DESCRIPTION OF THE SECOND EDITION OF *COMPUTATION AND PROBLEM SOLVING IN UNDERGRADUATE PHYSICS*

David M. Cook

Department of Physics
Lawrence University
711 E Boldt Way SPC 24
Appleton, WI 54911

Email: `david.m.cook@lawrence.edu`

22 February 2023

Note: More detailed information about this freely-downloadable book and associated files are available from the Physical Sciences Resource Center of the AAPT ComPADRE Digital Library `psrc.aapt.org/psrc/curricula/cpsup/`, a library that is part of the National Science Digital Library.

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 address 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, MATLAB, OCTAVE, or PYTHON; a computer algebra system, e.g. MAPLE, MATHEMATICA, or MAXIMA;[1] a graphical visualization tool, often available within both array processors and computer algebra systems; a standard scientific computer language, e.g., PYTHON, 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, e.g., UNIX or LINUX, Windows, or macOS; 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.

[1]IDL, MATLAB, MAPLE, MATHEMATICA, and MACSYMA were included in the first edition; in the second edition, MACSYMA was replaced with MAXIMA, and OCTAVE and PYTHON were added.

In a project started in the mid 1980's and supported between 1988 and 2003 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 studies need not be interrupted to deal with computational issues as an aside to their 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*; its first edition was completed in 2003 and has been in use at Lawrence and elsewhere (see Section 5 on page 7) since mid 2003. It has been continually revised and updated since 2003 and, in 2018, an updated and expanded second edition was made available. In early 2023, the second edition was expanded with the addition of a chapter on partial differential equations (PDEs) and a chapter on the large FORTRAN package MUDPACK, which contains numerous routines for applying multigrid techniques to the solutions of PDEs.

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 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 text materials for that course came into being in that year, have been steadily revised since, and are now included in the files currently available from the site identified in the note at the beginning of this description.

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 PYTHON and MATHEMATICA,

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, can be downloaded from the above-identified URL, but it will require editing to reflect local practices.

3 The Structure of *CPSUP*

The titles of the chapters in the second edition of *CPSUP* are listed in Table 1. Chapter 1 stands alone and provides background for the rest of the book. Chapters 2, 3, 4, and 5—only one of which would normally be present in any particular version—introduce specific array processors (IDL, MATLAB, OCTAVE, and PYTHON). Chapters 6, 7, and 8—again only one of which would normally be present in any particular version—introduce specific computer algebra systems (MAXIMA, MAPLE, and MATHEMATICA). Chapter 9, portions of which are prerequisite for *some* of the sections in later chapters, discusses structured programming, illustrating with examples in FORTRAN, C, IDL, MATLAB, OCTAVE, and PYTHON. Chapter 10 introduces the idea of a subroutine library, specifically the Numerical Recipes library. The remaining chapters address several important categories of computational processing. Chapters 11 and 12 focus on solving ordinary differential equations,[2] Chapter 13 discusses evaluating integrals, Chapter 14 treats finding roots, Chapter 15 fo-

[2]Introduced in Chapter 12, 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.

Table 1: Chapter titles in the second edition of *CPSUP*. A complete table of contents can be accessed from the URL identified in the note at the beginning of this description.

cuses on partial differential equations (including both finite difference and finite element methods), and Chapter 16 focuses on the MUDPACK package of FORTRAN subroutines for solving elliptic partial differential equations by multigrid techniques.[3]

Each of Chapters 11, 13, and 14 begins with a (generic) section in which several problems drawn from subareas of physics and using the computational technique on which 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. Chapter 15 also begins with a generic section outlining several problems whose solution involves solving partial differential equations and continues with numerous (optional) sections addressing those problems numerically using

[3]Like ODEPACK, MUDPACK is a standard subroutine package available at many supercomputing centers.

several of the numeric tools introduced in earlier chapters and exploiting both finite difference and finite element techniques. Chapter 12 uses LSODE to solve seven different problems involving systems of ordinary differential equations, some of which emerge after the basic equation has been discretized in the spatial dimensions. Chapter 16 uses components of MUDPACK to solve several different boundary value problems in two and three dimensions. The appendices introduce a publishing system (Appendix A on LaTeX) and a UNIX program for producing drawings (Appendix B on TGIF). Every chapter concludes with numerous exercises using the techniques of the chapter.

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.[4]

Despite the organization of the chapters by program or by computational technique, 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 (IDL, MATLAB, MAPLE, MATHEMATICA) have been used continuously since the early 1990s as the text to help sophomores learn the capabilities of several tools.[5] 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)

[4] In recent offerings, PYTHON and MATHEMATICA have replaced IDL and MAPLE.

[5] The PYTHON components were first used in the winter of 2019.

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.

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 and MUDPACK, chapters that were first drafted in the early 2000s and have now been added to those available to the world beyond Lawrence. *Computational Mechanics* has been offered annually at Lawrence since 2003, though its content has been continually revised as different teachers have assumed responsibility for the course. *Computational Physics* has been offered in alternate years since 2004, again with revisions introduced by different teachers.

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 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 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 since the early 1990s 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 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). Up until about 2020, the author published *CPSUP* on his own. During that time, some 1000 copies of various versions were provided for class use at a variety of institutions and, beyond the 70 copies given to the participants in the summer workshops, over 100 examination copies were sent to interested teachers around the country.

As the author approaches the end of his professional career, continued self-publishing has seemed increasingly less viable. To address that concern, these materials—and many others created at Lawrence over the years to embed a focus particularly on using computational resources in the undergraduate physics curriculum—have now been made available for free download from the Physical Sciences Resource Center of the AAPT ComPADRE Digital Library, a library that is part of the National Science Digital Library. For the time being, only completed texts will be placed at the URL `psrc.aapt.org/curricula/cpsup`.

All versions will contain Chapter 1 and the generic components of Chapters 9, 11, 13, 14 and 15. Components for a specific tool will be included only if the corresponding flag from those listed in Table 2 is set to TRUE when the version is created. Since there are 14 flags that can be independently set to one of two values (TRUE or FALSE), there are technically $2^{14} = 16384$ distinct versions. If all flags are set to TRUE, the result is a version containing 1871 pages and occupying a file either 34.2 MB or 39.2 MB in size![6] If all flags are set to FALSE, the resulting version contains 204 pages and occupies a file either 1.2 MB or 1.9 MB in size. In my mind, a version that includes only one of IDL, MATLAB, OCTAVE, and PYTHON, only one of MAXIMA, MAPLE, and MATHEMATICA, and LaTeX (because I have a particular fondness for that tool) is likely to be most useful; there are 12 such versions (24 if both linked and print versions are counted separately). Those 24 versions are provided for download from the URL identified in the note at the beginning of this description. Beyond those 24 versions, the available downloads include (a) a version with *all* components *except* the FORTRAN and C components, (b) a version containing the generic components and the FORTRAN components (including Numerical Recipes), LSODE, and MUDPACK), (c) a version containing the generic components and the C components (including Numerical Recipes), (d) a version containing *only* TGIF, and (e) a version containing only LaTeX. Separate subsections at the indicated URL provide for

[6] In the larger file, which is convenient for viewing on a screen, each internal reference is linked to the point referenced; in the smaller file, which is better if the file is to be printed, the references are present but are not linked to the point referenced.

Table 2: This table lists the available flags for assembling a particular version of *CPSUP*. For the most part, each of these options can separately be set to TRUE or FALSE, except that FORTRAN is prerequisite for LSODE and MUDPACK, and at least one of FORTRAN and C is prerequisite for NUMERICAL RECIPES.

IDL	MATLAB	OCTAVE	PYTHON
MAXIMA	MAPLE	MATHEMATICA	
FORTRAN	C	NUMERICAL RECIPES	
LSODE	MUDPACK		
LATEX	TGIF		

download of (a) solutions to selected exercises from *CPSUP* and (b) all program and data files referred to in the text..

In due time, the LaTeX source files for all components, PDF files for all figures not produced by `tikz`, and instructions for adding further components and producing additional versions may be added. At base (and very briefly), the process to create a PDF file containing the formatted text of a desired assemblage, including a Table of Contents and an Index, involves

- Creating a "flag" file of the proper `true-false` flags to select the desired components, and
- Processing the controlling file `assemble.tex` at least twice and possibly three times with `pdflatex`, each time specifying the name of the "flag" file and stipulating whether a linked or printable version is to be created.

The entire process takes about five minutes on the HP Windows computer the author now uses. The result is a file containing the desired components ready for viewing on a screen or printing.

6 Additional Information

Additional information about this book can be obtained by perusing the ComPADRE web site identified on page 1; reading a published review;[7] looking up the article "Computation in Undergraduate Physics: The Lawrence Approach", Am. J. Phys. **76**, 321–326 (April–May, 2008);[8] and/or, for the time being, contacting the author at the email address on page 1.

[7]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 audience and have similar broad objectives.

[8]This theme issue of the American Journal of Physics was published in conjunction with a Gordon Conference held 8–13 June 2008 on the topic "Computation and Computer-Based Instruction."