
Hans Petter Langtangen

A Primer on Scientific Programming with Python

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	13
1.3	Another Formula: Celsius-Fahrenheit Conversion	18
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	21
1.4	Evaluating Standard Mathematical Functions	22
1.4.1	Example: Using the Square Root Function	22
1.4.2	Example: Using More Mathematical Functions	25
1.4.3	A First Glimpse of Round-Off Errors	25
1.5	Interactive Computing	26
1.5.1	Calculating with Formulas in the Interactive Shell	27
1.5.2	Type Conversion	28
1.5.3	IPython	29
1.6	Complex Numbers	31

1.6.1	Complex Arithmetics in Python	32
1.6.2	Complex Functions in Python	32
1.6.3	Unified Treatment of Complex and Real Functions	33
1.7	Summary	35
1.7.1	Chapter Topics	35
1.7.2	Summarizing Example: Trajectory of a Ball	38
1.7.3	About Typesetting Conventions in This Book ..	39
1.8	Exercises	40
2	Basic Constructions	51
2.1	Loops and Lists for Tabular Data	51
2.1.1	A Naive Solution	51
2.1.2	While Loops	52
2.1.3	Boolean Expressions	54
2.1.4	Lists	56
2.1.5	For Loops.....	58
2.1.6	Alternative Implementations with Lists and Loops	60
2.1.7	Nested Lists.....	64
2.1.8	Printing Objects	65
2.1.9	Extracting Sublists.....	66
2.1.10	Traversing Nested Lists	68
2.1.11	Tuples	70
2.2	Functions	71
2.2.1	Functions of One Variable	71
2.2.2	Local and Global Variables.....	73
2.2.3	Multiple Arguments	75
2.2.4	Multiple Return Values	77
2.2.5	Functions with No Return Values	79
2.2.6	Keyword Arguments	80
2.2.7	Doc Strings	83
2.2.8	Function Input and Output	84
2.2.9	Functions as Arguments to Functions	84
2.2.10	The Main Program	86
2.2.11	Lambda Functions	87
2.3	If Tests	88
2.4	Summary	91
2.4.1	Chapter Topics	91
2.4.2	Summarizing Example: Tabulate a Function....	94
2.4.3	How to Find More Python Information	98
2.5	Exercises	99
3	Input Data and Error Handling	119
3.1	Asking Questions and Reading Answers	120

3.1.1	Reading Keyboard Input	120
3.1.2	The Magic “eval” Function	121
3.1.3	The Magic “exec” Function	125
3.1.4	Turning String Expressions into Functions	126
3.2	Reading from the Command Line	127
3.2.1	Providing Input on the Command Line	127
3.2.2	A Variable Number of Command-Line Arguments	128
3.2.3	More on Command-Line Arguments	129
3.2.4	Option–Value Pairs on the Command Line	130
3.3	Handling Errors	132
3.3.1	Exception Handling	133
3.3.2	Raising Exceptions	136
3.4	A Glimpse of Graphical User Interfaces	139
3.5	Making Modules	141
3.5.1	Example: Compound Interest Formulas	142
3.5.2	Collecting Functions in a Module File	143
3.5.3	Using Modules	148
3.6	Summary	150
3.6.1	Chapter Topics	150
3.6.2	Summarizing Example: Bisection Root Finding	152
3.7	Exercises	160
4	Array Computing and Curve Plotting	169
4.1	Vectors	170
4.1.1	The Vector Concept	170
4.1.2	Mathematical Operations on Vectors	171
4.1.3	Vector Arithmetics and Vector Functions	173
4.2	Arrays in Python Programs	175
4.2.1	Using Lists for Collecting Function Data	175
4.2.2	Basics of Numerical Python Arrays	176
4.2.3	Computing Coordinates and Function Values ...	177
4.2.4	Vectorization	178
4.3	Curve Plotting	179
4.3.1	The SciTools and Easyviz Packages	180
4.3.2	Plotting a Single Curve	181
4.3.3	Decorating the Plot	183
4.3.4	Plotting Multiple Curves	183
4.3.5	Controlling Line Styles	185
4.3.6	Interactive Plotting Sessions	189
4.3.7	Making Animations	190
4.3.8	Advanced Easyviz Topics	193
4.3.9	Curves in Pure Text	198
4.4	Plotting Difficulties	199
4.4.1	Piecewisely Defined Functions	199

4.4.2	Rapidly Varying Functions	205
4.4.3	Vectorizing StringFunction Objects	206
4.5	More on Numerical Python Arrays	207
4.5.1	Copying Arrays	207
4.5.2	In-Place Arithmetics	207
4.5.3	Allocating Arrays	208
4.5.4	Generalized Indexing	209
4.5.5	Testing for the Array Type	210
4.5.6	Equally Spaced Numbers	211
4.5.7	Compact Syntax for Array Generation	212
4.5.8	Shape Manipulation	212
4.6	Higher-Dimensional Arrays	213
4.6.1	Matrices and Arrays	213
4.6.2	Two-Dimensional Numerical Python Arrays	214
4.6.3	Array Computing	216
4.6.4	Two-Dimensional Arrays and Functions of Two Variables	217
4.6.5	Matrix Objects	217
4.7	Summary	219
4.7.1	Chapter Topics	219
4.7.2	Summarizing Example: Animating a Function ..	220
4.8	Exercises	225
5	Sequences and Difference Equations	235
5.1	Mathematical Models Based on Difference Equations ..	236
5.1.1	Interest Rates	237
5.1.2	The Factorial as a Difference Equation	239
5.1.3	Fibonacci Numbers	240
5.1.4	Growth of a Population	241
5.1.5	Logistic Growth	242
5.1.6	Payback of a Loan	244
5.1.7	Taylor Series as a Difference Equation	245
5.1.8	Making a Living from a Fortune	246
5.1.9	Newton's Method	247
5.1.10	The Inverse of a Function	251
5.2	Programming with Sound	253
5.2.1	Writing Sound to File	253
5.2.2	Reading Sound from File	254
5.2.3	Playing Many Notes	255
5.3	Summary	256
5.3.1	Chapter Topics	256
5.3.2	Summarizing Example: Music of a Sequence	257
5.4	Exercises	260
6	Files, Strings, and Dictionaries	269

6.1	Reading Data from File	269
6.1.1	Reading a File Line by Line	270
6.1.2	Reading a Mixture of Text and Numbers	273
6.1.3	What Is a File, Really?	274
6.2	Dictionaries	278
6.2.1	Making Dictionaries	278
6.2.2	Dictionary Operations	279
6.2.3	Example: Polynomials as Dictionaries	280
6.2.4	Example: File Data in Dictionaries	282
6.2.5	Example: File Data in Nested Dictionaries	283
6.2.6	Example: Comparing Stock Prices	287
6.3	Strings	291
6.3.1	Common Operations on Strings	292
6.3.2	Example: Reading Pairs of Numbers	295
6.3.3	Example: Reading Coordinates	298
6.4	Reading Data from Web Pages	300
6.4.1	About Web Pages	300
6.4.2	How to Access Web Pages in Programs	302
6.4.3	Example: Reading Pure Text Files	302
6.4.4	Example: Extracting Data from an HTML Page	304
6.5	Writing Data to File	308
6.5.1	Example: Writing a Table to File	309
6.5.2	Standard Input and Output as File Objects	310
6.5.3	Reading and Writing Spreadsheet Files	312
6.6	Summary	317
6.6.1	Chapter Topics	317
6.6.2	Summarizing Example: A File Database	319
6.7	Exercises	323
7	Introduction to Classes	337
7.1	Simple Function Classes	338
7.1.1	Problem: Functions with Parameters	338
7.1.2	Representing a Function as a Class	340
7.1.3	Another Function Class Example	346
7.1.4	Alternative Function Class Implementations	347
7.1.5	Making Classes Without the Class Construct	349
7.2	More Examples on Classes	352
7.2.1	Bank Accounts	352
7.2.2	Phone Book	354
7.2.3	A Circle	355
7.3	Special Methods	356
7.3.1	The Call Special Method	357
7.3.2	Example: Automatic Differentiation	357
7.3.3	Example: Automatic Integration	360
7.3.4	Turning an Instance into a String	362

7.3.5	Example: Phone Book with Special Methods . . .	363
7.3.6	Adding Objects	365
7.3.7	Example: Class for Polynomials	365
7.3.8	Arithmetic Operations and Other Special Methods	369
7.3.9	More on Special Methods for String Conversion .	370
7.4	Example: Solution of Differential Equations	372
7.4.1	A Function for Solving ODEs	373
7.4.2	A Class for Solving ODEs	374
7.4.3	Verifying the Implementation	376
7.4.4	Example: Logistic Growth	377
7.5	Example: Class for Vectors in the Plane	378
7.5.1	Some Mathematical Operations on Vectors	378
7.5.2	Implementation	378
7.5.3	Usage	380
7.6	Example: Class for Complex Numbers	382
7.6.1	Implementation	382
7.6.2	Illegal Operations	383
7.6.3	Mixing Complex and Real Numbers	384
7.6.4	Special Methods for “Right” Operands	387
7.6.5	Inspecting Instances	388
7.7	Static Methods and Attributes	389
7.8	Summary	391
7.8.1	Chapter Topics	391
7.8.2	Summarizing Example: Interval Arithmetics . . .	392
7.9	Exercises	397
8	Random Numbers and Simple Games	417
8.1	Drawing Random Numbers	418
8.1.1	The Seed	418
8.1.2	Uniformly Distributed Random Numbers	419
8.1.3	Visualizing the Distribution	420
8.1.4	Vectorized Drawing of Random Numbers	421
8.1.5	Computing the Mean and Standard Deviation ..	422
8.1.6	The Gaussian or Normal Distribution	423
8.2	Drawing Integers	424
8.2.1	Random Integer Functions	425
8.2.2	Example: Throwing a Die	426
8.2.3	Drawing a Random Element from a List	427
8.2.4	Example: Drawing Cards from a Deck	427
8.2.5	Example: Class Implementation of a Deck . . .	429
8.3	Computing Probabilities	432
8.3.1	Principles of Monte Carlo Simulation	432
8.3.2	Example: Throwing Dice	433
8.3.3	Example: Drawing Balls from a Hat	435

8.3.4	Example: Policies for Limiting Population Growth	437
8.4	Simple Games	440
8.4.1	Guessing a Number	440
8.4.2	Rolling Two Dice	440
8.5	Monte Carlo Integration	443
8.5.1	Standard Monte Carlo Integration	443
8.5.2	Computing Areas by Throwing Random Points ..	446
8.6	Random Walk in One Space Dimension	447
8.6.1	Basic Implementation	448
8.6.2	Visualization	449
8.6.3	Random Walk as a Difference Equation	449
8.6.4	Computing Statistics of the Particle Positions ..	450
8.6.5	Vectorized Implementation	451
8.7	Random Walk in Two Space Dimensions	453
8.7.1	Basic Implementation	453
8.7.2	Vectorized Implementation	455
8.8	Summary	456
8.8.1	Chapter Topics	456
8.8.2	Summarizing Example: Random Growth	457
8.9	Exercises	463
9	Object-Oriented Programming	479
9.1	Inheritance and Class Hierarchies	479
9.1.1	A Class for Straight Lines	480
9.1.2	A First Try on a Class for Parabolas	481
9.1.3	A Class for Parabolas Using Inheritance	481
9.1.4	Checking the Class Type	483
9.1.5	Attribute versus Inheritance	484
9.1.6	Extending versus Restricting Functionality	485
9.1.7	Superclass for Defining an Interface	486
9.2	Class Hierarchy for Numerical Differentiation	488
9.2.1	Classes for Differentiation	488
9.2.2	A Flexible Main Program	491
9.2.3	Extensions	492
9.2.4	Alternative Implementation via Functions	495
9.2.5	Alternative Implementation via Functional Programming	496
9.2.6	Alternative Implementation via a Single Class ..	497
9.3	Class Hierarchy for Numerical Integration	499
9.3.1	Numerical Integration Methods	499
9.3.2	Classes for Integration	501
9.3.3	Using the Class Hierarchy	504
9.3.4	About Object-Oriented Programming	507
9.4	Class Hierarchy for Numerical Methods for ODEs	508

9.4.1	Mathematical Problem	508
9.4.2	Numerical Methods	510
9.4.3	The ODE Solver Class Hierarchy	511
9.4.4	The Backward Euler Method	515
9.4.5	Verification	518
9.4.6	Application 1: $u' = u$	518
9.4.7	Application 2: The Logistic Equation	519
9.4.8	Application 3: An Oscillating System	521
9.4.9	Application 4: The Trajectory of a Ball	523
9.5	Class Hierarchy for Geometric Shapes	525
9.5.1	Using the Class Hierarchy	526
9.5.2	Overall Design of the Class Hierarchy	527
9.5.3	The Drawing Tool	529
9.5.4	Implementation of Shape Classes	530
9.5.5	Scaling, Translating, and Rotating a Figure	534
9.6	Summary	538
9.6.1	Chapter Topics	538
9.6.2	Summarizing Example: Input Data Reader	540
9.7	Exercises	546
A	Discrete Calculus	573
A.1	Discrete Functions	573
A.1.1	The Sine Function	574
A.1.2	Interpolation	576
A.1.3	Evaluating the Approximation	576
A.1.4	Generalization	577
A.2	Differentiation Becomes Finite Differences	579
A.2.1	Differentiating the Sine Function	580
A.2.2	Differences on a Mesh	580
A.2.3	Generalization	582
A.3	Integration Becomes Summation	583
A.3.1	Dividing into Subintervals	584
A.3.2	Integration on Subintervals	585
A.3.3	Adding the Subintervals	586
A.3.4	Generalization	587
A.4	Taylor Series	589
A.4.1	Approximating Functions Close to One Point... ..	589
A.4.2	Approximating the Exponential Function	589
A.4.3	More Accurate Expansions	590
A.4.4	Accuracy of the Approximation	592
A.4.5	Derivatives Revisited	594
A.4.6	More Accurate Difference Approximations	595
A.4.7	Second-Order Derivatives	597
A.5	Exercises	599

B	Differential Equations	605
B.1	The Simplest Case	606
B.2	Exponential Growth	608
B.3	Logistic Growth	612
B.4	A General Ordinary Differential Equation	614
B.5	A Simple Pendulum	615
B.6	A Model for the Spread of a Disease	619
B.7	Exercises	621
C	A Complete Project	625
C.1	About the Problem: Motion and Forces in Physics	626
C.1.1	The Physical Problem	626
C.1.2	The Computational Algorithm	628
C.1.3	Derivation of the Mathematical Model	628
C.1.4	Derivation of the Algorithm	631
C.2	Program Development and Testing	632
C.2.1	Implementation	632
C.2.2	Callback Functionality	635
C.2.3	Making a Module	636
C.2.4	Verification	637
C.3	Visualization	639
C.3.1	Simultaneous Computation and Plotting	639
C.3.2	Some Applications	642
C.3.3	Remark on Choosing Δt	643
C.3.4	Comparing Several Quantities in Subplots	644
C.3.5	Comparing Approximate and Exact Solutions	645
C.3.6	Evolution of the Error as Δt Decreases	646
C.4	Exercises	649
D	Debugging	651
D.1	Using a Debugger	651
D.2	How to Debug	653
D.2.1	A Recipe for Program Writing and Debugging	654
D.2.2	Application of the Recipe	656
E	Technical Topics	669
E.1	Different Ways of Running Python Programs	669
E.1.1	Executing Python Programs in IPython	669
E.1.2	Executing Python Programs on Unix	669
E.1.3	Executing Python Programs on Windows	671
E.1.4	Executing Python Programs on Macintosh	673
E.1.5	Making a Complete Stand-Alone Executable	673
E.2	Integer and Float Division	673
E.3	Visualizing a Program with Lumpy	674
E.4	Doing Operating System Tasks in Python	675
E.5	Variable Number of Function Arguments	678

E.5.1 Variable Number of Positional Arguments 679

E.5.2 Variable Number of Keyword Arguments 681

E.6 Evaluating Program Efficiency 683

E.6.1 Making Time Measurements 683

E.6.2 Profiling Python Programs 685

Bibliography 687

Index 689