MATHEMATICS OF COMPUTATION
Volume 67, Number 224, October 1998, Pages 1739–1751
S 0025-5718(98)01032-1

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.

16[65-02]—*Acta numerica* 1997, A. Iserles (Managing Editor), Cambridge University Press, New York, NY, 551 pp., $25\frac{1}{2}$ cm, hardcover, \$60.00

This book is Volume 6 (1997) of a series of survey articles on important developments in numerical mathematics and scientific computation written by authors who have made substantial contributions to the topics of their articles. The present volume contains eight articles ranging in length from 38 pages to 174 pages. A review of Volume 4 (1995) of this series by Vidar Thomée appeared in the April 1997 issue of Mathematics of Computation.

The first article, *Constructing cubature formulae*: *The science behind the art*, by Ronald Cools presents "a general, theoretical foundation for the construction of cubature formulae to approximate multivariate integrals." Quality criteria and several different ways to construct such approximations are discussed, as well as the characterization of minimal cubature formulae.

In *Wavelet and multiscale methods for operator equations*, Wolfgang Dahmen discusses the use of wavelets for the approximation of (mainly elliptic) partial differential equations. Many of the important tools of wavelet analysis can be found in this article along with more general mathematical ideas important to also understanding and analysing other multiscale approaches such as multigrid applied to finite element discretization of partial differential equations.

A new version of the fast multipole method for the Laplace equation in three dimensions by Leslie Greengard and Vladimir Rokhlin discusses a new version of the fast multipole method for the evaluation of potential fields in three dimensions. The scheme evaluates all pairwise interactions in large ensembles of particles, i.e., expressions of the form $\sum_{i=1,i\neq j}^{n}(q_i)/\|x_j-x_i\|$ for the gravitational or electrostatic potential and $\sum_{i=1,i\neq j}^{n}(q_i)(x_j-x_i)/\|x_j-x_i\|^3$ for the field, where $x_1,x_2,\dots,x_n$ are points in R^3 and $q_1,q_2,\dots,q_n$ are a set of real coefficients.

The article *Lanczos-type solvers for nonsymmetric linear systems of equations* by Martin J. Gutknecht introduces the basic forms of the Lanczos process and some of the related theory and describes in detail a number of solvers based on it. The author also discusses possible breakdowns of the algorithms and remedies for these breakdowns.

In the paper *Numerical solution of multivariate polynomial systems by homotopy continuation methods*, T. Y. Li discusses the use of homotopy methods to find all isolated solutions of a system of n polynomial equations in n unknowns and a variation of this problem in which it is desired to solve such systems for each of several choices of the coefficients in the system.

In *Numerical solution of highly oscillatory ordinary differential equations*, by Linda R. Petzold, Laurent O. Jay, and Jeng Yen, the authors consider several different classes of problems exhibiting oscillatory behavior, including linear oscillatory systems, rigid and flexible mechanical systems, problems in molecular dynamics, and problems from circuit analysis and orbital mechanics. For each class of problems, they discuss "the structure of the equations, the objectives of the numerical simulation, the computational challenges, and some numerical methods that may be appropriate."

In the article *Computational methods for semiclassical and quantum transport in semiconductor devices*, Christian Ringhofer gives an overview of recently developed numerical methods for several models used in semiconductor device simulation. These include Galerkin methods for the semiclassical and quantum kinetic systems and difference methods for the classical and quantum hydrodynamical systems.

The final article by Steve Smale entitled *Complexity theory and numerical analysis* deals with the complexity analysis of algorithms and involves upper bounds on the time required in a numerical algorithm and estimates of the probability distribution of the condition number of a problem. Among the problems discussed are the solution of polynomial equations, the solution of linear systems of equations, and the calculation of eigenvectors.

This and the other volumes of the series provide an excellent way to learn about current research in a wide variety of areas in numerical analysis and scientific computation.

Richard S. Falk

17[26A06, 34A50, 35A40, 65L60, 65M60, 65N30]—*Computational differential equations*, by K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, Cambridge University Press, New York, NY, 1996, xvi+538 pp., $22\frac{1}{2}$ cm, hardcover, $100.00

In the philosophy of Leibnitz, analysis and numerical computation are equally important aspects of mathematics and are not considered as separate elements. Unfortunately, this unified view has been neglected for the last hundred years. Mathematical education has treated these elements as distinct and has emphasized the symbolic, analytic aspect of mathematics. Now, however, with the advent of powerful computational tools it is possible to again fuse mathematical analysis and computation. Inspried by the philosophy to Leibnitz, the authors have written an ambitious text for undergraduate/graduate engineers and scientists which combines mathematical modelling, analysis and computation.

In order to plot a course through the vast landscape of material dealing with computation, the authors have chosen a particular method and used it to present a unified treatment of problems ranging from the fundamental theorem of calculus to partial differential equations. The Galerkin method uses linear combinations of special classes of functions to approximate the solutions of differential equations. The special classes of functions may be polynomials, piecewise polynomials, or trigonometric polynomials, to name a few. The Galerkin method can be viewed as a natural extension of the venerable technique of separation of variables, which uses expansions in term of eigenfunctions.

The tone for the book is set in the review chapter on Calculus. The fundamental theorem of the calculus is formulated as an initial value problem for a differential equation. The solution of this problem is then approximated by piecewise constant functions. The accuracy of the approximation is estimated and the estimate is used to suggest a variable mesh algorithm to compute the approximations, to within a given tolerance, as cheaply as possible. The student is immediately made aware of the practical nature of computation and the goal of doing it as efficiently as possible.

Along the way the student is given some basic results on polynomial interpolation and quadrature, and a review of linear algebra.

In Chapter 6 we see the Galerkin method introduced in two basic problems that will be studied in great detail in later chapters. Somewhat surprisingly one of them is the first-order linear equation

$$u'(t) + a(t)u(t) = f(t), \quad u(0) = u_0.$$

Rather than the usual numerical discussion using the Euler method and its refinements, we find polynomial approximations used to reduce the problem to finding solutions of linear systems. Approximation by global polynomials with the basis $1, t, t^2, \ldots, t^n$ leads to ill-conditioned systems of linear equations much as it does for least squares polynomial approximation. With this motivation, the authors turn to piecewise polynomial approximate solutions, limiting themselves to (discontinuous) piecewise constant and (continuous) piecewise linear functions.

In a second, more standard approach, the Galerkin finite element method is used to treat the boundary value problem

$$-u''(x) = f(x), \quad u(0) = u(1) = 0.$$

With the stage now set and the basic issues raised, the authors provide a chapter of numerical linear algebra. In Part II, they treat in detail two archetypal problems:

(1) Two point boundary value problems on an interval $[a, b]$ in Chapter 8, and
(2) Initial value problems for systems of first order equations in Chapter 9.

In both chapters the authors carefully discuss the a priori estimates for the Galerkin methods employed and also the a posteriori estimates needed to develop adaptive mesh algorithms. The discussion of error also includes comments about the error arising from the approximation of certain integrals by quadrature formulas. Estimates are made that take into account all the sources of error.

Part III deals with problems in more than one space variable, including the wave equation, the heat equation and the Poisson equation. These chapters provide rather brief derivations of the linearized equations in the usual way. Formulas for the solutions of the heat and wave equations are stated rather than derived. The emphasis in these later chapters is on the geometrical and concrete computational aspects of finite elements in two and three dimensions. The presentation reaches its most advanced level in Chapters 18 and 19, which discuss finite element approximate solutions to convection-diffusion equations in which convection dominates. After 20 chapters of rather concrete, computationally oriented discussion, the authors state and prove the Lax-Milgram lemma and illustrate its application with several examples. This mode of exposition is contrary to that of most mathematics texts and is quite effective.

The authors' writing style is clear and crisp. There is an abundance of quotations, many of them from Leibnitz, but also from sources ranging from Aristotle to Hank Williams. While interesting and amusing, not all of them seem particularly relevant.

Although the authors' stated goal is to fuse mathematical modeling, analysis and computation, the emphasis is clearly on computation. The authors' experience in computation gives them a perspective on the subject that allows them to stress the important issues. Rather than offer simply a collection of methods, they have chosen to unify the text by focusing on a single method—the Galerkin method. However, this comes at a cost because some other very effective techniques, particularly for ordinary differential equations, are not even mentioned.

Interesting problems are inserted throughout the text instead of being gathered together at the end of sections. They are often smaller steps in the proofs of larger propositions. In addition there are many computational projects which may be done with a collection of finite element codes developed at Chalmers University of Technology, Sweden. They are available at no charge at the Chalmers website. The web address that worked for me, different from the one given in the book, is `http://www.md.chalmers.se/Math/Research/Femlab/`.

The text would be suitable for a student having had the usual three semesters of calculus, linear algebra, a course in ordinary differential equations, and some experience with numerical analysis. The first two parts of the book could be used for an advanced undergraduate course, but Part III is graduate level material.

Because of its very definite point of view, this book does not fit the mold of the standard numerical analysis text or of the standard numerical analysis course. However, it is provocative and should be taken seriously by all faculty, not just those in applied mathematics. It takes a fresh look at some of the standard topics in undergraduate analysis, and the ideas of the text could be used in many courses in analysis and computation. While not without its limitations, this book provides a vision of computation and analysis that may become a model for the future.

Jeffery Cooper
Department of Mathematics
University of Maryland
College Park, Maryland 20742

18[65-01, 65FXX]—*Applied numerical linear algebra*, by James W. Demmel, SIAM, Philadelphia, PA, 1997, xi+419 pp., $25\frac{1}{2}$ cm, softcover, \$45.00

This book is intended as a textbook in numerical linear algebra for first-year graduate students in a variety of engineering and scientific disciplines. In the preface the author gives a list of goals he was trying to meet. After stating his target audience, he goes on to write, "2. It should be self-contained, assuming only a good undergraduate backgound in linear algebra. 3. The students should learn the mathematical basis of the field, as well as how to build or find good numerical software. 4. Students should acquire a practical knowledge for solving real problems efficiently. In particular they should know what the state-of-the-art techniques are in each area...". Finally he writes, "5. It should all fit in one semester...".

This is a difficult undertaking. In my opinion the author has fallen short in certain ways, but the book is excellent nevertheless. It is clearly written, though somewhat demanding, and it is well organized.

All of the standard topics are there, including linear equation solving, linear least squares problems, nonsymmetric and symmetric eigenvalue problems, and the singular value decomposition. The perturbation theory associated with each of these topics is discussed. The basic computational tools are developed, and the algorithms are derived. Convergence and backward error analyses are supplied where needed. Sometimes rigorous convergence proofs are provided, but in other cases informal arguments are given instead.

A serious effort has been made to discuss every state-of-the-art algorithm in the core areas of numerical linear algebra. Not surprisingly, emphasis has been placed on those that have been included in LAPACK, to which Demmel has been a major contributor. Numerous pointers to LAPACK (and to Matlab and Templates) are included. Wherever two or more algorithms for the same problem are given, advice on which is best in which situations is also given.

A long chapter on iterative methods includes not only the standard SOR and conjugate gradient theories, but also material on fast Poisson solvers, multigrid, and domain decomposition.

The final chapter is on iterative methods for eigenvalue problems. The focus is on the Lanczos algorithm for symmetric matrices, but there is also a brief discussion of the nonsymmetric problem.

Several other features are worth mentioning. Interesting and illuminating applications are sprinkled throughout the text. The impact of modern cache-based computer architectures on algorithm design is discussed. Relative perturbation theory and high-accuracy algorithms are included. The author maintains a homepage for the book (`http://www.siam.org/books/demmel/demmel_class`), at which one finds a list of errata, Matlab source code for examples and problems, and numerous links to software libraries and various information sources.

The book contains a huge amount of information within 400 pages. It follows that the presentation must be fairly steep. Indeed, many details are left to the reader, usually with scant hints. Most students will find this book to be tough sledding.

It is also fair to say that the book contains much more than could be covered in a single semester. In the preface the author gives advice on what to include in a one-semester course. The amount of material listed there is far more than I would undertake.

One apsect of the book's organization displeased me. It is difficult to find one's way from the problems back to the appropriate place in the text. For example, Question 6.7 on page 358 asks the student to prove Lemma 6.7. The way the numbering scheme is set up, Lemma 6.7 could be anywhere in Chapter 6, which is 96 pages long. I located Figure 6.7, Theorem 6.7, Example 6.7, and Definition 6.7 before finally finding Lemma 6.7, which turns out to be a list of fundamental properties of Chebyshev polynomials on page 296. Looking further, I found an Algorithm 6.7 and a displayed equation (6.7). There are several things that could have been done to make navigation easier. The questions could have been placed at the ends of sections rather than gathered at the end of each chapter. There could have been fewer independent numbering schemes. For example, there could have been a common numbering scheme for definitions, theorems, lemmas, propositions, and algorithms. Also, items could have been numbered by chapter.section, not just by chapter.

On the positive side, the book has a detailed table of contents and an extensive index, so it should be easy to use as a reference book.

I can recommend this book as a text for a class of outstanding, well-prepared students. It is also an excellent, up-to-date reference for experts.

David S. Watkins
Department of Pure and Applied Mathematics
Washington State University
Polman, Washington 99164-3113

19[65D25]— *Computational differentiation techniques, applicatitons, and tools*, Martin Berg, Christian Bischof, George Corliss, and Andreas Griewank (editors), SIAM, Philadelphia, PA, 1996, xv+421 pp., 25 cm, softcover, $65.00

This is one of the most interesting mathematical books that I have had in my hands for a long time. Its editors have strived to illuminate the ideas and the potential of computational differentiation by assembling a total of 36 contributions from nearly as many authors: each of them points a beam of light on the subject from a special direction, and the fascinating result is the appearance of a rather comprehensive image of this fast-growing, important area of scientific computing. Naturally, this also testifies to the skill of the organizers of the workshop in inviting the right people.

It is impossible to give credit to the individual contributions. Someone having a basic acquaintance with computational differentiation (this slightly wider notion has replaced the original term "automatic differentiation") may pick articles to his or her liking in any order, and will find many which offer interesting and relevant reading. Someone without prior knowledge of the subject should start with an introductory survey article (several of these papers have the appropriate flavor) and then will at least be able to gather an impression of the potential of computational differentiation for his or her own work. Gnerally, the articles maintain a nice balance between readability and technicality.

The volume should greatly help in advertising the important benefits that may be derived from computational differentiation in many areas of scientific computing, and it may help to attract a number of scientists into developping the subject further.

Hans J. Stetter

20[65K10, 90C26]—*Numerica: A modelling language for global optimization*, by Pascal Van Hentenryck, Laurent Michel, and Yves Deville, The MIT Press, Cambridge, MA, 1997, xvii+210 pp., 23 cm, softcover, $25.00

The overall subject area encompassing this book is the numerical solution of nonlinear systems of equations and constrained and unconstrained optimization. More precisely, the book describes certain techniques for finding *all* solutions to nonlinear systems of equations and to finding *global* optima. Until recently unrecognised by many researchers in the field, such computational methods both provide mathematical rigor and are applicable to many practical problems. The authors have a commercial implementation, ILOG Numerica, that embodies both their own variants of these methods and a modeling language to interface well with the methods and to be user friendly. The book introduces the underlying methods and the modeling language, presents computational results obtained with Numerica, and guides the reader in use of Numerica. Some details are given below.

1. Overall subject area

Part of the basis of the computational procedures in Numerica is *interval arithmetic*, in which operations are performed on intervals rather than numbers. The crucial fact about implementations of interval arithmetic with floating point arithmetic is *rigorous enclosure of ranges.* That is, if $\boldsymbol{x} = [\underline{x}, \overline{x}]$ is an interval (or, more generally, an interval vector), and $\boldsymbol{f}(\boldsymbol{x})$ is an interval value obtained by evaluating an expression f with interval arithmetic, then

$$\{f(x) \mid x \in \boldsymbol{x}\} \subseteq \boldsymbol{f}(\boldsymbol{x}),$$

i.e., the interval value contains the range. Since, with proper rounding control, this computational result contains the range with *mathematical rigor*, interval arithmetic is used in various automatic theorem proving contexts. Interval arithmetic is thus particularly elegant and powerful in handling systems of inequality constraints and in global optimization, where bounds on the range of an objective function are valuable.

For a simple illustrative example, suppose it is to be determined whether the constraint $x_1^2 - x_2 \geq 0$ is valid in the rectangle, i.e., in the interval vector $([1,2],[3,4])^T$. An interval evaluation of the left side of the constraint yields

$$[1,2]^2 - [3,4] = [1,4] - [9,16] = [-15,-5] < 0.$$

Thus, every point in $([1,2],[3,4])^T$ is infeasible with respect to the constraint $x_1^2 - x_2 \geq 0$.

Until recently, knowledge of interval arithmetic was either limited or eschewed. Many numerical analysts had not recognised that it is possible to compute rigorous bounds on ranges. Among those who had heard of interval arithmetic, a common objection was that the "bounds on the range are so pessimistic that they are useless." This is true for naive, inappropriate use of interval arithmetic, and overly optimistic early claims about interval arithmetic led to disbelief. However, research in the past twenty years has led to clearer views of the possibilities, more practical algorithms, and solution of significant application problems with interval methods. Numerica is a manifestation of such research.

The elements of interval arithmetic necessary for understanding the other subject matter are introduced clearly in *Numerica: A modeling language for global optimization.*

On the application level, the book *Numerica* addresses two main problems of nonlinear programming:

- The numerical solution of inequality-constrained nonlinear systems of equations,

$$\boxed{\begin{array}{lrcl} \text{Find } \textit{all} \text{ solutions to} & f(x) & = & 0 \\ \text{subject to} & g(x) & \leq & 0, \end{array}} \tag{1}$$

  where $f : \boldsymbol{x} \subset \mathbf{R}^n \to \mathbf{R}^n$ and $g : \boldsymbol{x} \to \mathbf{R}^{m_2}$, $m_2 \geq 0$, where $\boldsymbol{x}$ is an interval vector

$$\boldsymbol{x} = ([\underline{x}_1, \overline{x}_1], \ldots, [\underline{x}_n, \overline{x}_n])^T;$$

and

- Constrained nonlinear optimization:

$$
\text{(2)} \qquad \boxed{\begin{array}{ll} \text{minimize} & \phi(x) \\ \text{subject to} & \left\{ \begin{array}{lcll} c(x) & = & 0 & \text{and} \\ g(x) & \leq & 0 & \end{array} \right\}, \end{array}}
$$

where $\phi : \boldsymbol{x} \subset \mathbf{R}^n \to \mathbf{R}$, $c : \boldsymbol{x} \to \mathbf{R}^{m_1}$, and $g : \boldsymbol{x} \to \mathbf{R}^{m^2}$, where $\boldsymbol{x}$ is an interval vector, and $m_1 \geq 0$ and $m_2 \geq 0$. Here, a *global* optimum, that is, the lowest possible value of ϕ over the feasible set, is sought.

In problem 2, it is possible to consider all or some of the interval bounds $\underline{x}_i$ and $\overline{x}_i$ as constraints. In *Numerica*, the concept "min-stable", basically the assumption that the minimum of ϕ occurs in the interior of the compact search region $\boldsymbol{x}$, is introduced. If this assumption is violated, some or all of the bounds $x_i \geq \underline{x}_i$ and $x_i \leq \overline{x}_i$ can be introduced as explicit inequality constraints.

2. Main research contributions of the authors

The book *Numerica* is not a research monograph, but it does contain polished results of original work by the authors. These include

- a unique implementation of constraint solving, and
- a modeling language to fit both the users' thought processes and the solution process.

Constraint solving or constraint propagation is a fairly well-known technique in artificial intelligence in the form of *constraint logic programming*. As a simplified example of the underlying idea for interval constraint propagation, suppose we are interested in the portion of the box $\boldsymbol{x} = ([-1,1],[0,2])^T$ that is feasible with respect to the constraint system

$$
\begin{array}{lclcl} c(x) & = & x_1^2 + x_2^2 - 1 & = & 0, \\ g(x) & = & x_1 + x_2 & \leq & 0. \end{array}
$$

Then we can solve for one variable in terms of the others in each of the constraints, to obtain sharper bounds. For example, we can solve for x_2 in $c(x) = 0$ to obtain

$$
\begin{array}{lcl} x_2 & \in & -\sqrt{1 - \boldsymbol{x}_1^2} \cup \sqrt{1 - \boldsymbol{x}_1^2} = [-1,1], \quad \text{whence} \\ x_2 & \in & [-1,1] \cap [0,2] = [0,1]. \end{array}
$$

We may then solve for x_1 in $g(x) \leq 0$ to obtain

$$
\begin{array}{lcl} x_1 & \leq & -x_2 \in [-1,0], \quad \text{whence} \\ x_1 & \in & [-1,0]. \end{array}
$$

We have thus reduced the initial set $([-1,1],[0,2])^T$ to the smaller set $([-1,0],[0,1])^T$. In some cases, it is possible for this technique alone to reduce an initial region $\boldsymbol{x}$ to a point, or to prove that no feasible point exists in $\boldsymbol{x}$.

This basic idea can be used in various ways. For example, each variable can be solved in each equation, as above. Alternately, as in [1] or [5, Ch. 7], the constraints can be parsed, and the relationships among the intermediate results produced during evaluation of the expressions defining the constraints can be used. In *Numerica*, the authors have developed a special univariate bisection method to work with the original constraints. This method appears to be effective on many problems.

Some experts have implemented interval constraints in the logic programming (artificial intelligence) language PROLOG; see [2], [6], or [7]. A disadvantage of PROLOG implementations of algorithms for problems 1 or 2 is that execution can be extremely slow. One idea behind the authors' modeling language Numerica is increased efficiency.

The book *Numerica* itself needs to be read to grasp the full scope of the authors' new language. However, as an example, take Problem 4.3 from [3]:

$$\begin{array}{ll} \text{minimize} & x_1^{0.6} + (3(x_1+u_1))^{0.6} - 6x_1 - 4u_1 + 3u_2 \\ \text{subject to} & \left\{ \begin{array}{rcll} x_1 + 2u_1 & \le & 4 & \text{and} \\ 3x_1 + 3u_1 + 2u_2 & \le & 4 & \end{array} \right\}. \end{array}$$

A corresponding Numerica file is:

```
Variable:
    x1 in [1e-2..3];
    u1 in [0..2];
    u2 in [0..1];
Body:
  minimize
    x1^0.6 + (3*(x1+u1))^0.6 - 6*x1 - 4*u1 + 3*u2
  subject to
    x1 + 2*u1 <= 4;
    3*x1 + 3*u1 + 2*u2 <= 4;
Display:
  dynamic;
```

Others, as in [4] or [1], have developed special languages or systems for constraint techniques. However, [4] is a general system without specific regard to global optimization, while the algorithms underlying [1] are different.

3. Contents of the book

Chapter 1 contains an explanation of nonlinear programming and constrained global optimization problems, along with a guide to the rest of the book. This guide includes recommendations on which chapters are important for different classes of readers, such as engineers applying the package, researchers in the underlying methods, etc. In Chapter 2, a clear, abbreviated explanation is given of the main features of the Numerica modeling language. Chapter 3 deals with (i) the elements of interval analysis, (ii) a clear theory of what a solution is (to aid in the interpretation of results given by Numerica), and (iii) an explanation of constraint solving. In Chapter 4, examples are given of problems for which Numerica fails, such as problems with an infinite number of solutions. Chapter 4 then goes on to indicate how problems can be reposed to improve Numerica's performance (this includes restating the objective and constraints in various ways; considerations are based mainly on properties of interval arithmetic). Chapter 5, good for reference purposes, gives a complete description of Numerica's syntax. Chapter 6, entitled "The Semantics of Numerica", includes formal definitions of interval extensions of functions, as well as formal definitions of the relationship between the results Numerica gives and the exact solution to the original problem. The purpose of the formality in Chapter 6 is to aid intuitive understanding. Chapter 7 describes some details of the underlying algorithms used in the implementation of Numerica. Chapter 8 presents a sizable set of test problems, as well as numerous tables of performance results of running Numerica on these test problems. Two appendices give (i) a concise, diagramatic depiction of Numerica's syntax, and (ii) Numerica input files for the test problems.

4. Overall assessment and recommendations

The book, overall, is clearly written, and should be accessible to nonexperts. It can serve both as a user's guide to the Numerica package and as an introduction to various techniques of interval global optimization. Additionally, the implementation details contain some ideas of possible interest to researchers in the field. The test results give an indication of the practicality and scope of applicability of the Numerica package; additionally, the completely specified test problem set can serve for benchmarking.

In summary, the book is *not* a comprehensive overview or a comprehensive reference to research in interval techniques for global optimization. The book *is* both a good introduction to interval global optimization and constraint solving and a good user's guide to the Numerica package. Although not comprehensive, the book *does* contain items of interest to researchers in the techniques for global optimization and nonlinear systems.

References

[1] A. B. Babichev, O. B. Kadyrova, T. P. Kashevarova, A. S. Leshchenko, and A. L. Semenov, UniCalc, A novel approach to solving systems of algebraic equations, *Interval Computations*, **1993**, no. 2, 29–47. CMP 94:15

[2] F. Benhamou, Interval constraint logic programming. *Constraint Programming: Basics and Trends* (A. Podelski, editor), 1–21, Springer-Verlag, Berlin, 1995.

[3] C. A. Floudas and P. M. Pardalos, *A Collection of Test Problems for Constrained Global Optimization Algorithms*, Lecture Notes in Computer Science, no. 455, Springer-Verlag, New York, 1990. MR **91m:**90154

[4] E. Hyvönen and S. De Pascale, Interval computations on the spreadsheet, *Applications of Interval Computations* (R. B. Kearfott and V. Kreinovich, editors), Applied Optimization, 169–209, Kluwer, Dordrecht, Netherlands, 1996. MR **96m:**65052

[5] R. B. Kearfott, *Rigorous global search*: *Continuous problems*, Kluwer, Dordrecht, Netherlands, 1996. MR **97i:**90003

[6] H. Koch, A. Schenkel, and P. Wittwer, *Computer-assisted proofs in analysis and programming logic*: *A case study*, SIAM Rev. **38** (1996), no. 4, 565–604. MR **97k:**39001

[7] P. Van Hentenryck, *Constraint satisfaction in logic programming.* MIT Press, Cambridge, MA, 1989. MR **91f:**68015

R. Baker Kearfott
Department of Mathematics
University of Southwestern Louisiana
Lafayette, LA 70504

21[68Q40]—*Polynomial algorithms in computer algebra*, by F. Winkler, Springer-Verlag, Wien, New York, 1996, viii+270 pp., 24 cm, softcover, $69.00

Any author in the area of computer algebra is faced with dilemmas forced upon him by the subject matter. Opinions vary on what exactly is computer algebra, but also the angle from which to present the material, and with it the target audience, must be carefully considered.

In *Polynomial algorithms in computer algebra*, Franz Winkler laudably attempts to demarcate clearly what he considers to be "computer algebra", and what part of it he intends to cover. For him computer algebra is synonymous with symbolic algebraic computation, which is concerned with the design, analysis, implementation and application of algebraic algorithms, that is, algorithms dealing with exact (non-numeric) computation in algebraic domains, striving to algebraic solutions. Winkler's interests are in the subarea dealing roughly with "polynomials"—the title refers to the subject, not to the complexity of the algorithms!

Of importance for the potential reader is the point of view chosen by the author to present his subject matter. Computer algebra has been claimed to be both a branch of mathematics and of computer science. Inclusion of Winkler's text in Springer's new series of *Texts and Monographs in Symbolic Computation* does not in itself reveal a view either way, as the publisher seems to aim at practitioners of both disciplines (yet the silver rather than yellow colour of the covers may suggest otherwise). But although definitions for most algebraic notions are briefly recalled, it is clearly expected that the reader is familiar with the elementary theory of (finite/p-adic/number) fields, (finite/polynomial/power series) rings, and, to a lesser extent, groups. Despite its mathematical orientation, this monograph is *not* a systematic account of basic algebra by way of algorithms, but rather an introduction into some of the available (polynomial algebraic) algorithms for accomplishing certain tasks. An analysis of the computational complexity of most algorithms is included, but these and other remarks of an information theoretic nature (such as about computer representation of algebraic objects) are provided at a pleasantly restrained rate (to my taste).

The author also deserves praise for the way he has dealt with the problems of presenting algorithms and of referring to computer algebra *systems*. By presenting most algorithms in a clear and concise combination of plain English and mathematical notation, he is able to avoid the use of lower level programming languages (C, Pascal) and of higher level system languages (like that of MAPLE) and their idiosyncrasies. The text provides no pointers to existing implementations, and remains system independent this way. Yet, users of the major systems will have no difficulty in locating the relevant functions.

Winkler's approach can be outlined as follows: motivated by a few "practical" applications, he develops a toolkit of algebraic techniques and algorithms, and shows in the final chapters how the motivating applications can be rephrased as algebraic problems allowing treatment by the tools he has provided. After an introduction, which includes a discussion of the notion of computer algebra and some other preliminaries, a chapter on arithmetic in basic algebraic domains follows. Just like any developer of a new computer algebra system seems to have to struggle with yet again reimplementing algorithms for basic but crucial operations like long integer arithmetic, and (multivariate) polynomial arithmetic (in sparse and recursive form), Winkler is unable to avoid their fairly lengthy description here. As a consequence we have reached page 50 (out of 250) before getting to the six main chapters on techniques in polynomial algebra, and the three chapters on their application. The inconsistent and confusing notation for residue class rings, finite fields and p-adic rings in the first chapters perhaps only deserves mentioning because of the contrast with otherwise high standards of exposition and consistency, and the reasonably adequate TEX typography and layout.

The six main chapters are concerned with "computing with homomorphic images" (Chinese remaindering, p-adic lifting, discrete Fourier transform), common divisors of polynomials, factorization of polynomials (over finite fields, integers, and number fields), decomposition of polynomials, solving systems of linear equations, and Gröbner bases. Each of these provides a good introduction into the area with many clear examples and exercises. Usually the author cuts a more or less straight path through the jungle of results, and although he does point out certain byways (particularly in the very good bibliographic notes), it is not always clear what their status is. For example, Berlekamp's polynomial factorization algorithm is nicely covered, and the chapter notes mention the existence of alternatives (like Cantor-Zassenhaus), but their relative merits are not discussed at all.

The three motivating applications are quantifier elimination in real closed fields, requiring a decision procedure for systems of polynomial (in)equalities (used in robotics, for example); indefinite summation, in particular of hypergeometric functions; and parametrization of algebraic curves. Gosper's algorithm for solving the second problem and the sketch of Collins's cylindrical algebraic decomposition algorithm for the first both suffer didactically from requiring very little of the background material. The discussion of curve parametrization builds much more nicely on the previous chapters.

In summary, this book provides a good, accessible, and up-to-date account of a particular branch of algorithmic algebra that may perhaps be described as the type practised at RISC, Linz. To me it does not seem suitable as an introductory text on polynomial algebra by way of algorithms, but it will probably prove to be of value to users of computer algebra systems in getting a feeling for the background, capabilities, and difficulties that such systems have in dealing with problems involving polynomial arithmetic.

Wieb Bosma
Vakgroep Wiskunde
Universiteit van Nijmegen
Postbus 9010
6500 GL Nijmegen
The Netherlands
E-mail address: wieb@sci.kun.nl

22[00A69]—*Computer algebra in industry: Problem solving in practice*, Arjeh M. Cohen (Editor), Wiley, Chichester, 1993, x+252 pp., $23\frac{1}{2}$ cm, \$45.00

Computer algebra systems (e.g., Mathematica, MAPLE, etc.) are now familiar to the academic community. Courses are taught on their use and textbooks are being written using them, which in turn require the reader to use them to solve problems. An example is the differential geometry text by Gray [1]. The publicity for these systems uses phrases such as "environment for technical computing", and questions about how such systems have changed the practice of mathematics are being examined, for example, by Devlin [2]. There has not been much yet on applications of such systems in industry.

The book under review here is the result of the SCAFI Seminar held in Amsterdam in 1992 (I could not find the definition of the acronym in the text, but it must be something like Seminar on Computer Algebra For Industry). According to the Preface, the seminar theme was "to show how computer algebra in industry can be useful and cost effective". To that end, eleven applications from industry are described: three originate in the movement of fluids, three from the dynamics of mechanical systems, three from electronics, and two from other areas. The computer algebra systems used in the treatment of these applications include MAPLE, Mathematica, AXIOM, and REDUCE. The descriptions are fairly uniform in format. Each has a summary of the problem to be solved and how it was rendered into a mathematical model. Then there is a description of how computer algebra was applied to reduction of the model to a solvable form followed by results. Sometimes the computer algebra programs are supplied. The book begins with a chapter containing an instructive overview of what computer algebra is and what its problem solving scope is. The book closes with two chapters on interfacing computer algebra systems to systems more suitable to numerical computation.

I feel the most important feature of this book is that it deals primarily with the mathematical features of the applications treated, not the numerical. The reader is shown examples of how powerful tools can be brought to bear in dealing with aspects of the solutions that would be very time consuming if these tools were not available. As such, the book indeed demonstrates that computer algebra is changing the practice of mathematics, and it makes good reading for all mathematical practitioners, especially those in industry. I recommend it also to teachers who include computer algebra in their courses for use as a reference and sourcebook.

Finally, I want to especially recommend this book to those who have not used a computer algebra system yet. I am writing this review on my workstation which, as I write, is busy using the computer algebra system I have installed to do a numerical integration for me in the background. Before the numerical work, it did symbolic integration to reduce the integral to a more practical form. I cannot imagine working any other way now. Perhaps a reading of this book will make a convert out of you, too.

References

[1] Gray, Alfred, *Modern differential geometry of curves and surfaces with Mathematica*, Second Edition, CRC Press, 1997.

[2] Devlin, Keith, *The logical structure of computer-aided mathematical reasoning*, Amer. Math. Monthly, Vol. 104, No. 7, pp. 632–646, 1997. CMP 97:17

Bernard D. Rudin
IBM Austin Lab 4310
11400 Burnet Rd.
Austin, TX 78758-3406