

The gathering and analysis of experimental data are fundamental activities in science and engineering. *Mathematica® in the Laboratory* is a hands-on guide which shows how to harness the power and flexibility of *Mathematica* in the control of data-acquisition equipment and the analysis of experimental data. It is fully compatible with *Mathematica* 3.0.

The book is made up of two parts. The first explains how to use *Mathematica* to import, manipulate, visualize and analyze data from existing files. The generation and export of test data are also covered. The second part deals with the control of laboratory equipment. The use of *Mathematica's* *MathLink®* system as applied to instrument control, data processing, and interfacing is clearly demonstrated.

Many practical examples are given, which can either be used directly or adapted to suit a particular application. The book sets out clearly how *Mathematica* can provide a truly unified data-handling environment, and will be invaluable to anyone who collects or analyzes experimental data, including astronomers, biologists, chemists, mathematicians, geologists, physicists, and engineers.

Cambridge University Press & Assessment  
978-0-521-58137-0 — Mathematica® in the Laboratory  
Samuel Dick, Alfred Riddle, Douglas Stein  
Frontmatter

[More Information](#)

---

*Mathematica*® in the Laboratory

---

---

# Mathematica® in the Laboratory

---

---

**Samuel Dick**

*Royal Greenwich Observatory*

**Alfred Riddle**

*Macallan Consulting, California*

**Douglas Stein**

*Addison Wesley Interactive*

Cambridge University Press & Assessment  
 978-0-521-58137-0 — Mathematica® in the Laboratory  
 Samuel Dick, Alfred Riddle, Douglas Stein  
 Frontmatter  
[More Information](#)



Shaftesbury Road, Cambridge CB2 8EA, United Kingdom  
 One Liberty Plaza, 20th Floor, New York, NY 10006, USA  
 477 Williamstown Road, Port Melbourne, VIC 3207, Australia  
 314–321, 3rd Floor, Plot 3, Splendor Forum, Jasola District Centre, New Delhi – 110025, India  
 103 Penang Road, #05–06/07, Visioncrest Commercial, Singapore 238467

Cambridge University Press is part of Cambridge University Press & Assessment, a department of the University of Cambridge.

We share the University's mission to contribute to society through the pursuit of education, learning and research at the highest international levels of excellence.

[www.cambridge.org](http://www.cambridge.org)

Information on this title: [www.cambridge.org/9780521581370](http://www.cambridge.org/9780521581370)

© Cambridge University Press & Assessment 1997

This publication is in copyright. Subject to statutory exception and to the provisions of relevant collective licensing agreements, no reproduction of any part may take place without the written permission of Cambridge University Press & Assessment.

First published 1997

*A catalogue record for this publication is available from the British Library*

*Library of Congress Cataloging-in-Publication data*

Dick, Samuel.

Mathematica® in the laboratory / Samuel Dick, Alfred Riddle,

Douglas Stein.

p. cm.

Includes bibliographical references.

ISBN 0 521 49906 2

1. Automatic data collection systems - Computer programs.

2. Mathematica (Computer file) I. Riddle, Alfred. II. Stein,

Douglas. III. Title.

TS158.6.D53 1997

502'.85'53 - dc20

96-21448

CIP

ISBN 978-0-521-58137-0 Hardback

ISBN 978-0-521-49906-4 Paperback

Mathematica®, MathLink®, and MathSource® are registered trademarks of Wolfram Research.

Cambridge University Press & Assessment has no responsibility for the persistence or accuracy of URLs for external or third-party internet websites referred to in this publication, and does not guarantee that any content on such websites is, or will remain, accurate or appropriate. Information regarding prices, travel timetables, and other factual information given in this work is correct at the time of first printing but Cambridge University Press & Assessment does not guarantee the accuracy of such information thereafter.

To Alison and my parents

*Samuel Dick*

To Dad, whose workshop was my first lab, to P. D. Evans, who taught me that a good experimentalist is a physicist at heart, and to R. J. Trew, who said, "If you'll run my lab, I'll make sure no one touches your setups."

*Alfred Riddle*

To Susan, Amanda, and Joel, who had to leave me at home on weekends to work on this book. Also to my Mom and Dad, who allowed me to fire up the soldering iron and build radios and other gizmos in my pre-teen years.

*Douglas Stein*

# Contents

	Preface	<i>page</i> xv
<b>1</b>	<b>Importing data from files</b>	1
	1.1 File operations in general	1
	1.1.1 Locating files	2
	1.2 File types	4
	1.3 Data structures	4
	1.4 Simple ASCII files	6
	1.4.1 Numbers and strings from free-format ASCII files	6
	1.4.2 String–number conversion	9
	1.4.3 Files with embedded comments	10
	1.4.4 Variably named files	13
	1.4.5 File-content catalogs	17
	1.4.6 Numbers and strings from fixed-format ASCII files	20
	1.5 ASCII files created by spreadsheets	22
	1.6 <i>Mathematica</i> files	28
	1.7 DXF files	29
	1.8 Binary files	35
	1.9 References	38
<b>2</b>	<b>Visualizing data</b>	39
	2.1 The art of visualization	39
	2.2 Standard graph-plotting functions	40
	2.3 Augmenting standard <i>Mathematica</i> graphs	42
	2.4 Data fusion	43

---

**CONTENTS**

2.5	Coping with awkward data	48
2.5.1	Axis reversal	48
2.5.2	Complex numbers	53
2.5.3	Unwrapping cyclic phase data	54
2.6	Making your own graphs	56
2.6.1	Polar surface plot	57
2.6.2	Bar charts and Manhattan plots	64
2.7	Acquisition-system problems	74
2.7.1	Glitch removal	74
2.7.2	Cyclic errors	76
2.7.3	Stuck ADC or encoder bits	78
2.8	Listening to your data	79
2.8.1	Simple waveforms	80
2.8.2	Listening to the inaudible: Using modulation	82
2.9	References	86
<b>3</b>	<b>Data analysis</b>	<b>87</b>
3.1	Statistical characteristics	87
3.1.1	Preparing data	89
3.1.2	Statistics without numbers	90
3.2	Comparing theory and data	92
3.2.1	Generating theoretical results	92
3.2.2	Plotting theory and data	94
3.2.3	Plotting theory–data differences	97
3.3	Fitting functions to data	100
3.3.1	<i>Mathematica</i> 's built-in fitting functions	100
3.3.1.1	Linear fitting	101
3.3.1.2	Nonlinear fitting	104
3.3.2	Fitting functions to complex data	108
3.3.2.1	Simple RC filter	109
3.3.2.2	LC resonator	112
3.3.3	Spline-based fitting	119
3.3.4	Filtering data to remove noise	128



---

**CONTENTS**

	3.3.4.1	Rejecting outliers	128
	3.3.4.2	Smoothing and filtering	133
	3.4	References	141
<b>4</b>		<b>Generating test data</b>	142
	4.1	Creating perfect data	142
	4.2	Creating random numbers	145
	4.2.1	Adding noise to simulated data	146
	4.2.2	Random numbers from custom distributions	152
	4.3	References	157
<b>5</b>		<b>Exporting data</b>	158
	5.1	Basic file operations	158
	5.2	<i>Mathematica</i> output to ASCII files	158
	5.2.1	Evaluated and unevaluated expressions	159
	5.2.2	Definitions	159
	5.2.3	Expressions in C/FORTRAN/TeX forms	160
	5.2.4	Formatted numbers	161
	5.3	<i>Mathematica</i> output to binary files	164
	5.3.1	Numbers	165
	5.3.2	Strings	167
	5.4	References	168
<b>6</b>		<b>Introduction to instrument control and data acquisition</b>	169
	6.1	Anatomy of an instrument data acquisition system	170
	6.1.1	Standards in instrumentation	170
	6.1.2	Designing your software	172
	6.1.3	Layerology	173
	6.1.3.1	Top (application) layer	173
	6.1.3.2	Hardware assessment layer	173
	6.1.3.3	The computing layer	174
	6.2	Coping with your data	175
	6.2.1	Data assessment	176
	6.2.2	Transmission time	176



---

**CONTENTS**

6.2.3	Compression	177
6.2.4	Binary and ASCII forms	187
6.3	Safety in laboratory systems	188
6.4	References	189
<b>7</b>	<b>Understanding MathLink</b>	<b>190</b>
7.1	The pons asinorum: <code>addtwo</code>	192
7.2	Using <code>:Evaluate:</code> to include <i>Mathematica</i> code	195
7.3	Putting and getting arguments manually	196
7.4	Passing lists and arrays	198
7.5	Passing arbitrary expressions	202
7.6	Requesting evaluations by the kernel	204
7.7	Handling and recovery of <i>MathLink</i> errors	206
7.8	Handling and recovery of other errors	208
7.9	Troubleshooting and debugging	216
7.10	Large projects	218
7.11	Special topics	220
7.11.1	Unknown result length from your program	220
7.11.1.1	Loopback links	220
7.11.2	Making your function abortable	222
7.12	References	224
<b>8</b>	<b>Interfacing I: a simple serial link</b>	<b>225</b>
8.1	The serial port hardware	226
8.1.1	Electrical connections	226
8.1.2	Serial port parameters	228
8.2	Software specification	230
8.3	The Macintosh software	230
8.3.1	Creating the <code>rs232.π</code> project	231
8.3.2	The application C code	231
8.3.3	The <i>MathLink</i> template file	237
8.3.4	Building the executable program	239
8.3.5	The <i>Mathematica</i> code	239
8.4	The PC/Linux software	242

---

**CONTENTS**

8.4.1	Port configuration	242
8.4.2	Port use	243
8.5	Problems?	246
8.5.1	Checking the hardware	246
8.5.2	Debugging the C code	247
8.5.3	Response times	248
8.6	References	248
<b>9</b>	<b>Interfacing II: more advanced links</b>	<b>249</b>
9.1	Design issues	249
9.1.1	Layers	249
9.1.2	Polling versus synchronous function calls	250
9.1.2.1	<i>Mathematica</i> code	251
9.1.2.2	Macintosh Toolbox	251
9.1.2.3	Drivers	251
9.1.2.4	Summary	251
9.1.3	Looping	252
9.1.4	Latency	252
9.2	Handling bare hardware (no support libraries)	253
9.2.1	The template file: <code>SerialDataAcquisition.tm</code>	254
9.2.2	The C code: <code>SerialDataAcquisition.c</code>	255
9.2.3	Using <code>SerialDataAcquisition</code>	260
9.3	Handling hardware with vendor libraries	261
9.3.1	One-shot oscilloscope	262
9.3.1.1	The template file: <code>OneShotScope.tm</code>	262
9.3.1.2	The C code: <code>OneShotScope.c</code>	262
9.3.1.3	Using <code>OneShotScope</code>	266
9.3.1.4	Moving on	267
9.3.2	Continuous acquisition (one channel)	268
9.3.2.1	The template file: <code>Oscilloscope.tm</code>	269
9.3.2.2	The C code: <code>Oscilloscope.c</code>	270
9.3.2.3	Using <code>Oscilloscope</code>	274
9.3.3	Streaming data to disk	276

---

**CONTENTS**

	9.3.3.1	The template file: StreamToDisk.tm	277
	9.3.3.2	The C code: StreamToDisk.c	277
	9.3.3.3	Using StreamToDisk	282
	9.3.3.4	Moving on	282
	9.3.4	Generating a waveform	283
	9.3.4.1	The template file: Generate- Waveform.tm	283
	9.3.4.2	The C code: GenerateWaveform.c	284
	9.3.4.3	Using GenerateWaveform	288
	9.3.4.4	Moving on	289
	9.3.5	Additional projects	290
	9.3.5.1	Counting LEDs	290
	9.3.5.2	Times Square (block character generator)	290
	9.3.5.3	Home heating controls	290
	9.3.5.4	Elevator controls	291
	9.3.5.5	Dream away	291
	9.4	Utility routines	291
	9.4.1	The template file: MitlMain.tm	291
	9.4.2	The C code: MitlUtil.c	293
	9.5	References	299
<b>10</b>		<b>Interface hardware design</b>	<b>300</b>
	10.1	Simple buffer circuits	301
	10.1.1	Attenuators	301
	10.1.2	Amplifiers	302
	10.1.2.1	Inverting-summing amplifier	302
	10.1.2.2	Differential amplifier	303
	10.1.2.3	Going on	304
	10.2	Noise control	304
	10.2.1	Noise in general	305
	10.2.2	Reducing true noise	306
	10.2.3	Switch debounce	307
	10.2.4	Level detection	307

---

**CONTENTS**

10.3	Signal generator	309
10.4	References	310
	<b>Appendix</b>	311
A.1	Defining your own functions: a brief summary	311
A.2	Pure anonymous functions	312
	Index	317



## Preface

*Mathematica* is a powerful tool to have in your laboratory. But what do we mean by laboratory, and what exactly is a powerful tool? The type of laboratory that is uppermost in our minds will have ongoing experiments, instruments with displays and dials, and a computer. Perhaps the laboratory has rows of bottles with colored liquids and Petri dishes. Or it might instead have growing plants, an environmental chamber for testing a new solidstate device from a silicon foundry, or a telescope collecting light from a distant galaxy. The laboratory itself might even be an entire state, a country, or an ocean, and the data might be gathered by surveying aircraft, by satellite, or by radio telemetry from remote helicopter-supported ground sites.

Whenever there is both data gathering and data processing to be done, we believe you will find *Mathematica* powerful. Why? Because *Mathematica* can provide you with a “unified” environment within which data visualizing, symbolic mathematical modeling, instrument control, data acquisition, data analysis, and report generation are all possible.

Traditionally, engineers and scientists have required different software packages for instrument control, for data acquisition, and for analysis. For example, an instrument control program would setup and acquire data from, say, a digital voltmeter which is connected by some means to a computer. The control program would place data into a file on the computer’s disk. Another program would open that file, perhaps to graphically display the data, to compare the data with computer-generated data from a theoretical model, or to execute some statistical test. In our experience, it is not uncommon for several programs (some written by us) to be used to process only one set of data – because each program is limited in its functionality. We have all spent a lot of time transferring numbers between programs, not to mention learning how to make many pieces of software do exactly what we need.

In this book we show you how you can do all these tasks – from instrument set-up to complicated analysis – by using *Mathematica*. (For instrument control, you will also need to be able to create short programs in the

## PREFACE

high-level language C. The focus of your programming work, however, will always be at a high level: the assembler manual can stay firmly on the shelf.)

We believe *Mathematica* forms an ideal platform for laboratory work. It is flexible and comprehensive enough to allow us to analyze, plot, and present our data. Its ASCII file format is an added blessing because we can always extract our data from the *Mathematica* file. The power and flexibility of the *Mathematica* language may seem daunting at first, but very quickly we find that this power keeps us from running into frustrating – and time-consuming – limitations. The only problem to date for a laboratory user is that *Mathematica* lacks data acquisition functions. In response, this book provides data acquisition solutions for *Mathematica* and gives many examples and routines for data analysis and plotting with *Mathematica*. We hope *Mathematica* saves you time, increases your insight, and helps you explain your findings as much as it does for us.

With *Mathematica*, your choice of computer is also wide open. *Mathematica* is available on a wide range of computers: IBM PCs (running DOS, Windows, OS/2, or Linux) and RISC workstations, the Apple Macintosh series (including Power Macintosh), Sun systems, DEC Alpha OSF and RISC Ultrix workstations, Hewlett-Packard workstations, and Silicon Graphics machines – to name just the most common computers. Across all of these computers, *Mathematica* has a very consistent user interface. This consistency minimizes your learning time when you move between machines, or that of a colleague when you pass your work on.

But why write a book about how to use *Mathematica* in the laboratory? *Mathematica* is a large and many-featured software system. Our aim with this book is to show you time-saving direct routes to solutions in typical scientific and engineering application areas. Of course, we encourage you to explore as much of *Mathematica* as you can. Our book can only guide you so far.

It is not possible within the scope of any one book to cover all the possibilities and all the problems that everyone will encounter. We try to provide an overview of work methods and to show in a general way how we develop those methods into solutions. That is, we concentrate on describing and discussing the foundation techniques that you need to know in order to develop your own solutions, targeted to your needs. By expanding upon our examples, you will use one of *Mathematica*'s great strengths: its customizable extensibility.

## Structure

We have written the book in two main sections. In both sections it is our aim to provide a foundation layer of techniques that you will be able first to apply



immediately and then to augment, once your familiarity with *Mathematica* grows.

In the first five chapters, we show how you can use *Mathematica* to import data from existing files, to graph data, to fit functions to data, and to generate and export test data. A fresh start is not necessary: you can gain from using *Mathematica* with your existing archival data. In the later chapters, we discuss both the task of controlling laboratory equipment and *Mathematica*'s *MathLink* system which enables you to extend *Mathematica* into the area of instrument control or to write your own data processing functions. We show how you can control instruments using a serial link and a general interface card – all driven from *Mathematica*.

Please note that it is not necessary to read the book chapter by chapter. Dip in, browse through, and skip over as you like. The material in any one chapter is not necessarily related to the material in any other chapter.

## Conventions

Because we use *Mathematica* interactively, we have preceded our instructions to *Mathematica* by *In:* (and typeset them in **bold monospace font**). *Mathematica*'s subsequent reply is preceded by *Out:* (in plain monospace font). Where *Mathematica* generates graphical output, it follows immediately after the generating input; where *Mathematica* generates sound, it is marked -Sound-. We have set examples of non-*Mathematica* code and references to file names in plain monospace font, too.

## Mathematica 2.2 and 3.0

When this book is published, 3.0 will be the newest version of *Mathematica*. We appreciate that many folks will still be using version 2.2. In general, *Mathematica* 2.2 and 3.0 both work happily with our examples. Where we have found significant differences between the *Mathematica* versions, we have added the version number as a suffix to the *In:* and *Out:* indicators described above. For example, *In 3.0:*. Where we found no significant difference, we have used version 2.2 input and output formats. We have verified our code using the last *beta* version of 3.0 – it is, however, possible that some minor differences may exist between this *beta* version and the first commercially-released version.

## Electronic access to material

We have placed the example datafiles and extracts of the *Mathematica* and C code on *MathSource*, Wolfram Research's electronic resource for *Mathemat-*



---

## PREFACE

*ica* material. You can access *MathSource* by mail from the Internet, or from the Wolfram Research website.

### Acknowledgments

The authors thank the many people who have helped us write this book. Stephen and Conrad Wolfram, Prem Chawla, and Todd Gayley at Wolfram Research are thanked for useful conversations and for answering our puzzles.

The team at Cambridge University Press, especially David Tranah in Cambridge and Philip Meyler in New York, are thanked for their enthusiasm, their encouragement and support, and their patience.

DBS wishes to acknowledge Shawn Sheridan of Wolfram Research for many helpful discussions over the years on the design of *MathLink* and the proper design of *MathLink*-based applications. Without his painstaking work, the C-language interfaces for *MathLink* would not be the elegant and useful tools that they are. DBS also wishes to thank and acknowledge Rob Dye and Ed McConnell of National Instruments, who graciously arranged for the loan of the LAB-LC board and copy of the NI-DAQ libraries used in the examples of Chapter 9.

Samuel Dick  
Alfred Riddle  
Douglas Stein  
1996