

# RESOURCE LETTER

Resource Letters are guides for college and university physicists, astronomers, and other scientists to literature, websites, and other teaching aids. Each Resource Letter focuses on a particular topic and is intended to help teachers improve course content in a specific field of physics or to introduce nonspecialists to this field. The Resource Letters Editorial Board meets at the AAPT Winter Meeting to choose topics for which Resource Letters will be commissioned during the ensuing year. Items in the Resource Letter below are labeled with the letter E to indicate elementary level or material of general interest to persons seeking to become informed in the field, the letter I to indicate intermediate level or somewhat specialized material, or the letter A to indicate advanced or specialized material. No Resource Letter is meant to be exhaustive and complete; in time there may be more than one Resource Letter on a given subject. A complete list by field of all Resource Letters published to date is at the website [www.kzoo.edu/ajp/letters.html](http://www.kzoo.edu/ajp/letters.html). Suggestions for future Resource Letters, including those of high pedagogical value, are welcome and should be sent to Professor Roger H. Stuewer, Editor, AAPT Resource Letters, School of Physics and Astronomy, University of Minnesota, 116 Church Street SE, Minneapolis, MN 55455; e-mail: [rstuewer@physics.umn.edu](mailto:rstuewer@physics.umn.edu)

## Resource Letter CP-2: Computational Physics

Rubin H. Landau<sup>a)</sup>

*Department of Physics, Oregon State University, Corvallis, Oregon 97331*

(Received 19 November 2007; accepted 6 January 2008)

This Resource Letter provides a guide to print and electronic literature relevant to a computational physics course. The multidisciplinary aspect of computational physics and its relation to computational science is reflected in the sections Courses, Programs and Reviews, Journals, Conferences and Organizations, Books, Tools, Languages and Environments, Parallel Computing, and Digital Libraries. © 2008 American Association of Physics Teachers.

[DOI: 10.1119/1.2837814]

### I. INTRODUCTION

#### A. CP's multidisciplinary nature

In 1996 Paul DeVries published Resource Letter CP-1 on Computational Physics as an aide to those teaching a course in Computational Physics (CP) (Ref. 9). The number of resources now available has increased significantly, some of the original CP texts are now out of print, Java and Python are now popular programming languages, and the exponential growth in the power and pervasiveness of computers has made computation essential to all areas of science. (A review of the history of CP education appears in Ref. 2.) In fact, although it appears that most physics classes now incorporate computation in some way, I would not call them “CP” classes if their use of computation is just to improve physics education. However, I would if they incorporate and analyze modern computational techniques as used in research to solve problems.

In this Resource Letter I adopt the view [see Fig. 1(a)] that CP is a multidisciplinary field of study that encourages students to understand physics along with computer science and applied mathematics. Teaching such a course creates new scholarly materials that fuse chunks of knowledge together [the central circle in Fig. 1(a)]. A CP course thus acts as a bridge that connects the three disciplines and leads students on a path toward applying the disciplines simultaneously in the process of solving concrete problems.

The inclusion of the computational tools and methods of modern research into education is not restricted to physics. It is happening in all of the sciences and is often called “Computational Science and Engineering” (CSE). As such, we view CP as a subfield of Computational Science. (Section IX

provides a glossary of acronyms that the reader may find useful in reading this Letter and in searching the literature.) The similarities of techniques and methods throughout the computational sciences form a common educational base that permits its practitioners to solve realistic and practical problems in a broad range of subjects, and to extend education to include problems that do not have analytical solutions. The effectiveness and power of computation in so many fields has led some people to add simulation to experiment and theory as key avenues followed in searching for scientific truth [Fig. 1(b)].

Recent advances in computational science affect this Resource Letter in a number of ways. On the one hand, I have listed some CSE resources that I believe are of value in CP, even though they may not be derived from physics. (For example, Ref. 16 indicates strong similarities between CP and CSE undergraduate degree programs.) You will note in these resources the inclusion of many tools in addition to traditional academic materials; this reflects the practical problem-solving paradigm at the heart of computational sciences and the fact that these tools help define what computational physicists do and how they do it.

On the other hand, computation is now so integral to physics research that there are numerous papers and books combining computation with advanced or specialized topics. Consequently, I have had to leave out many excellent resources that are important for research, but which I do not view as general enough for a CP course. Accordingly, I would encourage another Resource Letter that collects resources that incorporate computation into the traditional physics disciplines. This would include pioneering books like Visscher's *Fields and Electrodynamics: A Computer-Compatible Introduction* for E&M, and introductory texts such as Chabay and Sherwood's *Matter & Interactions*, and

<sup>a)</sup>Electronic mail: [rubin@science.oregonstate.edu](mailto:rubin@science.oregonstate.edu)

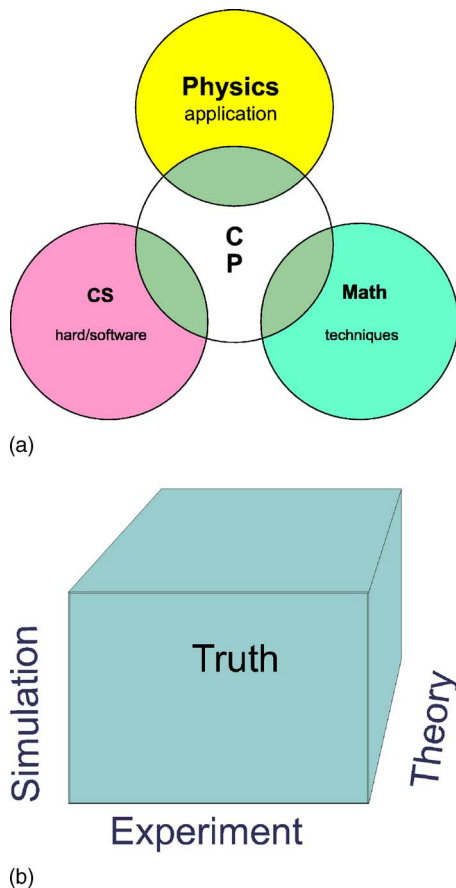


Fig. 1. (a) Computational Physics as both the union of physics with computer science and applied mathematics, and as a bridge connecting the three traditional disciplines [based on Ref. 16]. (b) The addition of simulation as a path toward scientific truth.

Christian and Belloni's *Physlet Physics*.

## B. The contents of CP

There is no standard curriculum for a course or degree in CP. Indeed, with only five CP degree programs in the entire country, and maybe another five CP minors/tracks, the field is still very much under development (see Section II for resources on various CSE programs and courses). A major challenge in a CP course, which is reflected in all of the texts in Section V A, is that much of the coverage needs to be of numerical methods, computational tools, and computer languages; it is hard to get into any significant physics until that is done.

When reflecting upon what subjects might be appropriate for future CP education, it is helpful to look to the future of computation, where the funding agencies envision petascale computing by 2011. (A petaflop is  $10^{15}$  floating point operations per second, which means that if Moore's law were applied to our desktop computers, we would have to wait until 2037 for a petaflop.) Although we can always repeat our old simulations with higher resolution and higher accuracy, petascale computing is most appropriate for new problems that involve multitime and multispace scales, and, accordingly, multiphysics. As laid out by the National Science Foundation (NSF), the future of CP includes quantum chromodynamics of particle interactions; radiative, dynamic and nuclear physics of stars and the collision of stars; heteroge-

neous catalysis on both semiconductor and metal surfaces; intermittency, stratification, and rotating turbulence in magnetic fluids; properties of matter under extreme conditions as calculated from first-principles simulations; molecular understanding of friction and lubrication; designing molecular electronic devices; generation and evolution of magnetic fields in planets and stars; modeling interaction of the Earth's magnetic field with coronal mass ejections; formation and evolution of galaxies; burning plasmas and magnetic confinement techniques; formation of planetary nebula by low Mach-number flows.

All of these subjects must be approached with a CSE toolkit including multigrid differential equation solvers, multi-scale (in time and space) physics and mathematics, optimized performance of large-scale parallel computations based on an understanding of HPC architectures with various forms of parallelisms, conversion of algorithms for use of partitioned global address space languages, multidimensional visualization tools, large-scale inverse-problem solutions, sophisticated sampling techniques, analysis of (apparently) uncorrelated multidimensional data sets distributed over vast networks, extensive use of scalable high performance computing (HPC) libraries, and collaboration with teams of CS and math specialists.

## II. COURSES, PROGRAMS, AND REVIEWS

There are less than 20 US undergraduate degree programs in all of the computational sciences, and five degree programs in CP. Websites and papers describing these programs, and those for minors, options, concentrations, tracks, foci, etc., as well as graduate programs are valuable CP resources.

### A. Programs

1. **B.S., Computational Physics, SUNY Buffalo**, a multidepartment effort, [undergrad-catalog.buffalo.edu/academicprograms/comphys.shtml](http://undergrad-catalog.buffalo.edu/academicprograms/comphys.shtml).
2. **B.A., Physics, Lawrence University**, *Computation in the Lawrence Physics Curriculum*, D. M. Cook, Proc. Int. Conf. on Computational Sci., ICCS 2001, ed. Vassil N. Alexandrov *et al.* (Springer, Berlin, 2001) 1074–1083; *Comput. Phys.* **11**, 240–245 (1997); **11**, 331–335 (1997); [www.lawrence.edu/dept/physics/ccli](http://www.lawrence.edu/dept/physics/ccli); *Computation in Undergraduate Physics: The Lawrence Approach*, Am. J. Phys. (2008).
3. **B.S., Computational Physics, Oregon State**, *Computational Physics for Undergraduates, the CPUG Degree Program at Oregon State Univ.*, R. H. Landau, *Comput. Sci. Eng.* **6**, 68–75 (2004); Program description at [www.physics.oregonstate.edu/CPUG](http://www.physics.oregonstate.edu/CPUG).
4. **B.S., Computer Physics, Illinois State**, *The Computer Physics Bachelor's Degree Program at Illinois State Univ.*, R. F. Martin, Jr., *Comput. Sci. Eng.* **8** (2006); possibly the first undergraduate CP program in the country, with CP classes starting in the 1970s, [www.phy.ilstu.edu/programs/CompPhys](http://www.phy.ilstu.edu/programs/CompPhys).
5. **Moderatorship, Physics and Computer Simulation, Trinity**, an undergraduate degree program at Trinity College, Dublin, [www.tcd.ie/Physics/undergraduate/computer\\_simulation](http://www.tcd.ie/Physics/undergraduate/computer_simulation).
6. **B.S., Computational Sci., SUNY Brockport**, *New Perspective on Computational Science Education*, O. Yaşar, K. S. Rajasethupathy, R. E. Tuzun, R. A. McCoy, and J. Harkin, *Comput. Sci. Eng.* **5**, 74–79 (2000).
7. **Multiple, Ralph Regula School of Computational Sci.**, Ohio's state-wide virtual school, contains some physics courses as well. Although in initial stage, may be a model for other states, [www.rscs.org](http://www.rscs.org).
8. **CSE Minors, Concentrations, or Tracks**, these change often, but include: **CP**, Abilene Christian, North Carolina State, Penn State-Erie, and Arkansas; **Computational Math**, Princeton, San Diego State, Central Florida, and Nebraska-Lincoln; **CSE**, Capital, Clark, Old Dominion, RPI, Salve-Regina, Syracuse, Wisconsin-Eau Claire,

## B. Courses, general education, and reviews

9. **Resource Letter CP-1: Computational Physics**, P. L. DeVries, *Am. J. Phys.* **64**, 364–367 (1996). Previous Resource Letter. (E)
10. **Computational Physics and the Undergraduate Curriculum**, H. Gould, *Comput. Phys. Commun.* **127** (1), 6–10 (2000).
11. **Computational Physics: A Better Model for Physics Education?**, R. H. Landau, *Comput. Sci. Eng.* **8**, 22–30 (2006). Survey results and relevant statistics.
12. **Computational Physics for Upper Level Courses, AAPT Topical Conference**, Davidson College, July 2007. There should be a number a papers appearing in the *Am. J. Phys.* (possibly September 2008) reporting on this conference. See too [www.opensourcephysics.org/CPC](http://www.opensourcephysics.org/CPC).
- Computational Physics Education: Coming of Age?**, R. F. Martin, Jr., *Proc. 2007 Int. Conf. on Comp. Math. Methods in Sci. and Engr.*, B. Wade, ed., 279–288, 2007.
13. **Computational Science Demands a New Paradigm**, D. E. Post and L. G. Votta, *Physics Today* **58** (1), 35–41 (2005).
14. **Teaching Computational Physics as a Laboratory Science**, R. L. Spencer, *Am. J. Phys.* **73**, 151–153 (2005).
15. **Survey of Computational Science Education**, C. Swanson, a long-running, high-quality survey that lists programs and courses in all computational sciences, and contacts entries to ensure viability. A somewhat dated 2003 edition is at [www.krellinst.org/services/technology/CSE\\_survey](http://www.krellinst.org/services/technology/CSE_survey).
16. **Elements of Computational Science Education**, O. Yaşa and R. Landau, *SIAM Rev.* **45**, 787–805 (2003); general program characteristics, contents, and balance of subjects.
17. **Graduate Education in Computational Science and Engineering**, SIAM Working Group on CSE Education, *SIAM Rev.* **43**, 163–177 (2001); describes CSE contents and programs, [www.siam.org/students/resources/report.php](http://www.siam.org/students/resources/report.php).
18. **Undergraduate Computational Science and Engineering Education**, SIAM Working Group on CSE Undergraduate Education, (apparently not published), [www.siam.org/about/pdf/CSE\\_Report.pdf](http://www.siam.org/about/pdf/CSE_Report.pdf).

## III. JOURNALS

There are many computer science and IEEE journals dealing with aspects of computing important to CP, and I specify just some next. The main journals containing discussions of computational education developments appear to be *Computing in Science & Engineering*, *SIAM Review*, and *American Journal of Physics*, with the others having an occasional education paper.

19. **Assoc. Computing Mach. J. on Ed. Resources in Computing**, [jeric.acm.org](http://jeric.acm.org).
20. **American Journal of Physics**, devoted to instructional and cultural aspects of physical science, with increasing interest in CP, [scitation.aip.org/ajp](http://scitation.aip.org/ajp).
21. **Annual Reviews of Computational Physics** (World Scientific, Singapore), contains at least nine volumes on computational research topics.
22. **Computer Physics Communications Program Journal**, started in 1969 as a pioneer in establishing importance of computer programs in the development of CP. Associated library now contains more than 2000 programs, [www.sciencedirect.com/science/journal/00104655](http://www.sciencedirect.com/science/journal/00104655).
23. **Computing in Science & Engineering (CiSE)**, joint publication of the IEEE Computer Society and the American Institute of Physics, merging (in 1999) AIP's *Computers in Physics* and IEEE's *Computational Science and Engineering*. More a readable magazine than an archival journal; September/October 2006 issue devoted to *Computation in Physics Courses*, [cise.aip.org](http://cise.aip.org).
24. **European Journal of Physics**, the European Physical Society's phys-

ics education journal, [www.iop.org/EJ/journal/0143-0807](http://www.iop.org/EJ/journal/0143-0807).

25. **HPCwire**, a weekly newsletter with plenty of advertisements, but also the latest news about high-performance computing trends, [www.hpcwire.com](http://www.hpcwire.com).
26. **IEEE Computer Magazine**, covers all aspects of computer science, computer engineering, technology, and applications, [www.computer.org/portal/site/computer/index.jsp](http://www.computer.org/portal/site/computer/index.jsp).
27. **IEEE Transactions of Learning Technologies**, slated to launch in 2008, will publish archival research papers on topics such as simulation systems for education, [www.computer.org/portal/site/transactions/](http://www.computer.org/portal/site/transactions/).
28. **Journal of Computational Physics** contains a broad spectrum of research papers, [www.elsevier.com/wps/find/journaldescription.cws\\_home/622866/description#description](http://www.elsevier.com/wps/find/journaldescription.cws_home/622866/description#description).
29. **Linux Journal**, a monthly magazine of the Linux community and a good place to learn about available tools, [www.linuxjournal.com](http://www.linuxjournal.com).
30. **Mathematics and Computers in Simulation**, *Transactions of IMACS*, [www.elsevier.com/wps/find/journaldescription.cws\\_home/505615/description#description](http://www.elsevier.com/wps/find/journaldescription.cws_home/505615/description#description).
31. **SIAM (Society for Industrial and Applied Mathematics) Review**, has been a leader in computational science/mathematics, and particularly CSE education, [epubs.siam.org/SIREV/sirev\\_toc.html](http://epubs.siam.org/SIREV/sirev_toc.html).
32. **SIAM Journal on Scientific Computing**, research articles on numerical methods and techniques for scientific computation, [www.siam.org/journals/sisc.php](http://www.siam.org/journals/sisc.php).
33. **Bull. ACM Special Interest Group in CS Ed. (SIGCSE)**, [www.sigcse.org/publications](http://www.sigcse.org/publications).
34. **Miscellaneous Education Journals**, (search on acronym) Ass. for Adv. of Computing in Ed. (AACE), *J Computers in Math. & Sci. Teach.* (JCMST), *Australian J. Ed. Tech.* (AJET), *Brit. J. Ed. Tech.* (BJET), *CS Ed.*, *Computers & Ed.*, *Computers in Ed. J.*, *Ed. Tech. & Soc.*, *Ed. Tech. Res. & Dev.* (ETR&D), *Ed. Tech. Rev.* (ETR), *Human-Computer Int.*, *IEEE Trans. on Ed.*, *Int. J Human-Computer Int.*, *Int. J. of Human-Computer Studies (IJHCS)*, *J. Res. on Tech. in Ed.* (JRTE), *J. of Ed. Computing Res.*, *J. of Engr. Ed.* (IJEE), *J. of Res. on Computing in Ed.* (JRCE), *User Modelling & User-Adapted Interaction*.

## IV. CONFERENCES AND ORGANIZATIONS

I hear and think often about the difficulties funding agencies and university faculties appear to have in dealing with multidisciplinary fields. Nevertheless, many CP practitioners find support in the resources and colleagues of the associated disciplines. In this section I indicate the websites of organizations that support computational science/physics. See also *Courses and Programs* in Section II and *Digital Libraries* in Section VIII.

### A. Conferences

35. **Assoc. for Computing Machinery**, supports advancing computing as a science and a profession; see also their digital library (Ref. 270), [www.acm.org](http://www.acm.org); **SIGCSE Technical Symposia on CS Ed.**, [db.grinnell.edu/sigcse/sigcse2008/Program/Program.asp](http://db.grinnell.edu/sigcse/sigcse2008/Program/Program.asp).
36. **APS Division of Computational Physics (DCOMP) Annual Meetings**, explores the use of computers in physics research and education. At times the division has stand-alone meetings, or with international conferences on CP, [units.aps.org/units/dcomp](http://units.aps.org/units/dcomp).
37. **International Conferences on Computational Science (ICCS)**, a series of yearly conferences in nice places: Krakow 2008, Beijing 2007, Reading, 2006, Atlanta 2005, Krakow 2004, Melbourne/St. Petersburg 2003, Amsterdam 2002, San Francisco 2001, [www.iccs-meeting.org](http://www.iccs-meeting.org).
38. **Conferences on Computational Physics (CCP)**, continues American and European Physical Societies' Physics Computing conferences: Boston 1989, Amsterdam 1990, San Jose 1991, Prague 1992, Albuquerque 1993, Lugano 1994, Pittsburgh 1995, Krakow 1996, Santa Cruz 1997, Granada 1998, Atlanta 1999, Brisbane 2000, Aachen 2001, San Diego 2002, Beijing 2003, Genova 2004, Los Angeles 2005, Gyeongju 2006, Brussels 2007, [ccp2007.ulb.ac.be](http://ccp2007.ulb.ac.be).



39. **European Physical Society Computation Group**, sponsors many research conferences, [www.eps.org/directory/epsentity\\_view?uid=Computational+Physics+Group](http://www.eps.org/directory/epsentity_view?uid=Computational+Physics+Group). See especially Ref. 45.
40. **girep: International Research Group on Physics Teaching**, maintains a series on conferences throughout Europe focused on innovation, [www.girep.org](http://www.girep.org).
41. **Geowall Consortium**, uses new projection technologies and fast graphics cards to visualize earth dynamics and aid understanding of spatial relationships, [geowall.geo.lsa.umich.edu/home.html](http://geowall.geo.lsa.umich.edu/home.html).
42. **Gordon Research Conferences**, in addition to focusing on traditional research topics, also examines issues related to computation and education, [www.grc.org](http://www.grc.org).
43. **IEEE Computer Society**, dedicated to advancing the theory, practice, and application of computer and information-processing technology, [www.computer.org/portal/site/ieeecs/index.jsp](http://www.computer.org/portal/site/ieeecs/index.jsp).
44. **IEEE Education Activities**, courses and more, [www.ieee.org/web/education/home](http://www.ieee.org/web/education/home).
45. **International Conference Multimedia Physics Teaching and Learning**, continues the series of workshops: Parma 2002, Prague 2003, Graz 2004, Berlin 2005, and Szeged 2006. Sponsored by the European Physical Society, [www.eps.org/conferences/europhysics\\_conferences/mptl-12/](http://www.eps.org/conferences/europhysics_conferences/mptl-12/).
46. **NSF's CISE Pathways to Revitalized Undergraduate Computing Education**, [www.nsf.gov/div/index.jsp?org=CNS](http://www.nsf.gov/div/index.jsp?org=CNS).
47. **SCXY Education Program**, workshop at the SuperComputing conferences for teachers designed to bring more computation into the classroom, [sc07.supercomputing.org/?pg=education.html](http://sc07.supercomputing.org/?pg=education.html).
48. **SIAM Conferences on Computational Science and Engineering**, promotes computational science and engineering as an academic discipline, and simulation as mode of scientific discovery, [www.siam.org/meetings/cse07](http://www.siam.org/meetings/cse07).
49. **Australian Computing Education Conference (ACE)**, a national conference of groups with long-term interest in nonstandard education, [www.acec2008.info](http://www.acec2008.info).
50. **ACM Special Interest Group in CS Ed. (SIGCSE) Technical Symposium**, focus on developing, implementing and/or evaluating computing programs, curricula, and courses, [www.cs.potsdam.edu/sigcse07/](http://www.cs.potsdam.edu/sigcse07/).
51. **International Computing Educational Resource (ICER) Workshop**, focuses on how people come to understand computational processes and devices, [www.cc.gatech.edu/conferences/icer2007](http://www.cc.gatech.edu/conferences/icer2007).
52. **Koli Calling – Baltic Sea Conference Computing Educational Resource**, teaching and learning practice with research in computing education, [cs.joensuu.fi/kolistelut](http://cs.joensuu.fi/kolistelut).

## B. Supercomputer centers and grids

The various supercomputer centers are not only an excellent resource for state-of-the-art computational tools and research, they also contain valuable resources classified under Education, Outreach, and Training. More recently, the grid consortia have also become valuable resources.

53. **Arctic Region Supercomputing Center**, [www.arsc.edu](http://www.arsc.edu).
54. **Cornell Theory Center**, [www.tc.cornell.edu](http://www.tc.cornell.edu).
55. **Department of Energy Advanced Scientific Computing Advisory Committee**, [www.sc.doe.gov/ascr/ASCAC/ASCAC.html](http://www.sc.doe.gov/ascr/ASCAC/ASCAC.html).
56. **Maui High Performance Computing Center**, affiliated with the Univ. of New Mexico, [www.mhpc.edu](http://www.mhpc.edu).
57. **National Center for Atmospheric Research (NCAR)**, [www.ucar.edu](http://www.ucar.edu).
58. **National Center for Supercomputer Applications at Univ. of Illinois**, a leader in outreach and visualizations, with petascale computing on the way, [www.ncsa.uiuc.edu](http://www.ncsa.uiuc.edu).
59. **National Energy Research Scientific Computing Center (NERSC)**, Department of Energy Office of Science facility focused on scientific discovery, [www.nersc.gov](http://www.nersc.gov).
60. **Oak Ridge National Center for Computational Sciences**, associated with various agencies and computers, [nccs.gov](http://nccs.gov).
61. **Ohio Supercomputer Center**, a diverse organization with a strong

emphasis on education (see also Ralph Regula School of Computational Science), [www.osc.edu](http://www.osc.edu).

62. **Open Science Grid (OSG)**, focused on meeting the computing and data management needs of researchers, especially collaborative science requiring high throughput computing, [www.opensciencegrid.org](http://www.opensciencegrid.org).
63. **Pittsburgh Supercomputing Center**, many programs, including strong education, workshops, outreach, and training for local facilities, as well as grants for computer time, [www.psc.edu](http://www.psc.edu).
64. **San Diego Supercomputing Center**, many programs, including strong education, workshops, outreach, and training for local facilities, as well as grants for computer time, [www.sdsc.edu](http://www.sdsc.edu).
65. **TerraGrid**, an open combination of leadership class resources at nine partner sites, with links to other grids, [www.teragrid.org/about/](http://www.teragrid.org/about/). Includes: Open Science Grid, Special Priority and Urgent Computing Environment, Massive Pulsar Surveys, Nat. Virtual Observatory, Linked Environments for Atmos. Discovery, Comp. Chem. Grid, Comp. Sci. and Engr. Online, Net. for Earthquake Engr. Sim., Net. for Comp. Nanotechn. and nanoHUB, TeraGrid Geographic Info. Sci. Gateway, CIG Sci. Gateway for Geodynamics Community, The Earth System Grid, Nat. Biomed. Comp. Resource, Developing Soc. Info. Data Grid, Neutron Sci. TeraGrid Gateway, Bio. and Biomed. Sci. Gateway, Open Life Sci. Gateway, The Telescience Project, Grid Analysis Environment, TeraGrid Visualization Gateway.
66. **Texas Advanced Computing Center**, many resources, with education under "EOT," [www.tacc.utexas.edu](http://www.tacc.utexas.edu).

## C. Groups with CSE education focus

67. **Engaging People in CyberInfrastructure (EPIC)**, a diverse team of educators with members throughout the country, [www.eotopic.org](http://www.eotopic.org).
68. **Krell Institute**, emphasizes the development of computational workforce and communicating science. See computational science fellowship programs, Undergraduate Computational Engineering and Sciences (UCES) Award, computational science and engineering education undergraduate programs listing, [www.krellinst.org](http://www.krellinst.org).
69. **Ralph Regula School of Computational Science (RRSCS)**, an Ohio statewide program at all levels of education in which participating colleges and universities confer degrees and certificates, [www.rscs.org](http://www.rscs.org).
70. **Shodor Education Foundation**, [www.shodor.org](http://www.shodor.org); **Computational Science Education Reference Desk (CSERD)**, a pathway project of the National Science Digital Library (Ref. 277), [www.shodor.org/refdesk](http://www.shodor.org/refdesk); **National Computational Science Institute (NCSI)**, provides on-the-road workshops, [www.computationalscience.net](http://www.computationalscience.net).
71. **Special Interest Group in CS Education, Assoc. for Computing Machinery (SIGCSE)**, [db.grinnell.edu/sigcse/sigcse2008/Program/Program.asp](http://db.grinnell.edu/sigcse/sigcse2008/Program/Program.asp); see especially their education links, [www.sigcse.org/topics](http://www.sigcse.org/topics).

## V. BOOKS

The books listed here tend to be of a general nature appropriate for course use. I do not list the large number of books dealing with computation in specific physics disciplines, although I do indicate some specialized applied math books. Some of these texts provide valuable resource lists in the form of annotated bibliographies indicating specialized books and articles for individual topics. There are some very good books that I leave out because their computer language is just too obsolete, although I do include some out-of-print books that are still available.

### A. CP books

72. **Computation and Problem Solving in Undergraduate Physics**, D. M. Cook. The self-published product of a project that incorporated various forms of computation throughout an undergraduate curriculum. Alternate versions available employing languages specified by individual readers. [www.lawrence.edu/dept/physics/ccli](http://www.lawrence.edu/dept/physics/ccli).

73. **Computer Modeling: From Sports to Spaceflight, From Order to Chaos**, M. A. Danby (Willmann–Bell, Richmond, 1997). Programs for MS-DOS are dated, but great subject mix. (E, I)
74. **Introduction to Computational Physics**, M. L. DeJong (Addison–Wesley, Reading, 1991). More elementary than most other texts, with detailed pseudocodes and Basic programs. Appears to be out of print. (E)
75. **A First Course in Computational Physics**, P. L. DeVries (Wiley, New York, 1994). Good coverage of the basic subjects with careful pedagogy. Microsoft FORTRAN codes on floppy disk. Appears to be out of print. (E)
76. **Understanding Molecular Simulation: From Algorithms to Applications**, 2nd ed., D. Frenkel and B. Smit (Academic Press/Elsevier, 2001, Amsterdam). Although specialized, molecular dynamics is a big part of CP, and this text has case study FORTRAN codes at [molsim.chem.uva.nl/frenkel\\_smit/README.html](http://molsim.chem.uva.nl/frenkel_smit/README.html). (I, A)
77. **Numerical Methods for Physics**, 2nd ed., A. L. Garcia (Prentice Hall, Upper Saddle River, 2000). Concise, yet covers a good number of computational and physics topics. MATLAB, FORTRAN, and C++ codes available from website, [www.algarcia.org/nummeth/nummeth.html](http://www.algarcia.org/nummeth/nummeth.html). (I)
78. **Computation in Modern Physics**, 3rd ed., W. R. Gibbs (World Scientific, Singapore, 2006). First half of text deals with an assortment of computational techniques, while second half applies them to quantum-mechanical problems in nuclear physics. (I, A)
79. **Computational Physics**, 2nd ed., N. Giordano and H. Nakanishi, (Prentice Hall, Saddle River, 2005). An excellent text with ample explanations and interesting applications. Presents only pseudocodes, which makes the text language independent, but with higher start-up costs. No CD, but author’s website provides some TRUE BASIC and FORTRAN codes, [www.physics.purdue.edu/~hisao/book](http://www.physics.purdue.edu/~hisao/book). (I)
80. **An Introduction to Computer Simulation Methods**, 3rd ed., H. Gould, J. Tobochnik, and W. Christian (Addison–Wesley, Reading, 2006). A new edition of a well-tested and excellent text, now incorporating Java and the Open Source Physics library (which has strong object-orientation and some overhead), but with a continued emphasis on statistical physics. (I)
81. **Computational Methods in Physics, Chemistry, and Biology**, P. Harrison, (Wiley, New York, 2001). An unusual, concise book that tries to inspire students by introducing many CP topics at an elementary level without much in the way of background. C codes in text. (E)
82. **Computer Simulation Using Particles**, R. W. Hockney, and J. W. Eastwood (Adam Hilger, Bristol, 1988). A lot can be simulated with just the motion of particles. (A)
83. **Introductory Computational Physics**, A. Klein and A. Godunov (Cambridge U. P., Cambridge, 2006). Careful treatment of a few basic topics, not many applications. Useful preparation for those using CERN’s ROOT framework. Programs mainly in C++, with calls to FORTRAN, using LINUX and free software packages. (I)
84. **Computational Physics**, S. E. Koonin (Benjamin, Menlo Park, 1986); FORTRAN edition, S. E. Koonin and D. Meredith (Westview Press/Perseus Books, Cambridge, 1998). Possibly the first text on CP; it is serious, but a challenge to use with undergraduates. BASIC and FORTRAN codes. (I–A)
85. **A Survey of Computational Physics**, *Introductory Computational Science*, R. H. Landau, M. J. Paez, and C. C. Bordeianu (Princeton U. P., Princeton, 2008). A fuller range of topics than Ref. 86, with a computational science and operational viewpoint. Procedural Java codes in the text, FORTRAN, and C codes and tutorials on the CD, and video lectures on the Web, [physics.oregonstate.edu/~rubin](http://physics.oregonstate.edu/~rubin). (I)
86. **Computational Physics, Problem Solving with Computers**, 2nd ed., R. H. Landau, M. J. Paez, and C. C. Bordeianu (Wiley–VCH, Berlin, 2007). Java version of the 1st ed., R. H. Landau and M. J. Paez (Wiley, New York, 1997). Extended survey in Ref. 85. (I)
87. **Computational Techniques in Physics**, P. K. MacKeown and D. J. Newman (Adam Hilger, Bristol, 1987). Some unusual and thoughtful applications, although not a broad spectrum of computational subjects. No programs in text, but a CD was provided by publisher. (I)
88. **An Introduction to Computational Physics**, 2nd ed., T. Pang (Cambridge U. P., Cambridge, 2006). First half of text deals with computational tools, second half with applications, including unusual ones on genetic programming and renormalization group. Java programs in text, with FORTRAN and C programs on author’s web page, [www.physics.unlv.edu/~pang/cp.html](http://www.physics.unlv.edu/~pang/cp.html). (I)
89. **Theoretical Physics on the Personal Computer**, 2nd ed., E. W. Schmid, G. Spitz, and W. Lösch (Springer–Verlag, Berlin, 2000). The 2nd edition of one of the earliest CP texts contains a great deal of physics and insight for this small book, although on limited topics. Fortran, CD. (I, A)
90. **A Physicist’s Guide to MATHEMATICA**, P. Tam (Elsevier, Oxford, 1997). Essentially a CP course using MATHEMATICA. (I)
91. **Computational Physics**, J. M. Thijssen (Cambridge U. P., Cambridge, 1999). A graduate/research-level text delving into the theoretical underpinnings of CP, with emphasis on condensed matter. Programming implications discussed, but no programs printed in text, no accompanying CD, and only few programs on the web, [www.tn.tudelft.nl/tn/People/Staff/Thijssen/comphybook.html](http://www.tn.tudelft.nl/tn/People/Staff/Thijssen/comphybook.html). (A)
92. **Computational Physics, An Introduction**, 2nd ed., F. J. Vesely (Plenum Press, New York, 2001). Focuses on computational techniques, with a few serious applications, and with concise discussions. No codes. (E–A)
93. **Introduction to Computer Simulation**, M. M. Woolfson and G. J. Perl (Oxford U. P., Oxford, 1999). Small number of topics. (E, I)

## B. Applied math and CSE books

94. **Numerical Methods that Work**, F. S. Acton (Math. Assoc. of Amer., Washington, DC, 2007) is a 1970 classic reissued by the MAA. (E, I)
95. **The Illustrated Wavelet Transform Handbook**, P. S. Addison (Institute of Physics Publishing, Bristol, 2002). (I)
96. **Computational Methods for Applied Science and Engineering**, M. G. Ancona (Rinton Press, Princeton, 2002), uses MATHCAD with digital text available. (E, I)
97. **Engineering Programming, C, MATLAB, and Java**, M. Austin and D. Chancogne (Wiley, New York, 1999). A set of four tutorials. (E)
98. **The DFT, An Owner’s Manual**, W. L. Briggs and V. E. Henson (SIAM, Philadelphia, 1995). Great name for serious study. (A); see too **Fast Fourier Transforms for Experimentalists**, D. Donnelly and B. Rust *Comput. Sci. Eng.* **7**, 80–88 (2005), one of a series of articles. (I)
99. **Numerical Methods for Engineers with Software and Programming Applications**, 5th ed., S. C. Chapra (McGraw–Hill, New York, 2006). MATLAB and EXCEL programs. An impressively broad view with strong pedagogy. (E, I)
100. **Computational Science Education Project**, dated, but a high-quality, early exploration into the use of the Web for computational education, [csep1.phy.ornl.gov/csep.html](http://csep1.phy.ornl.gov/csep.html). (E)
101. **Projects in Scientific Computing**, R. E. Crandall (Springer–Verlag, New York, 2000). Excellent selection of topics with focus on algorithms. (I)
102. **Numerical Algorithms with C**, G. Engeln–Müllges and F. Uhlig, translated by M. Schon and F. Uhlig (Springer, Berlin, 1996). Rivals and complements Press *et al.*’s Numerical Recipes. Also FORTRAN version available. (I)
103. **Computer Algebra Recipes, A Gourmet’s Guide to Mathematical Models of Science**, R. H. Enns and G. C. McGuire (Birkhäuser/Springer, Berlin, 2001). Good ideas for CP education projects. Also, *Computer Algebra Recipes for Mathematical Physics*. (E, I)
104. **Numerical Methods**, 3rd ed., J. D. Faires and R. Burden (Thompson Brooks/Cole Toronto, 2003). Also an important numerical analysis text by same authors. (I)
105. **Numerical Methods Using MATHCAD**, L. Fausett (Prentice Hall, Upper Saddle River, 2002). See also *Applied Numerical Analysis Using MATLAB*, 2nd ed. (Prentice Hall, Upper Saddle River, 2007), also designed for undergraduate science majors. (E, I)
106. **An Introduction to High Performance Scientific Computing**, L. D. Fosdick, E. R. Jessup, C. J. C. Schauble, and G. Domik (MIT Press, Cambridge, 1996). An exceptionally complete, yet readable, survey of computer-science basics and tools of computational science with a few applications. (I)
107. **Numerical Methods for Engineers and Scientists: An Introduction with Applications using Matlab**, A. Gilat and V. Subramaniam

(Wiley, New York, 2007). Also a similar Numerical Methods with MATLAB by same publisher. (I)

108. **Fundamentals of Wavelets**, J. C. Goswami and A. K. Chan (Wiley, New York, 1999). (I)
109. **Scientific Computing**, 2nd ed., M. T. Heath (McGraw-Hill, New York, 2002). A practical guide to numerical computing. No codes, but useful book site, <http://www.cse.uiuc.edu/heath/scicomp>. (E, I)
110. **Introduction to Scientific Computation and Programming**, D. T. Kaplan (Thompson, Brooks/Cole, Belmont, 2004). MATLAB code fragments in text. Fundamentals of computer science for scientists, with very nice examples. (E)
111. **A First Course in Scientific Computing**, *Symbolic, Graphic, and Numerical Modeling Using MAPLE, Java, MATHEMATICA, and FORTRAN 90*, R. H. Landau (Princeton U. P., Princeton, 2005). Also LATEX and alternate versions of text in MATHEMATICA and FORTRAN 90 on the CD along with notebooks, workbooks, and codes. (E)
112. **Numerical Methods for Mathematics, Science, and Engineering**, 2nd ed., J. Mathews (Prentice Hall, Englewood Cliffs, 1992). (I)
113. **Numerical Recipes**, 3rd ed. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling (Cambridge U. P., Cambridge, 2007). A groundbreaking classic from a scientist's point of view, with erudite writing no less. All codes from previous versions in all computer languages are included in this new edition. (I, A)
114. **Introduction to Computational Science**, Introduction to Computational Science: Modeling and Simulation for the Sciences Angela B. Shiflet, George W. Shiflet and Shiflet (Princeton U. P., Princeton, 2006). An introduction to computer modeling based on successful classes for liberal arts students. (E)
115. **High Performance Computing**, 2nd ed., C. Severance and K. Dowd (O'Reilly, Sebastopol, 1998). Contains useful, down-to-earth discussions of how architecture influences performance and how to benchmark and tune codes. (A)
116. **Atlas for Computing Mathematical Functions: An Illustrated Guide for Practitioners with Programs in FORTRAN and MATHEMATICA**, W. J. Thompson (Wiley, New York, 1997), appears to be out of print The Abramowitz and Stegun for the computer age. (I)
117. **Computing for Scientists and Engineers**, W. J. Thompson (Wiley, New York, 1992). One of the earlier books that introduced us to the field, and though thin, contains much insight about numerical methods. (E)
118. **Computational Methods in Physics and Engineering**, 2nd ed., S. S. M. Wong and S. S. Wong (World Scientific, Singapore, 2003). Full discussions of computational techniques from a physicist's point of view, but little in the way of applications. FORTRAN codes on CD and Web, [www.worldscibooks.com/physics/3365.html](http://www.worldscibooks.com/physics/3365.html). (I)
119. **A First Course in Computational Physics and Object-Oriented Programming with c++**, D. Yevick (Cambridge U. P., Cambridge, 2005). Basically a c++ programming course with a few examples. (I)
120. **Introduction to Scientific Programming**, *Computational Problem Solving Using MAPLE and C*, J. L. Zachary (Springer/Telos, New York, 1996). Early effort to teach CSE to lower-division students. (E)

## VI. TOOLS, LANGUAGES, AND ENVIRONMENTS

Everyone seems to agree that visualization is important in understanding physics concepts and understanding the output of computations. Accordingly, I place Visualization first in this section. Note that the modern role of visualization in CP includes (1) concurrent visualization and analysis, (2) on-the-fly interactive visualization, and (3) postsimulation visualization reading data.

In contrast to visualization, not everyone agrees whether compiled/interpreted languages or problem solving environments such as MAPLE and MATHEMATICA are best for a CP course. Indeed, disagreement may be even more pronounced when the advocates of a compiled language discuss which compiled language should be advocated. In practice, most CP texts adopt a compiled/interpreted language, while most texts that incorporate computation into traditional physics

disciplines adopt a problem-solving environment. This probably reflects the view that compiled languages show best how an algorithm is implemented and what level of precision is being demanded, while being in a form that encourages exploration; all items of importance in a CP course.

In my view, Java's attention to precision, useful error messages, and object-orientation make it good for scientific computing, while its universality, free compilers, and use in the commercial sector make it popular with students. However, it is not as efficient or as well supported for HPC and parallel processing as are FORTRAN and *c*, the latter two having highly developed compilers and many more scientific subroutine libraries available. FORTRAN, in turn, is still the dominant language for HPC, with FORTRAN 90/95 being a surprisingly nice, modern, and effective language; but alas, it is hardly taught by any CS departments, and compilers can be expensive. *C*, and its object-oriented brother *c++*, are good for HPC, have good free compilers and libraries available, but may be too flexible with precision and memory access for beginners to learn good scientific programming practices. Python, a new, interpreted language that has garnered a small but devoted following, is free, user friendly, good for beginning programming, and has a nice 3D graphics library. However, it is still developing, and while not intended for HPC, this is changing with time. In any case, I recommend that students become familiar with at least two compiled languages (preferably one being FORTRAN), and that they keep their focus on how clearly the algorithm is implemented and not on the minimum number of statements that is needed to do it.

As I use the term *problem-solving environment* in this Resource Letter, I refer to MAPLE and MATHEMATICA, and their less-popular cousins, such as DERIVE (now obsolete) and MACYSMA/MAXIMA. All are capable of symbolic manipulations. In our experience [which includes a 1:1 translation of the Maple version of our "First" book (Ref. 111) into Mathematica], these environments have much more in common than they do differences, and once you know one, learning the other is essentially a question of grammar. Another group of high-level languages that is sometimes also called problem-solving environments, include MATLAB (especially popular with engineering departments), its free cousin GNU OCTAVE, and MATHCAD. These are essentially command-line interfaces for running numerical calculations that employ built-in libraries (such as LAPACK) for linear algebra, mathematical methods, visualization, signal processing, and such. These environments are powerful and work well, but are more likely to separate the user more from the algorithms than compiled languages (a separation, I daresay, some users may prefer).

For pedagogical purposes in CP classes, I recommend that students use compile and execute commands directly from a shell. However, when projects get complicated, such as with multiple data sets or multiple computers, then an integrated development environment (IDE), also known as a studio or a computational framework, is often used by researchers and professional programmers. An IDE provides an integrated, tightly knit set of components that may include a code editor, compiler, a build tool, a version control system, a debugger, a class browser, an object inspector, and a class hierarchy diagram. Examples include CACTUS, KDEVELOP, ECLIPSE, NETBEANS, and ROOT.



## A. Visualization

121. **Advanced Visual Systems (AVS5)**, a commercial off-the-shelf (COTS) and expensive product with similar properties as OpenDX and Paraview, but maybe more so, [www.avs.com/avs5.html](http://www.avs.com/avs5.html). (I)
122. **CAIDA**, Cooperative Assoc. for Internet Data Analysis contains tools and information, [www.caida.org/tools](http://www.caida.org/tools). (I)
123. **ChromaDepth 3D Imaging Glasses**, inexpensive 3D vision via red-blue separation of color output, [www.3dglassesonline.com/3d-chromadepth-glasses](http://www.3dglassesonline.com/3d-chromadepth-glasses). (E)
124. **Discover Data Portal**, education about visualization, [education.sdsc.edu/discoverdata](http://education.sdsc.edu/discoverdata). (E)
125. **Geometric Tools for Computer Graphics**, P. J. Schneider and D. H. Eberly (Morgan Kaufmann/Elsevier, San Diego, 2003). How to implement the mathematics of visualizations. (I)
126. **Geo. Tech. Scientific Visualization Tutorials**, good for general ideas but not specific tools, [www.cc.gatech.edu/scivis/tutorial/tutorial.html](http://www.cc.gatech.edu/scivis/tutorial/tutorial.html). (E)
127. **Gnuplot** is a classic. It is a free, nearly universal, command-line driven, interactive data and function plotting utility. Although Grace produces higher, publication-quality 2D figures by default, Gnuplot is powerful and flexible, and our standard for 3D (surface) plots. Also for Windows, which Grace is not, [www.gnuplot.info](http://www.gnuplot.info). (E)
128. **Grace**, a free WYSIWYG 2-D plotting tool for the X Window System (Unix/Linux), or Windows with Cygwin. This descendant of *ACE/gr* and *Xmgr* is my choice for publication quality, 2D graphics, but it cannot create 3D (surface) plots, [plasma-gate.weizmann.ac.il/Grace](http://plasma-gate.weizmann.ac.il/Grace). (E)
129. **IDL, The Data Visualization and Analysis Platform**, a powerful, but costly COTS package, [www.ittvis.com/idl](http://www.ittvis.com/idl). (A)
130. **IN-SPIRE Visual Document Analysis**, information-visualization software developed by Pacific Northwest National Laboratory, [inspire.pnl.gov/](http://inspire.pnl.gov/). (A)
131. **JImage, Image Processing, and Analysis in Java**, powerful, but easier to use than NCSA Suite, [rsb.info.nih.gov/ij/](http://rsb.info.nih.gov/ij/). (I)
132. **NIST Scientific Applications and Visualization Group**, gives examples of visualization, [math.nist.gov/mcsd/sav/vis](http://math.nist.gov/mcsd/sav/vis). (E, I)
133. **MAPLE and MATHEMATICA** produce beautiful visualization of mathematical functions with analytic forms, but are less convenient for large numerical datasets (unless you are already familiar with Maple/Mathematica). See Section VI G. (E, I)
134. **MATLAB** provides outstanding visualization tools for both numerical data and simulations. Version 7.X adds improved interactivity. See Section VI F. (E, I)
135. **NCAR Graphics**, an open source time-tested UNIX package of FORTRAN and *c* utilities for contours, maps, vectors, streamlines, weather maps, surfaces, histograms, etc., [ngwww.ucar.edu](http://ngwww.ucar.edu). (A)
136. **NCSA Visualization Suite** is free and includes **Easy-Viz**, a simple tool designed for educators to visualize both 2D and 3D scalar data, and **NCSA Databridge**, an application that organizes data into common file formats for visualization, [education.ncsa.uiuc.edu/products/dvs.html](http://education.ncsa.uiuc.edu/products/dvs.html). (E)
137. **OPEN DX** is an open-source, LINUX/UNIX software package based on IBM's *Data Explorer* that provides industrial-strength visualization comparable to the COTS AVS, (Ref. 121). Capable of interactive 3D rendering and volume visualization (slicing and dicing) of N-D datasets. Uses graphical programs (networks) that resemble flowcharts, and can run under windows with Cygwin, [www.opendx.org](http://www.opendx.org). (I)
138. **OPENDX: Paths to Visualization**, D. Thompson, J. Braun, and R. Ford, (Visualization and Imagery Solutions, Missoula, 2001). (I)
139. **OPENGL**, a vendor-neutral low-level environment for developing portable, interactive 2D and 3D graphics applications. Essentially the assembler language of computer graphics, [www.opengl.org](http://www.opengl.org). (A)
140. **ParaView, Parallel Visualization Application**, a free, yet powerful and beautiful, multiplatform visualization tool for large datasets that can be used on a single or a parallel computer, [www.paraview.org/](http://www.paraview.org/) New. Tutorial at Ref. 145. (A)
141. **plplot**, a free, easy to use, cross-platform library for 2D and 3D plots, callable from within both compiled and interpreted languages; not to be confused with PtPlot, [plplot.sourceforge.net](http://plplot.sourceforge.net). (E)

142. **PtPlot**, a 2D data plotter and histogram tool implemented in Java, part of UCB's Ptolemy IDE (Ref. 210). Our choice for graphical output from within Java programs, [ptolemy.eecs.berkeley.edu/java/ptplot](http://ptolemy.eecs.berkeley.edu/java/ptplot). (E)
143. **Scientific Computing and Imaging Institute**, home for open source SCIRun, BioPSE & map3d, [software.sci.utah.edu/scirun.html](http://software.sci.utah.edu/scirun.html). (I, A)
144. **Starlight Information Visualization System**, graphically depicts *information*, not the usual type of plots, [starlight.pnl.gov](http://starlight.pnl.gov). (I, A)
145. **SDSC Intro. to 3D Visualization**, J. Sales, useful tutorials, [education.sdsc.edu/sc07/introtoviz/](http://education.sdsc.edu/sc07/introtoviz/). See too [vis.sdsc.edu](http://vis.sdsc.edu). (E, I)
146. **Visual Display of Quantitative Information**, 2nd ed., E. R. Tufte (Graphics Press, Cheshire, 2001). A standard. (E)
147. **VisIt**, a free interactive parallel-visualization and graphical-analysis tool, developed by DOE labs for viewing scientific data on Unix and PC platforms, [www.llnl.gov/visit](http://www.llnl.gov/visit). (I)
148. **Visualization Toolkit**, an open-source *c++* class library for 3D graphics, image processing and visualization, with interpreted interface layers for Tcl/Tk, Java, and PYTHON, [public.kitware.com/VTK](http://public.kitware.com/VTK). (I)
149. **VolView**, a COTS powerful volume-visualization tool appropriate for research and medical application, [www.kitware.com/products/volview.html](http://www.kitware.com/products/volview.html). (A)

## B. FORTRAN

150. **Computing for Scientists: Principles of Programming with FORTRAN 90 and *c++***, R. J. Barlow and A. R. Barnett (Wiley, New York, 1998). Teaches *c* and FORTRAN side by side, a useful tack when learners already know one. (E, I)
151. **FORTRAN 90/95 for Engineers and Scientists**, 2nd ed., S. J. Chapman (McGraw-Hill, New York, 2004). Practical and effective. (E, I)
152. **FORTRAN 90 and Computational Science**, The Computational Science Education Project, a dated but high-quality early exploration in Web education, [csep1.phy.ornl.gov/pl/pl.html](http://csep1.phy.ornl.gov/pl/pl.html). (E)
153. **Univ. of Liverpool FORTRAN 90 Course Notes**, A. C. Marshall. The place I look for good and full explanations, [www.liv.ac.uk/HPC/F90page.html](http://www.liv.ac.uk/HPC/F90page.html). (I)
154. **Essential FORTRAN 90 AND 95**, L. Meissner (The Fortran Company, Tucson, 1997). (E, I)
155. **FORTRAN 95/2003**, M. Metcalf, J. Reid, and M. Cohen (Oxford, New York, 2004). Clear and straightforward. (E, I)
156. **FORTRAN 90 Programming**, T. M. R. Ellis, I. R. Phillips, and T. M. Lahey (Addison-Wesley, Wokingham, 1994). Both FORTRAN 90 and 77 with academic discussions. (E, I)

## C. *c*, *c++* (Object Oriented)

157. **Computing for Scientists: Principles of Programming with FORTRAN 90 and *c++***, R. J. Barlow and A. R. Barnett (Wiley, New York, 1998). Teaches *c* and FORTRAN side by side, a useful tack when learners already know one. (E, I)
158. **C for Scientists and Engineers**, R. Johnsonbaugh and M. Kalin (Prentice Hall, Upper Saddle River, 1997). Meant as a text, with page, [condor.depaul.edu/~mkalin/cse](http://condor.depaul.edu/~mkalin/cse). (E, I)
159. **A Book on C**, 4th ed., A. Kelley and I. Pohl (Addison-Wesley, Upper Saddle River, 1997). A useful tutorial that includes information on converting to Java and *c++* from *c*. Electronic materials available from author, [www.soe.ucsc.edu/~pohl/abc4.html](http://www.soe.ucsc.edu/~pohl/abc4.html). (A-I)
160. **The *c* Programming Language**, 2nd ed., B. Kernighan and D. Ritchie (Prentice-Hall, Englewood Cliff, 1988). The classical way to learn its creators. (E, I)
161. **Accelerated *c++*, Practical Programming by Example**, A. Koenig and B. E. Moo (Addison-Wesley, Upper Saddle River, 2000). Exceptional discussion of practical object-oriented programming, with electronic materials online at [www.acceleratedcpp.com](http://www.acceleratedcpp.com). (E, I)
162. **Practical C Programming**, 3rd ed., S. Oualline, (O'Reilly, 1997, Sebastopol). Also **Practical *c++* Programming**, 2nd ed. Short and to the point. (E, I)
163. **The *c++* Programming Language**, 3rd ed., B. Stroustrup (Addison-Wesley, Upper Saddle River, 1997). Written by creator of *c++*, with electronic materials at his site, [www.research.att.com/~bs/3rd.html](http://www.research.att.com/~bs/3rd.html). (I, A)

## D. Java

164. **Java for Engineers and Scientists**, G. J. Bronson (Thompson, Brooks/Cole, Toronto, 2003). Some readers may appreciate the well-worded explanations. (E, I)
165. **Understanding Object-Oriented Programming with Java**, T. Budd (Addison–Wesley, Reading, 1998). The place to go for a deep understanding of object-oriented programming with Java. (I)
166. **Teach Yourself Java in 21 Days**, 5th ed., R. Cadenhead and L. Lemay. A popular tutorial, now updated to Java 6. (Samms/Pearson, Indianapolis, 2007). (E, I)
167. **Java for Engineers and Scientists**, 2nd ed., S. J. Chapman (Pearson/Prentice Hall, Upper Saddle River, 2004). Discusses Java in the language of scientists and teaches good scientific programming practice. My favorite Java book. (E, I)
168. **Introductory Java for Scientists and Engineers**, R. Davies (Addison–Wesley, Harlow, 1999). A good introduction to Java for someone who already knows C or FORTRAN. (E)
169. **Java, How to Program**, 7th ed., H. M. Deitel and P. J. Deitel (Prentice Hall, Upper Saddle River, 2007). Some colleagues swear by it, I find it an overload. (I, A)
170. **Introduction to Programming Using Java, Version 4.0**, D. Eck (2002). A free, yet serious and excellent textbook, math.hws.edu/javanotes. (I)
171. **Easy Java Simulations**, F. Esquembre, a software tool designed for the creation of Java simulations, www.um.es/fem/Ejs/Ejs\_en. (E, I)
172. **Java in a Nutshell, Desktop Quick Reference**, 5th ed., D. Flanagan (O'Reilly, Sebastopol, 2005). Useful, concise, but not for beginners. (I, A)
173. **JavaTech: An Introduction to Scientific and Technical Computing with Java**, C. S. Lindsey, J. S. Tolliver, and T. Lindblad (Cambridge U. P., Cambridge, 2005). An extensive collection of materials, also assembled as a Web course, www.particle.kth.se/~lindsey/JavaCourse/Book/. (E, I)
174. **NMATH Product Suite (CenterSpace Software)**, building blocks for various applications callable from any .NET language (one that produces programs that execute within the Microsoft .NET framework, e.g., C#, VISUAL BASIC. NET, J# and c++/CLI.), www.centerspace.net. (A)
175. **Sun Java Developer's site**, the official version of Java, java.sun.com. See also the documentation site java.sun.com/docs and tutorials at java.sun.com/developer/onlineTraining. (I, A)

## E. Python

176. **Learn to Program Using Python**, A. Gauld (Addison–Wesley, Reading, 2000); also online. Another Web book that got turned into paper, www.freenetpages.co.uk/hp/alan.gauld. (E, I)
177. **Python/Vpython**, official website for python, www.python.org.
178. **Python: Batteries Included**, special issue of Comput. Sci. Eng. 9, (2007).
179. **Python First: Introduction to Computing with Python**, A. Radenski. A commercial digital book/course with text, slides, labs and quizzes, studypack.com/comp/course/view.php?id=232. (E)
180. **Python Subroutine Libes**, see Ref. 267. (I)
181. **An Introduction to Python—The Python Tutorial**, G. van Rossum and F. L. Drake, Jr. (Network Theory Ltd, Bristol, 2006); also online. The official Python tutorial, www.network-theory.co.uk/python/intro. (E, I)
182. **Vpython**, a free package including Python, an interactive development environment, a 3D graphics package, and a module for array processing, www.vpython.org.

## F. MATLAB, OCTAVE, and MATHCAD

183. **MATLAB Programming for Engineers**, 3rd ed., S. J. Chapman (Thompson, Toronto, 2005). MATLAB as a technical programming language. Password-protected online site for instructors. (E)
184. **Mastering MATLAB 7**, D. C. Hanselman and B. Littlefield (Prentice Hall, Upper Saddle River, 2005). M files on line. (E, I)

185. **MATLAB Guide**, 2nd ed. D. J. Higham and N. J. Higham, (SIAM, Philadelphia, 2005). A good reference for all levels, with electronic media on the Web, www.siam.org/books/ot75. (E, I)
186. **Numerical Computing with MATLAB**, C. Moler, (SIAM, Philadelphia, 2004); also available on line, www.mathworks.com. The author is the creator of MATLAB. (E, I)
187. **GNU OCTAVE**, an open source, free high-level language that is mostly compatible with MATLAB and shares many of the same features, www.gnu.org/software/octave/. (E, I)
188. **MATHCAD**, an environment for creating, documenting, and sharing engineering calculations. Like Matlab, has various libraries and packages (packs). www.ptc.com/products/mathcad/mathcad14/promo.htm. (E, I)
189. **MATLAB Tutorial**, free online tutorial, www.mathworks.com/academia/studentcenter/tutorials/launchpad.html (E, I)

## G. MAPLE and MATHEMATICA

190. **MATHEMATICA in Theoretical Physics**, 2nd ed., G. Baumann (Springer–Verlag, Berlin, 2005). (I)
191. **MAXIMA** computer algebra system is a descendant of MACSYMA and is now available as open source, maxima.sourceforge.net. (I)
192. **Essential MAPLE 7, An Introduction for Scientific Programmers**, R. M. Corless (Springer, New York, 2002). Emphasis on MAPLE programming. No apparent update to recent MAPLE. (E, I)
193. **MATHEMATICA for Scientists and Engineers**, R. Gass (Prentice-Hall, Upper Saddle River, 1998). With many physics applications and M 3.0 notebooks on CD; appears to be out of print. (I)
194. **Introduction to MAPLE**, 3rd ed., A. Heck (Springer, New York, 2003). My favorite MAPLE book. MAPLE 8 in text, MAPLE 11 worksheets on Web, remote.science.uva.nl/~heck/Maplebook/. (E, I)
195. **Mathematical Methods Using MATHEMATICA, or Students of Physics and Related Fields**, H. Hassani, (Springer, New York, 2004). A supplementary text for use in a math methods course. (E)
196. **Maplesoft's MAPLE Documentation**. Both printed manuals and free electronic version available, www.maplesoft.com/documentation\_center. (I, A)
197. **MAPLE, A Comprehensive Introduction**, R. Nicolaides and N. Walkington (Cambridge U. P., Cambridge, 1996). Only MAPLE V, but a nice introduction. (E)
198. **MATHEMATICA Navigator**, H. Ruskeepaa (Elsevier, Oxford, 2004). An 844 page complete treatment. (E, I)
199. **Physics with MAPLE**, F. Y. Wang (Wiley–VCH, Berlin, 2006). Emphasis on the physics more than Maple. (I)
200. **The MATHEMATICA Book**, 5th ed., S. Wolfram, (Wolfram Media, Champaign, 2003). At 1488 pages this should be complete. (I, A)
201. **MATHEMATICA for Physicists**, 2nd ed., R. L. Zimmerman and F. I. Olness, (Addison–Wesley, San Francisco, 2003). For physicists, with many applications. MATHEMATICA 4 and 5 notebooks on Web, darkwing.uoregon.edu/~phys600. (I, A)

## H. Development environments, and debuggers

An hour to design, a day to write, and a week to debug. Debugging complicated scientific programs can be the most time-consuming part of computational physics. Good programming practices and systematic code development as well as these tools can help.

202. **Cactus Problem Solving Environment**, an open-development platform used over many years by large international collaborations of physicists and computational scientists, www.cactuscode.org.
203. **Condor**, a project aimed at supporting high-throughput computing on large collections of distributed computing resources, www.cs.wisc.edu/condor.
204. **Eclipse**, an open source-development platform running in Windows, MacOS, and Linux, www.eclipse.org.
205. **GNU Debugger (gdb)**, shows what is going on inside a program while it executes or crashes, sourceware.org/gdb.
206. **GuardSoft Relative Debugger**, new debugging technology of relative debugging, guardsoft.com.



207. **Java NETBEANS**, released by Sun and is now open source and free. Runs on WINDOWS, LINUX, MAC OS X, and SOLARIS. A well-respected alternative to Eclipse, [www.netbeans.org](http://www.netbeans.org).
208. **KDevelop**, the development environment for KDE, the free graphical desktop environment for Linux and Unix, [www.kdevelop.org](http://www.kdevelop.org).
209. **TotalView Debugger**, a debugger for the multicore age, [www.total-viewtech.com](http://www.total-viewtech.com).
210. **PTOLOMY**, a Java-based software framework used to assemble and control interaction of concurrent simulation components. Contains the Java plotting program, **PtPlot**, [ptolemy.berkeley.edu/java/ptplot](http://ptolemy.berkeley.edu/java/ptplot).
211. **ROOT**, object-oriented framework for large scale data analysis, a c++ replacement of the popular PAW program developed at CERN, [root.cern.ch](http://root.cern.ch).
212. **Your Printer**, probably the most common and powerful debugging tool; you just include print statements as you program, and comment them out for production runs.

## I. Markup languages

A markup language is used to write the source code for a document that gets compiled into presentation form. The publishing of scientific documents and their storage in digital libraries often use markup languages. [Push the menu buttons in your browser to *View/Page Source* to see the “marks” for a Hyper Text Markup Language (HTML) document.]

213. **Docbook**, *The Definitive Guide*, N. Walsh and L. Muellner (O’Reilly, Sebastopol, 2003). The official documentation for the DocBook document type definition, available online at [www.docbook.org/tdg/en/html/docbook.html](http://www.docbook.org/tdg/en/html/docbook.html).
214. **Google Docs**, for sharing your documents on line, [www.google.com/google-d-s/intl/en/tour1.html](http://www.google.com/google-d-s/intl/en/tour1.html).
215. **Hermes**, a semantic XML, MathML, Unicode e-publishing tool for LATEX authored scientific articles, [hermes.roua.org](http://hermes.roua.org).
216. **LATEX: A Document Preparation System**, 2nd ed., L. Lamport (Addison–Wesley, Upper Saddle River, 1994). My first and favorite for simplicity and conciseness.
217. **LATEXML: A LATEX to XML Converter**, [dlmf.nist.gov/LaTeXXML/](http://dlmf.nist.gov/LaTeXXML/).
218. **LATEX Companion**, 2nd ed., F. Mittelbach, M. Goossens, J. Braams, D. Carlisle, and C. Rowley (Addison–Wesley, Upper Saddle River, 2004).
219. **Guide to LATEX**, 4th ed., H. Kopka and P. W. Daly (Addison–Wesley, Upper Saddle River, 2004). Not as friendly as Lamport, but has most everything needed.
220. **Learning LATEX**, D. F. Griffiths and D. Higham (SIAM, Philadelphia, 1997). Short but effective guide to LATEX 2e.
221. **MATHML**, a product of the W3C Math working group, provides a low-level foundation for the inclusion of mathematical expressions in digital documents, [www.w3.org/Math/](http://www.w3.org/Math/), with white paper at [www.w3.org/Math/XSL](http://www.w3.org/Math/XSL).
222. **Future Scientific Digital Documents with MathML, XML, and SVG**, R. Landau, D. Vediner, P. Wattanakasiwich, and K. Kyle, *Comput. Sci. Eng.* **4**, 77–86 (2002).
223. **Vector Markup Language (VML)**, an XML application defining format for encoding vector information, [www.w3.org/TR/NOTE-VML.html](http://www.w3.org/TR/NOTE-VML.html).
224. **W3C Home**, the home of the Work Wide Web Consortium (WWW = “the Web”), [www.w3.org](http://www.w3.org).
225. **W3C Math Home**, the home of the W3C’s views on how Math should be incorporated into Web documents, [www.w3.org/Math](http://www.w3.org/Math).
226. **XML, Extensible Markup Language**, a Web markup language that separates contents of text from presentation, [www.w3.org/XML](http://www.w3.org/XML).
227. **XHTML, The Extensible HyperText Markup Language**, a reformulation of HTML 4 in XML 1.0, [www.w3.org/TR/xhtml1](http://www.w3.org/TR/xhtml1).
228. **XML FAQ**, P. Flynn, frequently-asked questions about the extensible markup language, [www.ucc.ie/xml](http://www.ucc.ie/xml).

## J. SQL: Database languages

Present and future computation is focusing more on data-intensive computing with distributed databases, which means that computational physicists should know something about dealing with databases. A database management system is a collection of programs for storing, modifying, and extracting information from a database. The standard query language is structured query language (SQL).

229. **Database Management Systems**, 3rd ed., R. Ramakrishnan and J. Gehrke (McGraw–Hill, New York, 2003), [pages.cs.wisc.edu/~dbbook](http://pages.cs.wisc.edu/~dbbook), a popular textbook (E, I).
230. **A First Course in Database Systems**, 3rd ed., J. Ullman and J. Widom (Prentice–Hall, Upper Saddle River, 2007). (E)
231. **SQL.org**, SQL information, including MySQL, PostgreSQL, Oracle, mSQL and Microsoft SQL, [www.sql.org](http://www.sql.org). (I, A)

## K. Concept mapping

Concept maps are useful for displaying the relationships among the elements of a knowledge base, with the concepts represented as nodes and with arrows indicating links among the concepts. They are also useful when placing materials into digital libraries, where the concepts get linked to chunks of information and thus become content maps. Maps are created with the tools:

232. **VUE, The Visual Understanding Environment**, Tufts Univ. Academic Technology, [vue.uit.tufts.edu](http://vue.uit.tufts.edu).
233. **AAAS Strands Maps**, traces paths followed by learners, [www.project2061.org/publications/rsl/online/GUIDE/CH2/RSTRAND0.PDF](http://www.project2061.org/publications/rsl/online/GUIDE/CH2/RSTRAND0.PDF).
234. **CITI Computing and Information Technology Interactive Digital Educational Library**, visualizes contents of digital library, [www.citidel.org](http://www.citidel.org).

## VII. PARALLEL COMPUTING

Parallel computing is the dominant form of high-performance computing. Even now there are multicore CPUs on desktops and laptops, with the immediate future seeing hundreds of thousands of CPUs with closely integrated communication and arithmetic units. Nevertheless, parallel computing is hard to implement for many scientific problems and now requires lower-level coding than used for serial programs. Most parallel programs utilize distributed memory computers with a message-passing interface known as MPI. A bright spot for the future of parallel computing is partitioned global address space (PGAS) languages, UPC for c and CAF for FORTRAN, and OpenMP (open multiprocessing), which should place less of a burden on the programmer.

235. **Argo Beowulf Cluster: MPI Commands and Examples** (Academic Computing and Communications Center, Univ. of Illinois at Chicago), [www.uic.edu/depts/acc/hardware/argo/mpi\\_routines.html](http://www.uic.edu/depts/acc/hardware/argo/mpi_routines.html).
236. **Web Pages for MPI and MPE**, (Argonne National Laboratory, Mathematics and Computer Science Division), [www-unix.mcs.anl.gov/mpi/www](http://www-unix.mcs.anl.gov/mpi/www).
237. **Parallel Computing Works!**, G. Fox, R. Williams, and P. Messina (Morgan Kaufmann, San Diego, 1994). Dated technology, but with valuable insights.
238. **PVM: Parallel Virtual Machine A User’s Guide and Tutorial for Networked Parallel Computing**, A. Geist, A. A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam (1994), (Oak Ridge National Laboratory, Oak Ridge), [/www.netlib.org/pvm3/book/pvm-book.html](http://www.netlib.org/pvm3/book/pvm-book.html).

239. **Parallel Programming with OpenMP** (Ohio Supercomputing Center, Science and Technology Support Group), [www.osc.edu/supercomputing/training/openmp/big/fslid.001.html](http://www.osc.edu/supercomputing/training/openmp/big/fslid.001.html).
240. **The OpenMP Application Program Interface**, a specification for parallel programming, [www.openmp.org](http://www.openmp.org).
241. **Parallel Programming in c with MPI and OpenMP**, M. J. Quinn (McGraw–Hill, New York, 2004). The computer science theory behind parallel programming, yet practical with many scientific examples. MPI codes in text.
242. **Parallel Programming with MPI**, P. S. Pacheco, (Morgan–Kaufmann, San Diego, 1997).
243. **Partitioned Global Address Space Languages**, *UPC, CAF OpenMP, UPC, CAF*, search the Web on acronym.
244. **Scientific Parallel Computing**, L. R. Scott, T. Clark, and B. Bagheri (Princeton U. P., Princeton, 2005). Focuses on the effect of algorithms, architecture, and languages on performance.
245. **How to Build a Beowulf**, T. Sterling, J. Salmon, D. Becker, and D. Savarese (MIT Press, Cambridge, 1999). Dated technology, but with valuable insights.
246. **Xgrid**, Apple technology for turning an ad hoc group of Macs into a cluster using MPI, [www.apple.com/server/macosx/technology/xgrid.html](http://www.apple.com/server/macosx/technology/xgrid.html).

## VIII. DIGITAL LIBRARIES

A major change in computing since CP-1 is the increased importance of networks, distributed datasets, CPU architectures designed for numerically intensive computing, and parallel processing. The networks permit us to get to distributed datasets and digital libraries quite easily, while the sophisticated architectures make the use of libraries containing subroutines designed for specific architectures crucial.

### A. Subroutine libraries

Although physicists often like to write their own software, I recommend the use of good software libraries as being faster and more reliable than the simple algorithms found in textbooks. These libraries typically contain the BLAS (basic linear algebra subprograms), LAPACK (linear algebra package), random-number generators, Fourier transforms, convolutions, correlation functions, sorting, searching, interpolation, and quadrature. Many of these libraries contain versions for specific architectures and parallel computers (SCALAPACK). In addition, there are also software libraries on specialized subjects such as finite elements, finite differences, signal processing, Fourier transforms, wavelet transforms, integral equations, and many of the other subjects you may find in a CP textbook; I suggest that you search the Web for specific needs.

247. **CERNLIB, Anaphe, and LCG at SLAC**, beware that these are somewhat outdated, [www.slac.stanford.edu/comp/physics/cernlib/cernlib.html](http://www.slac.stanford.edu/comp/physics/cernlib/cernlib.html). (A)
248. **CMLIB**, a free library of some 750 high-quality FORTRAN 77 subroutines for numerical computation. Maintained by the National Institute of Standards and Technology, which seems to have serviced it last in 1988, [gams.nist.gov/serve.cgi/Package/CMLIB](http://gams.nist.gov/serve.cgi/Package/CMLIB). (I)
249. **Computer Physics Communications Program Library**, started in 1969 as a pioneer in establishing the importance of programs in the development of CP. It now contains more than 2000 programs, with papers describing the programs published in the *Computer Physics Communications Journal*, [cpc.cs.qub.ac.uk/cpc](http://cpc.cs.qub.ac.uk/cpc). (A)
250. **FFTW, Fastest Fourier Transform in the West**, a C subroutine library for computing the discrete Fourier transform (DFT) in one or more dimensions for both real and complex data, [www.fftw.org](http://www.fftw.org). (A)
251. **NIST Guide to Available Math Software (GAMS)**, a cross-index and virtual repository of mathematical and statistical software components for use in computational science and engineering, [gams.nist.gov](http://gams.nist.gov). (I)
252. **GMP, the GNU Multiple Precision Arithmetic Library**, a free library for arbitrary precision arithmetic, [gmplib.org](http://gmplib.org). (I)
253. **Gnu Project, Scientific Library**, a leader in providing free software, including subroutine libraries and computational tools for UNIX/LINUX. The Gnu scientific library (GSL) for C and c++ is free, extensive, and open source, [www.gnu.org/software/gsl/](http://www.gnu.org/software/gsl/); manuals at [www.gnu.org/manual/manual.html](http://www.gnu.org/manual/manual.html). (E)
254. **IBM Engr. & Sci. Software Lib. (ESSL)** a complete, yet proprietary library optimized for IBM computers. Parallel ESSL (PESSL) supports 32-bit and 64-bit Fortran, C and c++ distributed-memory parallel-processing. Math Acceleration Subsystem provides scalar and vector libraries of tuned mathematical intrinsic functions, [www-03.ibm.com/systems/p/software/essl.html](http://www-03.ibm.com/systems/p/software/essl.html). (I)
255. **IMSL**, a commercial set of high-quality and high-performance mathematical and statistical functions written in C, c++, Java, and Fortran, [www.vni.com/products/imsl](http://www.vni.com/products/imsl). (A)
256. **JAMA, A Java matrix package**, part of the Java Numerics project at NIST, is a matrix library for Java (see too IMSL and NAG), [math.nist.gov/javanumerics/jama](http://math.nist.gov/javanumerics/jama). (I)
257. **Java Matrix Package, Jampack** A sibling matrix package to JAMA, developed at NIST and the Univ. of Maryland, [ftp://math.nist.gov/pub/Jampack/Jampack/AboutJampack.html](http://ftp://math.nist.gov/pub/Jampack/Jampack/AboutJampack.html). (I)
258. **JScience**, comprehensive Java library for the scientific community, even has complex matrix objects, which JAMA does not, [jscience.org](http://jscience.org).
259. **LAPACK, Linear Algebra Package**, an efficient, portable free library of FORTRAN 77 subroutines for common problems in linear algebra; a successor to LINPACK and EISPACK, [www.netlib.org/lapack](http://www.netlib.org/lapack). (I)
260. **LAPACK Users' Guide**, 3rd ed., E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen (SIAM, Philadelphia, 2000). (I)
261. **LAPACK95 Users' Guide**, V. A. Barker, L. S. Blackford, J. Dongarray, J. Du Croz, S. Hammarling, M. Marinova, J. Waśniewski, and P. Yalamov (Siam, Philadelphia, 2001). (I)
262. **Netlib** contains a variety of free libraries, including ones for vector and matrix algebra, sparse matrix algebra, fast Fourier transforms, and the famous BLAS and LAPACK, [www.netlib.org](http://www.netlib.org). (I)
263. **Numerical Algorithms Group (NAG)**, originally from UK labs, now a commercial set of high-performance mathematical and statistical functions for C, EXCEL, Java, LABVIEW, MAPLE, MATLAB, R, and VISUAL BASIC, [www.nag.co.uk](http://www.nag.co.uk); similar package, **Aspen HSL (formerly Harwell Subroutine Library)**, [www.aspentech.com/HSL](http://www.aspentech.com/HSL). (A)
264. **Open Source Physics**, a collection of curricular material for physics computation and physics education at all levels, [www.opensourcephysics.org](http://www.opensourcephysics.org). (I)
265. **Open Source Physics User's Guide with Examples**, W. Christian (Addison–Wesley/Pearson, Boston, 2006). (E, I)
266. **PORT (Portable, Outstanding, Reliable, and Tested) Mathematical Subroutine Library**, a free library of a wide range of FORTRAN 77 routines. Originated at Bell Labs research, [www.bell-labs.com/project/PORT](http://www.bell-labs.com/project/PORT). (I)
267. **Python Scientific Libraries, NumPy, and SciPy**. NumPy is the fundamental package for scientific computing, while SciPy extends it. NumPy contains N-D array object, linear algebra, Fourier transforms, random numbers, and tools for integrating FORTRAN code. SciPy provides numerical routines for things like numerical integration and optimization, [numpy.scipy.org](http://numpy.scipy.org), [www.scipy.org](http://www.scipy.org). (I, A)
268. **ScalLAPACK (Scalable LAPACK)**. Get program running on single processor with LAPACK, and then switch to SCALAPACK for parallel processing, [netlib2.cs.utk.edu/scalapack](http://netlib2.cs.utk.edu/scalapack). (A)
269. **ScalLAPACK Users' Guide**, L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley (SIAM, 1997, Philadelphia). (A)

### B. General Digital Library

270. **Assoc. for Computing Machinery (ACM) Digital Library** contains journals, magazines, transactions, proceedings, newsletters, publica-

tions by affiliated organizations, special interest groups, and ACM oral history interviews, [portal.acm.org/dl.cfm](http://portal.acm.org/dl.cfm).

271. **ACM Special Interest Group on Computer Science Education (SIGCSE)**, members' materials, [www.sigcse.org/topics/](http://www.sigcse.org/topics/).
272. **American Physical Society Journals**, just journals, but for more see *Compadre* portal (Ref. 269b), [publish.aps.org](http://publish.aps.org).
273. **Project Gutenberg**, first producer of free electronic books (ebooks), [www.gutenberg.org/wiki/Main\\_Page](http://www.gutenberg.org/wiki/Main_Page).
274. **IEEE Computer Society Digital Library**, access to 25 society magazines and transactions, and more than 1,700 conference proceedings, [www.computer.org/portal/site/csdl/index.jsp](http://www.computer.org/portal/site/csdl/index.jsp).
275. **arXiv.org**, open access to e-prints in Physics, Mathematics, Computer Science, Quantitative Biology and Statistics. Originally at Los Alamos, now at Cornell, [arxiv.org/](http://arxiv.org/).
276. **Merlot**, Multimedia Educational Resource for Online Teaching, [www.merlot.org/merlot](http://www.merlot.org/merlot).
277. **National Science Digital Library (NSDL)**, [nsdl.org](http://nsdl.org). The national online library for science, technology, engineering, and math. CP content still in need of growth and organization. Direct use of NSDL is less effective than using its pathways:
- (a) **CSERD, Computational Science Pathway**, provided by Shodor Education Foundation, [www.shodor.org/refdesk](http://www.shodor.org/refdesk). (E)
  - (b) **ComPADRE Physics and Astronomy Pathway**, a collaboration of AAPT, APS, AIP/SPS and AAS, [www.compadre.org/portal](http://www.compadre.org/portal), with plans to add a CP Library in 2008. (E)
  - (c) **Materials Science Pathway**, provided by MatDL at Kent State Univ. Contains simulations tools of interest, [matdl.org/repository](http://matdl.org/repository). (A)
278. **PhysNet, Physics Documents Worldwide**, lists of links to document sources, [physnet.uni-oldenburg.de/PhysNet/physdoc.html](http://physnet.uni-oldenburg.de/PhysNet/physdoc.html).
279. **Sourceforge**, the world's largest open source software development website is a priceless resource, [sourceforge.net/](http://sourceforge.net/).

## IX. GLOSSARY OF ACRONYMS

AAPT	American Association of Physics Teachers
ACM	Association for Computing Machinery
AIP	American Institute of Physics
BLAS	Basic Linear Algebra Subprograms
CERN	European Center for Nuclear Research
CiSE	Computing in Science & Engineering (journal)
CISE	Computer and Information Science and Engineering (NSF)
CS	Computer Science
COTS	Commercial off-the-shelf

CP	Computational Physics
CSE	Computational Science and Engineering
CPU	Central Processing Unit
DFT	Discrete Fourier Transform
GNU	GNU Software Foundation
HPC	High Performance Computing
IDE	Integrated Development Environment
IEEE	Institute Electrical and Electronic Engineers
KDE	K Desktop Environment (Linux)
LAPACK	Linear Algebra Package
MPI	Message passing Interface
NCAR	National Center Atmospheric Research
NCSA	National Center Supercomputer Applications
NIST	National Institute Science and Technology
NSDL	National Science Digital Library
NSF	National Science Foundation
OOP	Object-Oriented Programming
SDSC	San Diego Super Computer Center
SIAM	Society for Industrial & Applied Mathematics
SQL	Structured Query Language
W3C	World Wide Web Consortium
XML	Extensible Markup Language

*Note added in proof.* Colleagues have recommended some additions. In Matlab, Octave, and Mathcad: **Scilab**, an open source, free platform for numerical computation, similar to Matlab, that also contains Scicos, a graphical system modeler and simulator, [www.scilab.org](http://www.scilab.org). (E,I); **Modeling and Simulation in Scilab/Scicos**, S. L. Campbell, J.-P. Chancelier, and R. Nikoukhah (Springer, Berlin, 2006). A tutorial with an emphasis on the modeling and simulation tools. (E,I). In Java: **Java 2: The complete reference**, 7th ed., H. Schildt (Osborne, Worcester, 2006). An in-depth description of essentially all of Java's features, even the new ones (I,A).

## ACKNOWLEDGMENTS

The referees, Amy Bugg, Wolfgang Christian, Denis Donnelly, Kenneth W. Ford, Richard Gass, Charles Patterson, and David Winch have been exceptionally insightful and constructive in their reports. I thank them for making this a more valuable resource than the one I originally assembled, and for broadening a parochial view. Thanks also goes to Roger Stuewer for his excellent editing and patience.