

Contents

1	Computing with Formulas	1
1.1	The First Programming Encounter: A Formula	1
1.1.1	Using a Program as a Calculator	2
1.1.2	About Programs and Programming	2
1.1.3	Tools for Writing Programs	3
1.1.4	Using Idle to Write the Program	4
1.1.5	How to Run the Program	7
1.1.6	Verifying the Result	8
1.1.7	Using Variables	8
1.1.8	Names of Variables	9
1.1.9	Reserved Words in Python	10
1.1.10	Comments	10
1.1.11	Formatting Text and Numbers	11
1.2	Computer Science Glossary	14
1.3	Another Formula: Celsius-Fahrenheit Conversion	19
1.3.1	Potential Error: Integer Division	19
1.3.2	Objects in Python	20
1.3.3	Avoiding Integer Division	21
1.3.4	Arithmetic Operators and Precedence	22
1.4	Evaluating Standard Mathematical Functions	23
1.4.1	Example: Using the Square Root Function	23
1.4.2	Example: Using More Mathematical Functions	25
1.4.3	A First Glimpse of Round-Off Errors	26
1.5	Interactive Computing	27
1.5.1	Using the Python Shell	27
1.5.2	Type Conversion	28
1.5.3	IPython	29
1.6	Complex Numbers	32
1.6.1	Complex Arithmetics in Python	33

1.6.2	Complex Functions in Python	33
1.6.3	Unified Treatment of Complex and Real Functions	34
1.7	Summary	35
1.7.1	Chapter Topics	35
1.7.2	Example: Trajectory of a Ball	39
1.7.3	About Typesetting Conventions in This Book ...	40
1.8	Exercises	41
2	Loops and Lists	49
2.1	While Loops	49
2.1.1	A Naive Solution	49
2.1.2	While Loops	50
2.1.3	Boolean Expressions	52
2.1.4	Loop Implementation of a Sum	54
2.2	Lists	55
2.2.1	Basic List Operations	55
2.2.2	For Loops	58
2.3	Alternative Implementations with Lists and Loops	60
2.3.1	While Loop Implementation of a For Loop	60
2.3.2	The Range Construction	60
2.3.3	For Loops with List Indices	61
2.3.4	Changing List Elements	63
2.3.5	List Comprehension	63
2.3.6	Traversing Multiple Lists Simultaneously	64
2.4	Nested Lists	64
2.4.1	A Table as a List of Rows or Columns	65
2.4.2	Printing Objects	66
2.4.3	Extracting Sublists	67
2.4.4	Traversing Nested Lists	69
2.5	Tuples	71
2.6	Summary	72
2.6.1	Chapter Topics	72
2.6.2	Example: Analyzing List Data	75
2.6.3	How to Find More Python Information	78
2.7	Exercises	79
3	Functions and Branching	87
3.1	Functions	87
3.1.1	Functions of One Variable	87
3.1.2	Local and Global Variables	89
3.1.3	Multiple Arguments	91
3.1.4	Multiple Return Values	93
3.1.5	Functions with No Return Values	95
3.1.6	Keyword Arguments	96
3.1.7	Doc Strings	99

3.1.8	Function Input and Output	100
3.1.9	Functions as Arguments to Functions	100
3.1.10	The Main Program	102
3.1.11	Lambda Functions	103
3.2	Branching	104
3.2.1	If-Else Blocks	104
3.2.2	Inline If Tests	106
3.3	Mixing Loops, Branching, and Functions in Bioinformatics Examples	107
3.3.1	Counting Letters in DNA Strings	107
3.3.2	Efficiency Assessment	113
3.4	Summary	115
3.4.1	Chapter Topics	115
3.4.2	Example: Numerical Integration	116
3.5	Exercises	120
4	Input Data and Error Handling	137
4.1	Asking Questions and Reading Answers	138
4.1.1	Reading Keyboard Input	138
4.1.2	The Magic “eval” Function	139
4.1.3	The Magic “exec” Function	143
4.1.4	Turning String Expressions into Functions	144
4.2	Reading from the Command Line	145
4.2.1	Providing Input on the Command Line	145
4.2.2	A Variable Number of Command-Line Arguments	146
4.2.3	More on Command-Line Arguments	147
4.2.4	Option-Value Pairs on the Command Line	148
4.3	Handling Errors	152
4.3.1	Exception Handling	153
4.3.2	Raising Exceptions	156
4.4	A Glimpse of Graphical User Interfaces	158
4.5	Making Modules	161
4.5.1	Example: Interest on Bank Deposits	161
4.5.2	Collecting Functions in a Module File	162
4.5.3	Using Modules	167
4.6	Summary	169
4.6.1	Chapter Topics	169
4.6.2	Example: Bisection Root Finding	172
4.7	Exercises	180
5	Array Computing and Curve Plotting	187
5.1	Vectors	188
5.1.1	The Vector Concept	188
5.1.2	Mathematical Operations on Vectors	189
5.1.3	Vector Arithmetics and Vector Functions	191

5.2	Arrays in Python Programs	193
5.2.1	Using Lists for Collecting Function Data	193
5.2.2	Basics of Numerical Python Arrays	194
5.2.3	Computing Coordinates and Function Values	195
5.2.4	Vectorization	196
5.3	Curve Plotting	198
5.3.1	Matplotlib; Pylab	198
5.3.2	Matplotlib; Pyplot	202
5.3.3	SciTools and Easyviz	204
5.3.4	Making Animations	209
5.3.5	Curves in Pure Text	214
5.4	Plotting Difficulties	215
5.4.1	Piecewisely Defined Functions	216
5.4.2	Rapidly Varying Functions	218
5.5	More Advanced Vectorization of Functions	219
5.5.1	Vectorizing StringFunction Objects	220
5.5.2	Vectorization of the Heaviside Function	221
5.5.3	Vectorization of a Hat Function	224
5.6	More on Numerical Python Arrays	226
5.6.1	Copying Arrays	226
5.6.2	In-Place Arithmetics	227
5.6.3	Allocating Arrays	228
5.6.4	Generalized Indexing	228
5.6.5	Testing for the Array Type	229
5.6.6	Compact Syntax for Array Generation	230
5.6.7	Shape Manipulation	230
5.7	Higher-Dimensional Arrays	231
5.7.1	Matrices and Arrays	231
5.7.2	Two-Dimensional Numerical Python Arrays	232
5.7.3	Array Computing	235
5.7.4	Two-Dimensional Arrays and Functions of Two Variables	235
5.7.5	Matrix Objects	236
5.8	Summary	237
5.8.1	Chapter Topics	237
5.8.2	Example: Animating a Function	239
5.9	Exercises	243
6	Files, Strings, and Dictionaries	257
6.1	Reading Data from File	257
6.1.1	Reading a File Line by Line	258
6.1.2	Reading a Mixture of Text and Numbers	261
6.1.3	What Is a File, Really?	262
6.2	Dictionaries	266
6.2.1	Making Dictionaries	266

6.2.2	Dictionary Operations	267
6.2.3	Example: Polynomials as Dictionaries	269
6.2.4	Dictionaries with Default Values and Ordering . .	271
6.2.5	Example: File Data in Dictionaries	273
6.2.6	Example: File Data in Nested Dictionaries	274
6.2.7	Example: Comparing Stock Prices	278
6.3	Strings	282
6.3.1	Common Operations on Strings	282
6.3.2	Example: Reading Pairs of Numbers	286
6.3.3	Example: Reading Coordinates	288
6.4	Reading Data from Web Pages	291
6.4.1	About Web Pages	291
6.4.2	How to Access Web Pages in Programs	292
6.4.3	Example: Reading Pure Text Files	293
6.4.4	Example: Extracting Data from HTML	295
6.5	Writing Data to File	296
6.5.1	Example: Writing a Table to File	296
6.5.2	Standard Input and Output as File Objects	298
6.5.3	Reading and Writing Spreadsheet Files	300
6.6	Examples from Analyzing DNA	305
6.6.1	Computing Frequencies	305
6.6.2	Analyzing the Frequency Matrix	312
6.6.3	Finding Base Frequencies	315
6.6.4	Translating Genes into Proteins	317
6.6.5	Some Humans Can Drink Milk, While Others Cannot	322
6.7	Summary	323
6.7.1	Chapter Topics	323
6.7.2	Example: A File Database	325
6.8	Exercises	329
7	Introduction to Classes	341
7.1	Simple Function Classes	342
7.1.1	Problem: Functions with Parameters	342
7.1.2	Representing a Function as a Class	344
7.1.3	Another Function Class Example	350
7.1.4	Alternative Function Class Implementations	351
7.1.5	Making Classes Without the Class Construct	353
7.2	More Examples on Classes	356
7.2.1	Bank Accounts	356
7.2.2	Phone Book	358
7.2.3	A Circle	359
7.3	Special Methods	360
7.3.1	The Call Special Method	361
7.3.2	Example: Automagic Differentiation	361

7.3.3	Example: Automagic Integration.....	364
7.3.4	Turning an Instance into a String.....	366
7.3.5	Example: Phone Book with Special Methods	367
7.3.6	Adding Objects	369
7.3.7	Example: Class for Polynomials	369
7.3.8	Arithmetic Operations and Other Special Methods.....	373
7.3.9	Special Methods for String Conversion.....	374
7.4	Example: Class for Vectors in the Plane.....	375
7.4.1	Some Mathematical Operations on Vectors.....	376
7.4.2	Implementation	376
7.4.3	Usage	378
7.5	Example: Class for Complex Numbers	379
7.5.1	Implementation	380
7.5.2	Illegal Operations.....	381
7.5.3	Mixing Complex and Real Numbers.....	382
7.5.4	Special Methods for “Right” Operands	384
7.5.5	Inspecting Instances	386
7.6	Static Methods and Attributes	387
7.7	Summary	388
7.7.1	Chapter Topics	388
7.7.2	Example: Interval Arithmetics.....	389
7.8	Exercises	395
8	Random Numbers and Simple Games.....	413
8.1	Drawing Random Numbers	414
8.1.1	The Seed	414
8.1.2	Uniformly Distributed Random Numbers	415
8.1.3	Visualizing the Distribution	416
8.1.4	Vectorized Drawing of Random Numbers	417
8.1.5	Computing the Mean and Standard Deviation ...	418
8.1.6	The Gaussian or Normal Distribution	419
8.2	Drawing Integers	420
8.2.1	Random Integer Functions	421
8.2.2	Example: Throwing a Die.....	422
8.2.3	Drawing a Random Element from a List	422
8.2.4	Example: Drawing Cards from a Deck	423
8.2.5	Example: Class Implementation of a Deck	425
8.3	Computing Probabilities	428
8.3.1	Principles of Monte Carlo Simulation.....	428
8.3.2	Example: Throwing Dice	429
8.3.3	Example: Drawing Balls from a Hat.....	432
8.3.4	Random Mutations of Genes	434
8.3.5	Example: Policies for Limiting Population Growth	439
8.4	Simple Games	442

8.4.1	Guessing a Number	442
8.4.2	Rolling Two Dice	443
8.5	Monte Carlo Integration	446
8.5.1	Standard Monte Carlo Integration	446
8.5.2	Area Computing by Throwing Random Points ...	448
8.6	Random Walk in One Space Dimension	450
8.6.1	Basic Implementation	451
8.6.2	Visualization	452
8.6.3	Random Walk as a Difference Equation	452
8.6.4	Computing Statistics of the Particle Positions ...	453
8.6.5	Vectorized Implementation	454
8.7	Random Walk in Two Space Dimensions	456
8.7.1	Basic Implementation	456
8.7.2	Vectorized Implementation	457
8.8	Summary	459
8.8.1	Chapter Topics	459
8.8.2	Example: Random Growth	460
8.9	Exercises	466
9	Object-Oriented Programming	483
9.1	Inheritance and Class Hierarchies	483
9.1.1	A Class for Straight Lines	484
9.1.2	A First Try on a Class for Parabolas	485
9.1.3	A Class for Parabolas Using Inheritance	485
9.1.4	Checking the Class Type	487
9.1.5	Attribute Versus Inheritance	488
9.1.6	Extending Versus Restricting Functionality	489
9.1.7	Superclass for Defining an Interface	490
9.2	Class Hierarchy for Numerical Differentiation	492
9.2.1	Classes for Differentiation	493
9.2.2	A Flexible Main Program	496
9.2.3	Extensions	497
9.2.4	Alternative Implementation via Functions	500
9.2.5	Alternative Implementation via Functional Programming	501
9.2.6	Alternative Implementation via a Single Class ...	502
9.3	Class Hierarchy for Numerical Integration	504
9.3.1	Numerical Integration Methods	504
9.3.2	Classes for Integration	505
9.3.3	Using the Class Hierarchy	509
9.3.4	About Object-Oriented Programming	511
9.4	Class Hierarchy for Making Drawings	513
9.4.1	Using the Object Collection	514
9.4.2	Example of Classes for Geometric Objects	523
9.4.3	Adding Functionality via Recursion	528

9.4.4	Scaling, Translating, and Rotating a Figure . . .	531
9.5	Classes for DNA Analysis	534
9.5.1	Class for Regions	534
9.5.2	Class for Genes	534
9.5.3	Subclasses	539
9.6	Summary	541
9.6.1	Chapter Topics	541
9.6.2	Example: Input Data Reader	542
9.7	Exercises	548
A	Sequences and Difference Equations	557
A.1	Mathematical Models Based on Difference Equations . .	558
A.1.1	Interest Rates	559
A.1.2	The Factorial as a Difference Equation	561
A.1.3	Fibonacci Numbers	562
A.1.4	Growth of a Population	563
A.1.5	Logistic Growth	564
A.1.6	Payback of a Loan	566
A.1.7	The Integral as a Difference Equation	567
A.1.8	Taylor Series as a Difference Equation	569
A.1.9	Making a Living from a Fortune	571
A.1.10	Newton's Method	571
A.1.11	The Inverse of a Function	575
A.2	Programming with Sound	577
A.2.1	Writing Sound to File	578
A.2.2	Reading Sound from File	579
A.2.3	Playing Many Notes	580
A.2.4	Music of a Sequence	580
A.3	Exercises	583
B	Introduction to Discrete Calculus	593
B.1	Discrete Functions	593
B.1.1	The Sine Function	594
B.1.2	Interpolation	596
B.1.3	Evaluating the Approximation	596
B.1.4	Generalization	597
B.2	Differentiation Becomes Finite Differences	599
B.2.1	Differentiating the Sine Function	600
B.2.2	Differences on a Mesh	600
B.2.3	Generalization	602
B.3	Integration Becomes Summation	603
B.3.1	Dividing into Subintervals	604
B.3.2	Integration on Subintervals	605
B.3.3	Adding the Subintervals	606
B.3.4	Generalization	607

- B.4 Taylor Series 609
 - B.4.1 Approximating Functions Close to One Point.... 609
 - B.4.2 Approximating the Exponential Function 609
 - B.4.3 More Accurate Expansions..... 610
 - B.4.4 Accuracy of the Approximation..... 612
 - B.4.5 Derivatives Revisited 614
 - B.4.6 More Accurate Difference Approximations 615
 - B.4.7 Second-Order Derivatives 617
- B.5 Exercises 619
- C Introduction to Differential Equations 625
 - C.1 The Simplest Case 626
 - C.2 Exponential Growth 628
 - C.3 Logistic Growth..... 633
 - C.4 A Simple Pendulum 634
 - C.5 A Model for the Spread of a Disease 637
 - C.6 Exercises 639
- D A Complete Differential Equation Project 641
 - D.1 About the Problem: Motion and Forces in Physics 641
 - D.1.1 The Physical Problem 641
 - D.1.2 The Computational Algorithm 644
 - D.1.3 Derivation of the Mathematical Model..... 644
 - D.1.4 Derivation of the Algorithm 646
 - D.2 Program Development and Testing 648
 - D.2.1 Implementation 648
 - D.2.2 Callback Functionality 651
 - D.2.3 Making a Module 652
 - D.2.4 Verification 653
 - D.3 Visualization 655
 - D.3.1 Simultaneous Computation and Plotting..... 655
 - D.3.2 Some Applications 658
 - D.3.3 Remark on Choosing Δt 658
 - D.3.4 Comparing Several Quantities in Subplots 659
 - D.3.5 Comparing Approximate and Exact Solutions ... 660
 - D.3.6 Evolution of the Error as Δt Decreases 661
 - D.4 Exercises 665
- E Programming of Differential Equations..... 667
 - E.1 Scalar Ordinary Differential Equations 668
 - E.1.1 Examples on Right-Hand-Side Functions..... 668
 - E.1.2 The Forward Euler Scheme 670
 - E.1.3 Function Implementation 671
 - E.1.4 Verifying the Implementation 671
 - E.1.5 From Discrete to Continuous Solution 672
 - E.1.6 Switching Numerical Method 673

E.1.7	Class Implementation	674
E.1.8	Example: Logistic Growth	677
E.2	Systems of Ordinary Differential Equations	677
E.2.1	Mathematical Problem	678
E.2.2	Example of a System of ODEs	680
E.2.3	From Scalar ODE Code to Systems	681
E.2.4	Numerical Methods	684
E.2.5	The ODE Solver Class Hierarchy	685
E.2.6	The Backward Euler Method	688
E.2.7	Application 1: $u' = -u$	691
E.2.8	Application 2: The Logistic Equation	693
E.2.9	Application 3: An Oscillating System	700
E.2.10	Application 4: The Trajectory of a Ball	702
E.2.11	Further Developments of ODESolver	704
E.3	Exercises	705
F	Debugging	735
F.1	Using a Debugger	735
F.2	How to Debug	738
F.2.1	A Recipe for Program Writing and Debugging ..	738
F.2.2	Application of the Recipe	740
G	Migrating Python to Compiled Code	753
G.1	Pure Python Code for Monte Carlo Simulation	754
G.1.1	The Computational Problem	754
G.1.2	A Scalar Python Implementation	754
G.1.3	A Vectorized Python Implementation	755
G.2	Migrating Scalar Python Code to Cython	757
G.2.1	A Plain Cython Implementation	757
G.2.2	A Better Cython Implementation	759
G.3	Migrating Code to C	761
G.3.1	Writing a C Program	761
G.3.2	Migrating Loops to C Code via F2PY	762
G.3.3	Migrating Loops to C Code via Cython	764
G.3.4	Comparing Efficiency	765
H	Technical Topics	767
H.1	Different Ways of Running Python Programs	767
H.1.1	Executing Python Programs in IPython	767
H.1.2	Executing Python Programs on Unix	767
H.1.3	Executing Python Programs on Windows	769
H.1.4	Executing Python Programs on Macintosh	771
H.1.5	Making a Complete Stand-Alone Executable ...	771
H.2	Integer and Float Division	771
H.3	Visualizing a Program with Lumpy	772
H.4	Doing Operating System Tasks in Python	774

H.5 Variable Number of Function Arguments 776

 H.5.1 Variable Number of Positional Arguments 777

 H.5.2 Variable Number of Keyword Arguments 779

H.6 Evaluating Program Efficiency 781

 H.6.1 Making Time Measurements 781

 H.6.2 Profiling Python Programs..... 783

References 785

Index 787