Editors: Larry Engelhardt, lengelhardt@fmarion.edu | Stephen P. Weppner, weppnesp@eckerd.edu

# Numerical Literacy for Physics Undergraduates

R. Steven Turley | Brigham Young University

*Computational Methods for Physics* should be seriously considered as a textbook by anyone teaching computational physics to upper-division physics students. It would be one of my top choices to teach an upper-division computational physics class among some other books, such as *Numerical Method for Physics* by Alejandro L. Garcia, *Computational Physics* by Nicholas J. Giordano, and *Mathematica for Physics* by Robert L. Zimmerman and Fredrick I. Olness. The book includes good coverage of the standard techniques usually covered in a numerical methods course: ordinary differential equations, root finding, partial differential equations, integration, Fourier transforms, linear algebra, and minimization with the addition of a few topics of special interest to physicists—chaos, neural networks, and Galerkin methods. Almost all of the techniques are couched in the context of physics examples from the areas of relativistic and classical mechanics, electricity and magnetism, quantum mechanics, thermodynamics, and fluid dynamics.

## Exercises and Problems

To be useful, any computational physics textbook needs good exercises and problems to give students practice applying the ideas presented in each chapter. Franklin's book doesn't disappoint in this regard. Each chapter has a rich set of two types of exercises: pencil-and-paper problems, to give students practice in the ideas behind the techniques; and laboratory problems, to give students practice implementing the techniques covered in each chapter. The problems are nontrivial, at a good level to challenge upper-division physics students, and tied nicely to the ideas in each chapter.

For example, in the chapter on eigenvalues there are pencil and paper problems to do things like proving that a real symmetric matrix has real eigenvalues or finding approximate energies for an infinite square well with a small perturbation. The laboratory problems involve things like finding least squares fits to noisy data, solving the Schrödinger equation numerically, or finding the numerical solution to the same perturbed infinite square well problem.

Joel Franklin, *Computational Methods for Physics*, Cambridge Univ. Press, 2013, ISBN 978-1-107-03430-3, 414 pp.

## Organization and Approach

The book is organized around the techniques themselves rather than around the physical problems. So, for example, the chapters have headings like root-finding, integration, Fourier transform, and matrix inversion. This is in contrast to a textbook like Giordano's "Computational Physics" which is organized around particular physical problems. In this respect it is more like Garcia's "Numerical Methods for Physics" book.

The author takes a somewhat unique approach in terms of showing how to implement each technique. Although he chose to use the Mathematica language, he limited himself to a small subset of the rich set of tools Mathematica has for solving these kinds of problems. By limiting himself to this subset of procedural statements, he makes it relatively easy to translate the code examples into other languages such as Python, C, C++, Java, or Fortran. By avoiding the powerful tools already in Mathematica for solving these kinds of problems, he helps the students better understand the inner workings and applicability of each of the methods discussed. The downside of such an approach is that the students aren't exposed to tools they might find useful for their own research or how the problems might be solved more elegantly using a more conventional Mathematica style, such as that employed in Zimmerman and Olness's book, *Mathematica for Physics*.

Overall, Franklin has made the book quite readable; his enthusiasm and interest in the subject help the reader understand why each topic is interesting and important. The context of numerous physics examples will help physics students understand their importance. Generally, both physics and mathematical ideas are presented with enough background so that an average upper-division student can follow the development even if they haven't covered that material in a course before. I would recommend that the student have some programming experience, introductory courses in mechanics, thermal physics, and electricity and magnetism, as well as modern physics, differential equations, and partial differential equations before taking a course using this text.

The book has a website with electronic copies of the examples in the book and example projects from students. This is an important resource for the large fraction of physics students who learn computational techniques better by example than by explanation.

## Content

The textbook begins with an introductory chapter containing an overview of programming. It serves as a good introduction to the subset of Mathematica used in the book and introduces the important ideas of scaling units and analyzing computational complexity and efficiency. However, I believe a student with no prior programming background would struggle a bit if this were his or her first exposure to programming. Instead, this chapter probably serves best as a review of the ideas that students have already encountered in previous programming experiences.

The book contains a solid introduction to numerical issues that can plague an uninformed novice to numerical methods. The careful reader will understand how to analyze the convergence of methods, issues involving algorithm stability, how to ensure flux conservation, and the importance of comparing numerical solutions to canonical problems that can be solved analytically.

The book focuses on broad ideas and the most basic techniques for each area covered. A student who wanted to apply these techniques in a research project would need to utilize the suggestions for further reading in each chapter or go to the literature to learn about the most efficient or robust algorithms or about critical details for particular applications. For example, the discussion about dealing with finite grids in solving the Poisson equation includes little detail about how to effectively address the issue of boundary conditions at the edge of the grid.

As is inevitable with an introductory text of this sort, the author had to pick and choose which details he would include. If I were teaching a course from this text, I would want to supplement the text with more information about high-order quadrature techniques, signal processing, high-order surface representations, statistical error analysis, Monte Carlo simulations, additional minimization techniques (such as genetic algorithms and simulated annealing), and extending Galerkin methods to boundary integral equations. However, this isn't a serious weakness. The book provides a solid foundation on which these topics (or the favorite topics of other computational physicists) can easily be built.

This would be a good textbook for providing junior- and senior-level undergraduate students with a foundational background in numerical techniques as applied to physics. With its strong set of physical examples and the author's obvious enthusiasm for this subject, I believe it will be an engaging textbook for physics students, which will also serve as a good review or preview of foundational physics topics. Most of the problems introduced by the author are intractable analytically, giving the reader good motivation for mastering the computational techniques the author presents to expand their toolkit for solving important physical problems. With the author's approach of using a simple subset of Mathematica to illustrate implementations of basic algorithms, students will be able to focus on the essential elements associated with each approach, rather than getting lost in the minutia of more sophisticated procedures and more elegant implementations. ■

**R. Steven Turley** is a professor of physics at Brigham Young University (BYU), where he has taught undergraduate and graduate courses in general physics, modern physics, quantum mechanics, and computational physics (using Mathematica). Turley's research has been in the areas of nuclear physics, space physics, optics, atomic physics, and computational electromagnetics, and his computational research has used a wide variety of languages, applied to a wide variety of problems. Turley has a PhD in physics from MIT. Contact him at turley@byu.edu.