Preface

Given the current explosion of data gathering and storage, we face a shortage of trained scientists and engineers who are able to extract knowledge from such data. In order to solve the big problems of today and tomorrow, we need more people with adequate training, and we need to create better tools for extracting knowledge from huge collections of data. Building a new generation of tools will require interdisciplinary teams that can think in new ways. In order for these teams to be productive, they require a common language, and the fundamentals of scientific computing and visualization (SCV) form this language. Thus, the primary goal of *Mathematical Principles for Scientific Computing and Visualization* is to bring SCV tools to a diverse group of people.

Recently, a new trend has started in computer science departments: the move to bring computer- and technology-based knowledge to a broader group of people. This movement is called *informatics*.[1] Computer applications are ubiquitous today, influencing nearly every aspect of our lives. Computer scientists and engineers alone cannot keep pace with the developments. Thus, informatics departments are emerging at colleges and universities to train a new breed of specialists.

[1] This is a US term. In Europe, "informatics" is synonymous with "computer science."

Mathematical Principles for Scientific Computing and Visualization was written with the goals of informatics in mind. In our opinion, it is the first book on SCV that targets such a broad audience. It is intended for students and researchers in engineering and science-related areas, such as biology, geography, and psychology. It will appeal to students because it concisely covers a range of important topics from an application-oriented perspective. Professionals will find this book helpful as a review of key computational methods, or as an update to what they were taught. Individuals in either of these audiences, in the course of their professional applications or research pursuits, will be exposed to various software packages for solving problems, be it problems from statistics, applied mathematics, or scientific visualization, in addition to domain-specific software. This book is intended as a guide to understanding the mathematical principles that underly the more general software packages. That knowledge is important for the non-mathematician because naive and uneducated use of computing and visualization packages might produce meaningless or erroneous results.

This book is written in an informal style and is accessible to someone who is not a mathematician. The book has over 180 illustrations —and not only in the "visualization" part. The reader is led to understand many concepts through graphical examples. Practical suggestions for using the tools of SCV are given, and applications are described in the text and demonstrated with illustrations. Case studies, real-world examples of how one or more tools are used, are included for nearly all topics. Positive feedback on other book projects convinced us that this style of book is of great use to practitioners and people new to a field. (See http://www.farinhansford.com/books.html for a complete list of books by the authors.)

Review of Contents

The book has two basic parts: scientific computing (Chapters 4–11) and visualization (Chapters 12–16). The book is designed so that the reader can begin with either part; however, cross references are given when material is dependent on ideas discussed elsewhere. The first two chapters after the introduction, "Computational Basics" and "Coordinate Systems," introduce the reader to key concepts that are used throughout the book.

The part of the book that focuses on scientific computing begins with topics on numerical linear algebra with Chapters 4, 5, and 6: "Background: Numerical Linear Algebra," "Solving Linear Systems," and "Eigen-Problems."

The second component of the scientific computing part, Chapters 7–11, deals with numerical calculus topics: "Background: Numerical Calculus," "Data Fitting," "Computing Dynamic Processes," "Finding Roots," and "Computing with Multivariate Functions."

The visualization part of the book begins with the most basic tools, which are presented in Chapter 12: "Visualizing Empirical Data." In order to develop more advanced visualization tools, "Facets" is the next chapter, and it focuses on triangle meshes. Chapters 14 and 15, "Visualizing Scalar Values over 2D Data" and "Volume Visualization," present state-of-the-art visualization techniques. For deeper knowledge of visualization, the last chapter, "Background: Computer Graphics," provides details on how objects are rendered in the visualization process.

Each chapter concludes with a "Problems and Experiments" section. The problems are not rote calculations, but rather require some reflection on the topics presented. Experiments are designed to incorporate the use of a software package and thus give the reader hands-on experience with the methods. Only through experimentation does one realize the power and pitfalls of systems such as Mathematica, Matlab, and Maple.

Classroom Use

In a classroom setting, *Mathematical Principles for Scientific Computing and Visualization* targets junior or senior undergraduates. A background in basic computing skills is desirable, as well as some basic knowledge of calculus and linear algebra.

For a one-semester class (and for an audience of varying mathematics backgrounds), the first chapters on computing and coordinates, Chapters 2 and 3, are essential for forming the necessary foundation. If the students have a good mathematical background, then all chapters from Chapter 4 (linear algebra) to Chapter 11 (multivariate data) can be treated in depth, and those on visualization may be given a lighter treatment. Conversely, if students are computationally oriented, there ought to be an emphasis on the visualization part, Chapters 12–16, and a lighter treatment of the scientific computing part.

Website

The book's website is http://www.farinhansford.com/books/scv/index.html. The website contains teaching materials, as well as the figures and code used in the book. We used Mathematica for computations and for generating many of the figures in the book. Yet this is not a Mathematica-centered book: the text is designed so that readers may equally well use other packages such as Matlab or Maple. Reviews and errata will be posted on the site as well.

Acknowledgments

We want to thank the many people and organizations that gave permission for us to use their images. We credit the many contributors in source lines that appear at the ends of figure captions.

Again, thanks to the great team at A K Peters! They are a pleasure to work with.

Gerald Farin
Dianne Hansford
Arizona State University
May 2008