REVIEWS AND DESCRIPTIONS OF TABLES AND BOOKS

The numbers in brackets are assigned according to the American Mathematical Society classification scheme. The 1991 Mathematics Subject Classification can be found in the annual subject index of *Mathematical Reviews* starting with the December 1990 issue.

20[65–01, 68N15].—JEAN-ETIENNE ROMBALDI, *Algorithmique Numérique et ADA*, Collection Logique Mathématiques Informatique, Vol. 12, Masson, Paris, 1993, xx + 336 pp., 24 cm. Price: Softcover F 260.

This is a book on standard numerical methods which differentiates itself from other similar books by the fact that it contains implementations of the algorithms in the ADA language. Indeed, one of its aims, as claimed in the preface, is to show that ADA is a suitable language for mathematical programming. It is not meant to be a book on programming with ADA. In fact, the selection of numerical analysis topics that are covered is quite good. However, some of the methods presented are now outdated and should have been replaced by more modern ones. In particular, I am referring to the section on eigenvalue problems (Section 4, in Chapter 3), where all the methods based on characteristic polynomials may have been omitted and replaced by the usual QR/Householder approach.

This book comes at a time when object-oriented programming is gaining ground in many areas. Although FORTRAN, and increasingly C, are still the preferred languages in scientific computing, it is argued in this book that ADA has some important advantages, which makes it particularly suitable for scientific computing. In particular, it allows components to be reusable, a feature of object-oriented programming, and tends to be an effective language in which to develop and maintain large applications software.

This is the only book I know of, in French or English, which presents Numerical Analysis from the viewpoint of ADA, or any Object Oriented Programming language. Whether ADA is a good choice is hard to judge now. It is certainly an important language, but C++ is currently more popular, although there are currently some inefficiencies in C++ which make it a little unattractive for real scientific applications. This book would certainly be helpful to those wishing to start programming scientific applications in ADA as opposed to a more traditional language. It is its main attraction since there are alternative numerical analysis textbooks, both in French and in English, which offer a better coverage of the numerical analysis material. In addition, the book can also be viewed as a nice introduction to object-oriented programming and software development in scientific computing.

The following is a brief list of contents of the book. Chapter 1 is an overview of the language ADA with a nice historical perspective and some parts on motivation. It also presents Open ADA, a compiler which is available on PC platforms. Chapter 2 describes a library of utilities which is used throughout the book. These include graphics functions for PCs and some standard functions such as random number generators, timers, and standard mathematical functions. Some of these utilities have a somewhat limited portability, but many of them are not publicly available and can be very useful to a first ADA user who is undertaking a sophisticated application in ADA. The other chapters deal with numerical techniques: Matrix computations (Chapter 3), Nonlinear systems (Chapter 4), Approximation and Interpolation (Chapter 5), Numerical Integration and FFTs (Chapter 6), Solution of Ordinary Differential Equations (Chapter 7), and Partial Differential Equations (Chapter 8).

A diskette containing a library of utilities and the programs discussed in the book is provided. The software is meant to be used in a PC platform, which could be a limiting factor for those programming in a Unix environment.

Yousef Saad

Department of Computer Science
University of Minnesota
Minneapolis, MN 55455

21[65N30, 73K05, 73K10, 73K15].—G. Prathap, *The Finite Element Method in Structural Mechanics: Principles and Practice of Design of Field-Consistent Elements for Structural and Solid Mechanics*, Solid Mechanics and Its Applications, Vol. 24, Kluwer, Dordrecht, 1993, x + 14 pp., $24\frac{1}{2}$ cm. Price $139.00/Dfl.245.00.

As the subtitle indicates, this book deals with the design of field-consistent elements for structural mechanics applications. To paraphrase the author, 'this book is not intended to be a text-book, delineating the full scope of finite element methodology, nor is it a comprehensive handbook of modern finite element practice for the finite element engineer.' It is a monograph on the design of finite element models of certain structural theories, based on the field-consistency concept the author has been promoting over the last decade. In fact, the book is composed of the papers the author and his colleagues published on the topic.

The book is divided into twelve chapters. Chapter 1 provides a general introduction and the author stresses the importance of "consistency" and "correctness" in finite element formulations. Chapter 2 contains a description of Timoshenko beam elements and a discussion on the associated shear locking phenomenon. The author describes the a priori error associated with the so-called inconsistent formulation and calls his approach "mathematically rigorous". While the approach is variationally correct/consistent, the mathematical rigor can be achieved only through the use of functional analysis tools. No such tools are used in this book to bring more rigor to what is covered. Chapter 3 deals with the membrane locking phenomenon and the simple curved beam element. Essentially, the field-consistent concept is studied with respect to shear flexible beam elements (Chapter 2), curved beam elements, assumed-strain formulations, plane stress elements, plate elements, brick elements, shell elements (Chapters 3–10), and a few other special elements (Chapter 11). Finally, the author presents a philosophical summary of the contents of the entire book in Chapter 12.

There are several positive aspects to the monograph. It is written in a simple language (despite the author's inclination to use terms like "functional re-constitution"!). The author provides a detailed description of the field-consistency issues, and studies the application of the concept to a whole range of elements (from beam elements to shell elements). The emphasis on 'variational correctness' of the finite element formulations, the subsequent discussions on assumed-strain formulations, and the discussion of the formulations in a more general setting of the Hu-Washizu principle, make the monograph useful.

The reviewer also has a few critical observations. The organization of the book is a little confusing, although the overall structure is good. The organization within the chapters is not very clear. Since sections and subsections have not been identified in the contents page, the reader is not likely to get a good overview of the structure of individual chapters.

The author has a tendency to emphasize concepts repeatedly throughout the book. While repeating concepts and definitions once or twice helps in driving the point home, repeated references hinder the flow of reading. Most of the sections end with concluding remarks, the chapters end with conclusions, and the entire book is summarized in Chapter 12. Perhaps, the final chapter could have been much shorter.

An additional comment pertains to the author's extremely critical view of other finite element techniques to explain locking and measures to alleviate locking. While the author has done an admirable job of describing the field-consistency techniques, he fails to describe at length the other techniques (he has merely stated these techniques in words) and their drawbacks to justify his comments. Perhaps this might seem unnecessary to a person who is aware of these techniques, but to such a person this book will be less useful. However, for a student or a practicing engineer, a complete description of all available methods is necessary before pointing out their deficiencies. Also, the author is either not aware of, or he chose to ignore, many pertinent references on the subject.

While the book has its deficiencies, it will serve as a useful reference book on finite element models of beams, plates, and shells.

J. N. Reddy

Department of Mechanical Engineering
Texas A&M University
College Station, TX 77843

22[65N30, 65N50, 65N55].—Randolph E. Bank, *PLTMG: A Software Package for Solving Elliptic Partial Differential Equations, Users' Guide* 7.0, Frontiers in Applied Mathematics, Vol. 15, SIAM, Philadelphia, PA, 1994, xii + 128 pp., $25\frac{1}{2}$ cm. Price: Softcover $24.50.

This book is a "must have" for anyone planning on using the PLTMG software package for the solution of second-order elliptic boundary value problems in two dimensions. The software can be obtained at no cost by anonymous ftp from *netlib* at netlib.att.com in the directory netlib/pltmg, or from *mgnet* at casper.cs.yale.edu in the directory mgnet/pltmg. This book is the only user-oriented documentation available for PLTMG.

PLTMG solves nonlinear second-order elliptic partial differential equations in general two-dimensional regions, with Dirichlet boundary conditions on part of the boundary and natural boundary conditions on the remainder of the boundary. It uses finite element discretizations based on C^0 piecewise linear triangular elements, solves the nonlinear systems with a damped Newton method, and the linear systems with a conjugate gradient method preconditioned by the hierarchical basis multigrid method. It contains a pseudoarclength continuation procedure for parameter dependencies. The package also contains an initial mesh generator, graphics capabilities for several hardware devices, including the X-Window System and PostScript, and the sample test problems that were used to generate the figures in the book. PLTMG was originally developed over 10 years ago as a research code to study multigrid and adaptive refinement methods, and has matured into a stable, well-respected, and widely distributed software package. Version 7.0 was released in early 1994.

The book is organized in seven chapters. The first chapter introduces the class of problems that PLTMG can solve, and the global organization of the software package. The second chapter describes, in full detail, the various data structures that the users must understand in order to use the package to solve their own problem. The third and fourth chapters describe the two principal procedures of the package, mesh generation and equation solution. The fifth chapter describes how to invoke various graphical options available in the package. The sixth and seventh chapters describe the test driver program and example problems provided with the package.

I found the book to be very readable and well organized as a tutorial on PLTMG. All the information required to use the program to solve complicated boundary value problems is available. The depth of the discussions concerning the underlying mathematical methods is intentionally shallow, providing only the detail required to fully understand the user interface. An extensive bibliography is provided to point the interested reader to further details.

As a users' guide to a software package, this book could be improved in two ways. First, there is no provision for quickly learning to use a subset of the program's capabilities. I was able to run the example test problems after reading just Chapters 1, 6 and 7, but to write a program that will solve a different problem, it is necessary to read the entire book. A brief introduction, or "quick start" chapter that provides enough information to write a program that solves a simple problem (for example, Poisson's equation on a square) would be a valuable asset. The full detail contained in Chapters 2, 3 and 4 would still be required to write a program that solves more complicated "real-world" problems.

Second, although the book is an excellent tutorial, the organization makes it difficult to use the book as a reference manual. For example, in order to set the appropriate parameters before calling one of the major procedures, one must first look in the book's Table 2.4 to determine which parameters are used by the procedure, and then look up each of them in the index to determine what page describes the options for that parameter. The addition of a "quick reference guide" that summarizes the parameter values and procedure calling sequences would be very useful.

Anyone who intends to use the PLTMG software will need a copy of this book. Despite the aforementioned deficiencies, the book contains all the information needed to solve complicated boundary value problems, and is written in a clear, easy-to-understand style.

WILLIAM F. MITCHELL

Applied and Computational Mathematics Division
National Institute of Standards and Technology
Gaithersburg, MD 20899

23[65M55, 65N55].—P. W. HEMKER & P. WESSELING (Editors), *Contributions to Multigrid*, CWI Tract, Vol. 103, Centre for Mathematics and Computer Science, Amsterdam, 1994, viii + 220 pp., 24 cm. Price: Softcover Dfl. 60.00.

From the Preface: This volume contains a selection from the papers presented at the Fourth European Multigrid Conference, held in Amsterdam, July 6–9, 1993.

J. H. B.

24[65L05].—LAWRENCE F. SHAMPINE, *Numerical Solution of Ordinary Differential Equations*, Chapman & Hall, New York, 1994, x + 484 pp., $23\frac{1}{2}$ cm. Price $64.95.

That so many books with more or less this same title have appeared in recent years might lead one to expect nothing new in this volume. This is far from the case. Numerical methods for differential equations is a very difficult and important subject and, while its literature is extremely rich, it is far from mature. Recent textbooks, and more so monographs, are not so much personal expositions of a well-defined body of work but personal contributions to the development of a vital and rapidly-changing research area. Lawrence Shampine has been a significant contributor to the theory and practice of solving differential equations numerically for 20 years and it is his style, developed and honed through his own research and experience, that is stamped on this book.

The book is divided into eight chapters of which the first three are of an introductory nature. The first deals with "The Mathematical Problem" (of solving ordinary differential equations), the second with "Discrete Variable Methods", and the third "The Computational Problem". Chapter four on "Basic Methods" is followed by the theory of "Convergence and Stability". The last three chapters are on "Stability for Large Step Sizes", "Error Estimation and Control" and "Stiff Problems".

References are collected together for standard literature, works actually cited, and codes referred to. The book concludes with a brief appendix on mathematical tools used in the book.

The subject of solving differential equations numerically is a mix of theoretical knowledge, practical insight and computational technique. In the Shampine style, software is also an essential component and it is the balance in emphasis between this and the other aspects of the subject that makes this book especially attractive.

JOHN C. BUTCHER

Department of Mathematics and Statistics
University of Auckland
Auckland, New Zealand

25[34C35, 58F05, 65L06, 65L07].—J. M. SANZ-SERNA & M. P. CALVO, *Numerical Hamiltonian Problems*, Applied Mathematics and Mathematical Computation, Vol. 7, Chapman & Hall, London, 1994, xii + 207 pp., 22 cm. Price $48.50.

The story of numerical analysis is mostly about translating mathematical concepts and models into a quantitative medium, fleshing numbers on the formal mathematical skeleton. This, however, should not obscure the crucial role of *qualitative* aspects of computation. It is an illusion that, as soon as it comes to application, all questions are of a purely quantitative character. 'Will the satellite stay in stable orbit?', 'Does a mixture undergo combustion?', 'Will the species survive in a given environment?', are all qualitative questions. Presumably, they are modelled by differential equations. Presumably, these differential equations are solved numerically. Certainly, their numerical solution contains errors. Naively, the purpose of numerical analysis is to minimize the accretion of error, but this frequently misses the point of the whole calculation. Thus, let us consider the stability of a satellite and suppose that two alternative computational methods are available. The first produces a very small error which, however, consistently undershoots the elevation. The second is considerably more error-prone but gets the stability issue exactly right: the numerical orbit is stable if and only if so is the exact one. Little doubt that, for the specific purpose in hand, the second method is superior!

The emphasis on the recovery of qualitative attributes of a mathematical model, rather than just minimizing the error, is relatively a new one. It has led in the last decade to a profound new insight into computation and has changed the treatment of many important numerical problems. Arguably, the most significant advance has been associated with Hamiltonian problems.

A system of ordinary differential equations is said to be Hamiltonian if it can be represented in the form

$$\frac{d\mathbf{p}}{dt} = -\frac{\partial H(\mathbf{p},\mathbf{q})}{\partial \mathbf{q}},$$
$$\frac{d\mathbf{q}}{dt} = \frac{\partial H(\mathbf{p},\mathbf{q})}{\partial \mathbf{p}},$$

where H is a given C^1 function. A significant proportion of dynamical systems that occur in mechanics—classical and quantum alike—can be rendered in a Hamiltonian form. As Penrose comments, "Such unity of form in the structure of dynamical equations, despite all the revolutionary changes that have occurred in physical theories over the past century or so is truly remarkable!" [3].

The formulation of equations of motion by William Rowan Hamilton in the above form was highly significant for sound physical reasons. The letter $\mathbf{p}$ stands for positions and $\mathbf{q}$ for momenta of physical particles, whilst H is the total energy. This dichotomy between position and momentum (and the consignment of velocity to an inferior role) was a crucial conceptual advance that augured the development of quantum mechanics. The practical significance of Hamiltonian systems, however, has less to do with their august historical role than with their intricate dynamics.

To put it in a nutshell, the problem with Hamiltonian equations is that they possess too much structure. For starters, total energy is conserved, $H(\mathbf{p}(t), \mathbf{q}(t)) \equiv \mathrm{const}$. So is an infinity of other integral quantities. More importantly, the flow $\Phi_{H,t}(\mathbf{p}(0), \mathbf{q}(0)) = (\mathbf{p}(t), \mathbf{q}(t))$, which assigns the solution at fixed time t to an initial condition, has a *symplectic* property: if $\Phi_{H,t}(\Omega) = \tilde{\Omega}$, then the sum of the areas of the projections into all two-dimensional spaces (p_j, q_j) is the same for Ω and $\tilde{\Omega}$. Moreover, as long as we restrict our attention to simply-connected sets Ω, symplecticity characterizes Hamiltonian systems.

Symplecticity is at the root of a whole range of interesting phenomena. Thus, it implies that fixed points of the flow are surrounded in the phase space by stable centres—the phase plane of the humble linear equation $y'' + y = 0$ is a case in point.

Experienced applied mathematicians know that often there is a price to be paid for conservation. In the familiar case of nonlinear hyperbolic conservation laws, the price of conservation is discontinuity. For Hamiltonian systems it is the sheer complexity of the flow: chaotic trajectories are a rule rather than an exception. The situation becomes even more complicated in the case of numerical modelling—after all, centres are known as the most volatile dynamical objects. Every tiny perturbation (and what is a numerical error if not a perturbation?!) is likely to turn a centre into a (stable or unstable) spiral. Inasmuch as the numerical error per se might be small (at least, for solution intervals of moderate length), the qualitative picture becomes misleading.

Hamiltonian systems have been computed (often badly) for many decades, but only in the early Eighties had a serious effort been expanded to design algorithms that preserve symplecticity, thereby ensuring that the numerical solution is faithful to original differential equations. As often, both the impetus and early results originated in the work of mathematical physicists [1, 4]. They have employed algorithms based on generating functions which, their ingenuity notwithstanding, are labour-intensive and need be derived afresh—by a painful expansion into series—for virtually every new Hamiltonian.

A turning point in the history of symplectic integrators came in 1988, when three numerical analysts (F. M. Lasagni, J. M. Sanz-Serna and Y. B. Suris [2, 5, 7]) independently proved that, subject to a satisfaction of an algebraic condition, Runge-Kutta methods are symplectic. Specifically, if $A = (a_{k,l})$ is the RK matrix and $\mathbf{b} = (b_l)$ are the RK weights of an s-stage Runge-Kutta, the symplecticity condition is $b_k a_{k,l} + b_l a_{l,k} = b_k b_l$, $k, l = 1, 2, \ldots, s$. Similar conditions can be derived for Runge-Kutta-Nyström and for partitioned Runge-Kutta methods (the latter apply different schemes to $\mathbf{p}$ and $\mathbf{q}$). In particular, the familiar Gauß-Legendre methods are symplectic. Even more remarkably, in the important special case of the Hamiltonian being of the form $H(\mathbf{p}, \mathbf{q}) = \frac{1}{2}\mathbf{p}^T\mathbf{p} + V(\mathbf{q})$, the underlying equations can be solved by an *explicit* Runge-Kutta-Nyström method which is symplectic!

This pioneering work has been followed by a large number of publications and generated a great deal of interest, not only among numerical analysts but also in the nonlinear dynamical systems community. The present monograph of Jesus Maria Sanz-Serna and Mari Paz Calvo, both leaders in the symplectic field, is very timely indeed. It started its life as an *Acta Numerica* survey [6] and has been subsequently expanded to a book form.

Writing a monograph for numerical analysts, symplectic geometers and users of Hamiltonian systems—astronomers, molecular biologists, mathematical physicists, quantum chemists, engineers, fluid dynamicists—presents a significant challenge, because of the great variability in numerical and mathematical proficiency of the target audience. Some will know good dynamics and no numerics, others good numerics and no dynamics. Yet others will know bad numerics and bad dynamics. A good monograph should dispel ignorance without putting to the test the attention span of the more knowledgeable readers. Sanz-Serna and Calvo managed this task in a most splendid manner!

The first part of the book is devoted to an exposition of analytic aspects of Hamiltonian equations. The dosage of mathematical sophistication is just right to present main qualitative properties of Hamiltonian systems and highlight the importance of symplecticity, but the authors admirably resist the temptation to wade into the deep waters of symplectic geometry or the KAM theory—a detour which, although replete with beautiful mathematics, is not necessary to understand the subject matter of this monograph.

Having explained dynamics to numericists, the authors devote the second part of the book toward the explanation of numerics (specifically, Runge-Kutta methods) to dynamicists. This is inclusive of both theory—in the main, application of rooted trees in the derivation of order conditions—and implementation.

The first two parts of the book set the stage for its extensive survey of symplectic Runge-Kutta methods. The algebraic condition for symplecticity is derived for standard, partitioned and Nyström methods. Moreover, there is a detailed consideration of the surprising, yet important, result of L. Abia and J. M. Sanz-Serna, namely that order conditions and the symplecticity condition interact! As soon as the algebraic condition for symplecticity is imposed on the RK matrix, most order conditions go away. This makes the derivation of symplectic Runge-Kutta methods substantially simpler.

Available symplectic methods of all kinds—explicit, implicit, singly-diagonally implicit, composite—are surveyed in great detail. This is accompanied by extensive numerical experiments that demonstrate vividly that symplectic methods easily outperform state-of-the-art (nonsymplectic) Runge-Kutta schemes.

An important consequence of symplecticity is that the solution can be subjected to backward error analysis. It is possible to prove that a symplectic numerical trajectory samples (within a uniformly small error) the exact solution of nearly Hamiltonian equations. Therefore, *global* behaviour of the solution is captured correctly, since the numerical trajectory reflects the properties of this nearby Hamiltonian. However, symplecticity and all its benefits are lost as soon as the step size is being amended in the course of the numerical solution and governed by (local) error control considerations. This flies in the face of received numerical wisdom: *always* control the error, *always* use a variable-step algorithm.

There is no such thing as a free numerical lunch. Symplecticity means that the error cannot be controlled in the course of the solution: we must stick with an initial step length through thick and thin. Abandonment of symplecticity, on the other hand, spoils the qualitative picture, denies us the benefits of backward error analysis and in long-term integration brings about considerably faster accumulation of error. The authors debate this dilemma at some length and conclude that, provided the solution interval is relatively short, good (non-symplectic) Runge-Kutta methods, e.g., the Dormand-Prince algorithm, have the edge. However, there is little doubt that symplectic methods are superior when it comes to long-term integration.

The authors conclude with a long list of additional themes and extensions—generating functions, the Lie formalism, the Poisson bracket, generation of high-order symplectic Runge-Kutta... .

This is an important book on an important subject. As numerical analysis evolves, we are likely to witness growing interdependence of numerical and dynamical considerations. Numerical Hamiltonian equations are a showcase of this meeting of ideas and cultures, but its eventual influence is bound to spread significantly wider. No numerical analyst can afford to stay ignorant of this trend.

A. Iserles

Department of Applied Mathematics and Theoretical Physics
University of Cambridge
Cambridge CB3 9EW
England

1. P. J. Channell, *Symplectic integration algorithms*, Los Alamos National Laboratory Rep. AT-6:ATN 83-9 (1983).
2. F. M. Lasagni, *Canonical Runge-Kutta methods*, Z. Angew. Math. Phys. **39** (1988), 952–953.
3. R. Penrose, *The emperor's new mind*, Oxford Univ. Press, Oxford, 1989.
4. R. D. Ruth, *A canonical integration technique*, IEEE Trans. Nucl. Sci. **30** (1983), 2669–2671.
5. J. M. Sanz-Serna, *Runge-Kutta schemes for Hamiltonian systems*, BIT **28** (1988), 877–883.
6. ——, *Symplectic integrators for Hamiltonian problems: an overview*, Acta Numer. **1** (1993), 243–286.
7. Y. B. Suris, *Preservation of symplectic structure in the numerical solution of Hamiltonian systems*, Numerical Solution of Differential Equations (S. S. Filippov, ed.), Akad. Nauk USSR, Moscow, 1988, pp. 148–160.

26[65F10, 65-04].—Richard Barrett, Michael Berry, Tony F. Chan, James Demmel, June Donato, Jack Dongarra, Victor Eijkhout, Roldan Pozo, Charles Romine & Henk van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, Philadelphia, PA, 1994, xiv + 112 pp., $25\frac{1}{2}$ cm. Price: Softcover $18.00.

In 1971 James Wilkinson and Christian Reinsch edited a handbook of ALGOL programs for the solution of linear systems and eigenvalue problems [4]. The individual contributions had been previously published in *Numerische Mathematik* and were characterized by painstaking attention to detail, extensive testing, and complete documentation of both the algorithms and their implementations. Although the programming language ALGOL never caught on in the United States, the Handbook had an enormous influence on the later development of mathematical software. For my generation it was the place to turn to discover how to implement matrix algorithms. In addition it was the inspiration for a series of linear algebra packages: EISPACK [3] (a FORTRAN translation of the Handbook eigenvalue algorithms), LINPACK [2], and LAPACK [1].

The authors of the book under review are concerned that such packages do not serve all customers equally well. The packages, they claim, work well for people who want black boxes but not for people who want customized code for specialized applications. They go on to say that both groups can be satisfied by introducing templates. In their own words:

> A template is a description of a general algorithm rather than the executable object code or the source code more commonly found in a conventional software library. Nevertheless, although templates are general descriptions of key algorithms, they offer whatever degree of customization the user may desire. For example, they can be configured for the specific data structure of a problem or for the specific computing system on which the problem is run.

These claims, I believe, are not supported by the present volume, even for the simply structured iterative algorithms the authors consider. But by the way of compensation, the authors have produced a very useful survey of iterative methods circa 1994. For reasons I will get to later, I think any attempt to produce a set of templates will lead one willy-nilly into writing a survey. But for now, I will forget templates—the word itself does not seem to appear between the introduction and the appendices—and review the book as if it were titled *Iterative Methods for Linear Systems: An Algorithmic Survey.*

The meat of the book is contained in four chapters entitled Iterative Methods, Preconditioners, Related Issues, and Remaining Topics.

The chapter on iterative methods is devoted to describing the basic algorithms. The authors make the usual distinction between stationary methods, such as the Gauss-Seidel method, and nonstationary methods, such as the conjugate gradient method. Each method is illustrated by pseudocode consisting of a pleasing mix of mathematical notation and flow control. The coverage is quite complete, and the chapter is especially valuable as an introduction to Krylov methods for nonsymmetric systems, a hot topic at the moment. A regrettable omission is a treatment of look-ahead techniques, since some of the algorithms are not really functional without them. The chapter concludes with a well-organized summary.

The chapter on preconditioners is a concise survey. It treats Jacobi and block Jacobi preconditioners, incomplete factorization methods, and preconditioners based on underlying differential equations.

The chapter on related issues treats stopping criteria, data structures, and parallelism. The section on stopping criteria is highly technical and I think will prove more confusing than useful to the nonexpert. Little is said about what convergence criteria are suitable for specific algorithms, and the introduction may mislead some into thinking that an explicit residual computation must be done each time convergence is checked. The section on data structures, on the other hand, is a much needed survey of some of the conventions used in handling sparse matrices. Parallelism is a difficult topic, and the authors have handled it as well as they can in a small book. This section is really a collection of broad hints with references to the literature.

The last chapter is a miscellany, treating topics like domain decomposition and multigrid methods. The book concludes with three appendices, a glossary, and a bibliography with over two hundred entries.

The authors and their publisher, SIAM, are to be commended for making the book and its algorithms available electronically. The book itself suffers from minor lapses that are inevitable in a first edition. Moving some of the introductory material on preconditioning to the beginning of the chapter on iterative methods would help orient the reader. The highly technical term "gather" is introduced without definition. Some of the terms in the glossary did not make it to the index. But these are minor matters, easily corrected in the electronic version. I found the book an invaluable companion when I was teaching iterative methods, and I highly recommend it to anyone who has anything to do with the subject.

I cannot, on the other hand, recommend the idea of a template, at least as it is found in the present volume. There are three reasons. It is not new. It does not perform as advertised. And it represents a falling off from a high tradition.

The fact that templates are not new is easily seen by imagining someone reading the second chapter on iterative methods in isolation. He or she would have no idea that it was anything other than a standard exposition of the kind found in a book on matrix computations. True, theoretical aspects are not prominent and the pseudocode is in some cases slightly more elaborate than usual. But that can be accounted for by the algorithmic orientation of the authors. Without being told, our reader would never guess that a new idea had informed the exposition.

Regarding performance, the authors' assertion (see above) that their algorithms can be configured for the data structure in question is largely correct. But this says more about the simplicity of the matrix operations required to implement iterative methods than about the effectiveness of templates in general. In particular, the iterative methods treated here require nothing more than that the matrix operate on a vector, or some equally simple operation, which can be left to the user to implement. But it is hard to imagine a template for, say, an LU decomposition that permits the easy passage from a dense to a sparse matrix.

On other counts the templates do not fare as well. The authors evidently regard convergence checking as an implementation detail and lard their pseudocode with the statement "check convergence; continue if necessary." The user is presumably to go to the section on stopping criterion and figure out what to do. But it is not at all clear that anyone but an expert could arrive at a proper convergence test from the material presented there.

When the going gets tough, the authors punt. Regarding look-ahead strategies, they say, "This leads to complicated codes and is beyond the scope of this book." In treating the QMR method, which really needs look-ahead, they refer the reader to a black-box "professional implementation." If templates cannot handle this comparatively simple situation, how will they fare with more complicated algorithms—say a quasi-Newton optimization method?

All this, I believe, explains why the present book ended up as a survey. If you are going to publish algorithms abstracted from their implementations—the definition of template in the glossary—and still expect the user to arrive at working code, you have to give hints about how to write it. In this case the hints have expanded to fill three of the four substantive chapters of the book.

The scientific community has been well served by the tradition that spawned the Handbook, EISPACK, LINPACK, and LAPACK. Because of the exhaustive documentation of the first three packages—LAPACK provides only documentation on usage, not on algorithmic and implementation details—they have been studied and modified by people who want to learn and customize. They are in fact the true templates of numerical linear algebra, and it would be a shame if the best and brightest were to desert the tradition. As I said earlier, the authors have produced a very useful book that is worthy of wide distribution. But the reader should keep in mind that the best template is carefully crafted, thoroughly documented working code.

G. W. S.

1. E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK users' guide*, SIAM, Philadelphia, PA, 1992.

2. J. J. Dongarra, J. R. Bunch, C. B. Moler, and G. W. Stewart, *LINPACK users' guide*, SIAM, Philadelphia, PA, 1979.

3. B. T. Smith, J. M. Boyle, J. J. Dongarra, B. S. Garbow, Y. Ikebe, V. C. Klema, and C. B. Moler, *Matrix eigensystem routines—EISPACK guide*, 2nd ed., Springer-Verlag, New York, 1976.

4. J. H. Wilkinson and C. Reinsch, *Handbook for automatic computation, Vol.* II, *Linear algebra*, Springer, New York, 1971.

27[40A15, 30B70, 42C05, 33B15, 30E05].—S. Clement Cooper & W. J. Thron (Editors), *Continued Fractions and Orthogonal Functions: Theory and Applications*, Lecture Notes in Pure and Appl. Math., vol. 154, Dekker, New York, 1994, xiv + 379 pp., 25 cm. Price: Softcover $145.00.

The birth and early development of the general theory of orthogonal polynomials are found in the investigations of continued fractions by Stieltjes and Chebyshev. As the offspring grew, it soon went its separate way, developed an independent identity and all but forgot its roots. Recent years have seen a reconciliation of sorts as workers in orthogonal polynomials have rediscovered continued fractions and what they can do for the study of orthogonal polynomials and simultaneously, researchers in continued fractions have found new applications and generalizations involving orthogonal functions. Much of the credit for this renewal goes to the Colorado school of continued fractions and its Norwegian connection. The volume under review, containing the proceedings of a seminar-workshop held in Leon, Norway, reflects this fact, since the participants heavily represent the original members of this group and several generations of their students.

The longest of the sixteen papers in this volume, "Orthogonal Laurent polynomials on the real line," by Lyle Cochran and S. Clement Cooper, presents a comprehensive, self-contained survey of the title topic, which was first introduced by Jones, Thron and Waadeland in 1980. This summary should prove to be a useful introduction to this significant generalization of the classical moment problems and orthogonal polynomials. By contrast, a second fairly extensive survey, "Continued fraction representations for functions related to the Gamma function," by L. J. Lange, gives a number of continued fraction representations for a large variety of specialized complex functions which are related to the classical gamma function. Between these two expositions of generalization and specialization, the remaining papers included here present new results covering convergence questions and truncation error bounds for continued fractions and sequences of linear fractional transformations, applications to moment problems, orthogonal functions on the real line and the unit circle, Szegö and related polynomials, their zeros and applications to frequency analysis. In addition, the preface contains an interesting historical survey of the Colorado-Norway group which began with Wolf Thron and his early association with the late Arne Magnus.

T. S. Chihara

Department of Mathematics, Computer Science & Statistics
Purdue University Calumet
Hammond, IN 46323-2094

28[68-01, 68Q40].—Richard Zippel, *Effective Polynomial Computation*, Kluwer, Boston, 1993, xii + 363 pp., 24 cm. Price $87.50.

The field of computer algebra has gained widespread attention in recent years with the increased popularity and use of computer algebra systems such as Derive, Maple and Mathematica (and others) in the general scientific community. These systems have an extensive set of mathematical capabilities in such diverse areas as basic algebra (e.g., polynomial operations such as factorization and GCD computation) and analysis (e.g., determining closed-form solutions of integrals and differential equations). For mathematicians using these systems there is often a natural desire to learn more about the algorithms that are used in these systems. Unfortunately, there are very few suitable textbooks or survey articles that introduce mathematicians to these algorithms. Since the alternative is to search through a wide variety of research papers and Ph.D. theses, this makes the area a difficult one to begin research.

This text is meant as a one-semester course to introduce students and researchers to some of the fundamental algorithms used in computer algebra, in particular, for computation with polynomials. It can be used as a text for upper-year undergraduate or starting graduate students. The term "effective" used in the title could also be read as practical in the sense that the approach used is one of describing algorithms that work in practice, rather than only in theory.

The author makes the point that many of the algorithms of polynomial computation have their origins in computational number theory (e.g., Hensel lifting) and uses this as his starting point. The topics covered include continued fractions, solving Diophantine equations, and algorithms for polynomial computations such as factorization, interpolation, elimination and computation of greatest common divisors. Computational methods such as Chinese remaindering and Hensel lifting that are used to overcome the basic problem of intermediate expression swell are also covered. In addition to these deterministic methods the author discusses some heuristic and probabilistic techniques used in some computer algebra computations.

A point should be made that many of these algorithms are not what a typical mathematician would expect or has been taught. Indeed, this is one of the facts that makes this such an interesting field of study. As an example, consider the simple problem of computing the greatest common divisor of say two multivariate polynomials (a very common operation required by many algorithms in computer algebra). Every undergraduate mathematics student learns that the way to compute this would be to use Euclid's algorithm. It will come as a surprise to many that this is one of the last methods that one would want to use in this case. For example, this method has the problem that there is significant growth in the size of coefficients during the intermediate steps. Instead, one can use this text to learn the methods which are used in practice. The text includes in various chapters: polynomial remainder sequences; heuristic gcd computation; modular methods based on the use of Chinese remaindering; Hensel methods using p-adic arithmetic; and probabilistic methods. The last-named method includes the well-known "sparsemod" algorithm that came from the author's own Ph.D. thesis from the days when he was associated with the Macsyma project at MIT.

This book will be a valuable resource for anyone interested in pursuing research in the field of computer algebra. It is well written and easy to read, with good use made of examples to illustrate the mathematical difficulties that are encountered in polynomial computation. The book has numerous historical references throughout the text in addition to the interesting historical notes that are found at the end of each chapter. It does not cover the same variety of computer algebra topics as found in, for example, [1] but it does go into much more depth in the topics that it does discuss. My only complaint with the text is a minor one—I would like to have seen exercises at the end of each chapter. This would make it easier to use in an upper-level undergraduate course on computer algebra.

George Labahn

Department of Computer Science
University of Waterloo
Waterloo, Ontario N2L 3G1
Canada

1. K. O. Geddes, S. R. Czapor, and G. Labahn, *Algorithms for computer algebra*, Kluwer, Dordrecht, 1992.