

An Introduction to Programming and Numerical Methods in MATLAB

S.R. Otto and J.P. Denier

An Introduction to Programming and Numerical Methods in MATLAB

With 111 Figures

 Springer

S.R. Otto, BSc, PhD

The R & A
St Andrews
Fife
KY16 9JD
Scotland

J.P. Denier, BSc (Hons), PhD

School of Mathematical Sciences
The University of Adelaide
South Australia 5005
Australia

British Library Cataloguing in Publication Data

Otto, S. R. (Stephen Robert)

An introduction to programming and numerical methods in
MATLAB

1. MATLAB (Computer file) 2. Numerical analysis — Data
processing

I. Title II. Denier, J. P.

518'.02855

ISBN 1852339195

Library of Congress Control Number: 2005923332

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licences issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.

ISBN-10: 1-85233-919-5

ISBN-13: 978-185233-919-7

Springer Science+Business Media
springeronline.com

© Springer-Verlag London Limited 2005

The use of registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

Typesetting: Camera-ready by authors

Printed in the United States of America

12/3830-543210 Printed on acid-free paper SPIN 11317333

*For
Julie
and
Jill and Megan.*

Preface

This text provides an introduction to the numerical methods that are typically encountered (and used) in science and engineering undergraduate courses. The material is developed in tandem with MATLAB which allows rapid prototyping and testing of the methods. The package MATLAB (matrix laboratory) provides an environment in which students can learn to programme and explore the structure of the numerical methods. The methods included here are of a basic nature and only rely on material which should have been explored prior to the first year undergraduate stage.

The methods presented are supplemented with a set of tasks at the end of each chapter (full solutions of these are given in Appendix C). The tasks are introduced in such a way as to allow students to explore the topics as they evolve. Some are of a mathematical nature, but in the main they involve manipulating codes which are given in the text of the chapter (or section). Those tasks which we regard as being harder are marked with an asterisk. Throughout the text MATLAB commands appear using **this font** in the text. In the main the names of MATLAB commands reflect the commands rôle. We have taken particular care to highlight many of the problems that occur with interpreting the syntax of MATLAB commands. In Appendix B we provide a glossary of all MATLAB commands within the text and short examples of how these commands can be used. Reference is made to the comprehensive help facility within MATLAB: however examples are given which are similar to those given in other places within the text.

Throughout the text we derive the numerical techniques we use, but also emphasise that MATLAB's rich vocabulary provides commands for performing most of the fundamental tasks encountered in numerical methods. This approach serves to introduce students to the methods and also provides an

understanding of their inner workings.

Why Do We Need Numerical Methods?

Mathematics is an elegant and precise subject: however when numerical answers are required one sometimes needs to rely on approximate methods to obtain useable answers. There are many problems which simply do not have analytical solutions, or those whose exact solution is beyond our current state of knowledge. There are also many problems which are too long (or tedious) to solve by hand. When such problems arise we can exploit numerical analysis to reduce the problem to one involving a finite number of unknowns and use a computer to solve the resulting equations.

The text starts with a description of how we could perform some very basic calculations (that is, simply using the computer as a calculator). It then moves on to solving problems which cannot, in practice, be solved by hand. Sometimes the solution of these problems can become as intricate and involved as the original problems and requires almost as much finesse and care to obtain a solution. There are several options available to us, both in terms of language and also overall approach. In this book we elect to express our ideas in terms of the syntax of the computer package MATLAB. Once you have mastered the syntax of MATLAB it will be easier for you to learn other languages, if you should decide you need to.

The Structure of This Text

This text is designed to be used as a source of reference for MATLAB commands (mainly through the glossary in Appendix B) and the examples given in the chapters. This is an approach we have found works well with our students. The text gives an introduction to numerical methods and the manipulation of the quantities used therein (for instance matrices). At each stage, short codes are given to allow the reader to try their own examples.

Examples of code which can be typed at the MATLAB prompt will appear within the general text in **this font** (this font is also used for the names of MATLAB codes), while longer examples will be written

```
>> commands to be entered
```

Results

Those codes which are designed to be saved to a file will appear in boxes

```
a = 1;
```

(sometimes wider boxes will be used for codes with longer lines).

At the end of each section, or chapter, there are a variety of tasks which are designed to help the reader understand the topics they have just read. Solutions of these are given in Appendix C.

Chapter 2 concentrates on some aspects of programming. We also introduce another of MATLAB's powerful tools, namely its ability to produce high quality plots of data. Students are shown how to create codes and functions, which serve to augment MATLAB's built-in vocabulary. This chapter concludes with a discussion of the important topic of errors, both from the perspective of classical numerical analysis and also from human interaction which arise in the action of coding. We have found that students benefit enormously from debugging programmes and it is hoped that this will help them to hone these skills, which we consider to be essential.

The third chapter contains a discussion of looping and logical structures within MATLAB. Again the mathematics is developed in tandem. Finally concluding the first part of the book we give some examples of how MATLAB can be used to solve problems (using just algebra and exploiting MATLAB as an advanced calculator).

In Chapters 4 and 5 we meet some classical numerical methods, in the form of root finding and interpolation (and extrapolation). Although MATLAB has intrinsic functions which will perform most of these operations (`fzero` and `polyfit`) we have included a detailed description of both topics. This is aimed at helping the students to understand how these methods work, and where they can potentially fail. In Chapter 5 we discuss the general form of Newton forward differences, which are used in this chapter and subsequent ones (for instance to derive the formula for integration). In the next two chapters we explore the numerical analysis associated with integration and differentiation.

In Chapter 6 we return to the topic of matrices. We start by discussing the mathematical concepts of rank and linear independence. We also discuss eigenvalues and eigenvectors (and their interpretation for 2-by-2 matrices). The topic of numerical integration is taken up in Chapter 7 and in Chapter 8 methods for the numerical solution of ordinary differential equations are explored. Finally in Chapter 9 we use MATLAB to calculate some basic statistical quantities and also to explore some maps, some of which may exhibit chaotic behaviour.

The text finishes with appendices containing an introduction to the basics

of matrix algebra, a glossary of useful terms and solutions to all the tasks contained within the text. These appendices have deliberately been made quite wordy, since we believe that the material they contain is one of the important aspects of the book.

At the outset we expect readers to be mathematically literate to the level of being able to: and solve algebraic equations (for instance determine the roots of a quadratic and solve simultaneous equations); integrate and differentiate simple functions; solve separable differential equations (although examples are given within the text). Although exposure to complex numbers and matrices would be useful at the start of the text, it is not necessary as these concepts are introduced in some detail in the introductory chapter.

We would like to thank students of the University of Birmingham whose patience and comments have been invaluable in the construction of this text. In particular we would like to identify Sukhjinder Chana and Rob Ackeroyd for their careful proof reading and comments.

St Andrews, UK
Adelaide, Australia

S. R. Otto
J. P. Denier

Contents

1. Simple Calculations with MATLAB	1
1.1 Introduction and a Word of Warning	1
1.2 Scalar Quantities and Variables	2
1.2.1 Rules for Naming of Variables	4
1.2.2 Precedence: The Order in Which Calculations Are Per- formed	5
1.2.3 Mathematical Functions	8
1.3 Format: The Way in Which Numbers Appear	12
1.4 Vectors in MATLAB	13
1.4.1 Initialising Vector Objects	13
1.4.2 Manipulating Vectors and Dot Arithmetic	14
1.5 Setting Up Mathematical Functions	17
1.6 Some MATLAB Specific Commands	20
1.6.1 Looking at Variables and Their Sizes	22
1.7 Accessing Elements of Arrays	23
1.8 Tasks	24
2. Writing Scripts and Functions	27
2.1 Creating Scripts and Functions	27
2.1.1 Functions	30
2.1.2 Brief Aside	35
2.2 Plotting Simple Functions	36
2.2.1 Evaluating Polynomials and Plotting Curves	41
2.2.2 More on Plotting	44
2.3 Functions of Functions	49
2.4 Errors	51

2.4.1	Numerical Errors	51
2.4.2	User Error	54
2.5	Tasks	57
3.	Loops and Conditional Statements	63
3.1	Introduction	63
3.2	Loops Structures	63
3.3	Summing Series	68
3.3.1	Sums of Series of the Form $\sum_{j=1}^N j^p, p \in \mathbb{N}$	73
3.3.2	Summing Infinite Series	76
3.3.3	Summing Series Using MATLAB Specific Commands ...	79
3.3.4	Loops Within Loops (Nested)	82
3.4	Conditional Statements	83
3.4.1	Constructing Logical Statements	85
3.4.2	The MATLAB Command <code>switch</code>	88
3.5	Conditional loops	90
3.5.1	The <code>break</code> Command	92
3.6	MATLAB Specific Commands	92
3.7	Error Checking	94
3.8	Tasks	97
4.	Root Finding	103
4.1	Introduction	103
4.2	Initial Estimates	104
4.3	Fixed Point Iteration	109
4.4	Bisection	113
4.5	Newton–Raphson and Secant Methods	117
4.5.1	Derivation of the Newton–Raphson Method	117
4.6	Repeated Roots of Functions	123
4.7	Zeros of Higher-Dimensional Functions(*)	125
4.8	MATLAB Routines for Finding Zeros	128
4.8.1	Roots of a Polynomial	128
4.8.2	The Command <code>fzero</code>	128
4.9	Tasks	130
5.	Interpolation and Extrapolation	133
5.1	Introduction	133
5.2	Saving and Reading Data	134
5.3	Which Points to Use?	139
5.4	Newton Forward Differences and Lagrange Polynomials	141
5.4.1	Linear Interpolation/Extrapolation	147

5.5	Calculating Interpolated and Extrapolated Values	148
5.6	Splines	150
5.7	Curves of Best Fit	152
5.8	Interpolation of Non-Smooth Data	155
5.8.1	Insufficient Data Points	158
5.9	Minimisation of Functions and Parameter Retrieval	161
5.9.1	Parameter Retrieval	163
5.9.2	Using <code>fmins</code> for Parameter Retrieval	164
5.10	Tasks	166
6.	Matrices	169
6.1	Introduction	169
6.1.1	Initialising Matrices Within MATLAB	169
6.1.2	Matrix Operations	174
6.1.3	Operations on Elements of Matrices	180
6.1.4	More on Special Matrices	182
6.1.5	Matrices Containing Strings	185
6.2	Properties of Matrices and Systems of Equations	186
6.2.1	Determinants of Matrices	190
6.3	Elementary Row Operations	191
6.3.1	Solving Many Equations at Once	198
6.4	Matrix Decomposition	199
6.5	Eigenvalues and Eigenvectors	204
6.6	Specific MATLAB Commands	208
6.7	Characteristic Polynomials	212
6.8	Exponentials of Matrices	214
6.9	Tasks	217
7.	Numerical Integration	225
7.1	Introduction	225
7.2	Integration Using Straight Lines	226
7.2.1	Errors in the Trapezium Method	229
7.3	Integration Using Quadratics	230
7.4	Integration Using Cubic Polynomials	235
7.5	Integrating Using MATLAB Commands	237
7.6	Specific Examples of Integrals	238
7.6.1	Infinite Integrals and Removable Singularities	238
7.6.2	Indefinite Integrals	240
7.7	Tasks	242

8. Solving Differential Equations	247
8.1 Introduction	247
8.2 Euler's Method and Crank–Nicolson	247
8.2.1 Analytical Comparisons	253
8.3 Banded Matrices	259
8.4 Runge–Kutta Methods	263
8.5 Higher-Order Systems	266
8.5.1 Second-Order Systems	266
8.5.2 Bessel's equation	270
8.5.3 Airy's Equation	273
8.5.4 Shooting Methods	274
8.6 Boundary-Value Problems	278
8.7 Population Dynamics	282
8.8 Eigenvalues of Differential Systems	285
8.9 Tasks	286
9. Simulations and Random Numbers	291
9.1 Introduction	291
9.2 Statistical quantities	291
9.2.1 Averages	291
9.2.2 Other Statistical Measures	293
9.3 Random Numbers and Distributions	295
9.3.1 Normal Distribution	298
9.3.2 Calculating Probabilities	299
9.3.3 Permutations	300
9.4 Maps and White Noise	300
9.4.1 Modelling Discrete Systems	307
9.4.2 Periodicity and Chaos	309
9.4.3 Random Motion	316
9.5 Tasks	319
A. A Mathematical Introduction to Matrices	323
A.1 Special Matrices	330
A.2 Inverses of Matrices	331
B. Glossary of Useful Terms	335
B.1 Arithmetic and Logical Operators	335
B.2 Symbols	343
B.3 Plotting Commands	347
B.4 General MATLAB Commands	365

C. Solutions to Tasks	389
C.1 Solutions for Tasks from Chapter 1	389
C.2 Solutions for Tasks from Chapter 2	394
C.3 Solutions for Tasks from Chapter 3	400
C.4 Solutions for Tasks from Chapter 4	408
C.5 Solutions for Tasks from Chapter 5	414
C.6 Solutions for Tasks from Chapter 6	418
C.7 Solutions for Tasks from Chapter 7	429
C.8 Solutions for Tasks from Chapter 8	436
C.9 Solutions for Tasks from Chapter 9	454
Index	459