Editors: Steven F. Barrett, steveb@uwyo.edu
Rubin Landau, rubin@physics.oregonstate.edu

A Blended, Multimodal Access eTextBook in Computational Physics

By Rubin H. Landau, Manuel J. Paez, Cristian C. Bordeianu

Taking advantage of emerging technology, a free eTextBook is discussed that will improve college students' learning. The eTextBook is platform-independent and highly useable, with multiple executable elements.

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

Our goal in educational development is to stimulate a systemic change in college-level curricula, in which science and computation are better integrated, which in turn better prepares students and improves learning. By creating textbooks in computational physics (CP), we can advance the integration of modern computational techniques into physics courses.[1] Because faculty members often report that they don't feel knowledgeable enough to teach a multidisciplinary subject like CP, our textbooks provide adequate materials to help them teach courses incorporating computation even with varying levels of experience.

Figure 1a illustrates the problem-solving paradigm of science at the core of computational research, which is used to structure our books. It symbolizes the essence of *computational scientific thinking* (we must understand the mathematical and computational basis of algorithms as well as the physical theory to have all three work together successfully). We encourage students to be creative as they work on a series of projects. In this way, students become actively engaged and emotionally involved with the materials as they experience the excitement of personal research. This approach also helps them take pride in their work, become 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, effcient, and relevant approach than often found in standard physics curriculum. Although students might have to take fewer physics classes to make room for computation, they learn the physics, CS, and math better when placed in context, which increases the learning process's efficiency and effectiveness.

Individual students learn in different ways, so it's essential for learners to form mental models to understand abstract subjects, such as physics.[2–5] Using multiple senses to interact with the materials improves the ability to form these mental models (the reader's brain is more activated when there are more ways to view the material), resulting in better learning.[6] Thus, an eTextBook makes sense for learning computational science. It's been shown that interacting with a computer and a textbook in a trial-and-error mode—as opposed to lectures and readings—is a better way to learn how to compute, think reflectively, and write.[7–10] Because our textbook is about computational science, it's particularly effective for our eTextBook to integrate the simulation, multimedia, and networking capabilities of a computer.

Factors and Processes for Creating an eTextBook

We began considering the creation of an electronic textbook in 2007 when we completed our paper-based text for *A Survey of Computational Physics*.[1] That text, which already had many electronic enhancements on an accompanying CD, included program listings in Java and other languages. Python had become very popular, particularly for physics education, so we created another version of the text in this language. In light of the economic realities of nonprofit publishing, we realized that an eTextBook was the best way to proceed, with the inclusion of a separate Python edition.

We first realized the potential of electronic textbooks much earlier—in November 1994—during an early meeting of the Undergraduate Computational Engineering and Sciences (UCES) program. Then, individuals interested in computational science education shared ideas and placed 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 (mathematics on the Web is still an issue), it was also clear that the "hyper" part of hypertext meant that a text could use multimedia (such as animation, sonification of data, simulations, and computer visualizations) to enhance education.

We assembled 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) are now online in a number of repositories associated with the National Science Digital Library (http://nsdl.org; we list some of the repositories in the "Contents" section).[11] We adopted a Creative Common license (http://creativecommons.org) for our eBook with the belief that materials developed with public funding should be available to the public. However, we're also developing 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 versions of the text and lectures, and an instructor's guide. (The LaTeX source files are available for readers with sight disabilities to read even the equations.)

We designed the eTextBook for viewing with a PC featuring a large screen, the ability to run Java-based applets, and a Flash-enabled browser (see Figure 2). We also produced alternate versions of the text that are readable on tablets and Kindles, though with reduced functionality. These mobile versions are excellent supplements to the PC-based text. They're also PDF-based, since the native Kindle format is inadequate for equations and complicated tables, and we've produced "app" versions of the text that can provide functionality similar to that of a PC.

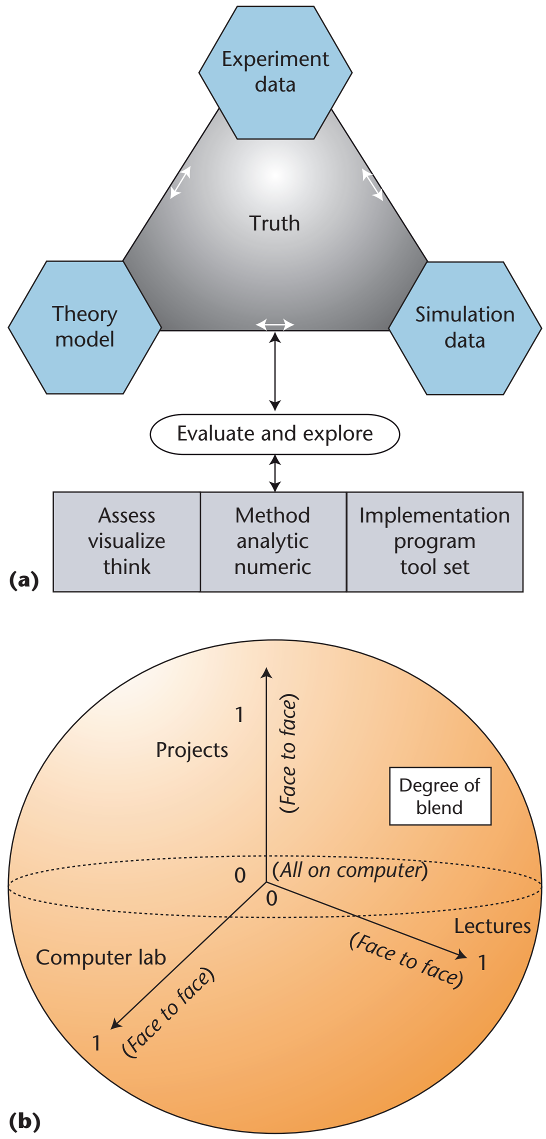


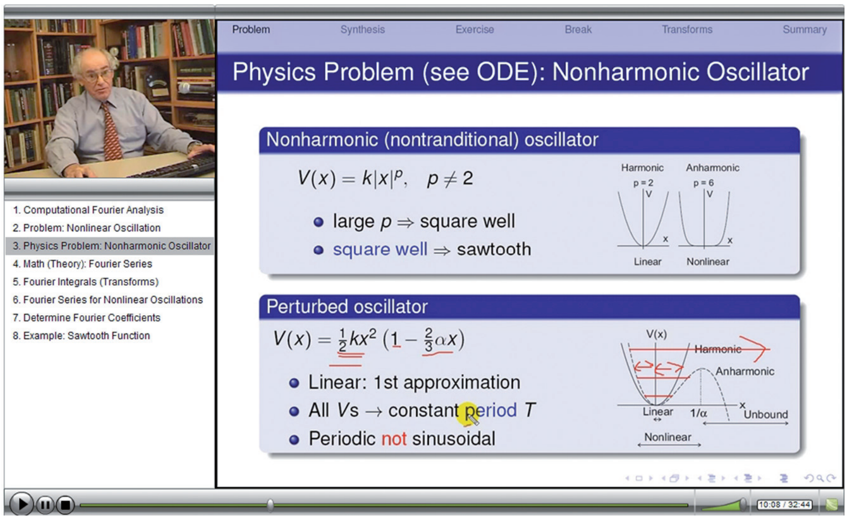
Figure 1. To help students learn how to think like a computational scientist, we use a variety of tools. (a) Our scientific problem-solving paradigm encourages students to be creative as they work. (b) A 3D space with projects, lectures, and lab axes—in which the degree of blend varies from 0 (all on computer) to 1 (all face-to-face)—uses multiple senses to activate the student's brain in more than one way.

Blended Courses

Over the past decade, we've 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 online enrollment (up 17 percent in 2009).[12] More than a decade of literature has shown that measurable learner outcomes result more from the quality of an entire course and its delivery than from the specific delivery media.[13,14] In general, online courses are capable of working as well as traditional ones, and blended or hybrid courses might be superior.[15,16] In the latter, faculty blend the course components, with each containing variable degrees of face-to-face, or face-to-computer learning (see Figure 1b).

We find that the effectiveness of a course is improved significantly when we can vary the degree of the blend to fit the local educational environment and students. Likewise, an eTextBook that contains video-based lecture modules, laboratory exercises, problems, and in-depth reading 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 take more work to create, as does an online course compared to a face-to-face one. The amount of work is reduced if we incorporate the creation of the eTextBook—or at least its major elements like video lectures and slides—into the development of a course.

There are multiple variables when comparing online to offline and blended courses, as well as when comparing traditional to fully electronic textbooks. In either case, isolating one element might not be reliable. Many students



(a)

```
""" 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
```

(b)

Figure 2. Samples from the eTextBook. (a) Screenshot of a video lecture module. (b) A Python code listing.

excel using online courses in which they can set their own hours and paces, but online courses don't work as well for students who are less motivated and who view the computer as a device for entertainment and socializing. These students often lack the self-discipline needed to sustain 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 are more likely to view the computer professionally.

Contents

To make the materials widely available, we posted the eTextbook on Compadre (www.compadre.org/psrc/items/detail.cfm?ID=11578), Merlot (www.merlot.org/merlot/viewMaterial.htm?id=604391), the Computational Science Education Reference Desk (CSERD; www.shodor.org/refdesk/Catalog/Item/2490/summary), and on the Oregon State University (OSU) science server.[11]

Video-Based Lectures

The eTextBook contains around 60 lecture modules covering almost every topic ("VideoLecs). The lecture modules are a mix of Flash, Java, HTML, and MPEG files. Each module opens a webpage containing a video picture-in-picture of a professor 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 (see Figure 2a). The modules are based on Greg Moses' eTeach developments,[17] and they use the Camtasia Studio commercial screen capture package. In the face of rapidly changing technologies, we relied on Camtasia's regular updates to keep our book current. It took five years for Sally Haerer (our director and producer) to produce the modules. Excellent sound quality and legible slides are essential for an eTextBook, but they create large file sizes, with the video modules occupying more than 14 Gbytes.

There's a great improvement in quality when we produce lectures in a studio setting with controlled sound, lighting, distance, and video monitors as opposed to 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 Courses" collection (www.teach12.com). We concluded that a scripted lecture is rarely engaging and frequently boring, although it sounds polished with few speaking errors. In contrast, our recordings feature a professor speaking in his office, as he normally would speak, referring to slides which contain only key points and equations. We find this to be more engaging and personal because of the natural spontaneity, depth, and imperfections that occur when a knowledgeable person speaks about a subject about which he obviously cares.

Python Codes

The éTextBook contains many simulations as downloadable Python codes. As a computer language, Python is attractive: it's free, available on essentially all platforms, and object-oriented. It's the easiest language to learn and can be extended to serious scientific computing via more than 19,811 packages (https://pypi.python.org/pypi). It's also a scripting language that can run or be integrated with legacy codes, and its arithmetic operations and control structures are similar enough to C and Java, making the numerical computing parts appear standard. All of the programs are also available in Java, C, and Fortran versions (available either in the commercial books or by purchasing them from the publisher).

As an example of our Python codes, consider the screenshots of a code listing in Figure 2b. Note that there aren't many lines of code. Half of these are used to solve Laplace's equation, and half are used to produce the surface plot of the resulting potential. The book's simulations are an invaluable learning resource due to their interactivity, visualization capabilities, direct connection to the physical theory and algorithms, and the straightforward programming style preferred by practicing scientists. Our goal was to have readers not only execute the codes, but also to examine the codes to see how they function and how to extend them 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, adding another dimension to the eBook.

Another exciting possibility for textbooks is the ability to run simulations without having to leave the text. In our original version of our eBook, the hyperref package's `run` command (http://en.wikibooks.org/wiki/LaTeX/Hyperlinks) in the LaTeX source file resulted in a PDF output that ran the codes directly. However, recent updates to Adobe's PDF readers forbid code execution. Although we recognize that there might be a concern for document security, 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 problem and has sponsored an "Executable Paper Grand Challenge" to improve future operating systems and document formats (www.executablepapers.com).

Although it's possible to copy and paste the eBook's PDF code listings into a Python development environment—such as the integrated development environment IDLE—it's unreliable because the different levels of indentations used to define the codes' structures might not be preserved. Accordingly, the caption of each code in the éTextbook links to a Python file that displays in a browser's window (for example, `"Codes/PythonCodes/LaplaceLine.py`). That listing can be copied reliably or executed directly. (Opening a saved copy of the code with a recent Firefox browser actually runs it!)

Executable Equations and Figures

Equations are wonderfully succinct and meaningful representations of mathematical relationships; it's a shame for electronic documents to include them only as pictures, because bitmaps can't be read by screen readers and don't contain information about the equation's meaning. The equations in our éText-Book convey their meanings well enough for a computer to process them with symbolic manipulation programs—such as Maple or Mathematica—or search the Web for other related documents. Whereas this would be more straightforward if our entire book was in a markup language like MathML, we've at least explored the possibility of executable equations in our PDF document by linking many of the equations to their corresponding MathML versions.

We demonstrate the effect here. The following equation for Lagrange interpolation would be linked to `"/xml/Equation2.xml`:

$$g(x) \simeq g_1\lambda_1(x) + g_2\lambda_2(x) + \cdots + g_n\lambda_n(x),$$

$$\lambda_i(x) = \prod_{j(\neq i)=1}^{n} \frac{x - x_j}{x_i - x_j} = \frac{x-x_1}{x_i-x_1}\frac{x-x_2}{x_i-x_2}\cdots\frac{x-x_n}{x_i-x_n}.$$

To open a window displaying an XML file (MathML is written in XML), click on the equation. 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'll 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. We would prefer a direct transfer execution of the symbolic program within the text, but again the current technology doesn't allow this.

Because every use of a numbered element in our éTextbook is linked to an original equation, figure, listing, and so on, another possibility is for equations to be linked to their derivations, or to a fuller explanation of their

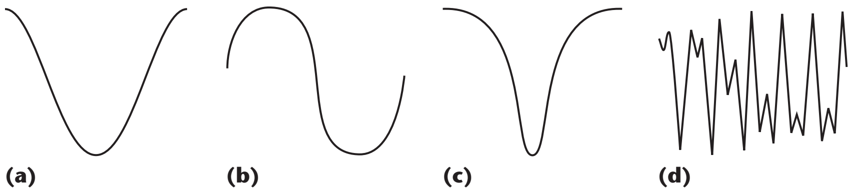


Figure 3. Linear and nonlinear oscillations. (a) Harmonic oscillator, (b) realistic pendulum, (c) anharmonic oscillator, and (d) particle driven in well. Links to their sonifications are at `"Sound`).

meaning (or both). This could make texts less threatening to those readers who feel mathematically challenged or who need review, which would otherwise interfere with the text's flow. Of course, this concept requires a text that can adapt to the knowledge level of the reader by accessing text materials included at different levels. Although this isn't technically hard to do, it's challenging to maintain the coherence and planned logic of a well-designed text. However, publishers such as Princeton Press have moved in this direction, permitting educators to assemble text by combining pieces from different text sources.

Sonification

Over the years, we've experimented with sound as a vehicle to involve another sense in visualizing data—a process known as *sonification*. One of our Web enhancements, "Visualizing Physics with Sound," explores sonification as an aide to sight-impaired readers, while another applet, HearData, converts a user's data into sound (`"/Applets` folder of the eTextBook).

We've found sonification 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 RandomNumbe < rate, then decay`) is so simple, and the physical origin of spontaneous decay so subtle, some convincing is helpful. So when the reader hears the simulation produce sounds like those coming from a Geiger counter, he's more likely to draw the correct conclusions. You can hear the simulation at `"/Sound/geiger.au`.

We've also found that sonification is effective when teaching nonlinear oscillations. Some of our techniques are collected at `"/Sound/sound.html` (see Figure 3). Although a graph of a nonlinear oscillation may look like a sine wave from a harmonic oscillator, it has higher-frequency overtones that sharpen the corners. You can see some differences in the graphs (Figure 3) or hear the differences in the sonifications as a "richness" of the nonlinear oscillator's sound. In contrast, the sonification of a driven particle in a square well sounds like noise (chaos).

Adobe Acrobat can read text out loud, which is another positive feature of PDF. Although the text doesn't have an automatically linked dictionary that searches for words as the cursor passes over them—as does the Kindle—the eTextBook has many specialized words linked to its glossary. And (for fun) if the reader doesn't wish to leave the page being read, she can click on the speaker logo near the linked word and hear the word spoken.

Accessibility

eBooks and their multimedia and multisensory enhancements can provide improved access for learners with various sensory challenges. We've cooperated with projects on this subject and have tried to make our eTextBook accessible and open to improved accessibility. So, even though PDF isn't a standard World Wide Web Consortium (W3C) format because it requires a plug-in, Adobe has recently improved PDF accessibility.[18,19] We've also made our LaTeX source files for the text and slides available as needed and the equations accessible with specialized readers.[20]

For obvious reasons, video isn't useful for some learners with visual impairments, but high-quality sound and LaTeX slides are. We've printed materials and slides for hearing-impaired learners, and all learners benefit from the ability to replay lecture modules and from attending lectures on their own schedules and at their own paces. Given the funding and need, we'd like to create an alternate, fully accessible version.

Take a Look for Yourself

Now that we've walked you through some of the features, we invite you to check out the current version. Visit http://science.oregonstate.edu/eBookWorking and download the PDF book file to a local file.

Open the PDF file with a standalone PDF reader (a browser plugin might not show the page numbers). Go to page iii for the table of contents (ToC) in bright blue. We use the standard blue to indicate links both to other parts of the text within the book and to external websites. Click any place in the ToC to jump to that part of the text. Return to the ToC (*Alt* + *leftarrow* returns to the previous text).

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, and so on. is linked. Return to page 4 and click on the tiny image of a person in a blue shirt. This will take you to one of our video lectures, which you can explore or watch in its entirety (about 20 minutes). See `"VideoLecs` for a listing of all the lectures.

On the bottom of page 4, there's 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.

At the top of page 5 is a listing for the code `Walk3D.py`. Click on the caption to open the code in a browser window. Copy and paste this code into IDLE to run it. You can also save it and open it again in Firefox to run it.

On page 6, click on the single equation and an XML/MathML version of it will open in a browser window. (We recommend Firefox for proper display.) Find the appropriate menu on your browser to view the "source." It's a rather lengthy MathML file in which each element of each symbol is defined.

Again on page 6, click on the word "algorithm" to jump to the glossary where the word is defined (*Alt* + *leftarrow*). Now click on the little picture of a speaker to hear the definition without leaving the page. Click on the icon of a movie projector at the bottom of the page to view a movie in an external player.

On the top of page 7 is Figure 1.2. (If you don't see a double pendulum, you might have to tell Acrobat that you "trust" this document to load a file.) You should then see a picture of a double pendulum. Click it to play a movie embedded within the page.

Evaluation and Assessment

Evaluating cyber-enabled learning is a challenge, involving multiple variables and an environment in which the role of computers and communications is seeing historically rapid change.[21] For example, we're working with the first generation of Web2 students, for whom electronic interaction is natural, who read news online more than from paper,[22] 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 CP courses. The courses grew from an original two quarters into an entire curriculum and undergraduate degree program in CP.[23] Several professors taught the courses, with one of our authors (Rubin Landau) leading the developments. The video lecture modules were developed over a period of five years and have been used in both blended and online classes. We've taught the courses since the late 1980s and published paper textbooks in 1997 and 2008. These texts have been used throughout the world in at least 50 schools. Both books have won prizes. We designed the courses, texts, and lecture modules around well-developed student learning outcomes.[23] We've had extensive assessment of the courses and lecture modules using student evaluations, pre- and post-course student interviews, and surveys conducted by an external evaluator (the Wisconsin Center for Education Research; www.wceruw.org/about.php). 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. As we've mentioned, the general format of the lecture modules was adapted from the successful eTeach project,[17] with the delivery style (nonscripted lecture plus slides) similar to the commercially successful and academically respected "Great Courses."

We first used a draft of the eTextBook and video lecture modules in a blended course, in which the instructor lectured from the lab. The evaluation was surprisingly positive in terms of student learning objectives and attitude. The only significant complaint 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 repeat parts of the lecture and were thankful that the professor's breaks to "sip tea" (a deliberate design element) also gave them breaks to gather their wits. More than once we've heard comments such as "this is what you told us yesterday in lecture"; to us, this indicates acceptance of the electronic lecture as part of the student's reality. At the end of the first term, the students were disappointed that we would return to live lectures. In recent times, one of our authors (Rubin Landau) has taught the CP courses purely online, which fits in well with his emeritus status, but is less effective for some of the students who need the face-to-face interaction.

The instructors for the blended course felt that the students were better prepared for lab than in the live lectures, although this could 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 discussions of higher quality. Landau taught the course the next term in the traditional format and found the students to be as well prepared as usual, if not more-so.

Landau has recently taught workshops at various schools and at SuperComputing conferences using and distributing the eTextBook. Many schools have used the developing forms of the eTextBook and have provided feedback. This is very promising. Usually, it's a slow process to change physics curriculum to include nontraditional topics (such as those in our texts). Generally, a small number of early adopters catch on, but institutions and organizations are slow to change.

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

- Students used the free eTextBook about 80 percent of the time and the paper textbook about 20 percent; all students used the eTextBook at least some of the time.
- About 40 percent of the students didn't read the instructions on how to use the text.
- Hyperlinks to figures, equations, and codes were used about 50 percent of the time, rarely with technical problems.
- The students consistently found the applets to be useful.
- None of the students printed pages from the eTextBook, although code listings were printed 10 percent of the time.
- They all agreed that the format was portable and worked on a variety of devices.
- About 30 percent of the students used the commenting, bookmarking, and highlighting features.
- All of the students reported that the written pages of the text were essential or fairly essential.
- Ninety percent of the students said that the lecture modules were essential or fairly essential.
- The shaded code listings were considered essential.
- Ninety percent of the students reported that the ability to run and see the results of simulations was essential or fairly essential.
- Students gave the eTextbook a rating of four stars out of five.

Overall, these assessments show that our eTextBook is successful in its goals.

> A good book has no ending.
> —R.D. Cumming

The eTextBook we created contains a wider variety of executable elements than is available currently in most eBooks. Some of these elements are essential, while others are explorations into what future texts might be like.

Textbooks have been around for a long time—a reflection that they're a highly adaptable literary genre. Nevertheless, we believe that books and textbooks must evolve with education, technology, and society. Even while education changes to include workshop approaches that emphasize student discoveries, reading textbooks—in their various forms—remains essential for true understanding. And while tablet computers are very attractive for viewing content, it still takes a computer to truly interact with content via universal tools, and especially so for learning a discipline involving computation. CiSE

Acknowledgments

This article is based on a talk presented at the Extreme Science and Engineering Discovery Environment (XSEDE12) Conference (https://www.xsede.org/web/xsede12). The eTextBook can be found at http://science.oregonstate.edu/eBookWorking. This work was supported in part by the US National Science Foundation (NSF) grant CCLI-0836971 and the NSF Engaging People in Cyberinfrastructure project.

References

1. R.H. Landau, M.J. Paez, and C.C. Bordeianu, *A Survey of Computational Physics: Introductory Computational Science*, Princeton Univ. Press, 2008.
2. J.M. Carroll and J.R. Olson, *Mental Models in Human-Computer Interaction: Research Issues about What the User of Software Knows*, tech. report, 1987; www.nap.edu/openbook.php?isbn=POD266.
3. C. Dede et al., "Multisensory Immersion as a Modeling Environnment for Learning Complex Scientific Concepts," *Modeling and Simulation in Science and Mathematics Education*, Springer-Verlag, 1999, pp. 282–319.
4. M.J. Farah et al., "Visual and Spatial Mental Imagery: Dissociable Systems of Representation," *Cognitive Psychology*, vol. 20, no. 4, 1988, pp. 439–462.
5. M.S. Donovan, J.D. Bransford, and J.W. Pellegrino, eds., *How People Learn*, Nat'l Academy Press, 1991.
6. M.C. Salzman et al., "A Model for Understanding How Virtual Reality Aids Complex Conceptual Learning," *Presence*, vol. 8, no. 3, 1999, pp. 293–316.
7. P. Davis, "How Undergraduates Learn Computer Skills," *THE Journal*, vol. 26, no. 69, 1999, pp. 68–71.
8. O. Hazzan and J.E. Tomayko, "Reflection and Abstraction in Learning Software Engineering's Human Aspects," *Computer*, vol. 38, no. 6, 2005, pp. 39–46.
9. *Shaping the Future: New Expectations for Undergraduate Education in Science, Math., Engineering, and Technology*, tech. report NSF96139, US Nat'l Science Foundation, 3 Oct. 1996; www.nsf.gov/publications/pub_summ.jsp?ods_key=nsf96139.
10. J. Larkin, "The Role of Problem Representation in Physics," *Mental Models*, Lawrence Earlbaum Associates, 1983, pp. 75–98.
11. R.H. Landau, M.J. Paez, and C. Bordeianu, *A Survey of Computational Physics:* Introductory Computational Science, Python Multimodal eBook beta version 4.3, Princeton Univ. Press, 2012; http://science.oregonstate.edu/eBookWorking.

12. I.E. Allen and J. Seaman, *Learning on Demand: Online Education in the United States, 2009,* report, The Sloan Consortium, 2009; http://sloanconsortium.org/publications/survey/learning_on_demand_sr2010.
13. T.R. Ramage, "The 'No Significant Difference' Phenomenon: A Literature Review," *J. Instructional Science and Technology*, vol. 5, no. 1, 2002, pp. 1–7.
14. A.G. Picciano, "Beyond Student Perceptions," *J. Asynchronous Learning Networks*, vol. 6, no. 1, 2002, pp. 21–40.
15. C.A. Twigg, *Improving Learning and Reducing Costs: Redesigning Large-Enrollment Courses*, The Pew Learning and Technology Program, 2003; www.thencat.org/monographs/mono1.pdf.
16. A.G. Picciano, "Blended Learning: Implications for Growth and Access," *J. Asynchronous Learning Networks*, vol. 10, no. 3, 2006, pp. 95–102.
17. G. Moses et al., "eTeach: New Technologies for Active Learning," *EnVision*, vol. 17, no. 3, 2001; www.sdsc.edu/pub/envision/v17.3/eteach.html.
18. R. Hudson, "PDFs and Accessibility," *Web Usability*, 1 May 2004; http://usability.com.au/2004/05/pdfs-and-accessibility.
19. The Nat'l Center on Accessible Information Technology in Education, "Is PDF Accessible?" *AccessIT*, Univ. of Washington, 2011, www.washington.edu/accessit/articles?2.
20. T.V. Raman, "AsTeR: Audio System for Technical Readings," *Information Technology and Disabilities J.*, vol. 1, no. 4, 1994; http://easi.cc/itd/volume1/number4/article2.html.
21. S. Ainsworth et al., *Cyberinfrastructure for Education and Learning for the Future: A Vision and Research Agenda*, tech. report REC-0449247, Computing Research Assoc., Nat'l Science Foundation, 2004.
22. T. Rosenstiel and A. Mitchell, *State of the News Media 2011: An Ann. Report on Am. Journalism*, report, Pew Research Center's Project for Excellence in Journalism, 2011; http://stateofthemedia.org/overview-2011.
23. R.H. Landau, "Computational Physics for Undergraduates: The CPUG Degree Program at Oregon State University," *Computing in Science & Eng.*, vol. 6, no. 2, 2004, pp. 68–75.

Rubin H. Landau is a distinguished professor and professor emeritus in physics from Oregon State University and the founder of the Computational Physics Degree Program. His research interests include few body systems in nuclear and particle physics, computational quantum mechanics, and scattering theory. Landau has a PhD in physics from the University of Illinois. He is a Fellow of the American Physical Society. Contact him at rubin@physics.oregonstate.edu.

Manuel J. Paez is a professor of physics at Universidad de Antioquia in Colombia. His research interests focus on computational physics algorithms and programming. He has a PhD in computational nuclear physics from Oregon State University. Contact him at mpaezenator@gmail.com.

Cristian C. Bordeianu is a physics and computational science teacher at Colegiul Militar Liceal (Ştefan cel Mare) at Câmpulung Moldovenesc, Romania. His research interests include computational, high-energy, and educational physics. Bordeianu has a PhD in physics from the University of Bucharest, Romania. Contact him at cbord1@gmail.com.
