



OBJECT-ORIENTED SCIENTIFIC PROGRAMMING—WITHOUT REINVENTING THE WHEEL

By Ramses van Zon

Damian Rouson, Jim Xia, and Xiaofeng Xu, *Scientific Software Design, The Object-Oriented Way*, Cambridge Univ. Press, 2011, ISBN: 978-0-521-88813-4, 382 pp.



This book on object-oriented design of scientific software presents design patterns that the authors have found to be applicable and useful in scientific computation. Largely driven by examples, the book includes many close-to-real scientific applications and uses the Fortran and C++ programming languages, with some emphasis on the former. The level is appropriate for moderately experienced programmers. As a textbook for a graduate course in scientific object-oriented *programming*, additional material would be needed because this book focuses on program *design*. More experienced scientific object-oriented programmers (who perhaps were never quite convinced about the utility of design patterns) might learn a trick or two as well—or might realize that they’ve occasionally reinvented the wheel.

Organization

The book is divided into three parts. The three chapters in Part I (“The Tao of Scientific OOP”) deal with general aspects of development costs, how object-oriented programming can help reduce these, and how this applies specifically in a scientific context.

Part II, “SOOP to Nuts and Bolts,” (where “SOOP” stands for scientific object-oriented programming), has six chapters that describe useful design patterns for scientific applications, such

as the abstract calculus pattern and the puppeteer pattern. The final part, “Gumbo SOOP,” contains chapters that discuss formal constraints and mixed-language programming, offer a brief introduction to parallel programming, and showcase multiphysics architectures.

The book also includes two appendices at the end—the first on the mathematical background used in the examples (Lagrange interpolation, linear and nonlinear solvers, partial differential equations, and finite difference approximations) and the second on the Universal Modeling Language (UML). It might have been more useful to have the latter as part of the main text, because UML diagrams aren’t intuitively understandable to the uninitiated, and they’re used throughout the book.

Patterns Are Useful, but What about Performance?

One of the book’s strengths is that it demonstrates the utility of design patterns in scientific computing by showing how they solve real software design problems. The book also shows that the choice of programming language is, to a large degree, irrelevant to the software design. The choice of C++ and Fortran 2003 is beneficial in that they both support abstract data-type calculus through operator overloading, a technique of which the authors are strong proponents. When well designed, such a calculus helps

with code maintenance by making the programmer’s intention much clearer and reducing the code’s size.

Another welcome aspect of the book is its discussion of the debugging qualities of codes. All too often, textbooks pretend that this aspect of programming doesn’t exist. Here, the debugging models are a bit crude and the discussion ignores some of the tools that facilitate and speed-up debugging (such as gdb and valgrind), but this doesn’t invalidate the main point: that a modular, object-oriented approach greatly reduces the amount of code to inspect, and thus reduces debugging time.

The detailed code examples are the book’s best feature. In fact, the book’s codes are not so much examples as an integral part of the text (despite being referred to as “figures”). The examples here stay closer to real scientific applications than those in most object-oriented textbooks. This helps the authors avoid both the easier problems (for which almost any design would work well) and artificially constrained problems (for which only one design works, but which you’re unlikely to ever encounter).

The book does have a few weaknesses as well. It has a slight philosophical slant, which can distract from the book’s valuable practical aspects. For example, the explanation of design patterns’ role in architecture is much longer than is needed for readers to appreciate the discussion of doing object-oriented programming

in Fortran 2003 (and the associated challenges) or how you might set up a multiphysics application. There's also a tendency to invoke the authority of the authors of the 1994 book, *Design Patterns: Elements of Reusable Object-Oriented Software*, rather than present an argument for why that particular pattern arises.

each component of each point of the ring, memory for the vector might have to be allocated dynamically. It's also confusing, because a dynamically sized vector is conceptually different from a fixed-sized array. In a scientific context, problem sizes tend to grow, so addressing this performance issue only at scale

array—which would have been useful in the current context to avoid the inefficiencies in using `std::vector`.

Finally, scientific or numerical libraries fall outside the book's scope, although using such libraries benefits the scalability of the code design. After all, each module that you don't maintain is one less module to debug. Libraries are also more likely to have been optimized for runtime performance.

This book makes a good case for the usefulness of design patterns and object-oriented programming for maintainable code, but disregards runtime performance and scientific libraries.

A more substantial weakness is that this text doesn't consider runtime performance, focusing instead on a scalable design. In fact, the authors suggest that scalable design will lead to scalable performance. This works at the coarse-grained level of the program's design, but is potentially counter-productive when applied to the fine-grained level. For example, the C++ implementation of the vortex class (which models a ring of 3D points) uses the `push_back` method of the standard template library class `std::vector` to set variables in an uninitialized vector, even though that vector will always hold three floating-point numbers. This is inefficient because, for

would require a rewriting of the code's fine-grained details, which is a risky undertaking that easily overshadows the initial advantage in maintainability. For low-level parts of your code, you should know and care about the inefficiencies of different object-oriented methods. In the current case, a simple fixed-size array would have sufficed and would likely be two to three times faster.

A minor, temporary weakness of the book is its use of Fortran 2003, which isn't fully supported by all mainstream compilers. In contrast, the authors avoid features from the C++11 standard and the Boost C++ library—such as smart pointers and a proper fixed-size

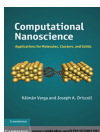
This book makes a good case for the usefulness of design patterns and object-oriented programming for maintainable code, but disregards runtime performance and scientific libraries. Still, it's one of those books that I wish I'd read earlier in my programming career. I found many design patterns familiar simply because I'd seen them before in my own code. I'll likely turn to this book in the future whenever I suspect a program design problem might be solved already.

Ramses van Zon is an application analyst and HPC computational science specialist at the University of Toronto's SciNet High Performance Computing Consortium. His research interests include non-equilibrium statistical physics and molecular dynamics simulations. van Zon has a PhD in theoretical physics from Utrecht University in The Netherlands. Contact him at rzon@scinet.utoronto.ca.

ALGORITHM PRIMER FOR NANOSCIENCE

By *Bruce Harmon*

Kálmán Varga and Joseph A. Driscoll, *Computational Nanoscience, Applications for Molecules, Clusters, and Solids*, Cambridge Univ. Press, 2011, ISBN: 978-1107001701, 444 pp.



Chemistry, solid-state physics, biology, and materials science are converging at the

nanoscale. High-powered instruments to visualize and probe at atomic-length scales have become ubiquitous in these disciplines, as have the

theoretical tools necessary to understand and explain the resulting physical structures and phenomena. Indeed, accurate computations and theoretical analysis are often necessary partners with experiment to assess the importance of the quantum manifestations at Ångström- and nanometer-length scales.

This book by Kálmán Varga and Joseph A. Driscoll is a welcome

educational venture into this important and fast-growing discipline. Because such a moving target is too complex and rapidly expanding to completely cover, the authors have narrowed their focus (somewhat) by restricting their aim “to provide a comprehensive program library and a description of advanced algorithms to help students and researchers learn novel methods and develop their own approaches.”

Purview

Part I of the book deals with 1D problems. The pedagogy is appropriate and the algorithm descriptions are well presented. The material can be covered quickly and might well suit a one-semester advanced undergraduate class, where the instructor could fill in some of the motivation and provide context before moving on to selected examples from the more advanced 3D topics in Part II of the book.

In Part II, the choice is rich, with topics including Monte Carlo, molecular dynamics, (time-dependent) density functional strategies, electron transport across single molecules, and the manipulation of atomic-orbital and plane-wave basis states for electronic structure calculations. The wide range and large number of advanced topics in Part II lets instructors select topics to supplement the teaching of a first-year graduate course in, say, solid-state physics, or even a multi-disciplinary course in computational nanoscience.

The Niche, the Missing, the Promise

As a student many years ago, I would have welcomed such a book when taking the canonical “Numerical Methods Course for Scientists” or the mathematical physics class. Nowadays, there are well-documented

subroutine libraries for most of the numerical methods, but the driver or main program for specific applications are still needed. There are also a growing number of complete codes available as open source maintained by dedicated groups of users. A recent publication, for example, listed 24 such codes.¹

These codes are frequently complex and might be impenetrable to the novice, so it’s noteworthy that the codes presented online for this book have been purposely simplified to increase


elsewhere is uninformative—though, admittedly, it gives the instructor an opportunity to drive home the concept. In 3D, the tetrahedron method to evaluate the electronic DOS is elegant and provides a lesson in numerical simplicity. Too bad it wasn’t included.

Eagerly give this book a passing grade for entering a brave new world, but I also look forward to a revised and updated second edition that has been polished by classroom experience and user feedback. Indeed, this

This book and associated software could join the vanguard of new instructional methodology as the Internet allows faster feedback, and the instructional content and value can be improved expeditiously.

their pedagogical value. If you’re teaching or taking a graduate-level class or seminar series with relatively disparate topics, you’ll greatly appreciate access to a collection of relevant codes and examples such as those given here. A well-written algorithm can be like a classic novel in revealing how components interact, the consequences of various actions, and the denouement.

Although I’m enthusiastic about the concepts that motivated this book and the selected topics, I’m a bit disappointed that one or two of my favorite topics didn’t receive adequate exposition. For example, the concept of the density of states (DOS) is quite basic, but the 1D example given is nearly useless, as the points in the energy bands with zero slope give an infinite density of states. The resulting plot of DOS(E) with vertical lines at those values of E and essentially zero

book and associated software could (and should) join the vanguard of new instructional methodology as the Internet allows faster feedback, and the instructional content and value can be improved expeditiously. 

Reference

1. Y. Saad, J.R. Chelikowsky, and S.M. Shontz, “Numerical Methods for Electronic Structure Calculations of Materials,” *SIAM Rev.*, vol. 52, no. 1, 2010, pp. 43–54.

Bruce Harmon is a distinguished professor in the Department of Physics and Astronomy at Iowa State University and a senior scientist at the US Department of Energy’s Ames Laboratory. His research interests include nanoscale magnetism, x-ray magnetic dichroism, phonons, and computational materials discovery. Harmon has a PhD in physics from Northwestern University. Contact him at harmon@ameslab.gov.

PURPOSE: The IEEE Computer Society is the world's largest association of computing professionals and is the leading provider of technical information in the field.

MEMBERSHIP: Members receive the monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others interested in the computer field.

COMPUTER SOCIETY WEBSITE: www.computer.org

Next Board Meeting: 1–3 Feb., Long Beach, CA, USA

EXECUTIVE COMMITTEE

President: John W. Walz*

President-Elect: David Alan Grier;* **Past President:** Sorel Reisman;* **VP, Standards Activities:** Charlene (Chuck) Walrad;† **Secretary:** Andre Ivanov (2nd VP);* **VP, Educational Activities:** Elizabeth L. Burd;* **VP, Member & Geographic Activities:** Sattupathuv Sankaran;† **VP, Publications:** Tom M.

Conte (1st VP);* **VP, Professional Activities:** Paul K. Joannou;* **VP, Technical & Conference Activities:** Paul R. Croll;† **Treasurer:** James W. Moore, CSDP;* **2011–2012 IEEE Division VIII Director:** Susan K. (Kathy) Land, CSDP;† **2012–2013 IEEE Division V Director:** James W. Moore, CSDP;† **2012 IEEE Division Director VIII Director-Elect:** Roger U. Fujii†

*voting member of the Board of Governors

†nonvoting member of the Board of Governors

BOARD OF GOVERNORS

Term Expiring 2012: Elizabeth L. Burd, Thomas M. Conte, Frank E.

Ferrante, Jean-Luc Gaudiot, Paul K. Joannou, Luis Kun, James W. Moore, William (Bill) Pitts

Term Expiring 2013: Pierre Bourque, Dennis J. Frailey, Atsuhiko Goto, André Ivanov, Dejan S. Milojicic, Paolo Montuschi, Jane Chu Prey, Charlene (Chuck) Walrad

EXECUTIVE STAFF

Executive Director: Angela R. Burgess; **Associate Executive Director, Director, Governance:** Anne Marie Kelly; **Director, Finance & Accounting:** John Miller; **Director, Information Technology & Services:** Ray Kahn; **Director, Membership Development:** Violet S. Doan; **Director, Products & Services:** Evan Butterfield; **Director, Sales & Marketing:** Chris Jensen

COMPUTER SOCIETY OFFICES

Washington, D.C.: 2001 L St., Ste. 700, Washington, D.C. 20036-4928

Phone: +1 202 371 0101 • **Fax:** +1 202 728 9614

Email: hq.ofc@computer.org

Los Alamitos: 10662 Los Vaqueros Circle, Los Alamitos, CA 90720-1314

Phone: +1 714 821 8380 • **Email:** help@computer.org

MEMBERSHIP & PUBLICATION ORDERS

Phone: +1 800 272 6657 • **Fax:** +1 714 821 4641 • **Email:** help@computer.org

Asia/Pacific: Watanabe Building, 1-4-2 Minami-Aoyama, Minato-ku, Tokyo 107-0062, Japan

Phone: +81 3 3408 3118 • **Fax:** +81 3 3408 3553

Email: tokyo.ofc@computer.org

IEEE OFFICERS

President: Moshe Kam; **President-Elect:** Gordon W. Day; **Past President:**

Pedro A. Ray; **Secretary:** Roger D. Pollard; **Treasurer:** Harold L. Flescher;

President, Standards Association Board of Governors: Steven M. Mills; **VP,**

Educational Activities: Tariq S. Durrani; **VP, Membership & Geographic**

Activities: Howard E. Michel; **VP, Publication Services & Products:** David

A. Hodges; **VP, Technical Activities:** Donna L. Hudson; **IEEE Division V**

Director: James W. Moore, CSDP; **IEEE Division VIII Director:** Susan K.

(Kathy) Land, CSDP; **President, IEEE-USA:** Ronald G. Jensen

revised 1 Dec. 2011



AIP

www.aip.org

The American Institute of Physics (AIP) is a not-for-profit membership corporation chartered in New York State in 1931 for the purpose of promoting the advancement and diffusion of the knowledge of physics and its application to human welfare. Leading societies in the fields of physics, astronomy, and related sciences are its members.

In order to achieve its purpose, AIP serves physics and related fields of science and technology by serving its member societies, individual scientists, educators, students, R&D leaders, and the general public with programs, services, and publications—*information that matters*.

The Institute publishes its own scientific journals as well as those of its member societies; provides abstracting and indexing services; provides online database services; disseminates reliable information on physics to the public; collects and analyzes statistics on the profession and on physics education; encourages and assists in the documentation and study of the history and philosophy of physics; cooperates with other organizations on educational projects at all levels; and collects and analyzes information on federal programs and budgets.

The scientists represented by the Institute through its member societies number more than 134 000. In addition, approximately 6000 students in more than 700 colleges and universities are members of the Institute's Society of Physics Students, which includes the honor society Sigma Pi Sigma. Industry is represented through the membership of 37 Corporate Associates.

Governing Board: Louis J. Lanzerotti* (chair), David E. Aspnes, Anthony Atchley,* Barry Barish, G. Fritz Benedict, J. Daniel Bourland, Curtis G. Callan, Charles W. Carter Jr, Timothy A. Cohn, David M. Cook, William J. Cook, Bruce H. Curran,* Morton M. Denn,* Alexander Dickson, Michael D. Duncan, H. Frederick Dylla* (ex officio), Janet Fender,* Judith Flippen-Anderson,* Brian J. Fraser,* Jaime Fucugauchi, Timothy Grove, Mark Hamilton, Warren W. Hein,* James Hollenhorst, Paul L. Kelley, Angela R. Keyser, Kate Kirby, Rudolf Ludeke,* Kevin B. Marvel,* Christine McEntee, Michael McPhaden, Patricia Mooney, Cherry Murray, Elizabeth A. Rogan, Charles E. Schmid, Joseph Serene,* Benjamin B. Snavely* (ex officio), David Sokoloff, Gene Sprouse, Hervey (Peter) Stockman, Quinton L. Williams. *Executive Committee member.

Management Committee: H. Frederick Dylla, Executive Director and CEO; Richard Baccante, Treasurer and CFO; Theresa C. Braun, Vice President, Human Resources; John S. Haynes, Vice President, Publishing; Catherine O'Riordan, Vice President, Physics Resources; Benjamin B. Snavely, Secretary.