Numerical Methods in Physics with Python

Bringing together idiomatic Python programming, foundational numerical methods, and physics applications, this is an ideal standalone textbook for courses on computational physics. All the frequently used numerical methods in physics are explained, including foundational techniques and hidden gems on topics such as linear algebra, differential equations, root-finding, interpolation, and integration. Accompanying the mathematical derivations are full implementations of dozens of numerical methods in Python, as well as more than 250 end-of-chapter problems. Numerical methods and physics examples are clearly separated, allowing this introductory book to be later used as a reference; the penultimate section in each chapter is an in-depth project, tackling physics problems that cannot be solved without the use of a computer. Written primarily for students studying computational physics, this textbook brings the non-specialist quickly up to speed with Python before looking in detail at the numerical methods often used in the subject.

Alex Gezerlis is Associate Professor of Physics at the University of Guelph. Before moving to Canada, he worked in Germany, the United States, and Greece. He has received several research awards, grants, and allocations on supercomputing facilities and is active in teaching at both undergraduate and graduate levels.

"I enthusiastically recommend *Numerical Methods in Physics with Python* by Professor Gezerlis to any advanced undergraduate or graduate student who would like to acquire a solid understanding of the basic numerical methods used in physics. The methods are demonstrated with Python, a relatively compact, accessible computer language, allowing the reader to focus on understanding how the methods work rather than on how to program them. Each chapter offers a self-contained, clear, and engaging presentation of the relevant numerical methods, and captivates the reader with well-motivated physics examples and interesting physics projects. Written by a leading expert in computational physics, this outstanding textbook is unique in that it focuses on teaching basic numerical methods while also including a number of modern numerical techniques that are usually not covered in computational physics textbooks."

Yoram Alhassid, *Yale University*

"In *Numerical Methods in Physics with Python* by Gezerlis, one finds a resource that has been sorely missing! As the usage of Python has become widespread, it is too often the case that students take libraries, functions, and codes and apply them without a solid understanding of what is truly being done 'under the hood' and why. Gezerlis' book fills this gap with clarity and rigor by covering a broad number of topics relevant for physics, describing the underlying techniques and implementing them in detail. It should be an important resource for anyone applying numerical techniques to study physics."

Luis Lehner, *Perimeter Institute*

"Gezerlis' text takes a venerable subject – numerical techniques in physics – and brings it up to date and makes it accessible to modern undergraduate curricula through a popular, open-source programming language. Although the focus remains squarely on numerical techniques, each new lesson is motivated by topics commonly encountered in physics and concludes with a practical hands-on project to help cement the students' understanding. The net result is a textbook which fills an important and unique niche in pedagogy and scope, as well as a valuable reference for advanced students and practicing scientists."

Brian Metzger, *Columbia University*

Numerical Methods in Physics with Python

ALEX GEZERLIS
University of Guelph

CAMBRIDGE UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

477 Williamstown Road, Port Melbourne, VIC 3207, Australia

314-321, 3rd Floor, Plot 3, Splendor Forum, Jasola District Centre, New Delhi - 110025, India

103 Penang Road, #05-06/07, Visioncrest Commercial, Singapore 238467

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9781108488846
DOI: 10.1017/9781108772310

First published 2020

A catalogue record for this publication is available from the British Library

ISBN 978-1-108-48884-6 Hardback
ISBN 978-1-108-73893-4 Paperback

Additional resources for this publication at www.cambridge.org/gezerlis

For Marcos

A human is a shadow's dream.
But when radiance from the heavens comes,
a glowing splendor upon mankind, and serene life.

Pindar

Contents

Preface

What we have loved,
Others will love, and we will teach them how.

William Wordsworth

This is a textbook for advanced undergraduate (or beginning graduate) courses on subjects such as Computational Physics, Computational Methods in Physics and Astrophysics, and so on. This could mean a number of things, so I start by going over what this book is *not*.

First, this is not a text that focuses mainly on physics applications and basic programming, only bringing up numerical methods as the need arises. It's true that such an approach would have the benefit of giving rise to beautiful visualizations and helping students gain confidence in using computers to study science. The disadvantage of this approach is that it tends to rely on external libraries, i.e., "black boxes". To make an analogy with non-computational physics, we teach students calculus before seeing how it helps us do physics. In other words, an instructor would not claim that derivatives are important but already well-studied, so we'll just employ a package that takes care of them. That being said, a physics-applications-first approach may be appropriate for a more introductory course (the type with a textbook that has the answers in the back) or perhaps as a computational addendum to an existing text on mechanics, electromagnetism, and so on.

Second, this is not a simulation-oriented textbook, revolving around the technology used in modern computational methods (e.g., molecular dynamics or Monte Carlo). This would have had the advantage of being intimately connected to actual research, but the disadvantage that it would assume students have picked up the necessary foundational material from elsewhere. To return to the analogy with non-computational physics, an introductory course on electromagnetism would never skip over things like basic electrostatics to get directly to, say, the Yang–Mills Lagrangian (assuming students will learn any necessary background material on their own) just because non-abelian gauge theory is more "current". Even so, an approach that focuses on modern computational technology is relevant to a more advanced course: once students have mastered the foundations, they can turn to state-of-the-art methods that tackle research problems.

The present text attempts to strike a happy medium: elementary numerical methods are studied in detail and then applied to questions from undergraduate physics, via idiomatic implementations in the Python programming language. When selecting and discussing topics, I have tried to prioritize pedagogy over novelty; this is reflected in the chapter titles, which are pretty standard. Of course, my views on what is pedagogically superior are mine alone, so the end result also happens to be original in certain aspects. Below, I touch upon some of the main features of this book, with a view to orienting the reader.

- **Idiomatic Python:** the book employs Python 3, which is a popular, open-source programming language. A pedagogical choice I have made is to start out with standard Python, use it for a few chapters, and only then turn to the NumPy library; I have found that this helps students who are new to programming in Python effectively distinguish between lists and NumPy arrays. The first chapter includes a discussion of modern programming idioms, which allow me to write shorter codes in the following chapters, thereby emphasizing the numerical method over programming details. This is somewhat counterintuitive: teaching more "advanced" programming than is usual in computational-physics books, allows the programming to recede into the background. In other words, not having to fight with the programming language every step of the way makes it *easier* to focus on the physics (or the math).
- **Modern numerical analysis techniques:** I devote an entire chapter to questions of numerical precision and roundoff error; I hope that the lessons learned there will pay off when studying the following chapters, which typically focus more on approximation-error themes. While this is not a volume on numerical analysis, it does contain a bit more on applied math than is typical: in addition to standard topics, this also includes modern techniques that haven't made it to computational-physics books before (e.g., compensated summation, automatic differentiation, or interpolation at Chebyshev points). Similarly, the section on errors in linear algebra glances toward monographs on matrix perturbation theory. To paraphrase Forman Acton [2], the idea here is to ensure that the next generation does not think that an obligatory decimal point is slightly demeaning.
- **Methods "from scratch":** chapters typically start with a pedagogical discussion of a crude algorithm and then advance to more complex methods, in several cases also covering state-of-the-art techniques (when they do not require elaborate bookkeeping). Considerable effort is expended toward motivating and explaining each technique as it is being introduced. Similarly, the chapters are ordered in such a way that the presentation is cumulative. Thus, the book attempts to discuss things "from scratch", i.e., without referring to specialized background or more advanced references; physicists do not expect theorems and lemmas, but do expect to be convinced.[1] Throughout the text, the phrases "it can be shown"[2] and "stated without proof" are actively avoided, so this book may also be used in a flipped classroom, perhaps even for self-study. As part of this approach, I frequently cover things like convergence properties, operation counts, and the error scaling of different numerical methods. When space constraints imposed a choice between not explaining a method well or omitting it entirely, I opted in favor of omission. This is intended as a "first book" on the subject, which should enable students to confidently move on to more advanced expositions.
- **Methods implemented:** while the equations and figures help explain why a method should work, the insight that can be gleaned by examining (and modifying) an existing implementation of a given algorithm is crucial. I have worked hard to ensure that these code listings are embedded in the main discussion, not tossed aside at the

[1] *Nullius in verba*, the motto of the Royal Society, comes to mind. The idea, though not the wording, can clearly be traced to Heraclitus's Fragment 50: "Listen, not to me, but to reason".
[2] An instance of *proof by omission*, but still better than "it can be easily shown" (*proof by intimidation*).

end of the chapter or in an online supplement. Even so, each implementation is typically given its own subsection, in order to help instructors who are pressed for time in their selection of material. Since I wanted to keep the example programs easy to talk about, they are quite short, never longer than a page. In an attempt to avoid the use of black boxes, I list and discuss implementations of methods that are sometimes considered advanced (e.g., the QR eigenvalue method or the fast Fourier transform). While high-quality libraries like NumPy and SciPy contain implementations of such methods, the point of a book like this one is precisely to teach students how and why a given method works. The programs provided can function as templates for further code development on the student's part, e.g., when solving the end-of-chapter problems. Speaking of the problems, these usually probe conceptual issues, edge cases, alternative implementations, or other numerical methods; instructors can get access to the solutions of more than 170 problems.

- **Clear separation between numerical method and physics problem:** each chapter focuses on a given numerical theme. The first section always discusses physics scenarios that touch upon the relevant tools; these "motivational" topics are part of the standard undergrad physics curriculum, ranging from classical mechanics, through electromagnetism and statistical mechanics, to quantum mechanics. The bulk of the chapter then focuses on several numerical methods and their implementation, typically without bringing up physics examples. The penultimate section in each chapter is a Project: in addition to involving topics that were introduced in earlier sections (or chapters), these physics projects allow students to carry out (classical or quantum) calculations that they wouldn't attempt without the help of a computer. These projects also provide a first taste of "programming-in-the-large". As a result of this design choice, the book may also be useful to beginning physics students or even students in other areas of science and engineering (with a more limited physics background). Even the primary audience may benefit from the structure of the text in the future, when tackling different physics problems.

Conscious efforts have been made to ensure this volume is reasonably priced: all figures are in grayscale, the page-count has been reduced by moving most of the introductory Python material to `www.numphyspy.org`, and a (relatively) inexpensive paperback version is also being published. In keeping with Wordsworth's verses, this book is dear to my heart; I hope the reader gets to share some of my excitement for the subject.

On the Epigraphs

I have translated 12 of the quotes appearing as epigraphs myself. In two more instances the original was in English and in the remaining case the translator is explicitly listed. All 15 quotes are not protected by copyright. The sources are as follows:

- Dedication: Pindar, *Pythian* 8, Lines 95–97 (446 BCE)
- Preface: William Wordsworth, *The Prelude*, Book Fourteenth (1850 CE)

- Chapter 1: Johann Wolfgang von Goethe, *Faust, Part I*, Lines 681–682 (1808 CE)
- Chapter 2: Aristotle, *Nicomachean Ethics*, Book I, Chapter III, 1094b (~330 BCE)
- Chapter 3: William Shakespeare, *King Lear*, Act I, Scene IV (~1605 CE)
- Chapter 4: Hesiod, *Works and Days*, Line 643 (~700 BCE)
- Chapter 5: Heraclitus, *Fragment 84* (~500 BCE)
- Chapter 6: Vergil, *Aeneid*, Book 7, Lines 310–311 (~19 BCE, translated 1697 CE)
- Chapter 7: Sophocles, *Ajax*, Lines 485–486 (~441 BCE)
- Chapter 8: Sophocles, *Oedipus Tyrannus*, Lines 120–121 (~429 BCE)
- P.S.: Diogenes Laërtius, *Lives and Opinions of Eminent Philosophers*, Book 6 (~220 CE)
- Appendix A: Plato, *The Republic*, Book 5, 453d (~380 BCE)
- Appendix B: Immanuel Kant, *Groundwork for the Metaphysics of Morals*, 2nd Section (1785 CE)
- Appendix C: Friedrich Nietzsche, *The Gay Science*, 381 (1877 CE)
- Bibliography: Plato, *Phaedrus*, 235d (~370 BCE)

Acknowledgments

My understanding of numerical methods has benefited tremendously from reading many books and papers, especially those by Forman Acton, David Ceperley, William Kahan, Nick Trefethen, and James Wilkinson; students looking for more reading material should start here. In the bibliography I mention only the works I consulted while writing.

I have learned a lot from my graduate students, my collaborators, as well as members of the wider nuclear physics, cold-atom, and astrophysics communities. This textbook is a pedagogical endeavor but it has unavoidably been influenced by my research, which is supported by (a) the Natural Sciences and Engineering Research Council of Canada, (b) the Canada Foundation for Innovation, and (c) an Early Researcher Award from the Ontario Ministry of Research, Innovation and Science.

I would like to thank my Editor at Cambridge University Press, Vince Higgs, for his considered advice, Margaret Patterson for her meticulous copyediting, as well as Henry Cockburn and Esther Miguéliz Obanos for answering my many questions. I am also grateful to the five anonymous reviewers for their positive feedback and suggestions for additions.

I wish to acknowledge the students taking my third-year computational-methods class; their occasional blank stares resulted in my adding more explanatory material, whereas their (infrequent, even at 8:30 am) yawns made me shorten some sections. I also want to thank Heather Martin for her advice on copyright, Eric Poisson for his wide-ranging comments on chapter drafts, and Martin Williams for personifying the advantages of teaching *hilaritatis causa*. I am indebted to my family, *inter multa alia*, for orienting me toward the subjective universal; Liliana Caballero provided wise counsel on the physics projects. While several people have helped me refine this book, any remaining poor choices are my responsibility.

Codes