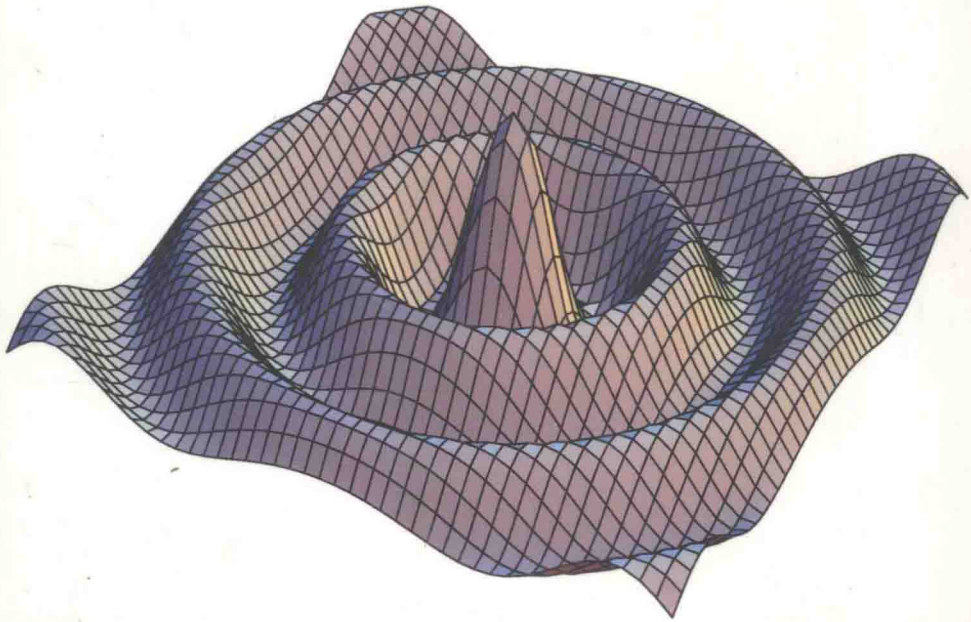


JOHN S. ROBERTSON

ENGINEERING
MATHEMATICS
WITH
MATHEMATICA



ENGINEERING MATHEMATICS WITH MATHEMATICA

John S. Robertson

*Professor of Applied Mathematics
United States Military Academy*

McGraw-Hill, Inc.

New York St. Louis San Francisco Auckland Bogotá Caracas
Lisbon London Madrid Mexico Milan Montreal New Delhi
San Juan Singapore Sydney Tokyo Toronto

ENGINEERING MATHEMATICS WITH MATHEMATICA

Copyright © 1995 by McGraw-Hill, Inc. All rights reserved. Printed in the United States of America. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a data base or retrieval system, without the prior written permission of the publisher.

This book is printed on acid-free paper.

2 3 4 5 6 7 8 9 0 DOC DOC 9 0 9 8 7 6 5

ISBN 0-07-053171-4

The editors were Maggie Lanzillo and Jack Shira; the production supervisor was Denise L. Puryear. R.R. Donnelley & Sons was printer and binder.

The illustration shown on the cover was generated by *Mathematica* and depicts the surface created by rotating the graph of the Bessel function of the first kind of order zero about the z -axis. In particular,

$$z = J_0(\sqrt{x^2 + y^2})$$

with $-15 < x < 15$ and $-15 < y < 15$.

This manual was typeset by the author with L^AT_EX, the document preparation system by Leslie Lamport built atop the T_EX engine of Donald Knuth. Special design macros were provided by ETP Services. The illustrations were prepared on a Macintosh IIci with *Mathematica* and MacDraw Pro.

Library of Congress Cataloging-in-Publication Data

**ENGINEERING MATHEMATICS
WITH MATHEMATICA**

International Series in Pure and Applied Mathematics

Ahlfors: *Complex Analysis*

Bender and Orszag: *Advanced Mathematical Methods for Scientists and Engineers*

Boas: *Invitation to Complex Analysis*

Brown and Churchill: *Fourier Series and Boundary Value Problems*

Buchanan and Turner: *Numerical Methods and Analysis*

Buck: *Advanced Calculus*

Chartrand and Oellerman: *Applied and Algorithmic Graph Theory*

Colton: *Partial Differential Equations*

Conte and de Boor: *Elementary Numerical Analysis: An Algorithmic Approach*

Edelstein-Keshet: *Mathematical Models in Biology*

Farlow: *An Introduction to Differential Equations and Their Applications*

Goldberg: *Matrix Theory with Applications*

Gulick: *Encounters with Chaos*

Hill: *Experiments in Computational Matrix Algebra*

Kurtz: *Foundations of Abstract Mathematics*

Lewin and Lewin: *An Introduction to Mathematical Analysis*

Morash: *Bridge to Abstract Mathematics: Mathematical Proof and Structures*

Parzynski and Zipse: *Introduction to Mathematical Analysis*

Pinsky: *Partial Differential Equations and Boundary-Value Problems with Applications*

Pinter: *A Book of Abstract Algebra*

Ralston and Rabinowitz: *A First Course in Numerical Analysis*

Ritger and Rose: *Differential Equations with Applications*

Robertson: *Engineering Mathematics with Mathematica*

Rudin: *Functional Analysis*

Rudin: *Principles of Mathematical Analysis*

Rudin: *Real and Complex Analysis*

Simmons: *Differential Equations with Applications and Historical Notes*

Small and Hosack: *Calculus: An Integrated Approach*

Small and Hosack: *Explorations in Calculus with a Computer Algebra System*

Vanden Eynden: *Elementary Number Theory*

Walker: *Introduction to Abstract Algebra*

ABOUT THE AUTHOR

John S. Robertson is a Professor of Applied Mathematics at the United States Military Academy. He received his Ph.D. in Mathematics from Rensselaer Polytechnic Institute. Since 1991 he has been a full professor at West Point. He is an active researcher in computational acoustics and computer applications to mathematics education. His work has been supported by the Office of Naval Research, the National Aeronautics and Space Administration, and a number of Army research agencies. He is a member of the Acoustical Society of America and the Society for Industrial and Applied Mathematics.

TO THE MEMORY OF DON ALVARO DEL PORTILLO
1914–1994
Deo omnia gloria!

PREFACE

This book is intended for use as a supplemental tool for courses in engineering mathematics, applied ordinary and partial differential equations, vector analysis, applied complex analysis, and other advanced courses in which *Mathematica*¹ is used. My goal in writing this text was to prepare a supplementary book that could be used to guide students through a series of laboratory exercises that would do the following:

- present cogent applications of the mathematics
- demonstrate the effective uses of the computational tool (in this case, *Mathematica*) to do the mathematics
- provide discussion of the results obtained by using *Mathematica*
- stimulate thought about and analysis of additional applications

Each chapter has been written so that the material it contains may be covered in a typical laboratory session of about 1-1/2 to 2 hours. The goals for every laboratory are stated at the beginning of each chapter. Mathematical concepts are then discussed within a framework of abundant engineering applications and problem-solving techniques using *Mathematica*. I have tried to keep the *Mathematica* instruction *per se* to a minimum, but have included enough material to get students up and running quickly.

Each chapter is followed by a set of exercises. Many of these are exploratory in nature and are intended to serve as a starting point for a student's mathematical experimentation. In addition, since most of the exercises can be solved

¹*Mathematica* is a registered trademark of Wolfram Research, Inc.

in more than one way, I have not provided an answer key for the student or instructor. Students should be encouraged to develop their own problem-solving skills with *Mathematica* and not just look for the “correct” answer.

Colleges and universities across the nation have been carefully re-examining the ways in which undergraduate mathematics is taught and done. The advent of computer algebra systems such as *Mathematica*, which can perform elaborate symbolic calculations, in conjunction with the rapidly expanding power of computers to function as graphic visualization devices have forced a critical re-thinking of how much of what is called higher mathematics ought to be taught and done.

The calculus reform movement has already borne much fruit. As it matures, the same style of innovative thinking must subsequently be brought to bear on virtually all the mathematics courses taken by upperclassmen. While core mathematics has been the object of substantial national attention, less emphasis has been placed on adapting advanced courses to the technology and the student expectations it brings with it.

For these reasons, one of the most promising areas in which to exploit computation is in “engineering mathematics,” a rubric which covers applied ordinary and partial differential equations, vector analysis, and applied complex analysis in courses normally taken junior and senior year. One approach to this problem is to write new textbooks with clearly-woven computational threads. I have taken a different approach by presenting discussion that is compatible with a broad range of engineering mathematics texts, as well as smaller, more specialized texts in differential equations and complex variables.

Although it might be desirable to make such a laboratory text independent of any particular software package, this goal is not yet in sight. There are simply too many differences in package front-ends and capabilities. In my view, a laboratory text must deal concretely with the details of a specific package. For a number of reasons, I have selected *Mathematica* as the computer algebra package for this text. *Mathematica* possesses a wealth of features which make it an excellent laboratory tool for engineering mathematics. In addition, *Mathematica* is available on a broad array of platforms—386 PCs, Macintoshes, Sun Sparcstations, IBM RS/6000s, etc.—and is found at many universities.

I would like to thank many, many people for their help while I prepared this book. My wife Julia alternately encouraged and cajoled me during the many months of its writing. Colonel Frank Giordano provided numerous useful insights and kept more than one wolf at bay. Dr. Norbert Carballo provided timely advice regarding the true purpose of the effort. Prof. Heidi A. Pattee of Oregon State University carefully read the manuscript and offered a myriad of suggestions which substantially improved the book. My editor, Maggie Lanzillo, was simply wonderful.

John S. Robertson

CONTENTS

Preface	xiii
1 Introduction	1
1.1 Goals	1
1.2 About <i>Mathematica</i>	1
1.3 Basic Algebra and Calculus Operations	2
1.4 Two-Dimensional Graphics	5
1.5 Three-Dimensional Graphics	7
Exercises	11
2 Vector Algebra	14
2.1 Laboratory Goals	14
2.2 Building Vectors	14
2.3 Vector Products	17
2.4 Determining Direction Cosines	18
2.5 Applications	19
Exercises	22
3 Manipulating Discrete Data	25
3.1 Laboratory Goals	25
3.2 Lists	25
3.3 Reading Lists of Data	27
3.4 Multidimensional Lists	32
Exercises	34
4 Matrices	37
4.1 Laboratory Goals	37
4.2 Products, Transposes, and Inverses	37

4.3	Powers, Eigenvalues, and Eigenvectors	40
4.4	Transition Probabilities	42
4.5	Application to Cryptography	44
	Exercises	47
5	Linear and Nonlinear Equations	49
5.1	Laboratory Goals	49
5.2	Linear Equations	49
5.3	Nonlinear Equations	52
5.4	Graphical and Numerical Solutions	54
	Exercises	56
6	First-Order ODEs	58
6.1	Laboratory Goals	58
6.2	First-Order Linear ODEs	58
6.3	Separable Equations	61
6.4	Homogeneous Equations	61
6.5	Special First-Order Equations	61
	Exercises	63
7	Second-Order Constant-Coefficient ODEs	65
7.1	Laboratory Goals	65
7.2	General Solutions	65
7.3	Using Initial Conditions	69
7.4	Damped Motion	70
7.5	Forced Motion	73
	Exercises	78
8	Laplace Transforms	79
8.1	Laboratory Goals	79
8.2	Simple Transforms	79
8.3	Application to Differential Equations	82
8.4	A Circuit Analysis Problem	83
	Exercises	86
9	The Simple Pendulum	87
9.1	Laboratory Goals	87
9.2	Model Formulation	87
9.3	A Special Case—The Inverted Pendulum	88
9.4	Periodic Oscillations of the Pendulum	90
	Exercises	91
10	Systems of ODEs	93
10.1	Laboratory Goals	93
10.2	Homogeneous Linear Systems	93
10.3	Nonhomogeneous Linear Systems	96
10.4	Chemical Mixing	99
	Exercises	102
11	Numerical Solutions of ODEs	104
11.1	Laboratory Goals	104

11.2	Second-Order Equations	104
11.3	Manipulating Multiple Solutions	106
11.4	Systems of ODEs	108
	Exercises	110
12	Variable Coefficient ODEs	112
12.1	Laboratory Goals	112
12.2	Cauchy-Euler Equations	112
12.3	Other Special Equations	114
12.4	Applications	117
	Exercises	119
13	Fourier Series	122
13.1	Laboratory Goals	122
13.2	Fourier Sine Series	122
13.3	Fourier Cosine Series	126
13.4	The <code>FourierTransform</code> Package	128
13.5	Fourier Series	129
	Exercises	133
14	The Heat Equation	135
14.1	Laboratory Goals	135
14.2	One-Dimensional Solution	135
14.3	Asymmetric Initial Temperature	136
14.4	Displaying Heat Flow Dynamics	139
14.5	Discussion	140
	Exercises	141
15	The Vibrating Bar	143
15.1	Laboratory Goals	143
15.2	Vibrating Bar	143
15.3	Separation of Variables	144
15.4	The Hinged-Hinged Bar	145
15.5	The Hinged-Clamped Bar	150
15.6	Examining Many Eigenvalues	156
	Exercises	160
16	The Vibrating Annulus	161
16.1	Laboratory Goals	161
16.2	Description of the Annulus	161
16.3	Separation of Variables Solution	162
16.4	Determination of the Eigenvalues	163
16.5	Sketching the Eigenmodes	165
	Exercises	167
17	Approximating Eigenvalues	168
17.1	Laboratory Goals	168
17.2	Motivation	168
17.3	Using Difference Equations	169
17.4	Generating the Coefficient Matrix	171

17.5	Application to Vibrating String	173
	Exercises	175
18	Vector Differentiation	177
18.1	Laboratory Goals	177
18.2	The Gradient, Divergence, and Curl	177
18.3	Other Vector Differentiation Operators	180
18.4	Establishing Vector Identities	181
	Exercises	182
19	Coordinate Systems	184
19.1	Laboratory Goals	184
19.2	Setting The Coordinate System	184
19.3	Changing Coordinates	186
19.4	Vector Multiplication in Other Systems	187
19.5	Vector Differentiation in Other Systems	188
19.6	Plotting in Other Coordinates	189
	Exercises	191
20	Vector Functions of a Single Variable	192
20.1	Laboratory Goals	192
20.2	Three-Dimensional Particle Motion	192
20.3	The <i>TNB</i> Coordinate System	195
20.4	Curvature and Torsion	196
	Exercises	198
21	Vector Integration	199
21.1	Laboratory Goals	199
21.2	The Divergence Theorem	199
21.3	Stoke's Theorem	201
	Exercises	204
22	Multi-Variable Optimization	205
22.1	Laboratory Goals	205
22.2	Description of the Trough	205
22.3	Locating Critical Points	206
22.4	The Second Derivative Test	209
22.5	Discussion	209
	Exercises	211
23	Visualizing Fields	213
23.1	Laboratory Goals	213
23.2	Gradient Fields	213
23.3	Divergence of a Vector Field	215
	Exercises	218
24	Complex Arithmetic	219
24.1	Laboratory Goals	219
24.2	Manipulating Complex Numbers	219
24.3	Complex Functions	222
	Exercises	224

25	Taylor and Laurent Series	225
25.1	Laboratory Goals	225
25.2	Taylor Series	225
25.3	Laurent Series	229
	Exercises	235
26	Residues	236
26.1	Laboratory Goals	236
26.2	Determining Residues	236
26.3	Integrating with Residues	238
26.4	Evaluating Real Integrals	239
	Exercises	240
27	Visualizing Complex Mappings	241
27.1	Laboratory Goals	241
27.2	Mapping Rectangular Regions	241
27.3	The Exponential Map	242
27.4	The Sine Map	243
27.5	The Logarithm Map	245
27.6	Other Mappings	248
	Exercises	248
28	Conformal Mappings	250
28.1	Laboratory Goals	250
28.2	Linear Fractional Transformations	250
28.3	Equilibrium Temperatures	254
28.4	Flow in a Corner	256
	Exercises	258
A	Summary of Functions and Operators	260
B	Unit Conversions and Constants	268
B.1	Physical Constants	270
B.2	Chemical Data	271
C	Enhancing Graphics	273
C.1	Two-Dimensional Plots	273
C.2	Three-Dimensional Plots	275
D	Generating FORTRAN and C Code	278
D.1	FORTRAN Example	278
D.2	C Example	279
	Bibliography	281
	Index	282

CHAPTER 1

INTRODUCTION

1.1 GOALS

- a. To become familiar with the basic syntax of *Mathematica*.
- b. To plot two-dimensional graphs.
- c. To plot three-dimensional surfaces.

1.2 ABOUT *Mathematica*

Mathematica is a powerful symbolic algebra tool that provides an extraordinarily rich variety of symbolic, graphical and numerical capabilities to anyone working with engineering mathematics.



FIGURE 1.1

Clicking on this (or a similar) icon starts *Mathematica* on the Macintosh.

Mathematica sessions are started differently on different machines. For example, on the Macintosh, you can click on a *Mathematica* icon, as shown in Fig. 1.1. This will open a *Mathematica* notebook into which you can start typing.

On a UNIX-based workstation, such as an IBM RISC System 6000, you would type `math` at the command prompt. In this case you will see the following text in your window:¹

```
Mathematica 2.0 for IBM RISC System 6000
Copyright 1988-91 Wolfram Research, Inc.
-- X11 windows graphics initialized --

In[1]:=
```

Refer to the *Local Guide* that came with your *Mathematica* implementation on starting *Mathematica* on other computer systems.²

Mathematica syntax, while initially a bit strange, is fairly easy to learn and remarkably consistent. As you work with it, keep the following rules in mind:

- *Mathematica* is case-sensitive.
- All *Mathematica* functions are capitalized.
- Function arguments are always delineated with square brackets, i.e. [...].
- Lists are always delineated with curly brackets, i.e. {...}.
- Variable ranges (for integration, plotting, and counting) are always built with lists.
- A double question mark ?? followed by the *Mathematica* function name will elicit a short help message. This is useful only if you already understand the command. There is no substitute for referring to the *Mathematica* book³ for definitive guidance on a particular function or operation.

1.3 BASIC ALGEBRA AND CALCULUS OPERATIONS

There are a few fundamental operations which must be mastered early on if the power of *Mathematica* is to be put to good use in the laboratory exercises. Consider the following expression:

¹As of this writing, notebooks are available for major UNIX platforms. With the release of *Mathematica* Version 2.2, you can also obtain a window directly to the *Mathematica* kernel on the Macintosh. Notebooks, though, are easier to manage, and their use should be *de rigueur* when available.

²In this and all subsequent chapters, input and output will be confined between the upward- and downward-facing horizontal brackets as shown in the above example. In addition, within a section, statements will always be consecutively numbered beginning with 1, e.g. `In[1]:=` and `Out[1]=`.

³Stephen Wolfram, *Mathematica: A system for doing mathematics by computer*, Addison-Wesley, Redwood City, CA, 1991.


```
In[1]:= (x + y)^2
```

```
Out[1]= (x + y)2
```

In this example, x and y are variables and the \wedge operator denotes exponentiation. *Mathematica* returns the result in so-called display form, not unlike the way we would write the result down on paper. Input lines are always numbered like `In[n]` and output lines like `Out[n]` where n is the sequential number of the line beginning with the $n = 1$ for the current *Mathematica* session.

Mathematica can be directed to expand the result with the `Expand[]` function:

```
In[2]:= Expand[%]
```

```
Out[2]= x2 + 2 x y + y2
```

The `%` symbol stands for the output of the immediately previous calculation. In this example, `%` is equivalent to `Out[1]`. This result of the last command can be further manipulated, say, by subtracting $4xy$ from it:

```
In[3]:= % - 4 x y
```

```
Out[3]= x2 - 2 x y + y2
```

Note that spaces between symbols designates *implied* multiplication. The use of the asterisk `*` makes the intention to multiply explicit, i.e., `4*x*y` is equivalent to `4 x y`.

In any case, to factor the result shown in `Out[3]` we use the `Factor[]` function:

```
In[4]:= Factor[%]
```

```
Out[4]= (-x + y)2
```