Preface

Much of the software available today is poorly written, inadequate in its facilities, and altogether a number of years behind the most advanced state of the art.
—Professor Maurice V. Wilkes, September 1973.

Scientific software is central to our computerized society. It is used to design airplanes and bridges, to operate manufacturing lines, to control power plants and refineries, to analyze financial derivatives, to map genomes, and to provide the understanding necessary for the diagnosis and treatment of cancer. Because of the high stakes involved, it is essential that the software be accurate and reliable. Unfortunately, developing accurate and reliable scientific software is notoriously difficult, and Maurice Wilkes' assessment of 1973 still rings true today. Not only is scientific software beset with all the well known problems affecting software development in general, it must cope with the special challenges of numerical computation. Approximations occur at all levels. Continuous functions are replaced by discretized versions. Infinite processes are replaced by finite ones. Real numbers are replaced by finite precision numbers. As a result, errors are built into the mathematical fabric of scientific software which cannot be avoided. At best they can be judiciously managed. The nature of these errors, and how they are propagated, must be understood if the resulting software is to be accurate and reliable. The objective of this book is to investigate the nature of some of these difficulties, and to provide some insight into how to overcome them.

The book is divided into three parts.

1. **Pitfalls in Numerical Computation.**
   We first illustrate some of the difficulties in producing robust and reliable scientific software. Well-known cases of failure by scientific software are reviewed, and the "what" and "why" of numerical computations are considered.

2. **Diagnostic Tools.**
   We next describe tools that can be used to assess the accuracy and reliability of existing scientific applications. Such tools do not necessarily improve results, but they can be used to increase one's confidence in their validity.

3. **Techniques for Building Reliable Scientific Software.**
   We describe a variety of techniques that can be employed to improve the accuracy and reliability of newly developed scientific applications. In particular, we consider the effect of the choice of programming language, underlying hardware, and the

parallel computing environment. We provide a description of interval data types, and their application to validated computations.

This book has been produced by the International Federation for Information Processing (IFIP) Working Group 2.5. An arm of the IFIP Technical Committee 2 on Software Practice and Experience, WG 2.5 seeks to improve the quality of numerical computation by promoting the development and availability of sound numerical software. WG 2.5 has been fortunate to be able to assemble a set of contributions from authors with a wealth of experience in the development and assessment of numerical software. The following WG 2.5 members participated in this project: Ronald Boisvert, Françoise Chaitin-Chatelin, Ronald Cools, Craig Douglas, Bo Einarsson, Wayne Enright, Patrick Gaffney, Ian Gladwell, William Gropp, Jim Pool, Siegfried Rump, Brian Smith, Van Snyder, Michael Thuné, Mladen Vouk, and Wolfgang Walter. Additional contributions were made by Kenneth W. Dritz, Sven Hammarling, Hans Petter Langtangen, Roldan Pozo, Elisabeth Traviesas-Cassan, Bill Walster, and Brian Wichmann. The volume was edited by Bo Einarsson.

Several of the contributions have been presented at other venues in somewhat different forms. Chapter 1 was presented at the Workshop on Scientific Computing and the Computational Sciences, May 28–29, 2001, in Amsterdam, The Netherlands. Chapter 5 was presented at the IFIP Working Group 2.5 Meeting, May 26–27, 2001, in Amsterdam, The Netherlands. Four of the chapters – 6, 7, 10, and 13 – are based on lectures presented at the SIAM Minisymposium on Accuracy and Reliability in Scientific Computing held July 9, 2001 in San Diego, California. Chapter 10 was also presented at the Annual Conference of Japan SIAM at Kyushu-University, Fukuoka, October 7-9, 2001.

The book has an accompanying website `http://www.nsc.liu.se/wg25/book/` with updates, codes, links, color versions of some of the illustrations, and additional material.

A problem with references to links on the internet is that as Diomidis Spinellis has shown in [Spinellis, 2003] the half-life of a referenced URL is approximately four years from its publication date. The accompanying website will contain updated links.

A number of trademarked products are identified in this book. Java, Java HotSpot, and SUN are trademarks of Sun Microsystems, Inc. Pentium and Itanium are trademarks of Intel. PowerPC is a trademark of IBM. Microsoft Windows is a trademark of Microsoft. Apple is a trademark of Apple Computer, Inc. NAG is a trademark of The Numerical Algorithms Group, Ltd. MATLAB is a trademark of The MathWorks, Inc.

While we expect that developing accurate and reliable scientific software will remain a challenging enterprise for some time to come, we believe that techniques and tools are now beginning to emerge to improve the process. If this volume aids in the recognition of the problems, and helps point developers in the direction of solutions, then this volume will have been a success.

Linköping and Gaithersburg, September 15, 2004.

Bo Einarsson
Project Leader

Ronald F. Boisvert
Chair, IFIP Working Group 2.5