press, Oxford.

Todd, John [1950]. Notes on modern numerical analysis. I. Solution of differential equations by recurrence relations, *Math. Tables Aids Comput. 4*, 39–44.

Todd, John [1954]. The condition of the finite segments of the Hilbert matrix, in *Contributions to the solution of systems of linear equations and the determination of eigenvalues*, Olga Taussky, ed., 109–116, NBS Appl. Math. Ser., v. 39, U.S. Government Printing Office, Washington, DC.

Todd, John [1979, 1977]. *Basic numerical mathematics*. v. 1: *Numerical analysis*, Internat. Ser. Numer. Math., v. 14, Birkhäuser, Basel-Boston; v. 2: *Numerical algebra*, Internat. Ser. Numer. Math., v. 22, Birkhäuser, Basel-Stuttgart.

Toselli, Andrea and Widlund, Olof [2005]. *Domain decomposition methods: Algorithms and theory*, Springer, Berlin.

Traub, J. F. [1964]. *Iterative methods for the solution of equations*, Prentice-Hall Ser. Automat. Comput., Prentice-Hall, Englewood Cliffs, NJ.

Traub. J. F. and Werschulz. A. G. [1998]. *Complexity and information*, Lezioni Lincee, Cambridge University Press, Cambridge.

Traub, Joe Fred and Woźniakowski, H. [1980]. *A general theory of optimal algorithms*, ACM Monograph Ser., Academic Press, New York.

Traub, Joseph Frederick, Wasilkowski, G. W., and Woźniakowski, H. [1983]. *Information, uncertainty, complexity*, Addison-Wesley, Reading, MA.

Traub, J. F., Wasilkowski, G. W., and Woźniakowski, H. [1988]. *Information-based complexity*, with contributions by A. G. Werschulz and T. Boult, Comput. Sci. Scient. Comput., Academic Press, Boston, MA.

Trefethen, Lloyd N. [2000]. *Spectral methods in MATLAB*, Software, Environments, and Tools, v. 10, SIAM, Philadelphia, PA.

Trefethen, L. N. and Weideman, J. A. C. [1991]. Two results on polynomial interpolation in equally spaced points, *J. Approx. Theory 65*, 247–260.

Trefethen, Lloyd N. and Bau, David,III [1997]. *Numerical linear algebra*, SIAM, Philadelphia, PA.

Troesch, B. A. [1976]. A simple approach to a sensitive two-point boundary value problem, *J. Comput. Phys. 21*, 279–290.

Tucker, Warwick [2011]. *Validated numerics: a short introduction to rigorous computations*, Princeton University Press, Princeton, NJ.

Turán, P. [1950]. On the theory of the mechanical quadrature, *Acta Sci. Math. Szeged 12*, 30–37.

Tveito, Aslak and Winter, Ragnar [2009]. *Introduction to partial differential equations: A computational approach*, Paperback reprint of the 2005 edition, Texts in Appl. Math., v. 29, Springer, Berlin.

Van Assche, Walter and Vanherwegen, Ingrid [1993]. Quadrature formulas based on rational interpolation, *Math. Comp. 61*, 765–783.

Van de Velde, Eric F. [1994]. *Concurrent scientific computing*, Texts in Appl. Math., v. 16, Springer, New York.

van der Laan, C. G. and Temme, N. M. [1984]. *Calculation of special functions: The gamma function, the exponential integrals and error-like functions*, CWI Tract, v. 10, Stichting Mathematisch Centrum, Amsterdam.

van der Vorst, Henk A. [2009]. *Iterative Krylov methods for large linear systems*, reprint of the 2003 original, Cambridge Monographs Appl. Comput Math., v. 13, Cambridge University Press, Cambridge.

Van Loan, Charles [1992]. *Computational frameworks for the fast Fourier transform*, Frontiers Appl. Math., v. 10, SIAM, Philadelphia, PA.

Varga, Richard S. [1990]. *Scientific computation on mathematical problems and conjectures*, CBMS-NSF Regional Conf. Ser. Appl. Math., no. 60, SIAM, Philadelphia, PA.

Varga, Richard S. [2000]. *Matrix iterative analysis*, second revised and expanded edition, Springer Ser. Comput. Math., v. 27, Springer, Berlin.

Verner, J. H. [1978]. Explicit Runge–Kutta methods with estimates of the local truncation error, *SIAM J. Numer. Anal. 15*, 772–790.

Walsh, J. L. [1969]. *Interpolation and approximation by rational functions in the complex domain*, 5th ed., Amer. Math. Soc. Colloq. Publ., v. 20, Amer. Math. Soc., Providence, RI.

Walter, Gilbert G. [1994]. *Wavelets and other orthogonal systems with applications*, CRC Press, Boca Raton, FL.

Walz, Guido [1996]. *Asymptotics and extrapolation*, Math. Res., v. 88, Akademie Verlag, Berlin.

Wang, Ze Ke, Xu, Sen Lin, and Gao, Tang An [1994]. *Algebraic systems of equations and computational complexity theory*, with a preface by H. W. Kuhn, Math. Appl., v. 269, Kluwer, Dordrecht.

Wanner, G., Hairer, E., and Nørsett, S. P. [1978]. Order stars and stability theorems, *BIT 18*, 475–489.

Watkins, David S. [2002]. *Fundamentals of matrix computations*, 2d ed., Pure Appl. Math. (New York), Wiley-Interscience, New York.

Watkins, David S. [2007]. *The matrix eigenvalue problem: GR and Krylov subspace methods*, SIAM, Philadelphia, PA.

Watson, G. Alistair [1980]. *Approximation theory and numerical methods*, Wiley-Interscience, Wiley, Chichester.

Watson, Layne T., Billups, Stephen C., and Morgan, Alexander P. [1987]. Algorithm 652 — HOMPACK: A suite of codes for globally convergent homotopy algorithms, *ACM Trans. Math. Software 13*, 281–310.

Weierstrass, K. [1891]. Neuer Beweis des Satzes, dass jede ganze rationale Function einer Veränderlichen dargestellt werden kann als Product aus linearen Functionen derselben Veränderlichen, *Sitzungsber. Königl. Akad. Wiss.* [Math. Werke, v. 3, 251–269.]

Weiss, Rüdiger [1996]. *Parameter-free iterative linear solvers*, Math. Res., v. 97, Akademie Verlag, Berlin.

Werner, Wilhelm [1982]. On the simultaneous determination of polynomial roots, in *Iterative solution of nonlinear systems of equations*, R. Ansorge, Th. Meis, and W. Törnig, eds., 188–202, Lecture Notes in Math., v. 953, Springer, Berlin.

Werner, Wilhelm [1984]. Polynomial interpolation: Lagrange versus Newton, *Math. Comp. 43*, 205–217.

Werschulz, Arthur G. [1991]. *The computational complexity of differential and integral equations: An information-based approach*, Oxford Math. Monographs, Oxford Sci. Publ., Oxford University Press, New York.

White, Robert E. [2007]. *Elements of matrix modeling and computing with MATLAB*, Chapman & Hall/CRC, Boca Raton, FL.

Wickerhauser, Mladen Victor [1994]. *Adapted wavelet analysis from theory to software*, A K Peters, Wellesley, MA.

Widlund, Olof B. [1967]. A note on unconditionally stable linear multistep methods, *BIT 7*, 65–70.

Wilkinson, J. H. [1994]. *Rounding errors in algebraic processes*, reprint of the 1963 original, Dover Publ., New York.

Wilkinson, James H. [1984]. The perfidious polynomial, in *MAA Stud. Math.*, v. 24: *Studies in numerical analysis*, Gene H. Golub, ed., 1–28, Math. Assoc. America, Washington, DC.

Wilkinson, J. H. [1988]. *The algebraic eigenvalue problem*, Monographs Numer. Anal., Oxford Sci. Publ., Oxford University Press, New York.

Wimp, Jet [1981]. *Sequence transformations and their applications*, Math. Sci. Engrg., v. 154, Academic Press, New York.

Wimp, Jet [1984]. *Computation with recurrence relations*, Applicable Math. Ser., Pitman, Boston, MA.

Wolfe, Michael Anthony [1978]. *Numerical methods for unconstrained optimization: An introduction*, Van Nostrand Reinhold, New York.

Woźnicki, Zbigniew Ignacy [2009]. *Solving linear systems: An analysis of matrix prefactorization iterative methods*, with a foreword by Richard S. Varga, Matrix Editions, Ithaca, NY.

Wright, K. [1970]. Some relationships between implicit Runge–Kutta, collocation Lanczos τ methods, and their stability properties, *BIT 10*, 217–227.

Xiu, Dongbin [2010]. *Numerical methods for stochastic computing: A spectral method approach*, Princeton University Press, Princeton.

Xu, Shu-fang [1998]. *An introduction to inverse algebraic eigenvalue problems*, Vieweg & Sohn, Braunschweig.

Young, David M. [2003]. *Iterative solution of large linear systems*, unabridged republication of the 1971 edition, Dover Publ., Mineola, NY.

Young, David M. and Gregory, Robert Todd [1988]. *A survey of numerical mathematics*; v. 1, corrected reprint of the 1972 original; v. 2, corrected reprint of the 1973 original, Dover Publ., New York.

Ypma, Tjalling J. [1995]. Historical development of the Newton–Raphson method, *SIAM Rev. 37*, 531–551.

Zayed, Ahmed I. [1993]. *Advances in Shannon's sampling theory*, CRC Press, Boca Raton, FL.

Zienkiewicz, O. C. and Taylor, R. L. [2000]. *The finite element method*; v. 1: *The basis*; v. 2: *Solid mechanics*; v. 3: *Fluid dynamics*; 5th ed., Butterworth-Heinemann, Oxford.

Zlatev, Zahari [1991]. *Computational methods for general sparse matrices*, Math. Appl., v. 65, Kluwer, Dordrecht.

Zwillinger, Daniel [1992a]. *Handbook of integration*, Jones and Bartlett, Boston, MA.

Zwillinger, Daniel [1992b]. *Handbook of differential equations*, 2d ed., Academic Press, Boston, MA.

Zygmund, A. [2002]. *Trigonometric series*, vols. 1, 2, 3d ed., with a foreword by Robert A. Fefferman, Cambridge Math. Lib., Cambridge University Press, Cambridge.

Index

W. Gautschi, *Numerical Analysis*, DOI 10.1007/978-0-8176-8259-0,