AN INITIATIVE TO SUPPORT COMPUTATIONAL SCIENCE AT THE UNIVERSITY OF VIRGINIA

P R E L I M I N A R Y   D R A F T: 7/9/06

By the:
U.Va. Task Force on Information Technology Infrastructure Supporting Research in Science and Engineering

CONTENTS

1. BACKGROUND

At the present time the University of Virginia is engaged in a number of efforts to assess its status and chart a direction for the future. Among these efforts are the development of the Case Statement for the upcoming capital campaign, and the development of a new 10 year strategic plan for the University. Thus this is a propitious moment to consider the future of computational science and engineering at the University.

Among the strategic objectives of the University of Virginia are the following goals: (1) Improve the University's core educational mission, (2) attract, retain, and support highly qualified faculty, (3) Develop the University's potential as a research institution together with cutting-edge research programs, (4) promote interdisciplinary collaboration, and (5) develop new modes of learning, teaching, and research.   This report discusses an emerging discipline that will prove essential if the University is to achieve its objectives: computational science. During recent years there have been dramatic advances in computational science that will revolutionize how problems are solved. The academic institutions that will emerge as world leaders will be those that recognize the significance of computational science to their strategic goals and act appropriately. This report is the first step toward formulating an effective strategy for the University of Virginia.

What is computational science?

Science is the process by which we observe the world, infer general principles that systematize those observations, and then deduce the observable consequences of those principles. These activities involve the collection and analysis of information.

Computational science is an emerging discipline and, as such, one whose definition is itself evolving. The traditional definition of computational science is that it undertakes an investigative approach to the understanding of systems (physical, biological, mechanical, etc.) through the use of mathematical models that are solved on high performance computers. Computational science grew out of necessity; the development of the modern computer was driven, in large part, by the need to solve complex equations in science and engineering. Today computational science includes not only simulation and numerical modeling, but also the powerful new ways in which data generated from experiments and observation can be manipulated and probed to gain scientific insight. These activities are often referred to as data mining and analysis. The broadest definition of computational science would encompass the development and application of problem-solving methodologies that rely on advanced computational resources, including processor, storage, network, and display technologies.

It is helpful to state what computational science is not. First, it is not computer science. Computer science is a specific discipline: the science of the computer itself. Computer science is often involved in important and ground-breaking computational science research, but there no requirement that all computational science directly involve computer science. Second, all uses of a computer in service of science are not necessarily computational science. The computer has undoubtedly become an indispensable tool in such activities as scholarly writing, literature research, production and manipulation of graphics and images, and experimental control and data acquisition. These activities are not normally regarded as computational science, even though they may require a significant amount of technical expertise. Third, computational science is not confined to supercomputing, i.e., the use of the most advanced and largest processors.

The purpose of computational science is the science, and its procedures follow the usual course of scientific inquiry. Research begins with a question, a conjecture, or a hypothesis to be tested and ends with a moment of discovery that constitutes at least a partial answer. The intermediate steps generally include framing the question in terms of a model; this most often means a determination of the equations governing the model. The scientist then looks for previously unrealized implications of this model. In computational science the governing equations must be put in terms amenable to solution by a numerical algorithm. This algorithm must then be converted to a machine-executable form through programming. This application must then be placed on a suitable hardware platform and, ideally, implemented in an appropriate and efficient manner for this platform. After testing and validation, the resulting program is used to execute a series of numerical experiments that have the potential to probe the question. The outcomes of these experiments are often themselves rather complex, requiring a comparable computational effort to analyze. Finally, out of this analysis one aims for new insights into the system under study. Computational science can be so effective in facilitating the exploration of data, that the outcomes often close the loop by assisting in the formulation of new questions for further research.

Many of these research steps are familiar; they are the same steps taken in any theoretical or experimental science. But computational science requires additional procedures and new areas of expertise. Discipline-trained scientists will not necessarily possess all of the requisite knowledge and, given the pace at which the technology evolves, may find it difficult even to maintain an appropriate level of expertise. Continual advanced training and research collaborations are essential.

Trends and History at the National Level

Twenty-five years ago computational science was severely hampered by limited resources. The computers that were available to the academic community were far from the state of the art, even for the era. They were located centrally and employed a time-sharing model that greatly increased the difficulty of use. Supercomputers existed, but they were primarily located within government labs and agencies, employed for programmatic efforts where the recognized importance of computation could easily justify the high costs of these machines.

The importance of computational science to fundamental research was acknowledged in a significant way in 1985 with the establishment of the National Science Foundation (NSF) supercomputing centers, dedicated to making state of the art computational resources available to a broad community of investigators. Concurrently, another important trend was underway, namely the rise of the desktop computer and the Internet. During the next decade the capabilities of the supercomputers located at national facilities increased rapidly, driving the state of the art and enabling ever more complex and sophisticated modeling by leading researchers. At the same time the wide availability of powerful desktop workstations began to make computational science more ubiquitous, increasing the number of practitioners within the scientific disciplines.

The most recent decade has seen the desktop workstation become increasingly widespread and powerful. A typical processor now has substantially greater capability than that of the Cray computers installed in the first supercomputing centers 20 years ago. Similarly, the capabilities of the Internet have increased to the point that 1 Gigabit bandwidth to the desktop is not uncommon. The facilities at the national centers have also seen remarkable improvements. The NSF centers now allocate roughly 100 million service units (SUs) per year, where the service unit is one standard processor hour. Leading research groups are routinely awarded more than one million SUs per year in support of their work, an amount of computing power substantially greater than that available to the entire national community only a short time ago.

The rapid pace of development continues at the national level. The NSF has established the national TeraGrid, which uses a dedicated high-speed network to link together an impressive array of computational resources located throughout the country. The NSF's plan for the next five years calls for the development of petascale computing capabilities, that is, processors capable of delivering 10^15 floating point operations per second and systems that can manipulate datasets containing on the order of 10^15 bytes.[1] Other federal agencies, e.g., DOE, NASA, and NIH, are also developing substantial computational capacity independent of the NSF initiatives. These agencies are making this investment because they recognize that ground breaking research will be produced with these facilities.

For a long time the emphasis has been on the hardware and the growth of the raw processor power. But this focus overlooks areas of equal or greater importance, such as the software that runs on these systems and the people who develop and use that software. Million-fold increases in processor speed have not been matched by commensurate improvements in these other areas. Programming tools, languages, and techniques have scarcely evolved; researchers continue to program in outdated languages or employ legacy codes that have not been updated since the punch-card era. They employ storage and analysis techniques that were adequate for megabytes of data but are completely ill-suited to the petabytes to come. While visualization techniques are well-developed as long as only (x,y) plots or two dimensional contours are required, datasets now often span an N-dimensional parameter space and may also include time-dependent evolution information. Both the software and the hardware required to display and interact with such data are still relatively undeveloped. Thus the requirements for computational science include not only discipline specific knowledge, but also up-to-date knowledge in areas such as algorithms, programming, optimization, data management, and visualization; a single researcher cannot hope to master all of these.

Dealing with these complexities will require the collaboration of many skilled people, but such people are generally not widely available, and the training to develop such people is lacking. Graduate and undergraduate curricula remained centered on the research methodologies employed when the faculty were educated, despite the revolutionary changes seen in the last few decades.

Twenty years ago researchers were clearly limited by the available hardware; they could easily devise projects that would exceed the capabilities of the fastest computers. Now the relation is inverted: the capabilities of the most powerful computers easily exceed the average researcher's ability to utilize them in a meaningful way.

Thus despite all of the progress of the past two decades, there remain significant concerns regarding the development of computational science. These concerns are quite clearly spelled out by the President's Information Technology Advisory

Committee (PITAC) in their June 2005 report "Computational Science: Ensuring America's Competitiveness." This report details both those steps that must be taken at the national level and those required within universities and research communities. The principal finding of the PITAC report is as follows:

Computational Science is now indispensable to the solution of complex problems in every sector, from traditional science and engineering domains to such key areas as national security, public health, and economic innovation. Advances in computing and connectivity make it possible to develop computational models and capture and analyze unprecedented amounts of experimental and observational data to address problems previously deemed intractable or beyond imagination. Yet, despite the great opportunities and needs, universities and the Federal government have not effectively recognized the strategic significance of computational science in either their organizational structures or their research and educational planning. These inadequacies compromise U.S. scientific leadership, economic competitiveness and national security.[2]

In what follows we shall draw upon the conclusions of this detailed and visionary report in making our recommendations for an initiative in computational science at the University. As one of the leading universities in the nation and the preeminent academic institution within the Commonwealth it is our responsibility to heed this call to action.

History at U.Va.

On a smaller scale the history of computational science at U.Va. mirrors the national trends. Twenty years ago academic computing largely consisted of a central time-share system and a few departmentally-based systems. In the late 80s, as workstations became available an increasing number of them were placed within science and engineering departments. In the early 90s the University carried out a comprehensive deployment of networking infrastructure. As the decade progressed, information technology and computing became an important tool for all departments, not just science and engineering. Meeting this new demand presented a significant institutional challenge; there was little opportunity to focus upon the special needs of computational science.

In an effort to redress this, the Research Computing Task Force (RCTF) was established in late 2000 to develop recommendations in support of computational science. The final report [3] made a number of specific recommendations in a range of areas. Several of those recommendations are relevant to the present initiative. The recommendation for the acquisition of medium-scale high-performance computing has been addressed to a modest degree through the subsequent acquisition of the Aspen, Birch, and Cedar clusters. The fact that those clusters quickly became oversubscribed certainly testifies to the importance of their acquisition. U.Va. has continued to make strategic investments in networking, including forefront Internet services, first as a participant in Internet2, and now with the National Lambda Rail Initiative. There has been less success in addressing the concerns raised by the RCTF report in advanced support and leadership. Detailed follow-up proposals to create a center for advanced computation have remained unrealized during the recent period of stringent budgets. There has been at best only modest and uncoordinated progress in the areas of faculty development and training, or the enabling of computational science collaborations, or setting up coordinated efforts to develop partnerships with or facilitate access to national resource providers.

One of the major findings of the RCTF was the need for continued guidance and leadership from faculty researchers in charting a course for the University in the rapidly developing area of computational science. Despite this recommendation there remains little faculty leadership or institutional guidance in furtherance of computational science at the University.

Institutional support for computation and information technology is vested with Information Technology and Communication (ITC). ITC has wide-ranging responsibilities throughout the University; assistance for computational science is but one minor component. ITC must provide support for such diverse elements as student computer labs, instructional and administrative computing, the campus-wide network and even the telephones. In an effort to elucidate the needs of researchers ITC has relied on faculty committees, both standing (e.g. the University Committee for Information Technology) which were not specifically devoted to computational science, or ad hoc, such as the Research Computing Task Force. Unfortunately these mechanisms for obtaining faculty advice and direction have been neither frequent nor sustained. Often ITC staff members are placed in a position where they must respond to or evaluate specific requests that originate from individual faculty members or from research groups, without the necessary information or guidance to place those requests in a broader institutional context.

The absence of any coordinated backing for computational science is currently having repercussions within the University. It makes recruiting at all levels --- graduate students, postdoctoral researchers, and faculty --- more difficult. It limits the ability of researchers to carry out breakthrough science. It diminishes researchers' ability to respond competitively to sponsored research opportunities.

On the other hand, the recent growth in complexity and capability within computational science has been so dramatic that almost all existing programs at universities throughout the nation may be inadequate. All universities must address these challenges and indeed several are beginning to do so. A partial list of initiatives underway at other universities is given in Appendix A. From these efforts best practices are beginning to emerge. We believe that a well-planned and coordinated initiative to enhance computational science will produce the following outcomes:

* Enable forefront research and strengthen research programs throughout the University

* Create a supportive environment to respond to new interdisciplinary research opportunities

* Attract the best faculty and students to the University

* Ensure the University's place as a top institution for science and engineering

* Provide the appropriate educational experience to both graduate and undergraduate students to prepare them for research and leadership roles in computational science

2. GOALS OF THIS INITIATIVE

The Need for a New Initiative

The rationale for a university-wide initiative to further computational science and engineering lies with its fundamental interdisciplinary nature. The natural alliances for computational science bridge school and departmental boundaries. All branches of science and engineering are increasingly dependent on access to high-performance computing resources. The technical skills and expertise necessary to make effective use of those resources are also not unique to any discipline. The University faculty have long shown a willingness, even eagerness, to develop collaborations that span disciplines and schools. Such efforts cannot be well served, however, by administrative structures that conform to rigid departmental and school boundaries. An infrastructure and administration to support the needs of computational science that lies outside of specific departments or schools can achieve efficiencies of scale, ensure access to a greater range of resources by a larger community, and promote research across discipline lines, as appropriate.

The NSF supercomputing centers represent examples of powerful processing resources maintained at the national level in service to researchers throughout the country. The PITAC report recommends similar national leadership efforts in the form of data repositories and software sustainability centers. University support for computational science does not mean providing resources of this scale locally. Rather it involves ensuring that the appropriate local resources are provided that enable university researchers to participate in and take advantage of these national initiatives. By the same token, this initiative is not about commodity computing, but must focus on providing advanced resources that specifically enable breakthrough computational science. We must assume that the university will provide the basic support and infrastructure for faculty and researchers to carry out their standard duties in research, teaching and administration.

In order for this initiative to succeed in its goals, there must be new investments made in computational science infrastructure. But the Task Force's greatest concern was the need to develop a culture of computational science and engineering at the University. More than hardware, this will require leadership, a skilled talent pool of experts in the many aspects of computational science, and an organizational structure that permits growth, flexibility and the ability to respond in a timely manner to rapidly changing requirements.

In considering all of these concerns and requirements, the Task Force has chosen to frame the goals of this initiative in terms of four categories:

(A) A structure for leadership

(B) Advanced support for sponsored research

(C) Expert support for computational science

(D) Education and training in computational science

The components of each of these are discussed in more detail below.

A. Leadership

Given the rapid pace of developments within computational science, the complexities and expense involved in undertaking support for computational science, and the wide range of disciplines for which computational science is essential, it is vital that the University develop and implement a strategic plan. This calls for leadership and active participation of the research faculty.

As a part of the work of this task force we examined computational science activities and initiatives at other institutions. One fact that emerged was that the most successful such initiatives had a director for computational science or equivalent. The diverse and inherently interdisciplinary aspects of computational science require a leader who can coordinate the needs and concerns among the many departments and schools that engage in computational science. <u>We recommend that such a responsibility be created at U.Va.</u> The charge to the individual with that responsibility would include:

*** work with senior university administration to develop and execute a plan for university computational science**

*** promote, develop, and manage external partnerships with other universities and research organizations**

*** oversee the staff of the advanced support team**

*** chair the faculty advisory committee**

*** oversee research policy and resource allocation**

The appointment should be at a senior level, such as an associate vice president. (Such a position was finally created at Virginia Tech after some experimentation with alternatives.) The rank of this appointment is needed to coordinate with senior administration for effective strategic planning and implementation for computational science throughout the University.

Second we recommend the establishment of a Faculty Advisory Committee for computational science. The purpose of the Advisory Committee is to provide leadership for and communication and collaboration among the computational science and engineering researchers and the expert community. The Committee will promote the development of partnerships in innovative and interdisciplinary research, provide guidance in advancing computational science at the University, and advise on policy issues and resource allocation. This advisory committee will be chaired by the director for computational science.

We anticipate that one of the major outcomes of an initiative to develop computational science and engineering will be an enhanced capability of competitive response to major new sponsored research opportunities. To this end we recommend that a Grant Development Officer for computational science be named. Many of the future calls for proposals from granting agencies will expect significant computational and interdisciplinary components. The Grant Development Officer for Computational Science and Engineering will identify new opportunities, work with researchers to assemble teams that can respond to such opportunities, and coordinate the technical aspects and arrange appropriate cost-sharing, resource commitment and other details in the resulting proposal.

Finally we recognize that this leadership group will require an appropriate level of administrative support staff.

B. Sponsored research development program

One of the primary goals of this initiative is to facilitate sponsored research. In the future we must systematically pursue opportunities in computational science at various agencies (NSF, NASA, DOE, NIH, etc.). Appendix B presents several brief synopses of breakthroughs that U.Va.'s researchers could achieve by employing advanced computational science techniques and resources. Enabling these breakthroughs is one of the most important potential outcomes of this proposed initiative.

As discussed above, the Grant Development Officer will work with researchers to search for appropriate funding opportunities in computational science, particularly those initiatives that have a strong interdisciplinary component, assisting in grant proposal writing, ensuring compliance with institutional and agency requirements in proposal preparation, and providing administrative support of funded research. There is a need for pre-proposal technical consulting to ensure the feasibility of the proposed research. To increase the competitiveness of proposals, there should be available matching funds from a computational science development program. Such backing should include fractional FTEs of support personnel. Faculty will be encouraged to develop interdisciplinary teams to respond to granting agency opportunities.

Increased success in obtaining sponsored research grants carries with it obvious pragmatic benefits. One of these is that assistance with sponsored research permits growth and sustainability of a significant computational science infrastructure. The Academic Computing Health Sciences (ACHS) center is an exemplar. ACHS supports many small or proof-of-principle projects by core funding, but as projects grow and increase in scope, significant amounts of staff time can be shifted to direct costs. In this manner ongoing funding is leveraged to maintain several times the number of personnel that could otherwise be employed.

Many significant future research opportunities will focus on the development of interdisciplinary teams that cut across the traditional school and departmental boundaries. But it is worth noting that even more traditional single-discipline, single-investigator projects will be more likely to be funded if they have a significant computational science component.

An appropriate way to facilitate the computational science research of faculty and staff is through a Development Program that provides start-up funds, seed money, and matching funds or the equivalent (e.g., staff time). A primary aim will be to increase the competitiveness of grant proposals, support pilot projects that will lead to more competitive grant proposals, and to increase the competitiveness of faculty and postdoctoral hires through support for their planned computational science efforts. Another means to advance the University's national profile in this area would be targeted computational science and engineering professorships, some at senior levels to provide experienced faculty, and others at the junior level to build up the computational community.

We anticipate that there will be new opportunities to develop collaborations and partnerships. Among these are new state initiatives and programs, collaborations with federal laboratories and agencies (e.g., Oak Ridge National Laboratory, SURA, National Institute of Aerospace, Jefferson Lab, MATP), Corporate Technology partners, and collaborations with other universities.

During the Spring 2006 semester, the Task Force membership became aware of a number of grant opportunities in computational science. One of these, the CI-TEAM program at the NSF, was deemed appropriate for a proposal and a sub-group of the task force was created to respond. Such a response was possible only because the interdisciplinary Task Force was already in place, and the resulting proposal is a modest example of the type of activity that could be enabled by this initiative. The CI-TEAM proposal activity is described in Appendix C.

Frontier research is becoming more and more a collaborative activity. An outcome of the activities proposed herein will be to create an environment that encourages and facilitates collaboration. A possible way to achieve this would be to establish a facility where researchers can work together with experts in a resource-rich environment. This leads naturally into the next major recommendation: the need for advanced computational science support.

C. Computational Science Resources

Often discussions of the needs of computational science devolve into discussions of hardware: processors, memory, disk space, bandwidth. These are obviously important, but not the only areas of investment in support of computational science. One of the clear conclusions of the Task Force is that now and in the future the scarce resource will be skilled people. They are what is truly important and the most valuable component in computational science. To this end we propose the establishment of the University of Virginia Alliance for Computational Science (UVACS).

The University of Virginia Alliance for Computational Science would consist of staff, postdoctoral researchers and graduate students who have expertise in a variety of areas in computational science, including hardware, software, algorithms, application development, and data analysis. These people would collaborate with faculty researchers on projects (most often sponsored research projects) and provide necessary technical expertise. The permanent staff provides a repository of knowledge and experience that will be shared with faculty, postdocs and students both through new research collaborations and through the education program. Staff would remain up-to-date on available hardware and software solutions, and their expertise would be called upon in developing grant proposals. Oversight for general directions and activities of this group would be provided by the faculty advisory committee, and the staff would report to the director for computational science.

A list of the resources that UVACS will coordinate and provide include:

1. Appropriate access to the appropriate processors, from desktop machines to supercomputers

2. Expertise in algorithms and programming

3. Use of scientific software and discipline-specific applications

4. Networking, including access to very high speed national backbone networks and Grid computing

5. Data repositories, data security and access, and data analysis

6. Display, including advanced visualization

7. Education and training

We briefly discuss each of these in turn. Because of the specific importance of education and training it is described as a distinct goal below. The group described here will, however, be important in enabling the education and training program.

1. The University should ensure local availability of the appropriate high performance computing hardware and software in support of computational science research. "Appropriate" in this context refers to a level to be determined, but which would lie between the "commodity" computing resources that each faculty member can reasonably expect, and the advanced resources available from partner organizations and national centers. One primary aim of local hardware resources will be to enable the development and testing of applications that may then migrate to more advanced facilities. Another goal would be to possess sufficient advanced computational hardware as is needed to support the education and training on advanced systems. Hardware and software resources may be obtained and identified as exclusively dedicated to these missions, or they may be identified from otherwise available university resources. Regardless, it is anticipated that such resources would be supported, maintained, and administered by ITC.

Researchers who require more significant processor resources should have well-identified routes to obtaining them. We envision that the University will develop relationships with partner organizations that have high-performance computational facilities, such as other SURA institutions, government labs (e.g. ORNL), and corporate technology partners. The group would identify opportunities, and help to develop proposals and requests to national facilities for advanced computing resources. Finally, there should be Cost Accounting Standards compliant procedures for obtaining resources through direct cost expenditures.

2. & 3. A primary responsibility of UVACS would be to provide algorithmic and programming expertise necessary to take advantage of high performance computing, both hardware and software. This would include code development, optimization, and parallelization. Consulting would be provided for computational science applications such as problem solving environments (Matlab, Mathematica, Maple), scientific libraries (IMSL, Scalapack), and discipline-specific applications (finite element, molecular dynamics, computational fluid dynamics, and the like). Applications support would include both use of specialized software packages (e.g. Ansys, Amber, and other standard programs) and original development by a research group in a compiled language.

4. Effective use of advanced networks such as the Lambda Rail requires expert support. For example, the staff would work with researchers at U.Va. and other institutions to develop, deploy and employ Grid software over regional and national scales. This effort would ensure that University researchers will be part of large scientific collaborations where pooled computing resources and data form an effective computing system across a wide-area network.

5. Large and complex data sets are growing rapidly, and raise issues of data repositories, data security, integrity, and access, knowledge extraction and analysis tools. Many disciplines share a common set of database challenges that can be addressed by advanced analytics applications (e.g. data-mining). The ultimate objective is to increase the effectiveness of scientific data understanding by providing the data

management underpinnings of analytic and visualization technology. The staff would provide the expertise in scientific data management to help achieve that objective.

6. The need for advanced visualization and display spans the full range of computational science activities and requires its own set of hardware and technical expertise. Hardware, software, and expert support are all required to enable advanced visualizations and a collaboration environment for computational science and discovery. The UVACS staff would provide expertise in data analysis and visualization, image processing, and management of the visualization facility.

The previous list of services can be categorized into four primary resource areas: high performance computing, scientific visualization, data management, and grid computing. Each area would require a primary support staff member and secondary support staff member. Most staff members would cover more than one area. Even so, it is anticipated that at least a dozen staff members would be necessary to adequately cover the four resource areas and work on various sponsored research projects.

Finally, the activities of UVACS should occur within a dedicated space, preferably within the Engineering and Sciences area of the Grounds. We recognize that this presents immediate practical difficulties at present. Space is at a premium and is tightly controlled by departments and schools who may be reluctant to cede space to inherently interdisciplinary activities such as this. Nevertheless we wish to affirm the principle that a professional activity such as proposed here needs quality work and meeting areas located near the researchers with whom they will partner.

D. Education, Training and Community Development

Despite the widespread perception of the importance of and potential for computational science, relatively little has been done to prepare students and researchers for the future. Institutional inertia undoubtedly plays a role, as does the interdisciplinary nature of computational science, in slowing the development and implementation of new instructional programs. As the PITAC report notes:

> ***...the number of graduates from computational science programs is inadequate to meet even current demand and it is far below the number that will be needed in the future....It is past time for universities to take action. They must examine their educational practices and organizational structures to provide and reward interdisciplinary and collaborative research and education. New structures, programs, and institutional incentives are urgently required.***

Even if the University initially could only partially underwrite the other requirements of computational science for researchers at the University, it is imperative to our fundamental duties as an educational institution to provide our students with the education and training they will need to be effective researchers, scientists and engineers.

The development and implementation of a computational science curriculum is complicated by the inherent intertwining of general interdisciplinary methodologies with discipline-specific applications. Individual departments might be prepared to teach the latter, but would find it difficult to do so without the former. Developing courses in general topics such as programming is usually not seen as appropriate or cost effective within a department. Such structural inhibitions can be overcome, however, with coordinated planning among departments and schools toward the goal of developing a formal program in computational science.

A first proposal for the development of a computational science curriculum is presented in the June 2006 proposal to the NSF, ``CI-TEAM Demonstration: Towards a Culture of Computational Science," A. Grimshaw PI, J. Hawley and S. Dexter Co-Is. The proposal development process is described briefly in Appendix B. The body of the proposal is attached as Appendix C. In summary the proposal calls for the development of two new courses: (1) a two semester, integrated undergraduate course to familiarize students with cyberinfrastructure and high performance computing, and (2) of a two semester, integrated graduate course in computational science to introduce beginning researchers to computational science and engineering. The unique aspects of these proposed courses include the development of a scalable curriculum that will include both general and discipline-specific components. Both course sequences will emphasize team projects to promote interdisciplinary collaboration, and will expose students to faculty and other students with common interests in computational science. The team-teaching style of the courses will also promote interdisciplinary collaboration and dissemination of knowledge to departmental faculty.

In the end the University might choose to develop a degree-granting program in computational science; there is a clear path of gradual development that leads toward such a goal. We might begin with several cross-listed courses taught by faculty from several departments that deal with the broad topics in computational science. Such courses might either lead directly to discipline-specific courses taught within departments, or may themselves incorporate aspects of such applications. These initial courses could provide a certification in computational science to graduate students enrolled within specific disciplines. As additional courses are developed this program could be expanded into a graduate minor in computational science, in conjunction with an existing degree program. Eventually, as experience is gained and the curriculum grows, it would be possible to develop a full degree program in computational science.

As a concrete example we describe a possible Interdisciplinary Graduate Minor in Computational Science. The goal of the Graduate Minor program is to establish a formal way for training the new generation of computational scientists and engineers with a solid background in the following areas:

* Computational mathematics and numerical methods;

* Scientific programming, general-purpose computational and visualization software;

* Modern high performance computing platforms and parallel programming;

* Discipline-specific computational methods and modern state of the art.

The first three areas deal with the aspects of computational science that are similar across disciplines. Currently, the students working on computational research projects are filling the gaps in these areas mostly by self-education, while taking discipline-specific graduate courses. The efficiency of such self-education is low. Moreover, the lack of systematic training leaves significant gaps in knowledge and understanding of the basic concepts and often leads to misconceptions.

The graduate minor program would provide students with the background needed for effective computational research and would better prepare them for future work in the quickly expanding and highly interdisciplinary area of computational science and engineering.

More specifically, the following academic requirements may be considered for the minor: M.S. degree minor: 9 credits with at least 6 credits from the core. Ph.D. degree minor: 15 credits with 9 credits from the core. The core courses include "Computational mathematics and numerical methods," "Scientific programming and algorithms for high performance computing," "Parallel computing for non-CS majors" and "High performance computing platforms." Additional discipline-specific courses are chosen from the approved list of courses with significant computational component. Examples of graduate computational courses currently offered at the University of Virginia are Computational Fluid Dynamics, Numerical Methods in Hydrology, Modeling in Materials Science, Computational Systems Biology, Computational Physics, Applied Numerical Methods in Mechanical Engineering.

In addition to providing a basic graduate-level educational background to students working in a general computational area, the benefits of the Minor in Computational Science (MCS) would include the following.

* Close contacts among graduate students working on diverse problems would naturally result in collaborations among different computational groups, sharing knowledge across disciplines and, eventually, formation of a vibrant informal computational science community at the University of Virginia;

* The minor would provide an effective way to start a serious computational effort in a research group/laboratory that does not have ongoing computational activities;

* The quality of the computational research would be facilitated by the solid and systematic knowledge of graduate students, leading to the recognition of the University of Virginia as a player in the computational area, expanded funding opportunities, etc.

* Having a minor in computational sciences would improve job opportunities for students;

* Depending on the success of the minor, establishment of a dedicated Computational Science graduate program may be considered in the future.

While developing a formal curriculum in computational science for our graduate and undergraduate students is a major goal of this initiative, it is also important to develop a less formal program of training for a wider community including students, postdoctoral fellows, and faculty. These individuals do not need course credits, but they still need education and training. Many researchers would certainly upgrade their skills if the information were made readily available to them in a convenient format. Thus the educational component of this initiative should include a regular program of courses, lecture series, mini-courses, workshops and a visiting seminar series. The content would vary widely, from discipline specific material, to advanced topics in computational science such as modern software environments, efficient use of HPC platforms, and other subjects of use to scientists who must develop and apply advanced applications as an essential part of their research. These courses would be taught by faculty, staff from the core computational science group, and other researchers drawn from existing areas of expertise within the University's science and engineering community. Seen more broadly, this program is a "community development" effort, to foster a culture of computational science in the sciences and engineering at the university.

3. SPECIFICS AND DEPLOYMENT PATH

This report has laid out the importance of an initiative in computational science in support of many of the University's strategic goals. In this section we discuss concrete steps that can be taken in the near term to begin to achieve the outcomes described above. This report alone cannot completely describe a successful University-wide program to support and enable world-class computational science. Instead we offer an outline of a possible deployment path along with some concrete suggestions for initial steps.

The following steps should be taken as soon as possible:

*** Identify faculty involved in computational science who could form a core of experts**

*** Form a faculty advisory committee for computational science**

*** Identify a sponsored projects development officer for computational science**

*** Identify existing expertise and interest in the core areas of UVACS support, such as visualization. Develop collaborations and communication among existing groups at the University**

*** Appoint a director as soon as possible - there are opportunities to be pursued**

and there is an immediate need for leadership if we are to achieve the other goals of this initiative

*** Identify existing expertise within ITC staff who could form the initial core computational group that would become UVACS**

*** Expand and extend existing workshops (presently given by the Research Computing Support Group of ITC) on topics in computational science**

*** Locate a suitable space to site the initial UVACS group**

*** Implement the activities of the NSF CI-TEAM proposal for a demonstration project in computational science education.**

*** Commence an interdisciplinary seminar series in computational science to build community**

*** Integrate these proposed activities into the vision for the Capital Campaign and for the University's long-range planning.**

*** Identify components of this initiative that would be suitable for specific funding opportunities within the Capital Campaign**

*** Actively pursue interdisciplinary funding opportunities in computational science and engineering. Identify procedures for providing matching or seed funds in support of sponsored research. Encourage inclusion of direct-cost support for UVACS staff.**

The next steps would follow naturally as resources become available, and as direct support through grants is applied. Of all the recommendations made herein, the task force consensus was that leadership is key. By establishing a plan and a framework for growth the University can develop a leading and active computational science community with minimal initial investment or risk.

4. CONCLUSION

The U.Va. Task Force on IT Infrastructure Supporting Research which was created in January 2006 to consider possible responses to the growing problems with computational science research support at the University. These problems include (among others) a lack of resources inhibiting ongoing research efforts, difficulty in retaining faculty with computational science research programs, lack of competitiveness in recruiting at the faculty, postdoc and graduate student levels, a decrease in the competitiveness of research grant proposals, and a difficulty in responding to interdisciplinary funding opportunities with strong computational science components. While initial discussions began with issues of hardware and infrastructure, the Task

Force quickly perceived that there were larger systemic issues that needed to be addressed. Computational science is an emerging discipline that is increasingly requiring specialized expertise and training more than specialized hardware. It is an inherently interdisciplinary activity that calls for the development of a ``culture of computational science" within the University.

In this report we have presented the arguments for developing such a culture at the University along with concrete steps that can be taken and some of the outcomes we can expect from taking them.

In the Introduction to this report we quoted the Principal Finding of the PITAC report regarding the strategic importance of computational science and the need for action. We conclude with one of their final recommendations:

Given all that depends on the field's vitality, it is imperative that the leaders in academia and the Federal government who are responsible for assuring the continued health of computational science spearhead the design and implementation of new multidisciplinary research and education structures that will assure the United States the advanced capabilities to address the 21st century's most important problems. In addition, the Federal government, in partnership with academia and industry, must commission --- and execute --- a multi-decade computational science roadmap that will direct coordinated advances in computational science and its underlying technologies, paving the way to greater breakthroughs in the many disciplines that will require these capabilities in the years ahead.[2]

What is true for the nation as a whole is no less true for any institution that wishes to participate in this future.

REFERENCES

1. NSF's Cyberinfrastructure Vision for 21st Century Discovery.

2. President's Information Technology Advisory Committee (PITAC) Report, "Computational Science: Ensuring America's Competitiveness," June 2005, http://www.nitrd.gov/pitac/reports/20050609_computational/computational.pdf

3. Final Report on Research Computing at the University of Virginia, March 2001, http://www.itc.virginia.edu/rctf/finalreport.html

4. SIAM Report on Graduate Education for Computational Science and Engineering, http://www.siam.org/students/resources/report.php

5. "Simulation-Based Engineering Science," Revolutionizing Engineering Science through Simulation, May 2006. A report of the National Science Foundation Blue Ribbon Panel on Simulation Based Engineering Science. http://www.ices.utexas.edu/events/SBES_Final_Report.pdf

APPENDIX A: Initiatives at other Institutions

Includes:

- Virginia Tech
- Princeton
- Cornell
- University of California, San Diego
- Penn State
- Harvard
- University of California, Los Angeles
- University of California, Santa Barbara
- University of North Carolina, Chapel Hill
- University of Texas, Austin
- University of Washington

<u>Introduction</u>

As part of the work of the task force, we surveyed the computational science activities at other institutions. The results are summarized in this Appendix. The following programs are described in terms of support for computational science research first, and then support for education in computational science second. The information for each of the schools was gathered by searching the school websites for terms relevant to computational science. The more detailed information for some schools, e.g., Virginia Tech and Princeton, were based on interviews/correspondence with principals involved with that school's computational science programs.

The aspects of support for computational science research that are considered are:

*Providing high performance computing platforms and system administration services
*Providing advanced visualization resources
*Providing user support for computational science application software
*Collaboration in applying computational science expertise to a domain-specific problems
*Fostering interdisciplinary collaborations, key to solving the most complex problems

The aspects of support for computational science education that are considered are:

*Training in the use of computational science application software
*Seminars and mini-courses on subject areas in computational science
*A formal curriculum in computational science
*Foster a culture/community of computational science

VIRGINIA TECH (Summary based on interview with Terry Herdman, Associate Vice President for Research Computing)

The research computing initiative at Virginia Tech is based in part on recommendations put forward in two separate task force reports beginning in Spring 2004. These recommendations were (1) create a new position, a director for research computing, who would oversee computational science efforts throughout the university and who would report to the CIO and to the VP for research; (2) create a new domain-specific application support group that would eventually consist of 30 to 40 people including professional staff, postdocs, and graduate students. The proposed budget for this initiative was $4 million per year; (3) Create a truly interdisciplinary education program including a certificate-granting graduate program in computational science, and a visiting lecture series including workshops, seminars and colloquia.

Recommendation (1) was achieved through the appointment of Terry Herdman to the position of associate VP for research computing. It was considered important that the person in this post be first an experienced computational science researcher, and that he have a title high enough in the organization to transcend schools and departments; that is, the position should be truly at a university-wide level. He has subsequently led efforts to implement the remaining recommendations. A Faculty advisory board is planned but not yet implemented. The research computing support group has begun modestly with approximately three staff people who are strategically employed to advance specific research projects. Additional staff are planned for the immediate future. A highly successful visiting scholar program was carried out to promote the development of the computational science community at Virginia Tech. The initiative to create a graduate program in computational science is currently stalled awaiting resolution of faculty and school issues related to creating an interdisciplinary program.

PRINCETON UNIVERSITY Computational Science Activities
(from phone interview with Professor James Stone, Department of Astronomy and PACM)

PICASso -- Program for Integrated Computer and Application Sciences

Initially funded by the NSF under the IGRET program for developing graduate education, this initiative is focused on developing courses, lecture series, minicourses, workshops, and visualization in computational science. The topics of the short courses vary widely, from discipline-specific material, to parallel computing using MPI, object-oriented programming, etc. Graduate credit is offered for these courses and a certificate in computation can be awarded upon completion of certain courses. These courses are taught by faculty, experts from the central IT program, researchers from the Princeton Plasma Physics Lab, and others. The program is coordinated by a director (who serves as the PI on the NSF grant) and a CS postdoc funded by the grant who is devoted half time to coordinating the activities. These activities are very useful for students and postdocs in particular, and may be the most important occurrence at Princeton in the area of computational science. This program is a ``community development" effort, to nurture a culture of computational science and provide needed training.

PICSciE -- Princeton Institute for Computational Science and Engineering

This program was developed by Professor Jerry Ostriker while he was Provost; he now serves as the director. The goal is to enhance research computing and to foster interdisciplinary research by providing resources for faculty, including hardware and systems. No new faculty lines are associated with this program but it does provide seed funds, and start-up funds for new computational hires within departments. Recently they have acquired a BlueGene system, a large Linux cluster, and an SGI Altix. They will have space in a new wing being built in the science library. PICSciE fills the role of a clearing house for grants and can supply matching funds to increase competitiveness. In addition to the director, there is an executive committee made up of faculty experts, and a system administrator seconded from the central IT organization. Currently there are no programming or research specialists within the institute although there is strong desire to hire such people.

PACM -- Program in Applied and Computational Mathematics

This program, while not a full department, functions like one and offers official university courses. It reports to the provost's office like other departments, and the budget is drawn directly from the provost. Joint appointments between this program and other disciplines are made.

CORNELL

Computational Science Research: The Cornell Theory Center (www.tc.cornell.edu) is a former NSF-sponsored national computing center and is now an interdisciplinary center focused on providing cyberinfrastructure services and research to researchers and educators. Cyberinfrastructure includes high-performance computing, data archiving and mining resources, high-speed networking, Web-based computing, visualization and applications expertise. The Cornell Theory Center (CTC) provides high-performance computing resources to advance and facilitate research in agriculture, genonmics, biology, materials science, and data-driven science. Part of CTC's mission is to create and support cross-university research alliances that rely on high-performance computing to achieve major advances. It also hosts a CAVE immersive visualization environment. There are approximately 40 members of the CTC staff.

Computational Science Education: Cornell offers a minor in computational science and engineering for Ph.D. students (www.cis.cornell.edu/cse/). The new associate Dean of faculty of Computing and Information Science wants to insure all students receive appropriate training computational science (http://www.cis.cornell.edu/cse/cse_wp2.htm). The Cornell Theory Center also offers extensive education and training in the use of Windows based HPC tools, as well as public outreach to K-12 education (www.tc.cornell.edu/services/education/).

UNIVERSITY of CALIFORNIA SAN DIEGO

Computational Science Research: The San Diego Supercomputing Center (www.sdsc.edu/) is an organized research unit of the University of California, San Diego primarily funded by NSF. As such, it hosts some of the most advanced HPC resources nationally. They support advanced visualization and collaboration in their Synthesis Center (www.syncenter.org/). The Strategic Applications Collaborations (SAC) and Strategic Community Collaborations (SCC) programs at SDSC are to enhance the effectiveness of computational science and engineering research conducted by academic users nationwide who use the SDSC HPC resources (www.sdsc.edu/user_services/sac/). The SDSC serves as a critical IT partner to large scale projects in life sciences, geosciences, engineering and other disciplines (www.sdsc.edu/projects.html) and employs nearly 400 staff.

Computational Science Education: The SDSC provides the training classes and workshops introduce new and current users to the high performance computing and visualization resources available at SDSC; and provide the programming skills necessary to use SDSC resources effectively and efficiently (www.sdsc.edu/user_services/training/). SDSC also supports and is aiding development of Computational Science Curriculum in various UCSD departments. It sponsors a computational science seminar program (www.sdsc.edu/CSSS/) and provides educational outreach to K-12 students and teachers (http://education.sdsc.edu/).

PENN STATE

Computational Science Research: The Graduate Education and Research Services (GEaRS) group is a part of a unit of Information Technology Services (http://gears.aset.psu.edu/). It is composed of three service groups: the High Performance Computing Group, the Visualization Group, and the Special Projects Group. The GEaRS group also provides seminars on HPC and visualization tools.

Computational Science Education: The Institute for Computational Science (www.ics.psu.edu) was created to foster educational and research activities in computational science, and to coordinate the graduate minor degree program in Computational Science (www.ics.psu.edu/minorCS.html). The Institute has a faculty advisory board of which the Institute Director is the chair. It sponsors a computational science seminar series and an annual university-wide 'Computation Day'.

HARVARD

Computational Science Research: The Initiative for Innovative Computing (http://iic.harvard.edu/) was created in response to Harvard Provost's 2004 Task Force on Science and Technology. The Initiative in Innovative Computing (IIC) uses computational resources to address issues at the forefront of data-intensive science and focuses on interdisciplinary collaborations that span traditional academic boundaries. The IIC strategy is to concentrate on adding cadre of experts on the needs and techniques of scientific computing to core Harvard staff, rather than on purchasing HPC hardware. IIC will have 6 program areas: analysis and simulation, instrumentation, visualization and communication, distributed computing, and databases and data provenance. A faculty advisory board accepts proposals for computational science projects requiring collaborative expertise, and oversight is provided by faculty steering committee. Core staffing level 20-25 in 2005 (virtual institute), will expand to 90-100 in 2010 (its own physical location).

Computational Science Education: The IIC provides a departmental cross-listing of computational science courses (http://iic.harvard.edu/courses.php) and hosts an invited speaker seminar series (http://iic.harvard.edu/events.php). It plans to offer new courses to both undergraduate and graduate students in data-intensive science. The IIC is in the initial planning stage for implementing a range of outreach programs for teachers, high-school students, and others that will illustrate of advances computational science and demonstrate how they impact society, today and tomorrow.

UCLA

Computational Science Research: The Academic Technology Services Division (www.ats.ucla.edu/default.htm) is jointly administered by the Vice Chancellor for Research and the Office of Information Technology. It is composed of three groups: Research Computing Technologies, the Visualization Portal, and Statistical Consulting, with approximately 40 staff members (www.ats.ucla.edu/About/). The Research Computing Technologies group (www.ats.ucla.edu/rct/) provides HPC support and training, as well as grant development support. The Visualization Portal staff support various scientific visualization activites.

The Institute for Digital Research and Education (IDRE) was recently established to elevate UCLA's position in computational and digital research and education by providing an integrated focus and critical infrastructure (www.idre.ucla.edu). Specifically, it will act as a catalyst for faculty from across departments and schools to form teams focused on solving scientific problems of national significance, and be a force for attracting the best faculty and graduate students in computational science and digital media. In addition, the IDRE will build up local computational resources and provide better access to national high-end computational platforms, as well as provide a facility for code sharing, development, and maintenance.

Computational Science Education: The IDRE plans to develop and coordinate curriculum across UCLA in the areas of computation, simulation, visualization, and the digital arts. It will provide a strong intellectual environment for faculty, researchers, and students though international workshops, seminar series, and conferences. It has already begun a bi-monthly lecture series.

UCSB

Computational Science Research: The Office of Information Technology maintains the

Supercomputing Division (www.oit.ucsb.edu/computing/supercomputing/) to helps UCSB faculty, researchers, and students access and use the High Performance Computing resources available from national supercomputing centers. Services include consultation on planning and architecture, help with establishing accounts, user and system support, and training.

Computational Science Education: UCSB offers an interdisciplinary Master's and Ph.D. degree emphasis in Computational Science and Engineering (CSE), in collaboration with the Departments of Chemical Engineering, Computer Science, Electrical and Computer Engineering, Mathematics, and Mechanical and Environmental Engineering (www.cse.ucsb.edu/). The CSE emphasis offers a broad multidisciplinary educational experience with strong foundations in both the enabling technologies of computer science and applied mathematics and in cutting-edge applications in engineering and science. The CSE emphasis provides both depth in a traditional academic area, and breadth in CSE technologies.

UNC, CHAPEL HILL

Computational Science Research: The central information technology organization for UNC is Information Technology Services. Within ITS is the division of Research Computing (http://its.unc.edu/reco/), which develops and maintains computing infrastructure for research support. This group provides user support and training for the HPC resources.

More recently, Daniel Reed, the Vice Chancellor for Information Technology and Chief Information Officer for the University of North Carolina at Chapel Hill since June 2004, has launched the Renaissance Computing Institute (http://www.renci.org/) as a major collaborative venture of Duke University, North Carolina State University, the University of North Carolina at Chapel Hill and the state of North Carolina. Primary components of its mission are to foster multidisciplinary collaborations with scientific communities, research institutes, businesses, government agencies, humanities and social science scholars, students, underserved audiences, artists and educators across the state of North Carolina, and to enable advancements in science, industry, education, the humanities and the arts by developing and deploying world-class computing resources and tools for data analysis, visualization and collaboration.

Computational Science Education: The Renaissance Computing Institute is involved in education and outreach through participation in Engaging People In Cyberinfrastructure (EPIC) and the Minority Serving Institutions (MSI) Network (www.renci.org/projects/epic.php). UNC also has a Bioinformatics and Computational Biology Training Program (BCB) that is a certificate granting program which works with various degree granting UNC departments to offer a specialization in Bioinformatics and Computational Biology, in addition to the coursework of a student's home department (http://bcb.unc.edu).

TEXAS

Computational Science Research: The Texas Advanced Computing Center (TACC) is a research center at the University of Texas at Austin reporting to the Office of the Vice President for Research (www.tacc.utexas.edu). TACC deploys and operates advanced computational infrastructure to enable computational research activities of faculty, staff, and students of UT Austin. TACC also provides consulting, technical documentation, and training to support users of these resources. TACC conducts research and development activities to produce new computational techniques and technologies that enhance the capabilities of advanced computing resources, and collaborates with computational researchers to apply advanced computational techniques in their research activities. Its main focus areas are high performance computing, scientific visualization, data information systems, and distributed & grid computing. TACC employs approximately 60 staff members.

The Institute for Computational Engineering and Sciences (www.ticam.utexas.edu) was created to function as an interdisciplinary research center for faculty and graduate students in computational sciences and engineering, mathematical modeling, applied mathematics, software engineering, and computational visualization. Organizationally, the Institute for Computational Engineering and Sciences (ICES) reports to the Vice President for Research, and draws faculty from seventeen participating academic departments. The Institute currently supports several research centers and numerous research groups, including the Center for Computational Geosciences and Optimization, the Center for Computational Materials, Center for Computational Molecular Science, Center for Distributed and Grid Computing, Center for Numerical Analysis, Center for Subsurface Modeling, and the Computational Visualization Center.

Computational Science Education: The ICES supports the Computational and Applied Mathematics Program, a graduate degree program leading to the M.S. and Ph.D. degrees in Computational and Applied Mathematics (www.ticam.utexas.edu/programs/). The ICES also supports a weekly seminar series (www.ticam.utexas.edu/seminars/).

U. WASHINGTON

Computational Science Research: The University of Washington has requested $3.0 million in funding from the state for a university-wide initiative to establish an interdisciplinary "E-Science Institute" which will house faculty, research scientists and graduate students with expertise in storing, organizing, mining, visualizing and interpreting such large volumes of data, what they refer to as "data engineering". The proposed institute objectives are to stimulate collaborative research, provide consulting in data engineering to scientists for research and proposal writing, provide Ph.D. level staff to work on data engineering aspects of science projects, and to provide education in data engineering to students, postdocs, faculty through regular courses, short courses, seminars, and workshops.

Computational Science Education: The proposed E-Science Institute would offer courses in computational science generally and data engineering specifically.

APPENDIX B: Examples of the Potential of Computational Science

----

In silico drug response prediction of 45,545 compounds to human bladder cancer using NCI-60 drug potency data and a novel co-regulation expression signature algorithm

Jae K. Lee, Ph.D.
Associate Professor of Biostatistics and Epidemiology
Director, U.Va. GeneChip/Microarray Bioinformatics Core, Department of Public Health Sciences

Since 1990, the Developmental Therapeutics Program (DTP) at the NCI has undertaken high throughput screening of novel compounds for the development of cancer chemotherapeutics. Thousands of compounds have been screened on a standardized cell line panel of 60 human cancer cell lines (NCI-60). However, this cell line panel does not include representation of a number of important tumor histologies such as urothelial bladder cancer which comprises the 5th most common cancer in the US and the most expensive cancer to manage overall. In an effort to address this limitation we have developed a novel computational algorithm that allows retrospective in silico screening of compounds previously evaluated in the NCI-60 assay for their effectiveness in bladder cancer. Using microarray profiling data from NCI-60 and a panel of 40 human bladder cells lines, here we describe a novel method that identifies a set of drug co-regulated chemosensitivity response biomarkers and use these and the Misclassification-Penalized Posterior (MiPP) method to design classification models that predict de novo drug responsiveness in a bladder cancer cell lines. Our model was validated by demonstrating 85-90% prediction accuracy of in vitro chemosensitivity of bladder cancer cell lines to cisplatin and paclitaxel. Finally, we employ this methodology for in silico cluster-computer screening of 45,545 compounds in the NCI database and identify not only those chemotherapeutics currently in clinical use in bladder cancer but also lead compounds that have significantly superior activity than the latter. In summary, this work provides a list of novel anticancer drug candidates for human bladder cancer with putatively superior effectiveness than current agents and by virtue of its applicability to any tumor type, greatly leverages the national investment in cancer drug screening.

Computational Materials Group, Department of Materials Science and Engineering, SEAS Leonid V. Zhigilei, lz2n@virginia.edu, Web page: http://www.faculty.virginia.edu/CompMat/

The research activities in the Computational Materials Group are focused on the development of new computational methods for materials modeling and application of these methods for numerical and theoretical analysis of the dynamic non-equilibrium processes in materials undergoing processing by short laser pulses, investigation of the microscopic mechanisms of phase transformations, properties of nanostructured and non-crystalline materials. One example project illustrated in the Figures below is focused on obtaining a fundamental understanding of the fast non-equilibrium processes induced by short pulse laser irradiation of metal surfaces as well as on the analysis of practical implications of the revealed physical picture for the advancement of laser technologies. In this program we develop a multi-scale computational model based on large-scale molecular dynamics simulations coupled with a continuum description of the laser light absorption by the conduction band electrons, the energy transfer to the lattice due to the electron-phonon coupling, and fast electron heat conduction. The model is applied to study the microscopic mechanisms of melting and recrystallization occurring under extreme superheating or undercooling conditions realized in short pulse laser processing, mechanisms of laser-induced plasticity, photomechanical damage, and ablation. In most of our simulations we use atomic-level models implemented in our own MPI parallel computer codes. A reliable description of the collective atomic motions induced by the fast laser energy deposition requires large computational cells that include from ~100,000 to several million atoms. Additional computing resources would allow us to expand the length-scale of our simulations up to a hundred of nanometers (hundreds of millions atoms), which would open up many exciting research opportunities. These include investigation of the microscopic mechanisms of crystal plasticity under extreme conditions of ultrafast laser induced deformation, analysis of the evolution of the shapes of liquid nuclei in an overheated crystal and testing the applicability of the homogeneous nucleation theory to the description of the nucleation of a new phase far from equilibrium.

DOC FILE INCLUDES IMAGES

Black Hole Astrophysics
John F. Hawley, Dept. of Astronomy, University of Virginia

For more than 30 years, astrophysicists have believed that black holes can swallow nearby matter and release a tremendous amount of energy as a result. Until recently, however, the mechanisms that bring matter close to black holes have been poorly understood, leaving researchers puzzled about many of the details of the process. Now, however, computer simulations of black holes are answering some of those questions and challenging many commonly held assumptions about the nature of this enigmatic phenomenon.

Researchers at the University of Virginia, along with collaborators at other institutions, have been developing and applying finite-difference computational codes for simulations of compressible magnetohydrodynamics (MHD) to astrophysical phenomena. One of our main research thrusts is the simulation of accretion disks around rotating black holes. The ultimate aim is to relate the fundamental physical processes that occur in the disk to the detailed observed time-dependent spectra. Achieving this goal will require substantial grid resolution over a large physical domain, from the black hole horizon outward over five orders of magnitude in radius. Long-term simulations are required as the orbital timescale increases as Kepler's law, namely orbital radius to the 3/2th power.

The primary resource drivers are the need for high spatial and temporal resolution, as well as the complexity of the physics and the simulation codes. An estimate of the resource required can be obtained from the product of the number of grid zones, times the number of time steps, times the number of floating point operations per time step per grid zone. Present simulations, using restricted radial and angular spatial domains (about 10^6 grid zones) and run for limited times (about 10^6 timesteps), and using only ideal MHD physics, require roughly 10^15 floating point operations. Large datasets (hundreds of GBytes) are typically generated that require significant post-processing analysis.

Advanced visualization is essential. The present simulations are sufficient for preliminary explorations of global dynamics, but not to connect to observations. Essential new physics includes radiative processes and radiation transfer, and photon tracing through the general relativistic spacetime. Along with increased resolution, these requirements easily boost the number of required flops by several orders of magnitude. However, as the capabilities of national resource systems approaches the petaflop level, we can anticipate that the simulations will finally achieve predictive modeling.

NICE IMAGES AVAILABLE
