Generally, textbooks on a topic such as C programming are evolutionary, not revolutionary. In contrast, Harry Cheng’s entry is an innovative textbook and programming environment that defines a new interactive paradigm for learning and using C and C++ that’s particularly well suited for scientists and engineers. Cheng has developed a cross-platform (Windows, Linux, Mac) C/C++ interpreter and environment called Ch that fully supports the 1989 ANSI C (C89) standard. It also supports major features of the C99 standard and C++. The package’s capabilities range from command-line interactive processing and embedded programming to a full integrated development environment (IDE) for program development and debugging. Ch is unique for scientific programming in that it includes appropriate data types (such as complex numbers and computational arrays); a complete 2D and 3D plotting package; and an extensive library of numerical analysis functions, including tools for data analysis, root finding/optimization, linear algebra, fast Fourier transforms, special functions, and numerical solutions to differential equations. The Ch interpreter system is thoroughly integrated into the textbook’s presentation, and thus offers a comprehensive package for C programming.

In my January 2010 programming course, the students were almost equally split between Mac and PC users. Cheng’s software and textbook package let us all focus on developing programs, without concerns about merging computer tools—that is, compilers and plotting and analysis packages—on multiple platforms.

**Overview: C and (a Bit) Beyond**

Chapter 2 begins with the requisite introduction of “Hello, World.” This chapter also gives students an overview of how to use the ChIDE system to run and debug programs, and how to use the Ch command shell to execute Unix-like commands and interactive Ch programs. Finally, Chapter 2 explains how to set up the environment to use an external compiler, such as gcc or Microsoft Visual C++, so students can compile and link executable programs. Thus, by the chapter’s end, students are ready to run their programs either interactively or using an external compiler/linker. Testifying to the Ch system’s strength, only a few students in the course decided to install a C/C++ compiler, and they did so only because they needed faster execution times for their final projects than were possible using the interpreted Ch language.

The bulk of the textbook—Chapters 3 through 16—offers a thorough introduction to the C programming language. Each chapter includes numerous example programs with pseudocode and flowcharts, and a large collection of end-of-chapter problems. Starting in Chapter 6, the textbook introduces the SoftIntegration Graphics Library (SIGL), which you can use to plot functions and data sets in 2D and 3D. Because my course strongly emphasizes numerical solutions to science and engineering problems, visualizing results is invaluable. This graphics package made it straightforward for students to plot their simulation results, whether on a Mac or a PC. Most chapters also included a “Making It Work” section with advanced topics such as defining compiler macros or using the `make` command. In a post-use survey, students reportedly felt that the book contained enough examples to help them with their assignments, but few had read these “Making It Work” sections.

Chapters 18 and 19 introduce basic concepts of C++ object-oriented programming, while Chapter 20 offers a more complete summary of the SIGL C++ graphics library. Placing this C++ material toward the book’s end insures that it doesn’t interrupt the flow of the introduction to C. However, when students got to this section, they felt like they were starting over, such as when they learned about stream input/output and enhanced C++ features such as classes and
Computing in Science & Engineering

Strengths and Shortcomings
By far, the textbook’s greatest strength is its close integration with the Ch interpreter system, which includes ChIDE, a scientific graphing package, and a full library of numerical analysis subroutines. Chapters 17, 21, and 22 introduce some of the Ch features that make it an excellent language choice for scientific programming, including complex numbers, computational arrays, and advanced numerical analysis subroutines.

The textbook concludes with an extensive chapter introducing Matlab, including useful tables that directly compare operators and functions in Matlab and Ch. Although the book is obviously focused on C programming, this Matlab chapter was one of the reasons I chose it. Throughout my course, I introduced Matlab structures—such as arrays, conditional execution, and loops—as students learned their C counterparts. Most homework sets included programming problems in both C and Matlab. Unfortunately, all of Chapter 23’s exercises involve having students use Matlab to solve corresponding exercises from earlier sections. It would be beneficial to incorporate some problems that focus more specifically on Matlab’s unique vector-based paradigm, particularly for solving array-based problems and linear algebra.

Finding, and one of the exercises requires students to modify it to implement Newton’s method. Chapter 10 includes sections on linear algebra and introduces the subroutine linsolve(), a built-in Ch function, and Clapack’s dgesv_() function. Beyond these examples and a brief discussion in Chapter 22 of some of Ch’s numerous built-in numerical functions, there’s no discussion of or exercises for other numerical techniques, such as numerical integration/differentiation and solving ordinary differential equations (ODEs).

In my view, there’s pedagogical value in having students code and understand basic algorithms, such as solving an ODE using a simple Euler, predictor/corrector, or fourth-order Runge-Kutta method. Doing so helps them understand some of the tradeoffs of accuracy, round-off error, execution times, and step size that are less readily apparent when using a canned function such as odes() (which is Ch’s built-in fifth-order adaptive step-size Runge-Kutta subroutine. Because science and engineering students are likely to encounter numerical modeling at some point in their academic or professional career, a programming course is an ideal place to empower them with an understanding of the basic principles of numerical analysis. Hopefully, a second edition of the textbook will incorporate more extensive discussion of these techniques, along with appropriate examples and exercises.

Obviously, an essential feature of this textbook package is the Ch interpreter system. In running several simple benchmark programs, code executes at about 1/1,000 the speed of compiled C code. In previous years, such a huge difference in performance greatly limited interpreters’ appeal.
However, because we now have faster processors, the Ch interpreter’s speed was more than adequate for all but the most demanding computational tasks. And, for the most part, the cross-platform compatibility and convenience of not having to compile and link code outweigh any performance penalty. The only significant difference between the Ch system’s student and professional editions is the limitation of arrays to 20,000 elements in the former; this was generally not a problem for the programs and projects in my course.

SoftIntegration’s technical support staff was highly responsive; when we notified them of bugs, they generally made a patched version available within a day. The extensive numerical analysis library’s Ch Language Environment Reference Guide includes well-documented subroutines for many of the numerical tasks that an engineering or science student is likely to encounter. However, because the book offers only a minimal introduction to numerical methods, students might not be in a position to fully utilize this library. This is because the book fails to introduce the advantages and disadvantages of numerical versus analytic techniques, so students might lack the sensitivity required to set up problems and interpret results in light of round-off error, step sizes, and so on.

The book includes a CD of the student edition of Ch for Windows, version 6.1 and indicates that students can download and license the corresponding version for Mac OS/X and Linux free of charge. However, as of this writing, SoftIntegration changed its licensing and now charges to purchase the latest student version (currently 6.3) from its website, though it does offer a 30-day free trial. Instructors can distribute the latest version of the student edition to their classes if they use the textbook and purchase a license for the software’s professional edition for their instructional lab. While the pricing—particularly for academic and student licenses—is very competitive, changing the student edition’s licensing might reduce the package’s appeal, particularly for students who want to learn C/C++ on their own.

The bottom line for a post-use review is one question: Would I use the text again? In the case of C for Engineers and Scientists. An Interpretive Approach, the answer is yes. I dealt with the book’s major weakness—the lack of an introduction to numerical methods—by including handouts and supplemental homework. And ultimately, the weakness was overwhelmed by the textbook’s exceptional strength: it let me introduce the C programming language within a cross-platform environment that enabled numerical analysis and visualization. Students agreed with this assessment; in their post-use survey, when asked whether the book should be used in future course offerings, all students answered “yes.” After using this textbook and the Ch interpreter, I have a new respect for the role of interpreters as an alternative to traditional compilers for introducing computer programming.

Reference

Tom Huber is a physics professor at Gustavus Adolphus College in St. Peter, Minnesota. He has extensive programming and modeling experience in both academia and industry, and frequently teaches a scientific programming course during the college’s month-long January term. Huber has a PhD in experimental nuclear physics from the University of Wyoming. Contact him at huber@gac.edu.

COMSOL MULTIPHYSICS FOR DUMMIES

By Christian Schröder


Roger W. Pryor, Multiphysics Modeling Using COMSOL—A First Principles Approach, has identified and filled the missing link in the existing literature on finite element modeling (FEM) using Comsol Multiphysics. This is a hands-on introduction that lets scientists and engineers quickly start building and solving models of different physical device structures to see how they behave on a computer before attempting to build physical prototypes. This book is an excellent resource, written to be accessible to rookies in the FEM field as well as offering valuable information for advanced users.

In the book’s introductory chapter, Pryor explains how to read and
use the book. He includes two detailed tables to help readers quickly find the specific concepts and techniques presented in various chapters. The introduction also offers the details required to get started with the software, including installation and hardware (which are important to the Comsol Multiphysics modeling environment’s performance) as well as coordinate systems.

In addition to these technical matters, the author also offers an initial introduction to several important aspects of the FEM modeling process by demonstrating how to create simple 1D models of one-, two-, and three-pane thermal insulation windows. Chapter 2 completes the introduction by addressing materials and material databases, including a discussion of how to obtain additional resources related to material properties. This topic is vital to the modeling process, as the accuracy and reliability of a model’s results depend heavily on the material properties used in the simulation.

**A Practical, Hands-On Guide**

In subsequent chapters, the author provides detailed hands-on practice of FEM modeling using Comsol Multiphysics with increasing complexity, ranging from 1D models based on the Korteweg-de-Vries equation for modeling soliton wave propagation (Chapter 3) to complex 3D models of a Helmholtz coil’s magnetic field (Chapter 8). In Chapter 9, Pryor discusses the concept of perfectly matched layers for modeling reflection-free boundaries in electromagnetic problems, and the final chapter deals with the bioheat equation. This last topic plays an important role in developing and analyzing new therapeutic medical techniques (such as killing tumors), and the author presents two bioheat models that demonstrate the effect of laser heating and microwave heating on biological tissue.

At first glance, 852 pages about FEM using a single tool might seem daunting. However, the book is not a mathematics book. Pryor completely leaves out the mathematical background of the FEM and numerical solving techniques. This is an excellent approach; many other books treat that topic in depth, and such knowledge isn’t particularly important for the practical modeler.

In contrast to other FEM books—even those tightly related to Comsol Multiphysics—this book isn’t dedicated to one special engineering field, such as mechanical engineering or computational fluid dynamics. Instead, it deals with a wide variety of application modes available in Comsol Multiphysics. Given this variety, it’s important that the book be organized such that readers can jump directly to chapters of interest, without missing important information from other chapters. This book achieves that and offers further aid by providing a consistent structure within each chapter (except Chapter 2, which deals with materials). First, each chapter offers an “In This Chapter” paragraph, which summarizes the content, followed by modeling considerations and physical background. The description of the actual modeling process also adheres to a specific pattern as follows:

- select one of the built-in solvers to solve the partial differential equations;
- specify the subdomain settings, including material properties and initial values;
- visualize, interpret, and discuss the results.

Finally, each chapter ends with a short summary.

Almost every page contains detailed screenshots that guide readers safely through the modeling process. In fact, by reading the book and simultaneously carrying out each modeling step on their computer screens, readers can quickly learn the basic ideas required for successful modeling with Comsol Multiphysics. The author also describes real-world applications for each model, which makes the text even more satisfying.

**Key Features and Drawbacks**

Perhaps the book’s most important feature is the notes that Pryor inserts between the paragraphs. Here, the reader gets key information taken directly from the author’s practical experience. This gives the impression of attending a “virtual” training course. It’s as if the author is standing right behind you while you’re reading and modeling, giving you important information that you need—without charging an extra training fee. Once again, the 852 pages don’t contain condensed scientific information. In fact, readers will find many identical paragraphs, phrases, and notes that reappear throughout the book. Although this adds length, it also frees
readers from needing to mark the pages where important information appears only once, which is usually the case in “traditional” science books.

This book is definitely not useful for the reader who is generally interested in FEM and numerical methods; it’s designed specifically for hands-on modeling with Comsol Multiphysics, and as such it’s readable only in combination with this software. Moreover, several chapters require additional modules (such as the Heat-Transfer-, AC/DC-, and Materials-Library-module) that aren’t included in the basic software package. Pryor provides a table in the introduction that shows which chapter requires which module; however, given that these additional modules aren’t free of charge, it would have been useful to mention the need for them on the book’s back cover to offer a clear warning to potential readers.

With the exception of Chapter 1, the author begins all technical chapters with the development of a model derived from models available in the Comsol Multiphysics installation package. Following this is a model variation and analysis of the results. Although each model’s construction is described and discussed in great detail, most models lack important information regarding the physical background, especially regarding the use of specific Comsol tool settings or application modes. In Chapter 1, for example, Pryor develops a series of 1D window models based on one-, two-, and three-glass panes. For the boundary conditions, he selects “Heat flux” and instructs the reader to enter a value of 15 in the “Heat transfer coefficient” window; however, he doesn’t explain the value’s origin or meaning.

From my own experience in teaching mathematical modeling with Comsol Multiphysics, I know that defining and setting the correct boundary conditions is very challenging for beginners. I therefore would expect a more detailed discussion about the types of boundary conditions in general and the source of this specific value. A similar situation occurs in Chapter 4. Here, Pryor introduces the concept of the moving mesh application mode in electrochemical polishing and the use of weak constraints in the Hall effect model. In both cases, he instructs readers to select a particular setting without offering any additional explanation of these powerful techniques. The only hint given is a reference to the Comsol Multiphysics Modeling Guide that comes with the software.

Two additional aspects of the book leave significant room for improvement in future editions. First, each chapter includes references, but most are links to Wikipedia or to the software’s manual. As readers will quickly realize, a simple Google search would yield the same information. Similarly, most of the “problems” that appear following each chapter simply repeat what’s presented earlier in the chapter, and few actually deal with interesting extensions of the material that has been discussed.

The book offers a highly useful DVD supplement containing all models developed and discussed in each chapter, as well as Microsoft Word documents containing all of the book’s figures and screenshots. The latter is very valuable for teaching purposes.

Generally, the book is well and clearly written, which is especially important for international users. The type setting of equations is sometimes rather poor. However, because the book doesn’t focus on mathematics, this is acceptable.

Pryor did a great job of putting together examples for many different application domains that can be solved by using Comsol Multiphysics. In each chapter, he clearly describes the modeling process, offers detailed discussion of possible modeling errors and results, and provides the most important background information. Together with the original Comsol handbooks provided with the Comsol Multiphysics software package, this book is an excellent starter and is highly valuable for teaching purposes. Advanced modelers who want to know more about mathematical or physical details can find them in other books on FEM. Several such books use Comsol Multiphysics, but typically focus only on specific application domains. In this regard, Pryor’s book is very much a missing link, offering a useful supplement to the literature on FEM modeling with Comsol Multiphysics.

Christian Schröder is a professor in the Department of Engineering Sciences and Mathematics at the University of Applied Sciences Bielefeld in Germany. His current research involves high-performance computing, computational material science, and engineering, with a focus on finite element modeling using COMSOL Multiphysics. Schröder has a PhD in theoretical physics from the University of Osnabrück in Osnabrück, Germany. Contact him at christian.schroeder@fh-bielefeld.de.

Selected articles and columns from IEEE Computer Society publications are also available for free at http://ComputingNow.computer.org.