CONTENTS

Preface **x**

1 Introduction **1**

2 Python programming for physicists **9**
2.1 Getting started 9
2.2 Basic programming 12
2.2.1 Variables and assignments 12
2.2.2 Variable types 14
2.2.3 Output and input statements 18
2.2.4 Arithmetic 23
2.2.5 Functions, packages, and modules 31
2.2.6 Built-in functions 35
2.2.7 Comment statements 37
2.3 Controlling programs with "if" and "while" 39
2.3.1 The if statement 39
2.3.2 The while statement 42
2.3.3 Break and continue 43
2.4 Lists and arrays 46
2.4.1 Lists 47
2.4.2 Arrays 53
2.4.3 Reading an array from a file 57
2.4.4 Arithmetic with arrays 58
2.4.5 Slicing 66
2.5 "For" loops 67
2.6 User-defined functions 75
2.7 Good programming style 84

3 Graphics and visualization **88**
3.1 Graphs 88

3.2 Scatter plots . . . 99
3.3 Density plots . . . 102
3.4 3D graphics . . . 111
3.5 Animation . . . 117

4 Accuracy and speed 126
4.1 Variables and ranges . . . 126
4.2 Numerical error . . . 128
4.3 Program speed . . . 134

5 Integrals and derivatives 140
5.1 Fundamental methods for evaluating integrals . . . 140
5.1.1 The trapezoidal rule . . . 141
5.1.2 Simpson's rule . . . 144
5.2 Errors on integrals . . . 149
5.2.1 Practical estimation of errors . . . 153
5.3 Choosing the number of steps . . . 155
5.4 Romberg integration . . . 159
5.5 Higher-order integration methods . . . 163
5.6 Gaussian quadrature . . . 165
5.6.1 Nonuniform sample points . . . 165
5.6.2 Sample points for Gaussian quadrature . . . 168
5.6.3 Errors on Gaussian quadrature . . . 175
5.7 Choosing an integration method . . . 177
5.8 Integrals over infinite ranges . . . 179
5.9 Multiple integrals . . . 182
5.10 Derivatives . . . 188
5.10.1 Forward and backward differences . . . 188
5.10.2 Errors . . . 189
5.10.3 Central differences . . . 191
5.10.4 Higher-order approximations for derivatives . . . 194
5.10.5 Second derivatives . . . 197
5.10.6 Partial derivatives . . . 198
5.10.7 Derivatives of noisy data . . . 199
5.11 Interpolation . . . 202

6 Solution of linear and nonlinear equations 214
6.1 Simultaneous linear equations . . . 214
6.1.1 Gaussian elimination . . . 215

6.1.2 Backsubstitution . . . 217
6.1.3 Pivoting . . . 221
6.1.4 LU decomposition . . . 222
6.1.5 Calculating the inverse of a matrix . . . 231
6.1.6 Tridiagonal and banded matrices . . . 232
6.2 Eigenvalues and eigenvectors . . . 241
6.3 Nonlinear equations . . . 250
6.3.1 The relaxation method . . . 250
6.3.2 Rate of convergence of the relaxation method . . . 255
6.3.3 Relaxation method for two or more variables . . . 261
6.3.4 Binary search . . . 263
6.3.5 Newton's method . . . 268
6.3.6 The secant method . . . 273
6.3.7 Newton's method for two or more variables . . . 275
6.4 Maxima and minima of functions . . . 278
6.4.1 Golden ratio search . . . 279
6.4.2 The Gauss–Newton method and gradient descent . . . 286

7 Fourier transforms 289
7.1 Fourier series . . . 289
7.2 The discrete Fourier transform . . . 292
7.2.1 Positions of the sample points . . . 297
7.2.2 Two-dimensional Fourier transforms . . . 299
7.2.3 Physical interpretation of the Fourier transform . . . 300
7.3 Discrete cosine and sine transforms . . . 304
7.3.1 Technological applications of cosine transforms . . . 308
7.4 Fast Fourier transforms . . . 310
7.4.1 Formulas for the FFT . . . 313
7.4.2 Standard functions for fast Fourier transforms . . . 315
7.4.3 Fast cosine and sine transforms . . . 318

8 Ordinary differential equations 327
8.1 First-order differential equations with one variable . . . 327
8.1.1 Euler's method . . . 328
8.1.2 The Runge–Kutta method . . . 331
8.1.3 The fourth-order Runge–Kutta method . . . 336
8.1.4 Solutions over infinite ranges . . . 340
8.2 Differential equations with more than one variable . . . 343

8.3 Second-order differential equations . . . 347
8.4 Varying the step size . . . 355
8.5 Other methods for differential equations . . . 364
8.5.1 The leapfrog method . . . 364
8.5.2 Time reversal and energy conservation . . . 367
8.5.3 The Verlet method . . . 371
8.5.4 The modified midpoint method . . . 374
8.5.5 The Bulirsch–Stoer method . . . 377
8.5.6 Interval size for the Bulirsch–Stoer method . . . 387
8.6 Boundary value problems . . . 388
8.6.1 The shooting method . . . 388
8.6.2 The relaxation method . . . 392
8.6.3 Eigenvalue problems . . . 392

9 Partial differential equations 404
9.1 Boundary value problems and the relaxation method . . . 406
9.2 Faster methods for boundary value problems . . . 414
9.2.1 Overrelaxation . . . 414
9.2.2 The Gauss–Seidel method . . . 415
9.3 Initial value problems . . . 418
9.3.1 The FTCS method . . . 419
9.3.2 Numerical stability . . . 425
9.3.3 The implicit and Crank–Nicolson methods . . . 432
9.3.4 Spectral methods . . . 435

10 Random processes and Monte Carlo methods 444
10.1 Random numbers . . . 444
10.1.1 Random number generators . . . 445
10.1.2 Random number seeds . . . 449
10.1.3 Random numbers and secret codes . . . 450
10.1.4 Probabilities and biased coins . . . 453
10.1.5 Nonuniform random numbers . . . 457
10.1.6 Gaussian random numbers . . . 460
10.2 Monte Carlo integration . . . 464
10.2.1 The mean value method . . . 468
10.2.2 Integrals in many dimensions . . . 470
10.2.3 Importance sampling . . . 472
10.3 Monte Carlo simulation . . . 476

10.3.1 Importance sampling and statistical mechanics 476
10.3.2 The Markov chain method 479
10.4 Simulated annealing 490

11 Using what you have learned **502**

Appendices:
A Installing Python 508
B Differences between Python versions 510
C Gaussian quadrature 514
D Convergence of Markov chain Monte Carlo calculations 520
E Useful programs 523

Index **532**