Mathematica: A Problem-Centered Approach

By Roozbeh Hazrat
Springer-Verlag, London, 2010
ISBN 978-1-84996-250-6

Review by
John D. Carter
Seattle University
February 2011

As we endeavor to solve increasingly complex problems, computer algebra systems (CAS) are becoming more and more important to, or perhaps necessary for, our work. We use CAS to help with calculations too tedious, messy, or time consuming to do by hand. Mathematica, developed by Wolfram Research Inc., is one of the most popular computer algebra systems available today. It is used by a broad range of people: beginning calculus students to experienced research scientists. Dr. Hazrat's book aims to introduce and teach Mathematica to both beginners and those looking to understand some of the finer details of programming in Mathematica.

Dr. Hazrat's book arose from a undergraduate-level course he taught over a five-year period. It introduces Mathematica's features and abilities by presenting detailed, worked-out example problems. Although the book includes two sections directly related to calculus, it is probably not a good text for use in a Mathematica laboratory portion of an introductory calculus class because it focuses on lists, sets, logical operations, and repeated operations. The book is much more well suited to a course in number theory or programming.

The book begins by quickly introducing how to use Mathematica as a calculator. Variables, simple algebraic calculations, functions, and basic Mathematica notation are discussed in the remainder of the introductory chapters. Nothing too exciting, just basic notation and syntax.

The heart of the text begins in Chapter 3 where lists are introduced. Lists are collections of data and are one of the basic Mathematica programming constructs. A variety of problems related to lists are introduced. For example, the book presents a simple one-line program that determines when $2^n + 1$ is prime when $n \in [1, 1000]$. The solution of $\{1, 2, 4, 8, 16\}$ is found in less than one second on a typical laptop computer. Although this example is pretty simple, the sample program demonstrates how to efficiently work with lists in Mathematica.

Chapter 5 presents some logic and set theory. The chapter begins by presenting Mathematica code for Boolean expressions. Shortly thereafter, simple unions, intersections, and other standard list/set operations are introduced. Demonstrating the wide range of problems presented in the book, the chapter ends with a discussion of the Collatz conjecture: *If one repeatedly applies the function*

$$f(x) = \begin{cases} \frac{x}{2} & \text{if } x \text{ is even} \\ 3x+1 & \text{if } x \text{ is odd} \end{cases}$$

to any positive integer one will eventually arrive at 1. This is an open problem that was proposed by Lothar Collatz in 1937. Obviously, Mathematica cannot prove the conjecture. However, the book contains an easy Mathematica program that can be used to test the conjecture for any number.

Chapters 6 through 8 introduce sums, products, loops, and substitution rules. These chapters are my favorites because I learned the most from them. Most importantly, I learned a trick to make a number of my existing Mathematica programs faster. The first example in Chapter 7 includes sample code that shows a sequence of commands that can be used to replace do loops. This sequence of commands does the same thing as a do loop, but is noticeably faster. Chapter 8 focuses on substitution and replacement rules. These rules are another important programming construct in Mathematica. They allow one to substitute a variable with a value without assigning the value to the variable. This ability allows one to study repeated fractions and to perform repeated tests of function values over a predetermined sequence of values. Mathematica excels at these repeated, tedious calculations.

Mathematicians are often very interested in pattern matching. This is the focus of Chapter 9. Problem 9.2 presents two methods for determining which of the first 50,000 primes have 2009 embedded in them (for example, 420097 is prime and contains the pattern 2009). It turns out that there are 26 such numbers. Problem 9.5 contains code that finds all words in the Mathematica dictionary that contain the letters "c", "u", "t", and "e". Although this later problem isn't directly very useful, it demonstrates how to write code that efficiently searches for given patterns.

Chapters 10 and 11 cover multiply-defined and recursive functions respectively. These chapters contain several detailed examples that demonstrate how to define and use piecewise functions. They also show methods for recursive sequences and functions. For example, Chapter 11 starts with a program that recursively defines the Fibonacci numbers. Again the program itself isn't that useful as Mathematica has an intrinsic function that returns the Fibonacci numbers. The benefit of including this example is that it is simple and clearly demonstrates how to define recursive functions in Mathematica.

The book concludes with discussions of basic linear algebra, graphs, and calculus-related commands. Of these, the most thorough is the chapter on graphs. The author demonstrates Mathematica's robust graphing ability by running through a series of example plots. Among many others, examples of polar plots, random walks, and surface plots are included. Although these examples are brief, they contain all of the details necessary to create plots of your own favorite functions.

In summary, Hazrat's book contains many short Mathematica programs that solve simple mathematical problems. The programs demonstrate how to use a wide range of Mathematica commands to solve a wide array of problems. Finally I would like to add that I enjoyed reading this book and learning a number of Mathematica "tricks" while doing so.