3. The Illumination Problem;
4. Orbits in the Planar Three-Body Problem;
5. The Internal Field in Semiconductors;
6. Some Least Squares Problems;
7. The Generalized Billiard Problem;
8. Mirror Curves;
9. Smoothing Filters;
10. The Radar Problem;
11. Conformal Mapping of a Circle;
12. The Spinning Top;
13. The Calibration Problem;
14. Heat Flow Problems;
15. The Penetration of a Long Rod into a Semi-infinite Target;
16. Heat Capacity of a System of Bose Particles;
17. Compression of a Metal Disc;
18. Gauss Quadrature; and
19. Symbolic Computation of Explicit Runge-Kutta Formulas.

The presentation is unique, and extremely interesting. I was thrilled to read this text, and to learn the powerful problem-solving skills presented by these authors. I recommend the text highly, as a learning experience, not only to engineering students, but also to anyone interested in computation.

F. S.

18[68-01, 68Q40]—*Maple* V *by example*, by Martha L. Abell and James P. Braselton, AP Professional, Boston, MA, 1994, xii+500 pp., 23 cm, softcover, $39.95

This book appears to be aimed at first-year undergraduate students who are not specializing in either mathematics or computer science but are using mathematics and computers purely as tools, and are trying to use Maple for the first time. It presents computational models for a range of standard elementary mathematical tasks to which Maple can be applied. It is analogous to a book of recipes rather than a book about cookery, and whilst it presents the mathematical background to some topics, it discusses hardly any of the programming background.

After a general introduction, the book has chapters discussing the following topics: basic arithmetic and algebra; calculus; sets, lists and tables; matrices and vectors; more on linear algebra; a long and rather laboured chapter on differential equations; and finally a chapter showing the obligatory plots of weird shapes in three dimensions. The book has a detailed index—so detailed that it indexes two occurrences of ellipsis, neither of which refers to its Maple significance. At the back of the book is a gratuitous tear-out "Quick Reference" card that reminds the reader that, among other things, "+" is a "frequently used abbreviation" for addition, and briefly illustrates 23 elementary Maple commands in the general areas of calculus, linear algebra and graphics.

Students, especially the less academic, seem to like copious explicit examples, and that requirement is certainly met by this book. However, students tend to believe what is printed in textbooks in preference to what their lecturers write or say in class. Hence, authors of student textbooks have a considerable responsibility to get it right. Unfortunately, the authors of this book have not got it right, because it contains a prodigious number of errors, misguided examples and confusing assertions.

For example, the following all occur on page 20 alone. The Maple trigonometric function `arccos` is misspelled as `arcos`. It is asserted that "`e ^ (−5)` could have been entered instead of exp(−5)", whereas in fact `e` has no special significance in Maple and it is `E` that represents the exponential number (the base of natural logarithms)—Maple is case-sensitive. The authors write "Notice that Maple returns an exact value unless otherwise specified with `evalf` or `evalc`." In fact, the value returned by `evalc` is as exact as its argument, and it is only `evalf` that forces approximate numerical evaluation of exact expressions. Such errors would be less disastrous in a monograph aimed at experienced Maple users, but unfortunately there are no pearls of wisdom for such readers in this book.

The book describes Maple V Release 2, which is the release before the current one. However, it is almost impossible to describe the latest version of software that is developing as rapidly as computer algebra systems unless one is a member of the inner circle of developers. A less excusable aspect of this book is that all descriptions of the user interface are specific to the Macintosh implementation. Hence, the illustrations of the graphical user interface and the discussion of details of its operation will be either inappropriate or wrong for users of Maple on other systems. For example, the comments about the distinction between the *Return* and *Enter* keys, and some comments about the Help system, are wrong in the context of Microsoft Windows.

There are a number of subtleties to Maple that seem to cause users fairly widespread confusion. Some are conceptual difficulties common to many computer algebra systems. One of these is the need to use free (i.e., unassigned) variables in some contexts, which the book confuses totally. It is *never* necessary to clear or unassign a variable immediately before assigning to it, but this is done (inconsistently) in perhaps half the examples in this book.

I will discuss a few of the more pervasive of the many other errors, inconsistencies and sources of confusion in the book.

Maple supports functions or mappings as objects independent of any arguments. This is an elegant and useful feature, which is not provided by all computer algebra systems. However, it is important not to confuse *functions* with their values, which are *expressions* obtained by applying the functions to arguments. This book does nothing to reduce this confusion. For example, on page 42, the authors write "The following example illustrates how to graph several functions using operator notation". In fact, nothing at all that could be regarded either as a Maple function or as Maple operator notation appears in the example. Three variables are (unnecessarily) unassigned and then assigned *expressions* containing the free variable x (which could usefully have been unassigned to ensure that it is free, but has not been). These three expressions are then plotted together using the normal Maple syntax for plotting expressions (which requires the free variable to be specified). This example illustrates a lack of understanding of the significance of free variables and the distinctions among functions, expressions and operators.

Maple supports sequences, lists and sets as distinct (but closely related) primitive data types, and it supports vectors as special cases of arrays. At various points the book confuses these data types. On page 33, within the space of four lines, the book uses both notations (x, y) and $\langle x, y\rangle$ to represent vectors in the context of writing *vector*-valued functions, but in fact the examples define functions that return *lists*.

Whilst it is true that many Maple functions accept lists and vectors interchangeably, they are fundamentally different data types. For example, it is possible to assign to the elements of a Maple vector but not to the elements of a Maple list. Similarly, on page 225 the book asserts that "A matrix is simply a list of lists...", which is not true, although the two structures have a natural one-to-one correspondence that Maple uses in some contexts.

On page 177 the authors assert that "the command $\texttt{seq(f(i), i = 0..n)}$ creates the list $\texttt{[f(1), f(2), ..., f(n)]}$", which is not true: the list must be explicitly constructed from the sequence by using square brackets. On page 186 the authors present an example that completely confuses sets and sequences. The output shown would not be produced by the input. To produce the output shown would require the addition of explicit set braces around the sequences, in which case it would be preferable to use the conventional infix syntax "`setone union settwo`" rather than the more obscure prefix syntax "`` `union`(setone, settwo) ``". On page 334, the authors create a *set* of expressions (which the authors call functions) and then map a function over the set to create a new set. It is meaningless to use sets in this context: they should be lists, because the ordering of the elements is *essential* in order to retain the correspondence between sources and images under the map. On page 347, the authors construct a list of expressions (but the output contains a spurious comma, thereby creating nonsense), which they then call a set. There is a lamentable lack of precision in the authors' use of sequences, lists and sets.

On page 35, the authors assert that "... in general, $'' '' '' \ldots ''$ (k times) refers to the kth most recent output." This is true only for $1 \leq k \leq 3$, which is not made clear. There are subtle dangers attached to the use of the $''$ (ditto) operator in a worksheet environment because ditto refers to the value computed most recently in time rather than in space. Anyone who teaches Maple must have observed that this confuses inexperienced users, but I found no mention of this danger in the book.

When illustrating the process of taking a limit, the authors evaluate an expression on a set of points in the neighbourhood of the limit point. They take great trouble to randomize this set of points, but I see no reason to believe that randomness plays any role in taking limits, and its role is not explained. I found this discussion particularly misleading.

On page 260 the book advocates the use of the cut-and-paste facilities (I assume) provided by the (Macintosh) graphical user interface as a way of outputting data to a file for later re-input to Maple. This is generally a tortuous process because it is necessary both to ensure that linear character-based (`lprint`) output is used *and* to proceed via another application such as an editor. Maple provides several *much better* and more elegant mechanisms. The `save` command can be used to save data in either ASCII or binary format, and `writeto`, `appendto` and `printf` can be used to output arbitrary data to a file. Several analogous functions are provided to read data back into Maple. These functions all existed in Release 2, but I could not find any of them mentioned in the book—they certainly do not appear in the index.

On page 300 the authors illustrate the use of rotation matrices to rotate a square. Apart from two syntax errors, they omit to specify the plot option "`scaling = CONSTRAINED`", with the consequence that their square is plotted as a rectangle which is then distorted into a parallelogram as it is rotated. This is much more likely to obscure the underlying geometry than to illuminate it!

This book represents a good idea that is unfortunately marred by what I assume was haste and carelessness, and perhaps some lack of understanding of the more subtle aspects of Maple. The publishers must surely also take some blame for not ensuring adequate proofreading.

FRANCIS J. WRIGHT
SCHOOL OF MATHEMATICAL SCIENCES
QUEEN MARY AND WESTFIELD COLLEGE
UNIVERSITY OF LONDON
LONDON E1 4NS, ENGLAND

19[49-02, 49K40, 65K05]—*Nonlinear programming*, by Olvi L. Mangasarian, Classics in Applied Mathematics, Vol. 10, SIAM, Philadelphia, PA, 1994, xvi+220 pp., 23 cm, softcover, $28.50

This influential book on the theory of nonlinear optimization was first published in 1969, and went out of print a few years ago. SIAM has now republished it as part of a series "Classics in Applied Mathematics".

This beautiful book contains a lucid exposition of the mathematical foundations of optimization. The presentation is always simple and concise, and contains complete and rigorous proofs of most of the results. This is a formative book, which can be read by a student with a good background of calculus of several variables, in that it exposes the reader to most of the basic mathematical ideas of nonlinear optimization. The book is self-contained: the excellent appendices and the introduction review all the mathematical concepts needed to understand the material of the book.

There is a very good treatment of convexity and its generalizations—quasi-convexity and pseudoconvexity. Optimality and duality are covered in great generality, and the exposition of theorems of the alternative is a pleasure to read. There is also an extensive discussion of constraint qualification. All of these results make the book a valuable reference.

Much work has been done in convexity and duality in the 25 years since this book was published, and our views of the foundations of mathematical programming have changed. Nevertheless, the theory developed in this book remains central to the foundations of nonlinear optimization, and the style remains as effective and delightful today as when the book was first published. Every person interested in nonlinear optimization should own this book.

JORGE NOCEDAL
DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
NORTHWESTERN UNIVERSITY
EVANSTON, IL 60208-0001

20[49-02, 49M35, 49M45, 65K05]—*Interior-point polynomial algorithms in convex programming*, by Yurii Nesterov and Arkadii Nemirovskii, SIAM Studies in Applied Mathematics, Vol. 13, SIAM, Philadelphia, PA, 1994, x+405 pp., 26 cm, $68.50[1]

The appearance of Karmarkar's method ten years ago opened a new chapter in the study of complexity in mathematical programming, which has since resulted

[1] This is an abridged version of a review that appeared in OPTIMA, a newsletter of the Mathematical Programming Society.