Los Alamos Computer Science Institute

Statement of Work for Academic Participants

Part I: Overview and Management

Introduction

The *Los Alamos Computer Science Institute (LACSI)* was created to foster computer science and computational science research efforts at the Los Alamos National Laboratory (LANL) that are both internationally recognized and relevant to the goals of LANL. LACSI is a collaboration between LANL and the Rice University Center for Research on High Performance Software, along with partners at the University of Houston (UH), the University of Illinois at Urbana-Champaign (UIUC), the University of Tennessee at Knoxville, and the University of New Mexico in Albuquerque. LACSI has major components on site at LANL and at Rice University.

LACSI personnel include member researchers, called *Fellows of the Institute*, both on and off site. The on-site researchers are called *LANL Fellows*, while the off-site researchers are referred to as *Academic Fellows*. While most of these will be at Rice, Academic Fellows can be at any participating institutions. The initial list of fellows is included below by institution:

Rice University: Ken Kennedy, Bob Bixby, Liliana Borcea, Alan Cox, Keith Cooper, John Dennis, Nate Dean, Mark Embree, Mike Fagan, Rob Fowler, Richard Hanson, Matthias Heinkenschloss, Y. Charlie Hu, Guohua Jin, Petr Kloucek, John Mellor-Crummey, Dan Sorensen, Bill Symes, Frank Toffoletto, Richard Tapia, Linda Torczon, Yin Zhang, Willy Zwaenepoel.

University of Houston: Lennart Johnsson, Barbara Chapman, Roland Glowinski, Yuri Kuznetsov, Jaspal Subhlok.

University of Illinois at Urbana-Champaign: Dan Reed, Ruth Aydt, Celso Mendes.

University of Tennessee at Knoxville: Jack Dongarra.

University of New Mexico: Deepak Kapur, Ed Angel, Tom Caudell, Arthur Maccabe, Pat McCormick.

Goals:

The Los Alamos Computer Science Institute was 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 LANL.
- To achieve a level of prestige in the computer science community that is on a par with the best computer science departments in the nation.
- To pursue computer science research that is relevant to the goals of High Performance Computing (HPC) programs at LANL.

- To ensure that there remains a strong focus on high-performance computing in the academic computer science community.

To achieve these goals, the LACSI was established as a collaborative effort with the Center for Research on High Performance Software (HiPerSoft) at Rice University and some of its partner institutions. The purpose of this collaboration is to support joint research on high-performance scalable computing that is relevant to the overall LANL goals and to foster a strong relationship between LANL and the academic partners, especially Rice University, the University of Houston, the University of Illinois, the University of Tennessee, and the University of New Mexico.

Management:

LACSI is managed by an executive committee consisting of senior Fellows from both on and off site. This committee conducts a regular review process, described below.

Executive Committee: The management of the computer and computational science activities will be the responsibility of the Executive Committee, which consists of senior researchers at LANL and the participating institutions. Current membership of the Executive Committee is as follows:

- Andy White, Institute Director, Chair, LANL
- Ken Kennedy, Vice-Chair, Rice
- Jack Dongarra, Univ. of Tennessee, Knoxville
- Lennart Johnsson, UH
- Deepak Kapur, UNM
- John Mellor-Crummey, Rice
- Rod Oldehoeft, LANL
- Dan Reed, UIUC
- Dan Sorensen, Rice
- Rick Stevens, ANL
- John Thorp, LANL
- Linda Torczon, Rice

This group will be expanded (or reduced) in the future as appropriate to meet the management needs of the Institute.

The Institute Director will chair the executive committee and the Rice Project Director (initially Kennedy) will serve as vice-chair.

LACSI Symposium:

The *LACSI Symposium* is both a meeting to showcase results of LACSI research efforts and a forum for presenting outstanding research results from the national community in areas overlapping the LACSI technical vision. This will be a traditional conference-style meeting with participation by LACSI Fellows, outside scientists from academic partner institutions, and other scientists from the community at large.

Executive Committee:

The Executive Committee is responsible for planning and reviewing LACSI activities on a regular basis, establishing new directions along with new goals and modified milestones. The Executive Committee will evaluate progress based on the quality and relevance of the research being done, and how the goals of the institute are being met.

In performing this evaluation and planning process, the Executive Committee will be assisted by the LACSI Planning Committee, which consists of all members of the Executive Committee plus additional project leaders from LANL and the Academic Partners.

Based on the outcomes of its reviews of the research and the other institute activities, the Executive Committee, assisted by the LACSI Planning Committee, may 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 will be identified and work will be initiated. The resulting work will be evaluated in subsequent reviews.

Collaboration Management

Rice University will be the lead (prime) on the contract for all academic partners. The Academic Partners Management Committee, a committee of senior Academic Fellows drawn from the participating academic institutions, will coordinate activities of the academic partners. Current membership of this committee includes Ken Kennedy (chair), Keith Cooper, Alan Cox, Jack Dongarra, Matthias Heinkenschloss, Deepak Kapur, Lennart Johnsson, John Mellor-Crummey, Dan Reed, Dan Sorensen, and Linda Torczon. Membership in this committee may change, as determined by the committee itself.

Contract Term:

With this contract, the LACSI project will be entering its fourth year of operation. This contract will be for a five year period, beginning October 1, 2001 and ending September 30, 2006.

During the third year of activity in this contract period (year 6 overall for LACSI), a review committee, consisting of external computer and computational scientists along with LANL staff members not affiliated with LACSI, will conduct an evaluation to determine the extent to which LACSI is meeting its research goals. Based on the results of this review, the LACSI contract will be either:

- extended for an additional five year term (beginning October 1, 2004 - the fourth year of this contract period, and the seventh year overall for LACSI), or

- phased out over the final two years of the current contract period, with a budget phase down as explained in the contractual terms.

If the contract is renewed for an additional five year term, a similar review will be held during the new third year of the extended contract with similar options for further extensions or phase down.

The review committee will evaluate LACSI research progress according to two criteria::

1. The quality of the research from the perspective of computer and computational science. The principal question to be answered is whether the research is of high quality and is producing results that are innovative, long-term in focus, and of high relevance to the general area of high-performance computing.
2. The relevance of the research to LANL application themes. The key question would be to what extent the work is making a contribution to the application activities on LANL computing platforms and what the prospects are for such contributions in the near or distant future.

Modes of Interaction and Collaboration

A principal goal of LACSI is to foster collaborative relationships between LANL participants and the participating academic institutions. These collaborations fall into four broad categories: joint recruiting, joint research activities (including on-site visits), technical meetings, and industrial partnerships.

Recruiting

If LACSI 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. Academic faculty at Rice can serve on search committees for LACSI, providing an academic perspective and knowledge of the computer science community drawn from years of faculty recruiting.

In addition, LACSI positions might be made more attractive by including an appointment in either the Rice Computer Science Department or the Computational and Applied Mathematics Department. Such appointments are subject to Rice University policy and guidelines.

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 LANL. This would allow time for the clearance process to complete while the researcher begins work on a LACSI effort. 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 LANL by faculty from Rice and other academic institutions.

Collaborative/Joint Research Activities

Collaborative research activities between the academic partners and LANL scientists are strongly encouraged. Both Rice and LANL will provide space for long-term visitors from other LACSI sites. In addition, LACSI will establish programs for visiting faculty (including sabbaticals), graduate students, and undergraduate students at LANL.

Joint Meetings

In addition to the annual symposium, LACSI will sponsor technical workshops on special topics of interest to the participants. These will typically involve outside researchers and will lead to published proceedings where applicable.

Industrial Partnerships

LACSI partners will establish joint relationships with selected industrial partners, particularly those critical to the success of ASCI and follow-on programs (e.g., IBM, The Portland Group, Compaq, etc.).

Computational Resources

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

In the following research section, all academic deliverables for individual projects will include an annual 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, Barbara Chapman, Keith Cooper, Alan Cox, Rob Fowler, Richard Hanson, Y. Charlie Hu, Guohua Jin, Ken Kennedy, Dan Reed, Linda Torczon, Willy Zwaenepoel

High-level Compilation and Compiler-based Tools

Investigators: John Mellor-Crummey, Barbara Chapman, Rob Fowler, Guohua Jin, Ken Kennedy

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 on conventional large-scale systems requires 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. Latency tolerance will be critical for high performance.

To exploit whatever configuration is available at run time, applications must be able to adapt accordingly. Flexible parallelization strategies must be developed that can effectively exploit a range of execution environments.

Principal areas of focus within this project include:

Compiler-assisted development tools. Achieving the highest possible performance with programs may require complex restructuring that cannot be proven safe without exploiting user knowledge. Our goal is to capitalize on user knowledge that is not, or not easily, derivable by a compiler. In this effort, we will develop tools to assist users in creating code that can readily adapt to a wide variety of runtime environments. We will investigate using "self tuning" (an approach that has successfully been employed by libraries such as ATLAS) for improving node performance. We aim to develop semi-automatic support to assist in extensive rewriting of programs. For example, semi-automatic support is needed for transforming key fragments of arbitrary codes into a format that is amenable to an ATLAS-style self tuning strategy. Furthermore, although exposing parallelism inherent in program regions may require modifications that cannot be automated, a tool may be able to interact with the user to determine the need for change and help automate parts of the software adaptation task.

Node Performance: A principal problem facing scientific codes on systems composed of commodity microprocessors is ineffective use of multi-level memory hierarchies. We will investigate strategies for improving utilization of multi-level memory hierarchies within a single thread of control using a combination of computation restructuring to improve temporal reuse along with data restructuring to improve spatial locality and reduce conflicts. This effort will include compile-time approaches such as loop fusion and program transformations to improve register allocation of array values, along with run-time approaches such as sparse matrix dissection to improve temporal locality in irregular computations.

Multiprocessor Performance: Challenges for achieving high performance include analyzing and transforming programs within and across procedures to improve parallelism, partitioning data and computation to exploit parallelism effectively, and optimizing communication to minimize volume, frequency, and exposed latency. Topics we will investigate in this area will include data-parallel compiler technology, multiprocessor time skewing - a partitioning strategy that improves node locality and reduces communication, and compiler technology for explicitly parallel languages such as Co-array Fortran and UPC. In addition, we will investigate compiler assistance for employing fault-tolerant strategies to cope with inevitable node failures that arise frequently on very large-scale systems

This research will consider 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 project 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, developing compiler analyses and transformations to automate application of these techniques to the extent possible, and incorporating these techniques into compilers and tools that can be used by application scientists.

In addition to technology for conventional large-scale parallel systems, we will begin to explore compilation technology appropriate for promising petaflops architecture designs, including processor-in-memory systems.

Milestones:

Year 4: Analyze applications in order to understand current approaches to parallelization, the extent of potential parallelism, and the degree to which it is exploited. Explore how to improve latency tolerance in codes. Adapt SGI open-source Pro64 compilers for C and Fortran 90 into a toolkit for source-to-source program transformation tools for improving memory hierarchy of scientific programs. Produce a prototype tool for improving memory hierarchy utilization.

Year 5: Develop tool support for studying applications and the implications of existing parallelization strategies. Continue to study applications with LANL scientists and develop compiler and run-time technology to implement restructuring transformations that address bottlenecks uncovered in ongoing experimentation. Deploy prototype compiler for an explicitly parallel language. Lay foundation for compiler-assisted tools to support massively parallel programming.

Year 6: Continue work on compiler and runtime techniques for improving performance on large-scale machines, including research on compiler and tools for exploiting massive parallelism more effectively, analysis and optimization of programs with explicit synchronization, and begin investigation of compiler technology for petaflop computing. Develop approaches for input of user information to tool and its exploitation. Consider interaction with compiler technology that may directly exploit user knowledge.

Years 7-8: Investigate techniques for optimization of object-oriented programs**,** dynamic compilation techniques that can use runtime estimates to optimize code on-the-fly; and generation of configurable code to support runtime adaptation. Investigate use of pattern matching and other technologies to support code modification.

Compilation Issues for High-Performance Microprocessors

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

Project Description: This investigation (formerly called IA 64 Compilation Issues) focuses on developing the compiler technology required to obtain a reasonable fraction of the performance available on high-end microprocessors, such as the IA 64. The premise of the research is that code written in high-level languages, such as Fortran, C, C++, or Java, should execute well on these machines. This requires that the compiler manage many different aspects of program execution. Among the areas that this project will explore are:

> *Instruction-level parallelism (ILP):* The compiler must transform the input program so that it contains enough ILP to keep the functional units busy. It must schedule the code so that the processor finds those opportunities. It may need to manage placement of data and operations on the chip (i.e., in distinct clusters or even distinct processors) as well.

Memory hierarchy management: The compiler must ensure that operands are available, in registers or cache, when an operation needs them. To do this, the compiler must transform the program to ensure that its locality matches the memory hierarchy of the target system. Example transformations include blocking, prefetching, and streaming.

Handling new architectural features: As microprocessors evolve, new features appear that are intended to improve performance. Recent examples include predication, speculation, and clustered register sets. Before applications see the benefits of these features, compilers must have effective mechanisms for using them.

The primary goals of this project are to develop practical compilation techniques to address the problems that keep current compilers from generating efficient code for high-performance microprocessors, and to work at transferring these techniques, and other best-practice techniques, into commercial and open-source compilers.

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

Milestones:

Year 4: Investigate scheduling, operand placement, and predication issues on the IA-64. Evaluate performance of open source compilers for IA-64 using LANL applications or abstracts of them. Use LANL applications (or abstracts of them) to study optimization sequences for adaptive compilation.

Year 5: Work with developer communities to improve open-source IA-64 compilers. Demonstrate prototype transformations and report on their effectiveness. Investigate strategies for improved convergence in adaptive compilation.

Year 6: Continue developing, demonstrating, and publishing transformations that improve performance in the open-source IA-64 compilers. Work with vendors to move adaptive compilation techniques into commercial systems.

Years 7-8: 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.

Performance Tools

Investigators: Dan Reed, Ruth Aydt, Barbara Chapman, 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. Principal areas of focus within this project include:

Compiler-assisted Performance Tools (Mellor-Crummey, Fowler): Understanding how to achieve top performance with an application on an architecture requires understanding where and how an application's demands don't match an architecture's capabilities. We will design and build language-independent tools that explain program performance by collecting, correlating, and displaying information about program structure, transformations by optimizing compilers, and dynamic performance measurements.

Intelligent Performance Toolkit (Reed): To optimize the behavior of such complex applications, performance analysis software must also evolve, replacing simple measurement tools with tools that provide deep integration of compile-time transformations, measurement and analysis. Moreover, time varying resource availability and demands will require 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.

We will develop a new generation of software tools that integrate compilation research described previously with real-time performance measurement, adaptive control systems, and intelligent data visualization and control. The goal of this work is to create an intelligent performance toolkit that allows software developers to create and runtime systems to negotiate "performance contracts" among software components and hardware/software systems.

Milestones:

Year 4: Deploy HPCView tools for use by LANL application scientists, work with them to identify application performance bottlenecks, and identify ways compiler and run-time technology could ameliorate them. Expand dynamic instrumentation capability and continue compilation integration efforts begun earlier. Validate and extend performance tuning toolkits and visualization systems based on user experience.

Year 5: Refine application signature performance model to automatically detect application phase changes and trigger loading of phase-specific performance models.

Year 6: Develop support for multilingual, multi-model performance optimization based on real-time measurement, visualization and adaptive control.

Year 7: Extend performance contract specification and monitoring infrastructure to support hierarchical contract verification.

Year 8: Develop "control panel" for the application and contract monitoring system. This panel would allow the user to observe the resource selection, application launching, and contract monitoring process, allowing the user to adjust tolerance levels, substitute models, select new resources interactively, and view historical data about previous predictions and measurements.

OpenMP

Investigators: Alan Cox, Y. Charlie Hu, Willy Zwaenepoel,

Project Description: The objective of this project is to develop portable shared memory programming support that scales from small clusters of workstations and SMPs to the ASCI supercomputers. Our approach is to support OpenMP — an emerging shared memory programming API standard — directly on top of the TreadMarks software distributed shared memory (DSM) system, which runs on networks of workstations and PCs and SMPs. In the first three years of this project, we have developed the basic system and studied its performance using various benchmark applications. We focused in particular on understanding the limitations of the system in terms of scalability, and develop solutions to the primary scalability problem, the fan-out of data after sequential sections of code. We have also worked with Kuck and Associates, Inc. (KAI), now part of Intel to transition this technology into a commercial product.

We will continue to work with KAI to complete this technology transfer, and to incorporate some of the scalability approaches we have developed into the commercial version. The resulting tool should have the stability and the support necessary to port application codes to it and evaluate its performance with these codes. Using that experience we will refine the tool. In addition, we will investigate the impact of data distribution directives on OpenMP and the use of new primitives to exploit multi-level parallel machines such as networks of SMPs.

Milestones:

Year 4: Refine and complete work on scalability issues.

Year 5: Demonstrate the fully integrated compiler/runtime system that supports efficient and portable OpenMP programming.

Year 6: Investigate the impact of data distributions on compilation and performance of OpenMP.

Year 7: Investigate specific ways of taking advantage of multi-level parallel machines in OpenMP.

Year 8: Develop and aid in the development of large-scale parallel application in OpenMP on networks.

B. Component Architectures for Rapid Application Development and Composition in a Networked Environment

Investigators: Ken Kennedy, Barbara Chapman, Keith Cooper, Jack Dongarra, Richard Hanson, Lennart Johnsson, Dan Reed, Jaspal Subhlok

The overarching goal of this activity is to develop component architectures that can be used to support rapid prototyping of portable parallel and distributed applications. These architectures would be the basis for frameworks for applying advanced compilation techniques, run-time system elements, and programming tools to prepare applications for execution on scalable parallel computer systems and distributed heterogeneous grids. To succeed, this effort will need to accomplish two long-term goals.

1. It must develop *frameworks for integrating existing components* rapidly and conveniently into complete applications. These frameworks must be able to produce efficient applications from scripts within reasonable compile times. In addition, they must be able to integrate components written in different languages, particularly Fortran and object-oriented languages like C++ and Java. Finally, the frameworks must support the generation of applications that execute with reasonable and reliable efficiency in a distributed computing environment.

2. 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 frameworks described above. The algorithms must be general, portable, and usable in a variety of situations.

Component Architectures for Problem Solving Environments in a Networked Environment

Investigators: Ken Kennedy, Barbara Chapman, Keith Cooper, Jack Dongarra, Richard Hanson, Lennart Johnsson, Dan Reed, Jaspal Subhlok

Generation of Problem-Solving Systems Through Component Integration

The goal of this project is to develop compiler technologies and library designs that will make it possible to automatically construct domain-specific development environments for high-performance applications. This project will develop advanced compiler technology to construct high-level programming systems from domain-specific libraries. Programs would use a high-level scripting language such as Matlab to coordinate invocation of library operations. Scripting languages typically treat library operations as black boxes and thus fail to achieve acceptable performance levels for compute-intensive applications. Previously, researchers have improved performance by translating scripts to a conventional programming language and using whole-program analysis and optimization. Unfortunately, this approach leads to long script compilation times and has no provision to exploit the domain knowledge of library developers.

To address these issues we are pursuing a new approach called "telescoping languages", in which libraries that provide component operations accessible from scripts are extensively analyzed and optimized in advance. In this scheme, language implementation would consist of two phases. The offline translator generation phase would digest annotations describing the semantics of library routines and combine them with its own analysis to generate an optimized version of the library, and produce a language translator that understands library entry points as language primitives. The script compilation phase would invoke the generated compiler to produce an optimized base language program. The generated compiler would (1) propagate variable property information throughout the script, (2) use a high-level "peephole" optimizer based on library annotations to replace sequences of calls with faster sequences, and (3) select specialized implementations for each library call based on parameter properties at the point of call.

We will use this strategy to construct a high-level application development environment for an application of interest to LANL based on Matlab. This system could be seen as a flexible replacement for POOMA.

We also plan to extend these programming systems to prepare applications for execution on computational grids. If this effort is to succeed, 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 includes distributed computing interfaces. 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++.

With these considerations in mind, we plan to pursue research in three important directions:

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:

1. Tools for defining and building scripting languages
2. Translation of scripting languages to standard intermediate code
3. Frameworks for generating optimizers for scripting languages that treat invocations of components from known libraries as primitives in the base language.
4. Optimizing translation of intermediate language to distributed and parallel target configurations
5. Tools for integrating existing code
6. Demonstration of these techniques in specific applications of interest to ASCI and LANL.

An important goal of this effort is to make it possible to build highly efficient applications from script based integration of pre-defined components. Building on the component architecture efforts described above, we will pursue the novel strategy of "telescoping languages" to make it possible to extend existing languages through the use of software components.

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.

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. Initially, this work will focus on Java, through the use of the JaMake high-level Java transformation system developed at Rice. Later we will consider extensions to other object-oriented languages. This effort will be undertaken as a collaboration with the CartaBlanca group at LANL.

Milestones:

Year 4: Preliminary prototype Matlab framework for supporting PSE development delivered to LANL, including Matlab front end integrated with Rice optimization framework. Specification of interface for incorporation of libraries into the telescoping languages framework. Demonstration of JaMake transformation framework applied to LANL Java programs.

Year 5: Prototype telescoping languages framework based on Matlab delivered to LANL, demonstrated on a PSE derived from an application domain to be selected in collaboration with LANL. Refine the JaMake high performance compilation system for Java and apply to components of the LANL CartaBlanca project on rapid prototyping of algorithms.

Application Development Support for Distributed Computing

The key challenge in building grid applications is to construct applications that are adaptive to changes in the execution environment and that 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. In addition we will pursue research on middleware to support optimal resource selection in grid environments. The work will proceed in two major subactivities:

Compilation of Configurable Object Programs. This research will explore the challenges in compiling programs to be dynamically configurable 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. This work will amplify the NSF-funded GrADS effort and apply it to applications of interest to LANL. A significant component of this work will be the development of performance models and mapping strategies that can be used to automatically retarget applications to different parallel and distributed computing platforms.

*Frameworks for Building Adaptive Network Applications***.** This project will perform fundamental research and develop middleware for "optimal" resource selection in shared distributed environments. The key challenge is predicting the performance of an application on a set of nodes on a shared network with dynamically changing resource availability. The planned framework for resource selection will have the following main components:

- Development of the notion of a *performance signature*, which represents application resource requirements and predicts application performance under different network condition such as different speed links and processors and different competing loads and traffic
- Development of a *resource graph* representation for the topology and status of currently available resources such as CPU and memory at nodes and latency and bandwidth on network links.
- Development of algorithms for *graph based node selection*, which map an application represented by its performance signature to the network represented by a resource graph.

Milestones:

Year 4: Demonstration of the GrADS execution system. Preliminary tools for constructing performance models and mapping strategies for configurable object programs. Code and methodologies to discover the application level structure and behavior from network level measurements, with application to NAS benchmarks.

Year 5: Demonstration of an application preparation and execution using the GrADS framework. Demonstration of a tool to generate the topology of simple networked clusters, including link capacities, traffic and sharing.

Numerical Component Libraries

Investigator: Lennart Johnsson, Jack Dongarra, Roland Glowinski, Richard Hanson, Yuri Kuznetsov

The principal goal of this effort is to develop libraries of components that can be effectively used in rapid prototyping systems of the sort this project proposes to produce. Key considerations will be production of scientific computing components that can be used reliably on a variety of computation platforms, especially parallel and distributed systems. This will entail establishing an architecture that provides a high degree of flexibility while maintaining efficiency and robustness across a broad spectrum of computational configurations. Important subactivities include:

1. *Standards*. Initiate and lead community effort towards an interface standard that allow for the creation of efficient implementations of common library routines for equation solvers, BLAS, etc. on a broad spectrum of architectures and heterogeneous distributed environments.
2. *Adaptable Libraries*. 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.
3. *Fast Algorithms*. Several 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.
4. *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.
5. *Extending the ATLAS* technology so it can gather key parameters of target hardware platforms and optimize selected kernel or software components across a range of platforms.

Milestones:

The five year outline of the project consists of four mutually dependent tracks:

- Development of new algorithms.
- Development of code generation and optimization infrastructure.
- Generation and implementation of adaptive software libraries on different parallel architectures.
- Demonstration of use of software libraries in important applications.

Year 4-5:

- Initiate an attempt to develop a community consensus for APIs for FFTs.
- Release of a FFT library for MPI applications supporting complex-to-complex, real-to-complex and complex-to-real transforms.
- Release of a FFT library for threaded applications supporting complex-to-complex, real-to-complex and complex-to-real transforms.
- Development of a single processor adaptive library for Sine and Cosine transforms.
- Performance modeling for FFTs.
- Release of a code generation and optimization toolbox for FFTs.
- Demonstration of the parallel complex FFT library in at least one application.
- Release of a Sine and Cosine library for MPI applications.
- Release of a complex FFT library for applications using MPI+threads.
- Release of an OpenMP interface for the FFT library.
- Release of out-of-core FFT transforms.
- Performance modeling for complex and real transforms.
- Development of a code generation and optimization toolbox for finite element problems.
- At least preliminary investigations into most, if not all of: (a) generalization of ATLAS's configure routines, (b) code generation for GEMV and GER, (c) code generation for some Level 1 routines, (d) SMP support via pthreads, (e) packed storage optimizations, (f) wide and narrow band optimizations, (g) sparse optimizations, (h) algorithmic tweaks and higher level routines, (i) distribute the recursive LU, Cholesky and QR software.

Year 6-8:

- Demonstration of the Sine and Cosine library in at least one application.
- Release of a Sine and Cosine library for applications using MPI+threads.
- Release of an OpenMP interface for the Sine and Cosine library.
- Release of a Lagrangian finite element library of arbitrary dimension and order for simplexes and hypercubes for MPI applications.
- Development of a code generation and optimization toolbox for Legendre and Spherical transforms.
- Development of a code generation and optimization toolbox for finite difference (convolution) problems.
- Release of a library for Legendre and Spherical transforms for MPI applications.

- Release of a library for Legendre and Spherical transforms for threaded applications.
- Demonstration of the Lagrangian finite element library in at least on application.
- Release of a Lagrangian finite element library of arbitrary dimension and order for simplexes and hypercubes for threaded and MPI+threads applications.
- Release of an OpenMP interface for the Lagrangian finite element library.
- Performance modeling for the finite element library functions.
- Demonstration of the Legendre and spherical transform library in at least one application.
- Release of a library for Legendre and Spherical transforms for applications using MPI+threads.
- Release of an OpenMP interface for the Legendre and Spherical transform library.
- Performance modeling for parallel spherical transforms.
- Release of a library for finite difference (convolution) problems.
- Demonstration of the finite difference (convolution) library in at least one application.
- Performance modeling for the finite difference library.

C. Computational Mathematics

Large-Scale Nonlinear Optimization Algorithms and Software

Investigators: Yin Zhang, Richard Tapia

Project Description: The objective of this project is the development of algorithms and software for a number of large-scale nonlinear optimization problems. One primary research area will be in linear and nonlinear semi-definite programming, i.e., optimization problems with matrix-valued variables and/or constraints. Another emphasis will be on the development of novel algorithms for large-scale graph-partitioning problems based on nonlinear optimization.

Relevance to LANL: This investigation will support computational infrastructure building for emerging applications. It will provide reliable and enabling tools for important computational tasks that need to be solved in LANL's applications and predictability studies. In particular, effective graph-partitioning algorithms will have important applications and immediate impact in parallel computations.

Milestones:

Year 4: Perform algorithmic research and design, conduct testing and develop prototype software. Two prototype software packages are currently under development: one for large-scale linear semi-definite programming, another for a computational geometry problem called Maximum Volume Ellipsoid problem. Work with LANL scientists to identify common research interests and to establish joint projects.

Year 5: Continue to perform algorithmic research and other tasks. In particular, we will study new algorithms that extend our graph-partitioning techniques from Max-Bisection, where they have been proven effective, to Min-Bisection problems. Finish the two prototype software packages started in Year 1. Work with LANL scientists to help disseminate the applicability and capacity of optimization techniques and to learn necessary details of their applications.

Years 6-8: Study approaches to solving large-scale nonlinear semi-definite programming problems and develop experimental codes. Develop software for the Min-Bisection problems. Refine the previously developed software and test on DOE/LANL-related applications whenever appropriate. Develop parallel version for those codes with proven values for large-scale applications. Document research results in technical reports.

Parallel Tools for the Analysis of Linked Subsystems

Investigators: Matthias Heinkenschloss, John Dennis

Project Description: The objectives of this project are the development of parallel tools for the coupling of simulations of aspects of a problem to full system simulations on which optimization-enabled 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 development of optimization algorithms suitable to use with these simulations, 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) Optimization algorithms suitable for use with such simulations; (4) Sensitivity and adjoint equation approaches for nonlinear error analyses; and (5) 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 support Newton and non-Newton optimization. It will aide in the analysis of error propagation in coupling of simulation codes.

Milestones:

Year 4: Explore domain decomposition (DD) formulations for full system analysis and optimization. Study solution of mixed variable problems via response modeling. Implement prototypes.

Year 5: Expand and refine DD framework and solution of mixed variable problems. Apply to problems relevant to LANL.

Years 6-8: Expand linked subsystem approaches to multidisciplinary problems and problems with non-deterministic data. Interface with emerging applications. Document the results in technical reports and implement prototypes.

Code-Based Sensitivity Analysis Project

Investigator: Mike Fagan

Project Description: The goal of this project is to develop source-to-source code augmentation 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 simulations written in Fortran 90/95, C, C++, Java, and combinations of these languages, (2) Techniques for computing first- and higher-order derivatives of parallel codes using the forward and adjoint modes of automatic differentiation, (3) Transformation techniques that extend simulations "at a point'" to simulations "in a neighborhood of a point'" for use in validation and verification. Interesting "neighborhood'" extensions we are currently considering include intervals, wavelets, probability distributions and generalized ranges, (4) 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 4: Investigate availability, acquire, and evaluate various language scanners and parsers. Year 1 focus is primarily Fortran 90/95 compiler infrastructure. Of particular interest is the recently released open source HP Fortran 90 offering.\\

Years 5-8: Investigate computational techniques for sensitivity analysis. Construct general purpose augmentation framework that can use different language front-ends. 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, Mark Embry

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.

We intend to develop a new method for simulating a shift-invert spectral transformation without direct matrix factorization. This transformation is required to accelerate convergence in difficult cases such as computing clustered or interior eigenvalues.

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 compute clustered or interior eigenvalues without direct matrix factorization, (3) Demonstration of terascale performance for stability analysis calculations associated with ocean models, (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.

A specific LANL application will be to use of eigenvalue methods from P_ARPACK for linear stability and bifurcation analysis of ocean models. We plan to apply P_ARPACK to find the most unstable modes of an ocean model. Preliminary work completed includes updating ARPACK and P_ARPACK codes to be compatible with the latest version of LAPACK. We have also completed preliminary work on the development an approximate shift-invert spectral transformation that will accelerate convergence of eigenvalue calculations without requiring a sparse-direct factorization and without requiring very accurate iterative solves. This will enable the solution of much larger problems in a matrix-free manner. We intend to work with LANL scientist Balu Nadiga to apply this to stability and bifurcation analysis of both idealized ocean models and full ocean models.

The ultimate goal currently lies with the full ocean model. We hope to find a parameter regime where the model goes to a steady state, and then look into the stability of that state and get the most unstable eigenmodes.

Milestones:

Year 4-5: Continue design and implementation of scalable parallel eigenvalue methods for terascale architectures. Develop simulation of shift-invert spectral transformation. Continue study of reduced basis methods and applications of regularization.

Years 6-8: Complete implementation of scalable parallel eigenvalue software for terascale architectures. Design interface for relevant applications. Apply the methods to stability analysis of ocean models. Incorporate available technology from compiler and parallel tools projects. 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: Nate Dean, Bob Bixby

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 two most important classes of such problems are mixed integer programming models (MIPs for short) and network optimization models. MIPs 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 that are discrete, but the MIP becomes more difficult to solve than the corresponding linear programming problem. Networks are commonly used to model complex systems in many areas including transportation, communication, scheduling, chemistry, computing, and biology.

The inherent difficulty of discrete optimization in general is that it tests the limits of computing. Applying discrete optimization to areas such as compiler design, data mining, and parallel systems will be the 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 is important in the development of high performance compiler tools.

Milestones: In each case the results will be documented in technical reports.

Year 4-5: Study applications and computational aspects of large scale problems involving discrete optimization and networks. Develop algorithms for their solution.

Years 5-7: Develop prototype software for selected DOE/LANL-related applications in collaboration with researchers at LANL.

Years 7-8: Optimize and test software on DOE/LANL-related applications.

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

Investigator: Petr Kloucek, Frank Toffoletto

Project Description: This project is to pursue applied research in the area of meso-scale numerics, which is relevant to a number of applications currently being investigated at LANL. These applications include modeling of elastic properties of foams, thermodynamics of twinned shape memory alloys, and the study of the origin of instabilities in the Earth's Magnetosphere.

We will continue our collaboration with the LANL T3-groups of Dr. Jeremiah Brackbill and of Dr. Brian VanderHeyden. We collaborate with the Brackbillπs group on development of coupling techniques for connection of stochastic and deterministic computational models. We plan to incorporate special finite element techniques such as nonconforming rough FEs and nonconforming space-time FEs into the Java Rapid Prototyping Environment developed by VanderHeyden's group within the LANL CartaBlanca project.

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 4: Generalization of the Subgrid Projection Method to two and possibly three dimensions. Translation of the FEAT-based operator calculus into Java High Performance Environment

Year 5-6: Incorporation of mesoscale numerical module based on the nonconforming rough FEs and non-conforming space-time FEs into CartaBlanca; modeling and control of the thermodynamics of martensitic transformation to provide vibration control. Completion of the development of the coupling of stochastic and deterministic models including the Fokker-Planck or Landau-Vlasov-Maxwell equations with Magneto-hydrodynamics

Year 7-8: Computational modeling of instabilities in Aurora and Magnetosphere; computational modeling of the reconnection. Incorporation of the particle-continumm coupling module based on the Landau/Vlasov-MHD interface into the CartaBlanca RPE

Methods and Software for Inverse and Control Problems

Investigators: Bill Symes, Liliana Borcea

Project Description: Develop high performance, reusable, extensible software for inverse and control problems using modern programming methods. Approach: propose standard interface software for definition of problems in this class, implementation of algorithms. Prototype realization: the Hilbert Class Library (HCL), a collection of C++ classes supporting and implementing the most effective algorithms for control and parameter estimation in large, complex systems.

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 4-8:

- Continued development of HCL and applications. Version 2.0 in alpha state, result of loose collaboration with DoE researchers and others. Chief focus: specification of vector interface which accommodates arbitrary user-defined operations without requiring users to alter vector code, related operator and function interfaces.
- Shannon Scott (MA, 2001): implemented component architecture using CORBA and MPI to realize parallel vector and operator classes without invading core HCL code. Solved acoustic control model problem with no loss of efficiency compared to native Fortran/MPI implementation.

- Eric Dussaud (current PhD student): proposes to use HCL, component framework, and abstract time stepping classes in satellite data assimilation application for magnetosphere structure studies; collaboration with Rice space physics group.
- Mark Gockenbach, Fei Wang (MTU): implications of adaptive gridding for automatic differentiation in control/inverse context; further development of component framework and remote/parallel classes.

Efficient Parallel Solvers for Multiphase Flow in Strongly Heterogeneous Porous Media

Investigators: Roland Glowinski, M.Buksas (LANL), M.Hyman (LANL), Yuri Kuznetsov, Lennart Johnsson L.Winter (LANL)

The simulation of multiphase flow in porous media involves outstanding issues in Modeling and Scientific Computing. Among the computational issues there is the necessity of solving time-dependent nonlinear diffusion equations with highly discontinuous coefficients and anisotropic behavior. Very fast solvers are required for the following reasons:

- Due to the random nature of some of the data, Monte-Carlo techniques may have to be used, each sampling requiring the solution of a time-dependent problem involving a diffusion operator among others.
- The iterative solution of inverse problems (via adjoint equation techniques, for example) may require a large number of simulations and also the backward solution of adjoint equations.The adjoint equation technique for inverse problems will be effective only if there exist fast solvers for the direct time dependent equations modeling the flow, and for their adjoints.

An unfortunate feature of the above problems is the fact that the heterogeneity and anisotropy of the diffusion coefficients make unfeasible the use of uniform (or quasi-uniform) finite element or finite volume meshes. On the other hand a well-known advantage of unifrom meshes is that they allow the use of fast solvers (such as multigrid or cyclic reduction based algorithms, for example) of optimal computational complexity. In order to overcome the above difficulties, and obtain efficient solution methods we intend to investigate a computational methodology based on the following components:

- Time discretization by operator-splitting in order to treat optimally the various operators associated to the physics and numerics of the problem under consideration.
- Space discretization by mixed/hybrid finite element or by finite volume methods in order to better represent the geometry complexities.
- Use of logically rectangular, locally refined meshes fitting the various lines or surfaces of discontinuity encountered in the problems.
- Parallelizable domain decomposition methods with the ability (via Lagrange multiplier based mortar techniques) to take into account possibly non-matching meshes at internal interfaces.
- Efficient preconditioners for the parallel solution of the saddle-point problems associated to mixed/hybrid finite element approximations and mortar techniques.

- Binary tree techniques to minimize the memory storage requirements associated to the implementation of adjoint techniques when solving inverse problems.

The investigators intend to further investigate the simulation of wave propogation in heterogeneous media. Besides the importance of these problems to LANL scientists we want to take advantage of the many methodological commonalities between these problems and those discussed in paragraph 3.Indeed the domain embedding methods we employ to treat wave propagation around obstacles, apply to the solution of diffusion problems as well.

We intend to collaborate with the related LANL/UH LASCI project "Parallel Numerical Methods for the Diffusion and Maxwell Equations in Heterogeneous Media on Strongly Distorded Meshes". The parallelization of the mesh generators and solvers will be done in cooperation with Johnsson, Chapman and Subhlok and collaborators in the Department of Computer Sciences at UH.

Milestones:

Year 4-5: Parallel algorithms and numerical results for 2D/3D diffusion type equations by mixed finite-element and volume methods on rectangular grids in strongly heterogeneous media.

Year 6: Parallel codes for the preconditioned iterative solution of the 2D/3D diffusion problems discretized by mixed finite element or finite volume on locally refined, logically rectangular meshes.

Year 7-8:. Integration of the diffusion solvers into the flow in porous media simulators. Investigation of the fast solution of inverse problems.

PARALLEL NUMERICAL METHODS FOR THE DIFFUSION AND MAXWELL EQUATIONS IN HETEROGENEOUS MEDIA ON STRONGLY DIST0RTED MESHES

Investigators: Yuri Kuznetsov, M.Berndt (LANL), Roland Glowinski, Lennart Johnsson, D.Moulton (LANL), M.Shashkov (LANL)

Project Description: Efficient parallel numerical methods for the diffusion and Maxwell equations in highly heterogeneous anisotropic media is a hot topic for scientists and engineers working in computer simulation of complex physical phenomena. This statement is very relevant to several research groups at LANL and UH.

The researchers from the LANL part of the project are very experienced in accurate and physically consistent approximations to the diffusion and Maxwell equations on strongly distorted meshes as well as in applications of advanced numerical methods to real life problems.

The researchers from UH have long term experience in discretization of partial differential equations by mixed and hybrid finite element methods. They also hold the world-leading positions in designing of efficient parallel iterative solvers based on a combination of domain decomposition, fictitious domain and multilevel techniques.

The first main objective of the project is to develop and investigate new accurate, physically consistent and convenient for applications approximations to the 3D diffusion and Maxwell equations with highly heterogeneous anisotropic coefficients on strongly distorted logically rectangular meshes. The second main objective of the project is to design, investigate, and implement on parallel computers new fast iterative solvers for large-scale algebraic systems resulting from the above discretizations.

New algorithms and experimental software will be delivered to LANL for testing, verification and evaluation as well as for applications to the problems of LANL interests

Milestones:

Year 4-5:

Algorithms and experimental parallel code for 3D diffusion equations on logically rectangular strongly distorted meshes.

Algorithms and experimental code for the numerical solution of 3D Maxwell equations with piece-wise constant coefficients on rectangular meshes. This solver will be used as a preconditioner for the approximations on distorted meshes.

Years 6-8:

Algorithms and numerical experiments for 3D convection-diffusion equations on logically rectangular distorted meshes.

Algorithms and parallel software for 3D Maxwell equations by mimetic/mixed finite elements with applications.

21-AUG-2001