Python Scripting for Computational Science

Hans Petter Langtangen

Simula Research Laboratory
and
Department of Informatics
University of Oslo

Table of Contents

1 Introduction 1
1.1 Scripting versus Traditional Programming 1
1.1.1 Why Scripting is Useful in Computational Science 2
1.1.2 Classification of Programming Languages 4
1.1.3 Productive Pairs of Programming Languages 5
1.1.4 Gluing Existing Applications 6
1.1.5 Scripting Yields Shorter Code 7
1.1.6 Efficiency 8
1.1.7 Type-Specification (Declaration) of Variables 9
1.1.8 Flexible Function Interfaces 11
1.1.9 Interactive Computing 12
1.1.10 Creating Code at Run Time 13
1.1.11 Nested Heterogeneous Data Structures 14
1.1.12 GUI Programming 16
1.1.13 Mixed Language Programming 17
1.1.14 When to Choose a Dynamically Typed Language 19
1.1.15 Why Python? 20
1.1.16 Script or Program? 21
1.2 Preparations for Working with This Book 22

2 Getting Started with Python Scripting 27
2.1 A Scientific Hello World Script 27
2.1.1 Executing Python Scripts 28
2.1.2 Dissection of the Scientific Hello World Script 29
2.2 Reading and Writing Data Files 32
2.2.1 Problem Specification 32
2.2.2 The Complete Code 33
2.2.3 Dissection 33
2.2.4 Working with Files in Memory 36
2.2.5 Efficiency Measurements 37
2.2.6 Exercises 38
2.3 Automating Simulation and Visualization 40
2.3.1 The Simulation Code 41
2.3.2 Using Gnuplot to Visualize Curves 43
2.3.3 Functionality of the Script 44
2.3.4 The Complete Code 45
2.3.5 Dissection 47
2.3.6 Exercises 49
2.4 Conducting Numerical Experiments 52
2.4.1 Wrapping a Loop Around Another Script 53

2.4.2 Generating an HTML Report . . . 54
2.4.3 Making Animations . . . 56
2.4.4 Varying Any Parameter . . . 57
2.4.5 Exercises . . . 60
2.5 File Format Conversion . . . 60
2.5.1 The First Version of the Script . . . 61
2.5.2 The Second Version of the Script . . . 62

3 Basic Python . . . 65
3.1 Introductory Topics . . . 65
3.1.1 Recommended Python Documentation . . . 65
3.1.2 Testing Statements in the Interactive Shell . . . 67
3.1.3 Control Statements . . . 68
3.1.4 Running an Application . . . 69
3.1.5 File Reading and Writing . . . 70
3.1.6 Output Formatting . . . 72
3.1.7 Exercises . . . 73
3.2 Variables of Different Types . . . 74
3.2.1 Boolean Types . . . 74
3.2.2 The None Variable . . . 75
3.2.3 Numbers and Numerical Expressions . . . 76
3.2.4 Lists and Tuples . . . 78
3.2.5 Dictionaries . . . 84
3.2.6 Splitting and Joining Text . . . 87
3.2.7 String Operations . . . 88
3.2.8 Text Processing . . . 89
3.2.9 The Basics of a Python Class . . . 91
3.2.10 Determining a Variable's Type . . . 93
3.2.11 Exercises . . . 96
3.3 Functions . . . 101
3.3.1 Keyword Arguments . . . 102
3.3.2 Doc Strings . . . 103
3.3.3 Variable Number of Arguments . . . 103
3.3.4 Call by Reference . . . 104
3.3.5 Treatment of Input and Output Arguments . . . 106
3.3.6 Function Objects . . . 107
3.4 Working with Files and Directories . . . 108
3.4.1 Listing Files in a Directory . . . 109
3.4.2 Testing File Types . . . 109
3.4.3 Copying and Renaming Files . . . 110
3.4.4 Removing Files and Directories . . . 111
3.4.5 Splitting Pathnames . . . 111
3.4.6 Creating and Moving to Directories . . . 112
3.4.7 Traversing Directory Trees . . . 112
3.4.8 Exercises . . . 115

4 **Numerical Computing in Python** 121
4.1 A Quick NumPy Primer 123
4.1.1 Creating Arrays 123
4.1.2 Array Indexing 124
4.1.3 Array Computations 126
4.1.4 Type Testing 127
4.1.5 Hidden Temporary Arrays 129
4.1.6 Exercises 130
4.2 Vectorized Algorithms 131
4.2.1 Arrays as Function Arguments 132
4.2.2 Slicing 133
4.2.3 Remark on Efficiency 134
4.2.4 Exercises 136
4.3 More Advanced Array Computing 137
4.3.1 Random Numbers 137
4.3.2 Linear Algebra 139
4.3.3 The Gnuplot Module 139
4.3.4 Example: Curve Fitting 142
4.3.5 Arrays on Structured Grids 144
4.3.6 File I/O with NumPy Arrays 146
4.3.7 Reading and Writing Tables with NumPy Arrays 147
4.3.8 Functionality in the Numpytools Module 150
4.3.9 Exercises 152
4.4 Other Tools for Numerical Computations 156
4.4.1 The ScientificPython Package 156
4.4.2 The SciPy Package 161
4.4.3 The Python–Matlab Interface 165
4.4.4 Some Useful Python Modules 166
4.5 A Database for NumPy Arrays 167
4.5.1 The Structure of the Database 168
4.5.2 Pickling 170
4.5.3 Formatted ASCII Storage 171
4.5.4 Shelving 172
4.5.5 Comparing the Various Techniques 173

5 **Combining Python with Fortran, C, and C++** 175
5.1 About Mixed Language Programming 175
5.1.1 Applications of Mixed Language Programming 176
5.1.2 Calling C from Python 176
5.1.3 Automatic Generation of Wrapper Code 178
5.2 Scientific Hello World Examples 180
5.2.1 Combining Python and Fortran 181
5.2.2 Combining Python and C 186
5.2.3 Combining Python and C++ Functions 192
5.2.4 Combining Python and C++ Classes 194

5.2.5 Exercises 198
5.3 A Simple Computational Steering Example 198
5.3.1 Modified Time Loop for Repeated Simulations 199
5.3.2 Creating a Python Interface 200
5.3.3 The Steering Python Script 202
5.3.4 Equipping the Steering Script with a GUI 205
5.4 Scripting Interfaces to Large Libraries 207

6 Introduction to GUI Programming 211
6.1 Scientific Hello World GUI 211
6.1.1 Introductory Topics 211
6.1.2 The First Python/Tkinter Encounter 214
6.1.3 Binding Events 217
6.1.4 Changing the Layout 218
6.1.5 The Final Scientific Hello World GUI 223
6.1.6 An Alternative to Tkinter Variables 224
6.1.7 About the Pack Command 225
6.1.8 An Introduction to the Grid Geometry Manager 227
6.1.9 Implementing a GUI as a Class 229
6.1.10 A Simple Graphical Function Evaluator 231
6.1.11 Exercises 233
6.2 Adding GUIs to Scripts 235
6.2.1 A Simulation and Visualization Script with a GUI 235
6.2.2 Improving the Layout 238
6.2.3 Exercises 241
6.3 A List of Common Widget Operations 242
6.3.1 Frame 245
6.3.2 Label 245
6.3.3 Button 247
6.3.4 Text Entry 247
6.3.5 Balloon Help 249
6.3.6 Option Menu 250
6.3.7 Slider 250
6.3.8 Check Button 251
6.3.9 Making a Simple Megawidget 251
6.3.10 Menu Bar 252
6.3.11 List Data 254
6.3.12 Listbox 255
6.3.13 Radio Button 258
6.3.14 Combo Box 259
6.3.15 Message Box 260
6.3.16 User-Defined Dialogs 262
6.3.17 Color-Picker Dialogs 263
6.3.18 File Selection Dialogs 266
6.3.19 Toplevel 267

6.3.20 Some Other Types of Widgets . . . 268
6.3.21 Adapting Widgets to the User's Resize Actions . . . 269
6.3.22 Customizing Fonts and Colors . . . 271
6.3.23 Widget Overview . . . 273
6.3.24 Exercises . . . 275

7 Web Interfaces and CGI Programming . . . 281
7.1 Introductory CGI Scripts . . . 282
7.1.1 Web Forms and CGI Scripts . . . 283
7.1.2 Generating Forms in CGI Scripts . . . 285
7.1.3 Debugging CGI Scripts . . . 287
7.1.4 Security Issues . . . 289
7.1.5 A General Shell Script Wrapper for CGI Scripts . . . 290
7.2 Making a Web Interface to a Script . . . 292
7.2.1 A Class for Form Parameters . . . 292
7.2.2 Calling Other Programs . . . 295
7.2.3 Running Simulations . . . 296
7.2.4 Getting a CGI Script to Work . . . 297
7.2.5 Using Web Services from Scripts . . . 300
7.2.6 Exercises . . . 302

8 Advanced Python . . . 305
8.1 Miscellaneous Topics . . . 305
8.1.1 Parsing Command-Line Arguments . . . 305
8.1.2 Platform-Dependent Operations . . . 308
8.1.3 Run-Time Generation of Code . . . 309
8.1.4 Exercises . . . 310
8.2 Regular Expressions and Text Processing . . . 311
8.2.1 Motivation . . . 312
8.2.2 Special Characters . . . 315
8.2.3 Regular Expressions for Real Numbers . . . 316
8.2.4 Using Groups to Extract Parts of a Text . . . 320
8.2.5 Extracting Interval Limits . . . 320
8.2.6 Extracting Multiple Matches . . . 325
8.2.7 Splitting Text . . . 329
8.2.8 Pattern-Matching Modifiers . . . 330
8.2.9 Substitution and Backreferences . . . 333
8.2.10 Example: Swapping Arguments in Function Calls . . . 333
8.2.11 A General Substitution Script . . . 337
8.2.12 Debugging Regular Expressions . . . 338
8.2.13 Exercises . . . 340
8.3 Tools for Handling Data in Files . . . 350
8.3.1 Writing and Reading Python Data Structures . . . 350
8.3.2 Pickling Objects . . . 352
8.3.3 Shelving Objects . . . 354

8.3.4 Writing and Reading Zip Archive Files . . . 355
8.3.5 Downloading Internet Files . . . 356
8.3.6 Binary Input/Output . . . 357
8.3.7 Exercises . . . 359
8.4 Scripts Involving Local and Remote Hosts . . . 359
8.4.1 Secure Shell Commands . . . 360
8.4.2 Distributed Simulation and Visualization . . . 361
8.4.3 Client/Server Programming . . . 363
8.4.4 Threads . . . 363
8.5 Classes . . . 365
8.5.1 Class Programming . . . 365
8.5.2 Checking the Class Type . . . 369
8.5.3 Private Data . . . 370
8.5.4 Static Data . . . 370
8.5.5 Special Attributes . . . 371
8.5.6 Special Methods . . . 371
8.5.7 Multiple Inheritance . . . 373
8.5.8 Using a Class as a C-like Structure . . . 373
8.5.9 Attribute Access via String Names . . . 374
8.5.10 Example: Turning String Formulas into Functions . . . 375
8.5.11 Example: Class for Structured Grids . . . 376
8.5.12 New-Style Classes . . . 379
8.5.13 Implementing Get/Set Functions via Properties . . . 379
8.5.14 Subclassing Built-in Types . . . 381
8.5.15 Copy and Assignment . . . 383
8.5.16 Building Class Interfaces at Run Time . . . 386
8.5.17 Building Flexible Class Interfaces . . . 389
8.5.18 Exercises . . . 396
8.6 Scope of Variables . . . 399
8.6.1 Global, Local, and Class Variables . . . 399
8.6.2 Nested Functions . . . 401
8.6.3 Dictionaries of Variables in Namespaces . . . 402
8.7 Exceptions . . . 404
8.7.1 Handling Exceptions . . . 405
8.7.2 Raising Exceptions . . . 406
8.8 Iterators . . . 407
8.8.1 Constructing an Iterator . . . 407
8.8.2 A Pointwise Grid Iterator . . . 409
8.8.3 A Vectorized Grid Iterator . . . 413
8.8.4 Generators . . . 415
8.8.5 Some Aspects of Generic Programming . . . 417
8.9 Investigating Efficiency . . . 421
8.9.1 CPU-Time Measurements . . . 421
8.9.2 Profiling Python Scripts . . . 424
8.9.3 Optimization of Python Code . . . 425

9 Fortran Programming with NumPy Arrays 429
9.1 Problem Definition 429
9.2 Filling an Array in Fortran 432
9.2.1 The Fortran Subroutine 432
9.2.2 Building and Inspecting the Extension Module 433
9.3 Array Storage Issues 435
9.3.1 Generating an Erroneous Interface 435
9.3.2 Array Storage in C and Fortran 437
9.3.3 Input and Output Arrays as Function Arguments 438
9.3.4 F2PY Interface Files 444
9.3.5 Hiding Work Arrays 448
9.4 Increasing Callback Efficiency 449
9.4.1 Callbacks to Vectorized Python Functions 449
9.4.2 Avoiding Callbacks to Python 452
9.4.3 Compiled Inline Callback Functions 453
9.5 Summary 456
9.6 Exercises 456
9.6.1 Exercises 456

10 C and C++ Programming with NumPy Arrays 461
10.1 C Programming with NumPy Arrays 462
10.1.1 Basics of the NumPy C API 462
10.1.2 The Handwritten Extension Code 464
10.1.3 Sending Arguments from Python to C 465
10.1.4 Consistency Checks 466
10.1.5 Computing Array Values 467
10.1.6 Returning an Output Array 469
10.1.7 Convenient Macros 470
10.1.8 Module Initialization 471
10.1.9 Extension Module Template 472
10.1.10 Compiling, Linking, and Debugging the Module 474
10.1.11 Writing a Wrapper for a C Function 475
10.2 C++ Programming with NumPy Arrays 478
10.2.1 Wrapping a NumPy Array in a C++ Object 479
10.2.2 Using SCXX 481
10.2.3 NumPy–C++ Class Conversion 483
10.3 Comparison of the Implementations 492
10.3.1 Efficiency 492
10.3.2 Error Handling 495
10.3.3 Summary 496
10.3.4 Exercises 497

11 More Advanced GUI Programming 503
11.1 Adding Plot Areas in GUIs 503
11.1.1 The BLT Graph Widget 504
11.1.2 Animation of Functions in BLT Graph Widgets 510
11.1.3 Other Tools for Making GUIs with Plots 512
11.1.4 Exercises 514
11.2 Event Bindings 517
11.2.1 Binding Events to Functions with Arguments 517
11.2.2 A Text Widget with Tailored Keyboard Bindings 520
11.2.3 A Fancy List Widget 523
11.3 Animated Graphics with Canvas Widgets 526
11.3.1 The First Canvas Encounter 527
11.3.2 Coordinate Systems 528
11.3.3 The Mathematical Model Class 532
11.3.4 The Planet Class 533
11.3.5 Drawing and Moving Planets 535
11.3.6 Dragging Planets to New Positions 536
11.3.7 Using Pmw's Scrolled Canvas Widget 540
11.4 Tools for Simulation & Visualization Scripts 542
11.4.1 Restructuring the Script 543
11.4.2 Representing a Parameter by a Class 545
11.4.3 Improved Command-Line Script 559
11.4.4 Improved GUI Script 560
11.4.5 Improved CGI Script 561
11.4.6 Parameters with Physical Dimensions 562
11.4.7 Adding a Curve Plot Area 564
11.4.8 Automatic Generation of Scripts 566
11.4.9 Applications of the Tools 567
11.4.10 Allowing Physical Units in Input Files 572
11.4.11 Converting Input Files to GUIs 576

12 Tools and Examples 579
12.1 Running Series of Computer Experiments 579
12.1.1 Multiple Values of Input Parameters 580
12.1.2 Implementation Details 583
12.1.3 Further Applications 588
12.2 Tools for Representing Functions 592
12.2.1 Functions Defined by String Formulas 592
12.2.2 A Unified Interface to Functions 594
12.2.3 Interactive Drawing of Functions 600
12.2.4 A Notebook for Selecting Functions 606
12.3 Solving Partial Differential Equations 612
12.3.1 Numerical Methods for 1D Wave Equations 613
12.3.2 Implementations of 1D Wave Equations 616
12.3.3 Classes for Solving 1D Wave Equations 622

12.3.4 A Problem Solving Environment 629
12.3.5 Numerical Methods for 2D Wave Equations 635
12.3.6 Implementations of 2D Wave Equations 638

A Setting up the Required Software Environment 649
A.1 Installation on Unix Systems 649
A.1.1 A Suggested Directory Structure 650
A.1.2 Setting Some Environment Variables 650
A.1.3 Installing Tcl/Tk and Additional Modules 651
A.1.4 Installing Python 652
A.1.5 Installing Python Modules 654
A.1.6 Installing Gnuplot 658
A.1.7 Installing SWIG 658
A.1.8 Summary of Environment Variables 659
A.1.9 Testing the Installation of Scripting Utilities 659
A.2 Installation on Windows Systems 660

B Elements of Software Engineering 665
B.1 Building and Using Modules 665
B.1.1 Single-File Modules 665
B.1.2 Multi-File Modules 668
B.1.3 Debugging and Troubleshooting 670
B.2 Tools for Documenting Python Software 672
B.2.1 Doc Strings 673
B.2.2 Tools for Automatic Documentation 674
B.3 Coding Standards 678
B.3.1 Style Guide 678
B.3.2 Pythonic Programming 682
B.4 Verification of Scripts 687
B.4.1 Automating Regression Tests 687
B.4.2 Implementing a Tool for Regression Tests 692
B.4.3 Writing a Test Script 695
B.4.4 Verifying Output from Numerical Computations 696
B.4.5 Automatic Doc String Testing 700
B.4.6 Unit Testing 702
B.5 Version Control Management 704
B.5.1 Getting Started with CVS 705
B.5.2 Building Scripts to Simplify the Use of CVS 709
B.6 Exercises 709