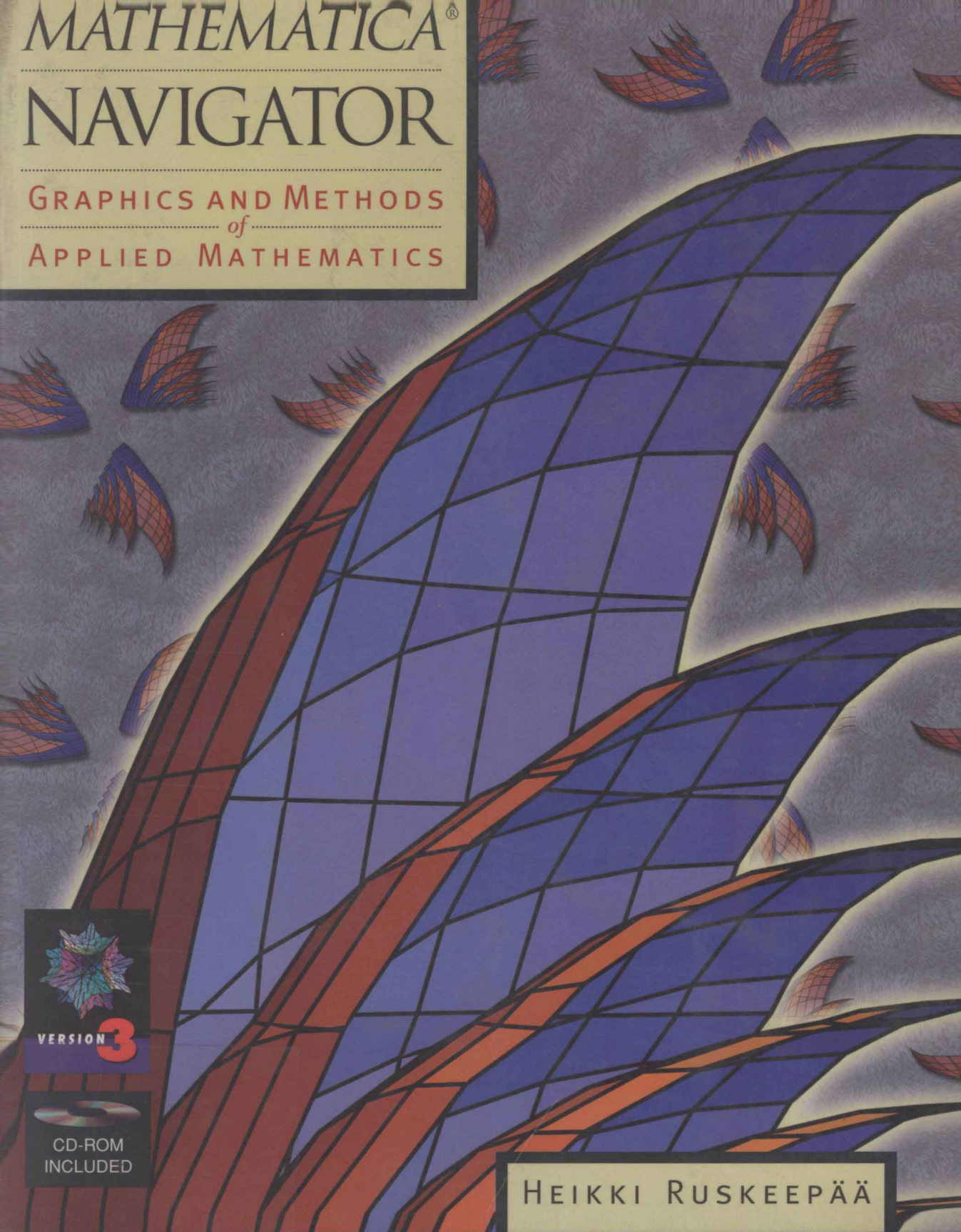
MATHEMATICA®
NAVIGATOR

GRAPHICS AND METHODS of APPLIED MATHEMATICS



VERSION 3

CD-ROM INCLUDED

HEIKKI RUSKEEPÄÄ

Mathematica®
Navigator

Graphics and Methods of Applied Mathematics

Heikki Ruskeepää
Department of Applied Mathematics
University of Turku, Finland

ACADEMIC PRESS
San Diego London Boston
New York Sydney Tokyo Toronto

This book is printed on acid-free paper. ♾

Academic Press
525 B Street, Suite 1900, San Diego, CA 92101-4495, USA
http://www.apnet.com

Academic Press
24–28 Oval Road, London NW1 7DX, UK
http://www.hbuk.co.uk/ap/

Library of Congress Cataloging-in-Publication Data

Ruskeepää, Heikki.
Mathematica navigator : graphics and methods of applied mathematics / Heikki Ruskeepää.
p. cm.
Includes bibliographical references and index.
ISBN 0-12-603640-3 (pbk. manual : alk. paper) — ISBN 0-12-603641-1 (CD-ROM)
1. Mathematica (Computer file). I. Title.
QA76.95.R87 1998
510'.285'53042—dc21 98-22863
CIP

Printed in the United States of America
98 99 00 01 02 03 IP 9 8 7 6 5 4 3 2 1

Mathematica®
Navigator

To Marjatta

Preface

What is the difference between an applied mathematician and a pure mathematician?
An applied mathematician has a solution for every problem,
while a pure mathematician has a problem for every solution.

The aim of this book, *Mathematica Navigator: Graphics and Methods of Applied Mathematics*, is

- to introduce the reader to *Mathematica* and
- to emphasize graphics, methods of applied mathematics and numerical analysis, and programming.

So, we navigate the reader through *Mathematica* and try to give an overall introduction. Often we slow down somewhat when an important or interesting topic of applied mathematics or numerical analysis is encountered, to investigate it in more detail, often using graphics and programming.

Mathematica has four main areas: graphics, symbolic calculation, numerical calculation, and programming. In the mathematical parts of the book in particular, all four areas are constantly present. Other parts of the book specialize in graphics, programming, and other topics.

■ Previous Knowledge

No previous knowledge about *Mathematica* is assumed. In other words, this book may be used as an introduction to *Mathematica*.

On the other hand, we assume some knowledge of various topics in pure and applied mathematics. We study, for example, approximation, partial differential equations, and statistics, without giving detailed introductions to these topics. If you are not acquainted with a topic, you can skip the chapter or section considering that topic (or you can browse through the chapter or section and try to get an idea of the subject, or even learn something about it).

Also, to understand the numerical algorithms it is useful for the reader to have some knowledge of the simplest numerical methods. Often we introduce briefly the basic ideas

of a method (or they may become clear from the examples or other material presented), but usually we do not derive the methods. If a topic is unfamiliar to you, then consult a textbook on numerical analysis, such as Skeel & Keiper (1993).

■ Special Aspects

The book covers a substantial part of the material in *Mathematica*. However, some topics are emphasized, some are given less emphasis, and some are even excluded. Next we explain these special aspects of the book.

■ Going Deeper and Wider

We have had the goal of studying at least important topics in some depth and width. This may mean detailed explanations and clarifying examples and applications. It may also mean introducing topics for which there is little or no material in the manuals. Here are some examples of both special emphasis (marked with *) and special topics (marked with -) in the book:

Graphics:

* graphics of functions (two chapters)
  - stereographic figures
  - graphics of four-dimensional functions
* fine-tuning graphics with options (two chapters)
  - using text, legends, and other primitives in graphics
* graphics of data (two chapters)
  - visualizations of real-life data
  - plotting irregular three-dimensional data
  - one- and multiway dot plots
  - graphics of four- and higher-dimensional data

Mathematics:

* differential calculus
* integral calculus
  - numerical quadrature
* equations
  - iterative methods of solving linear equations
* optimization
  - classical optimization with equality and inequality constraints
  - constrained nonlinear optimization

* interpolation
  - interpolation of irregular three-dimensional data
* approximation
  - graphical diagnostics of least-squares fits
  - least-squares and Chebyshev approximation of functions
* differential equations
  - visualizing solutions of systems of nonlinear differential equations
  - numerical solution of linear and nonlinear boundary value problems
* difference equations
  - nonlinear difference equations
  - bifurcation diagrams, Lyapunov exponents, fractal images
* partial differential equations
  - series solutions
  - solving elliptic problems by the finite difference method
* probability
  - simulation of several stochastic processes
* statistics
  - visualizing confidence intervals and errors in statistical tests
  - confidence intervals and tests for probabilities

Programming

* three styles of programming (procedural, functional, and rule-based)
* emphasis on functional programming
* many examples of programming

■ Writing Our Own Programs

Mathematica has many ready-to-use commands for symbolic and numerical calculations and for graphics. Nevertheless, we often present our own programs, too. We think that such implementations can be of some value, in spite of the fact that they are not as fine and powerful as *Mathematica*'s built-in commands.

- First, a self-made implementation shows clearly how the algorithm works. You know (or should know) exactly what you are doing when you use your own implementation. The ready-made commands are often like black (or gray) boxes, because we do not know much about the methods.
- Also, writing our own implementations teaches us programming. We present short programs throughout the book (especially in Chapters 15–25). In this way, we hope that the reader will become steadily more familiar with programming and she or he is encouraged to practice program writing.

- In addition, a self-made implementation can be pedagogically worthwhile. For example, we implement Euler's method for differential equations. It has almost no practical value but, as the simplest numerical method for initial value problems, has a certain pedagogical value. Also, programming a simple method first may help us to tackle a more demanding method later.

We will write programs, for example, for

Sputnik, stereogram, Mandelbrot image;

gradient, Hessian, Laplacian, Jacobian, divergence, numerical derivatives;

Simpson's method, trapezoidal method, Gaussian quadrature;

norms, Gauss–Seidel method, Newton's method, secant method;

Lagrange's multipliers, Dawidon–Fletcher–Powell method;

Lagrange's interpolation, Newton's interpolation, Chebyshev interpolation;

least-squares method for data and functions, Chebyshev approximation;

Euler's method, Runge–Kutta method, methods for linear and nonlinear boundary value problems;

bifurcation diagram, Lyapunov exponent;

finite difference method for elliptic partial differential equations;

simulating random walks, Wiener process, gambler's ruin process, branching process, Markov chain, Poisson process, birth and death process, M/M/1 queue;

showing simulated confidence intervals and errors of statistical tests, confidence interval and test for probabilities.

■ Other Special Aspects

We have integrated the so-called packages tightly into the material. Instead of presenting a separate chapter about packages, each package is explained in its proper context. In this way, it is hoped that the reader's attention will be drawn more effectively to the many excellent packages.

We have structured the book so that finding a topic is easy. Usually a topic is considered in one and only one chapter or section, so that you need not search in several places to find the whole story. Each numerical routine is also presented in the proper context, after the corresponding symbolic methods. This helps you to find material for solving a given problem: it is usually best to try a symbolic method first, and if this fails, to resort to a numerical method.

Topics of a "pure" nature such as number theory, finite fields, quaternions, or graph theory are not considered in this book. Commands for string, box, and notebook manipulation are treated only briefly. We do not consider *MathLink* (a part of *Mathematica* enabling interaction between *Mathematica* and external programs).

■ Contents

The contents of the book can be divided into six main parts:

- Introduction (Chapters 1 and 2)
- Housekeeping (Chapters 3 and 4)
- Graphics (Chapters 5 through 10)
- Expressions (Chapters 11, 12, and 13)
- Programming (Chapters 14 and 15)
- Mathematics (Chapters 16 through 25)

The first two chapters introduce *Mathematica* and give a short overview.

The next two chapters consider notebooks, inputs and outputs of *Mathematica*, customizing *Mathematica*, packages, reading and writing data, and saving memory and computing time. You may skip these chapters until you need them.

Then we go on to graphics. One of the finest aspects of *Mathematica* is its high-quality graphics, and one of the strongest motivations for studying *Mathematica* is to learn to illustrate mathematics with figures. The material on graphics is divided into six chapters: graphics for two-dimensional functions (Chapters 5 and 6), graphics for three- and four-dimensional functions (Chapters 7 and 8), and graphics for data (Chapters 9 and 10).

In the next three chapters we study the basic objects of *Mathematica*, namely expressions. We consider numbers in *Mathematica* and emphasize topics about floating-point numbers and precision of numerical routines. We explain how to manipulate expressions, and we consider special types of expressions. Chapter 13 introduces the very basic objects of *Mathematica*, namely lists. Matrices are expressed as lists, and so we also consider matrices in Chapter 13.

The following two chapters are about user-defined functions and programming. We consider special types of functions (such as composite, piecewise-defined, periodic, recursive, and implicit functions), so-called pure functions (very important in *Mathematica*), modules (functions consisting of several commands), and packages (collections of functions). Three types of programming are introduced: procedural, functional, and rule-based (object-oriented programming is not considered). Programming is one of the strongest points of *Mathematica*. It is often amazing how concisely and efficiently we can write a program even for a somewhat complex problem. We can also efficiently write programs for graphics production. Functional and rule-based programming are generally recommended because they are efficient, but procedural programming can also be used.

In the remaining ten chapters we study different areas of pure and applied mathematics. The chapters are in pairs of more or less related topics:

- Differential Calculus • Integral Calculus
- Equations • Optimization
- Interpolation • Approximation
- Differential and Difference Equations • Partial Differential Equations
- Probability • Statistics

Topics of traditional differential and integral calculus include derivatives, Taylor series, limits, integrals, sums, and transforms.

We consider linear, polynomial, and transcendental equations, and linear and nonlinear programming and classical optimization.

In interpolation we have the usual interpolating polynomial, a piecewise-interpolating polynomial or surface, and splines. In approximation we distinguish approximation of data and functions. For the former we can use the linear or nonlinear least-squares method, while for the latter we have, for example, Chebyshev and minimax approximation.

Differential equations can be solved symbolically and numerically. We can solve first- and higher-order equations, systems of equations, and initial and boundary value problems. We also consider difference equations. For linear difference equations we can possibly find a solution in a closed form, but nonlinear difference equations must be studied in other ways. For partial differential equations we show how to use the Laplace transform, how to handle series solutions, and how to solve some problems numerically with the method of lines or the finite difference method.

Mathematica contains information about most of the known probability distributions. Simulation of various random phenomena is done nicely with random numbers. Statistical topics include descriptive statistics, smoothing, confidence intervals, tests of hypotheses, and linear and nonlinear regression.

■ Recommendations

If you are a newcomer to *Mathematica*, then Chapter 1 "Getting Started" is mandatory and Chapter 2 "A Quick Tour" is strongly recommended. You can also browse Chapter 3 "Notebooks, Customizing, and Packages" and perhaps also Chapter 4 "Data, Memory, and Time," so that you know where to go when you encounter the topics of these chapters. After that you can proceed more freely. However, read Section 12.1, because it contains some very basic techniques used constantly for expressions.

If you have some previous knowledge of *Mathematica*, you can probably go directly to the chapter or section you are interested in, with the risk, however, of having to go back to study some background material. Again, read Section 12.1.

■ Some Notes

Version and Environment The book has been written with *Mathematica* version 3.0.1, using a Macintosh. Mathematical formulas of the book are also written with *Mathematica*.

Mathematica works in much the same way in various environments, with only the keyboard shortcuts of menu commands varying for the most part. To some extent, we mention the shortcuts for the Microsoft Windows and the Macintosh environment.

Version 3 of *Mathematica* contains many new and modified commands and features as compared with version 2. We have denoted such new properties with [✵❸].

Saving Space Some techniques have been used to save space in the book. One of them is using a small size for the figures. When you produce graphics, use a larger size, to make the figures clearer and more impressive (the default size is, in fact, much larger).

Often we produce a series of figures, showing various aspects of a topic, but to save space, we usually do not show the separate figures one after another, but we gather them together and show them either superimposed or side by side. We hope you do not find the technique too confusing or difficult. After you have learned the technique, you will probably feel comfortable with it. You may observe that it also saves space on the screen and helps you to make comparisons between the figures.

A third method to save space is to write simple results of commands next to the command, in the same row. Again, we hope this does not confuse you. The command and its result can always be identified from the style of the font: the command is in the **`bold style`** and the result in the `plain style`.

Many commands of *Mathematica* have so-called options for modifying the commands in some ways. All options have a default value, but we can give other values. When listing the options, we give either all possible values of them or some examples of values, but we do not explicitly mention what the default values are, to save space. The default value is always the first value mentioned. After that come other possible values or examples of other values.

The CD-ROM The CD-ROM that comes with the book contains the entire book. With some easy steps you can install the book to the Help Browser of *Mathematica* (the CD-ROM contains installation instructions). With the Help Browser you can then easily find and read sections of the book and copy material from the book to your document. With the Help Browser you can see all the figures of the book in color (color figures appear especially in Sections 7, 8, 10, and 23) and show all animations explained in the book.

In addition, the CD-ROM contains some data. **`visdata`** is a collection of data sets relating to Cleveland (1993) (the collection is reproduced with the permission of the publishers, Hobart Press). All the data sets are explained and illustrated in the book mentioned. We visualize some of the data sets (a few of them are used in a slightly modified version, and are also included). Other data sets of the CD-ROM are **`hare`** and **`lynx`** from Burghes & Borrie (1981) (reproduced with the permission of the authors) and **`precipitation`** from Guttorp (1995) (reproduced with the permission of the publishers, Chapman & Hall). These sets, too, are used in some examples of this book.

Some packages in the CD-ROM contain most programs in the book.

Comments Please send us comments! Comments on anything are welcome: misprints, errors, unclear topics, topics requiring more material, topics lacking, topics of no use, etc.

■ Acknowledgments

Four sources have been invaluable in preparing this book. *The* Mathematica *Book* by Stephen Wolfram (Wolfram 1996) is, of course, the most important source, followed by the manual for the packages (Wolfram Research 1996). The article Keiper (1993) explains most of the numerical methods. The anecdotes at the beginning of the chapters are from the wonderful

book by MacHale (1993) (the anecdotes are reproduced or adapted with the permission of the publishers, Boole Press, 26 Temple Lane, Dublin 2, Ireland).

I have been lucky enough to enjoy excellent working conditions at the Department of Applied Mathematics of the University of Turku. For this my sincere thanks are due to professors Ulla Pursiheimo and Mats Gyllenberg. I thank Robert M. Dickau (Wolfram Research, Inc.), Veikko Keränen (Rovaniemi Institute of Technology), Simo Kivelä (Helsinki University of Technology), and Pertti Näykki (Business Advantage Development) for useful comments and suggestions.

I am very grateful to Rosemary Mackenzie (University of Turku) for patiently and accurately correcting my English (until now I believed my English was quite good...). My wife Marjatta has done a great job in proofreading the manuscript, but actually her contribution has extended far beyond the usual proofreading (until now I believed my text was quite correct and clear ...). Many, many thanks to both of you! (Any errors remaining are mine.)

For financial support I express my deep gratitude to Turun Yliopistosäätiö (Turku University Foundation), Academic Press, Suomen tietokirjailijat (The Association of Finnish Non Fiction Writers), and Suomalaisen kirjallisuuden edistämisvarojen valtuuskunta (The Commission for the Promotion of Finnish Literature).

Heikki Ruskeepää
Department of Applied Mathematics
University of Turku
FIN-20014 Turku
Finland
ruskeepa@utu.fi

About the CD-ROM

The CD-ROM contains:

- the entire text of the book,
- all animations discussed in the book,
- all data sets illustrated in the book—plus several other data sets,
- packages containing most programs developed in the book.

The CD-ROM contains detailed instructions for installing the entire book to *Mathematica* 3's Help Browser, giving users direct access to the material of the book via the Help Browser. Users can then:

- read all of the text,
- experiment with the examples,
- run all of the animations,
- see the colors of the figures,
- copy *Mathematica* 3 code from the book to their own notebooks.

The data sets on the CD-ROM allow users to work out examples where data sets are illustrated graphically.

The CD-ROM can be read using Windows, Macintosh, and Unix computers.

Contents