

Programming with **MATLAB**[®] for Scientists

A Beginner's Introduction

Eugeniy E. Mikhailov



CRC Press
Taylor & Francis Group

Programming with **MATLAB**[®] for Scientists *A Beginner's Introduction*

This book offers an introduction to the basics of MATLAB[®] programming to scientists and engineers. The author leads with engaging examples to build a working knowledge, specifically geared to those with science and engineering backgrounds. The reader is empowered to model and simulate real systems, as well as present and analyze everyday data sets. In order to achieve those goals, the contents bypass excessive “under the hood” details, and instead gets right down to the essential, practical foundations for successful programming and modeling. Readers will benefit from the following features:

- Teaches programming to scientists and engineers using a problem-based approach, leading with illustrative and interesting examples
- Emphasizes a hands-on approach, with “must know” information and minimal technical details
- Utilizes examples from science and engineering to showcase the application of learned concepts on real problems
- Showcases modeling of real systems, gradually advancing from simpler to more challenging problems
- Highlights the practical uses of data processing and analysis in everyday life



CRC Press
Taylor & Francis Group
an **informa** business

www.crcpress.com

CRC Press titles are available as eBook editions in a range of digital formats

ISBN: 978-1-138-57004-7



9 781138 570047

Mikhailov

Progress with MATLAB[®] for Scientists

CRCPress

Programming with MATLAB for Scientists

A Beginner's Introduction

Eugeniy E. Mikhailov



CRC Press

Taylor & Francis Group

Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

MATLAB® and Simulink® are trademarks of the MathWorks, Inc. and are used with permission. The MathWorks does not warrant the accuracy of the text or exercises in this book. This book's use or discussion of MATLAB® and Simulink® software or related products does not constitute endorsement or sponsorship by the MathWorks of a particular pedagogical approach or particular use of the MATLAB® and Simulink® software.

Published in 2017 by CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2017 by Taylor & Francis Group, LLC

CRC Press is an imprint of Taylor & Francis Group

No claim to original U.S. Government works

Printed in the United States of America on acid-free paper

10 9 8 7 6 5 4 3 2 1

International Standard Book Number-13: 978-1-138-57004-7 (hardback)

This book contains information obtained from authentic and highly regarded sources. Reprinted material is quoted with permission, and sources are indicated. A wide variety of references are listed. Reasonable efforts have been made to publish reliable data and information, but the author and the publisher cannot assume responsibility for the validity of all materials or for the consequences of their use.

No part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (<http://www.copyright.com/>) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data

Names: Mikhailov, Eugeny E., 1975- author.
Title: Programming with MATLAB for scientists : a beginner's introduction / Eugeny E. Mikhailov.
Description: Boca Raton, FL : CRC Press, Taylor & Francis Group, [2017]
Identifiers: LCCN 2017031570 | ISBN 9781498738286 | ISBN 1498738281
Subjects: LCSH: MATLAB. | Science--Data processing. | Engineering mathematics--Data processing. | Numerical analysis--Data processing.
Classification: LCC Q183.9.M55 2017 | DDC 510.285/53--dc23
LC record available at <https://lcn.loc.gov/2017031570>

Visit the Taylor & Francis Web site at
<http://www.taylorandfrancis.com>

and the CRC Press Web site at
<http://www.crcpress.com>

Programming with MATLAB for Scientists

A Beginner's Introduction

Preface

Intended Audience

This book is intended for anyone who wants to learn how to program with MATLAB and seeks a concise and accessible introduction to programming, MATLAB, and numerical methods. The author hopes that readers will find here all necessary materials for handling their everyday computational and programming problems. Even more seasoned readers may find useful insights on familiar methods or explanations for puzzling issues they might encounter.

We will start with simple concepts and build up a skill set suitable to model, simulate, and analyze real-life systems. Additionally, this book provides a broad overview of the numerical methods necessary for successful scientific or engineering work. We will get familiar with a “lore” of computing, so you will know what to look for when you decide to move to more advanced techniques.

The book is based on material of the one semester “Practical Computing for Scientists” class taught at the College of William & Mary for students who have not yet declared a major or students majoring in physics, neuroscience, biology, computer science, applied math and statistics, or chemistry. The students who successfully took this class were at all levels of their academic careers; some were freshmen, some were seniors, and some were somewhere in between.

Why MATLAB?

A couple words about MATLAB, as it is our programming language of choice. MATLAB has a good balance of already implemented features, which are important for scientists and for ease of learning. MATLAB hides a lot of low-level details from users: you do not need to think about variable types, compilation processes, and so on. It just works. You can also do a calculation on a whole array of data without tracking every element of the array. This part is deep inside of MATLAB.

From an instructor’s point of view, you do not need to worry about the installation of MATLAB for your class. It is easy and students are capable of doing it alone. More importantly, it looks and works the same on variety of operational systems, such as Windows, Mac, and Linux. MATLAB produces exactly the same result on all computers.

From a student’s point of view, MATLAB is probably the most frequently required programming language for an engineering or scientific position. Therefore, if you learn MATLAB now, you likely will not need to retrain yourself to another industry standard programming language.

MATLAB has a downside: it is expensive to purchase if your school or workplace does not provide it. This is not a big worry; you can do exercises from all the chapters, except the data fitting, with a free alternative: GNU Octave. The fitting in Octave uses a different set of commands, but everything else will work the same (you might need minor tweaking for more advanced options).

What Is not Covered in This Book?

This book does not extensively cover MATLAB's commands. There is no reason to write another manual for MATLAB (which has an excellent one already) or redo tutorials available on the web.

This book is also not a substitute for a book explaining the ins and outs of numerical methods. Whenever possible, we discuss what interesting things can be done with a numerical method and do not bother with the most efficient implementation. However, the beginning of the book is an exception. There, the basics of programming are explained via implementations of some numerical method algorithms (which often have MATLAB's built-in equivalents already).

How to Read This Book

If you have programmed before, you can skip the majority of Part I, but make sure that you are fluent with the element operations described there, the differences between array and element-wise operations, and array slicing.

If you are scientist, then the plotting and fitting materials are a must. Make sure that you have read the fitting chapter. If you need to learn anything important about data analysis, then learn this.

The material in Part III is somewhat optional, although the author strongly recommends the optimization problem chapter (see Chapter 13). It is amazing to see how many other problems are essentially optimization problems and can be solved with methods presented there. The time to use the material in this section will probably come in upper undergraduate classes.

As you get more and more fluent with programming, reread the good programming practice materials in Section 4.4 and try to implement more and more techniques from there.

Data files and MATLAB listings locations

All MATLAB's listing and required data files used in the book are available on the web.* The PDF version of this book contains direct links to such files.

MATLAB[®] is a registered trademark of The MathWorks, Inc.

For product information, please contact:

The MathWorks, Inc.

3 Apple Hill Drive Natick,

MA 01760-2098 USA

Tel: 508 647 7000 Fax: 508-647-7001

E-mail: info@mathworks.com

Web: www.mathworks.com

* Please visit http://physics.wm.edu/programming_with_MATLAB_book.

Contents

<i>Preface</i>	<i>xi</i>
I Computing Essentials	1
1 Computers and Programming Languages: An Introduction	3
1.1 Early History of Computing	3
1.2 Modern Computers	4
1.2.1 Common features of a modern computer	4
1.3 What Is Programming?	5
1.4 Programming Languages Overview	5
1.5 Numbers Representation in Computers and Its Potential Problems	7
1.5.1 Discretization—the main weakness of computers	7
1.5.2 Binary representation	8
1.5.3 Floating-point number representation	8
1.5.4 Conclusion	9
1.6 Self-Study	10
2 MATLAB Basics	11
2.1 MATLAB's Graphical User Interface	11
2.2 MATLAB as a Powerful Calculator	14
2.2.1 MATLAB's variable types	14
2.2.2 Some built-in functions and operators	15
2.2.3 Operator precedence	16
2.2.4 Comments	17
2.3 Efficient Editing	17
2.4 Using Documentation	18
2.5 Matrices	19
2.5.1 Creating and accessing matrix elements	19
2.5.2 Native matrix operations	21
2.5.3 Strings as matrices	24
2.6 Colon (:) Operator	25
2.6.1 Slicing matrices	25
2.7 Plotting	26
2.7.1 Saving plots to files	28
2.8 Self-Study	29

3	Boolean Algebra, Conditional Statements, Loops	31
3.1	Boolean Algebra	31
3.1.1	Boolean operators precedence in MATLAB	32
3.1.2	MATLAB Boolean logic examples	32
3.2	Comparison Operators	33
3.2.1	Comparison with vectors	34
3.2.2	Comparison with matrices	34
3.3	Conditional Statements	35
3.3.1	The if-else-end statement	35
3.3.2	Short form of the “if” statement	36
3.4	Common Mistake with the Equality Statement	36
3.5	Loops	36
3.5.1	The “while” loop	36
3.5.2	Special commands “break” and “continue”	38
3.5.3	The “for” loop	38
3.6	Self-Study	40
4	Functions, Scripts, and Good Programming Practice	41
4.1	Motivational Examples	41
4.1.1	Bank interest rate problem	41
4.1.2	Time of flight problem	42
4.2	Scripts	43
4.2.1	Quadratic equation solver script	43
4.3	Functions	45
4.3.1	Quadratic equation solver function	46
4.4	Good Programming Practice	47
4.4.1	Simplify the code	47
4.4.2	Try to foresee unexpected behavior	48
4.4.3	Run test cases	48
4.4.4	Check and sanitize input arguments	49
4.4.5	Is the solution realistic?	50
4.4.6	Summary of good programming practice	51
4.5	Recursive and Anonymous Functions	51
4.5.1	Recursive functions	51
4.5.2	Anonymous functions	52
4.6	Self-Study	54
II	Solving Everyday Problems with MATLAB	57
5	Solving Systems of Linear Algebraic Equations	59
5.1	The Mobile Problem	59
5.2	Built-In MATLAB Solvers	61
5.2.1	The inverse matrix method	61

5.2.2	Solution without inverse matrix calculation	62
5.2.3	Which method to use	62
5.3	Solution of the Mobile Problem with MATLAB.....	63
5.3.1	Solution check	64
5.4	Example: Wheatstone Bridge Problem	65
5.5	Self-Study	66
6	Fitting and Data Reduction	69
6.1	Necessity for Data Reduction and Fitting	69
6.2	Formal Definition for Fitting.....	70
6.2.1	Goodness of the fit	70
6.3	Fitting Example	71
6.4	Parameter Uncertainty Estimations	73
6.5	Evaluation of the Resulting Fit	74
6.6	How to Find the Optimal Fit.....	75
6.6.1	Example: Light diffraction on a single slit	77
6.6.2	Plotting the data	77
6.6.3	Choosing the fit model.....	78
6.6.4	Making an initial guess for the fit parameters	79
6.6.5	Plotting data and the model based on the initial guess	80
6.6.6	Fitting the data	81
6.6.7	Evaluating uncertainties for the fit parameters	82
6.7	Self-Study	84
7	Numerical Derivatives	87
7.1	Estimate of the Derivative via the Forward Difference	87
7.2	Algorithmic Error Estimate for Numerical Derivative.....	88
7.3	Estimate of the Derivative via the Central Difference	89
7.4	Self-Study	91
8	Root Finding Algorithms	93
8.1	Root Finding Problem.....	93
8.2	Trial and Error Method.....	94
8.3	Bisection Method	95
8.3.1	Bisection use example and test case	97
8.3.2	Possible improvement of the bisection code	100
8.4	Algorithm Convergence.....	100
8.5	False Position (<i>Regula Falsi</i>) Method	101
8.6	Secant Method	102
8.7	Newton–Raphson Method	103
8.7.1	Using Newton–Raphson algorithm with the analytical derivative	105
8.7.2	Using Newton–Raphson algorithm with the numerical derivative	106

8.8	Ridders' Method	106
8.9	Root Finding Algorithms Gotchas	108
8.10	Root Finding Algorithms Summary	109
8.11	MATLAB's Root Finding Built-in Command	110
8.12	Self-Study	110
9	Numerical Integration Methods	113
9.1	Integration Problem Statement	113
9.2	The Rectangle Method	114
9.2.1	Rectangle method algorithmic error	116
9.3	Trapezoidal Method	116
9.3.1	Trapezoidal method algorithmic error	117
9.4	Simpson's Method	118
9.4.1	Simpson's method algorithmic error	118
9.5	Generalized Formula for Integration	119
9.6	Monte Carlo Integration	119
9.6.1	Toy example: finding the area of a pond	119
9.6.2	Naive Monte Carlo integration	120
9.6.3	Monte Carlo integration derived	120
9.6.4	The Monte Carlo method algorithmic error	121
9.7	Multidimensional Integration	122
9.7.1	Minimal example for integration in two dimensions	122
9.8	Multidimensional Integration with Monte Carlo	123
9.8.1	Monte Carlo method demonstration	124
9.9	Numerical Integration Gotchas	124
9.9.1	Using a very large number of points	124
9.9.2	Using too few points	124
9.10	MATLAB Functions for Integration	125
9.11	Self-Study	126
10	Data Interpolation	129
10.1	The Nearest Neighbor Interpolation	129
10.2	Linear Interpolation	130
10.3	Polynomial Interpolation	132
10.4	Criteria for a Good Interpolation Routine	134
10.5	Cubic Spline Interpolation	134
10.6	MATLAB Built-In Interpolation Methods	135
10.7	Extrapolation	136
10.8	Unconventional Use of Interpolation	136
10.8.1	Finding the location of the data crossing $y = 0$	136
10.9	Self-Study	138

III	Going Deeper and Expanding the Scientist’s Toolbox	139
11	Random Number Generators and Random Processes	141
11.1	Statistics and Probability Introduction	141
11.1.1	Discrete event probability	141
11.1.2	Probability density function	142
11.2	Uniform Random Distribution	142
11.3	Random Number Generators and Computers	143
11.3.1	Linear congruential generator	144
11.3.2	Random number generator period	145
11.4	How to Check a Random Generator	145
11.4.1	Simple RNG test with Monte Carlo integration	146
11.5	MATLAB’s Built-In RNGs	148
11.6	Self-Study	148
12	Monte Carlo Simulations	149
12.1	Peg Board	149
12.2	Coin Flipping Game	151
12.3	One-Dimensional Infection Spread	152
12.4	Self-Study	160
13	The Optimization Problem	161
13.1	Introduction to Optimization	161
13.2	One-Dimensional Optimization	162
13.2.1	The golden section optimum search algorithm	163
13.2.2	MATLAB’s built-in function for the one-dimension optimization	165
13.2.3	One-dimensional optimization examples	165
13.3	Multidimensional Optimization	167
13.3.1	Examples of multidimensional optimization	168
13.4	Combinatorial Optimization	174
13.4.1	Backpack problem	174
13.4.2	Traveling salesman problem	178
13.5	Simulated Annealing Algorithm	183
13.5.1	The backpack problem solution with the annealing algorithm	185
13.6	Genetic Algorithm	192
13.7	Self-Study	193
14	Ordinary Differential Equations	195
14.1	Introduction to Ordinary Differential Equation	195
14.2	Boundary Conditions	197
14.3	Numerical Method to Solve ODEs	197
14.3.1	Euler’s method	197
14.3.2	The second-order Runge–Kutta method (RK2)	199

14.3.3	The fourth-order Runge–Kutta method (RK4)	200
14.3.4	Other numerical solvers	200
14.4	Stiff ODEs and Stability Issues of the Numerical Solution	201
14.5	MATLAB's Built-In ODE Solvers	203
14.6	ODE Examples	203
14.6.1	Free fall example	203
14.6.2	Motion with the air drag	206
14.7	Self-Study	212
15	Discrete Fourier-Transform	213
15.1	Fourier Series	213
15.1.1	Example: Fourier series for $ t $	215
15.1.2	Example: Fourier series for the step function	217
15.1.3	Complex Fourier series representation	218
15.1.4	Non-periodic functions	219
15.2	Discrete Fourier Transform (DFT)	219
15.3	MATLAB's DFT Implementation and Fast Fourier Transform (FFT)	221
15.4	Compact Mathematical Notation for Fourier Transforms	222
15.5	DFT Example	222
15.6	Self-Study	226
16	Digital Filters	229
16.1	Nyquist Frequency and the Minimal Sampling Rate	229
16.1.1	Under-sampling and aliasing	230
16.2	DFT Filters	232
16.2.1	Low-pass filter	233
16.2.2	High-pass filter	234
16.2.3	Band-pass and band-stop filters	235
16.3	Filter's Artifacts	237
16.4	Windowing Artifacts	239
16.5	Self-Study	241
	References	243
	Index	245