Proposal for a Graduate Level Minor in Scientific Computation at Indiana University

October 10, 199fi

Scientific Computing Program Steering Committee

1 Introduction

Indiana University has begun an interdisciplinary, interdepartmental program in Scientific Computing. The program is a formal recognition of changes in science that have been prominent for the last 15 years, changes that have introduced a powerful and entirely new mode of research. The increasing availability of high performance computers has led to a method of scientific inquiry based on mathematical models solved by means of numerical computations, analyzed and viewed by means of advanced computer graphics. Carrying out research by these means is necessarily interdisciplinary, calling on advanced skills in areas that span many classical divisions of academia. However, the current Ph.D. system inhibits Ph.D. students from taking courses in more than one other field, their minor area. We propose the creation of a new Ph.D. minor program, allowing the interdepartmental education necessary to better equip our students to carry out research within this new paradigm.

2 Scientific Computing

Scientific computation is the development, study, and implementation of methods for solving engineering and science problems on computers. It draws upon, but is distinct from, “computer science” which studies issues such as the mathematics of information transmission, storage, and processing, the modeling of human reasoning, complexity theory, and principles of software design. Although these issues can be important in some scientific computations, the concentration here is on the *overall* process of computer solution of problems. As a simplified example, solving a fluid dynamics problem requires several areas of expertise. Formulating the physical problem usually lies in physics, chemistry, astronomy, or geology. Deriving a mathematical model is a classical mathematics problem, and selecting a solution process lies in engineering or mathematics. Choosing algorithms to numerically solve the mathematical model is in the area of numerical analysis, while selection of computer language and details of implementation of the resulting algorithms is traditionally a task for computer science and engineering. Validation and analysis of the results are in the field of statistics and experimental science, while display of the results is a problem in the area of engineering, human perception, and art.

Twenty years ago it was standard for a scientist solving a problem to master all the relevant areas; the choices and complexity of solution methods on all the levels were relatively limited. Now, it is no longer reasonable to expect an expert to be familiar with all of the necessary fields of research involved. Each level has become a separate research area, increasingly insulated from the other levels. Unquestionably the results produced on each level have made tremendous contributions to science, technology, and society, but the effective solution of modern scientific problems requires new communication and collaboration between the levels.

At the same time, the complexity and importance of these computational problems has grown. In part this is because computational science has become a new mode for making scientific inquiry that now stands alongside analytical theory and physical experiment as the basis of scientific process. It represents a fundamental change in our view of science. In the classical model of science physical experiment suggested ideas which led to theories which led to predictions which are tested by experiment. As theories have become more complex the process of deriving predictions has become dependent on calculations only possible by computer. Furthermore, as our understanding of nature has expanded, the frontiers of our knowledge have pushed past the easy targets of direct experiment and observation. Now the experiments we need to perform often are not physically practical. For example, what happened in the first few microseconds of the big bang or how is global warming affected by the fires in the Amazon? Consequently science has turned to the computer to serve as a "numerical laboratory" where experiments can be conducted in an artificial environment over which the scientist has complete control. For example, if an astronomer has a hypothesis about a theory about the evolution of stars, his or her theory can be used to drive a simulation which may then disprove the hypothesis in a more direct (and often more conclusive) manner than could be done by searching the heavens.

Every discipline of science has now come to rely on this new third branch of the scientific process in some form. Some new areas, such as nonlinear systems and chaos studies, could not have been born without techniques of computational science.

The growing importance for the world at large of this new paradigm is illustrated by, among other projects, the High Performance Computing Program of the Office of Science and Technology Policy. This specifies a list of "Grand Challenges", economic and scientific problems of great potential for the nation. Grand challenge problems being actively targeted at Indiana University include:

- Climate and global change modeling and prediction
- Quantum chromodynamics for high energy physics.
- Turbulence modeling in fluid flows.
- Oil and gas recovery enhancements to help locate and efficiently extract natural resources.
- Speech recognition by machines, to ease the use and programming of computers.
- Understanding earthquakes.
- Ground water modeling for solving environmental problems.

The solution of these problems requires the coordination and collaboration of scientists from many fields, and the training of researchers in the development of the requisite mathematical, software, and hardware tools.

In isolation, classical academic departments are neither prepared nor equipped to educate and train computational scientists. The proliferation of new parallel machine architectures alone has created new fields of research, and the experience of the past 15 years has shown that efficient use of these machines can only be achieved by an integrated approach, beginning with the choice of physical and mathematical models, down through the implementation of computational kernels that effectively use small–scale architectural features such as vector registers and memory caches. A few researchers have achieved great success in using this new mode of scientific inquiry, but there is now a strong need for people systematically educated in a new paradigm. Indiana University has begun an interdisciplinary program in Scientific Computation, with the goal of producing such interdisciplinarians.

3 Benefits of a Scientific Computation Minor

We foresee several benefits from this program, for students, faculty, and the University at large. In particular, it can achieve

- The further encouragement and faciliation of interdisciplinary research in computational science at IU.
- The ability of students to take courses in scientific computing and satisfy the Ph.D. minor requirements, without having to take all the minor courses from just one department.
- Greater visibility that will allow us to attract better students to graduate studies at IU.
- Providing a formal recognition of the interdisciplinary work of graduating students, helping them in starting their careers.
- Providing the training that is currently sought by industry, academia, and national laboratories.
- The ability to attract better faculty candidates in all the involved departments. The hiring of a scientific computing program faculty member in 1993 attracted hundreds of applications, many of them outstanding young researchers.
- More efficient utilization of teaching resources by coordinating the course offerings and their timings, and the ability to offer more varied and in–depth courses by making them practically available to a larger body of students.
- Enhancement of our ability to obtain large grants in interdisciplinary scientific endeavors.

4 Current Scientific Computation Faculty at IU

The Scientific Computation program has been driven by the research and needs of many faculty members at IU. This indicates both the need for the field and the existence of a large number of researchers interested in computational problems as well as a large number of current projects in the field. IU has created two faculty positions specifically in scientific computing, in the Computer Science and Chemistry Departments. A partial listing of scientists involved in scientific computation includes:

- Randall Bramley, Computer Science
- Haldan Cohn, Astronomy
- Ernest Davidson, Chemistry
- Charles Delwiche, Biology
- Richard Durisen, Astronomy
- Kent Dybvig, Computer Science
- Dennis Gannon, Computer Science
- Steve Girvin, Physics
- Steve Gottlieb, Physics
- Andrew Hanson, Computer Science
- Charles Horowitz, Physics
- Mike Jolly, Mathematics
- Alan Kostelecky, Physics
- Allan MacDonald, Physics
- Glenn Martyna, Chemistry
- Greg Olyphant, Geological Sciences
- Peter Ortoleva, Chemistry
- Robert Port, Linguistics-Computer Science
- Gary Pavlis, Geological Sciences
- Vishnu Ranganathan, Geological Sciences
- Brian Serot, Physics
- Peter Shirley, Computer Science
- Jim Swihart, Physics
- Roger Temam, Mathematics
- David Wise, Computer Science
- S.Y. Lee, Physics

Research grants specifically for scientific computing awarded to IU faculty over the last three years total over 5.5 million dollars and several Ph.D. students have completed a degree in the area over the same time period. Table 1 summarizes some of this activity; some grants are joint between several faculty members, but only one principal investigator is listed. When shared, only the portion of the grants directly attributable to scientific computing is shown.

Researcher	Grad Students Directly Involved in Sci Comp	Grant Totals (approx for 3 yrs)
R. Bramley	2	210,000
H. Cohn	2–4	285,000
E. Davidson	5	750,000
R. Durisen	1	100,000
D. Gannon	8	1,300,000
S. Gottlieb	0	200,000
A. Hanson	2	335,000
M. Jolly	2	260,000
P. Ortoleva	9	1,200,000
R. Temam	3	300,000
G. Pavlis	-	520,000

Table 1: Partial List of Student and Grant Activity in Scientific Computing at IU–Bloomington

5 Current Scientific Computation Projects

There are a large number of research projects on campus that are already involved with scientific computation; a few are listed below. They range over a broad spectrum of departments including Astronomy, Chemistry, Computer Science, Geological Sciences, Linguistics, Mathematics and Physics. In addition to these, departments such as Theater, Biology and Economics have projects that potentially will use or already have used computational science methods.

5.1 Fortran Tools, pC++ and Parallel Programming

Professor Dennis Gannon, Department of Computer Science and CICA.

CICA (Center for Innovative Computer Applications) currently has a 3-year, NSF funded project to investigate new ideas in parallel programming technology. The initial goal of this project is to aid several local users in the development of completely parallelized versions of their important applications. However, the major goal of this project is to develop new software tools and algorithm libraries to help in the processes of parallelizing and optimizing scientific computations.

Our Fortran project is designed around a set of tools for performance analysis and prediction as well interactive and automatic parallelization of programs written in FORTRAN77, FORTRAN90, or the PCS and D dialects. The emphasis of this effort is to exploit compiler technology in the performance prediction and analysis process. This information allows the programmer to focus more quickly on the algorithmic and data structure issues that are causing performance problems, but it is also the key to predicting the scalability of a given computation.

In our pC++ project we are investigating a model of programming and an experimental programming language and compiler project that we hope will achieve the goals of portability and scalable performance. Our approach is similar in spirit to the FORTRAN-D project from Rice University, but because it is based on the object oriented language C++, it can take advantage of inheritance and abstraction to provide a radically different view of libraries that encapsulate parallelism. Our goal is to raise the level of mathematical abstraction that scientific programmers work with in the design of their codes. By doing so, the compiler can get closer to the mathematical foundations of the application and do a much better job of managing the parallel resources. In addition, a higher level mathematical formulation enables us to work with high level libraries that are much richer in algebraic and topological structure. These libraries can be optimized by hand for each parallel architecture and easily integrated with the users application code.

Gannon's group has implemented the pC++ programming language and environment on a variety of machines including VAX8800, Alliant FX/8, Alliant FX/2800, and BBN GP1000 systems. Initial results show great promise, but the work on the application areas below will be continuing for the next several years.

5.2 Applications in Geochemistry

Professor Peter Ortoleva, Departments of Chemistry and Geological Sciences, is investigating the use of homogenization theory to study geochemical processes such as the development of petroleum reservoirs in a sedimentary basin. The idea behind homogenization theory is to build numerical techniques which allow phenomena that operate on different spatial or temporal scales to be simulated. Much like multigrid techniques used to solve partial differential equations work by separating different solution wavelengths and solving them on different grid resolutions, his homogenization approach allows certain physical variables to be solved on local fine grids while a large global coarse grid is used to resolve the dynamics on the longer scales.

The principal differences between computation based on homogenization theory and a multigrid scheme is that there are only a few length scales required to describe the geochemical problems and that the fine grid describing the short wavelength dynamics need not be global in extent. The data structures arising from the theory consist of a coarse grid of fine grids. The coarse grid covers the entire domain and resolves the long wavelength behaviors. The fine grids are associated with vertices of the coarse grid and represent a "patch" of short wavelength behavior that is representative of the behavior in the coarse grid cells nearby.

Because of this two-level decomposition, we see that this computation is a natural candidate for parallel execution (the fine grid solutions are all independent and the coarse grid is small enough for a vector solution in one processor). We have already started initial experiments on this computation and developed special graphical tools to visualize the results on our existing parallel platforms.

5.3 Applications in Computational Fluid Dynamics

Professor Roger Temam, Department of Mathematics, is a leading authority on the theory of numerical methods for solving the Navier-Stokes equations. His current projects include the development of new multiresolution algorithms based on some dynamical system concepts, such as the approximation of attractors. This includes the development of the Nonlinear Galerkin method (spectral, pseudo-spectral and finite element discretizations) and the Incremental Unknowns Method (finite differences). The aim and the emphasis is on solving very large problems from computational fluid dynamics.

5.4 Algorithm Development in CFD

Professor Randall Bramley, Department of Computer Science. Programs that solve incompressible flow problems in computational fluid dynamics (CFD) spend most of their time in the solution of linear systems of equations.

Research currently is concentrating on solving these systems more accurately and faster by two approaches. First, parallelism is introduced into the solution process by means of a domain partitioning method. This is needed both when a new parallel approach such as Schwartz decomposition is used, and when existing algorithms are implemented on parallel machines with distributed resources. Partitionings of the physical domain lead to block structures of the linear systems, which can then be mapped to different processors. Currently, automatic methods for partitioning nontrivial three-dimensional problems require good starting dividers for the domain, and can frequently give undesirable decompositions with unnecessarily long interfaces (which increase the amount of communication between processors) or with unbalanced subdomains (which gives poor load balancing between processors). In collaborations with workers in France and Canada, approaches are being developed that will allow existing CFD computer programs (which may comprise tens of thousands of lines of code) to be adapted to parallel machines. These will allow retaining the large investment made in such codes, and since they do not require the introduction of new parallel algorithms. The lengthy and expensive process of validation of the codes is reduced. Secondly, an outstanding problem for incompressible fluid flows (such as those found inside turbomachinery, nuclear reactors, or in chemical engineering plants) is maintaining the *null-divergence* condition on the velocity fields. This condition is a direct consequence of physical conservation laws, and even small violations of it can quickly cause computed solutions to depart from physical relevance. Standard linear solvers either fail to adequately maintain this condition, or are based on assumptions that do not hold in the CFD problems of interest. New algorithms for solving constrained linear systems are being developed that maintain the crucial null-divergence condition to within machine precision. These solvers are applicable to many scientific computation problems which require solving constrained systems of equations.

5.5 Applications in Neural Networks

Professor Robert Port, Departments of Linguistics and Computer Science. The study of perceptual systems that recognize patterns presented only over time is important to speech perception, melody perception and the recognition of brief transient noises like squeaks and chirps (such as those found in birds' songs). Real-time systems are needed to process these patterns as they occur. In addition, it appears that human processing of these patterns is tempo invariant, since complex patterns can be discriminated just as accurately at unexpected tempos as when the tempo is constant Our recent work has demonstrated recurrent neural networks that can process temporal patterns in a rate-invariant way. After training a network on a pattern at a single tempo, the pattern can be recognized despite rate variation without any retraining. This follows since the dynamic representation remains the same in Dynamic Auditory Memory. This work shows that rate-invariant pattern recognition can be achieved by means of a stable rate-independent trajectory through the state space of activations. Such a representation replaces the static 'time-window' method of storing and analyzing temporal patterns employed by both conventional speech recognition systems and other neural-network models. Since systems of this general form do not require an external control mechanism to determine what should be done next, they provide an alternative model for many other dynamic cognitive processes including visual cognition and sentence understanding.

The parallelization of the neural network software will begin by designing a special library of pC++ collections and classes for general multistage recurrent networks. These libraries will be derived from existing optimized linear algebra classes.

5.6 Applications in Radiosity

Professor Peter Shirley, Department of Computer Science, is primarily involved with software for rendering algorithms based on variations of radiosity computations and ray tracing. Our objective here is to provide efficient parallelizations of that work. This is not new to Professor Shirley. While at CSRD in Urbana he wrote one of the first production parallel graphics raytracers, which ran on an Alliant FX/8 and the experimental Cedar System. Our goal is to develop a parallel implementation of Shirley's existing C++ radiosity library. We should be able to see a 100 fold improvement in the rendering times for single frames and animation sequences over the current Silicon Graphics based implementation.

5.7 Applications in Physics

Professor Steven Gottlieb of the Physics Department is interested lattice gauge techniques in quantum chromodynamics. Quantum chromodynamics (QCD) is a quantum field theory that elementary particle physicists believe describes the strong interaction. This force is one of four fundamental forces. The others are electromagnetism, gravitation and the weak interaction. Although we can write down the equations of QCD, they are nonlinear differential equations and we cannot solve them analytically. QCD, like any field theory, has its variables defined on the continuum of space-time.

The most promising approach so far seems to be to approximate the continuum theory by a discrete lattice of points. With a finite, but very large, number of variables we can simulate the system on a computer. There are two types of variables in QCD. In the lattice formulation, we have 3×3 complex matrices to describe the gluons of the theory. There is one matrix for every link of the lattice. These gluons are exchanged by quarks and are responsible for the forces between the quarks. The forces are so strong that the quarks are bound together into particles we observe in nature, such as the proton or pi meson. The quarks themselves are described by three-component complex vectors. The interactions between the quarks and gluons involve products of the matrices and vectors. The gluons have a self-interaction that involves products of the matrices on different links of the lattice. The most time-consuming part of the calculation involves inverting a matrix for the quark field.

The state of the art today including the effects of virtual quark-antiquark pairs is to do a simulation on a $16^3 \times 32$ lattice. The dimension of the fermion matrix is 294,912. The matrix itself is sparse, but the inverse, of course, is not. We generally only need one or a few columns of the inverse. Conjugate gradient is the leading algorithm for doing the inversion. Typically, several hundred iterations are required for convergence. About 600 floating point operations are required for each site of the lattice, for each interaction. In a typical simulation, a few times 10^5 inversions are required to carry out a simulation. If the theory is treated in the "quenched approximation" that neglects the virtual quark-antiquark pairs, a lattice of size $32^3 \times 64$ can be used.

Professor Gottlieb and his research group have ported their software for this computation to the Intel Paragon, Connection Machine CM5, Cray T3D and IBM SP1. Major computations are underway on each of these platforms at NSF and DOE supercomputer centers. A graduate student in computer science is porting the code to C++, which may provide a unique opportunity to test the pC++ software described above because the unique mix of algebraic abstractions used in the computation are a good test of the utility of the new language.

Professors Steven Girvin and A. H. MacDonald of the Physics Department head the Theoretical Condensed Matter Group and have have an active interest in the computational sciences. Their primary physics interest is in the quantum mechanics of many-particle systems. Of particular current interest are high-temperature superconductors, the quantum Hall effect, and electrodynamics of low-dimensional electron gases.

Some of the numerical techniques used are similar to those used by quantum chemists: truncation of the Hilbert space of interest to a large but manageable size based on physical grounds, followed by exact diagonalization of large (typically sparse) Hamiltonian matrices. The path integral formulation of quantum mechanics affords another approach, using techniques similar to those developed in lattice gauge theories of high-energy physics. One discretizes space and time onto a lattice and considers the statistical mechanics problem defined by the set of all possible space-time trajectories of particles on this lattice. Numerical renormalization group methods and finite-size scaling methods are very important techniques used to scale up the results from finite lattices to the physical continuum.

The condensed matter group presently has a cluster of SUN 3/60's and 5 HP-735 workstations and associated X-terminals. Heavy use is also made of the RS-6000 cluster provided by Indiana University, and the supercomputer facilities at NCSA. The lattice problems are amenable to parallel algorithms are being examined for use in the near future.

5.8 Applications in Astronomy (I)

Professor Richard Durisen of the Astronomy Department has two relevant projects, in star formation and transport mechanisms in planetary rings.

5.8.1 Star Formation

The formation of stars and planetary systems involves the collapse of gaseous interstellar clouds from light year dimensions down to the size of our own solar system, a change in length scale by a factor of a more than a hundred thousand. The clouds which collapse to form stars are known to be rotating;

and the importance of centrifugal force grows as the cloud collapses. One goal of star formation theory is to determine how and under what conditions collapse leads to formation of a binary or multiple star system or of a single star with planets. A number of different transport mechanisms and instabilities can play a role in this process. Most of these cannot be detected in action by direct observations, because even the nearest star forming regions are many hundreds of light years away; and so the occurrence and behavior of proposed mechanisms can only be studied by means of computer simulations. For many years, Professor Durisen and his collaborators have investigated the dynamics of rapidly rotating, self-gravitating gas disks and stars with particular attention to global instabilities which may break up a single object into two or more pieces. Simulations of these processes are made with a 3D hydrodynamics code using a cylindrical grid whose axis is the rotation axis. The current version is second order in both space and time; and typical grid sizes for the most rotationally flattened disks include several hundred thousand computational cells. Up to now, computations have been run on high-level workstations and vector supercomputers; but the FORTRAN code is quite suitable for parallelization. An earlier version written in pC++ tested favorably on several parallel machines. Durisen's group is planning to develop a FORTRAN90 version of the code for use on the CM5 at the National Center for Supercomputing Applications.

5.8.2 Transport Mechanisms in Planetary Rings

Observations of Saturn's rings by the Voyager spacecraft in the last decade has revealed an unanticipated wealth of structure in the rings. Through the 1980's, many of the edges as well as some spiral features have been explained by gravitational interactions between the ring particles and either the large external moons of Saturn or kilometer-sized moonlets embedded within the ring system. Nevertheless, the vast majority of edges and bands within the rings remain mysterious. Since 1984, Professor Durisen and his students have collaborated with Dr. Jeffrey N. Cuzzi of NASA-Ames Research Center on the effects of meteoroid impacts onto ring particles. Millimeter-sized meteoroids will strike the icy meter-sized ring particles at tens of kilometers per second, blasting out tens of thousands of times their own mass from the ring particle. The debris will spray tens to thousands of kilometers across the rings. The meteoroid pounding is continuous and, over time, causes a net exchange of material and orbital angular momentum between neighboring ring regions, a process called ballistic transport.

Prof. Durisen and IU students have developed a computer simulation code to model the evolution of the rings due to ballistic and viscous transport. The calculations, although only one-dimensional in space, are computationally intensive because they involve multiple integrals over the speed and direction of the ejected debris. This research has already shown that some edges and radial banding in the rings are likely to be due to ballistic transport. Fits to observed ring structures have implications for the material strength and composition of ring particles, the rate of meteoroid influx, and the age of the Saturn ring system.

5.9 Application in Astronomy (II)

Professors Haldan Cohn and Phyllis Lugger, Department of Astronomy, have been investigating the dynamical evolution of stellar systems with graduate students Robert Grabhorn and Paul Bode. This is an application of the very large N-body problem of gravitational astrophysics. Two different numerical approaches are being used: (1) direct integration of the Fokker-Planck kinetic equation which follows the evolution of the particle distribution functions, and (2) direct integration of the N-body equations of motion which follows the orbits of the individual particles. The former method is being used for globular star clusters ($N \sim 10^6$ stars) and the latter is being used for galaxy clusters ($N \sim 10^{15}$ stars).

Pairwise gravitational interactions between stars in dense clusters cause a long-term evolution of clusters towards ever greater central density — a process termed core collapse. The Fokker-Planck code solves a time-dependent, differential-integro equation that describes the diffusion of stars in orbit-space due to these occasional gravitational encounters. This code is usually run on high-performance workstations including a dedicated HP 735. Recent work has focused on the role of compact binary stars in star cluster evolution. These binaries act as "cosmic eggbeaters" to oppose the tendency of clusters to undergo core collapse. This leads to an interesting chaotic oscillation of cluster cores between low and high density states. Planned extensions of this work include increasing the physical realism of the binary treatment using Monte Carlo simulation techniques.

Gravitational interactions among the luminous and dark matter components in galaxy clusters drive an evolution of both the internal structure of the galaxies and the spatial distribution of the galaxies in the cluster. Galaxies tend to settle into the central regions of the clusters where galaxy-galaxy interactions are frequent. These interactions may result in both galaxy growth through mergers and shrinkage through the loss of mass to the cluster background. These complex processes are best followed through a direct solution of the $3N$ coupled, second-order, differential equations of motion. The most computationally intensive aspect of this computation, the determination of the gravitational force, is accomplished by use of a tree algorithm for approximating the spatial distribution of the particles representing the luminous and dark matter components. The simulations have been run on an HP 735 and a Cray Y-MP, using as many as 10^5 particles to simulate a 100-galaxy cluster. A parallelized version of the code is now being run on the IU Intel Paragon. Ultimately, a total of about 10^6 particles will be used to simulate a 1000-galaxy cluster.

5.10 Functional Programming Languages and Algorithms for Multiprocessing

Professor David Wise, Department of Computer Science We are developing multiprocessing matrix algorithms that represent matrices as quaternary trees within heap memory. A recent result is a version of Gaussian elimination for integer matrices that applies uniformly to dense and to sparse matrices (though is particularly well suited to the latter). Pivoting that reduces the magnitudes of intermediate results, that increases parallelism, and that avoids fill-in can all be efficiently done (without additional traversal) as part of each successive elimination step.

5.11 Solution of Dissipative Partial Differential Equations

Professor M. Jolly, Department of Mathematics This project includes the analysis and computational study of nonlinear Galerkin methods for dissipative partial differential equations (PDEs). These methods are based on the theory of inertial manifolds, which provides a rigorous connection between the long time behavior of certain PDEs and that of specially constructed finite-dimensional ordinary differential equations. The emphasis is on capturing the proper dynamics, rather than on transient portions of solutions.

Visualization in phase space plays important role in this effort. Current work involves the construction and rendering of 2-D (un)stable manifolds of stationary and periodic solutions. Using interactive computer graphics to display the intersection of two such manifolds, we can gain a geometric understanding of global bifurcations, which signal drastic changes in the behavior of the physical system.

5.12 Applications in Chemistry

Professor Glenn Martyna, Department of Chemistry, is investigating the structural and dynamical properties of chemical systems at an atomic level of detail. This is accomplished by combining ideas from statistical mechanics and electronic structure to formulate atom based mathematical models. The models, usually coupled sets of differential equations, are then solved on high speed computers. Diverse systems such as liquids metals, impurities in quantum mechanical solids and hydrocarbons in zeolites are currently being examined.

The computational end of Professor Martyna's research involves finding new mathematical frameworks in which to express either the equations of interest or their method of solution. The results are used to produce more efficient algorithms. For example, symplectic multiple time step integration techniques and improved canonical dynamics methods both developed by Professor Martyna have led to large reductions of computer time in a number of applications. However, it is also important and desirable to reduce the computer time used to solve a given problem by employing a parallel machine. Ideally, this reduction is on the order of the number of atoms/molecules in the problem, a significant savings. Fortunately, the structure of the atom based methods used in Professors Martyna's research is such that this can be accomplished. Work in this regard is underway and will continue as new algorithms are developed and implemented.

5.13 Applications in Geophysics

Professor Gary Pavlis, Department of Geological Sciences, is part of the geophysics group. The work is nearly completely computational in nature, and can be characterized as a mix of data processing, scientific visualization, and simulations. At present he has two major ongoing projects that relevant to the computational sciences program.

1. Fundamentals of seismic wave propagation

Our group has been part of a major effort in the past four years to field a new generation of seismic arrays with the objective of extending the frequency bandwidth of these arrays outside the conventional band of 1 to 10 Hz. This work has relevance to monitoring of a future comprehensive test ban treaty on nuclear weapons as all methods for discrimination of nuclear explosions from earthquakes require recording over much wider frequency bands. For this reason our work has focused on arrays designed to record seismic wavefields at higher frequencies (10 to 50 Hz) and at lower frequencies (0.05 to 1 Hz).

These practical goals require improved knowledge of fundamental properties of seismic wavefields. We are approaching this is a new way through two major tools. First, we have developed a new application of scientific visualization to better understand the nature of waves recorded by the state-of-the-art seismic arrays we have operated. Three-dimensional graphics have revolutionized our basic understanding of how complex seismic wave propagation is in the real world. These tools allow us to easily visualize the actual nature of ground motion in an earthquake over a wide frequency range, and to sort out different types of scattering (through a process called beamforming) that drastically complicate seismic observations, especially at higher frequencies. The second tool we use to address these problems are classic to newer state-of-the-art methods of signal processing. We use a wide range of this class of techniques to investigate signal spectra, correlations, and stacking characterists. These provide important quantititive tools to verify hypotheses revealed by the more subjective visualization techniques.

2. Surface wave discriminants

We have a newly funded, three year project, to develop a new technique for the discrimination of nuclear explosions from earthquakes. This project involves developing new signal processing techniques to combine the use of phase matched filters with low frequency seismic array measurements to enhance low amplitude, dispersed, surface wave packets. These signals can commonly have signal-to-noise ratios of much less than one, but they can be extracted because they have a well defined structure. The initial focus of this project will be on algorithm development to try to make this idea work effectively. Ultimately, we expect to apply this technique to enhance and

Department:	CSci	Math	Astro	Physics	Chem	Geo
Number in Minor:	5–10	10–15	0–5	5–10	5–10	5

Table 2: Estimated Number of Students to Minor in Scientific Computing at IU–Bloomington

quantify the energy in surface wave packets produced by small earthquakes and nuclear explosions. Our assertion is that this approach will allow us to extend the most successful existing discrimination method (the so called body wave-surface wave magnitude discriminant) to a much lower limit. This is important as low yield nuclear explosions, at present, are difficult to distinguish from small earthquakes.

6 Course Offerings

A large number of IU courses currently address scientific computation issues. Basic graduate courses can be generally organized into four categories: numerical analysis, behaviour of systems, scientific visualization, and high performance computing. Details of the courses can be found in the Appendix, along with a partial list of undergraduate courses directly relevant to scientific computing.

7 Number of Students Affected

Based on enrollments in currently offered graduate courses and informal polling in the different departments, we expect at least 30 students to minor in the area. Table 2 gives an approximate breakdown of their numbers. A minor program will affect much larger numbers of students, who will take courses without actually minoring in the field. For example, astronomy students routinely minor in physics, because of the close connection between the two fields. Nevertheless, it is likely that astronomy students will be interested in high performance computing courses. Similarly, the scientific computing and visualization courses attract students from areas as diverse as theater design and optics.

8 Minor in Scientific Computing

The number of courses required for a student with a minor in scientific computing will be the same as the number of courses the student's home department requires for an outside minor, preferrably drawn from outside the student's major department. At least two courses must be from outside the student's department, and any courses drawn from within the student's department for credit towards the Scientific Computing minor must be approved in advance. The courses normally are from the list in the Appendix, but need not be restricted to that list. The courses should be chosen in consultation with the student's major advisor, and a member of the Scientific Computation Program Committee. Current members of that Committee and their academic departments are:

- Randall Bramley (Computer Science)
- Richard Durisen (Astronomy)
- Dennis Gannon (Computer Science)
- Steve Gottlieb (Physics)
- Andy Hanson (Computer Science)
- Mike Jolly (Mathematics)
- Glenn Martyna (Chemistry)
- Greg Olyphant (Geological Sciences)
- Peter Ortoleva (Chemistry)

The course Introduction to Scientific Computing I (C671) has been created as a standard introductory course for students in the program. Students entering with a background in computational science and engineering, in consultation with their advisor on the Scientific Computing Committee, may omit that course from their curriculum.

9 Graduate Program

At a recent conference on computational science and engineering education, funded by the Department of Energy, representatives from universities, industries, and national laboratories agreed on the need for students with strong training in computational methods, backed by a solid footing in an engineering or science applications area. Students will seek a degree with one component in a classical area like Physics, Chemistry, Computer Science or

Mathematics and the second component in Scientific Computation. A Ph.D. in Scientific Computation would focus on a problem from a classical scientific discipline, but its primary goal would be the advancement of the algorithmic techniques, tools, and systems that determine the state of the art in this new science. For example, a chemistry student interested in free boundaries arising in reaction-transport systems might have as a primary result a set of tools (software or hardware) to study such problems and visualize the results. This would likely not be considered a classical chemistry Ph.D. and it does not fit in the mold of traditional computer science. But, as a Ph.D. in computational science, it would be both useful to other computational scientists as well as valuable to researchers in traditional chemistry. Because of this, students apply through one of the existing academic departments, and indicate their interest in the Scientific Computation program at that time.

A Relevant Undergraduate Courses

Because scientific computing draws on and is applied to many fields of endeavor, the following list is only for 400–level courses which are of specific use in the area. A truly complete list would include most 400–level courses in mathematics and computer science, as well as a large fraction of those in most of the sciences.

- Mathematics: M471–M472 (numerical analyis), M441–M442 (partial differential equations), M447–M448 (mathematical models and applications) M463–M464 (probability)
- Computer Science: C428 (computer architecture) C481–C482 (computer graphics)
- Chemistry: C361–C362 (physical chemistry)

B Scientific Computing Course Descriptions

Computer Solution of Linear Systems, C690 in CSci

Usual Instructor: R. Bramley

Prereq: Fortran programming knowledge; Ability to create and edit files in Unix; Linear algebra comparable to an undergraduate course: Gaussian elimination, condition numbers, vector norms; Advanced calculus: gradients.

Frequency/Timing: Taught once, Fall 1993. Plan to teach once every year or two.

Topics: Origins, classification, and characteristics of linear systems. Direct methods for dense systems. Direct methods for sparse systems; graph interpretation. Iterative methods: Stationary, Krylov subspace methods. Preconditioning: incomplete factorizations, polynomial, reduced systems. Advanced topics: updating factors, rank and condition estimation, eigenvalue problems.

Target audience: Upper level undergrads and beginning graduate students from mathematics, computer science, other sciences.

Notes: This course emphasizes the practical implementation of the methods on modern high performance computers, not their mathematical analysis.

Scientific Visualization, C690 in CSci

Usual Instructor: A. Hanson, Peter Shirley

Prereq: no strong ones required, but "helpful background" is calculus, analytic geometry, numerical analysis, computer graphics.

Frequency/Timing: Taught once, Spring 93/94. Plan to teach in alternate years, alternating with machine vision.

Topics: 1,2,3,4 dimensional data; histograms, images, scalar fields, volume rendering. Human factors. Imaging via "cartoon" techniques, level sets, slices. Vector fields in 1–3 dimensions. Interactive techniques. Color maps. Database organization and access. Object oriented interfaces. Off-line techniques for large problems. Special project: visualize data set from a scientific discipline.

Target audience: The course is intended to introduce basic techniques of scientific visualization to students with a variety of research backgrounds. Among the areas from which we would like to encourage participation are cognitive science, chemistry, physics, astrophysics, mathematics, biology, and, of course, computer science.

Notes: Some difference in topics between instructors.

Physico-Chemical Modeling and Nonlinear Dynamics, C668 in Chemistry

Usual Instructor: Peter Ortoleva

Frequency/Timing: Alternate years

Prereq: 3 semesters of Calculus

Topics: Describing reaction, transport, mechanical dynamics. Processes and dynamics. Conservation laws and phenomenologies. Discretizations (finite difference and finite elements). Adaptive gridding. Gridding for moving boundaries. Computational homogenization for multiple scale problems. Multi-gridding. Acceleration/extrapolation techniques.

Target audience: Juniors, Seniors and graduate students.

Notes:

Computing Applications in Physics, P410 in Physics

Usual Instructor: Steve Gottlieb, Chuck Horowitz

Prereq: Electricity and magnetism (P332). Introductory computer programming (C201 or C301) ODE's, matrix algebra, complex variables, Fourier analysis. Physics prerequisites can be waived, but students are expected to be able to program in C, Fortran or Pascal.

Frequency/Timing: Has been taught once or twice per year since fall 1988.

Topics: Intro to Unix environment. Classical systems (mechanics, deterministic systems): Coffee Cooling (1st order ODE), falling objects (2nd order ODE), oscillatory motion, chaos, quadrature and monte carlo integration. Symbolic computation with Mathematica, introducing plotting, numerical and symbolic integration, biased estimators and the jackknife technique. Nondeterministic systems: Percolation, curve fitting, fractals, Monte Carlo.

Target audience: Upper level undergraduate or graduate students. Course is designed to introduce students to computational science, beginning with UNIX in a workstation environment and continuing to pursue a variety of topics that allow the students to develop programming and data analysis skills.

Notes: Students learn to write shell scripts, awk commands, 2-dimensional graphics (of publication quality), histogramming and autocorrelations. In dealing with deterministic systems, control of numerical errors is emphasized. Quality of random number generators is discussed under nondeterministic systems. *Introduction to Computer Simulation Methods*, by H. Gould and J. Tobochnik has been used as a text, with *Numerical Recipies* as a recommended reference.

Computational Physics, P609 in Physics

Usual Instructor: Steve Gottlieb, Chuck Horowitz

Frequency/Timing: Has been offered once per year for several years.

Topics: When taught by Gottlieb, similar topics to what has been done in P410, though the pace is somewhat faster. Horowitz has run this as a lab course based on *Computational Physics*, by S. Koonin.

Target audience: Graduate students in computer science, mathematics or a physical science who would like to learn how to use a computer for problem solving.

Notes: This and P410 are the Physics department's bread and butter courses for teaching students to use computers for research. They should be among the core courses of the concentration in Scientific Computing.

Monte Carlo Methods in Physics, P700 in Physics

Usual Instructor: Chuck Horowitz

Prereq: Undergraduate courses in quantum mechanics and/or statistical mechanics.

Frequency/Timing: Approx. every other year, last taught spring 1992-93.

Topics: Introduction to Monte Carlo methods (random variables, sampling methods, Monte Carlo integration). The Metropolis algorithm for sampling a general distribution. Applications to simulations of classical fluids and variational Monte Carlo calculations for Bose and Fermi quantum fluids. Green's Function Monte Carlo and solutions of integral equations. Mechanics.

Target audience: Graduate students in physics, chemistry and computer science.

Notes:

Introduction to Scientific Computation I and II, C671–C672 in CSci

Usual Instructor: Randy Bramley

Prereq: Fortran or C programming knowledge (C202, C301 or C302). Ability to create and edit files in Unix. Linear algebra; vectors, matrices, vector norms (M303 or M301). Advanced calculus; gradients, ordinary differential equations, PDE's (M343).

Frequency/Timing: To be taught Spring 1993. May be annually taught.

Topics: Intro and history. Characteristics of scientific and enginering computations. High performance computer architectures. Useful software tools; awk, sed, grep, perl, graphics tools, netlib. Numerical methods overview. Performance analysis.

Target audience: Seniors or beginning graduate students.

Notes: Planned as the first core course for students in computational sciences.

High Speed Scientific Computation, C690 in CSci

Usual Instructor: Dennis Gannon

Prereq: C or Fortran, linear algebra.

Frequency/Timing: Taught last spring (1993), and may be annual

Topics: Parallel and vector computer architecture. Timing models for regular computations. Programming paradigms:] HPF, message passing, object oriented. Dense computational kernels and matrix computations. Performance models and analysis issues. Compilation Techniques.

Target audience: Graduate students from all fields of science with some computer experience who want to learn about parallel computers, algorithms, and programming.

Notes:

Analysis of Numerical Methods, M571 in Math

Usual Instructor: M. Jolly

Prereq: Undergraduate course work in ordinary and partial differential equations, mathematical analysis, knowledge of a programming language.

Frequency/Timing: Every odd year in the fall semester.

Topics: Topics: Finite difference schemes for PDEs ... von Neumann stability analysis, spurious effects such as dissipation and dispersion, ADI methods for higher space dimension problems, theory of well-posed problems, effect of smoothness of initial condition on rate of convergence. Lax-Richtmyer Theorem (consistent and stable is equivalent to convergent). Simple cases of linear hyperbolic, parabolic, and elliptic PDEs are discussed in detail. A few nonlinear examples are added here along the way.

Target audience: Graduate students seeking a thorough mathematical understanding of finite difference schemes. This is not a cookbook course, nor is it one where we prove the existence and uniqueness of solutions for each PDE considered. The style is however, that of "example-theorem-proof". While there will be one or two computational projects, most homework assignments consist of problems of the type: "show that ...". A good feel for the level of the course is indicated by the main text, J.C. Strikwerda, Finite Difference Schemes and Partial Differential Equations.

Notes: Undergraduate courses in numerical analysis and graduate courses in differential equations are helpful but not required.

Analysis of Numerical Methods, M572 in Math

Usual Instructor: M. Jolly

Prereq: Same as M571, but it is NOT necessary to have taken M571.

Frequency/Timing: Every even year in the spring (following M571).

Topics: Finite element methods for PDEs ... interpolation estimates, convergence results, effect of smoothness, discontinuous Galerkin methods. A brief, yet rigorous treatment of the conjugate-gradient method for solving linear systems will be given. Examples include Poisson equation in 2D, convection diffusion equation (boundary layers, streamline diffusion), exterior Dirichlet and Neumann problems (single and double layer potentials)

Target audience: Similar to that for M571. Some FEM software (PLTMG, and perhaps FIDAP) will be used to illustrate some of the numerical phenomena discussed. The main text is C. Johnson, Numerical Solution of Partial Differential Equations by the Finite Element Method .

Inverse Methods in Geophysics, G612 in Geological Sciences

Usual Instructor: Gary Pavlis:

Prereq: Linear algebra.

Frequency/Timing: Approximately every other year; last taught Fall 1991.

Topics: : Introduction to basic statistics. Theory of generalized inverses with applications to geophysics. Robust estimation techniques for multivariable inversion problems and relation to generalized inverses. Linear inversion of problems with infinitely many unknowns: solution, resolution and propagation of errors tradeoff analysis.

Target audience: Graduate students in geophysics, physics, chemistry, and computer science interested in practical aspects of solving large scale inverse problems. Current focus is in geophysics, but the techniques used have application in all departments involved in this proposed program.

Geophysical Signal Analysis, G514 in Geological Sciences

Usual Instructor: Albert Rudman

Prereq: Physics P222, Mathematics M343 or M313

Frequency/Timing: Once per year

Topics: : Construction, analysis, and interpretation of geophysical signals. Filter theory, spectral analysis, signal to noise enhancement, transform theory, seismic wave propagation, computer applications.

Target audience: Graduate students in geophysics, but should be of interest to students in other fields requiring basic background in signals processing.

Methods in Exploration Seismology, G614 in Geological Sciences

Usual Instructor: Albert Rudman

Prereq: G514

Frequency/Timing: occassionally with demand, last taught 1989.

Topics: : Analytic and digital analysis of wave propagation. Inverse problems. Migration of seismic data in time and f-k space. Generation of seismograms in time and frequency domains. Computer applications.