UCES: An Undergraduate CSE Initiative

Thomas L. Marchioro II, David M. Martin, and W. Donald Payne

In 1993 the Applied Mathematical Sciences division of Ames Laboratory, with funding from the US Department of Energy, launched an educational initiative to improve the teaching of computational science and engineering at the undergraduate level. Parallel initiatives had already been created to strengthen computational science education at the kindergarten through 12th-grade level (Adventures in Supercomputing[1]) and at the graduate level (the Computational Science Education Project[2–4] and the Computational Science Graduate Fellowship Program[5]). All of these projects seek to unite teachers of CSE in a collaborative effort to develop effective, innovative instructional materials, many of which have recently been distributed in hypertext form on the World Wide Web.

With the general aim of promoting computational science approaches to teaching scientific analysis, the Undergraduate Computational Engineering and Science (UCES) Project began a series of formative meetings and workshops. These planning and strategy sessions brought together a wide range of individuals from academia, government research laboratories, and industry; people in mathematical, scientific, and engineering fields interested in pursuing innovative approaches to developing and implementing instructional CSE materials. Our discussions coalesced around three objectives:

(1) forming a cohesive curricular approach to computational science;
(2) identifying and strengthening a computational science and engineering community; and
(3) developing a readily accessible archive of instructional materials in CSE, including authoring tools that allow instructors to tailor materials to their specific needs.

UCES has begun addressing each of these goals and coordinating activities in support of them.

The UCES approach to CSE education

Computational science and engineering is at once both marginal and fundamental to the traditional disciplines that it touches. Marginal because CSE is not a "discipline" in the classic sense—nor, in the UCES view, should it aspire to disciplinary status. Unlike some subject areas—mathematics and computer science, for instance—the role of CSE is not as a body of knowledge but as a methodology, a set of strategies, a problem-solving perspective. CSE should not supplant traditional disciplines, but complement and buttress them. Nonetheless, CSE is fundamental precisely in the way that methodologies are fundamental, fused inevitably to certain philosophical approaches that shape how problems are viewed as well as solved.

CSE is not, of course, a clearly identifiable community[6] or a univocal interdisciplinary movement. Sometimes its identity seems to center in a loose cluster of interests such as visualization, computational error analysis, computer networking, and computational tools for numerical and symbolic analysis. At least these are the most visible areas of activity in a broad attempt to realize two of the goals laid out in SIAM's 1987 National Computing Initiative:

- "support for computational science and engineering research in the interdisciplinary mode, combining applied mathematics, computational science, and scientific computing applied to fundamental problems in science and engineering" and
- "innovative undergraduate and graduate curricula to educate engineers, scientists, economists, and so on in the effective use of advanced computing technology."[7]

Furthermore, the report noted: "In many areas of science and engineering, advanced computation methods are becoming as important as experimental and analytical methods, yet few curricula include even basic courses in computational mathematics. While universities are beginning to gain access to supercomputers through federally sponsored programs, they still need to be encouraged to develop courses and curricula in this area."[7]

At the undergraduate level this philosophy means that CSE fundamentals must be integrated into the curriculum, not merely added as a new course or set of courses. At a practical level, adding more courses to already overextended curricular requirements is not feasible in most programs. At a philosophical level, UCES views problem-centered strategies for teaching CSE as integrative and synthesizing. At the very least such strategies need to be incorporated into existing courses, but many programs can go a step further to replace certain existing computer-based courses with interdisciplinary counterparts built on a problem-model-assessment pedagogy.

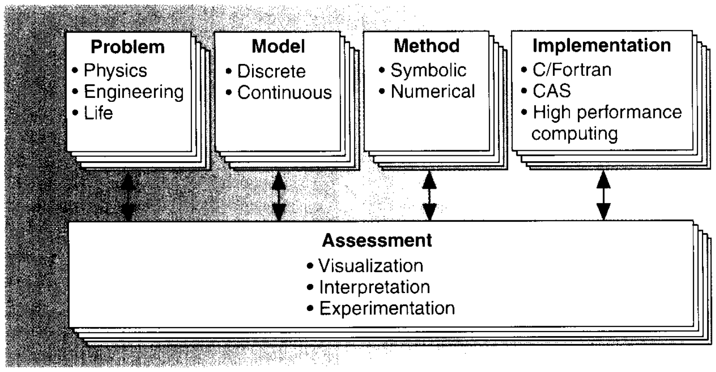

Figure 1. The UCES paradigm of CSE education.

Building curriculum

Those assembled at the first UCES meeting agreed that the analytical and problem-solving skills associated with computational science were becoming foundational to many scientific and engineering disciplines. Students needed these skills for both their academic and workplace careers, yet those skills were not often addressed in the curriculum in a clear and focused way. With that in mind UCES members decided to concentrate their initial efforts on three tasks: analyzing successful CSE programs, shaping a basic CSE curriculum, and deriving a general model for teaching computational concepts through problem-based modules. At the curricular level, they identified several key areas for course development: a general introductory course in scientific computation at the freshman level, a more advanced course in scientific programming and/or introductory numerical analysis at the sophomore level, and selected courses for juniors and seniors, such as courses in parallel computing and computational physics.

With the freshman course as the first goal, the group agreed to develop a set of basic concepts along with strategies for teaching these concepts. The aim was to produce a series of modules that could be used either independently or linked in a single, coherent course. Materials needed to be flexible if teachers in various institu-

Breast Cancer Diagnosis via Linear Programming

Michael C. Ferris and O.L. Mangasarian, University of Wisconsin

Linear programming, used in many business and operations research applications, is a technique for finding an optimum combination from among many feasible combinations of variables. In this project, which is intended for junior- and senior-level undergraduates, we show the students how linear programming can be used to diagnose breast cancer, based on experience gained in building a computer-based diagnostic system that is currently in use at University of Wisconsin Hospitals.[1,2] The project was developed in conjunction with a new computational book we are writing on linear programming with Matlab and has been used several times for teaching purposes at the University of Wisconsin.

The project uses the Wisconsin Breast Cancer Database, made public by William H. Wolberg of the University of Wisconsin Hospitals and available from the University of California–Irvine Repository of Machine-Learning Databases (ftp.ics.uci.edu/pub/machine-learning-databases/breast-cancer-wisconsin). Each data point in this database represents a quantification as a nine-dimensional vector[3,4] of a confirmed benign or malignant fine needle aspirate (FNA). Magnified slides of benign (Figure A) and malignant (Figure B) FNAs are shown.

The database has one line per sample, each with 11 attributes determined by the oncologist from the FNA: 1–sample code number; 2–class (2 for benign, 4 for malignant); 3–clump thickness; 4–uniformity of cell size; 5–uniformity of cell shape; 6–marginal adhesion; 7–single epithelial cell size; 8–bare nuclei; 9–bland chromatin; 10–normal nucleoli; 11–mitoses. For purposes of the project, the data have been split into a training set and a testing set. For both of these, the student uses attributes 3 to 11 (each of which takes on a value between 0 and 10) to form a nine-dimensional vector representing each case as a point in nine-dimensional real space.

In generating the diagnostic program, the student begins with the training set consisting of two disjoint point sets (benign and malignant) in $\mathbb{R}^9$ and generates a discriminant function that distinguishes between the two sets. This function is subsequently used on the testing set to determine whether new aspirates are benign or malignant.

The discriminant function is determined by solving a single linear program.[1,5] The size of the database requires a reasonably sophisticated linear-programming code that takes into account key numerical ideas from linear algebra and mathematical programming; these ideas are crucial for this application of computational science to medical diagnosis.

The discriminant function $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} - \gamma$ has the property $f(\mathbf{x}) \geq 0$ if $\mathbf{x}$ is malignant, $f(\mathbf{x}) < 0$ if $\mathbf{x}$ is benign. This determines a plane $\mathbf{w}^T\mathbf{x} = \gamma$ that separates malignant and benign points in $\mathbb{R}^9$. The variables $\mathbf{w}$ and γ are found by minimizing the sum of the distances to the dividing plane whenever a point is on the incorrect side of the plane. Using standard techniques, this can be expressed as the following linear program:

$$\min_{\mathbf{w},\gamma,\mathbf{y},\mathbf{z}} \left\{ \frac{1}{m}\mathbf{e}^T\mathbf{y} + \frac{1}{k}\mathbf{e}^T\mathbf{z} \,\middle|\, \boldsymbol{M}\mathbf{w} - \mathbf{e}\gamma + \mathbf{y} \geq \mathbf{e}, \right.$$
$$\left. -\boldsymbol{B}\mathbf{w} + \mathbf{e}\gamma + \mathbf{z} \geq \mathbf{e}, \mathbf{y} \geq 0, \mathbf{z} \geq 0 \right\}$$

Here $\mathbf{e}$ is an appropriately dimensioned vector of 1's, $\mathbf{y}$ and $\mathbf{z}$ are error vectors, and m and k are the number of malignant and benign points which are stored as matrices $\boldsymbol{M}$ and $\boldsymbol{B}$. The solution determines the plane, and hence the discriminant function.

To implement this project in the classroom, three things are required:

- the data for the project, obtained by anonymous FTP from ftp.cs.wisc.edu: math-prog/teaching;

tional settings were to integrate them into their curricula. To present these modules clearly to students, it was decided to structure them around five basic components: problem, model, method, implementation, and assessment (see Figure 1). This UCES paradigm stresses the journey from physical problem to computational solution, emphasizing assessment of all results, often through visualization.

The UCES paradigm is not meant to be restrictive, and some materials (such as background discussions on Taylor series and root finding) deviate from it markedly, but it provides a cohesive overall approach that is consistent with the philosophy just delineated. In the case of the introductory course, the UCES paradigm has led to a course in "scientific problem-solving with a computer" as a replacement for the traditional introductory course in procedural programming that often does not provide students with a problem-solving context.

Paralleling the discussions of content and approach were practical questions about appropriate computational tools, including software packages such as Matlab, Maple, and Mathematica. What common tools could form a reliable environment, yet one adaptable to diverse instructional sites? UCES initially selected Maple and Mathematica, with support for other programs to be added later. A parallel discussion focused on the best delivery system for UCES materials. Although gopher, FTP, and e-mail were viewed as possible delivery mechanisms, the development of the World Wide Web and user-friendly browsers in 1994 eventually made the Web the primary delivery mechanism, and the evolving Hypertext Markup Language (HTML) standard the choice for formatting. Since the fundamental goal of UCES is to effect widespread cultural changes in computational education, we focused on the Web as the primary communication mechanism.[8] Using the Web allows UCES to "spread the word" both to our primary audience and to the "general public" of the Internet. Our educational materials are written for this type of electronic publication; print or hybrid versions, while planned in some cases, are considered secondary. In addition to professional participants, student assistants have helped with processing materials and creating tools for interactive courseware development. We envision a future in which electronic dissemination plays an important, perhaps dominant, role in education, and hope to demonstrate the value of the developing digital communication infrastructure through high-quality electronic publishing.

Building community

UCES has helped to solidify the CSE community through its many workshops

---

- a linear programming code (publicly available or an implementation of a version developed in the course); and
- simple plotting routines.

We use the Matlab numerical package for all of the above in our teaching. The data can be loaded easily, a satisfactory linear-programming code is easily implemented, and the built-in plotting routines are more than adequate.

Students are evaluated on the following assignments:

(1) Formulate the problem as a linear program. Solve the problem using the matrices ***M*** and ***B***, which can be determined from the training.dat file, with field 2 determining whether the sample is benign or malignant. Determine how good the solution is by counting the number of misclassified points on the training set.

(2) Test out your discriminant function on the testing set. Determine the number of misclassified points on the testing set.

(3) Suppose that the oncologist wants to use only two of the nine attributes for diagnosis. Determine which pair of attributes is most effective in determining a correct diagnosis, again by using only the training set (with these two features) to determine the discriminant function and then using the testing set to determine the number of misclassified points. Print out the number of misclassified points for each combination.

(4) Using the best answer from (3) above, plot all the testing points on a 2D figure using Matlab's built-in plotting routines. Use "o" for benign points and "+" for malignant points in your plot. Then use Matlab to draw in the calculated line $\mathbf{w}^T\mathbf{x} = \gamma$. Check to see if the number of misclassified points agrees with your plot.

Michael C. Ferris and O.L. Mangasarian teach in the Computer Science Department at the University of Wisconsin. They can be reached at 1210 West Dayton Street, Madison, WI 53706; e-mail, {ferris, olvi}@cs.wisc.edu.

Figure A. Benign cells.

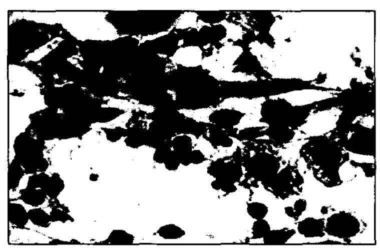

Figure B. Malignant cells.

References

1. O.L. Mangasarian, W.N. Street, and W.H. Wolberg, "Breast Cancer Diagnosis and Prognosis via Linear Programming," *Operations Research*, Vol. 43, 1995, pp. 570–577.
2. W.H. Wolberg, W.N. Street, and O.L. Mangasarian, "Image Analysis and Machine Learning Applied to Breast Cancer Diagnosis and Prognosis," *Analytical and Quantitative Cytology and Histology*, Vol. 17, No. 2, Apr. 1995, pp. 77–87.
3. W.H. Wolberg and O.L. Mangasarian, "Multisurface Method of Pattern Separation for Medical Diagnosis Applied to Breast Cytology," *Proc. Nat'l Academy of Sciences, U.S.A.*, Vol. 87, 1990, pp. 9,193–9,196.
4. W.H. Wolberg, W.N. Street, and O.L. Mangasarian, "Breast Cytology Diagnosis via Digital Image Analysis," *Analytical and Quantitative Cytology and Histology*, Vol. 15, No. 6, Dec. 1993, pp. 396–404.
5. K.P. Bennett and O.L. Mangasarian, "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets," *Optimization Methods and Software*, Vol. 1, 1992, pp. 23–34.

and collaborative efforts, and through its professional and student recognition programs. The need for such project activities was determined in part by the special nature of CSE education in 1993. It was still a new, only partly delineated field, one growing rapidly across disciplines in potentially exciting but unpredictable ways. While a stronger sense of community has been a valuable by-product of UCES efforts, a more direct and visible step in promoting computational science was needed. To this end UCES instituted a pair of awards programs.

In 1994 UCES initiated a national Undergraduate Computational Science Award program to recognize and reward educators who have achieved excellence in CSE education (see the sidebar on pp. 70–71 for an example of an award-winning project). The awards program also serves to locate outstanding instructional projects that could be added to the UCES educational archive. Cash awards were presented for thirteen submissions, and four others were granted honorable mention. All winners (and the honorable mentions) were the guests of the DoE at an awards exposition and banquet in Washington, D.C., including an "electronic poster session" at which the entries were displayed. This gathering provided an opportunity for teachers from various disciplines and institutions to focus on the computational science aspects of their instruction and to be invigorated by the diversity of projects represented. Topics drawn from various disciplines focused on problems with computation-based solutions. A second annual Undergraduate Computational Science Award program and banquet was held this year. A number of the winners' projects will be transformed into interactive hypertext modules and made available through the Web.

Another awards program, focusing on undergraduates, was inaugurated this year. In some ways, the student program most clearly illustrates the evolution of the UCES project and the effects of its community-building on computational science instruction. Students in courses that had incorporated instructional material from the UCES electronic archive were able to submit class projects for electronic publication on the UCES archive. The students shaped their instructional modules according to the five-part UCES paradigm designed to produce clearly focused, pedagogically sound units that could be easily shared and easily adapted to other courses and institutions. These materials highlighted computationally based interactivity using the Web, with the intent being both to increase student motivation and to have the projects reach a wider audience. [The next issue of *CS&E* (Winter '95) will discuss the two awards programs in more detail.—*Ed.*]

Building courseware

At the center of the UCES notion of on-line, interactive courseware is the concept of instructional modules. Each module is a self-contained teaching unit limited in scope to a single concept, an illustration of a concept, or a computationally based problem. It is structured as a miniweb of interactive hypertext documents, formatted in HTML for Web viewing, and aimed at a specific undergraduate audience. The goal in creating such modules is to generate a database of discrete instructional units that can be assembled in various configurations by individual teachers. These units can easily be linked to other modules to form a customized body of material suitable for complete CSE courses or specific course units.

Some of the UCES modules now available on-line cover such topics as public-key encryption, variable-mass rockets, the electrostatic potential in a wire, and Buffon's Needle. The public-key encryption module, for example, asks students to address two of the major drawbacks of standard cryptography: securing a channel for transmitting a decoding key and identifying the sender of the message. In working through the problem, students learn how to determine the solution of congruency equations using the Euclidean algorithm and how to use symbolic algebras for integer factorization. Another module poses this question: From the Kitty Hawk sand dune in North Carolina, how far out into Albemarle Sound can one see? Among other instructional goals, the module illustrates the value of diagrams in deriving a suitable model, introduces students to the concept of floating-point error, and shows how algebraic simplification can convert an unstable computation into a stable one.

Articles and essays about CSE education are welcome.
CSE Education department editor Alvin I. Thaler, thaler@nsf.gov.

Many of the modules contain an interactive component that encourages the student to experiment with the concepts being discussed as they are encountered. For example, the public-key encryption module contains encryption/decryption panels (accessed through the Web's forms capability) which give the reader an opportunity to analyze various cryptographic solutions. UCES members have developed several pieces of custom software that make the construction of interactive educational materials straightforward. Several different models of interactivity are supported, although all use the Web and a browser such as Netscape or Mosaic. UCES hopes to release all of this software shortly into the public domain.

All UCES educational materials have been classroom tested, usually at two or three separate sites. A review board for new materials has recently been formed, and eventually all materials "published" on the UCES archive will have passed through a rigorous peer review process.

In its first two years, UCES has identified a clear educational need, set realistic goals for improving CSE education, and begun developing functional modules that could benefit undergraduate teaching for a broad audience. An on-line journal is being considered as a means of publicizing UCES activities and providing information of general interest to the CSE community.

Yet many practical and philosophical issues remain, even ones as fundamental as the way in which information is best structured for effective learning. Do the sequential, incremental organizational patterns of traditional course syllabi function in the multilinear, interactive environments of the Internet? Can teachers and researchers in diverse disciplines and institutions work collaboratively at a national level to shape the way a field is taught? Will hypermedia methods of instruction motivate students as independent learners? These and many other issues, like the changing nature of copyright in global digital environments, will figure prominently in the next stage of the UCES Project as it seeks practical solutions to the shifting educational needs in CSE. Educators who are interested in using UCES materials in the classroom or contributing to the UCES archive or collection of WWW-based educational tools are encouraged to contact the project at the address below. ◆

Acknowledgments

This work is funded through the Ames Laboratory, which is operated for the US Department of Energy by Iowa State University under contract no. W-7405-ENG-82. UCES is funded by DoE through the HPCC Initiative and the Office of Scientific Computing.

Current members of UCES are affiliated with Ames Laboratory, Appalachian State University, Capital University, Clemson University, Colorado School of Mines, Colorado State University, Cornell College, Cray Research, Duke University, the IEEE Computer Society, Mississippi State University, National Center for Supercomputing Applications, North Carolina State University, Oregon State University, Sandia National Laboratory, San Diego State University, Swarthmore College, TELOS/Springer-Verlag, University of California at Berkeley, University of Colorado, University of Illinois at Urbana-Champaign, University of Michigan, University of Tulsa, University of Utah, University of Washington, US Department of Energy, and Waterloo Maple.

Thomas L. Marchioro II is a theoretical/computational physicist who serves as associate program manager of UCES. David M. Martin is professor of materials science and engineering at Iowa State University, and the program manager of UCES. W. Donald Payne is a professor of English and rhetoric at Iowa State University and has a longstanding interest in issues of electronic publication. Readers may direct questions and comments to Marchioro at 126 Wilhelm Hall, Ames Laboratory, Ames, IA 50011-3020; e-mail uces@ameslab.gov.

References

1. Uniform Resource Locator: http://ams.ameslab.gov/AiS/national/index.html.
2. V.M. Umar, M.R. Strayer, and C.E. Oliver, "An On-Line Introduction to Computational Science via CSEP," *IEEE Computational Science & Eng.*, Vol. 1, No. 4, 1994, pp. 6–9.
3. V.M. Umar, M.R. Strayer, and C.E. Oliver, "Building an Electronic Book on the Internet: CSEP—An Interdisciplinary Syllabus for Teaching Computational Science at the Graduate Level," *Proc. IEEE Conf. Frontiers in Education*, IEEE, Piscataway, N.J., 1994, pp. 430–433.
4. Uniform Resource Locator: http://csep1.phy.ornl.gov/csep.html.
5. Uniform Resource Locator: http://ams.ameslab.gov/CSGF/home.html.
6. Some sense of CSE may be gleaned from the 13 academic programs described in J.R. Rice, "Academic Programs in Computational Science and Engineering," *IEEE Computational Science & Eng.*, Vol. 1, No. 1, 1994, pp. 13–21.
7. H.J. Raveche, D.H. Lawrie, and A.M. Despain, *A National Computing Initiative: The Agenda for Leadership, Report of the Panel on Research Issues in Large-Scale Computational Science and Engineering*, SIAM, Philadelphia, 1987.
8. Uniform Resource Locator: http://uces.ameslab.gov/uces/home.html.