A\_Primer\_Scientific\_Programming\_Python\_c00

Third Edition

Hans Petter Langtangen

**Preface**

The aim of this book is to teach computer programming using examples from math- ematics and the natural sciences. We have chosen to use the Python programming language because it combines remarkable expressive power with very clean, simple, and compact syntax. Python is easy to learn and very well suited for an introduction to computer programming. Python is also quite similar to MATLAB and a good language for doing mathematical computing. It is easy to combine Python with compiled languages, like Fortran, C, and C++, which are widely used languages for scientific computations.

El objetivo de este libro es enseñar programación informática utilizando ejemplos de matemáticas y ciencias naturales. Hemos elegido utilizar el lenguaje de programación Python porque combina un notable poder expresivo con una sintaxis muy clara, simple y compacta. Python es fácil de aprender y muy adecuado para una introducción a la programación informática. Python también es bastante similar a MATLAB y un buen lenguaje para realizar cálculos matemáticos. Es fácil combinar Python con lenguajes compilados, como Fortran, C y C++, que son lenguajes ampliamente utilizados para cálculos científicos.

The examples in this book integrate programming with applications to mathe- matics, physics, biology, and finance. The reader is expected to have knowledge of basic one-variable calculus as taught in mathematics-intensive programs in high schools. It is certainly an advantage to take a university calculus course in parallel, preferably containing both classical and numerical aspects of calculus. Although not strictly required, a background in high school physics makes many of the ex- amples more meaningful.

Los ejemplos de este libro integran la programación con aplicaciones a las matemáticas, la física, la biología y las finanzas. Se espera que el lector tenga conocimientos básicos de cálculo de una variable, tal como se enseña en los programas intensivos de matemáticas en las escuelas secundarias. Sin duda, es una ventaja tomar un curso universitario de cálculo en paralelo, preferiblemente que contenga aspectos tanto clásicos como numéricos del cálculo. Aunque no es estrictamente obligatorio, tener conocimientos de física en la escuela secundaria hace que muchos de los ejemplos sean más significativos.

Many introductory programming books are quite compact and focus on listing functionality of a programming language. However, learning to program is learning how to *think* as a programmer. This book has its main focus on the thinking pro- cess, or equivalently: programming as a problem solving technique. That is why most of the pages are devoted to case studies in programming, where we define a problem and explain how to create the corresponding program. New constructions and programming styles (what we could call theory) is also usually introduced via examples. Particular attention is paid to verification of programs and to finding errors. These topics are very demanding for mathematical software, because the unavoidable numerical approximation errors are possibly mixed with programming mistakes.

Muchos libros introductorios a la programación son bastante compactos y se centran en enumerar las funciones de un lenguaje de programación. Sin embargo, aprender a programar es aprender a pensar como programador. Este libro se centra principalmente en el proceso de pensamiento, o lo que es lo mismo: la programación como técnica de resolución de problemas. Por eso, la mayoría de las páginas están dedicadas a estudios de casos de programación, en los que definimos un problema y explicamos cómo crear el programa correspondiente. También se suelen introducir nuevas construcciones y estilos de programación (lo que podríamos llamar teoría) mediante ejemplos. Se presta especial atención a la verificación de programas y a la búsqueda de errores. Estos temas son muy exigentes para el software matemático, porque los inevitables errores de aproximación numérica posiblemente se mezclen con los errores de programación.

By studying the many examples in the book, I hope readers will learn how to think right and thereby write programs in a quicker and more reliable way. Re- member, nobody can learn programming by just reading – one has to solve a large amount of exercises hands on. The book is therefore full of exercises of various types: modifications of existing examples, completely new problems, or debugging of given programs.

Espero que, estudiando los numerosos ejemplos que aparecen en el libro, los lectores aprendan a pensar correctamente y, por lo tanto, a escribir programas de forma más rápida y fiable. Recuerden que nadie puede aprender a programar sólo leyendo: hay que resolver una gran cantidad de ejercicios prácticos. Por eso, el libro está repleto de ejercicios de distintos tipos: modificaciones de ejemplos existentes, problemas completamente nuevos o depuración de programas dados.

To work with this book, I recommend using Python version 2.7. For Chaps. 5–9 and Appendices A–E, you need the NumPy and Matplotlib packages, preferably also the IPython and SciTools packages, and for Appendix G, Cython is required. Other packages used in the text are nose and sympy. Section H.1 has more infor- mation on how you can get access to Python and the mentioned packages.

Para trabajar con este libro, recomiendo utilizar la versión 2.7 de Python. Para los capítulos 5 a 9 y los apéndices A a E, se necesitan los paquetes NumPy y Matplotlib, preferiblemente también los paquetes IPython y SciTools, y para el apéndice G, se requiere Cython. Otros paquetes utilizados en el texto son nose y sympy. La sección H.1 tiene más información sobre cómo obtener acceso a Python y a los paquetes mencionados.

There is a web page associated with this book, http://hplgit.github.io/scipro- primer, containing all the example programs from the book as well as information on installation of the software on various platforms.

**Python version 2 or 3?** A common problem among Python programmers is to choose between version 2 or 3, which at the time of this writing means choosing between version 2.7 and 3.5. A common recommendation is to go for Python 3, because this is the version that will be further developed in the future. However, there is a problem that much useful mathematical software in Python has not yet been ported to Python 3. Therefore, Python version 2.7 is the most popular version for doing scientific computing, and that is why also this book applies version 2.7.

A widely used strategy for software developers who want to write Python code that works with both versions, is to develop a common version for Python 2 and 3. For the programs in this book, a common version can easily be produced by first developing for version 2.7 and then *automatically* convert the code by running the futurize program. Section 4.10 demonstrates how this is done in simple cases.

The Python 2.7 code in this book sticks to all modern constructions that are backported from version 3 such that the code becomes as close as possible to the equivalent Python 3 code. At any time, you can just run futurize to see the differ- ences between your Python 2.7 version and the corresponding Python 3.5 version.

**Contents**

Chapter 1 introduces variables, objects, modules, and text formatting through examples concerning evaluation of mathematical formulas. Chapter 2 presents programming with while and for loops as well as lists, including nested lists. The next chapter deals with two other fundamental concepts in programming: functions and if-else tests.

El capítulo 1 presenta variables, objetos, módulos y formato de texto a través de ejemplos relacionados con la evaluación de fórmulas matemáticas. El capítulo 2 presenta la programación con bucles while y for, así como listas, incluidas las listas anidadas. El capítulo siguiente trata otros dos conceptos fundamentales de la programación: funciones y pruebas if-else.

How to read data into programs and deal with errors in input are the subjects of Chap. 4. Chapter 5 introduces arrays and array computing (including vectorization) and how this is used for plotting y D f .x/ curves and making animation of curves. Many of the examples in the first five chapters are strongly related. Typically, for- mulas from the first chapter are used to produce tables of numbers in the second chapter. Then the formulas are encapsulated in functions in the third chapter. In the next chapter, the input to the functions are fetched from the command line, and validity checks of the input are added. The formulas are then shown as graphs in Chap. 5. After having studied Chaps. 1–5, the reader should have enough knowl- edge of programming to solve mathematical problems by what many refer to as “MATLAB-style” programming.

Chapter 6 explains how to work with dictionaries and strings, especially for in- terpreting text data in files and storing the extracted information in flexible data structures. Class programming, including user-defined types for mathematical com- putations (with overloaded operators), is introduced in Chap. 7. Chapter 8 deals with random numbers and statistical computing with applications to games and random walks. Object-oriented programming, in the meaning of class hierarchies and inheritance, is the subject of Chap. 9. The key examples here deal with building toolkits for numerical differentiation and integration as well as graphics.

Appendix A introduces mathematical modeling, using sequences and differ- ence equations. Only programming concepts from Chaps. 1–5 are used in this appendix, the aim being to consolidate basic programming knowledge and apply it to mathematical problems. Some important mathematical topics are introduced via difference equations in a simple way: Newton’s method, Taylor series, inverse functions, and dynamical systems.

Appendix B deals with functions on a mesh, numerical differentiation, and nu- merical integration. A simple introduction to ordinary differential equations and their numerical treatment is provided in Appendix C. Appendix D shows how a complete project in physics can be solved by mathematical modeling, numerical methods, and programming elements from Chaps. 1–5. This project is a good exam- ple on problem solving in computational science, where it is necessary to integrate physics, mathematics, numerics, and computer science.

How to create software for solving ordinary differential equations, using both function-based and object-oriented programming, is the subject of Appendix E. The material in this appendix brings together many parts of the book in the context of physical applications and differential equations.

Appendix F is devoted to the art of debugging, and in fact problem solving in general. Speeding up numerical computations in Python by migrating code to C via Cython is exemplified in Appendix G. Finally, Appendix H deals with various more advanced technical topics.

Most of the examples and exercises in this book are quite short. However, many of the exercises are related, and together they form larger projects, for example on Fourier Series (3.21, 4.21, 4.22, 5.41, 5.42), numerical integration (3.11, 3.12, 5.49, 5.50, A.12), Taylor series (3.37, 5.32, 5.39, A.14, A.15, 7.23), piecewise constant functions (3.29–3.33, 5.34, 5.47, 5.48, 7.19–7.21), inverse functions (E.17–E.20), falling objects (E.8, E.9, E.38, E.39), oscillatory population growth (A.19, A.21, A.22, A.23), epidemic disease modeling (E.41–E.48), optimization and finance (A.24, 8.42, 8.43), statistics and probability (4.24, 4.25, 8.23, 8.24), hazard games (8.8–8.14), random walk and statistical physics (8.32–8.40), noisy data analysis (8.44–8.46), numerical methods (5.25–5.27, 7.8, 7.9, A.9, 7.22, 9.15–9.17, E.30– E.37), building a calculus calculator (7.34, 9.18, 9.19), and creating a toolkit for simulating vibrating engineering systems (E.50–E.55).

Chapters 1–9 together with Appendices A and E have from 2007 formed the core of an introductory first semester bachelor course on scientific programming at the University of Oslo (INF1100, 10 ECTS credits).

**Changes from the fourth to the fifth edition** Substantialchangeswereintroduced in the fourth edition, and the fifth edition is primarily a consolidation of those changes. Many typos have been corrected and many explanations and exercises have been improved. The emphasis on unit tests and test functions, especially in exercises, is stronger than in the previous edition. Symbolic computation with the aid of SymPy is used to a larger extent and integrated with numerical computing throughout the book. All classes are now new-style (instead of old-style/classic as in previous editions). Examples on Matplotlib do not use the pylab module any- more, but pyplot and MATLAB-like syntax is still favored to ease the transition between Python and MATLAB. The concept of closures is more explicit than in earlier editions (see the new Sect. 7.1.7) since this is a handy and popular construction much used in the scientific Python community. We also discuss the difference between Python 2 and 3 and demonstrate how to use the future module to write code that runs under both versions.

The most substantial new material in the fifth edition appears toward the end of Chap. 5 and regards high-performance computing, linear algebra, and visualization of scalar and vector fields. Although this material is not used elsewhere in the book, many readers have requested basic recipes when going from one to two variables or from vectors to matrices later when solving more advanced problems and using the book as their programming reference. The new matrial in Chap. 5 was written jointly with Dr. Øyvind Ryan.

**Acknowledgments** This book was born out of stimulating discussions with my close colleague Aslak Tveito, and he started writing what is now Appendix B and C. The whole book project and the associated university course were critically de- pendent on Aslak’s enthusiastic role back in 2007. The continuous support from Aslak regarding my book projects is much appreciated and contributes greatly to my strong motivation. Another key contributor in the early days was Ilmar Wilbers. He made extensive efforts with assisting the book project and establishing the uni- versity course INF1100. I feel that without Ilmar and his solutions to numerous technical problems the first edition of the book would never have been completed. Johannes H. Ring also deserves special acknowledgment for the development of the Easyviz graphics tool back in the days when Python plotting was a hassle, and later for his maintenance of software associated with the book.

Professor Loyce Adams studied the entire book, solved all the exercises, found numerous errors, and suggested many improvements. Her contributions are so much appreciated. More recently, Helmut Büch worked extremely carefully through all details in Chaps. 1–6, tested the software, found many typos, and asked critical questions that led to lots of significant improvements. I am so thank- ful for all his efforts and for his enthusiasm during the preparations of the fourth edition. The fifth edition has benefited much from Hakki Eres’ careful examination of the fourth edition. He found several typos and code errors, some of which go back to the first edition.

Special thanks go to Geir Kjetil Sandve for being the primary author of the computational bioinformatics examples in Sects. 3.3, 6.5, 8.3.4, and 9.5, with con- tributions from Sveinung Gundersen, Ksenia Khelik, Halfdan Rydbeck, and Kai Trengereid. I am also greatful to Øyvind Ryan’s work with linear algebra and visu- alization of scalar and vector fields in Chap. 5.

Several people have contributed with suggestions for improvements of the text, the exercises, and the associated software. I will in particular mention Ingrid Eide, Ståle Zerener Haugnæss, Kristian Hiorth, Timothy Keough, Arve Knudsen, Espen Kristensen, Tobias Vidarssønn Langhoff, Martin Vonheim Larsen, Kine Veronica Lund, Solveig Masvie, Håkon Møller, Rebekka Mørken, Mathias Nedrebø, Marit Sandstad, Helene Norheim Semmerud, Lars Storjord, Fredrik Heffer Valdmanis, and Torkil Vederhus. Hakon Adler is greatly acknowledged for his careful reading of early various versions of the manuscript. Many thanks go to the professors Fred Espen Bent, Ørnulf Borgan, Geir Dahl, Knut Mørken, and Geir Pedersen for for- mulating several exciting exercises from various application fields. I also appreciate the cover image made by my good friend Jan Olav Langseth.

This book and the associated course are parts of a comprehensive and successful reform at the University of Oslo, called *Computing in Science Education*. The goal of the reform is to integrate computer programming and simulation in all bachelor courses in natural science where mathematical models are used. The present book lays the foundation for the modern computerized problem solving technique to be applied in later courses. It has been extremely inspiring to work closely with the driving forces behind this reform, especially the professors Morten Hjorth-Jensen, Anders Malthe-Sørenssen, Knut Mørken, and Arnt Inge Vistnes.

The excellent assistance from the Springer system over the years, in particular Martin Peters, Thanh-Ha Le Thi, Ruth Allewelt, Peggy Glauch-Ruge, Nadja Kroke, Thomas Schmidt, Patrick Waltemate, Donatas Akmanavicius, and Yvonne Schlat- ter, is highly appreciated, and ensured a smooth and rapid production of all editions of this book.

Oslo, February 2016 Hans Petter Langtangen

1 Computing with Formulas . . . . . . . . . . . . . . . . . . . . . . . . . . 1

Computación con fórmulas

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

Funciones y ramificaciones

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

3.1.10The 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

Datos de entrada y manejo de errores

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