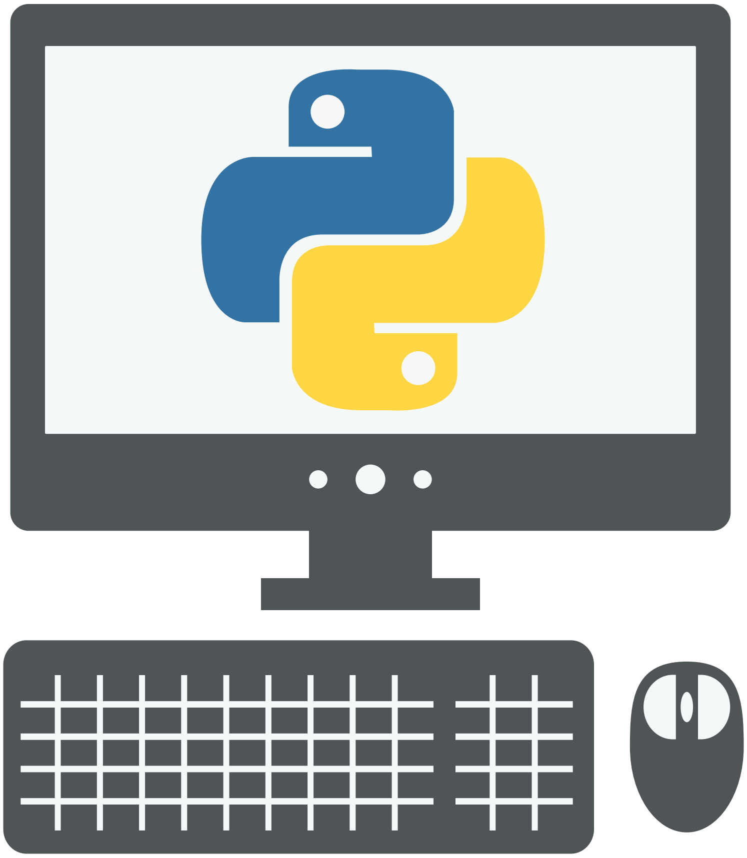

Perfectly Python

Michael Jay Schillaci | Harris Corporation

J. Wang, *Computational Modeling and Visualization of Physical Systems with Python*, Wiley, 2016.

Written in a no-nonsense style, Jay Wang's *Computational Modeling and Visualization of Physical Systems with Python* provides the ever-growing community of Python learners and users with a thorough text and valuable resource. Ideally suited for use as a course text in advanced undergraduate or graduate curricula, this relatively slim volume offers coverage that is broad and deep, and the presentation is well thought out. With standard topics ranging from planetary motion, dynamics, and chaos to oscillation, waves, electricity, and magnetism, the text also includes excellent chapters on simple random problems and thermal systems. It distinguishes itself from others by also covering both time-dependent and time-independent methods in quantum mechanics as well as classical and quantum scattering. The author wisely deals with the problem of delivering sufficient depth of coverage by providing tutorial-like examples, complete code listings, and exercises at the end of each chapter.

The introductory chapter provides a wealth of information for the novice programmer. It begins with an overview of the syntax and control structures of the interpretive language and also discusses file handling, compatibility concerns for 2.7x and 3.xx conversion, and speed and extensibility (via Numba, Cython, F2Py, and so on), issues that will appeal to the Python acolyte. All programmers will be happy that the text also includes material on plotting with Matplotlib and Numpy arrays with explicit code examples for creation, indexing, and slicing. The discussion on multidimensional arrays and universal functions is terse, but its inclusion points to the volume's usefulness as both a textbook and as a handy reference resource for programmers

As some of my own research is in classical and post-Newtonian theories of gravity, I found the text's treatment of orbital motion to be of particular interest. The author does a wonderful job of presenting the basic physics and then discussing the importance of units and scales before introducing the Runge-Lenz vector.

and researchers. Indeed, the entire text is replete with notes on debugging and optimization, and its many references to appendices and online resources makes for a tight if sometimes dense read.

The integration of the Python code and the nascent background material is seamless and well accomplished. From a pedagogical viewpoint, the author's delivery is excellent, and the chapter structure generally progresses from coverage of the theoretical background (equations) to the logical steps needed for a programmed solution. These are often given in numbered or bulleted lists (pseudo code), a feature that again points to the volume's ease of use for beginning students. Some readers will find the background material rife with vectors and other dense mathematical notation, so educators who use the text might have to unpack some of it for their students. Where the book really delivers is that most of the problems posed in the main text include a full implementation for a coded solution and visualization of the results.

The beginning chapters dealing with classical problems, such as projectile motion with and without resistance, do an excellent job of covering the basic physics but also demonstrate the rationale behind some of the most widely used numerical methods (such as Runge-Kutta and leapfrog) so that students are prepared to use these methods to solve systems of equations efficiently. These goals, of course, require some straightforward use of arrays and visualization, and the text stands out by providing tight and extensible code and also by demonstrating simple methods for creating scenes that can greatly enhance the final product.

As some of my own research is in classical and post-Newtonian theories of gravity, I found the text's treatment of orbital motion to be of particular interest. The author does a wonderful job of presenting the basic physics and then discussing the importance of units and scales before introducing the Runge-Lenz vector. He then takes the time to discuss why time transformations via step-size adjustments must be used. The volume also goes well beyond the basics as exemplified by the inclusion of material on precession due to relativistic corrections. Here, the author walks the reader through straightforward methods to create animations and plots that demonstrate the success of Einstein's theory in explaining the precession of Mercury, introducing readers to advanced methods using radial velocity datasets. While these points might not be of interest to all readers, this same care and attention to detail—and extension beyond the basics—is found throughout the book.

For those readers more interested in how programming aspects come to bear on the scientific content covered, I found many gems while reading through the volume and running many of the programs. For example, in visualizing harmonic motion, the author first introduces standard numerical methods and power spectrum and then uses SciPy's linear algebra modules to demonstrate matrix diagonalization methods and normal mode identification. He then takes the opportunity to extend these methods to model the displacement of a string under a load and demonstrates how the Lapack routines are used to solve the resulting linear system of equations. This sequential and cumulative approach makes the book more accessible to novice programmers and more relevant to those who are more advanced. In this regard, the bulk of the provided code is well documented and expertly written. The author almost always utilizes procedural programming methods by implementing a series of functions, and most speed concerns are mitigated with the @ jit compiler decorator. In some cases, object-oriented methods are implemented, and the bulk of reading and writing of data files is handled using the pickle module.

The scientific scope of the book will likely be intimidating to some readers. This is particularly true with the material on electromagnetic phenomena where finite-element, mesh, and interpolation methods are discussed, as these methods assume a level of mathematical knowledge and sophistication that many (undergraduate) readers might not have. Significantly, the inclusion of the system matrix and mesh generation material in this chapter allows more robust and realistic simulations to be developed throughout the remainder of the text. The sophistication of the numerical techniques will also challenge some readers. For example, when tackling important problems in quantum mechanics, a spatially discretized leapfrog method is introduced. The text does a good job of outlining the reasons and steps needed to implement this algorithm and then applies it to the quantum oscillator and free-fall problems. In his characteristic fashion, the author then extends the discussion beyond the scope of most other books and also demonstrates methods for visualizing the position and momentum space amplitudes for 2D quantum waves.

It's also worth noting that the text's material on simple random systems does a great job of introducing a stochastic model for Brownian motion and also discusses importance sampling methods. The text illustrates its depth here by delving into statistical theory and giving a comprehensive treatment of nonuniform distributions, providing a guided project to show how these can be generated via the rejection method. With similar completeness, the author covers more advanced material on thermodynamic and scattering processes, and develops code that's relevant to many areas of pure and applied research.

With exhaustive, wide-ranging, and careful coverage of many areas in classical, quantum, and relativistic physics, Jay Wang's book deserves to become a standard in undergraduate and graduate curricula where scientific computing plays a role. Its collection of expertly documented and written Python code also makes it an ideal desk reference for programmers and researchers alike that is, in short, perfectly Python. ■

Michael Jay Schillaci is the former managing director of the McCausland Center for Brain Imaging at the University of South Carolina, served as associate professor of physics at Roberts Wesleyan College, and currently works as an imaging scientist and software engineer in the geospatial systems division at Harris Corporation. His research interests include computational physics, cognitive neuroscience, and classical theories of gravity. Schillaci has a PhD in physics from the University of Arkansas at Fayetteville. He's a member of the American Physical Society. Contact him at Michael.Schillaci98@gmail.com.