
Contents

1	The First Few Steps	1
1.1	What Is a Program? And What Is Programming?	1
1.2	A Matlab Program with Variables	3
1.2.1	The Program	3
1.2.2	Dissection of the Program	4
1.2.3	Why Not Just Use a Pocket Calculator?	5
1.2.4	Why You Must Use a Text Editor to Write Programs	6
1.2.5	Write and Run Your First Program	6
1.3	A Matlab Program with a Library Function	7
1.4	A Matlab Program with Vectorization and Plotting	8
1.5	More Basic Concepts	10
1.5.1	Using Matlab Interactively	10
1.5.2	Arithmetics, Parentheses and Rounding Errors	11
1.5.3	Variables	11
1.5.4	Formatting Text and Numbers	12
1.5.5	Arrays	14
1.5.6	Plotting	15
1.5.7	Error Messages and Warnings	18
1.5.8	Input Data	19
1.5.9	Symbolic Computations	19
1.5.10	Concluding Remarks	21
1.6	Exercises	22
2	Basic Constructions	25
2.1	If Tests	25
2.2	Functions	27
2.3	For Loops	32
2.4	While Loops	35
2.5	Reading from and Writing to Files	36
2.6	Exercises	38
3	Computing Integrals	47
3.1	Basic Ideas of Numerical Integration	48
3.2	The Composite Trapezoidal Rule	49

3.2.1	The General Formula	51
3.2.2	Implementation	52
3.2.3	Alternative Flat Special-Purpose Implementation	54
3.3	The Composite Midpoint Method	57
3.3.1	The General Formula	58
3.3.2	Implementation	58
3.3.3	Comparing the Trapezoidal and the Midpoint Methods	59
3.4	Testing	60
3.4.1	Problems with Brief Testing Procedures	60
3.4.2	Proper Test Procedures	61
3.4.3	Finite Precision of Floating-Point Numbers	62
3.4.4	Constructing Unit Tests and Writing Test Functions	64
3.5	Vectorization	67
3.6	Measuring Computational Speed	69
3.7	Double and Triple Integrals	69
3.7.1	The Midpoint Rule for a Double Integral	69
3.7.2	The Midpoint Rule for a Triple Integral	73
3.7.3	Monte Carlo Integration for Complex-Shaped Domains	76
3.8	Exercises	80
4	Solving Ordinary Differential Equations	87
4.1	Population Growth	88
4.1.1	Derivation of the Model	89
4.1.2	Numerical Solution	91
4.1.3	Programming the Forward Euler Scheme; the Special Case	94
4.1.4	Understanding the Forward Euler Method	97
4.1.5	Programming the Forward Euler Scheme; the General Case	97
4.1.6	Making the Population Growth Model More Realistic	98
4.1.7	Verification: Exact Linear Solution of the Discrete Equations	101
4.2	Spreading of Diseases	102
4.2.1	Spreading of a Flu	102
4.2.2	A Forward Euler Method for the Differential Equation System	105
4.2.3	Programming the Numerical Method; the Special Case	105
4.2.4	Outbreak or Not	106
4.2.5	Abstract Problem and Notation	108
4.2.6	Programming the Numerical Method; the General Case	109
4.2.7	Time-Restricted Immunity	111
4.2.8	Incorporating Vaccination	111
4.2.9	Discontinuous Coefficients: a Vaccination Campaign	114
4.3	Oscillating One-Dimensional Systems	115
4.3.1	Derivation of a Simple Model	115
4.3.2	Numerical Solution	117
4.3.3	Programming the Numerical Method; the Special Case	117
4.3.4	A Magic Fix of the Numerical Method	120
4.3.5	The 2nd-Order Runge-Kutta Method (or Heun's Method)	122
4.3.6	Software for Solving ODEs	123
4.3.7	The 4th-Order Runge-Kutta Method	130

4.3.8	More Effects: Damping, Nonlinearity, and External Forces	133
4.3.9	Illustration of Linear Damping	136
4.3.10	Illustration of Linear Damping with Sinusoidal Excitation	137
4.3.11	Spring-Mass System with Sliding Friction	138
4.3.12	A Finite Difference Method; Undamped, Linear Case	141
4.3.13	A Finite Difference Method; Linear Damping	143
4.4	Exercises	144
5	Solving Partial Differential Equations	153
5.1	Finite Difference Methods	155
5.1.1	Reduction of a PDE to a System of ODEs	156
5.1.2	Construction of a Test Problem with Known Discrete Solution	158
5.1.3	Implementation: Forward Euler Method	158
5.1.4	Application: Heat Conduction in a Rod	160
5.1.5	Vectorization	165
5.1.6	Using Odespy to Solve the System of ODEs	165
5.1.7	Implicit Methods	166
5.2	Exercises	169
6	Solving Nonlinear Algebraic Equations	177
6.1	Brute Force Methods	178
6.1.1	Brute Force Root Finding	179
6.1.2	Brute Force Optimization	181
6.1.3	Model Problem for Algebraic Equations	182
6.2	Newton’s Method	183
6.2.1	Deriving and Implementing Newton’s Method	183
6.2.2	Making a More Efficient and Robust Implementation	186
6.3	The Secant Method	189
6.4	The Bisection Method	191
6.5	Rate of Convergence	193
6.6	Solving Multiple Nonlinear Algebraic Equations	196
6.6.1	Abstract Notation	196
6.6.2	Taylor Expansions for Multi-Variable Functions	196
6.6.3	Newton’s Method	197
6.6.4	Implementation	198
6.7	Exercises	199
References		203
Index		205

List of Exercises

Exercise 1.1: Error messages	22
Exercise 1.2: Volume of a cube	23
Exercise 1.3: Area and circumference of a circle	23
Exercise 1.4: Volumes of three cubes	23
Exercise 1.5: Average of integers	23
Exercise 1.6: Interactive computing of volume and area	23
Exercise 1.7: Update variable at command prompt	24
Exercise 1.8: Formatted print to screen	24
Exercise 1.9: Matlab documentation and random numbers	24
Exercise 2.1: Introducing errors	38
Exercise 2.2: Compare integers a and b	38
Exercise 2.3: Functions for circumference and area of a circle	38
Exercise 2.4: Function for area of a rectangle	39
Exercise 2.5: Area of a polygon	39
Exercise 2.6: Average of integers	40
Exercise 2.7: While loop with errors	40
Exercise 2.8: Area of rectangle versus circle	40
Exercise 2.9: Find crossing points of two graphs	40
Exercise 2.10: Sort array with numbers	41
Exercise 2.11: Compute π	41
Exercise 2.12: Compute combinations of sets	41
Exercise 2.13: Frequency of random numbers	42
Exercise 2.14: Game 21	42
Exercise 2.15: Linear interpolation	42
Exercise 2.16: Test straight line requirement	43
Exercise 2.17: Fit straight line to data	43
Exercise 2.18: Fit sines to straight line	44
Exercise 2.19: Count occurrences of a string in a string	45
Exercise 3.1: Hand calculations for the trapezoidal method	80
Exercise 3.2: Hand calculations for the midpoint method	80
Exercise 3.3: Compute a simple integral	80
Exercise 3.4: Hand-calculations with sine integrals	80
Exercise 3.5: Make test functions for the midpoint method	81
Exercise 3.6: Explore rounding errors with large numbers	81

Exercise 3.7: Write test functions for $\int_0^4 \sqrt{x} dx$	81
Exercise 3.8: Rectangle methods	81
Exercise 3.9: Adaptive integration	82
Exercise 3.10: Integrating x raised to x	83
Exercise 3.11: Integrate products of sine functions	83
Exercise 3.12: Revisit fit of sines to a function	83
Exercise 3.13: Derive the trapezoidal rule for a double integral	84
Exercise 3.14: Compute the area of a triangle by Monte Carlo integration	84
Exercise 4.1: Geometric construction of the Forward Euler method	144
Exercise 4.2: Make test functions for the Forward Euler method	145
Exercise 4.3: Implement and evaluate Heun's method	145
Exercise 4.4: Find an appropriate time step; logistic model	145
Exercise 4.5: Find an appropriate time step; SIR model	146
Exercise 4.6: Model an adaptive vaccination campaign	146
Exercise 4.7: Make a SIRV model with time-limited effect of vaccination	146
Exercise 4.8: Refactor a flat program	146
Exercise 4.9: Simulate oscillations by a general ODE solver	146
Exercise 4.10: Compute the energy in oscillations	147
Exercise 4.11: Use a Backward Euler scheme for population growth	147
Exercise 4.12: Use a Crank-Nicolson scheme for population growth	148
Exercise 4.13: Understand finite differences via Taylor series	148
Exercise 4.14: Use a Backward Euler scheme for oscillations	149
Exercise 4.15: Use Heun's method for the SIR model	150
Exercise 4.16: Use Odespy to solve a simple ODE	150
Exercise 4.17: Set up a Backward Euler scheme for oscillations	150
Exercise 4.18: Set up a Forward Euler scheme for nonlinear and damped oscillations	151
Exercise 4.19: Discretize an initial condition	151
Exercise 5.1: Simulate a diffusion equation by hand	169
Exercise 5.2: Compute temperature variations in the ground	170
Exercise 5.3: Compare implicit methods	170
Exercise 5.4: Explore adaptive and implicit methods	171
Exercise 5.5: Investigate the θ rule	171
Exercise 5.6: Compute the diffusion of a Gaussian peak	172
Exercise 5.7: Vectorize a function for computing the area of a polygon	173
Exercise 5.8: Explore symmetry	173
Exercise 5.9: Compute solutions as $t \rightarrow \infty$	174
Exercise 5.10: Solve a two-point boundary value problem	175
Exercise 6.1: Understand why Newton's method can fail	199
Exercise 6.2: See if the secant method fails	199
Exercise 6.3: Understand why the bisection method cannot fail	200
Exercise 6.4: Combine the bisection method with Newton's method	200
Exercise 6.5: Write a test function for Newton's method	200
Exercise 6.6: Solve nonlinear equation for a vibrating beam	200