Collaborative Initiatives in the Distance Education of Applied Mathematics

A. Jaun[1], P. Cotton[2], T. Hurtig[1], T. Johnson[1], T. Rylander[3], N. Stainier[4], L. Villard[4]

[1] Royal Institute of Technology, 100 44, Stockholm, Sweden

[2] Morgan Stanley, 1585 Broadway, New York, NY 10036-8293, USA

[3] University of Illinois, Urbana, IL 61801-2991, USA

[4] Ecole Polytechnique Fédérale, 1015 Lausanne, Switzerland

E-mail of main author: jaun@kth,se

Abstract. A web based technology for the distance education of applied mathematics has been used to share courses in financial engineering and numerical analysis. Remote participation occurred through virtual classrooms, with a network of 5 teachers from 4 universities and firms who advised up to 29 students working simultaneously in 3 different countries. The strength of the method resides in a problem-based learning environment that uses numerical experiments in the syllabus and assignments where the students explain with their own words, algebra and Java programs how to solve practical problems.

1 Introduction

Starting from a graduate course taught in 1997 using a problem based learning methodology, six years trials and developments make it possible today to offer courses in numerical analysis [1] and financial engineering [2] for a relatively large number of students from the Swedish Netuniversity—in a manner that compares favorably with traditional classes. Apart from the expected gain allowing a flexible learning at any time and from everywhere on the Internet, the shift from a classroom to the digital world creates new opportunities to involve sister universities and specialists from the corporate world: in the last edition of the course *Numerical methods for partial differential equations* [1], a network of 5 teachers from 4 universities and firms for example advised 29 students working simultaneously in 3 different countries.

This paper report on our experience and after a short review in section 2 of the motivation behind the developments, section 3 describes the instructional tools that have been created to support student-teacher interactions in virtual classrooms. Section 4 deals with the organizational and pedagogical aspects, showing how the same technology has been used in different manners: intensive summer schools taught locally in Stockholm, continued education courses shared with sister universities and a lifelong learning on the Internet.

2 Background and motivation

With the spreading of computers and information technology, students increasingly expect an education that is illustrated with practical and even realistic examples. The *problem based learning* (PBL) methodology is in principle an excellent answer; the large burden of supervising this form of education, however, makes it impractical even for middle sized classes with 30 participants. Much smaller classes are generally taught independently by neighboring departments, often in a conventional manner and at a relatively high cost.

Distance learning technologies can be used to create virtual classrooms with students from departments and schools that are located in different countries. Well prepared material and partly automated supervision from a server could make the problem based learning more attractive for larger classes. Pooling students from various disciplines has a tremendous development potential in terms of opening the scientific horizon and building bridges across cultural barriers. Teachers from sister universities having various research interests add new colors to the subject and as a group are more available to answer questions quickly. Professionals can exploit the flexibility of distance learning to participate either as students or teachers, in an education that becomes more relevant for a continued education outside academia.

From our experience, a major difficulty with a problem based learning at a distance, is to provide the right amount of guidance to a large number of students when it is needed rapidly. An important issue is what can be done to avoid the frustration of learners who are left without a clue, get side-tracked or lost, while others easily find the solution. Another issue for courses in applied mathematics is the question of showing students who are not familiar with LaTeX or Java, simple ways to write mathematical formulas and programs without spending several lessons on a technology that is unrelated with the subject of the course.

The *solutions templates* we proposed two years ago [3] could be an excellent way to provide hints, using milestones and cross-checks to guide students through realistic problems that usually require a number of steps to reach the solution. Performing simple modifications of the formulas that are initially provided in the solution templates may be sufficient to acquire the technology simply through the context.

Although it is premature to draw conclusion about the cost efficiency of such a problem based learning at a distance, this article will show that part of the vision described above has indeed been reproduced in middle-sized classes and with a high degree of satisfaction from the students.

3 Instructional technology

Most of the technology has been designed in collaboration with the authors, using a distance learning software that is developed, maintained and distributed by Lifelong-Learners [4]. Convinced by the value of an open learning for the society at large, a regular web browser (Netscape, Explorer) and a free video player (RealVideo) are used to visualize all the material from the course web sites [1, 2]. Student specific data (assignments, solutions, messages, etc) and supervision (corrections, advice, assessment, etc) are restricted for registered users; fortunately, the former have so far always been sponsored, making the course environment available free of charge for regular students from sister universities.

Here is a list of educational tools that can be combined in different manners to suit the work style from different students.

3.1 Open learning area

The syllabus with an automatic form of supervision can be accessed directly on-line, or downloaded and studied off-line to limit the connection time with a modem.

Java-powered syllabus. More than a hyper-linked document from conventional class, the syllabi of both courses have been redesigned to take advantage of Java applets. Relatively advanced topics such as the evolution of stock market prices can be described in very simple terms using simulations. Figure 1 shows an example of a web page combining hyper-linked text, mathematical formulas and the JBONE applet illustrating a random walk. Numerous experiments are suggested and are performed by changing editable parameters in the applet. They serve not only to verify the understanding, but also to develop an intuition for the subject. Modifications carried out by the learners themselves are one of the best form of automatic supervision that can be provided at a distance.

Computer quiz. Questions with a multiple choice of predefined answers have been automated and can be corrected simply by following a link back into the syllabus. This simpler form of automatic supervision ensures that the learner correctly read and understood the syllabus.

Student reports. Small projects, representing of the order of a full-time week of work, are sometimes carried out at the end of graduate courses to apply the knowledge to specific areas of interest. Some of the reports are of sufficient quality and interest that we now publish them on the course web site.

3.2 Registered users area

The students need to register and identify themselves through a login in order to tailor the course material to the right level, store individual solutions in the form of web pages and benefit from the supervision of a human teacher.

Student profile. Every learner starts his studies with peers in a virtual classroom, where the level of difficulty, the assignments and course requirements are fixed once for all. It is indeed important to allow for a continuous improvement of the course material without disrupting the work of students who sometimes take several years to fulfill the requirements.

Levels of difficulty. Harnessing the power of numerical simulations in financial engineering [2], a new syllabus has been designed to explain the stock option payoff dynamics both to engineers who are familiar with advanced mathematics and to financiers who generally prefer to avoid formulas. The syllabus and the assignments are tailored according to the student profile, provide questions and explanations at the chosen level of difficulty.

RealVideo recordings. Hand-held digital cameras and editing software make it easy now to produce amateur recordings of lectures and to compress them in RealVideo format. The level of professionalism everybody is used to from TV makes the experience as painful for the teacher as it is funny for the student... Fact is that a majority watch the video recordings to complement the readings from the syllabus. Video-conferences have been used to various degrees of satisfaction for joint sessions with Stockholm, Gothenburg and Lausanne; rather than to replace regular lectures, a video-conference may be useful at the beginning of a course to create social contacts to demonstrate the possibilities and the limitations of interacting at a distance.

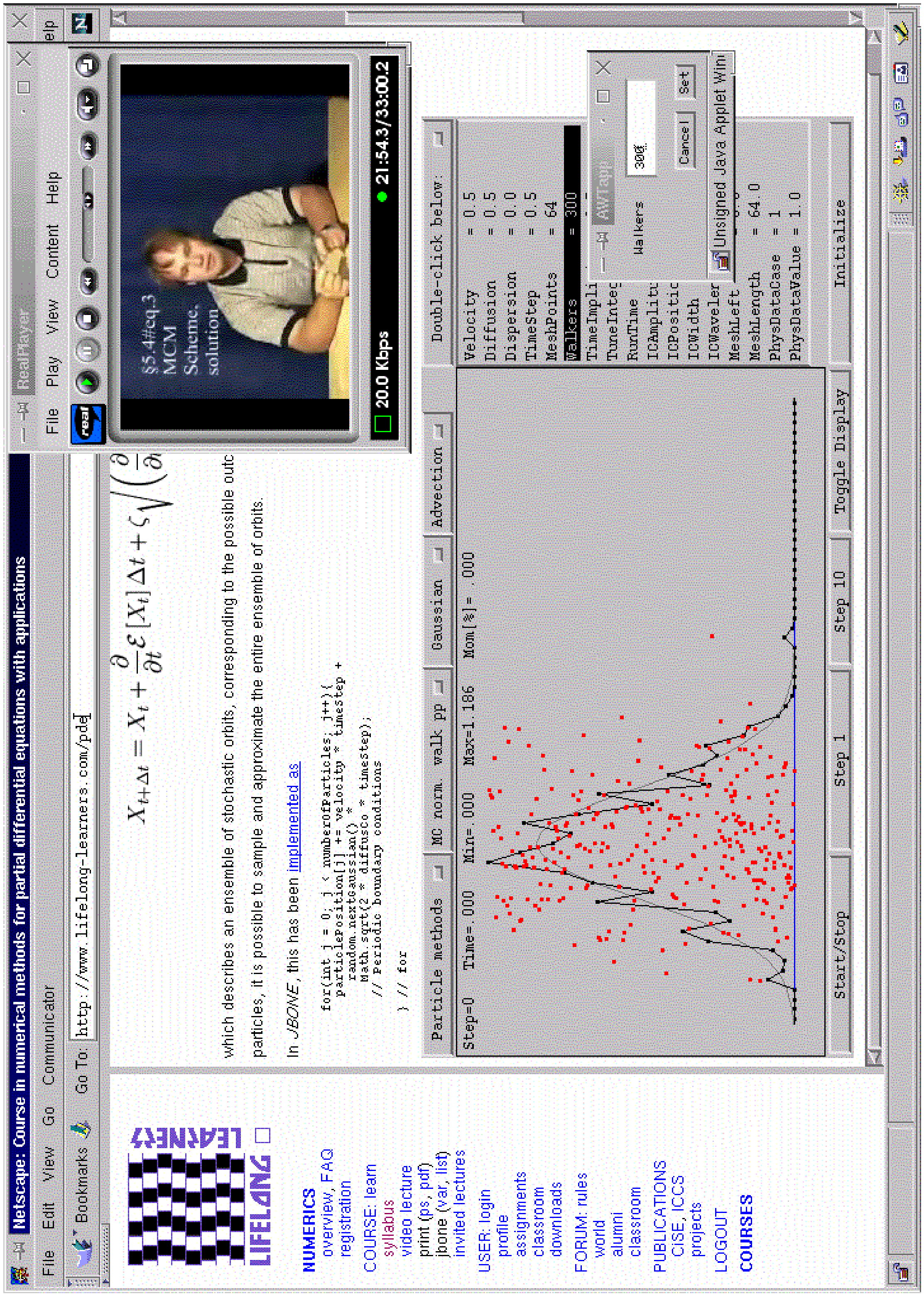

Figure 1. Web browser displaying hyper-linked text, formulas and numerical experiments using editable parameters in the JBONE applet—reproducing arguments that are discussed in the RealVideo lecture.

Assignments are at the core of any problem based learning. They vary both in scope and difficulty, but typically require 1-2 days each to work through a whole series of steps that foster a practical understanding of the subject. Starting from a physical model, the students derive a mathematical formulation using LaTeX to write formulas, implement their numerical scheme in Java, choose relevant parameters and discuss the simulations that are visualized in their own applet. Figure 2 (back) shows how text areas appear below the handout: these can be submitted from a regular browser and are automatically compiled into Java-powered solution web pages. Conservation properties are monitored in the applet to signal potential mistakes and prove to be a very powerful form of automatic supervision.

Templates. Rather than starting derivations from scratch with empty windows, we now provide guidance using editable examples and let the students cross-check intermediate steps. With a clever design, such templates greatly reduce the amount of questions that are difficult to answer rapidly in large classes, limit the frustration of students who are stuck without killing the creativity of those who don't need them. At the beginning of the course, simple modifications of LaTeX and Java templates are incredibly effective to show students with very little programming experience, how to use these technologies.

Corrections. Although a part can be pre-programmed and delivered automatically at any time, the best form of supervision requires personal contact with a human teacher, usually within a day or two. Since most of the details have already been corrected, the teacher can focus on the global picture, open the horizon with questions for the best students and spend more time explaining the mistakes of the weakest. This interaction occurs entirely through web pages and can be provided from anywhere on the Internet; it is therefore possible to involve several teachers from sister universities in the same virtual classroom.

Virtual classrooms. To create a feeling of belonging to the same class and develop human contacts with peers, the name, affiliation and a picture of each student is displayed in the classroom area. Eventually, the students get to know each other through forum discussions and the links to the best solutions, which we usually share with everyone who finishes a certain assignment. This creates a healthy form of competition where every student tries hard to appear on a very exclusive list.

Discussion forums. Questions and answers from peers are usually quick and competent; we therefore reward them with bonus points in the same manner as the assignments. A limited amount of supervision is required to ensure that everything that is discussed is correct, but the light shed from different angles is usually precious and documents difficulties that can be addressed in subsequent editions of the course.

Download. Keeping the solutions accessible anywhere from the Internet proves to be useful when moving abroad and it is very rewarding for the teacher to see how former students occasionally log back to their account. Reassembling all the LaTeX inputs into a printable document and inserting the Java schemes into the source code, students can download and document their work independently from the server.

Evaluation. The simplicity of filling-in and collecting electronic forms makes it possible not only to evaluate the entire course at the end, but also to optimize the path leading there as opinions changes from chapter to chapter.

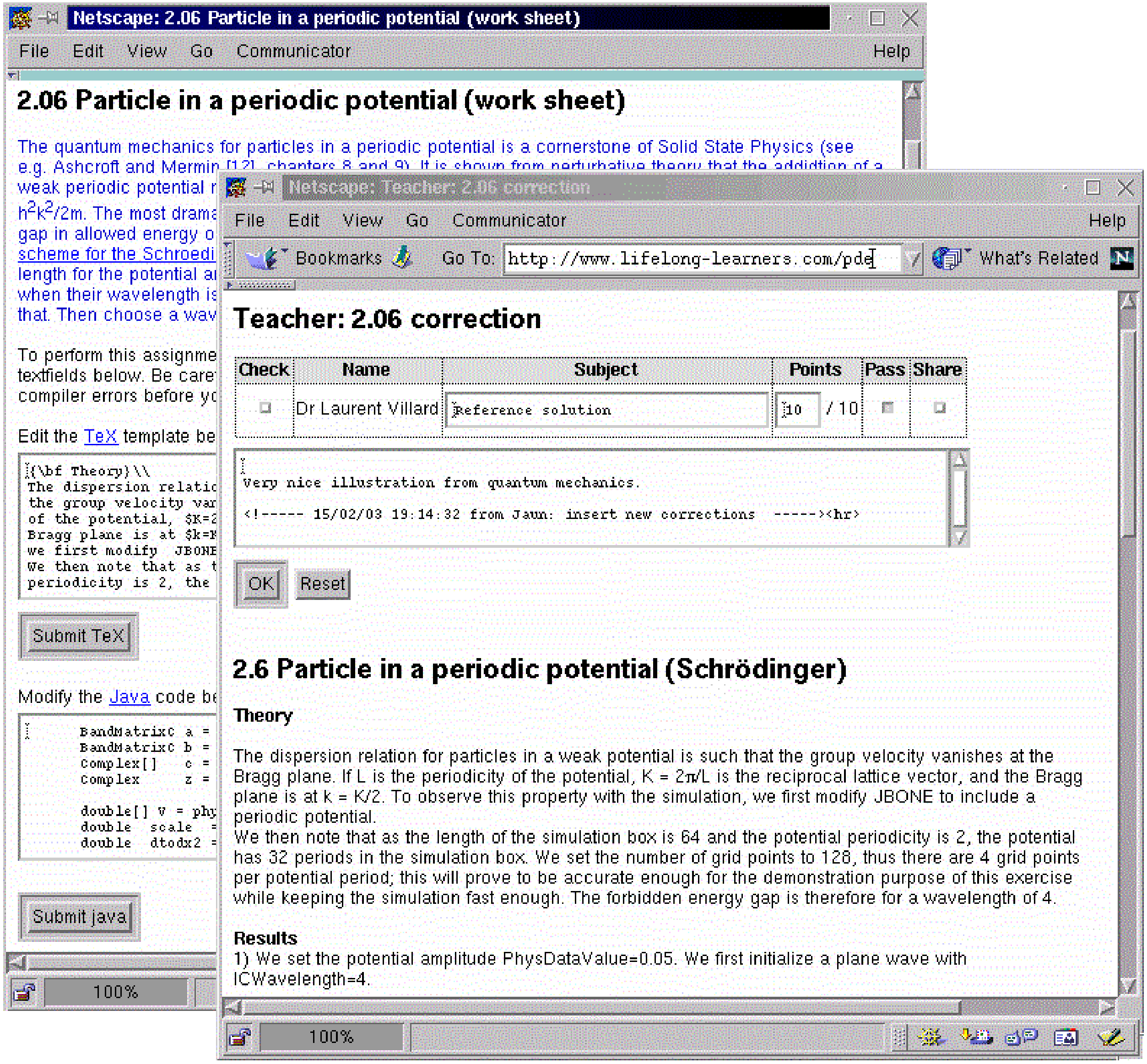

Figure 2. The window in the back illustrates how the assignments are carried out starting from a handout (top): gradual modifications of templates show the students how to submit their own analysis combining text, formulas (TeX), program subroutines (Java), run parameters and pictures. The window in the front illustrates how teachers visualize the Java-powered solutions (bottom), correct potential mistakes, evaluate and finally accepts the solution (top).

3.3 Teacher area and course development

Our experience shows that moving from a conventional classroom to a distance-learning setup is best performed gradually, using 1-2 months developments for each step: modify an existing applet for an early version of the Java-powered syllabus during the first year, add a few web exercises for local students in the second edition and take a limited number of remote learners in the third. The software from Lifelong-Learners [4] has for example been tailored in this manner to gradually replace the course in numerical analysis with material in financial engineering.

Syllabus source code. A single enriched LaTeX source produces the printed and the web edition of the syllabus, the computer quiz and the assignments. A `Makefile` updates the material by calling the `latex2html` and `tth` compilers [5, 6] and embedding all the modifications in the course environment.

Applet source code. The advantage of modifying an older applet such as JBONE is that with age, the applet has been tested and finally run on most of the web browsers. Our experience shows that in spite of being a standard, Java behaves differently on browsers and operating systems that have each its own versions and bugs. A `Makefile` updates the executable and the documentation calling the `javac`, `javadoc` compilers and publishes them on the web.

Assignments and templates. New templates are easily created and maintained for consecutive editions of the same course, using the experience from previous years to gradually improve the automatic supervision at a distance.

Monitoring the progress. Having the possibility to check all the solutions at any time, it becomes much easier to monitor the progress of 30 participants at a distance than it is to follow the progress from a conventional class locally. An accurate monitoring is often perceived as motivating by the students who notice that the teacher cares; it also allows the teacher to detected potential difficulties before it is too late, helping to reducing the large dropout rates that often characterize distance-learning courses.

3.4 Administration and server

While the students can access all the material from all the most common operating systems (Windows, UNIX, MacOS, etc), exclusively open source software has been used to serve the course from a Linux workstation.

Hardware requirements. Even a modest PC connected to 10 MBits/sec network is sufficient to handle 30 students working simultaneously in an intensive course. Relying heavily on the server availability, it is however safe to install a backup server on a different network.

Software requirements includes an Apache server featuring a PHP parser and an interface to a Postgres SQL database. The latex2html, tth compilers have to be installed to support mathematical formulas and javadoc, javac to compile the applets. Most of the software is usually included in recent Linux distributions.

Installation and maintenance. After decompressing the archive and tailoring the configuration file, a `Makefile` is used to perform the entire installation: create and populate a database, install the templates and the compilers, compile the syllabus and the applet source codes, publish it all on the web. Backups and maintenance of the student accounts are performed in the same manner and can be scheduled for automatic execution every day.

Source code. Full access is given to the source code, allowing other teachers to tailor the software and develop it further for specific needs. Dynamic web pages are created with PHP scripts that can simply be edited.

4 Teaching experience in virtual classrooms

A variety of setups have been tried as the course gradually evolved from a problem based learning offered to five students in Stockholm to a full scale distance-learning for up to fifty students from the Swedish Netuniversity and independent learners on the Internet. We here only discuss experience acquired from courses with at least one half of remote participants.

Summer University 2000-2001 (KTH, EPFL, CTH, UU). Three weeks intensive courses in numerical methods [1] have been organized twice in a partnership with four sister universities: the Royal Institute of Technology in Stockholm (KTH, 16 students in 2001), the Swiss Federal Institute of Technology in Lausanne (EPFL, 6), the Chalmers Institute of Technology in Göteborg (CTH, 5) and the Uppsala University (UU, 2), under the patronage of the Summer University of Southern Stockholm (SUSS).

On both occasions, the course was kicked off with an e-mail sent to the schools mailing lists; a link to the Java-powered syllabus proved to be an excellent advertisement that has also been forwarded to colleagues. After registration, a second message sent to the participants described the organization of the event and tried to resolve ahead of time most of the difficulties that can easily disturb the beginning of a course: every student verified the browser capability of displaying both applets and formulas, bookmarked the address of backup servers and printed a paper version of the syllabus.

A video conference was used for the first lecture and later, with various degrees of satisfaction, for optional tutorials. The primary achievement was probably less pedagogical than to reassure the teachers, when it was not entirely clear how well the software would support the interactions with students that were thousands of kilometers apart. Figure 3 shows that RealVideo recordings have been successfully introduced in 2001 along with the rest of the distance-learning software. It is likely that we will not use much video-conferencing anymore in 2003.

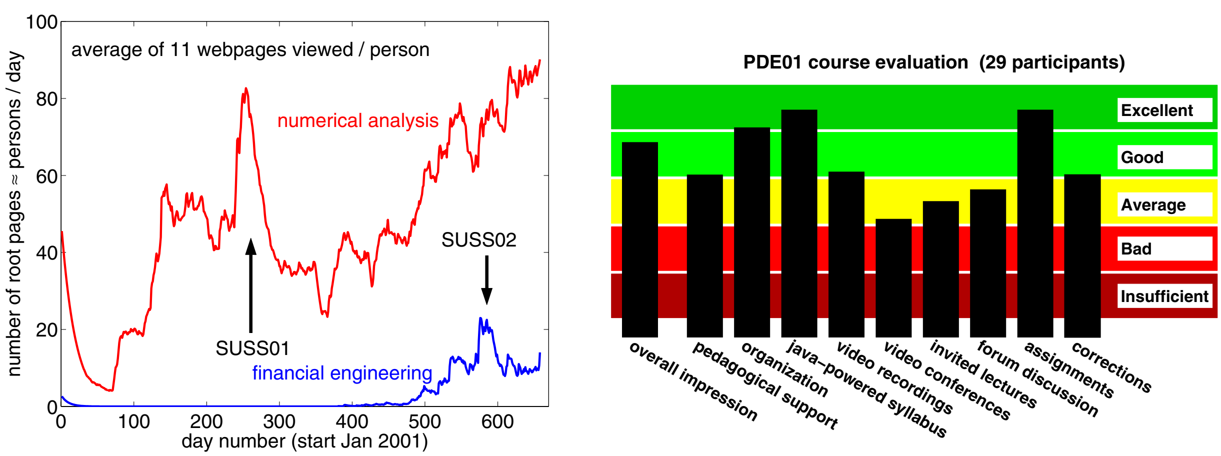

Figure 3. Growing web site traffic with two peaks during the two Summer University courses (left) and the evaluation from students who participated from four universities in 2001 (right).

Previous experience justified our relatively high expectations from the problem based learning environment; templates, however, have been first introduced in 2001, hoping to improve the guidance and to avoid having to answer the same questions every year in the users forum. While the desired effect has partly been observed, further improvements in the templates will probably have a much larger long term impact as the courses gains maturity.

With light speed technology now taking care of most of the operations—it took several minutes back in 1997 to edit a file, compile and save the result from a simulation manually, an operation that has now been reduced to less than 3 seconds—we were surprised to hear that the students become increasingly demanding: it was suggested that we replace the standard Java compiler by a quicker one to gain another second! On a more serious tone, two students complained for having to wait as much as 12 hours to get back a correction from the teachers. Of course, both the students and the teachers knew that it usually takes a week before a limited number of exercises are corrected in a conventional setup with 30 students. Nevertheless, such feed-back puts pressure on the teachers who face an additional disappointment in figure 3, when they find out that the personal efforts they make for real-time human support (forum discussions, corrections) is finally rated below the pre-programmed answers from the computer.

Open learning 2001-2003. The Java-powered syllabus has been shared with learners everywhere on the Internet and the web site traffic in figure 3 shows that nearly one hundred persons now connect to the course every day. Clearly, every surfer is not necessarily a learner with an average of 11 pages viewed per person. Nevertheless, the benefit is not negligible with more than 500 users who registered for a personal account (free of charge for academics) and we know from encouragements, that a dozen colleagues at least use our material for their own classes. Private lessons have been offered on a private basis, as part of the activity teachers are allowed to pursue outside their professional duties.

Swedish Netuniversity 2003. Both courses will be held for the first time under an national agreement where every student from a Swedish university can learn at a distance, obtain a formal accreditation, while the hosting institution gets rewarded financially for this service.

5 Conclusions

Open source software has been used to hold distance-learning courses in numerical analysis and financial engineering, in a manner that compares favorably with traditional classes. Teachers from different universities and firms provided human support for 29 students from geographically remote locations, adding colors to the material distributed anytime and everywhere on the Internet. A problem based learning proves to be effective for middle-sized classes, when templates and an automatic form of supervision can be used to guide the students from previous experience. This highlights the importance of gradual improvements in the software, where teachers adapt the pre-programmed material to gradually supervise more students.

Acknowledgements

This work is supported in part by the Summer University of Southern Stockholm, Morgan Stanley, Lifelong-Learners and the Swiss National Science Foundation.

References

[1] Swedish Netuniversity course *Numerical Methods for Partial Differential Equations*, http://www.lifelong-learners.com/pde/nu

[2] Swedish Netuniversity course *Hedging your Portfolio: Options, Swaps and Derivatives*, http://www.lifelong-learners.com/opt/nu

[3] A. Jaun, T. Johanson, J. Hedin, *Teaching Numerical Methods for PDEs with the Internet*, Comput. Sci. Eng. 3 (2001) 83

[4] Software distributed by http://www.lifelong-learners.com

[5] N. Drakos, *Text to Hypertext conversion with LaTeX2HTML*, Baskerville 3 (1993) 12 http://cbl.leeds.ac.uk/nikos/tex2html/doc/latex2html/latex2html.html

[6] I. Hutchinson, *TTH: a TeX to HTML translator*, http://hutchinson.belmont.ma.us/tth

Biographies

André Jaun is an Associate Professor of Computational Physics at NADA at the Royal Institute of Technology. He teaches conventional courses in Stockholm, distance-learning courses for the Swedish Netuniversity and open-learning courses on the Internet. His research interests span from the numerical modeling in fusion energy sciences to the pricing of options for the stock market.
Peter Cotton earned his Ph.D. in Mathematics from Stanford University and is now working with the modeling of Credit Derivatives for Morgan Stanley. He participated as an external lecturer in the distance-learning courses.
Tomas Hurtig is a Ph.D. student at the Alfvén Laboratory at the Royal Institute of Technology. He has been an assistant for the distance-learning of both numerical methods and financial engineering. His research interests include numerical modeling and experimental plasma physics.
Thomas Johnsson is a Ph.D. student at the Alfvén Laboratory at the Royal Institute of Technology. He has been an assistant twice for the distance-learning of numerical methods. His research is focused on the numerical modeling of ion cyclotron resonance heating in fusion plasmas.
Thomas Rylander is a post-doctoral associate at the Center for Computational Electromagnetics, University of Illinois at Urbana-Champaign. He has been an assistant for the distance-learning of numerical methods. His research focuses on finite element methods in electromagnetics.
Nadine Stainier is a pedagogical adviser at the Center for Research and Support of Training and its Technologies at the Swiss Federal Institute of Technology in Lausanne, where she assists educational staff for teaching quality and innovations.
Laurent Villard is a senior scientist at the Centre de Recherches en Physique des Plasmas at the Swiss Federal Institute of Technology in Lausanne. Besides his research in numerical simulations of instabilities and turbulence in magnetically confined plasmas, he teaches undergraduate general physics and was responsible for the Swiss node in the collaborative teaching of numerical methods.