Book Editors: Jovanovic, J. and Chiong, R. (2012).
Technology Enhanced Learning
Santa Rosa, California: Informing Science Press. (pp. xx-xx).

Chapter 1

An eTextBook in Computational Physics with Multiple Executable Elements

Rubin H. Landau, Oregon State University, Corvallis
Manuel J. Paez, University of Antioquia, Medellin
Cristian Bordeianu, University of Bucharest

Work supported in part by the National Science Foundation via Grant CCLI-0836971 and via the Engaging People in Cyberinfrastructure project.

Abstract

A complete eTextBook with multiple executable elements has been created, and Web-based versions have been placed in repositories affiliated with the US National Science Digital Library. The book, which is aimed at upper-division undergraduates, contains files format and executable elements chosen to be platform independent, highly useable and free. While future technologies and operating systems promise vastly improved executable books, the created eTextbook highlights some of the features possible with existing technologies. The project combines some 20 years of Computational Physics textbook developments and 15 years of Web enhancement developments into a prototype eTextBook that includes text, computational laboratories, demonstrations and video-based lecture modules.

Introduction, a Short History of Textbooks

The Romans, Chinese, Egyptians, Jews, Indians, and other ancient cultures all had texts, with Paleo-Hebrew being used by the Sumerians for texts in the 10th century BCE or earlier. Some texts, like Homer's Odyssey from the 8th century BCE, were spoken, with the change to printed texts did not coming easy; Socrate' lament that the change in media from spoken word to printed text has led to a decrease in students' ability to memorize stories was recorded by his student Plato. By the 4th century CE, Aelius Donatus had developed a form of the textbook that is still popular today, namely, a question and answer *catechism*. The printing of books in quantity was made possible by Gutenberg's invention of moveable type around 1439, with the next hundred years seeing books becoming more affordable and common. It may be said that by the 17th century, Descarte already set the foundation of what we now call *virtual reality* when he clarified the distinction between the concepts of objective reality (what's really there) and subjective reality (what we perceive when we read and imagine).

True textbooks probably started with the New England Primer and Noah Webster's Grammar, which provided a streamlined approach to education by combining grammar rules with moral maxims. Modern textbooks in the USA probably began in the 19th century as the newly found normal schools and land-grant universities started turning out large numbers of teachers. This not only led to the end of the minimal classrooms with one teacher speaking and students at multiple grade leveles memorizing facts, but also changed the style of teaching. These modern educators were taught objective teaching in which students developed understanding via experiences and the teachers taught by example, as is still common.

Electronic versions of textbooks may have started in the 1960s with programmed learning, which is essentially the Q&A catechism in modern form. In 1971, Michael Hart founded Project Gutenberg (2012) and entered its first eBook, the US Declaration of Independence. The popularity of eBooks rose in the 1990s as the developing internet made file transfers easier, as new formats for books were developed, and as pirate sites started contaminating the internet.

The internet also increased the popularity of Paul Ginsparg's ePrint arXiv for scientific preprints based at Los Alamos National Laboratory. Previous to this, preprints were mailed out before publications with a distribution that by its very nature was inequitable. The Web archive of preprints and anti-preprints (ones that have been published) led to a transformation in the world of scientific publication away from the traditional paper form for journals and towards self publishing.

By 1998 we already had libraries holding eBooks, although it was not until 2003 that they started lending them. In 2001 the Creative Commons (2012) licensing concept was created to provide "a flexible range of protections and freedoms for authors, artists, and educators". And finally, in 2010, Amazon sold more eBooks than paper ones.

Motivations

If I asked people what they wanted, they would have said "faster horses." -Henry Ford

Our Computational Physics Textbooks

The overall motivation for our educational developments has been to stimulate a systemic change in college-level curricula in which science and computation are better integrated. We believe this provides better preparation of students and improved learning. Our prime motivation for creating textbooks in Computational Physics (Landau et al., 2011, 2008, 2007) is to advance the integration of modern computational techniques into undergraduate physics courses. And because faculty members often do not feel knowledgeable enough to teach a multidisciplinary subject like CP, we want our textbooks to provide enough materials so that faculty with various levels of experience in scientific computing can teach courses incorporating computation.

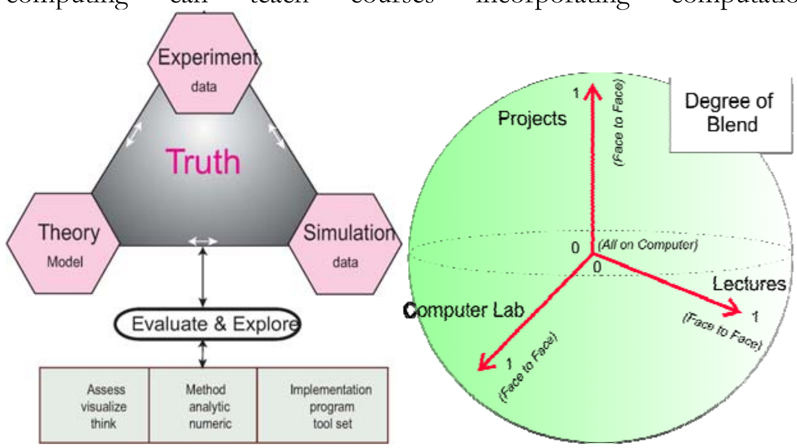


Figure 1: Left: The scientific problem solving paradigm used to structure the CP materials. Right: A 3-D space of Projects, Lectures and Lab axes, in which the degree of blend varies from 0 (all on computer) to 1 (all face to face)

Figure 1 left illustrates the scientific problem-solving paradigm that is at the core of computational research and that is used to structure our books. We use it within a projects approach in which we encourage students to be creative as they work on a series of projects. In this way students get actively engaged and emotionally involved with the materials as they experience the excitement of personal research. They also tend to take pride in their work, get familiar with a large number of techniques and tools, acquire confidence in making different project pieces work together, and learn valuable job skills.

From a pedagogical perspective, education following the problem-solving paradigm is a more balanced, efficient and relevant approach than often found in a standard physics curriculum. Although students may have to take fewer physics classes in order to make room for computation, they tend to learn the physics, CS, and math better when placed in context (Yasar, 2001). There is increased efficiency and effectiveness when materials are learned in context and with purpose, in this case to solve a problem.

Motivation for Electronic Text

Our practical motivation for an eTextBook came in 2007 as we completed our paper-based text *A Survey of Computational Physics,* published by Princeton University Press (Landau et al*.,* 2008; Princeton, 2012). That text, which already had many electronic enhancements on an accompanying CD, included programs listings in Java, with other language versions on the CD. We realized then that the Python language had become very popular, and particularly for physics education, and so approached our editor with the suggestion of another version of the text using Python. After we had some of the economic realities of non-profit publishing explained to us, we realized that an eTextBook was the best way to proceed with a separate Python edition. Although the business model for electronic publishing was even less clear then than it is now, having a free online version of the eTextBook seemed acceptable with PUP, especially in light of their interests advancing publishing and education.

Our realization of the great potential of an electronic text began much earlier, in November 1994, in an early meeting of the Undergraduate Computational Engineering and Sciences (2012) program. Then, individuals interested in computational science education (including Landau) started to share ideas and place HTML versions of their developments on the UCES computer, which resided on this new thing called the World Wide Web. Although it was clear from the start that HTML had problems with mathematics that LaTeX did not, it was also clear that the "hyper" part of hypertext meant that a text could use multimedia to enhance education. (Mathematics on the Web is still an issue, as we discuss later.) Specifically, the use of animations, sonification of data, simulations, and computer visualizations seemed to be ideal elements to enhance education.

Another of our motivations for creating an electronic text was the realization that different learners learn in different ways, and that it is essential for learners to form mental models in order to understand abstract subjects such as physics (Carroll & Olson, 1988; Dede et al, 1999; Farah et al., 1988; National Research Council, 1991). Accordingly, the ability to form these mental models is improved by using multiple senses to interact with the materials since providing more ways to view our materials permits more parts of the reader's brain to get activated in the "viewings, and applying more resources to the task results in better learning (Salzman et al., 1999).

Finally an eTextbook makes good sense when learning a computational science. It has been shown that interacting with a computer and a textbook in a trial-and-error mode is a better way to learn how to compute, how to think reflectively, and how to write, than are lectures and readings (Davis, 1999; Hazzan & Tomayko, 2005; National Science Foundation, 1996; Larkin, 1993). Accordingly, having an eTextBook that can integrate the simulation, multimedia and networking capabilities of a computer seem natural for a textbook in a computational science..

Table 1: The applets incorporated into the text. In the eTextBook, each entry is a link to the applet. A Web page with the same table and active links is at http://science.oregonstate.edu/eBookWorking/Applets/.

Chaotic Pendulum	4 Centers Chaotic Scattering	Planetary Orbits
Waves on a String	Normal Modes	Cellular Automata Sierpinski
Solitons	Spline Interpolation	Relativistic Scattering
Lagrange Interpolation	Young's 2 Slit Interference	Wavelet Compression
Starbrite (H-R Diagram)	Data Sonification	Photoelectric Effect
Sonifying Physics	Lissajous Figures	Heat Equation 1, 2
Wavepacket-Wavepacket Collision Movies	Fractals: Sierpin, Fern, Tree, Film, Column, Dla	Wave Function SqWell, HarmOs, Asym, PotBarrier
Molecular Dynamics	2 Centers Satellite	Feynman Path Integral

In 1996 we started a Computational Physics Education Research group at Oregon State University as part of our involvement with the Northwest Alliance for Computational Science and Engineering (Northwest Alliance for Computational Science and Engineering). We developed HTML tutorials in various forms for the Web, and even converted a Unix book to a Web tutorial with a built-in Unix shell (Coping with Unix, 2012) that was used at supercomputer centers throughout the US. Later when Java was introduced, we developed an extensive collection of Applets (Applets, 2012), which are small Java programs that get loaded from a browser, but run on the user's computer. In Table 1 are the applets that are now part of our eTextBook. In 1997 we published a Computational Physics textbook and placed these various electronic enhancements to the text on the Web (Landau et al., 1998).

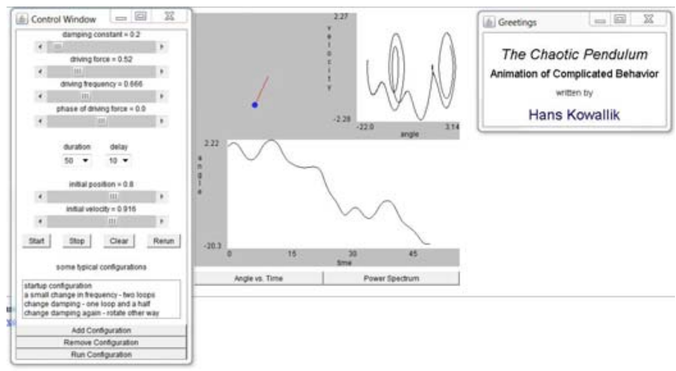


Figure 2: A screen shot of a chaotic pendulum applet found at http://science.oregonstate.edu/eBookWorking/Applets/JAVA_pend/

As an example of our applets, in Figure 2 we present a screenshot of one of our applets on the chaotic pendulum. The control panel on the left lets the student interact with the simulation and use it as a test for their own program.

This applet is a good demonstration of how multimedia can be used to make learning abstract concepts easier (Hazzan & Tomayko, 2005). In this case the abstract concept is phase space, a space in which position is viewed as the independent variable and velocity or momentum is viewed as the dependent variable (the plot with ellipse-like curves). On the one hand, the picture of a pendulum actually swinging is a concrete example of chaotic motion with its combination of oscillations with multiple periods and "over the top" rotations. Seeing this motion occur as the displacement θ versus time graph on the bottom is being drawn, breathes life into what otherwise appeared like just a graph. Likewise, clicking on the *Power Spectrum* button provides, literally, another (virtual) space in which to view the motion, one in which the ellipses of different sizes ("cycles") correlate with major Fourier components in the power spectrum. Seeing the phase space plot being drawn along with the θ versus *t* plot, the Fourier spectrum and the animation adds meaning at different levels of understanding to what is otherwise the abstract concept of motion in phase space. And because modern classical dynamics is greatly simplified by viewing it as geometry in phase space, these visualizations help the student make that transition to a simpler, yet more abstract, viewpoint.

Blended Courses

Over the last decade we have heard varied and often strong opinions regarding the efficacy of online versus traditional university classes. Regardless of the opinions, each passing year continues to see an increase in the fraction of online enrollment [17% in 2009 (Sloan, 2009)]. Over a decade of literature has shown that measurable learner outcomes have more to do with the quality of an entire course and the quality of its delivery than with the specific delivery media used (Twigg, 2003; Ramage, 2001). A common finding is that online courses are capable of working as well as traditional ones, and that *blended* or *hybrid* courses may be superior (Picciano, 2006, 2002, 1998; Pew, 2011). In the latter, faculty blend the course components, with each containing variable degrees of face-to-face, or face-to-computer learning (Figure 1 right).

The ability to vary the degree of blend to fit the local educational environment and students appears to significantly improve the effectiveness of a course. Likewise, having an eTextBook that contained video-based lecture modules, laboratory exercises, problems and in-depth reading would mean that it can be used as the core of an online course, in a blended course, or for a standard course. eTextBooks are thus more flexible than printed ones, but they do take more work to create, as does an online course as compared to a traditional one. The amount of work would be reduced if creation of the eTextBook, or at least its major elements like video lectures and slides, were incorporated into the development of a course.

There are multiple variables coming into play when comparing online to offline and blended courses, as well as when comparing traditional and fully electronic textbooks. In either case, isolating one element may not be reliable. Problems in online courses do arise with less-mature and less-motivated students who view the computer as a device for entertainment and socializing, and who lack the self-discipline needed for sustained effort on a regular basis without a formal class structure. However, this is less of an issue with students already specializing in computational science who tend to view the computer professionally.

Features of Our eTextBook

We have worked at assembling a model of what a future eTextBook might be like, while still using currently available, free, and common technologies. The entire 500 pages of it (700 in paper with its larger margins) is now online in a number of repositories (Landau et al., 2011) associated with the US National Science Digital Library (1996). We have adopted a Creative Common license for our eBook as our way of supporting the idea that materials developed with public funding should be available to the public. However, we are also working with the publisher to develop a business model in which there are fees for extra features such as a printed copy of the text, a DVD with multi-language codes or lecture modules, source codes for the slides, mobile device versions of the text and lectures, and an instructor's guide. [The LaTeX source files are available for readers with sight disabilities who can use software readers (Raman, 1994) to read even the equations.]

Our original plan with the eBook was to have the words, equations and figures be executable in multiple ways, but still have a format that is readable on a computer and that can be printed out to look like a real book. We wanted the package to provide multiple "views"of Computational Physics as well as contain most everything needed for a full, blended course. Ideally, we wanted the book's executable elements to be encapsulated into the text so that readers could do their explorations without leaving the text. This would include the ability to link to and execute archives of codes, data, applets, tutorials and scripts. We see no reason why readers shouldn't be able to reproduce a good fraction of the results and figures shown in books and scientific papers.

In order to implement our original plan, we wanted to create the book as a MathML/XML file (W3C Math Home; Extensible Markup Language) with scalar vector graphics figures (SVG). This would permit a high level of accessibility and executability, the ability to incorporate various Web technologies, as well as be viewable on a variety of devices. However, by the time we completed about one-third of our video lecture modules (the most time-intensive part of the project), it became clear that the neither the general world, nor the scientific world, nor the browser creators had adopted MathML to the point where we could expect many readers to use it. Accordingly, we changed our plan and decided to present our eTextBook as a pdf file with internal and external links (Landau et al., 2011). There are many advantages to the pdf format:

● It is readable on most every computer platform (universal) and free readers are available.

● It is readable on tablets and mobile devices.

● We can retain our LaTeX source file for the book and just output it as pdf.

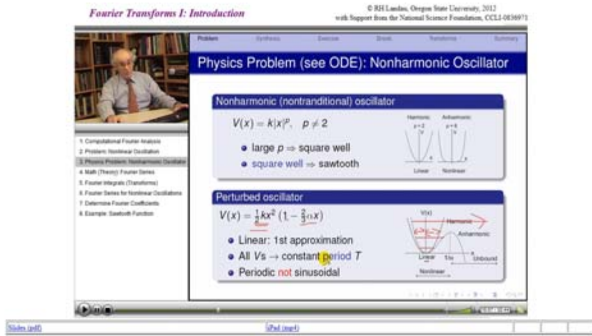
● The *hyperref* macro package (LaTeX/Hyperlinks) for LaTeX automatically inserts internal and external links into the pdf. It also automatically creates links to all figures, equations, tables, sections, references, and index and table-of-contents entries. This is important since hand setting links into pdf after each compilation would be time consuming.

● Most people know how to use pdf readers and their features.

● Acrobat (also called Adobe Reader) has extensive controls for navigation and viewing.

● Acrobat Pro has extensive editing features that permit an instructor to customize the book by removing parts, by adding parts, and by inserting comments. Likewise students can annotate the text.

Figure 3: Screenshot of a video lecture module. The full module is at http://www.science.oregonstate.edu/eBookWorking/VideoLecs/Fourier_Intro/FourierIntro.html



The page format of the eTextBook was designed for viewing with a PC that has a large screen, that can run Java-based applets, that has a Flash-enabled browser, and that has enough memory to store the videos lectures, codes, animations and applets locally. However, technology did not stand still while we were creating the book, so now we have also produced alternate versions of the text that are readable on tablets and eReaders, though with reduced functionality, and that stores the auxiliary files on the Web as opposed to locally. Considering that the video lecture modules occupy over 14 GB, in the future, and particularly for mobile devices, cloud storage would be the preferred approach.

We envision the mobile versions of the eTextBook as portable supplements to the PC-based text; to be read, listened to, and watched more than as a platform for code execution. At present, the mobile versions of the text are also pdf-based since the native book-reading formats are not adequate for equations and complicated tables. However, we are in the process of producing mobile “app”versions of the text designed for specific mobile platforms, and these should provide functionality similar to that of a PC.

Video-Based Lectures

The eTextBook contains some 60 lecture modules that cover almost every topic in the book. They are listed and linked to (in the pdf version of this chapter) in Table 2. The modules are based on developments by Greg Moses at the University of Wisconsin (eTeach), and use the

Camatasia Studio commercial screen capture package (Camtasia Studio). In the face of rapidly changing technologies, we rely on Camtasia's regular updates to keep our book from becoming obsolete while we were working on it. The modules were directed and produced by Sally Haerer and took over five years to produce. We decided that excellent sound quality was essential, as well as high legibility for the slides. The price paid for that is large files, with the modules occupying over 14 GB in total.

The lecture modules are a mix of Flash, Java, HTML and Mpeg. As seen in Figure 3, each module opens a Web page containing a video picture-in-picture of a professor discussing and demonstrating the material in his office, coordinated dynamic slides (sometimes with red scribbling on them), a dynamic table of contents, and links to codes and applets. In the eBook we indicate a video lecture with a marginal icon linked to a lecture.

Having the lectures produced in a studio setting with controlled sound, lighting, distance and video monitors results in a great improvement in quality compared to that of "live" lectures where the subject moves and often faces away from the viewer. In preparation for our recordings, our CP group viewed a number of online university courses and lectures, including some from the commercially successful *Great Lectures Collection* (The Teaching Company). We concluded that a scripted "lecture", while sounding polished with few speaking errors, is rarely engaging but frequently boring. In contrast, having a professor speak in his office, as he normally would from a slide containing just key points and equations, is more engaging and personal because it contains the spontaneity, depth, and imperfections that occur naturally when a knowledgeable person thinks and speaks about a subject they obviously care about.

Because of their dynamics, the preferred format for the video lecture

Table 2: The video lecture modules incorporated into the text. In the eTextBook each entry is a link to a lecture. A Web page with the same table and active links is at (Video Lectures).

CP: Education	8.1 N-D Search	10.4 DFT Alias	14.10 Parallel C	17.9 Heat, C-N
1.1 Intro to CP	8.3 Matrix Co	10.8 Fast FT	14.14 HPC Lab	18.1String Wave
1.3 C Basics	8.5 Interpolation	11.1 Wavelet I	14.15 HPC II	18.3 Cat Waves
1.5 Number Rep	8.8 Least Sq Fit	11.4 Wavelet II	15.1 Magnets	18.5 WavePacks
1.5 IEEE Floats	9 ODEs	11.5 Wavelet III	15.7 Feynman I	18.9 FDTD EM
1.5 Mach Prec	9.5 ODE Algor	12.1 Bugs	15.8 Feynman II	19.1 Solitons
2 Errors	9.5 ODE Lab	12.10 NL Pend I	16 MD I	19.6 Fluids
4 OOP I	9.10 QM Eigen	12.12 Pend II	16 MD II	20.1 Integl Eqn
5.1 Random MC	9.14 Chaos Scatt	12.14 Pend III	17 PDE Intro	20.3 Integl Eqn
5.3 MC Simulate	10 Int Fourier	13.1 Fractals I	17.2 PDE Elect	
6 Integration	10.4 DFT I	13.3 Fractals II	17.4 Electro II	
7 Differentiation	10.4 DFT II	14 HPC	17.10 Finite Ele	
7.7 Searching	10.5 Filters	14 HPC II	17.16 Heat Eqn	

modules is Abode Flash and Java. Yet we have also created MPEG-4 versions of the lecture modules for Apple mobile devices that cannot run Flash. That eliminates some of their interactivity and controls, but preserves the excellent sound and legibility, both important properties. A more universal approach for video would be to use HTML5, which Apple touts as a modern alternative (Apple). However, at present HTML5 is still under development, and so not a viable option.

Executable Codes/Paper

The free eTextbook posted on Compadre (2011), Merlot (2011) and CSERD contains many simulations as Python codes that the user can download. As a computer language, Python is very attractive: it appears to us as the easiest language to learn, it is free, it is available on essentially all platforms, it can be extended to serious scientific computing via over 19,811 packages (PyPI), it is also a scripting language that can run or be integrated with legacy codes, it is object oriented, and its arithmetic operations and control structures are similar enough to C and Java to make the numerical computing parts of it appear standard.

We have also created versions of all programs in Java, C and Fortran, and they are available in the commercial books or by separate purchase from the publisher.

As an example of our Python codes, consider the screenshots of a shaded code listing in Figure 3 and its output in Figure 4. Note that there are not many lines of code in Figure 3, with about half used to solve Laplace's equation for a line charge, and half used to produce the surface plot of the resulting potential in Figure 4. The book's simulations are an invaluable learning resource due to their interactivity, their

Listing 1.1 A sample code, **LaplaceLine.py**.

```
""" LaplaceLine.py:  Solution of Laplace's eqtn with 3D matplot """

from numpy import * ;      import pylab as p;    import matplotlib.axes3d as p3

print("Initializing")
Nmax = 100; Niter = 70; V = zeros((Nmax, Nmax), float)          # float maybe Float

print "Working hard, wait for the figure while I count to 60"
for k in range(0, Nmax-1):  V[k,0] = 100.0                             # line at 100V

for iter in range(Niter):                                  # iterations over algorithm
    if iter%10 == 0: print iter
    for i in range(1, Nmax-2):
        for j in range(1,Nmax-2): V[i,j] = 0.25*(V[i+1,j]+V[i-1,j]+V[i,j+1]+V[i,j-1])
x = range(0, Nmax-1, 2);  y = range(0, 50, 2)                # plot every other point
X, Y = p.meshgrid(x,y)

def functz(V):                                             # Function returns V(x, y)
    z = V[X,Y]
    return z

Z = functz(V)
fig = p.figure()                                                     # Create figure
ax = p3.Axes3D(fig)                                                    # plot axes
ax.plot_wireframe(X, Y, Z, color = 'r')                           # red wireframe
ax.set_xlabel('X')                                                    # label axes
ax.set_ylabel('Y')
ax.set_zlabel('Potential')
p.show()                                          # display fig, close shell to quit
```

Figure 3: An example of a code listing. Clicking on the caption opens the code in a browser (or runs it, depending upon system settings). This figure is linked to http://science.oregonstate.edu/eBookWorking/Codes/PythonCodes/LaplaceLine.py

visualization capabilities, their direct connection to the physical theory and the algorithms, and the direct programming style preferred by practicing scientists. Our goal was to have readers not only execute the codes, but also to have them look inside the codes in order to see how they function and to extend them as needed for their own projects and research. (Feedback indicates that many researchers find the text and especially the codes valuable.) In this way the codes constitute a virtual laboratory that adds another dimension to an eBook.

Running simulations without having to leave the text is an exciting possibility for textbooks. As we originally engineered the eBook, use of the hyperref package's *run* command (LaTeX/Hyperlinks) in the La-

TeX source file resulted in a pdf output that ran the codes directly. However, Abode apparently considered this a security risk and recent updates to their pdf readers forbid code execution. *Although document security is a legitimate concern, forbidding executable codes in documents limits the potential ways in which scientific and educational materials can be communicated and used.* Indeed, Elsevier Publishers has recognized this as a general and important problem, and has sponsored an *Executable Paper Grand Challenge* aimed at improving future operating systems and document formats so that applications can execute code without risking security (Executable Paper).

Although it is possible to copy and paste the eBook's pdf code listings into a Python development environment (IDLE), this is unreliable because the different levels of indentations used to define the codes' structures sometimes are not be preserved. However, the identations are preserved if they are stored in an HTML page stored as text. Accordingly, the caption of each code in the eTextbook (as well as a python snake head we use as an icon) links to a display of the code within a browser's window, and that listing can be reliably copied or executed directly. (Opening a saved copy of the code with a recent Firefox actually runs the code!)

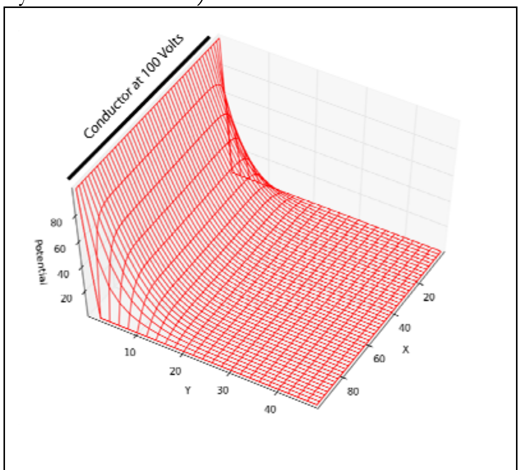


Figure 4: Output of the code LaplaceLine.py in Figure 3. The actual output can be rotated and zoomed.

The traditional techniques for incorporating executable codes into Web content is via applets written in Java, and the text is linked to the applet collection shown in Table 1 (Applets). Although you can interact with applets through their control panel just fine, looking inside them may not be educational for our intended readers because their graphics and interactivity often require complicated coding. (Python applets are still being developed.)

Executable Equations

Equations are such wonderfully succinct and meaningful representations of mathematical relationships, that it is a shame to have electronic documents include them only as pictures. A shame because bitmaps cannot be read by screen readers and contain no information about the meaning of the equation that can be used elsewhere. Accordingly, we would like the equations in our eTextBook to convey their meaning well enough so that a computer can process them with symbolic manipulation programs such as *Maple* or *Mathematica,* or search the Web for other documents related to these equations. Whereas this would be (more) straightforward if our entire book were in a markup language like MathML (discussed earlier), we have at least explored the possibility of executable equations in our pdf document by linking many of the text's equations to their corresponding *MathML* (W3C Math) versions.

We can demonstrate the effect if you are reading an electronic copy of this chapter. The equation below for Lagrange interpolation is a link to http://science.oregonstate.edu/eBookWorking/xml/Equation2.xml

$$
\begin{aligned}
g(x) &\simeq g_1\lambda_1(x) + g_2\lambda_2(x) + \cdots + g_n\lambda_n(x), \\
\lambda_i(x) &= \prod_{j(\neq i)=1}^{n} \frac{x-x_j}{x_i-x_j} = \frac{x-x_1}{x_i-x_1}\frac{x-x_2}{x_i-x_2}\cdots\frac{x-x_n}{x_i-x_n}.
\end{aligned}
$$

A displayed equation that is linked to an xml version that can be executed.

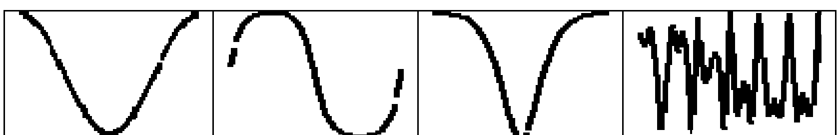
Clicking on this equation or the linked URL requests a browser to open up a window displaying an .xml file [MathML is written in xml (W3C Math)]. If your computer associates .xml files with Mozilla Firefox, the browser window should display a beautiful and scalable version of this equation. If you click on *Tools/Web Developer/Page Source*, you will see that this really is a MathML file and not a bitmap. This MathML source code can now be copied to and manipulated by symbolic manipulation programs. *A direct transfer or execution of the symbolic program within the text would be preferable, but again it seems like we must wait for the technology.*

Because every use of a numbered element in our eTextbook is linked to original equation, figure, listing, *etc.*, , another possibility is to have equations linked to their derivations, or to a fuller explanation of their meaning (or both). This could make texts less threatening to those readers who feel mathematically challenged, or just need some extra review which would otherwise interfere with the flow of the text. Of course this concept is related to having a text that can adapt to the knowledge level of the reader by accessing text materials included at different levels. While not hard to do technically, maintaining the coherence of an entire good text would be a challenge. However, publishers such as Princeton Press have moved in this direction by permitting educators to assemble text by combining pieces from different texts (Princeton).

Sonification

We have experimented over the years with using sound as a vehicle to involve another sense in visualizing data, a process known as *sonification*. Indeed, one of our Web enhancements, *Visualizing Physics with Sound*, explores sonification as an aide to sight-challenged readers, while another applet, *HearData*, converts a user's data into sound (Applets).

We have found sonifications to be particularly useful in our spontaneous decay simulation (which produces its own sound), and in the study of nonlinear oscillations. Specifically, the algorithm that simulates spontaneous decay *(if RandomNumber < rate, then decay)* is so simple, while the physical origin of spontaneous decay is so subtle, that some convincing is helpful. So when the reader hears the simulation produce sounds just like those coming from a Geiger counter, it helps reinforce the conclusion that nature must be doing something like our simulation. You can hear the simulation by clicking on the speaker, or http://www.science.oregonstate.edu/eBookWorking/Sound/geiger.au



Harmonic Oscillator	Realistic Pendulum	Anharmonic Oscillator	Particle Driven in a Square Well

Figure 5: Linear and nonlinear oscillations linked to their sonifications (Sound).

We have also found that sonification is very effective when teaching nonlinear oscillations. Some of our techniques are collected at (Sound) and shown in Figure 5. Although a graph of a nonlinear oscillation may look like a sine wave from a harmonic oscillator, it has higher-frequency overtones that tend to sharpen up the corners. You can see some differences in the graphs in Figure 5 or hear (the graphs are linked) the differences in sonifications as "richness" of the nonlinear oscillator's sound. In contrast, the sonification of a particle driven in square well sounds like noise, which is the sound of chaos.

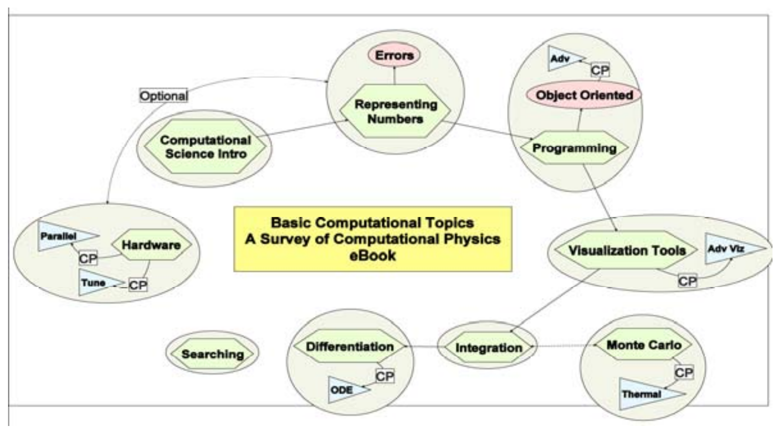


Figure 6: A concept map of the basic topics in the text (the physics map is more complicated). Each map object is linked to the appropriate section in the text.

Adobe Acrobat does have the ability to read text out loud, and that is another positive feature of having the book in pdf. Although the text does not yet have an automatically-linked dictionary that look up words as the cursor passes over them, as does the *Kindle*, the eTextBook does have many specialized words linked to its Glossary. And (for fun) if the reader does not wish to leave the page being read, he or she can click on a speaker logo near the linked word and hear the word spoken (GlossSound).

Executable Figures

Figures are often confusing for learners and especially those with disabilities because the figures contain elements that may be abstractions, as well as design elements that are irrelevant to the essential features. Our original plan was to have the eBook's figures in a *Vector Markup Language,* such as interactive Scalar Vector Graphics (SVG), with individual elements annotated to demonstrate their meaning, and various layers removable for simplification. This would also permit viewing by tactile readers with game-like feedback.

Although interactive SVG are not uncommon on the Web, being able to embed them in a pdf document readable by Acrobat still appears to be a research problem requiring a custom plugin. Doing this in an eTextbook meant for the general public seemed unwise at present, and so we have not included them in the eTextbooks we have placed online (Landau et al., 2011).

Nevertheless, our eTextbook does contain two concepts maps created with the Visual Understanding Environment (VUE), and we have inserted some interactivity into them. A concept map (Figure 6 is an example) shows in graphical form the relationships among the subjects in a knowledge field, analogous to the lines of force surrounding electric charges. (In the present case the knowledge field is the eBook.) Such diagrams have found use as a tool for organizing and representing the connections within a field of knowledge and the paths used to learn them. We included it as an alternative to the tradition format of a table of contents (which we also include with full internal linkages). Although you cannot try it with Figure 6 in this Word document, in our pdf eTextBook each map element is linked to related pages within the text or to slides of the lectures.

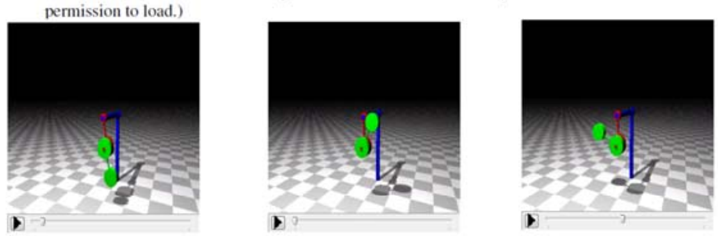


Figure 1.2 An embedded movie of a double pendulum simulation. (You may have to reduce reader's size to give permission to load.)

The text also uses various symbols and fonts to help clarify the type of material being dealt with. These include:

Figure 7: A composite showing three frames of an embedded animation side-by-side.

Animations are important in learning for the obvious reason that they make their subject come alive like nothing else can. They are included in a number of ways throughout the text. They are part of most of our applets and many of our simulations. In addition, we have movie projector icons that link to Mpeg files and media players, for example, http://science.oregonstate.edu/eBookWorking/Movies/2dsol.mpg).

Other figures in the text are encapsulated animations, which work right within the text. We give a simulation of this in Figure 7, which is actually three screen shots pasted together side-by-side; the real animation is in the center frame only. The effect of reading a text in which the figures are moving reminds us of the living newspapers in the Harry Potter movies (Harry, 2009). However, because animations encapsulated within the text increases the size of our pdf file significantly, we have used them sparingly.

Accessibility

In what is really a large and separate book in its own right, eBooks and their multimedia and multisensory enhancements can provide improved access for learners with various sensory disabilities. We have cooperated with projects on this subject and have tried to make our eTextBook accessible and open to improved accessibility. So, even though pdf is not a standard W3C format (it requires a plugin), Adobe has improved pdf accessibility in recent times (Hudson, 2004; AccessIT, 2011). In addition, we have made our LaTeX sources files for the text and slides available as needed, and even the equations in them are accessible with specialized readers (Raman, 1994).

Video by its very nature is not useful for the visually disabled. However, the high-quality sound and LaTeX slides are useful. Learners, and especially dyslexic ones, tend to benefit from being able to stop and replay the lectures modules, while hearing-disabled learners will have printed materials and slides to view. Sight-disabled readers have high-quality sound as well as the LaTeX source of the slides and text, in which even the equations are readable (Raman, 1994). And all students benefit from being able to attend lectures on their own schedules and at their own pace. Given the funding and need, we would gladly create an alternate fully accessible version.

Take a Look for Yourself

1. Go to our book page (Landau et al., 2011) and download pdf file to a local one.

2. Open the pdf file with a stand-alone pdf reader (a browser plugin might not show the page numbers).
3. Go to page iii and you should see the table of contents (TOC) in bright blue. As is traditional, we use blue to indicate links, both locally to other parts of the text and externally to Web servers. Click anyplace in the TOC and jump to that part of the text. Return to the TOC (*Alt + leftarrow returns to previous*).
4. Jump to page 4 where you will find Section 1.2, *Using the Feature of This Book*. Click on the 1.1 in Figure 1.1 in the first paragraph. This is an example of how every figure, equation, code, section, *etc.* is linked.
5. Back on page 4, look for a tiny image of a person in a blue shirt and click on it. This should take you to one of our video lectures, which you can explore, or spend the next 20 minutes watching. See (Video Lectures) for a listing of all the lectures.
6. On the bottom of page 4 there is a yellow "Applet"icon as well as the words "Chaotic Scattering". Clicking on either of these will take you to an applet on Classical Chaotic Scattering. See (Applets) for a listing of all the applets.
7. At the top of page 5 is a listing for the code **`Walk3D.py`**. Click on the caption to have a browser open with the code in a browser window. Copy and paste this code into (IDLE) to run it. Alternatively, save it and then open it again with Firefox to run it.
8. On page 6, click on the single equation there and an XML/MathML version of it should open in a browser window (we recommend Firefox for it to display properly). Find the appropriate menu on your browser so that you can view the "source". It should be a rather lengthy MathML file in which each element of each symbol is defined.
9. Again on page 6, click on the word "algorithm", to jump to the Glossary where the word is defined (*Alt + leftarrow*).
10. Now click on the little picture of a speaker to hear the definition without leaving the page.
11. On the bottom of page 6 you will find an icon of a movie projector. Click on it to view a movie in an external movie player.
12. On the top of page 7 you will find Figure 1.2. (If you do not see a double pendulum you may have to click here and tell Acrobat that you trust this document to load a file.) You should then see a picture of a double pendulum, which if clicked will play a movie embedded within the page.
13. You are on your own now to go explore and learn some computational physics.

Evaluation and Assessment

Evaluating cyber-enabled learning is a grand challenge problem, with multiple variables involved as well as a society in which the role of computers and communications is seeing a historically rapid change (Ainsworth et al., 2004). For example, we are just now working with the first generation of Web2 students who view electronic engagement and interactions as completely normal, who read news online more than from paper (Pew, 2011), and who buy, and presumably read, more electronic books from Amazon than paper ones.

Our eTextBook evolved from our paper textbook, which in turn evolved from over a decade's worth of class notes for our Computational Physics courses (CP Courses). The courses grew from an original two quarters into an entire curriculum and undergraduate degree program in Computational Physics (Landau, 2004). A number of professors taught the courses, with Landau leading the developments. The video lecture modules were developed over five years and used in both blended and online classes. We have taught the courses since the late 80's, and have published paper text books based on it in 1997 and 2008. These texts have been used throughout the world at approximately 50 schools that we know of. Both books have won prizes.

The developed courses, texts and lecture modules were designed around well-developed student learning outcomes (Landau, 2004). The courses and lecture modules have had formative and summative assessment via student evaluations, pre- and post-course student interviews and surveys conducted by an external evaluator (Wisconsin). The texts have had pre- and post-publication reviews, market surveys conducted by several publishers, feedback from faculty and students using the materials at other schools, and multiple discussions with collaborators. The general format of the lecture modules was adapted from the successful eTeach project (eTeach), with the delivery style (nonscripted lecture plus slides) similar to the commercially successful and academic respected *Great Courses* (The Teaching Company).

The first time we used a draft of the eTextBook and video lecture modules was in a blended course in which the lecture time was spent with the students in the lab. The evaluation was surprisingly positive in terms of student learning objectives and attitude. The only significant complain was related to difficulties in reading the DVD we used to distribute the draft and lectures. The students commented on the effectiveness of being able to stop and repeat parts of the lecture, and were thankful that the professor's breaks to sip tea (a deliberate design element) also gave the viewers breaks to gather their wits. More than once we have heard comments like "this is what you told us yesterday in lecture"; to us this indicated some level of acceptance of the electronic lecture as part of the student's reality. At the end of the first term, the students were disappointed that the next term would go back to live lectures. (We now have all terms completed.) In recent times, Landau has taught the Computational Physics courses purely online, which fits in well with his emeritus status, but loses some effectiveness for some of the students who appear to need the face-to-face intereaction.

The instructors for the blended course both had the impression that the students were better prepared for lab than when we had live lectures, although this may be due in part to our requirement that the students view the lectures before the lab. The instructors also concluded that the students' projects were at least of the same quality as before, with student questions and discussion of higher quality. Landau did teach the course the next term in the traditional format, and found the students to be as well prepared as usual, if not more so.

In recent times Landau has taught workshops at various schools and at the SuperComputing (SC) conferences using and distributing the eTextBook. A number of schools are now using those preliminary versions and providing feedback. This is good since in general changing the physics curriculum to include nontraditional topics, such as in our texts, is a slow process with a small number of early adopters catching on, but with institutions and organizations slow to change.

The detailed external assessment surveyed how much use each of the different features of the book received and how helpful was each. Here are some of the results:

- The students used the free eTextBook about 80% of the time and the paper one about 20%. All students used the eText some of the time.
- About 40% of the students did *not* read the instruction on how to use the text.
- Hyperlinks to figures, equations and codes were used about 50% of the time, with technical problems rare.
- There was a rather even distribution in all ranking categories as to the usefulness of the applets.

- None of the students bothered printing pages from the eTextBook, although code listings were printed some 10% of the time.
- None of the students thought the eTextbook lacked portability.
- About 30% of the students used the commenting, bookmarking and highlighting features possible with pdf.
- 100% of the students thought the written pages of the text were essential or fairly essential.
- 90% of the students thought the lecture modules were essential or fairly essential.
- The shaded code listings were considered essential.
- Running and seeing the results of simulations were considered essential or fairly essential by 90% of the students.
- The eTextbook was rated 4 stars out of 5.

Summary and Conclusions

A good book has no ending. –R.D. Cumming

We have created and placed online a complete eTextBook in Computational Physics that contains a wider variety of executable elements than present in most current eBooks. Some of these elements are essential, while others are explorations as to what future texts may be like. We have also indicated a number of features that would be useful in future eTexts, but which are not feasible with current technologies. Our choices for format and technologies were made to provide wide acceptance, high usability, platform independence, and no cost to readers. In December 2011 the full book was made available on a number of pathways (Landau et al., 2011) to the US National Science Digital Library (NSDL).

Textbooks have been around for a long time and that probably reflects their being a highly adaptable literary genre. Nevertheless, we believe that books and textbooks need to change in accord with the changes occurring in education, technology and society. Even while education changes to include workshop approaches that emphasize students discovering knowledge on their own, it seems that reading textbooks in their various forms remains essential for true understanding of a subject. And finally, while tablet computers have many attractions for viewing content, at present a PC or Mac still seems best for truly interacting with content via universal tools, and especially for learning a discipline involving computation.

References

AccessIT, The National Center on Accessible Information Technology in Education. (2011) Retrieved from http://www.washington.edu/accessit/articles?2

Ainsworth, S., Honey, M., Johnson, W. L., Koedinger, K., Muramatsu, B., Pea, R., Recker, M. and Weimar, S. (2004). *Cyberinfrastructure for Education and Learning for the Future: A Vision and Research Agenda,* Retrieved from http://archive.cra.org/reports/cyberinfrastructure.pdf

Apple**'s** view on HTML5 and web standards. Retrieved from http://www.apple.com/html5/

Applets from *A Survey of Computational Physics, Python Edition.* Retrieved from http://www.science.oregonstate.edu/eBookWorking/Applets/

Camtasia Studio Screen Recording and Presentation, Retrieved from http://www.techsmith.com/camtasia.asp

Carroll, J. M. and Olson, J. R. (1988). Mental Models in Human-Computer Interaction. M. Helander (Ed.), *Handbook of Human Computer Interaction.* Amsterdam: North Holland.

comPADRE, Physics & Astronomy Education Communities, Retrieved from http://www.compadre.org/portal/

CP Courses**,**

-Scientific Computing I, II:

http://www.physics.oregonstate.edu/~rubin/COURSES/ph265/

http://www.physics.oregonstate.edu/~rubin/COURSES/ph464/

-Computational Physics I, II:

http://www.physics.oregonstate.edu/~rubin/COURSES/ph465/

http://www.physics.oregonstate.edu/~rubin/COURSES/ph466/

Creative Commons. Help to share knowledge and creativity with the world. Retrieved from http://creativecommons.org/

CSERD, The Computational Science Education Reference Desk. Retrieved from http://www.shodor.org/refdesk/Catalog/Item/2490/summary

Coping with Unix, An Interactive Survival Kit. Retrieved from http://www.physics.oregonstate.edu/~rubin/nacphy/UNIX/

Davis, P. (1999). How Undergraduates Learn Computer Skills: Results of a Survey and Focus Group, *T.H.E Journal, 26*, 69. Retrieved from http://www.thejournal.com/articles/14120

Dede, C., Salzman, M., Loftin, R. B. and Sprague D. (1999). Multisensory Immersion as a Modeling Environnment for Learning Complex Scientific Concepts, Computer Modeling and Simulation in *Science Education*, eds. N. Roberts, W. Feurzeig, W. and B. Hunter, New York: Springer-Verlag.

eTeach, Learning on Demand. University of Wisconsin. Retrieved from http://www.sdsc.edu/pub/envision/v17.3/eteach.html

Executable Paper Grand Challenge. A contest created to improve the way scientific information is communicated and used. Retrieved from http://www.executablepapers.com

Extensible Markup Language. Retrieved from http://www.w3.org/XML/

Farah, M.J., Hammond, K.M., Levine, D.N. and Calvanio, R. (1988). Visual and spatial mental imagery: dissociable systems of representation, *Cognitive Psychology, 20,* 439-462.

GlossSound, a spoken definition of double precision. http://science.oregonstate.edu/eBookWorking/GlossarySound/doubleprecison.wav

Project Gutenberg. The first producer of free ebooks. Last modified on 20 August 2012. Retrieved from http://www.gutenberg.org/

'Harry Potter-style' video ads to be run inside U.S. paper magazine**.** (2009)**.** Retrieved from http://www.dailymail.co.uk/news/article-1207944/Harry-Potter-meets-Minority-Report-video-adverts-magazines-reality.html

Hartman, J. L. and Truman-Davis, B. (2000) Factors related to the satisfaction of faculty teaching online courses at the University of Central Florida. In J. Bourne and J. C. Moore (eds.), *Online Education 2: Proceedings of the 2000 Sloan Summer Workshop on Asynchronous Learning Networks.* Needham, MA. Retrieved from http://www.mendeley.com/research/factors-relating-satisfaction-faculty-teaching-online-courses-university-central-florida/

Hazzan, O. and Tomayko J. E. (2005). Reflection and Abstraction in Learning Software Engineering's Human Aspects, *IEEE Computer,* 39-46.

Hudson, R. (2004). PDF and Accessibility. Retrieved from http://www.usability.com.au/resources/pdf.cfm

IDLE, An Integrated Development Environment for Python. Retrieved from http://docs.python.org/library/idle.html

Jiang, M. and Ting, E. (2000). A study of factors influencing students' perceived learning in a web-based course environment. *International Journal of Educational Telecommunications 6(4)*, 317–338.

Landau, R. H. (2004) Computational Physics for Undergraduates, the CPUG Degree Program at Oregon State University. *IEEE Computing in Sci & Engr*, *6,* 68-75.

Landau, R. H., Kowallik, H. and Paez M. J. (1998). Web-Enhanced Undergraduate Course and Book for Computational Physics, *Computers in Phys*., *12,* 240-247. Retrieved from http://www.aip.org/cip/pdf/landau.pdf

Landau, R. H., Paez, M. J. & Bordeianu, C.. (2008). *A Survey of Computational, Physics, Introductory Computational Science*. Princeton, NJ: Princeton University Press;
ibid (2007) *Computational Physics*. Berlin: Wiley-VCH.

Landau, R. H., Paez M. J. & Bordeianu C. (2011). *A Survey of Computational Physics, Python Multimodal eBook, Version ß4.3, Introductory Computational Science*. Retrieved from http://www.science.oregonstate.edu/eBookWorking

Larkin, J. (1993). The role of problem representation in physics, 75-98, *Mental Models*, D. Gentner and A. Stevens Eds. Hillsdale, NJ: Lawrence Earlbaum Associates.

LaTeX/Hyperlinks (hyperref package). Retrieved from http://en.wikibooks.org/wiki/LaTeX/Hyperlinks

Merlot, Multimedia Educational Resource for Learning and Online Teaching. Retrieved from http://www.merlot.org/merlot/viewMaterial.htm?id=604391

Multimedia Educational Resource for Learning and Online Teaching (Merlot). Retrieved from http://www.merlot.org/merlot/viewMaterial.htm?id=604391

National Research Council (1991). *How People Learn*. M. S. Donovan, J. D. Bransford and J. W. Pellegrino Eds. Washington DC: National Academy Press.

NSDL, The National Science Digital Library. Retrieved from http://nsdl.org/

National Science Foundation. (1996). *Shaping the Future: New Expectations for Undergraduate Education in Science, Mathematics, Engineering, and Technolgy*. Retrieved from http://www.nsf.gov/publications/pub_summ.jsp?ods_key=nsf96139

Northwest Alliance for Computational Science and Engineering Computational Physics Research Group. Retrieved from http://physics.oregonstate.edu/~rubin/nacphy/

Pew Research Center's Project for Excellenece in Journalism. (2011). *State of the News Media 2011*. Retrieved from http://stateofthemedia.org/.

Physical Science Resource Center (PSRC, Compadre). Retrieved from http://www.compadre.org/psrc/items/detail.cfm?ID=11578

Picciano, A. G. (1998). Developing an asynchronous course model at a large, urban university. *Journal of Asynchronous Learning Networks 2(1)*. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.131.9318

Picciano, A. G. (2002). Beyond student perceptions: Issues of interaction, presence, and performance in an online course. *Journal of Asynchronous Learning Networks 6(1)*. Retrieved from http://faculty.weber.edu/eamsel/research%20groups/on-line%20learning/picciano%20(2002).pdf

Picciano, A.G. (2006). Blended Learning: Implications for Growth and Access. *Journal of Asynchronous Learning Networks, 10 (3)*. Retrieved from http://sloanconsortium.org/jaln/v10n3/blended-learning-implications-growth-and-access

Princeton University Press. Retrieved from http://press.princeton.edu/

PyPI - the Python Package Index (with links to Python). Retrieved from http://pypi.python.org/pypi

Ramage, T. R. (2001). *The "No Significant Difference" Phenomenon: A Literature Review. e-Journal of Instructional Science and Technology, 5*(1). Retrieved from http://spark.parkland.edu/ramage_pubs/1/

Raman, T. V. (1994). AsTeR: Audio System for Technical Readings. Retrieved from http://easi.cc/itd/volume1/number4/article2.html;
Emacspeak, the Complete Audio Desktop, Retrieved from http://emacspeak.sourceforge.net

Salzman, M. C., Dede, C., Loftin, R. B. and Chen J. (1999). A Model for Understanding How Virtual Reality Aids Complex Conceptual Learning; *Presence*: *8(3)* 293-316.

Shea, P., Fredericksen, E, Pickett, A., Pelz, W. and Swan, K. (2000) Measures of learning effectiveness in the SUNY Learning Network. In J. Bourne and J. C. Moore (eds.), *Online Education 2: Proceedings of the 2000 Sloan Summer Workshop on Asynchronous Learning Networks.* Needham, MA. Retrieved from http://www.suny.edu/sunytrainingcenter/files/ALNWorkshop2000.pdf

The Sloan Consortium. (2009) *Learning on Demand: Online Education in the United States, 2009*. Retrieved from http://sloanconsortium.org/publications/survey/learning_on_demand_sr2010

Sloan-C Annual Consortium, Blended LearningConference and Workshop. Retrieved from http://www.blendedworkshop.uic.edu

Sound**,** The NACSE Physics Group. Visualizing Physics with Sound. Retrieved from http://science.oregonstate.edu/eBookWorking/Sound/sound.html

SVG, Scalable Vector Graphics. Retrieved from http://www.w3.org/Graphics/SVG/

The Teaching Company. The Great Courses. Retrieved from http://www.teach12.com

Twigg, C.A. (2003*)*. Improving Learning and Reducing Costs. National Center for Academic Transformation, RPI. Retrieved from http://www.thencat.org/articles.html

UCES, The Undergraduate Computational Engineering and Sciences Program. Retrieved from http://www.krellinst.org/csgf/awards-contests/undergraduate-computational-engineering-and-sciences-uces-award-program

VUE, The Visual Understanding Environment. Tufts University. Retrieved from http://vue.uit.tufts.edu/

Wisconsin Center for Education Research, formerly Learning through Evaluation, Adaptation and Dissemination Center. Retrieved from http://www.wceruw.org/about.php

W3C Math Home. Retrieved from http://www.w3.org/Math/

Yasar, O. (2001). Computational science education: standards, learning outcomes and assessment. *Computational Science-ICCS , Int. Conf.,* V.N. Alexandrov et al., eds., Lecture Notes in Comput. Sci. 1159-1169. Berlin: Springer-Verlag.

Editors: Jovanovic, J. and Chiong, R. (Eds.). (2012).
Technology Enhanced Learning.
Santa Rosa, California: Informing Science Press.

Authors

Rubin Landau is a Professor of Physics Emeritus at Oregon State University. He has directed a basic research program in computational particle physics, and now develops educational materials for computational physics. He is the author or co-author of five texts in physics and computational physics, as well as the founder of the undergraduate degree program in Computational Physics, in which he still teaches some classes online. Landau is currently the Education Editor for the IEEE/AIP *Computing in Science & Engineering* magazine, a co-editor of a Computational Physics book series for Taylor and Francis, and sits on the XSEDE TEOS Advisory Committee. Email: rubin@science.oregonstate.edu

Manuel Paez is a Professor of Physics Emeritus at Universidad de Antioquia, Medellin, Colombia. His research speciality is computational nuclear physics, and his teaching has been in physics and computational physics, with programming in many languages. He has authored a number of educational manuals and videos, as well as coauthored three Computational Physics texts. Email: mpaezenator@gmail.com

Cristian Bordeianu is a researcher in the Faculty of Physics at the University of Bucharest, Romania, and teaches physics and computer science in technical high schools. He has over twenty years of experience in developing educational software for physics classes in high schools and universities. He is co-author of three texts in computational physics. Email: cristian.bordeianu@brahms.fizica.unibuc.ro

Index

Abstract concepts, 5, 6
Accessibility, 8, 19, 20
Animations, 7, 19, 21
Applets, 6, 9, 10, 11, 15, 17, 21, 23
Assessment, 21, 23
C language, 13
Camtasia Studio, 11
Code execution, 6, 10, 13, 14, 21
Computational Physics, 3, 22, 23
Concept maps, 18
Contents, 7, 20
Courses
  blended, 7, 23
  online, 7, 8
Electronic textbooks, 2
eReaders, 10
eTextbook
  contents, 8
Evaluation, 21, 23
Executable
  elements, 9
  equations, 15
  figures, 18
  Paper, 14
Flash, 11, 12
Fortran, 13
Gutenberg, 2
HTML, 5, 6, 11, 14
HTML5, 12
Hyperlinks, 5, 16, 20
hyperref package, 9, 14
Java, 11, 12, 13
LaTeX, 9
MathML, 9, 15
Mobile devices, 9, 10
Motivation, 3, 4
Mpeg, 11, 12
Multimedia enhancements, 5
Online version, 4, 8
pdf pros and cons, 9, 14
Problem solving paradigm, 4
Projects approach, 4
Python, 4, 12, 13, 14, 15
Simulations, 5, 12, 13, 19
Sonification, 16
Sound quality, 11

Studio production, 11
Text readers, 17, 20
Textbook history, 1
Video lectures, 10, 20
Virtual reality, 2, 7, 13
Web browser, 14, 16, 20, 21
World Wide Web, 5
xml, 9, 16, 21