Review of *An Introduction to Parallel and Vector Scientific Computing*
by Ronald W. Shonkwiler and Lew Lefton, Cambridge University Press, 2006
Review by David H. Bailey[1]

On one hand, the field of high-performance scientific computing is thriving beyond measure. Performance of leading-edge systems on scientific calculations, as measured say by the Top500 list [2], has increased by an astounding factor of 8000 during the 15-year period from 1993 to 2008, which is slightly faster even than Moore's Law. Even more importantly, remarkable advances in numerical algorithms, numerical libraries and parallel programming environments have led to improvements in the scope of what can be computed that are entirely on a par with the advances in computing hardware. And these successes have spread far beyond the confines of large government-operated laboratories — many universities, modest-sized research institutes and private firms now operate clusters that differ only in scale from the behemoth systems at the large-scale facilities.

In the wake of these recent successes, researchers from fields that heretofore have not been part of the scientific computing world have been drawn into the arena. For example, at the recent SC07 conference, the exhibit hall, which long has hosted displays from leading computer systems vendors and government laboratories, featured some 70 exhibitors who had not previously participated.

In spite of all these exciting developments, and in spite of the clear need to present these concepts to a much broader technical audience, there is a perplexing dearth of training material and textbooks in the field, particularly at the introductory level. Only a handful of universities offer coursework in the specific area of highly parallel scientific computing, and instructors of such courses typically rely on custom-assembled material. For example, the present reviewer and Robert F. Lucas relied on materials assembled in a somewhat ad-hoc fashion from colleagues and personal resources when presenting a course on parallel scientific computing at the University of California, Berkeley, a few years ago [1].

Thus it is indeed refreshing to see the publication of the book *An Introduction to Parallel and Vector Scientific Computing*, written by Ronald W. Shonkwiler and Lew Lefton, both of the Georgia Institute of Technology. They have taken the bull by the horns and produced a book that appears to be entirely satisfactory as an introductory textbook for use in such a course. It is also of interest to the much broader community of researchers who are already in the field, laboring day by day to improve the power and performance of their numerical simulations.

The book is organized into 11 chapters, plus an appendix. The first three chapters describe the basics of system architecture including vector, parallel and distributed memory systems, the details of task dependence and synchronization, and the various programming models currently in use — threads, MPI and OpenMP. Chapters four through nine

[1]Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA, dhbailey@lbl.gov. Supported in part by the Director, Office of Computational and Technology Research, Division of Mathematical, Information and Computational Sciences, U.S. Department of Energy, under contract number DE-AC02-05CH11231.

provide a competent introduction to floating-point arithmetic, numerical error and numerical linear algebra. Some of the topics presented include Gaussian elimination, LU decomposition, tridiagonal systems, Givens rotations, QR decompositions, Gauss-Seidel iterations and Householder transformations. Chapters 10 and 11 introduce Monte Carlo methods and schemes for discrete optimization such as genetic algorithms.

One highly commendable aspect of this book is that it presents, in detail and without apology, actual code segments illustrating many of the algorithms and techniques described in the text. These are mostly in Fortran and C, but there are also a few in languages such as Java. These code examples are interspersed in the main exposition, with longer examples in the Appendix. These examples include, in many cases, parallel programming constructs, say using MPI or OpenMP, and note potential pitfalls in real-world implementations.

There are a few drawbacks that this reviewer could list: (1) there are a noticeable number of editing glitches — spelling errors, run-on sentences and the like — even though in general the text is very readable; (2) some notation is not the standard usage, such as using "MFLOPS" instead of the preferred "Mflop/s"; (3) some of the exposition is disappointingly out-of-date, such as descriptions of the Cray X-MP and Y-MP vector computers, which haven't been in widespread usage for at least ten years; (4) the book includes a section on quantum computing, which is interesting and well-written, but it not connected in any credible way to the remainder of the book; and (5) the book omits a number of topics readers might expect to be included in a text of this scope.

Some of the omissions noted by this reviewer include a discussion of fast Fourier transforms (FFTs), a section on n-body problems and the general class of dynamically adaptive computations, and at least a brief explanation of how a physical system described by a set of differential equations can be discretized and converted into a sequence of linear algebra computations. Some additional discussion on debugging and performance analysis would also have been helpful. Thus instructors and others who wish to cover such topics will have to supplement the text with some additional material as needed.

But in general it is a well-written and useful reference. It will be interesting to see if it becomes widely adopted as a text for coursework in the field.

References

[1] David H. Bailey and Robert F. Lucas, "Applications of Parallel Computers," available at `http://crd.lbl.gov/~dhbailey/cs267`.

[2] "The Top 500 Supercomputer Sites," available at `http://www.top500.org`.