Editor: Denis Donnelly, donnelly@siena.edu

TEACHING COMPUTATIONAL METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS USING THE WEB

By André Jaun, Johan Hedin, and Thomas Johnson

COMPUTATIONAL METHODS ARE PART OF THE PROBLEM-SOLVING SKILLS THAT PROFESSIONALS WORKING IN QUANTITATIVE FIELDS NEED. ADVANCED-LEVEL TEXTBOOKS PROVIDE THE MATHEMATICAL FOUNDATION FOR ONE SPECIFIC APPROACH; however, they miss the overview and examples that are generally necessary to choose the right method and implement a practical solution. Internet technology can be of great value in this context. Thus, we created a problem-based learning environment where knowledge is acquired by performing well-defined tasks. We also switched from a teaching-centered course to a learning-centered course–the students choose their own focus, order and pace as they explore the material (see http://pde.fusion.kth.se).

Since 1997, the concept has been validated three times in summer courses with about 15 participants geographically dispersed around Stockholm and Göteborg. It is now being tested all year round with distance-learning students on the Web.

A distance learning setup

Prior to registration, Web links to the course notes and former student projects enable potential participants to decide whether to enroll. This way of proceeding reflects the current trend towards a free market for university courses and is well adapted to offering teaching services outside traditional school boundaries, such as sister universities and private companies.

Every morning during the first two weeks of class, the students either attend a lecture in a videoconference room or they download a video recording from the Internet. Both are optional and serve as an introduction to a second, active learning phase in which the students run experiments in a Java-powered Web version of our course notes.[1] They do their assignments in a regular Web browser and then submit their work electronically to the teacher for correction. Discussion forums exploit the ability of quick learners to answer simple questions from peers and enable the teacher to focus on problems where his expertise is most useful.

Classroom discuss Web page equivalents of the course notes and are broadcast by videoconference to remote participants. To keep the students attention span, the lessons were relatively short (30-40 minutes). We would quiz them periodically to stimulate discussion locally before the conclusions are shared between remote classrooms.

After an analytical derivation, we use the Jbone applet (*J*ava *b*ed for *one*-dimensional problems) to test new schemes directly in the Web browser, adding an unprecedented animation and interactivity to the lecture. This is valuable when we compare the numerical properties of different time evolution schemes. The students use menus to select the equation (advection diffusion, Burger's shock waves, Korteweg-De Vries solitons, Black-Scholes options, or Schrödinger) and the initial condition (box, Gaussian, cosine, soliton, put option, or wavepacket), and they controll the parameters (velocity, diffusion, dispersion, time step, etc.) by editing text fields directly in the Web browser. A mouse click starts the simulation, making it easy, for example, to illustrate the linear instability that occurs when the time step gets too large in an explicit finite-difference advection scheme, or to show how the more subtle aliasing in spectral methods affects a nonlinear train of solitons.

Active learning

An advantage of using widespread, platform-independent technology is that the students can reproduce and modify the demonstrations from their offices and homes. Repeating the line of thought from the classroom, the text and figures take the reader through a series of analytical derivations that yield a computational scheme. Hyperlinks point to the relevant sections in the code and show how every algorithm has been implemented. Default parameters are preset to illustrate specific properties, but they can be modified to verify if students have correctly understood a topic.

For example, consider Brownian motion and diffusion; both are fundamental in science and engineering and often hard for undergraduates to understand.

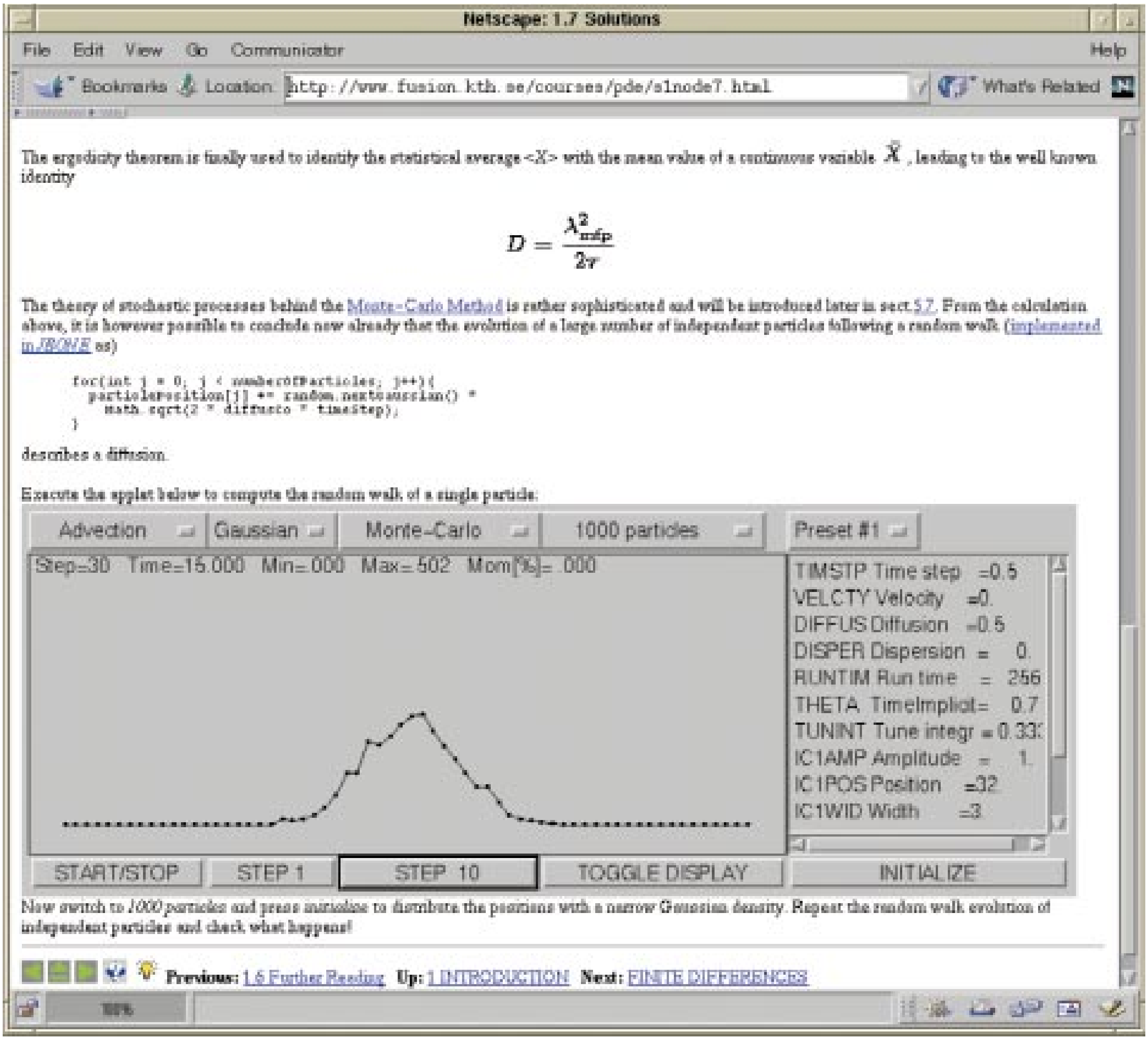

Figure1. Screen capture of the Web browser displaying an analytical formula and the algorithm with hyperlinks into the source code and the Jbone applet—all after executing a Monte Carlo integration with 1,000 particles to illustrate the connection between the motion of random walkers and diffusion.

An analytical derivation of the RMS displacement $\sqrt{\langle \xi_i^2 \rangle}$ in a particle's random walk connecting the diffusion coefficient to the square of the mean free path divided by the collision time $D = \lambda_{mfp}^2 / 2\tau_c$ might enlighten a few, but it is likely to lose most students in the algebra. Using the Monte Carlo evolution from the applet displayed in Figure 1, it is simple to demonstrate first how the random motion of a single particle can be described with the Java code:

```
for (int j=0; j<numberOf
        Particles; j++){
    particlePosition[j] +=
        random.nextGaussian() *
        Math.sqrt(2 * diffusCo
          * timeStep);
}
```

Distributing 1,000 particles to approximate an initial Gaussian distribution, the same algorithm immediately shows what diffusion is. Change the initial condition to a cosine or a square box and it is equally rewarding to demonstrate how short wavelength modes decay more rapidly than long wavelengths.

We designed the exercises in the first class session to familiarize students with electronic publishing on the Web. The templates show how to use the building blocks in relevant schemes to assimilate Tex and Java directly in the context of the exercise. For the first exercise, we provided model equations and program structures the students can edit directly in the browser and adapt to what they want to say. Last year, we decided to distribute a list of all the solution Web pages and let the students compare and discuss the results with each other.

Some students chose to carry out an additional one-week project, applying their favorite method to a topic of interest such as the Black-Scholes equation for a European call option, a tunable finite-elements integration for the Schrödinger equation, a mesh refinement procedure, an iterative solver, and so on. Given the small amount of time allocated for each project, the scope remained limited. By cross-checking each other's reports on the Web, the students got an overview of a rather broad range of applications.

A single Tex source file generated both the printed compendium and the hyperlinked Web pages. By running open-source translators such as latex2html,[2] tth,[3] and scripts embedded in a makefile, you can produce the static Web material effectively with no additional cost to what is already required to print a compendium and slides. Although writing the Jbone applet from scratch was challenging for the teacher, the object-oriented language and the code's encapsulated structure gradually enabled students with little programming experience to create their own schemes and modify existing ones. A substantial amount of documentation (programming tree, keyword index) is created by javadoc, the open source program, automatically using the javadoc utility—part of the standard Java development kit.

News groups or discussion forums prove to be an ideal tool, allowing a sufficient number of participants to interact in a geographically distributed environment. Some supervision was required, but instead of answering the same question a dozen times (often for organizational matters), the teacher intervened once for an announcement, then spent the rest of the time clarifying informal discussions and helping students understand the subject in their own language. We now reward relevant contributions to discussions, as well as completed assignments, to encourage interaction between students and create the feeling of belonging to the same virtual classroom.

Considerable steps have been taken since we first taught the course in a conventional manner in 1997. It is easy to publish course notes on the web, but from a pedagogical point of view, however, the largest benefit comes from the problem-based learning environment. This is much more delicate to implement with remote participants when firewalls, different computer platforms, and multiple versions of software and operating systems can quickly become a technology nightmare.

In its full electronic form, the course clearly requires a well-maintained Web server. Because of the "ask once, answer to all" nature of the discussion forums, assistants can be employed very efficiently and the overall teaching load is finally similar to a conventional setup.

Some flexibility is required from both the lecturer and the students in order to exploit the new possibilities and work around the weaknesses of distance learning. Our experience, however, shows that the pedagogical content is by no means reduced if we can use technology to support a problem-based learning context, including a forum allowing the students to discuss and understand the material. Is it reasonable to expect students to pay for courses if the material is readily accessible on the Internet? Our experience shows that those students who can attend classes locally or through videoconference still do so for the stimulation and discussion they get from the teacher and peers. Those who cannot attend strongly value the flexibility to deliver their work when and where they like, and they still benefit from close personal supervision from a teacher. CiSE

Acknowledgments
The Summer University of Southern Stockholm and Ericsson partly funded this work.

References

1. A. Jaun, J. Hedin, and T. Johnson1, Numerical Methods for Partial Differential Equations, TRITA-ALF-1999-05 (1999), http://pde.fusion.kth.se
2. N. Drakos, Text to Hypertext conversion with LaTeX2HTML, Baskerville (1993) 12, http://cbl.leeds.ac.uk/nikos/tex2html/doc/latex2html/latex2html.html
3. L. Hutchinson, TTH: a TeX to HTML translator. http://hutchinson.belmont.ma.us/tth

André Jaun is an assistant professor in computational physics at the Alfvén Laboratory of the Royal Institute of Technology in Stockholm. His technical interest include the modeling of magneto-hydrodynamic fluids, the heating and stability of plasmas and the development of educational methods for distance education. He belongs to the American Physical Society and the Swiss Alpine Club. Contact him at jaun@fusion.kth.se.

Johan Hedin is a senior system architecture developer of software for health care at eCare AB in Stockholm. He recently acquired his Ph.D. in fusion plasma physics. His technical interests include the radio frequency heating of plasmas, parallel computing, Monte-Carlo simulations and object oriented methods for distributed environments. Contact him at johan@eCare.se.

Thomas Johnson is a graduate student in fusion plasma physics at the Alfvén Laboratory of the Royal Institute of Technology in Stockholm. His technical interests are the numerical modeling of radio frequency heating and current drive in tokamaks. Contact him at thomasj@fusion.kth.se.