

Book Review

Numerical Recipes in C++: The Art of Scientific Computing

2nd ed.; W. Press, S. Teukolsky, W. Vetterling, and B. Flannery;
Pub Date: February 2002; ISBN: 0-521-75033-4;
1064 pp.; \$70.00 (hardcover)

Numerical Recipes in C++: The Art of Scientific Computing is the recently updated, C++ version of the well-known reference *Numerical Recipes in C*. Written by four professionals from academia and industry, it provides overviews to a wide variety and seemingly exhaustive list of numerical algorithms. It expands brief descriptions of the practical applications of each topic into more advanced theoretical discussions and then provides ANSI/ISO C++ implementations of each algorithm. It also concludes each section with a helpful list of further reading on each topic. This text features numerical procedures for linear algebra, function operations, statistical tests, Fourier methods, differential equations, random number generation, and many others, as well as more computer science oriented subjects including sorting, checksums, and data compression.

As a result of the extensive list of topics covered by *Numerical Recipes in C++*, fully understanding the theory in each section will most likely require prior exposure to the presented topic. In some cases, a mathematical educational background may be required while in others, such as statistical testing and sorting, familiarity will most likely suffice. As a result, undetectable misapplication of many of the algorithms may occur if the underlying mathematics are misunderstood. The reader must also make considerations for what works best in practice in contrast to theory. However, the authors have made an excellent attempt at outlining the limitations of each method, which will assist the reader in correctly applying each algorithm. Overall, the theory is discussed in a concise and easy to understand language, which is much easier to ingest than a typical mathematics text.

The section "Error, Accuracy, and Stability" provides essential insight into the fundamental limitations of numerical computation, and understanding these limitations is also vital to the correct application of the provided source code. Throughout the text this issue is addressed sporadically with reference to individual algorithms. Without this understanding, gross miscalculations may occur as a result of the computational representation of floating point numbers in code that is otherwise semantically correct.

However, *Numerical Methods in C++* makes a good attempt at explaining and reiterating the importance of this subtlety.

Numerical Methods in C++ seldomly discusses the space requirements of each algorithm, though the time requirements of each code segment are extensively covered in a loose language, simple enough for even novice programmers to understand: e.g. in reference to singular value matrix approximation, "If your matrix is approximated by a small number K of singular values, then this computation of A_x takes only about $K(M+N)$ multiplications, instead of MN for the full matrix." In some instances supplementary sections are even included that outline the running time of specific methods, such as matrix inversion in the section titled "Is Matrix Inversion an N^3 Process?"

In spite of the well detailed treatment of the numerical theory in this book, in some instances the code is disappointing. As a result of cryptic meaningless variable names, little white space, and sparse comments, it is clear that space was a consideration when this text was assembled. Furthermore, other than the use of vector and matrix classes, most of the code samples in this text do not utilize the advanced functionality of C++, i.e. few object oriented features are employed. This is most likely a result of the translation of the code from the earlier version of the book written in C. However, array and vector indexing has been edited to start a zero rather than one, which was a major source of confusion with this book's predecessor. Otherwise, despite the lack of object oriented style, the code samples provide elegant implementations of the featured algorithms.

Overall, *Numerical Recipes in C++* contains descriptions of an unparalleled collection of numerical methods, and it is written in a language that is easy to understand. A few new functions have been added to this second edition of this text, however for most practical applications the earlier edition or the C version will suffice. With prior familiarity with the subject material, it is an excellent tool for learning how to correctly apply and implement the contained algorithms. Furthermore, the authors have done an excellent job of alerting the reader to many of the problems that can arise from implementing these algorithms in a computational framework. Overall, despite its shortcomings, *Numerical Recipes in C++* will be a valuable addition to any numerical programmer's book shelf.

Frederick Crimins
Tufts University
March 27, 2002