CONF-9411137--2

Teaching Interdisciplinary Computational Science from an Electronic Book on the Internet

C. Edward Oliver
Michael R. Strayer
Oak Ridge National Laboratory*
P. O. Box 2008
Oak Ridge, TN 37831-6259

Verena M. Umar
Vanderbilt University
105 Computer Center
Nashville, TN 37240

To be presented at the Frontiers in Education Conference, San Jose, CA, November 2 -6, 1994; to be published in the Proceedings.

*Managed by Martin Marietta Energy Systems, Inc. under contract DE-AC05-84OR21400 with the U.S. Department of Energy

DISCLAIMER

Teaching Interdisciplinary Computational Science from an Electronic Book on the Internet

C.Edward Oliver, Michael R.Strayer
Oak Ridge National Laboratory*

Verena M.Umar
Vanderbilt University

INTRODUCTION

The advances in high performance computing technology have led to great changes in the way applied and pure sciences are being carried out. These advances have left curricula at the graduate level in many disciplines omitting a whole body of essential research tools that students have to seek out on their own to carry out their research. In many cases the skills are not difficult to assimilate and the student develops the required knowledge during the process of carrying out research. Usually this is not a systematic approach, and it often leaves the student with a fragmented knowledge of some of the increasingly important concepts and areas relating to computing. By offering a general introduction and overview to networks, computer architectures, scientific visualization, programming languages, a variety of operating systems and hypertext authoring tools students get a broad comprehensive view of modern high performance computing methodologies. Students from all disciplines benefit from such a course. The survey of such tools is ideally suited for an interdisciplinary curriculum. As high performance computing becomes an essential component of the university curriculum, instructors need to be prepared to incorporate these concepts and methods in their own research and curricula. At this point in time educators are uncertain as to how to proceed, and there appears to be a lack of concensus on the curriculum for computational science [1,2].

The Computational Science Education Project was initiated in September 1991, by the Department of Energy to develop a syllabus for teaching interdisciplinary computational science. The interdisciplinary nature of the project is intended to contribute to national technological competitiveness by producing a body of graduates with the necessary skills to operate effectively in high performance computing environments [3]. Presented elsewhere at this conference is a paper describing the CSEP e-book in greater detail. The title of said paper is: Building an Electronic Book on the Internet: "CSEP - an interdisciplinary syllabus for teaching computational science at the graduate level."

Besides the curriculum issues there is presently no textbook that surveys modern high performance tools. Although many new books deal with some of these issues in depth, there is no set of materials for a general introduction to computation and networks and it is unlikely that there will be one since the development of modern technology is so fast paced. Following the D.O.E. initiative on high performance computing education the CSEP has produced materials of this nature, and is reaching out to the computational science community to provide curriculum aid to instructors.

TWO ILLUSTRATIVE CURRICULE

The CSEP materials have been taught at a number of universities. Two sample courses will be described next, one at the University of Tennessee, out of the Physics Department, and the other at Colorado State University, out of the Mathematics Department.

At the University of Tennessee the course was a two-semester laboratory course with three credit hours described as being "A Course in Modern High Performance Computation for the Physical and Life Sciences". Prerequisites of the course were some initial familiarity with the use of computers for practical calculation, communication, and graphical display. Although other languages such as C were employed on occasion, the working language of the course was Fortran.

The first semester covered: An Overview of Computational Science, Computer Architectures, Networks and Performance Analysis, Scientific Visualization, Computational languages, and Linear Systems. The second semester addressed Numerical Methods for Ordinary Differential Equations, Numerical Methods for Partial Differential Equations, and Random Number Generators. Special projects were assigned drawn partially from the case studies and from material suggested by the students. These projects involved team work and involved probing some aspect of high performance computing technology in depth.

The teaching laboratory offered 12 Sun Sparc-1 workstations with a high-end server. The e-book was provided to the students locally from the server to reduce network loading. The student projects also required access to supercomputers; a MasPar, a CM-5, an iPSC/860, and a KSR-1.

Students from a number of departments attended this course. They came from mathematics, economics, nuclear engineering, physics, zoology, computer science and microbiology. The laboratory was available for 60 hours during a semester and they were officially required to attend three hours per week. The course was self-study with no lectures, but during the 60 hours the laboratory was staffed by either teaching assistants or faculty. The grading was pass/fail and required the students to electronically submit all of the exercises covered in the CSEP syllabus.

In contrast the Colorado State Course was a combination of lectures and laboratory work receiving 3 hours of credit per semester. Here again the students spanned diverse disciplines, including mathematics, computer science, range science, industrial engineering, civil engineering, and mechanical engineering.
The Colorado State course seemed to attract approximately the same number of students and distribution of disciplines as the University of Tennessee course, approximately 16 students from 6-8 departments. The students had access to a CM-5, a Paragon, a Cray-2, and workstations for the laboratory work.

This course, titled "Introduction to High Performance Computing and Visualization" covered the following areas in the first semester: HPC - An Overview, Scientific Visualization, Computer Networks, Computer Architectures, Programming Languages, Principles of Vectorization, Principles of Prallelization, Performance Programming, and Systems of Linear Equations. The second semester covered the remaining chapters on Selected Methods and one case study.

Overall students at both the University of Tennessee and Colorado State University gave the CSEP syllabus high ratings. All students enjoyed the courses and greatly appreciate having the public domain tools at their disposal for their research work. The e-book contains more material than can be comfortably taught, but generally more exercises and examples were needed.

ORGANIZATIONAL ISSUES

In setting up a new computational science course there are several issues besides the administrative preparations. Workstations, high performance computing resources and networks need to be made available to the students. Workstations are needed to read the e-book, experiment with visualization tools and as computing platforms. To only read the e-book the platform needs to have a WWW browser installed that supports graphics and forms, for example NCSA's Mosaic for the Macintosh. The systems used for reading CSEP material should be connected to the Internet. CSEP is setting up a number of mirror sites, which will be seen on the homepage at the URL: http://csep1.phy.ornl.gov/csep.html , and choosing a site that is geographically close to your institution will improve the viewing of the e-book. In practice it is preferable to set up a local mirror site to improve response.

To experiment with the visualization tools a Unix workstation is required. The easily accessible public domain software used for CSEP needs to be installed. C and Fortran compilers are recommended for the exercises. A F90 compiler and C++ are optional, but required for some of the exercises.

Access to remote high performance platforms is needed in particular for the case studies. You may contact an NSF center or National Laboratory for a class account. Such preparations can require several months.

SAMPLE GRAPHICS FROM EXERCISES

Figures 1 through 4 show graphics generated from exercises in various of the CSEP chapters using public domain software. Figure 1 shows a two-dimensional vector field, which is the gradient of the function $x^2 - y^2$, as generated with plotmtv from sample data obtained by a Fortran code provided to the students. Figure 2 shows spring winds over the Sea of Japan as generated with the Generalized Mapping Tool [4,5]. This example is taken from the Ocean Models case study, which discusses the visualization techniques and provides a database for the wind analysis. Figure 3 display Chaos Esthethique, which is discussed in the case study on Nonlinear Mappings. They illustrate a conservative and a dissipative mapping corresponding to stable and unstable configurations of an accelerator beam. The plots were generated with xmgr. Figure 4 was taken from the chapter on Computer Architectures and shows how to construct an n-dimensional hypercube.

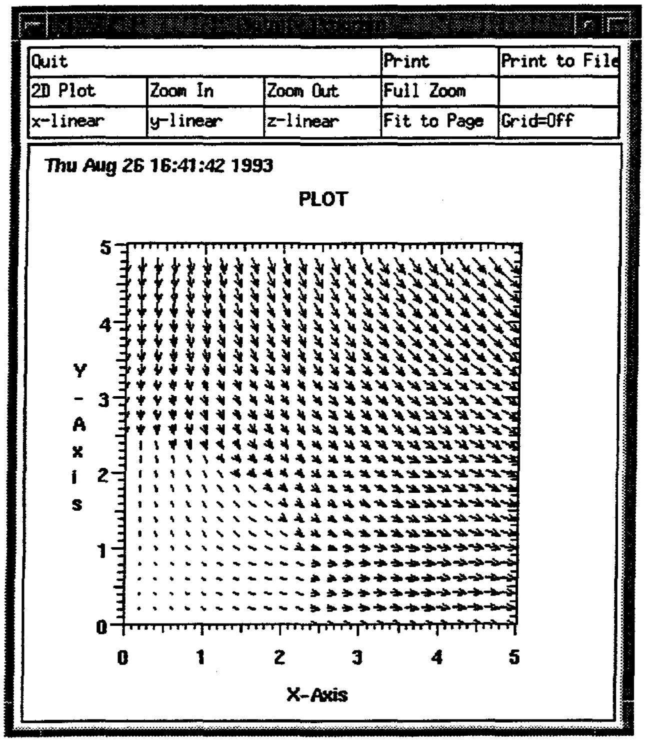

Figure 1

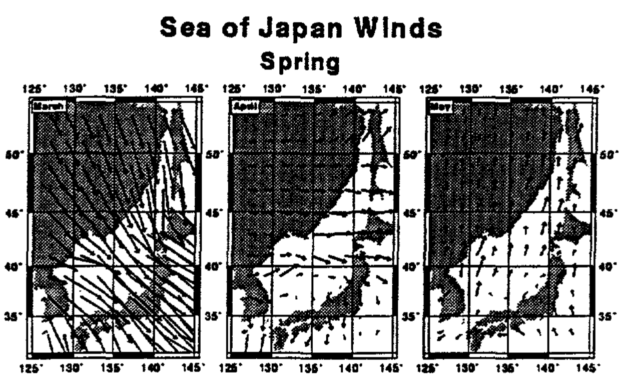

Figure 2

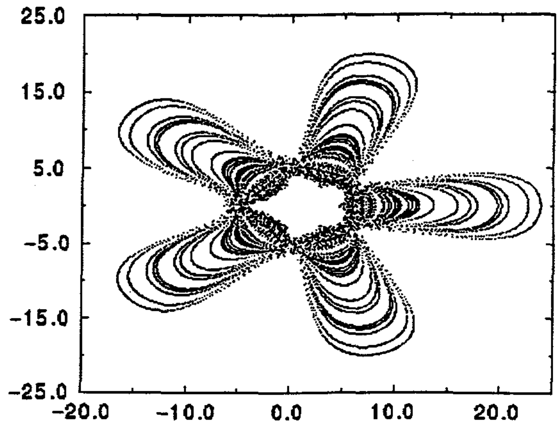

Figure 3a

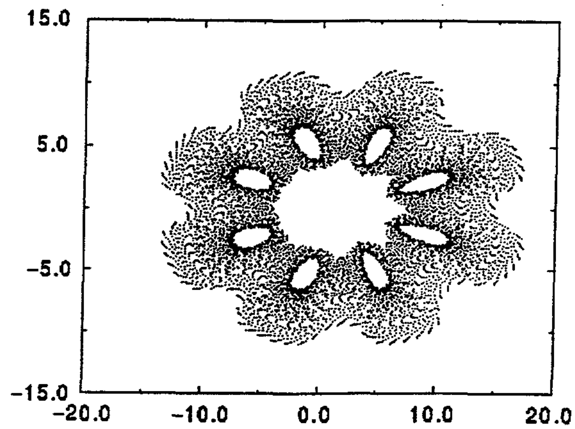

Figure 3b

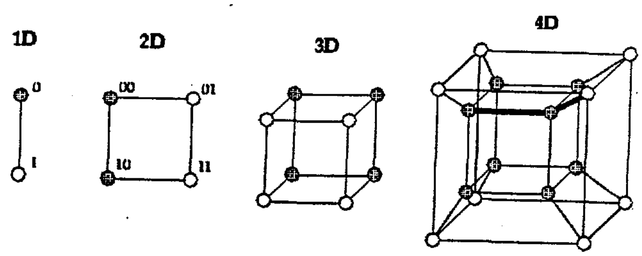

Figure 4

CONCLUSION

All students exhibited newly acquired hpc skills by the end of the year and appreciated had access to a whole structure of public domain tools appropriate for carrying on research in their disciplines. The greatest progress was made in the areas of visualization and networking. The students were generally enthusiastic about the networking and visualization issues in the book. This is probably due to the universal nature of these subject areas. The computational analysis was more difficult and students from physics and mathematics faired better than the students from biology and economics doing less well. The case studies are clearly for advanced study by students of particular disciplines. The engineering students were generally more in tune with the syllabus.

FIGURE CAPTIONS

Figure 1 - Two-dimensional vector field, that is the gradient of the function $x^2 - y^2$, generated with plotmtv.
Figure 2 - Spring Winds over the Sea of Japan generated with the Generalized Mapping Tool - taken from the section of the Ocean Models case study that discusses the effects of wind stress on the ocean currents.
Figure 3 a - Conservative mapping for Chaos Esthetique. The data set for the plot was generated from a Fortran program that implemented the mapping, and the plot was created from the data set using xmgr.
Figure 3 b - Dissipative mapping for Chaos Esthetique. The data set for the plot was generated from a Fortran Program that implemented the mapping, and the plot was created from the data set using xmgr.
Figure 4 - How to construct an n-dimensional cube, copy an (n-1)-dimensional cube, then connect corresponding nodes in the original and the copy. In these figures nodes from the original are colored gray.

REFERENCES

[1] - Geoffrey Fox, Proceedings of the National Academy of Arts and Sciences, 1992.
[2] - The Computational Science Education Project (CSEP), Proceedings of IEEE Frontiers in Eduction Conference, Vol. 121, 1993.
[3] - High Performance Computing and Information: Toward a National Information Infrastructure. OSTP 1994.
[4] - P. Wessel and W.H.F. Smith, Free Software helps map and display data, EOS Trans. AGU, 72, 441, 445-446, 1991.
[5] - W.H.F. Smith and P. Wessel, Gridding with continuous curvature splines in tension, Geophysics, 55, 293-305, 1990.

*****

C. Edward Oliver

Dr. C. Edward Oliver received his Ph.D. in Mathematics in 1969 from the University of Alabama. He is currently the Associate Director of Oak Ridge National Laboratory for Computing, Networking, Informatics, and Education.

Michael R. Strayer

Dr. Michael R. Strayer received his Ph.D. in Physics from M.I.T. He is presently the Section Head of Theoretical and Computational Physics at Oak Ridge National Laboratory.

Verena M. Umar

Verena M. Umar received an M.S. in Applied Mathematics from Claremont Graduate School and an M.S. in Computer Science from Vanderbilt University. She is currently Editorial and Technical Assistant to the Computational Science Education Project and part of the Research Support group at the Vanderbilt University Computer Center.

---

*Managed by Martin Marietta Energy Systems, Inc. under contract DE-AC05-84OR21400 with the U.S. Department of Energy.