

Contents

Preface	xiii
1 Introduction	1
1.1 Learning computing	1
1.2 Background references	3
1.2.1 Computer programming in general	4
1.2.2 Maple in particular	4
1.2.3 Algorithms in general	5
1.2.4 Computer algebra algorithms	5
1.2.5 Numerical algorithms	5
1.2.6 Online references	6
1.3 Fundamental terminology	6
1.4 Maple interfaces	8
1.5 The Maple worksheet interface	11
1.5.1 The online help system	11
1.5.2 Balloon Help	13
1.5.3 Use of the mouse	13
1.5.4 Useful keyboard shortcuts	15
1.5.5 New lines and execution groups	15
1.5.6 Opening and executing worksheets	17
1.5.7 Application and document windows	17
1.5.8 Averting disaster	17
1.5.9 My worksheet shrank!	19
1.5.10 Don't work in the dark	19
1.5.11 Compatibility with Maple V	19
1.6 Exercises: the worksheet interface	20
2 Review of Basics	23
2.1 Maple syntax: special symbols	23
2.1.1 Variables, assignments, equations, procedures, and mappings	24
2.1.2 Quotes	25
2.1.3 Brackets	28

2.1.4	Mathematical symbols	30
2.1.5	Programming symbols	33
2.2	Assignment, evaluation, substitution, and simplification	41
2.2.1	Assignment and evaluation	42
2.2.2	Evaluation and substitution	46
2.2.3	Multiple substitutions	48
2.2.4	Simplification	49
2.3	Common errors	51
2.3.1	Juxtaposition	51
2.3.2	Upper and lower case letters	52
2.3.3	The exponential function and the letter e	53
2.3.4	Assignments and equations	54
2.4	Sequences, lists, and sets	55
2.4.1	Parallel assignment	55
2.4.2	Functions and their argument sequences	57
2.4.3	Automatic simplification; inert functions	57
2.4.4	Functions and operators	58
2.4.5	Operations on sequences, lists, and sets: <code>op</code>	59
2.4.6	Operations on elements of sequences, lists, and sets: <code>op</code> , <code>[]</code>	60
2.4.7	The empty sequence: <code>NULL</code>	63
2.4.8	Constructing sequences, lists, and sets	64
2.4.9	Relatives of sequences: sums and products	66
2.5	Characters and strings	68
2.6	Exercises	70
3	Two-Dimensional Plotting	73
3.1	Simple plotting	73
3.1.1	Plotting graphs of functions	73
3.1.2	Plotting graphs of “strictly numerical” functions	76
3.1.3	Plot ranges	77
3.1.4	Multiple plots	78
3.1.5	Parametric plotting	79
3.1.6	Plotting points	81
3.1.7	Plot options	82
3.1.8	Plotting discontinuous functions	84
3.1.9	Displaying and manipulating plots	85
3.2	The <code>plots</code> package and animation	88
3.2.1	Aside: packages and the <code>with</code> command	88
3.2.2	Simple animation: <code>animate</code>	89
3.2.3	Combining and animating plots: <code>display</code>	90
3.2.4	Changing default plotting options: <code>setoptions</code>	93
3.2.5	Annotating plots: <code>textplot</code>	93
3.2.6	A simple geometrical example	94
3.3	A graphical investigation of the cardioid	94

3.3.1	The basic geometry	95
3.3.2	The geometry in more detail	97
3.3.3	Animating the geometry	98
3.4	Exporting plots (and worksheets)	100
3.5	Exercises: the cycloid	101
3.6	Appendix: code to plot the figures	103
4	Three-Dimensional Plotting	105
4.1	The <code>plot3d</code> function	106
4.1.1	Graphs of real-valued functions of two real variables	106
4.1.2	Parametric plotting of surfaces	107
4.1.3	Multiple three-dimensional plots	110
4.2	Three-dimensional plotting options	110
4.3	Interacting with three-dimensional plots	112
4.4	Using the <code>plots</code> package	112
4.4.1	Setting default options and plotting text: <code>setoptions3d</code> , <code>textplot3d</code>	112
4.4.2	Combining and animating three-dimensional plots: <code>animate3d</code> , <code>display3d</code>	112
4.4.3	Plotting curves in space: <code>spacecurve</code>	113
4.4.4	Plotting contours: <code>contourplot3d</code> , <code>contourplot</code>	114
4.4.5	Plotting gradient fields: <code>gradplot</code>	117
4.4.6	Implicit plotting: <code>implicitplot</code>	118
4.5	Three-dimensional bouncing ball animation	119
4.6	Exporting three-dimensional plots	121
4.7	Exercises	123
4.8	Appendix: code to plot the figures	125
5	Numerical and Semi-Numerical Computation	127
5.1	Number systems	128
5.1.1	Number systems in mathematics	128
5.1.2	Number systems in Maple	131
5.1.3	Number systems in other languages	139
5.2	Floating-point computation in Maple	139
5.2.1	Floating-point evaluation: <code>evalf</code> , <code>evalhf</code>	140
5.2.2	Solution of algebraic equations: <code>solve</code> , <code>fsolve</code>	142
5.2.3	Numerical integration (quadrature): <code>evalf/Int</code>	146
5.2.4	Solution of differential equations: <code>dsolve</code>	148
5.3	Multiple roots	150
5.4	Aliases and algebraic numbers	151
5.4.1	Alias	151
5.4.2	Algebraic numbers	152
5.4.3	Implicit dependence	153
5.5	Computing with modular integers	154
5.6	Exercises	156

5.7	Appendix: code to plot the figures	158
6	Numerical Linear Algebra	159
6.1	Linear algebra in Maple	159
6.2	Numerical linear algebra	160
6.3	Interactive linear algebra	161
6.4	Viewing large matrices and vectors	164
6.5	Using the <code>LinearAlgebra</code> package	166
6.6	Solving systems of linear equations	167
6.7	Special matrices	169
6.8	Eigenvalue problems	172
6.9	Ill-conditioned problems	176
6.10	Under- and over-determined problems	180
6.10.1	Under-determined problems	180
6.10.2	Over-determined problems	181
6.10.3	Solution by singular value decomposition	182
6.10.4	Least squares solution	184
6.11	Exercises	185
7	Logic and Control Structures	187
7.1	Relational operators: <code><</code> , <code><=</code> , <code>></code> , <code>>=</code> , <code>=</code> , <code><></code>	187
7.2	Boolean algebra	191
7.3	Boolean operators in Maple: <code>and</code> , <code>or</code> , <code>not</code>	191
7.3.1	Truth tables in Maple	192
7.3.2	Boolean expressions	193
7.4	Conditional execution: <code>if</code>	194
7.4.1	Simple conditional execution: <code>if</code>	194
7.4.2	Alternative execution: <code>else</code>	195
7.4.3	Multiple alternative execution: <code>elif</code>	196
7.5	Piecewise-defined functions	198
7.5.1	Piecewise-defined functions using <code>if</code>	198
7.5.2	Piecewise-defined functions using <code>piecewise</code>	201
7.5.3	Continuity class	202
7.5.4	Conditional evaluation: the ‘ <code>if</code> ’ function	204
7.6	Loops: <code>do</code> , <code>while</code> , <code>for</code>	204
7.6.1	The general controlled loop: <code>while</code>	205
7.6.2	Nested control structures	206
7.6.3	Iterating over an integer range: <code>for ... to</code>	207
7.6.4	Iterating over a structure: <code>for ... in</code>	210
7.6.5	Other ways to control loops: <code>break</code> , <code>next</code>	211
7.6.6	The <code>for ... to</code> -loop implemented as a <code>while</code> -loop	212
7.6.7	Double loops	212
7.7	Exercises	216

8	Procedures and Recursion	219
8.1	Defining and using procedures	220
8.1.1	Returning versus printing values	220
8.2	General procedure definition syntax	221
8.3	Variable scope	222
8.4	Procedure arguments	225
8.5	Special identifiers in procedures	227
8.5.1	Variable numbers of arguments: <code>nargs</code> , <code>args</code>	227
8.5.2	Returning unevaluated: <code>procname</code>	229
8.6	Terminating procedure execution	231
8.7	Examples: integer factorization	232
8.7.1	Aside: <code>ifactor</code>	236
8.8	Displaying Maple procedures	237
8.9	Procedure options and remember tables	238
8.9.1	Remember tables	239
8.10	More about variable scope	241
8.10.1	Lexical scoping	241
8.10.2	Returning local variables	242
8.11	Recursion	244
8.11.1	The factorial function	244
8.11.2	Recursion and stacks	245
8.11.3	Avoiding infinite recursion	246
8.11.4	Recursive summation	247
8.11.5	Recursive power computation	248
8.12	Procedures that output plots	250
8.13	Recursive plotting	253
8.14	Animating the Koch snowflake fractal	255
8.14.1	The snowflake construction	256
8.14.2	Basic geometry	256
8.14.3	Constructing a single side	257
8.14.4	Animating the “growth” of a snowflake	258
8.14.5	Code to plot the snowflake construction	259
8.15	Program design	259
8.15.1	Understand the problem	260
8.15.2	Design the algorithm	261
8.15.3	Write the program	261
8.15.4	Test the program	262
8.16	Exercises	263
9	Operators and Functions	267
9.1	Operators	267
9.1.1	Syntax	267
9.1.2	Inert operators	268
9.1.3	Active operators	270
9.2	Converting between expressions and mappings	273

9.3	Composition of mappings and functions	274
9.4	The derivative operator D	275
9.4.1	Partial derivatives	278
9.4.2	Specifying conditions on differential equations	279
9.5	Set operations	279
9.5.1	Union	279
9.5.2	Intersection	280
9.5.3	Difference	281
9.5.4	Symmetric difference	281
9.5.5	Membership	282
9.5.6	Set algebra	284
9.5.7	Power sets	284
9.6	Examples of recursive integer and polynomial functions	286
9.6.1	Euclidean division	286
9.6.2	Greatest common divisors	287
9.6.3	Digits and coefficients	288
9.7	Exercises	291
10	Data Types	295
10.1	Primitive types: <code>whattype</code>	296
10.2	Use of type information: error checking, polymorphism	298
10.3	Alternative types: $\{type1, type2, \dots\}$	298
10.4	The main Maple data types	299
10.4.1	Numerical types	299
10.4.2	Algebraic types	300
10.4.3	Boolean types	300
10.4.4	Structural types	300
10.4.5	Programming types	301
10.5	Type testing: <code>type</code> and <code>::</code>	301
10.5.1	Error-checking of procedure arguments using <code>::</code>	301
10.5.2	General type testing: <code>type</code> or <code>::</code>	302
10.5.3	Quoting type specifications	303
10.6	Boolean combinations of types: <code>And</code> , <code>Or</code> , <code>Not</code>	304
10.7	Structured or nested data types	305
10.7.1	Generic structured types	306
10.7.2	Specific structured types	307
10.7.3	Type-testing commuting operators	308
10.7.4	Type-testing sequences	310
10.8	Special procedure-argument types	311
10.9	Examples of polymorphic procedures	313
10.9.1	Active relational operators	314
10.9.2	A more general maximum function	315
10.9.3	A more general (rational) factorial function	318
10.9.4	A more general digit-sequence function	320
10.10	Defining new types	321

10.11 Exercises	323
11 Conventional Programming	329
11.1 Operations on structures	330
11.1.1 Mapping over structures: <code>map</code>	330
11.1.2 Mapping on other arguments: <code>map2</code>	331
11.1.3 How is <code>map</code> defined?	332
11.1.4 Mapping over arrays and tables	334
11.1.5 Selecting from structures: <code>select</code> , <code>remove</code>	335
11.2 Implementing vector and matrix algebra	337
11.2.1 Vector addition	338
11.2.2 Scalar product	339
11.2.3 Matrix multiplication	341
11.3 Data processing in Maple	342
11.3.1 Outputting to a file	342
11.3.2 Inputting from a file	343
11.3.3 Elementary statistics	344
11.3.4 Other methods of input and output	346
11.3.5 Aside: offline processing	346
11.4 Interactive programs in Maple	347
11.5 Error handling: <code>try</code> , <code>catch</code>	351
11.6 Debugging	354
11.6.1 <code>infolevel</code>	356
11.6.2 <code>printlevel</code>	357
11.6.3 <code>debug</code> and <code>trace</code>	359
11.6.4 Interactive debugging	360
11.7 Exercises	364
12 Algebraic Programming	369
12.1 Univariate polynomial algorithms	369
12.1.1 Squarefree factorization	369
12.1.2 Sturm sequences	373
12.1.3 Finding zeros by interval halving	377
12.2 Multivariate polynomial algorithms	381
12.2.1 Generalized polynomial division	381
12.2.2 Gröbner bases	388
12.3 Exercises	397
13 Spreadsheets	399
13.1 Introduction	400
13.2 Copying and moving cells	402
13.3 Working with sequences	405
13.4 Tabulating data	406
13.5 Programming spreadsheets	410
13.6 Case study: the simple pendulum	412

13.6.1	Exact solutions for small and large amplitude	413
13.6.2	Approximate numerical solution	415
13.6.3	Numerical computation of the period	418
13.6.4	Numerical computation of the amplitude	421
13.7	Exercises	423
14	Text Processing	425
14.1	Text files	426
14.2	Counting lines, words, and characters	428
14.2.1	Counting words and characters within a line	428
14.2.2	Counting lines, words, and characters in a file	433
14.3	Text formatting	436
14.4	A Markov chain algorithm	440
14.5	Exercises	444
15	Object Orientation and Modules	445
15.1	Introduction to object orientation	446
15.2	Block matrices	447
15.3	Modules as records	448
15.4	Modules in more detail	450
15.5	Module constructors	451
15.6	Modules as objects	455
15.7	Data encapsulation	457
15.8	Accessing block matrix elements	463
15.9	Square and inverse block matrices	464
15.10	Modules and types	466
15.11	Block matrix multiplication	467
15.12	Limitations of modules as objects	469
15.13	Operator overloading	470
15.14	Using a private Maple library	477
15.15	Modules as packages	481
15.16	Adding online help	485
15.17	Exercises	487
	Bibliography	489
	Index	491