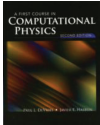




A GREAT TEXT FOR MATLAB-SPECIFIC COMPUTATIONAL PHYSICS COURSES (AND A GOOD TEXT OTHERWISE)

By Eric Ayars

Paul DeVries and Javier Hasbun, *A First Course in Computational Physics*, 2nd ed., Jones & Bartlett Publishers, 2010, ISBN: 978-0763773144, 448 pp.



This is a clearly written, readable, and well-balanced introduction to computational physics. It uses Matlab for the language, and although the text is tightly integrated with that language, it's not inseparable. The book's level is appropriate for upper-division undergraduate or introductory graduate-level courses. It thoroughly covers the basics of computational physics, with little if any extraneous topics.

This book is not ideal for a course in which the students have no previous programming experience. The authors expect that students will be familiar with at least one programming language (preferably Matlab). They provide an appendix that gives a useful Matlab introduction for those completely unfamiliar with the package; other than that, there's no attempt to *teach* Matlab or any other language. In short, it's a book on computational physics, not on computation.

Strengths

By far, the book's best feature is the delightfully clear and understandable explanations of the more complicated aspects of computational physics. The authors' explanation of the fast Fourier transform (FFT), for example, quickly and easily explains its function, why it requires that the number of samples be a power of two, and why

it's so much faster than the discrete Fourier transform. Although I've been teaching computational physics for years, their explanation is so clear that it made me realize that I hadn't yet fully grasped the technique's elegance.

This level of elegance in the explanations extends throughout the book. In their introduction to integration formulae, the authors skip the usual graphical pictures of the sum method and the trapezoid rule. Instead, they go straight to a Taylor expansion of the integrand, and in two easy-to-follow pages they pop out the trapezoid rule, Simpson's method, and two higher-order methods as well.

There's a tendency among students in a computational physics course to jump right in to the code, throwing processing power at a problem without much consideration. This book fights that first instinct throughout. It repeatedly reminds students to think about the *problem* rather than the *program*. It provides a healthy emphasis on good coding practice, and it demonstrates good practice in the sample code given.

Some might consider the book's small topic selection a weakness, but from my own experience in teaching computational physics, I consider it a strength. The volume of material is perfect for a one-semester course. The book does an excellent job of

thoroughly covering the basics needed for general computational work, without trying to cover less fundamental topics that students are unlikely to need outside of specialized computational physics research. It's what students need, without the standard "second half of the book," which is full of topics that no class ever quite reaches.

Weaknesses

It might seem odd to list the elegance of the authors' explanations as both a strength and a weakness, but that elegance sometimes cuts both ways. Despite the authors' warnings about using "clever" code, they sometimes use clever explanations where a more basic explanation would be more valuable to most students. To use an example given earlier: students who are already familiar with basic integration techniques will undoubtedly benefit from the ease with which the authors derive—and then move beyond—Simpson's method. But a student who lacks previous experience in computational physics will probably have a difficult time following the authors directly into Romberg integration.

Another weakness of the text is a somewhat haphazard arrangement of topics. For example, the chapter on Fourier analysis contains large sections on chaos in nonlinear differential equations. These are certainly related topics in the sense that Fourier

analysis is a powerful tool for analyzing chaotic systems, and in fairness to the authors there's nothing disjointed about the flow from one topic to the next. They talk about FFTs, then power spectra, then the van der Pol oscillator, then Lyapunov exponents, and it all makes sense; but it's a bit disconcerting to have to occasionally glance at the page header to remind yourself of the chapter's overall topic—which is not chaos theory!

In my opinion, the text's biggest weakness is that it's written around Matlab, which is an expensive and proprietary software. However, the authors rarely do anything that *requires* Matlab, so it's possible to work around this issue. Python—with the SciPy package—duplicates nearly all of the Matlab toolbox, and if you

choose to go this route, the book's Matlab examples can be translated into Python with relative ease.

If you're teaching an upper-level undergraduate or introductory graduate-level computational physics course, you're using Matlab, and your students have some prior programming experience, then use this book. Minor quibbles aside, it's the best text I've seen for that particular combination of course characteristics.

For the rest of us, this book still might be a good choice; but it's not as obvious a choice. For starters, if your course lacks a programming prerequisite, you should expect to spend considerable class time on programming basics. If your course is taught

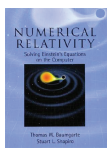
using a language other than Matlab, then after translating some example code, this text could work very well. The book's best qualities don't depend on the Matlab examples, and libraries exist for other languages that allow the same functionality as Matlab. It's an excellent book, and well deserving of serious consideration for any computational physics course.

Eric Ayars is an associate professor of physics at California State University, Chico. In addition to computation, his research interests include development and applications of microcontroller systems for physics and cross-disciplinary undergraduate research. Ayars has a PhD in physics from North Carolina State University. Contact him at ayars@mailaps.org.

ENGINEERING COSMIC COLLISIONS

By **Richard Matzner**

Thomas W. Baumgarte and Stuart L. Shapiro, *Numerical Relativity*, Cambridge Univ. Press, 2010, ISBN: 978-0521514071, 720 pp.



Numerical Relativity by Thomas Baumgarte and Stuart Shapiro is an extraordinary book. It's *big*, containing 597 pages, with another 100 pages of appendices, references, and index; it's also *dense*, capturing much of the recent advances in computational modeling via Einstein's description of gravity, or general relativity. With the Laser Interferometer Gravitational Wave Observatory (LIGO) now actively searching for gravity waves, this topic is particularly timely.

A book entitled *Numerical Relativity* could include special relativity—a

topic subsumed by general relativity—but there's little particular attention paid to the topic here. This is a reasonable choice, because no book can cover all aspects of any modern field, and this book is satisfyingly thorough in what it does cover. For example, if you want to understand how to model the merger of compact astrophysical objects (such as spinning black holes) and what signal you'd observe in a gravitational wave detector, this book is essential. There's none other like it. Moreover, the book's contents will allow smart graduate students who have experienced guides to carry out a wide swath of research in a very important subject.

The authors designed this to be useful as a textbook, so it contains “300 exercises scattered throughout the book.” However, it's definitely aimed at the expert. It would be a real feat for a naive reader to absorb the mass of detail presented here, all of which is directly relevant to the subject.

Early Chapters

The book's contents alternate between analytical relativity and analytic questions directly relevant to numerical computation. There's little in the way of computational analysis, but the book contains detailed descriptions of the approaches and procedures necessary to model strong-gravity systems, and it tells you how to carry them out.

The background material includes:

- coordinate choices and appropriate metrics for a general relativistic computation;