DESCRIPTION OF
COMPUTATION AND PROBLEM SOLVING IN UNDERGRADUATE PHYSICS

David M. Cook

Department of Physics
Lawrence University
Box 599
Appleton, WI 54912

Voice: 920-832-6721 FAX: 920-832-6962

Email: david.m.cook@lawrence.edu

CPSUP website: http://www.lawrence.edu/dept/physics/ccli

31 March 2007

1 Preamble, History, and Objectives

As they pursue research in physics, physicists regularly find that skills in performing many different supporting tasks are crucial to success. At one point or another, for example, a practicing physicist will encounter the need to solve algebraic equations; solve ordinary differential equations; solve partial differential equations; evaluate integrals; find roots, eigenvalues, and eigenvectors; acquire and analyze data; graph functions of one, two, and three variables; graph experimental data; fit curves to data; manipulate images; prepare reports and papers; and probably several other tasks. Computers have an immense potential both to assist in the use of analytic methods and to support numerical methods when analytic methods fail. Physics educators are now almost universally agreed that undergraduate physics curricula must acquaint physics majors with computational approaches to these tasks alongside the traditional analytic approaches so that, as practicing physicists in the 21-st century, they are familiar in the context of physics problems with the capabilities of a spreadsheet, e.g., EXCEL; an array processing program, e.g. IDL or MATLAB; a computer algebra system, e.g. MAPLE, *Mathematica*, or MACSYMA; a graphical visualization tool, often available within both array processors and computer algebra systems; a standard scientific computer language, e.g., FORTRAN, C, or C++, particularly for creating driving programs to invoke publicly or commercially available subroutines; a program for data acquisition, e.g., LabView; and perhaps other more specialized tools. Further, to make effective use of these tools, we believe that the student must be acquainted with the main capabilities of at least one operating system, preferably UNIX or LINUX; be fluent in the use of a versatile text editor and of a program for creating drawings; be fluent in the use of a publishing package, e.g., LaTeX; and be fluent in the use of a presentation program, e.g., PowerPoint.

In a project started in the mid 1980's and supported since 1988 by three grants from the W. M. Keck Foundation, three grants from the National Science Foundation, and matching moneys and other funds from Lawrence University, we in the Department of Physics at Lawrence University have been developing curricular components that (1) support efforts to acquaint students with computational procedures and resources early enough so that they will be motivated and prepared to use these resources *on their own initiative* when circumstances warrant and so that later work need not be interrupted to deal with computational issues as an aside to its main purpose, and (2) provide students with both the background and the confidence to support informed reading of vendor manuals, which usually do a splendid job of listing capabilities exhaustively but typically burden the beginner with initially irrelevant refinements and fail to illustrate adequately how even the rudimentary capabilities can be combined to perform useful tasks. Over the years, a wide assortment of instructional materials aimed at helping us at Lawrence to fulfill these objectives has been drafted and tested and redrafted. The third of the NSF grants, which was awarded to Lawrence University in February of 2000, supported the writing of a book that brings together the materials developed at Lawrence in the previous decade and expands on those materials, a book whose primary objective is to introduce intermediate-level (i.e., sophomore) physics students to a selected spectrum of computational tools, help them learn enough of the tools' capabilities to know what the tools can do in application to problems in physics, and build their confidence both in using the tools and in reading vendor-supplied documentation. The book is titled *Computation and Problem Solving in Undergraduate Physics*, hereafter *CPSUP*; it has been completed and in use at Lawrence and elsewhere (see Section 5 on page 7) since mid 2003.

With these objectives in mind, *CPSUP* consciously focuses on helping students *get started*; it is not designed to be comprehensive or exhaustive, either in laying out the capabilities of any particular computational resource or in discussing numerical algorithms, nor does it—in the first edition, anyway—address *all* of the above identified tasks. Quite explicitly, for example, *CPSUP* does not address laboratory-related activities like data acquisition and analysis. It is also not a book about computational physics; it focuses on uses of computational tools but does so very explicitly in the context of problems in physics. Indeed, the sophomore course at Lawrence for which *CPSUP* is the text would not in any way replace a course in computational physics. Rather, the materials treated in *CPSUP* provide strong background for a subsequent, junior-senior level course in computational physics, which—we believe—would be substantially enhanced if students came to it already familiar with the resources on which this book concentrates. Indeed, we introduced exactly that second course at Lawrence in September, 2004 (and I am drafting text materials for that course).

2 Challenges Faced in Structuring *CPSUP*

One major difficulty in creating materials on computational topics emerges because different potential users favor different hardware platforms and different software packages. Especially in the computational arena, the variety of options and combinations is so great that

any single choice (or coordinated set of choices) is bound to limit the usefulness of the product to a small subset of all potential users. To address this difficulty, *CPSUP* is assembled from a wide assortment of components, some of which—the generic components—will be included in all versions and others of which—those specific to particular software packages—will be included only if the potential user requests them. Thus, the specific software and hardware discussed in *CPSUP* can be tailored to the spectrum of resources available at the instructor's site. Two versions may well differ in numerous respects. One may include the generic components and the components that discuss IDL, MAPLE, C (including numerical recipes), and LaTeX while another may include the generic components and the components that focus on MATLAB, *Mathematica*, and FORTRAN (including Numerical Recipes).

A second major difficulty in creating materials on computational topics exists because some aspects of local environments are unique to individual sites. Local rules of citizenship, the features and resources of the local operating system, policies governing the structuring of public directories, the assigning of accounts and passwords, and the scheduling of backups and after-hours access, licensing restrictions on proprietary software, the means to launch particular application programs, to compile user-written FORTRAN and/or C programs, and to access printers, and numerous other aspects are subject to considerable local variation. *CPSUP* does not constrain local options in these matters. Instead, its users must draft a site-specific supplement, called the *Local Guide*, to which individuals should refer for site-specific particulars. A LaTeX template for that guide, specifically the guide used at Lawrence, is available to users of *CPSUP*, but it will require editing to reflect local practices.

3 The Structure of *CPSUP*

The titles of the chapters in (the first edition of) *CPSUP* are listed in Table 1. Chapter 1 stands alone and provides background for the rest of the book. Chapters 2 and 3—only one of which would normally be present in any particular version—introduce specific array processors.[1] Chapters 4, 5, and 6—again only one of which would normally be present in any particular version—introduce specific computer algebra systems.[2] Chapter 7—which is optional (though its content is prerequisite for *some* of the sections in later chapters)—discusses structured programming, illustrating with examples in FORTRAN and/or C. Chapter 8 introduces the idea of a subroutine library, specifically the Numerical Recipes library. The remaining chapters address several important categories of computational processing (Chapters 9 and 10 on solving ordinary differential equations,[3] Chapter 11 on evaluating integrals, and Chapter 12 on finding roots).

Each of Chapters 9, 11, and 12 begins with a (generic) section in which several problems drawn from subareas of physics and using the computational technique on which

[1] OCTAVE is a shareware version of MATLAB, and future editions perhaps should consider including OCTAVE as an option.

[2] MACSYMA is no longer commercially available but a shareware version called MAXIMA is a possible substitute.

[3] LSODE is a component in the ODEPACK package of FORTRAN subroutines for solving a wide variety of ODEs and is a standard subroutine available at many supercomputing centers.

	Preface
	Acknowledgements
	Disclaimer and Testing Log
	Table of Contents
Chapter 1	Preliminaries
Chapter 2	Introduction to IDL
Chapter 3	Introduction to MATLAB
Chapter 4	Introduction to MACSYMA
Chapter 5	Introduction to MAPLE
Chapter 6	Introduction to *Mathematica*
Chapter 7	Introduction to Programming
Chapter 8	Introduction to Numerical Recipes
Chapter 9	Solving Ordinary Differential Equations
Chapter 10	Introduction to LSODE
Chapter 11	Evaluating Integrals
Chapter 12	Finding Roots
Appendix A	Introduction to LaTeX
Appendix B	Introduction to TGIF
Appendix Z	Contacting Software Vendors
	Index

Table 1: Chapter titles in *CPSUP*. A complete table ot contents can be accessed from links at the web site `http://www.lawrence.edu/dept/physics/ccli`.

the chapter focuses are laid out. Each of those chapters then continues with one or more (optional) sections in which some of the identified problems are addressed symbolically with whatever computer algebra systems are included in the version, a (generic) section on numerical approaches to the category of problem on which the chapter focuses, and several (optional) sections in which some of the problems laid out in the first section are addressed numerically with whatever array processors, computer algebra systems, and programming languages are included in the version. Each chapter concludes with numerous exercises using the techniques—both symbolic and numerical—of the chapter. The (optional) appendices introduce a publishing system (Appendix A on LaTeX) and a program for producing drawings (Appendix B on TGIF).

Table 2 indicates the options from which a potential user would choose in communicating which components were to be included in the version assembled for his or her use.[4] In time (and in subsequent editions), discussion of other packages and languages (OCTAVE, MAXIMA, MathCAD, REDUCE, DERIVE, JAVA, Word, ...) and other general computational techniques (statistical data analysis, linear and non-linear curve fitting, image processing, Monte Carlo techniques, finite difference and finite element approaches

[4]Since any one of the eleven options can be set to true of false independently of the others, thare are technically $2^{11} = 2048$ different versions of *CPSUP*. Only a *very* few of these versions, however, are at all sensible.

- ☐ IDL
- ☐ MATLAB
- ☐ MAPLE
- ☐ *Mathematica*
- ☐ MACSYMA
- ☐ Programming
  - ☐ FORTRAN. including Numerical Recipes
    - ☐ LSODE
  - ☐ C, including Numerical Recipes
- ☐ LaTeX
- ☐ TGIF

Table 2: This table indicates the available options for assembling a particular version of *CPSUP*. Items whose selection boxes are vertically aligned can be selected independently. Indenting of selection boxes conveys prerequisites. Thus, for example, selecting LSODE requires selection of FORTRAN, which in turn requires selection of Programming.

---

to partial differential equations, ...) may be added, either written by the author of these materials or—possibly—contributed by additional authors. Chapters on partial differential equations, MUDPACK (a package of FORTRAN subroutines for solving elliptic PDEs), and MARC/Mentat (a commercial package for applying finite element methods to PDEs) are being drafted and tested in a new course at Lawrence.

The order of presentation *in the book* does not compel any particular order of treatment *in a course or program of self-study.* To be sure, some later sections depend on some earlier sections, but the linkages are not particularly tight. In the Lawrence context, for example, we start with the chapters and appendices introducing IDL, MAPLE, and LaTeX, then address the IDL and MAPLE portions of the chapter on ordinary differential equations (ODEs), the FORTRAN portions of the chapter on programming, the FORTRAN portions of the chapter on ODEs, and the chapter introducing LSODE, and finally address the IDL, MAPLE, and FORTRAN portions of the chapters on integration and root finding.

Despite the organization of the chapters by program or by computational technique involved, the focus throughout is on physical contexts. The materials are designed to be used in conjunction with intermediate level courses, not introductory courses. While the illustrations of computational procedures highlight significant physical contexts and most of the examples and suggested exercises emerge from interesting physical situations, the objective is for students to become both fluent and wary in using computational resources in application to these physical situations, not to dwell excessively on the microscopic details of numerical analysis or to teach them the underlying physics (except insofar as successful computer-based solution of problems underscores the power of the fundamental physical ideas). The students are assumed to have completed an introductory survey course and to be embarking on intermediate level studies as they undertake a study of this book. We focus not so much on the set up of the situations—that is assumed to be the province of

other courses—as on computer-based symbolic and numerical techniques and strategies for determining the solution once the set up is complete. Examples are drawn from classical mechanics, classical electricity and magnetism, thermal physics, quantum mechanics, curve fitting, DC and AC circuit theory, optics, and several other areas.

4 Potential Uses

At Lawrence, those portions of *CPSUP* relating to software packages we support have been used for more than a decade and a half as the text to help sophomores learn the capabilities of several tools. From the early 1990s until the 2002–03 academic year, I offered an elective sophomore course titled *Computational Tools in Physics*. In outline, the course introduced in order (1) LaTeX (to be used subsequently for *all* submitted work), (2) the general capabilities of IDL and MAPLE, (3) the use of IDL, MAPLE, and LSODE (which provides a vehicle for introducing FORTRAN) for solving ODEs in conjunction with a concurrently taken course in classical mechanics, (4) the use of IDL, MAPLE, and Numerical Recipes for evaluating integrals in conjunction with a concurrently taken course in electromagnetic theory, and (5) the use of IDL, MAPLE, and Numerical Recipes for finding roots, with examples drawn from quantum eigenvalue problems and vibration analysis. The course met formally once every two weeks throughout the entire academic year for a lecture/demonstration orienting students to the next task. Thereafter, students worked their way through the tutorial portions of the text and then worked three or four substantial assigned exercises. Both the segment on ODEs and the segment on integration ended with each student undertaking a three to four week project on a computational exercise of his or her choice. Even though this course is no longer offered, detailed syllabi for a past offering of this course are retained on the Lawrence website.[5]

Starting in 2002–03 (and reflecting a desire to move computational topics into a *required* part of our program for physics majors), we abandoned the former elective course and restructured our required sophomore course in classical mechanics to focus about 50% of its time on computational tools, creating a new course called *Computational Mechanics*. *CPSUP* is, of course, the text for the computational dimensions of this new course. In 2004–05, when for the first time both junior and senior majors had completed *Computational Mechanics*, we introduced a junior/senior course titled *Computational Physics*, which lists *Computational Mechanics* as a prerequisite, focuses on approaches to partial differential equations, and uses as its text the chapters on partial differential equations, MUDPACK, and MARC/Mentat now being drafted for the projected second edition of *CPSUP*. Syllabi for these two courses are also available on the Lawrence website.[6]

As illustrated by our uses at Lawrence, *CPSUP* is a suitable text for a sophomore course aimed at introducing physics majors at that level to computational approaches to

[5] Go to `http://www.lawrence.edu/dept/physics/curriculum/listing.html`, scroll down to Physics 270, and click on the link for the syllabus to that course. Note that, at the time of the posted syllabi, Physics 270 used MACSYMA rather than MAPLE as the symbolic manipulator.

[6] Go to `http://www.lawrence.edu/dept/physics/curriculum/listing.html`, scroll down to Physics 225 for *Computational Mechanics* or to Physics 540 for *Computational Physics*, and click on the link for the syllabus at either location.

problems in physics. Through selection of sections and choice of the examples in mechanics or in electromagnetic theory or ..., the text would also function well as a supplement to support introduction of computational approaches in the standard courses in the intermediate-level program of a typical physics major. Since it is in many places structured as a tutorial, *CPSUP* could also support self-studies, independent studies, or tutorials by individual students interested in learning how to apply computation to physics—and occasional students at Lawrence have used the materials in that mode. Finally, especially after a student has spent some time with some of the chapters, *CPSUP* should be a useful reference work to have in one's library.

Broadly, *CPSUP* is a flexible text that can be used in a variety of ways—including as the text for stand-alone courses, as a supplement to existing courses, and as the text for tutorials and self-study—to support introduction of computational approaches to problems in physics alongside the more traditional analytic approaches. The author sincerely hopes that the experience at Lawrence over the past two decades has produced a text that will be valuable in many contexts as increasing numbers of departments strive to incorporate computation into their upper-level physics curricula.

5 Status of *CPSUP* and Publishing Requirements

The drafting of (the first edition of) *CPSUP* was complete by 1 August 2003, and all necessary permissions were by then in hand. During the summers of 2001, 2002, and 2003, four week-long workshops supported by an NSF grant and attended by a total of 70 physics faculty members from around the United States were held at Lawrence. Several of the faculty members who participated in those workshops beta-tested at least portions of various drafts of *CPSUP* and have provided feedback that helped the author to refine the exposition. Early in the project, the author explored publication with several possible publishers and even signed a contract with one publisher. As it turned out, the capacity of that publisher to provide the level of customizability required for relatively small orders economically was not up to what *CPSUP* requires, and that original contract was terminated (amicably on both sides). Since that termination, the author has been publishing *CPSUP* on his own. Table 3 summarizes the distribution and use of copies since the drafting was complete in August 2003.

Over the longer term, the author would like, however, to interest a commercial publisher in *CPSUP*. The product requires a capacity to produce relatively small numbers of copies of customized assemblages of the generic components with a user-specified selection of the optional components. Unfortunately (and this is the major requirement that the publishing procedures of two or three years ago could not meet), the customization is quite microscopic. In only some cases does the customization amount to omitting or including entire chapters. If, for example, the MAPLE components are not selected, then the MAPLE chapter would be omitted but also the MAPLE *sections* in the chapters on ordinary differential equations, integration, and root finding would be omitted. The construction of the customized book itself as well as of the table of contents and index must respect omissions not only of occasional entire chapters but also of sections within chapters.

Version	# Institutions	# copies
Class-sized Adoptions		
Maverick	6	53
IDL-MAPLE+other components	5	38
IDL-*Mathematica*+other components	5	92
MATLAB-MAPLE+other components	6	47
MATLAB-*Mathematica*+other components	11	106
TOTAL	33	336
Single Examination Copies Sent		
Full	10	10
IDL-MAPLE+other components	8	8
IDL-*Mathematica*+other components	11	11
MATLAB-MAPLE+other components	16	16
MATLAB-*Mathematica*+other components	31	31
TOTAL	76	76

Table 3: Summary of distribution of self-published copies of *CPSUP* from 1 August 2003 to 31 March 2007. Versions contain IDL-MAPLE, IDL-*Mathematica*, MATLAB-MAPLE, or MATLAB-*Mathematica* and various selections from the other options. The few versions containing more than one of IDL and MATLAB and/or more than one of MACSYMA, MAPLE, and *Mathematica* are identified as "maverick". In the column labeled '# institutions', institutions to which copies were sent in more than one year are tallied once for each year. Beyond the above-enumerated distribution, seventy copies of the full version (as it existed at the time of each workshop) are in the hands of those who participated in the workshops, and four different institutions have purchased a total of seven copies, mostly of the full version, as library acquisitions.

Thus, *the publication of CPSUP will require a process that can fairly easily assemble the desired version for each user from the appropriate files. Further, since most users are likely to be at liberal arts colleges, the process must be economical even for relatively small orders, since the typical order is likely to be for somewhere between 5 and 20 copies.*

In the author's self-publication of *CPSUP*, this requirement is met as follows: LaTeX is used as the type-setting package. All text is incorporated in the first instance in LaTeX source files, which include not only the text but embedded formatting instructions and indexing commands. All figures are in PostScript and are included in the text with the LaTeX `\includegraphics` command. Generic components are included automatically when the controlling file `assemble.tex` is processed by LaTeX. Inclusion of optional components is controlled by Boolean flags that are set to `true` or `false` in a file that is almost the first thing to be read when `assemble.tex` is processed by LaTeX. To create a PostScript file of a desired version involves

- Creating a file of the proper `true-false` flags.[7]
- Running LaTeX on `assemble.tex`, specifying the name of the "flag" file.
- Running LaTeX on `assemble.tex` a second time to set internal references.
- (sometimes) Running LaTeX on `assemble.tex` a third time to account for pagination that changes when correct references are inserted.
- Running the program `makeindex` to format the index.
- Running LaTeX on `assemble.tex` one more time to include the proper index.
- Running `dvips` to create the PostScript file ready for printing.

The entire process takes about five minutes on the HP LINUX workstation in the author's office. In the self-publishing operation, the PostScript file is then transmitted electronically to a local printing business which provides it as input to a Xerox machine that prints and binds the desired number of copies, almost without further human intervention.[8] The author knows not the feasibility of this particular approach for commercial publishing, but it or something equivalent is essential for *CPSUP*.

6 Additional Information

Additional information about this book can be obtained by perusing the project web site identified on page 1, reading a review that appeared recently,[9] or contacting the author at the email address or the phone number on page 1.

[7] Probably someone more skilled than the author could create a web-based order form that would create the necessary "flag" file automatically from selections made by the end-user when placing an order.

[8] The author even imagines that the entire operation could be automated so that the user would submit an order on line and the bound copies would appear at the output of the appropriate machine without any attention from a person along the way.

[9] See R. Thorsten Clay, Review of *A First Course in Computational Physics and Object-Oriented Programming with C++* by David Yevick, *A First Course in Scientific Computing: Symbolic, Graphic, and Numeric Modeling* by Rubin H. Landau, and *Computation and Problem solving in Undergraduate Physics* by David M. Cook (Am. J. Phys. **74** 653 (July, 2006)). The three books reviewed are quite different from one another but are aimed more or less at the same audienceand have similar broad objectives.