Integrating Teaching and Research in HPC: Experiences and Opportunities

M. Berzins[1], R.M. Kirby[1], and C.R. Johnson[1]

School of Computing and Scientific Computing and Imaging Institute,
University of Utah, Salt Lake City, UT, USA

Abstract. Multidisciplinary research reliant upon high-performance computing stretches the traditional educational framework into which it is often shoehorned. Multidisciplinary research centers, coupled with flexible and responsive educational plans, provide a means of training the next generation of multidisciplinary computational scientists and engineers. The purpose of this paper is to address some of the issues associated with providing appropriate education for those being trained by, and in the future being employed by, multidisciplinary computational science research environments.

1 Introduction

The emerging multidisciplinary area of Computing, as distinguished from traditional Computer Science, is the study and solution of a new class of multidisciplinary problems whose solution depends on the combination of state-of-the-art computer science coupled with domain-specific expertise in such areas as medicine, engineering, biology, and geophysics. In a Computing Research Association article [1], Foley describes Computing as the integration of Computer Science and other disciplines to address problems of wide interest as illustrated in Figure 1. Multidisciplinary Computing is one of the fastest growing research areas in the US and Europe.

Examples of typical multidisciplinary computing problems are:

- How can we efficiently store, model, visualize and understand the mass of data generated by the human genome program?
- How might we model, simulate and visualize the functions of the heart and brain to better diagnose and treat cardiac and neural abnormalities with a view to improving the quality of life?
- How might we compute solutions to realistic physical models of dangerous situations such as explosions with a view to improving safety?

The next wave of industry growth will focus on opportunities resulting from the answers to questions such as these. Examples of Computing efforts at the University of Utah include the School of Computing, Scientific Computing and Imaging (SCI) Institute, the Department of Energy (DOE) ASCI Center for

V.S. Sunderam et al. (Eds.): ICCS 2005, LNCS 3515, pp. 36–43, 2005.

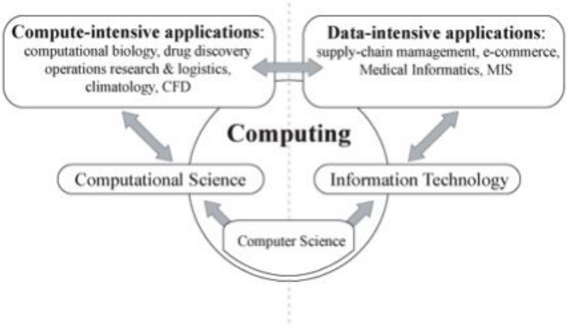

Fig. 1. Relationships between Computing, Computer Science and Applications, adapted from J. Foley's CRA article [1]

the Simulation of Accidental Fires and Explosions (C-SAFE), the NSF Grid-Computing Environment for Research, and Education in Computational Engineering Science, among several others.

The objective of this paper is to present an educational model that bridges the research mission of university computing research activities with the educational mission of the university in a synergistic way that benefits both the university and the student. We present a new University of Utah program that provides educational opportunities specifically enhanced by interaction with on-campus computing research activities. This program is a Ph.D. program in Computing with emphasis tracks in Scientific Computing, Computer Graphics and Visualization, and Robotics, offered through the School of Computing. It is worth stressing that these are not developments in isolation. In 1998, 31 graduate programs in computational science at U.S. Universities had been created. As of 2003, the number had grown to 47. In addition, since 1998, 16 new undergraduate degree programs in computational science had been created. The Computing track in Scientific Computing benefits from, and builds upon, the current M.S. degree program in Computational Engineering and Science (CES) [2].

The paper is organized as follows. In Section 2, we will present the research missions and research results from two large computing research centers that reside on the University of Utah campus. In Section 3, we will present details concerning the new Computing graduate degree program, with specific emphasis on how this educational programs provide a win-win situation for both the research missions of the centers and the educational mission of the university. We will use an example from a high performance computing course to illustrate the intertwined nature of classroom education and research education. We conclude in Section 4 with a summary and discussion of our findings concerning this integrated approach.

2 Multidisciplinary Research Efforts at Utah

To accurately understand and appreciate the environment in which these programs were developed, we will present a discussion of two current research centers at the University of Utah. The first of these is the Center for the Simulation of Accidental Fires and Explosions (C-SAFE), funded by the U.S. Department of Energy, which represents a center whose focus is the physical sciences and engineering. The second is the Center for Bioelectric Field Modeling, Simulation, and Visualization funded by the U.S. National Institutes of Health (NIH), which represents a center whose focus is in biomedicine and bioengineering. These two centers represent research efforts rich in opportunity for integrating teaching and research in high-performance computing.

2.1 Center for the Simulation of Accidental Fires and Explosions (C-SAFE)

C-SAFE is funded under the Department of Energy's Accelerated Strategic Computing Initiative (ASCI) program. The primary goal of C-SAFE focuses specifically on providing state-of-the-art, science-based tools for the numerical simulation of accidental fires and explosions, especially within the context of handling and storage of highly flammable materials. In Figure 2 (left) we present a visualization of a fire calculation which required the efforts of computational scientists, mathematicians and engineers. The objective of C-SAFE is to provide a system comprising a problem-solving environment (the Uintah PSE) [3, 4] in which fundamental chemistry and engineering physics are fully coupled with non-linear solvers, optimization, computational steering, visualization and experimental data verification.

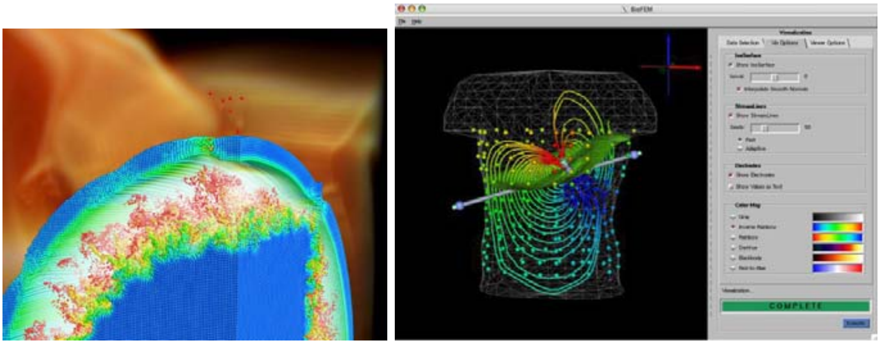

Fig. 2. C-SAFE (left): A simulation of an explosive device enveloped in a jet fuel fire, just after the point of explosion. Particles representing the solid materials (steel and HMX) are colored by temperature, and the gases (PBX product gases and fire) are volume rendered. **NCRR BioFEM PowerApp (right)**: A modeling, simulation and visualization environment for bioelectric fields. Shown here is a visualization showing the results from a finite element simulation of electric current and voltage within a model of the human torso

One of the major educational challenges posed by this environment is balancing the need to lay a firm foundation in high-performance computing "fundamentals" while at the same time exposing students to the practical issues that arise in large-scale high-performance codes as used by C-SAFE. Often times concepts and tools are taught serially across different courses and different textbooks (and with a variety of application domains in mind), and hence the interconnection between the education and the practical is not immediately apparent. Of particular importance to the mission of C-SAFE is the ability of the software to use large numbers of processors in a scalable way but also to be able to use adaptive meshes in both space and time as a means of changing resolution in order to increase the fidelity of the computation. These aims may be conflicting unless great care is taken. In Section 3.2 we present a description of a high-performance computing and parallelization course offered as part of the Computing Program which attempts to address this issue.

2.2 Center for Bioelectric Field Modeling, Simulation, and Visualization

In 2000, one of the authors (CRJ) saw the need for interdisciplinary biomedical computing research as expressed in the following [5]:

> "[R]evolutionary solutions to important science and technology problems are likely to emerge from scientists and engineers who are working at the frontiers of their respective disciplines and are also engaged in dynamic interdisciplinary interactions. . . . [B]iomedicine is now particularly well poised to contribute to advances in other disciplines and to benefit substantially from interactions with those disciplines."

In keeping with this vision, Johnson *et al.* initiated the NIH-funded Center for Bioelectric Field Modeling, Simulation, and Visualization at the University of Utah. The motivation for this Center comes from the confluence of scientific imagination and the maturation of the technology required to pursue new ideas. As computers have become more and more powerful, their users have acquired the potential ability to model, simulate, and visualize increasingly complex physical and physiological phenomena. To realize this new potential there have also been concomitant advances in computer software such as graphical user interfaces, numerical algorithms, and scientific visualization techniques. This combination of more powerful devices and the software to use them has allowed scientists to apply computing approaches to a continually growing number of important areas—such as medicine and, in particular, the important field of bioelectricity.

The mission of the Center is:

- To conduct technological research and development in advanced modeling, simulation, and visualization methods for solving bioelectric field problems.
- To create and disseminate an integrated software problem solving environment for bioelectric field problems called BioPSE [6] which allows interaction

between the modeling, computation, and visualization phases of a bioelectric field simulation as illustrated in Figure 2 (right).

One of the educational challenges within this type of environment is to develop a curriculum which instills good software engineering practices within the context of user-driven scientific computing software. Portability, extensibility, usability and efficiency all compete in this type of software environment; most Computing training focuses on one or two of these issues, but does not show how to balance the competing interests of these areas to create a product which meets the mission as stated above. The Computing degree infrastructure described in Section 3 is designed to accommodate these type of needs.

3 Integrating Research and Teaching

Students participating in high-tech research areas with our faculty are at present limited to academic program choices that do not currently reflect either the changing multidisciplinary demands of employers in industry nor the actual breadth and multidisciplinary nature of their research training and achievements. While many of these students participate in the high-quality Computer Science graduate program, their multidisciplinary needs and aspirations are somewhat different from those satisfied by conventional Computer Science, which provides more emphasis on learning about computer hardware, operating systems, and theory, and less on how to solve real-world interdisciplinary computing problems.

To bridge the gap between the high-performance programming and computing needs of the research centers as described above, we envisage an integrated research and teaching environment which provides sufficient structure to instill foundational scientific computing knowledge while providing sufficient freedom to individualize a program of study to the student's research and professional needs. The bridge has been built within the new Computing Degree offered by the School of Computing at the University of Utah, which is described in the next section.

3.1 Computing Degree Program

Two key features of our new Computing graduate degree structure are particularly designed to meet this student expectation. Not only is the new Computing degree designed to integrate knowledge from many starting points (engineering, mathematics, physics, medicine), but its track structure makes it possible to build natural and student-centered collaborative academic programs across the University. The Computing degree structure operates at both Masters and Doctoral level and is interdisciplinary through its track structure. Each track has a minimum of six faculty members who form a Track Faculty Committee. This track structure makes it possible for the Computing degree to be applicable to emerging multidisciplinary problems with a maximum of efficiency in a sound academic manner. We note that academic tracks have been shown to be a successful mechanism for offering a variety of educational opportunities within a larger degree option.

The current tracks existing under the umbrella of the Computing Degree are: (1) Scientific Computing, (2) Computer Graphics and Visualization and (3) Robotics. Our focus in this paper is on the scientific computing track.

The Scientific Computing track trains students to perform cutting edge research in all of the aspects of the scientific computing pipeline: mathematical and geometric modeling; advanced methods in simulation such as high-performance computing and parallelization; numerical algorithm development; scientific visualization; and evaluation with respect to basic science and engineering. Students apply this knowledge to real-world problems in important scientific disciplines, including combustion, mechanics, geophysics, fluid dynamics, biology, and medicine. Students integrate all aspects of computational science, yielding a new generation of computational scientists and engineers who are performing fundamental research in scientific computing, as well as being interdisciplinary "bridge-builders" that facilitate interconnections between disciplines that normally do not interact. Our mission is to provide advanced graduate training in scientific computing and to foster the synergistic combination of computer and computational sciences with domain disciplines.

The scientific computing track requires only four "fundamental" courses: Advanced Scientific Computing I/II, Scientific Visualization, and High-Performance Computing and Parallelization. These four courses are designed to provide sufficient breadth in computing issues as to allow individual faculty members to then use the remaining course hour requirements to individually direct a student's program of study to meet that student's research needs. In the following section, we describe in depth one of the four aforementioned classes, with the specific intent of showing how it fulfills the gap-filling need described earlier.

3.2 Computing Degree Program - "High-Performance Computing and Parallelization" Course

In this section we take one example from the Scientific Computing track of the new Computing degree and relate it to the C-SAFE research in high performance computing.

The course entitled "High Performance Computing and Parallelization" is intended to make it possible to understand parallel computer architecture at a high level; to write portable parallel programs using the message passing system MPI; and to understand how to construct performance models for parallel programs. The course covers the use of workstation networks as parallel computers and issues such as data decomposition, load balancing, communications and synchronization in the design of parallel programs. Both distributed memory and shared memory programming models are used. Performance models and practical performance analysis are applied to multiple case studies of parallel applications. The course is based on the books [7, 8] with background material from [9] and from a number of research papers such as [10, 4, 11, 3].

The course assignments involve writing parallel programs on a parallel computing cluster. One issue that arises in the teaching of this material is the conflict between the students being able to learn quickly and possibly interactively if at all possible against the normal mode of batch production runs. Often the best way to resolve this conflict is through the purchase of a small teaching cluster.

Simple Performance Analysis. In understanding parallel performance it is first necessary to understand serial performance as the concepts that occur on parallel machines such as memory hierarchy are also present on serial machines in the shape of cache and *tlb* effects [8]. In understanding parallel performance and scalability the concepts of Isoefficiency, Isomemory and Isotime are all important and are often the most difficult topics for the students to grasp. Isoefficiency studies consider how fast the problem size has to grow as the number of processors grows to maintain constant efficiency. Isotime studies consider how fast the problem size has to grow as the number of processors grows to maintain constant execution time. Isomemory studies consider how fast the problem size has to grow as the number of processors grows to maintain constant memory use per processor.

These metrics may be defined for a problem of size n whose execution time is $T(n, p)$ on p processors and lead to a number of conclusions, see [10, 11]:

(i) If the Isotime function keeps (T(n,1)/p) constant then the Isotime model keeps constant efficiency, and the parallel system is scalable.
(ii) If execution time is a function of (n/p) then the Isotime and Isoefficiency functions grow linearly with the number of processors, and the parallel system is scalable.
(iii) If the Isotime function grows linearly then the Isoefficiency function grows linearly, and the parallel system is scalable.
(iv) If Isoefficiency grows linearly and the computational complexity is linear then the Isotime grows linearly, and the parallel system is scalable.

Martin and Tirado [11] quote an illuminating example from linear algebra characterized by a multigrid problem size of N^2 for which Isomemory and Isotime require $N^2 = p$ while for Isoefficiency $N^2 = p^2$. In this case if problem size is scaled with Isotime (and memory) execution time is constant and efficiency decreases slowly. In their example a 128x128 problem on 2 processors needs to go to 512x512 on 8 processors for Isoefficiency, rather than 256x256 on 8 processors for Isotime performance.

The importance of such results for C-SAFE, as it moves towards an adaptive parallel architecture of a very complex multi-physics code, is that they provides a good theoretical base for those involved in the development of the load balancing algorithms needed to make effective use of large numbers of processors on the latest generation of machines.

4 Summary and Discussion

Multidisciplinary research has become an integral part of the research landscape, and its importance will continue to grow in the future. How discipline-centered university programs adapt to the changing nature of research will directly impact scientific and engineering progress in this next century. More tightly coupled integration of research and teaching is mandatory. The University of Utah's Computing Degree Program as described in this paper provides a mechanism solid enough to provide stability to students while progressive enough to adapt to varying needs of both the student and the research centers with which the students interact.

Acknowledgments

This work was supported by NIH NCRR grant 5P41RR012553-02 and by awards from DOE and NSF. The SCIRun and BioPSE software are available as open source from the SCI Institute website: www.sci.utah.edu.

References

[1] Jim Foley. Computing > computer science. *Computing Research News*, 14(4):6, 2002.

[2] Carleton DeTar, Aaron L. Fogelson, Chris R. Johnson, Christopher A. Sikorski, and Thanh Truong. Computational engineering and science program at the University of Utah. In *Proceedings of the International Conference on Computational Science (ICCS) 2004, M. Bubak et al, editors, Lecture Notes in Computer Science (LNCS) 3039, Part 4*, pages 1202–1209, 2004.

[3] J.D. de St. Germain, J. McCorquodale, S.G. Parker, and C.R. Johnson. Uintah: A massively parallel problem solving environment. In *Proceedings of the Ninth IEEE International Symposium on High Performance and Distributed Computing*, August 2000.

[4] S. G. Parker. A component-based architecture for parallel multi-physics PDE simulation. In *International Conference on Computational Science (ICCS2002) Workshop on PDE Software*, April 21–24 2002.

[5] Focus 2000: Exploring the Intersection of Biology, Information Technology, and Physical Systems.

[6] BioPSE: Problem Solving Environment for modeling, simulation, and visualization of bioelectric fields. Scientific Computing and Imaging Institute (SCI), http://software.sci.utah.edu/biopse.html, 2002.

[7] B. Wilkinson and M. Allen. *Parallel Programming: techniques and applications using networked workstations and parallel computers (Second Edition).* Prentice Hall, Inc., Englewood Cliffs, N.J., 2004.

[8] S. Goedecker and M. Hoisie. *Performance Optimization of Numerically Intensive Codes.* SIAM, Philadelphia, PA, USA, 2001.

[9] P.S. Pacheco. *Parallel Programming with MPI.* Morgan Kaufmann, 1997.

[10] M. Llorente, F. Tirado, and L. Vázquez. Some aspects about the scalability of scientific applications on parallel computers. *Parallel Computing*, 22:1169–1195, 1996.

[11] Ignacio Martin and Fransisco Tirado. Relationships between efficiency and execution time of full multigrid methods on parallel computers. *IEEE Transactions on Parallel and Distributed Systems*, 8(6):562–573, 1997.