Los Alamos Computer Science Institute

Statement of Work for Academic Participants

Part I: Overview and Management

Introduction

To foster computer science and computational science research efforts at Los Alamos that are both internationally recognized and relevant to the goals of Los Alamos National Laboratory (LANL), we propose to establish *the Los Alamos Computer Science Institute* (LACSI). The Institute will be a collaboration between LANL and the Rice Center for Research on Parallel Computation (Rice CRPC), along with its partners at University of Houston (UH) and the University of Illinois at Urbana-Champaign (UIUC). The Institute will have major component on site at LANL and at Rice University.

Institute personnel will include member researchers, called *Fellows of the Institute*, both on and off site. The on-site group will be called *LANL Fellows*, while the off-site group will be referred to as *Academic Fellows*. While most of these will be at Rice, Academic Fellows can be at any participating institutions. Initially, there will be such fellows at Rice, University of Houston (Lennart Johnnson, Barbara Chapman, Jaspal Subhlok), and University of Illinois at Urbana-Champaign (Dan Reed, Ruth Aydt).

Goals

The Los Alamos Computer Science Institute will be founded with these goals:

- To build a presence in Computer Science research at LANL that is commensurate with the strength of the Physics community at Los Alamos.
- To achieve a level of prestige in the computer science community that is on a par with top computer science departments in the nation
- To pursue computer science research that is relevant to the goals of High Performance Computing (HPC) programs at Los Alamos. and
- To ensure that there remains a strong focus on high-performance computing in the academic computer science community.

To achieve these goals, the LACSI will establish a collaboration with the Center for Research on Parallel Computation at Rice University and some of its partner institutions. The purpose of this collaboration is to support collaborative research on high-performance scaleable computing that is relevant to the overall LANL goals and to foster a strong relationship between LANL and the CRPC partners, especially Rice University, University of Houston and University of Illinois.

Management

The institute will be managed by an executive committee consisting of senior Fellows from both on and off site. This committee will engage in a continuous review process, described below. It will be assisted in its task by a Computer and Computational Science Advisory Committee, which will include senior researchers in these areas from institutions other than those affiliated with the institute.

Executive Committee

The management of the computer and computational science activities will be the responsibility of an executive committee consisting at senior researchers at LANL and the participating institutions. Initially the executive committee would consist of the following individuals:

- Andy White, LANL, Acting Institute Director, Chair, LANL
- Lennart Johnsson, PI, University of Houston
- Ken Kennedy, PI, Rice
- Dan Reed, PI, UIUC
- John Reynders, LANL

One goal of the effort is to recruit a senior computer or computational scientist to LANL to serve as Director of the Institute. The Institute Director would chair the executive committee and the Rice Project Director (initially Kennedy) would serve as vice-chair.

Reviews

Progress reviews will be an important component of the management strategy. There will be two reviews conducted annually:

1. One review would evaluate the research efforts from the point of view of computer and computational science. The principal question to be answered by this review was whether the research is of high quality and producing results that are innovative and of high relevance to the general area of high-performance computing. In addition to members of the Institute, outside experts on computer science, especially the members of the Advisory Committee would be recruited to participate in this review.

2. The second review would evaluate activities in terms of their relevance to and participation in LANL application themes. The key question would be to what extent the work is making a contribution to the current application activities on LANL computing platforms and what are the prospects for such contributions in the future. LANL application scientists would participate in these reviews along with Fellows of the Institute and Advisory Committee Members.

The reviews would be taken into account in the planning activities of the Executive Committee, establishing new directions along with new goals and modified milestones. The contract will have a one year term with four one year continuance options. The continuance option will be evaluated yearly by the LANL members of the executive committee.

At the end of the first three years of activity, a review committee consisting of external computer and computational scientists along with LANL staff members not affiliated with the Institute, would conduct an evaluation to determine the extent to which the Institute was meeting its goals.

This contract will be for a maximum of five years with a follow-on contract dependent upon the results of the review.

Planning Process

The planning process for the Institute will be driven by the annual review cycle. Based on the outcomes of those reviews, the Executive committee will propose a collection of projects to be undertaken, along with goals for those projects, and identify projects to phase out. Institute Fellows to lead the new efforts would be identified and work would be initiated. The resulting work would be evaluated in subsequent annual reviews. An important criterion used in these evaluations is the extent of relevance to the long and short-term future benefits for LANL applications.

The Academic Planning and Coordination Committee, a committee of senior Academic Fellows drawn from the participating academic institutions would coordinate activities at the Academic Partners.

Interaction/Collaboration

A principal goal of the Institute is to foster collaborative relationships with the participating academic institutions. These collaborations would fall into four broad categories: joint recruiting, on-site visits, technical meetings, and industrial partnerships.

Recruiting

If the Los Alamos Institute is to succeed in its goal of invigorating computer science and computational science research at LANL, it will need to successfully recruit some of the best computer science researchers in the nation. Rice University and the other academic partners can assist in this endeavor in several ways. First academic faculty at Rice can serve on search committees for the LANL Institute, providing an academic perspective and knowledge of the computer science community drawn from years of faculty recruiting.

In addition, positions at the Los Alamos Institute can be made more attractive by including a faculty appointment in either the Rice Computer Science Department or the Computational and Applied Mathematics Department. Such appts are subject to Rice polices and guidelines. Such faculty appointments are typically at one of three levels:

1. *Adjunct Professor*, which is a courtesy appointment typically accorded a distinguished researcher at another institution
2. *Visiting Professor*, which is used for faculty at other institutions who are on a one- to three-year visit on site at Rice; and
3. *Faculty Fellow*, which is used for faculty-level researchers who are supported entirely on grants and contracts. A Faculty Fellow can apply for grants from Rice as a Principal Investigator.

One strategy for recruiting is to have new researchers spend a year or two at Rice as Visiting Faculty or Faculty Fellows before coming to Los Alamos. This would give time for the clearance process to complete while getting them started on relevant research right away. Other staff members might return to Rice for extended visits later in their careers. Of course, we will also make ample provision for long-term visits to Los Alamos by faculty from Rice and other academic institutions.

On-Site Visits

Both Rice and Los Alamos would provide space for long-term visitors from other sites in the Institute. In addition, the Institute would establish programs for visiting faculty (including sabbaticals) graduate students and undergraduate students at LANL.

Joint Meetings

In addition to the annual technical reviews, the Institute would regularly sponsor technical workshops on special topics of interest to the participants. These would typically involve outside researchers and would lead to published proceedings wherever possible.

Industrial Partnerships

The LANL Institute and participating institutions would establish joint relationships with selected industrial partners, particularly those critical to the success of ASCI and follow-on programs (e.g., SGI, IBM, Portland Group).

Computational Resources

The academic partners would be provided access to ASCI computing platforms at LANL on a predetermined basis for development and testing. The process would make it possible to allocate a small cluster of nodes each week and a larger cluster once a month. It is understood that dedicated access may be needed for key tests and performance analyses.

In the following research section, all Rice deliverables for individual projects will include a yearly report describing the project, any affiliated software and its application to LANL problems. Source code affiliated with any project will be delivered to LANL upon request. Such software will be available for use or modification by the United States government, LANL and the University of California without restriction or fee.

Part II: Research

A. Compilation, Systems, and Performance Evaluation of Large Scale Parallel Machines

Investigators: John Mellor-Crummey, Sarita Adve, Vikram Adve, Keith Cooper, Alan Cox, Rob Fowler, Y. Charlie Hu, Ken Kennedy, Dan Reed, Linda Torczon, Willy Zwaenepoel

Compilation strategies

Investigators: John Mellor-Crummey, Sarita Adve, Vikram Adve, Rob Fowler, Ken Kennedy, Jack Dongarra and Antoine Petitet

Project Description: The objective of this project is to develop compiler and run-time technology that will help application developers achieve a high fraction of peak performance on large-scale parallel computing systems. Achieving this objective will require eliminating obstacles to efficiency at multiple levels: within a single processor, within a symmetric shared-memory multiprocessor node, within a cluster of nodes sharing memory with non-uniform memory access (NUMA) latency, and between NUMA shared-memory systems coupled with a message passing interconnection network. A major challenge is to improve utilization of multi-level memory hierarchies within a single thread of control using a combination of data restructuring to improve spatial locality and reduce conflicts, computation restructuring to improve temporal reuse, and software prefetching to improve latency tolerance. Additional challenges for achieving high performance range from analyzing and transforming programs within and across procedures to expose massive parallelism, partitioning data and computation to exploit multi-level parallelism effectively, and scheduling computation and communication to avoid resource contention and load imbalance. All of these issues will need to be addressed in the context of scientific applications that employ sophisticated methods for processing large-scale data sets, including the use of unstructured meshes and dynamic adaptation of data structures. Procedurally, this effort will involve experimentation (including simulation, measurement, and analysis of applications) to identify the most significant performance bottlenecks, developing and prototyping techniques for improving performance in such applications, and developing compiler analysis and transformations to automate application of these techniques to the greatest extent possible.

Milestones: (Year 1) Work with LANL developers to analyze representative applications (including the compact apps) to identify the key classes of bottlenecks; develop and test program restructuring strategies for addressing these bottlenecks; begin assembling compiler analysis and transformation infrastructure to assist in the diagnosis and remediation of performance bottlenecks. A particular focus of this work will be on memory hierarchy optimizations for irregular computations. In Year one, ATLAS technology would be adapted to ASCI needs. **(Years 2-5)** Develop appropriate compiler and run-time technology to implement restructuring transformations that address bottlenecks uncovered in our experimentation during year 1. We expect this to include optimization techniques for irregular and adaptive codes, analysis and transformations to exploit massive parallelism, analysis and optimization of programs with explicit synchronization, compilation strategies improving efficiency of programs for hybrid shared-memory and message-passing systems, techniques for optimization of object-oriented programs**,** dynamic compilation techniques that can use runtime estimates to optimize code on-the-fly; generation of configurable code to support runtime optimization, and integrated tools that support development of portable high-performance codes by providing fine-grain control over compiler optimizations to retarget codes for particular architectures.

IA 64 Compilation Issues

Investigators: Keith D. Cooper, Sarita Adve, Vikram Adve, John Mellor-Crummey, Ken Kennedy, Linda Torczon

Project Description: The objective for this investigation is to develop compiler technology to attain a reasonable fraction of the available performance on IA 64 platforms when running code compiled from classical languages, like dialects of Fortran, C, and C++. Many distinct issues contribute to performance on machines like the IA 64. For high performance, a compiler must:

Keep the functional units busy. This requires the compiler to transform the input program so that it has enough IP to sustain the computation rate. It requires instruction-scheduling techniques that can convert available ILP into dense schedules -- for simple loops, for loops with control flow, and for straight-line code.

Have operands ready for each instruction. This will involve transforming programs to match their locality to the memory hierarchy of the target system, including real applications of blocking, prefetching, and (perhaps) streaming. Once the data is actually on-chip, the compiler may need to manage instruction and data placement with respect to the clustered register file, along with the classic problems of allocation and scheduling.

Handle predication with a holistic approach. If-conversion is not the whole answer. Open issues include understanding the tradeoff between predication, with its sparser execution pattern, and branching to denser code; predicate register management; the interaction between predicate lifetimes and instruction placement in the scheduler; and minimizing the impact of predicate evaluation on overall application performance.

The IA 64 is complex enough that we will need realistic architectural simulations to understand program behavior, the impact of various transformations on performance, and where real potential for improvement exists.

To address these issues, we have brought together a team that includes expertise in analysis and translation for parallel systems (V. Adve, K. Kennedy, J. Mellor-Crummey), in managing memory hierarchies and locality (V. Adve, K. Cooper, J. Mellor-Crummey, K. Kennedy), in low-level code generation issues (K. Cooper, L. Torczon) and in architectural simulation (S. Adve).

Milestones: (Year 1) Investigate a broad array of issues that arise in "wide" machines; work with LANL personnel to characterize application performance issues; work with Intel to get necessary non-disclosure presentations; work internally to connect the various tools together into a testbed for our ideas. **(Year 2)** Begin simulation studies of application performance on the IA 64; tailor investigations begun in year 1 to IA 64-specific issues including memory hierarchy management, managing predicated execution, and handling the clustered register architecture; pursue direct relationships with IA 64 system vendors to transfer results into their commercially available tools. **(Year 3)** Demonstrate prototype implementations of the transformations developed in year 2; report on simulation studies of the effectiveness of these transformations and (Intel willing) hardware validation of performance. Work with LANL application developers to improve, directly, the performance of their codes; launch investigations of new problems revealed by studies in previous years. **(Years 4-5)** Work with LANL applications developers and vendor compiler groups to identify new problems exposed by processor and system architecture that are amenable to compiler-based improvement and develop techniques to address them.

Tools

Investigators: Dan Reed, Vikram Adve, Ken Kennedy, John Mellor-Crummey, Rob Fowler

Project Description: Parallel computing is rapidly evolving to include heterogeneous collections of distributed and parallel systems. Concurrently, applications are becoming increasingly multidisciplinary, with libraries and other application components implemented using diverse programming languages, models and parallelization strategies. In consequence, it is now extraordinarily difficult to achieve high fractions of peak hardware performance on large-scale parallel systems, emerging networks of workstations, or wide area computational grids.

To optimize the behavior of such complex applications, performance analysis software must also evolve, replacing simple measurement tools with deep integration of compile-time transformations with measurement and analysis. Moreover, time varying resource availability and demands will necessitate increasing use of real-time, adaptive performance optimization and just-in-time compilation. This integration, based on user-specified, compiler-synthesized, and measurement-validated performance contracts will enable creation of a new generation of nimble, high-performance applications.

Milestones: (Year 1) Compiler-based performance scalability estimation, measurement, and visualization, targeting shared memory systems (memory hierarchies), NOWs, and wide area networks; performance specification systems based on performance contracts. **(Year 2)** Performance contract validation and constraint-based performance tuning. **(Year 3)** Integrated adaptive performance tuning and just-in-time compilation based on resource availability, together with software distribution and support. **(Year 4)** Validation and extension of performance tuning toolkits based on user experience. **(Year 5)** Multilingual, multi-model performance optimization based on real-time measurement and adaptive control.

B. Component Architectures for Distributed Parallel Computing and Rapid Application Development and Composition

Investigators: Ken Kennedy, Vikram Adve, Alan Carle, Keith Cooper, Lennart Johnsson, Jaspal Subhlok, Joe Warren

The overarching goal of this activity is to develop a component architecture that can be used to support portable parallel and distributed compilation and to apply it to application development support and problem-solving environments for distributed heterogeneous grids. To succeed, this work will need to accomplish three long-term goals.

- It must develop a framework for integrating existing components rapidly and conveniently. The framework must be able to integrate components written in different languages, particularly Fortran and object-oriented languages like C++.

- It must develop a strategy for producing code from collections of components that exhibits reliable efficiency on Grid configurations and high performance on scalable parallel machines. This will involve some sort of whole-program analysis and optimization framework. Furthermore the program preparation process must be reorganized to support this activity. In particular the compilation framework must be divided into smaller components that can be invoked at various times in the program preparation process.

- It must develop a collection of components for use in science and engineering applications. This collection should be ideally suited for use in the rapid prototyping framework described above. The algorithms must be general, portable, and usable in a variety of situations.

Component Architectures for Problem Solving Environments

Investigators: Ken Kennedy, Vikram Adve, Alan Carle, Keith Cooper, Jaspal Subhlok, Jack Dongarra and Antoine Petitet

The goal of problem-solving environment research is produce systems that make it possible for unsophisticated end users to rapidly prototype and experiment with new applications developed from existing libraries of components. The ultimate goal would be to use extensive computation to make the resulting applications as efficient as applications written by professionals. Components would be primarily focused on parallel and distributed computing in science and engineering but many of the techniques would be applicable to all computation domains.

The primary strategy for this activity is to compose applications through the use of (possibly visual) scripting languages. These languages are in wide use today but suffer because the programs that are composed using them are typically not efficient enough for repetitive use in a production situation. Furthermore, there are critical issues of correctness and reliability that may be too technical for the end-user programmer. These issues must be dealt with automatically in the PSE, possibly through the use of extensive computation.

If this effort is to succeed in the grid environment, it must take into account two important realities. First many components will be constructed using object-oriented languages, so techniques for optimizing such languages are critical. Second, the execution environments for the resulting programs are likely to be distributed, so the implementation must take into account the performance implications of distributed systems, even if the applications are compiled together. For these reasons, basing a significant portion of the work on the Java programming language makes sense. Java is portable and has distributed computing interfaces built in. However, we must overcome one major drawback of Java if it is to be used in scientific computation, namely its less-than optimal performance. Although we intend to focus on Java, many of the strategies developed for Java will extend to other object-oriented languages such as C++.

This research will focus on four important directions:

Component Architectures for Integration. This effort will focus on the design and specification of components that can be used in a PSE for high-performance computation. Significant issues will be flexibility and adaptability of the components to both the computations in which they are incorporated and the platforms on which they will be executed. In addition these components must have architectures that permit the effective management of numerical accuracy.

Toolkits for Building Problem-Solving Systems. The effort will also focus on the production of tools for defining and building new domain specific PSEs, including:
- Tools for defining and building scripting languages
- Translation of scripting languages to standard intermediate code
- Frameworks for interprocedural optimization of standard intermediate code supporting multiple languages
- Optimizing translation of intermediate language to distributed and parallel target configurations
- Component libraries for inclusion in scripting systems
- Demonstrations in specific applications domains

Compilation of Object-Oriented Languages. As mentioned above, high-performance compilation strategies must be developed for object-oriented languages such as Java and C++. This should include interprocedural techniques such as inlining driven by global type analysis and analysis of multithreaded applications. This work would also include new programming support tools for high-performance environments.

Integration of Heterogeneous Components and Models. In the future, components and applications will be written in more than one language. To support this kind of application development we propose to work on a number of key supporting technologies including:
- Frameworks for integrating components in diverse languages and models
- Support for data interchange via a uniform interface to a virtual "dataspace"
- Runtime support to optimize data transfer and minimize buffering
- Support for language heterogeneity including C, C++, Fortran and Java.
- Aggregate import and export, e.g. parallel arrays and other compound objects
- Customization for specific coordination and data exchange patterns (e.g. streaming data, broadcasts, concurrent shared objects), that lead to context specific communication and buffering optimizations.
- Support for large and persistent data objects.

Milestones: (Year 1) Definition of component architecture to foster smooth integration, even for numerical components with variable accuracy. Demonstration of a preliminary global optimizing compiler for Java. Demonstrate a preliminary framework for integrating components to produce efficient parallel programs. Adapt the NetSolve environment to ASCI needs and work with LANL investigators to demonstrate ASCI-like application domains. **(Years 2-5)** Produce a high performance compilation system for Java and extend the techniques to C++. Produce a complete PSE and toolkit for integrating component into efficient parallel and distributed systems. Produce an interprocedural framework capable of integrating PSE components with high performance.

Distributed Computing

Investigators: Ken Kennedy, Vikram Adve, Keith Cooper, Lennart Johnsson, Jaspal Subhlok

The key challenge in building grid applications is to construct applications that are adaptive to changes in the execution environment and which can detect and correct performance problems automatically.

In this activity, we will explore the meaning of network-aware adaptive applications and what the implementation and optimization challenges are for such applications.

The work will proceed in two major subactivities:

Frameworks for Building Adaptive Network Applications. This research will develop a framework for building "network-aware" applications that can adapt their execution dynamically to meet performance and Quality-of-service goals. The main components of this research are as follows:

- A uniform interface to the network state, including the workloads on the computation nodes and the capacities and traffic on the network links.
- Runtime support for efficient dynamic migration of applications on a network.
- Algorithms to drive dynamic selection of nodes and migration decisions based on application structure and network state.

Compilation of Reconfigurable Object Programs. This research will explore the challenges in compiling programs to be dynamically reconfigurable in the sense described above. It will explore programming models and their translation to efficient collections of tasks that can be targeted to a variety of Grid computing platforms. The long-term goal is to support component-based implementation of applications for grids using PSEs of the sort described in the previous section. A critical technology will be adaptivity to changing performance characteristics of the components of a distributed computing platform. This adaptivity will require reoptimization and migration of components and data.

Milestones: Work with compilation group and ASCI representatives to define model applications for evaluation of technology, then develop sample codes for test and evaluation of compiler directives and compilation based support for dynamic load-balancing. **(Year 1**) define five representative application code samples and implement those for one language and execution environment. (**Year 2)** Implement and evaluate for all ASCI execution environments. (**Year 3**). Optimization and additional sample codes.

Numerical Component Libraries

Investigator: Lennart Johnsson, Jack Dongarra and Antoine Petitet

- Initiate and lead community effort towards an interface standard that allow for the creation of efficient implementations of common library routines (e.g., FFT, eigenanalysis, equation solvers, BLAS, etc) on a broad spectrum of architectures and heterogeneous distributed environments.

- Creation of polyalgorithmic, parameterized library functions that allows for the run-time selection/optimization of algorithm selection and scheduling based on the problem at hand and the execution environment.

- Fast Algorithms. A number of key ASCI applications are based on mathematical models requiring evaluation of interparticle interactions. We propose to develop library support for use of multipole and multipole like O(N) methods in ASCI applications.

- Algorithms for Graphics and Visualization. Included are algorithms for the manipulation and visualization of extremely large data sets, generation and refinement of unstructured simplicial grids, along with multiresolution analysis over those grids, data management via wavelets, and feature definition and extraction.

Milestones: (Year 1) Development of a draft community standard for common scientific software functions in large scale simulations through a series of workshops. Demonstration of efficient FFT routines, including "out-of-core" routines on ASCI platforms. Demonstration of linear time, adaptive, N-body solver. (**Year 2)** Real-to-complex and complex-to-real FFT for ASCI platforms. Prime factor transforms. Fast Legendre and spherical transforms. (**Year 3)** Polyalgorithmic solvers.

C. Computational Mathematics

Large-Scale Nonlinear Optimization Algorithms and Software

Investigators: Richard Tapia and Yin Zhang

Project Description: The objective of this project is the development of reliable algorithms and parallel software for large-scale nonlinear constrained and unconstrained optimization problems, including systems of nonlinear equations. The emphasis will be on constrained least-squares problems for large-scale data-fitting and parameter identification. The algorithms will be based on interior-point methodology. In some of the algorithms and codes, scalability will be the primary focus; in others, reliability will be the emphasis.

Relevance to LANL: This investigation will support computational infrastructure building for emerging applications. It will provide reliable and enabling tools for important computational tasks, such as large-scale constrained data-fitting and solution of nonlinear-systems, that need to be solved in LANL's applications and predictability studies.

Milestones: **(Year 1)** Perform algorithmic research, algorithm testing, and initial software design of reliable algorithms and parallel software for large-scale nonlinear constrained and unconstrained optimization problems, including systems of nonlinear equations. **(Years 2--5)** Rice will develop prototype software tools using existing linear algebra libraries, refine the code and test it on DOE/LANL-related applications. Develop a preliminary parallel version of the code. Document research results in technical reports.

Parallel Tools for the Analysis of Linked Subsystems

Investigators: John Dennis, Matthias Heinkenschloss, Joe Warren

Project Description: The objective of this project is the development of parallel tools for the coupling of simulations of aspects of a problem to full system simulations on which decisions can be based. This requires the development of a mathematical and algorithmic framework that supports the coupling of expensive subsystem simulations with varying accuracy, the identification of parameters in these linked systems, and the error prediction in linked simulations through nonlinear backward and forward analyses (if the outputs are allowed to vary in a certain range, how accurate must the parameters be; if the parameters vary in a certain range, what is the variation in the outputs). The research issues that will be investigated are: (1) Parallel algorithms for the solution of blocked nonlinear systems with very expensive and inexact function evaluations; (2) Parallel algorithms for parameter identification governed by expensive and inaccurate simulations; (3) Sensitivity and adjoint equation approaches for nonlinear error analyses; and (4) Implementation of these algorithms in an object-oriented way in parallel computing environments.

Relevance to LANL: This investigation will support the LANL effort in emerging applications, which require the coupling of subsystem simulations. It will provide mathematical and algorithmic frameworks for error propagation in coupling of simulation codes.

Milestones: **(Year 1)** Design an object-oriented framework for linked subsystem simulations and provide an initial implementation of basic algorithms in this framework. Apply this framework to problems relevant to LANL. **(Years 2--5)** Expand and refine the subsystem analysis tools and their application to emerging applications. Document the results in technical reports and implement prototypes.

Code-Based Sensitivity Analysis Project

Investigator: Alan Carle

Project Description: The goal of this project is to develop source-to-source code transformation tools to convert massively-parallel, high-fidelity, physics-based computer simulation codes into codes that are suitable for use in the context of accurate and efficient sensitivity analysis. The research issues that will be investigated are: (1) Automatic differentiation of Fortran 90/95, including techniques for computing first- and higher-order derivatives of parallel codes using the forward and adjoint modes of automatic differentiation; (2) Automatic extension of simulations "at a point" to simulations "in a neighborhood of a point" for use in validation and verification by replacing computations on floating point values with computations on intervals, wavelets, probability distributions, or other representations of ranges of values (generalized ranges); and (3) Novel ways in which information about code structure can be used to improve/accelerate sensitivity analysis.

Relevance to LANL: Sensitivity analysis plays an important role in developing a deep understanding of the results of computational simulations and the underlying mathematical models that drive the simulation. Sensitivity analysis is a key component in techniques for parameter identification and quantification of uncertainty.

Milestones: (Year 1) Investigate availability, acquire, and evaluate Fortran 90/95 compiler infrastructure. Investigate computational techniques for sensitivity analysis. **(Years 2--5)** Develop prototype code augmentation infrastructure for Fortran 90/95 and explore representations, and local rules, for generalized ranges. Investigate techniques for global propagation. Complete automatic differentiation tool for Fortran 90/95 using code augmentation infrastructure and test. Develop prototype of generalized range propagation tool. Complete generalized range propagation tool and test. Improve automatic differentiation tool and generalized range propagation tool based on user feedback. Document research results in technical reports.

Eigenvalue Methods and Software for ASCI-MPP systems

Investigator: Danny Sorensen

Project Description: This investigation will develop methods and parallel software for large eigenvalue problems and related applications. The project will extend the capabilities of the highly successful P_ARPACK eigenvalue software. This software provides enabling technology in numerous application areas. It also can serve as an excellent test bed and point of interaction for new compiler technology. Two additional related areas which impact other projects in the computational mathematics effort are reduced basis methods for dynamical systems and control of processes governed by PDE and regularization of large discrete ill-posed problems such as those arising from inverse problems. Research goals include: (1) The ability to handle increases of two or more orders of magnitude in problem size (current technology is 10^6 to 10^7 variables); (2) The ability to scalably determine a significant percentage of the eigenpairs (increases from 10% to 50% of the total are needed for selected applications in electronic structure applications); (3) Demonstration of terascale performance for a set of simulations that are central to understanding material properties; (4) Development of new reduced basis methods for dynamical systems and control of processes governed by PDE and provision of new partial balanced realizations with optimal H-infinity error estimates (there is potential for several orders of magnitude increase in time scale for molecular dynamics); and (5) Application of recently developed methods for regularizing discrete ill posed problems to parameter estimation and other inverse problems.

Relevance to LANL: This research will support the LANL effort in emerging applications, which require eigen-analysis, SVD, or principle component analysis. It will support DOE and LANL efforts such as electronic structure calculations on materials and chemical reaction dynamics. The project will provide enabling technology for simulation of composite materials and their decay modes, control systems arising in linked-subsystems investigation, and parameter estimation and inverse problems.

Milestones: (Year 1) Complete initial design of scalable parallel eigenvalue software for terascale architectures. Design interface for relevant applications. Incorporate available technology from compiler and parallel tools projects. Initiate study of reduced basis methods and applications of regularization. **(Years 2--5)** Expand and refine the eigenvalue software. Continue investigation and software design for reduced basis methods, regularization schemes and their application to emerging applications. Document results in technical reports and implement prototypes.

Integer/Mixed Integer Programming and Large Scale Problems

Investigators: <u>Bill Cook</u>, Bob Bixby, David Applegate, Nate Dean

Project Description: Discrete optimization is a critical component of computational engineering. It is used to solve practical problems that involve choosing the best alternative from a field of possibilities. Perhaps the most important class of discrete optimization problems are the mixed integer programming models, or MIPs for short. These problems differ from linear programming problems in that some of the variables are required to take on only integer values. These integer variables allow modeling of decisions which are discrete in nature, but their inclusion makes MIP problems much more difficult to solve than the corresponding linear programming problems.

The inherent difficulty of MIP, and of discrete optimization in general, tests the limits of available computing platforms. Moving discrete optimization onto distributed, parallel systems will be a focus of Rice's research in this area, together with fundamental work on solution techniques for very large systems.

Relevance to LANL: Large scale simulations in the area of predictability will certainly need to include discrete components. The projects in discrete optimization will make tools available to modelers in this area. Discrete optimization, and MIP in particular, will also be important in the development of high performance compiler tools.

Milestones: (Year 1) Study computational aspects of discrete optimization and MIP. **(Years 2--5)** Develop prototype software for the solution of large-scale discrete models. Document the results in technical reports.

Methods and Tools for the Solution of Non-smooth, Multi-scale, Coupled Models

Investigator: <u>Petr Kloucek</u>

Project Description: The focus of this research project is on the development of computational methods and tools for a prediction of meso-scale phenomena. The computational techniques and massively parallel algorithms developed for this project will include domain decomposition coupled to operator splitting, nonlinear Galerkin and functional multigrid methods, adaptive mesh refinement of unstructured grids, and a technique for connection of non-matching grids.

Relevance to LANL: The proposed mathematical and computational framework is directly related to the predictability of low-dimensional physical events within systems with continuum multifractal structure. Such phenomena include dynamic fracture, large instabilities in random systems, pattern formation, grain growth, and nucleation.

Milestones (Year 1) Design a massively parallel algorithm for the computation of multi-fractal structures in crystalline materials. **(Years 2--5)** Apply this method to model high-frequency loading of shape-memory alloys. Concentrate research on a development of novel computational techniques for non-smooth non-equilibrium systems. Document the results in technical reports and implement prototypes.

Methods and Software for Inverse and Control Problems

Investigators: Bill Symes, Mark Gockenbach

Project Description: Characterization and control of complex dynamical systems, such as wave propagation in heterogeneous materials, requires coupling of very expensive and inexact simulations, often involving subsimulations interacting at vastly different scales, with appropriate optimization and linear algebra algorithms. These algorithms must incorporate features not found in contemporary library code, such as management of the interaction of simulation and optimization errors, scalability with maximal architecture neutrality, and transparent management of data storage amongst devices of varying latency. It is virtually inconceivable that such tools could be built, maintained, and successfully transferred to end users without the employment of object-oriented programming methodology.

Progress towards these objectives will be achieved through extension and enhancement of an existing object-oriented numerical software library which already incorporates many of the desired features. Applications to inverse problems and geophysical signal processing and to control of structures subject to dynamic loads will serve as testbeds. Prior experience in coupling to automatic differentiation, automatic class generation from numeric kernels, and adjoint state approach will be leveraged to provide powerful rapid prototyping tools.

Relevance to LANL: This investigation will support the LANL effort in simulation, identification, and control of complex multiscale phenomena through elaboration of an established object-oriented approach to this class of problems.

Milestones: (Year 1) Demonstrate object-based coupling of automatic differentiation, automated class construction, adjoint state sensitivity, and control of large complex systems. Accomplish initial design of distributed vector base class. Implement Arnoldi-based trust-region regularization of ill-posed inverse problems, subspace methods for multiple scales, and interior-point codes for inequality constraints. Perform a preliminary exploration of Java's suitability as a possible replacement for C++. **(Years 2--5)** Perform further investigations of algorithm structure, transparent data and operator distribution, and integration of graphical interfaces. Document the results in technical reports.