RESEARCH IN MATHEMATICS AND COMPUTER SCIENCE AT ARGONNE:

September 1989 through February 1991

Gail W. Pieper

Mathematics and Computer Science Division

March 1991

This work was supported by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.
# Contents

1 MCS Division Highlights 

2 MCS Scientific Staff
   2.1 Permanent Staff
   2.2 Wilkinson Fellow
   2.3 Postdoctoral Researchers

3 User Facilities
   3.1 Advanced Computing Research Facility
   3.2 The Intel Touchstone DELTA System
   3.3 ACRF Users

4 Research Supported by DOE – Office of Energy Research
   4.1 Algorithms and Software
      4.1.1 Numerical Linear Algebra (C. Bischof, M. Jones, P. Plassmann, P. Tang)
      4.1.2 Optimization (J. Moré, P. Plassmann, S. Wright)
      4.1.3 Automatic Differentiation and the Numerical Analysis of Nonlinear Systems (A. Griewank)
      4.1.4 Domain Decomposition Algorithms (W. Gropp, B. Smith)
      4.1.5 Optimal Control and Boundary Value Problems (S. Wright)
      4.1.6 Function Approximation Techniques (W. J. Cody, P. Tang)
      4.1.7 Quadrature (J. Lyness)
   4.2 Applied Mathematics
      4.2.1 Fluid Dynamics (I-L. Chern, W. Gropp, G. Leaf)
      4.2.2 Conservation Laws (G.-Q. Chen, I-L. Chern)
      4.2.3 Combustion (H. Kaper, G. Leaf)
      4.2.4 Reaction-Diffusion Systems (H. Kaper, M. Kwong)
      4.2.5 Superconductivity (H. Kaper, M. Kwong)
   4.3 Computer Science
      4.3.1 Logic-based High-Performance Computing (I. Foster, E. Lusk, W. McCune, S. Winker)
      4.3.2 Software Tools for Parallel Computing (W. Cowell, I. Foster, E. Lusk, R. Stevens)
      4.3.3 Automated Reasoning (E. Lusk, W. McCune, R. Stevens, S. Winker, L. Wos)
4.3.4 Abstract Programming (J. Boyle) ........................................... 24
4.3.5 Automated Program Verification (K. Dritz) ................................ 25

5 Research Supported by DOE – Office of Health and Environmental Research 27
5.1 Molecular Biology (I. Foster, R. Overbeek, S. Winker) ...................... 27
5.2 Parallel Climate Modeling (I-L. Chern, I. Foster, W. Gropp, R. Stevens) ...... 27

6 Special Activities Supported by Non-DOE Funding 29
6.1 NSF Funding ........................................................................... 29
6.1.1 CRPC ............................................................................... 29
6.1.2 Summer Institutes in Parallel Computing ................................. 30
6.1.3 Minority Internship in Parallel Computing ............................... 30
6.1.4 Workshop on Asymptotic Analysis and Numerical Solution of PDEs ...... 30
6.2 SDI Funding ........................................................................... 30
6.3 U.S. Army and Air Force .......................................................... 32
6.4 U.S. Army .............................................................................. 32

7 Technology Transfer 33
7.1 Motorola ............................................................................... 33
7.2 U.S. Air Force .......................................................................... 33

8 Educational Activities 34
8.1 A-PRIME – Argonne Program in Mathematics Education ............... 34
8.2 Special Course in Supercomputing Offered to Minority Students ...... 34
8.3 Visitors Program ....................................................................... 35

A Scientific Staff 36

B Administrative and Technical Support Staff 37

C Professional Activities 38

D Scientific Impact 40

E Meetings, Workshops, and Classes 57

F Special Term Appointments and Other Visitors 59

G Seminars 62
MATHMATICS AND COMPUTER SCIENCE RESEARCH AT ARGONNE

September 1989 through February 1991

by

Gail W. Pieper

Abstract

This report reviews the research activities in the Mathematics and Computer Science Division at Argonne National Laboratory for the period September 1989 through February 1991. The body of the report gives a brief look at the MCS staff and the research facilities and then discusses the diverse research projects carried out in the division. Projects funded by non-DOE sources are also discussed, and new technology transfer activities are described. Further information on staff, visitors, workshops, and seminars is found in the appendixes.

1 MCS Division Highlights

The past eighteen months have been both an exciting and rewarding period for the Mathematics and Computer Science (MCS) Division. With continued funding from the Applied Mathematical Sciences subprogram of the U.S. Department of Energy (DOE), we explored diverse topics in algorithm and software design, applied mathematics, and computer science, developing new techniques and tools to advance the state of the art in scientific computing. In 1990, we became actively involved in DOE's CHAMMP Climate Modeling Program and received funding for one of only five pilot projects set up to investigate mathematics and computer science issues related to global climate modeling. We have also taken a lead role in the PCN project—an effort supported by the NSF Science and Technology Center for Research in Parallel Computation to develop and implement a new parallel programming language. And, in December we began coordinating Argonne's participation in the Concurrent Supercomputing Consortium, providing scientists with access to the Intel Touchstone DELTA System—the world's most powerful computer.

The eighteen-month period has also been highlighted by the formulation of new strategies, the development of new software, and the discovery of new science:

- design of the LANZ software package for solving the large, sparse, symmetric generalized eigenproblem—a problem of significant importance in structural engineering (Section 4.1.1);

- development of partitioning techniques that lead to almost linear speedups on function-gradient evaluations and Hessian-vector products for partially separable functions, which arise in many large-scale optimization problems (Section 4.1.2);

- design of the automatic differentiation package ADOL-C, incorporating a new technique that enables researchers to calculate second and higher partial derivatives automatically and accurately, without the high storage cost of traditional methods (Section 4.1.3);
• completion of SPECFUN, a collection of programs for the computation of the most important special functions of mathematical physics (Section 4.1.5);

• establishment of a rigorous framework for 2 by 2 nonlinear hyperbolic systems of conservation laws, proving a long-standing conjecture about the propagation of oscillations (Section 4.2.2);

• use of our logic programming toolkit to identify tetra-loop composition in ribosomal RNA, recognized as making a solid contribution to biology (Section 4.3.1);

• formulation of new strategies applied by an automated reasoning program to discover more elegant axiom systems, long sought by the world’s leading mathematicians (Section 4.3.3);

• demonstration of the scalability of the spectral transform method, vital for parallel climate modeling on future teraop computers (Section 5.2.1).

In early 1991, DOE (together with other federal agencies including the Department of Defense, NASA, and the National Science Foundation) was poised to implement its program for High Performance Computing and Communications (HPCC). The goal of this new federal initiative is to enhance U.S. competitiveness by supporting enabling technologies for high-performance computing. The program consists of four parts, emphasizing high-performance computing systems, algorithms and software development, networking technology, and basic research and human resources. Given our current program of research activities, we view the HPCC Program as an exciting opportunity to apply our expertise in parallel computing to some of the Grand Challenges of computational science. Towards this end, we have initiated several collaborative projects with scientists in other divisions at Argonne (Materials Science, Chemistry, Biology, and others), as well as outside Argonne, formulated a major new project in software development for linear algebra (Parzival), and started an educational project to develop visualization techniques that will help current and future teachers of mathematics better understand the fundamental concepts of mathematics (A-PRIME, in collaboration with Argonne's Division of Educational Programs and Chicago State University). These activities complement our ongoing core research programs in mathematics and computer science. They will become more visible as the HPCC Program is implemented in the next few years.

A vital part of our activities also involves encouraging the transfer of our research accomplishments to commercial applications. For example, we enhanced our software analysis tools and incorporated them into a commercial software product offered by the Numerical Algorithms Group, Inc., through a technology transfer project sponsored by NAG and the State of Illinois. We also collaborated with Motorola on the use of abstract programming and program transformation techniques for police radio communications, developed a new computational strategy (in collaboration with scientists at Mobil) that improves the reliability and speed of critical point calculations important for simulating enhanced oil recovery processes, and designed a new algorithm for the power function which has been adopted by Sun Microsystems and MIPS Computer Inc. More recently, we have begun exploring automated reasoning techniques in the design of logic circuits based on arithmetic logic units provided by Computational Logic, Inc.

This has also been a banner period for awards. Three of our staff members were honored during this period:
• G.-Q. Chen was named an Alfred P. Sloan Research Fellow. Chen holds an Argonne–University of Chicago Fellowship in Mathematics and has been active in the study of differential equations important in fluid dynamics.

• W. J. Cody received an Outstanding Contribution Award from the IEEE Computer Society for helping to draft ANSI/IEEE Std. 854-1987, the IEEE Standard for Radix-Independent Floating-Point Arithmetic.

• C. Bischof received the "Best Paper" award at the Conpar90/VappIV Joint Conference on Vector and Parallel Processing held in Zurich, Switzerland, on September 10–13, 1990, for his work on "An Adaptive Blocking Strategy for Matrix Factorizations."

In the remaining sections of this report, we highlight these and other activities in the Mathematics and Computer Science Division from September 1989 through February 1991. First we give a brief look at our scientific staff and at our research facilities. Then we review specific research activities conducted by the division members. Included also are a discussion of projects funded by non-DOE sources and a description of several new ventures undertaken to promote the transfer of technology within the Laboratory and to industry. A complete list of division members, their publications, and their professional activities is provided in the appendices.

We conclude with a list of people who visited the MCS Division to conduct seminars, participate in workshops, or collaborate on special projects as part of our extensive visitors program. We encourage university faculty, postdocs, and graduate and undergraduate students, as well as staff members from other laboratories and industry, to collaborate with our regular staff, communicating new ideas and contributing new challenges to the division's research programs.
The success of our research is directly attributable to the energy and expertise of our scientific staff. This year our permanent staff was joined by several postdoctoral fellows and by researchers with joint appointments at other institutions.

2.1 Permanent Staff

Here we give a brief look at our permanent staff members—their past achievements and their current research. In particular, we welcome four new members: I-Liang Chern, Ian Foster, William Gropp, and Stephen Wright.

Christian H. Bischof initially joined our division as the first Wilkinson Fellow in 1988 and became a regular staff member in 1989. He has played an active role in the development of portable algorithms for the LAPACK project. He is currently working on numerical linear algebra software for parallel machines and parallel extensions of the ADOL-C automatic differentiation package.

James M. Boyle has been active for several years in the area of program transformation. His efforts led to the development of the TAMPR program transformation system, which was used in the development of LINPACK. His current interest is in abstract programming and its use, along with program transformation, to develop highly efficient parallel programs for high-performance computers.

I-Liang Chern has been working on hyperbolic conservation laws and computational fluid dynamics. Currently, he is involved in Argonne's global climate modeling studies, designing new algorithms for the shallow-water equations on the sphere.

William J. Cody, Jr., was instrumental in establishing the IEEE standard for floating-point arithmetic, now implemented in numerous chips. He also developed much of FUNPACK, a collection of special function subroutines, and wrote a book entitled *Software Manual for the Elementary Functions*. He has recently completed another collection of portable special function routines and test programs, called SPECFUN.

Wayne R. Cowell played a major role in the development of the early PACKs—EISPACK and LINPACK—at Argonne. More recently, he coordinated the Toolpack project to develop portable software tools, and currently is extending this work to the study of a Fortran-oriented programming environment for advanced-architecture computers.

Kenneth W. Dritz is participating in the preparation of ISO standards for Ada and takes an active role in studying Ada as a language for parallel programming. He has contributed chapters to several books on Ada and has developed and taught tutorials and workshops on the language.

Ian T. Foster has been actively involved in the study of concurrent logic programming and its applicability in large-scale parallel computing. His particular interests are in the development of high-level programming languages (e.g., Strand and PCN) and their application in global climate modeling, molecular biology, and automated theorem proving.

Burton S. Garbow is best known for his pioneering work on EISPACK. He also collaborated with his colleagues on the development of MINPACK and the Toeplitz package. His recent work has included implementation of Toolpack on several of the advanced computers in our ACRF.
Andreas Griewank brings a strong background in computational mathematics and applied analysis. His current research focuses on automatic differentiation and the numerical analysis of nonlinear systems.

William D. Gropp was recently appointed Deputy Scientific Director of the Advanced Computing Research Facility. His research interests include domain-decomposition and preconditioning methods for partial differential equations, with particular attention to their performance on parallel computers. He is also studying the scalability of numerical methods for global climate models.

Hans G. Kaper serves as the division director. He also continues to carry out research in nonlinear differential equations.

Man Kam Kwong carries out research in applied analysis. His interests are in boundary value problems, nonlinear differential equations, and symbolic programming languages.

Gary K. Leaf was instrumental in the development of DISPL, a package widely used by scientists and engineers throughout the United States. He is currently analyzing problems in fluid dynamics and combustion.

Ewing L. Lusk was one of the implementors of the reasoning programs LMA and ITP and has been actively involved in automated reasoning since joining the division several years ago. He is also one of the principal developers of a parallel programming approach that implements synchronization through monitors written as macros. He is currently Scientific Director of the Advanced Computing Research Facility.

James N. Lyness carries out research in the general area of numerical quadrature. He has constructed quadrature software and has developed evaluation methods for various classes of numerical software. At present, he is working on techniques for the Fourier-Laplace transforms and on the development of lattice rules for high-dimensional quadrature.

William W. McCune has been actively engaged in automated reasoning research. His recent development of the powerful automated theorem-proving program OTTER has already proved valuable in solving problems in mathematics and logic.

Jorge J. More played a lead role in the development of MINPACK, a collection of high-quality optimization subroutines now distributed worldwide. Currently he is working on an expanded version of the collection, with a focus on large-scale optimization and linearly constrained minimization software. He is also writing a book on the theoretical and practical aspects of optimization.

Ross A. Overbeek helped develop the LMA/ITP system that is widely used in the United States today. He is currently involved in exploring the use of advanced computational technologies (particularly logic programming) in genome sequencing.

Ping Tak Peter Tang joined our staff after a year of working in our division supported by special funding from the SDIO Systems/Phase 1 BM/C3 Technology Office. He is mainly interested in approximation theory and highly accurate numerical methods. His recent activities include design and testing of elementary function algorithms and development of robust algorithms for numerical linear algebra.

Larry Wos was winner (jointly with Steven Winker) of the first American Mathematical Society prize for current achievements in automated theorem proving. He has been active in the field for over twenty years. He introduced the notion of strategy to automated reasoning, devised new inference rules that have made problem solving easier, and even named the field of automated...
reasoning. He has co-authored a book on automated reasoning that has become the standard text in the field and has written a second book discussing open research problems in automated reasoning.

Stephen J. Wright carries out research in computational mathematics and parallel numerical methods. His recent work has been on stable parallel algorithms for ordinary differential equations and optimal control and on algorithms for constrained and nonsmooth optimization.

2.2 Wilkinson Fellow

The Wilkinson Fellowship is awarded competitively to a young researcher in computational mathematics. The fellowship is named after James H. Wilkinson, F.R.S. (1918–1986). Previous Wilkinson Fellows were Christian Bischof and Magnus Ewerbring.

Barry F. Smith is our current Wilkinson Fellow. His research includes analytic and computational aspects of domain decomposition. In particular, he is interested in the development of algorithms for the solution of multicomponent problems that arise in structural mechanics.

2.3 Postdoctoral Researchers

The MCS Division provides postdoctoral fellowships to promising researchers in various areas of computer science and mathematics. This year we have four postdocs.

Gui-Qiang Chen is the recipient of the first Argonne–University of Chicago Fellowship in Mathematics. He is actively engaged in developing new methods for the solution of nonlinear conservation laws and the equations of fluid dynamics.

Mark T. Jones is conducting research in software and algorithms for the parallel solution of the large symmetric generalized eigenproblem as it arises in structural engineering and physics. He is also interested in the parallel iterative solution of sparse systems of linear equations as they arise in applications.

Paul E. Plassmann has been working on parallel algorithms for problems arising in two areas: large-scale nonlinear optimization and sparse matrix factorizations. His most recent investigations have focused on the development of parallel interactive solvers for ill-conditioned, large sparse matrices.

Steven K. Winker has been investigating the feasibility of using logic programming for applications in molecular biology. He has also been active in automated reasoning research.
3 User Facilities

3.1 Advanced Computing Research Facility

To conduct the experimentation vital to much of our mathematics and computer science research, we operate an Advanced Computing Research Facility (ACRF). Currently, the ACRF includes eight commercial multiprocessing computers:

- A Thinking Machines Connection Machine – Model 2 with 16,384 nodes, each having 8 kilobytes of memory.
- An Active Memory Technology DAF 510/8 with 1,024 one-bit processors, each with 8 kilobytes of memory.
- A BBN TC-2000 (Butterfly II) with 45 processors, each having 4 megabytes of memory.
- A BBN GP-1000 (Butterfly I) with 96 processors, each having 4 megabytes of memory.
- An Alliant FX/8 system with 8 vector processors sharing 32 megabytes of memory.
- An Encore Multimax system with 20 processors sharing 64 megabytes of memory.
- A Sequent Symmetry S81 system with 26 processors sharing 32 megabytes of memory.
- A Stardent Titan graphics supercomputer with 4 vector processors and 64 megabytes of memory.

We have also acquired a Solbourne Enterprise server with 256 megabytes of RAM, upgradable to 1.1 gigabytes of RAM. This machine is an early representative of the large-memory workstation of the mid-nineties and will allow the ACRF to investigate a wide variety of issues that will face systems designers during the next decade. We hope to add 256 megabytes to make this machine especially effective for experiments in molecular biology, automated reasoning, computer algebra, and automatic differentiation.

Our objective in acquiring this diversity of machines is twofold. It provides an opportunity to attack questions regarding the use of each type of architecture (some with global memories and a small number of processors, some with local memories and many processors, etc.). Furthermore, the diversity of machines is essential for developing algorithms and programming techniques that achieve high performance on a range of different computer architectures, yet require little or no change to move from one system to another.

Our staff members access the ACRF computers through scientific workstations (principally Suns and NeXTs). These workstations provide a sophisticated and responsive environment for program development and graphics output.

3.2 The Intel Touchstone DELTA System

Early in 1991, Argonne became a charter member of the Concurrent Supercomputing Consortium (CSC). The CSC is headed by the California Institute of Technology (Caltech) and consists of more than a dozen national laboratories, universities, and funding agencies, as well as Intel Scientific
Computing, Inc. The purpose of the CSC is to advance the state of the art of scientific computing by providing access to high-performance massively parallel computers. Caltech has acquired a 576-node Intel Touchstone DELTA System, which will be available to CSC participants in the early summer of 1991. The DELTA System will be accesible for Argonne users through the ACRF. Argonne's participation in the CSC is supported by various programmatic divisions, as well as through special laboratory funds for high-performance computing.

As part of the acquisition of the Intel Touchstone DELTA System, the ACRF acquired an iPSC/860 machine to be used as a staging platform for software development for the larger machine. In November 1990, the ACRF also sponsored a short course to familiarize Argonne computational scientists with the Touchstone architecture.

### 3.3 ACRF Users

To encourage use of the ACRF advanced computers by off-site users, we have linked the ACRF to various national networks. By December 1990, the ACRF had more than 1100 registered users, of whom more than 400 were active each month.

We also encourage use of the facility by sponsoring an extensive visitors program, conducting special workshops on issues in advanced computing, hosting user group meetings, and offering three-day classes in parallel computing. Of particular note are the following:

- We held the fourth annual two-week Institute in Parallel Computing in September 1990. This institute introduces graduate students and postdoctoral researchers to new techniques in parallel programming and to state-of-the-art computer systems.
- We sponsored the first Affiliates Workshop for our industrial and academic affiliates.
- We began specialized classes to supplement the introductory courses we have offered for several years. Among the first of these specialized classes was one on PCN (a high-level parallel programming language developed at Argonne and Caltech); we also assisted in the first course on parallel programming for computational chemists.

![ACRF Resource Distribution](image1.png)

![ACRF Use by Discipline](image2.png)

**Figure 1: ACRF Users, 1990**
4 Research Supported by DOE – Office of Energy Research

Research in the Mathematics and Computer Science Division is focused on the development of algorithms and software, the analysis and application of mathematical methods for the solution of scientific problems, and the development of computer science methodology and tools to advance the state of the art of scientific computing.

4.1 Algorithms and Software

An essential part of our research program involves designing algorithms for the numerical solution of problems common to many scientific and engineering problems, and implementing these algorithms on high-performance computers.

4.1.1 Numerical Linear Algebra (C. Bischof, M. Jones, P. Plassmann, P. Tang)

Our current efforts in numerical linear algebra focus on the development of algorithms and software for solving large sparse-matrix problems on parallel machines. In particular, we are developing efficient software for orthogonal decompositions and variants of the conjugate gradient and Lanczos algorithms. Orthogonal methods are robust and reliable and are applicable to the solution of any equation system, be it overdetermined, underdetermined, or numerically difficult. On the other hand, for many problems, iterative methods based on conjugate gradient techniques have proven to be successful, but the choice of the right preconditioner remains problematic. We are pursuing methods that blend the power of those two approaches in producing fast and reliable codes for solving equations systems on distributed-memory machines. We are also incorporating these advances into codes for solving generalized eigenvalue problems, a fundamental task in engineering and physics problems.

The goal is to produce algorithms and software that are well suited for large-scale parallel environments and that can be used to solve numerically difficult problems reliably (as well as numerically easy problems with great speed). These characteristics are crucial for the study of ill-posed problems that arise in many applications (for example, reservoir modeling and atmospheric temperature profiles). The problems are challenging because they are large and usually sparse and their solution is highly sensitive to small perturbations in the data.

A major achievement has been the development of the LANZ software package for solving the large, sparse, symmetric generalized eigenproblem on vector supercomputers and shared-memory parallel computers. This problem is of significant practical importance in, for example, structural engineering where it arises in the vibration and buckling problems. The LANZ software combines the Jones and Patrick algorithm for indefinite factorization with advances in sparse matrix technology developed at Boeing Computer Services. LANZ has been successfully installed on computers such as the CRAY Y-MP and the Encore Multimax as well as Sun workstations. Documentation for the software has been produced and distributed to many users.

We continued our work on "incremental condition estimation," a monitoring and prediction technique useful in a variety of matrix factorizations. We gained a better understanding of the numerical behavior of this technique and are developing a toolbox that allows a user to easily
assemble incremental condition estimators with varying degrees of accuracy and cost and to custom-tailor an estimator to the matrix structure at hand. The software was incorporated into the orthogonal decomposition codes in LAPACK, a portable linear algebra library. Using our codes in the development of the MINPACK-2 software, B. Averick (joint appointment with University of Minnesota) obtained a two-fold speedup on the Alliant FX/8 over previous approaches. As an extension of this work, and in collaboration with G. Shroff (Caltech), we developed a monitoring technique for updates that are common in numerical optimization methods.

Matrix techniques have recently received significant attention in the signal processing community. In collaboration with C.-T. Pan (Northern Illinois University), we developed a matrix updating algorithm that is suitable for SIMD machines. Experimental results on the DAP-510 confirmed the superiority of this approach over standard methods.

We also made significant progress on improving the preconditioned conjugate gradient method, one of the most successful methods for solving large, sparse linear systems. In particular, we developed a new algorithm for computing incomplete Cholesky preconditioners, and we demonstrated the use of graph coloring heuristics to generate matrix reorderings that minimize the amount of interprocessor communication for the incomplete factorization method. Experiments on large structures problems have shown the computational promise of these improvements, especially on parallel implementations of this method.

Finally, we developed and implemented a new parallel sparse orthogonal factorization algorithm on a hypercube architecture. An advantage of this new algorithm is that the data structures used for the factorization can be reused to solve the matrix problem required in the solution of a optimization subproblem.

4.1.2 Optimization (J. Moré, P. Plassmann, S. Wright)

Optimization problems arise in a wide range of scientific, engineering, and economic applications. Most problems of interest involve a large number of variables, either because they are discretizations of infinite-dimensional problems (for example, optimal control) or because they are models of large and complex discrete systems (for example, airline scheduling). Our research centers on the development of algorithms and software for the solution of large-scale nonlinear optimization problems. Of particular interest are the optimization problems that arise in geophysical oil prospecting, modeling of inhomogeneous superconducting materials, and chemical kinetics.

A major accomplishment was a performance study of algorithms for the solution of large-scale bound-constrained problems on parallel computers. This research was presented by invitation at the Workshop on Large-Scale Optimization at Cornell University. The study centers on the solution of two large-scale optimization problems: the elastic-plastic torsion problem and the journal bearing problem. Performance issues were illustrated with the GPCG algorithm of Moré and Toraldo. This algorithm uses the gradient projection method to select an active set and the conjugate gradient method to explore the active set defined by the current iterate. We demonstrated that significant improvements in the performance of the GPCG algorithm can be obtained by using partitioning techniques in the parallel environment offered by the Alliant FX/8 and the Sequent Symmetry. We also demonstrated that these partitioning techniques lead to almost linear speedups on function-gradient evaluations and Hessian-vector products for partially separable functions. Typical speedups are 7.2 on the 8-processor Alliant and 17.6 on 20 of the processors of the
Sequent Symmetry. These results are important because partially separable functions arise in many significant large-scale optimization problems—in particular, when the Hessian of the optimization problem is sparse.

Another major achievement was the design of an interior-point algorithm for linearly constrained optimization. The obvious approach is to form a sequence of barrier functions by taking combinations of the (nonlinear) objective function and logarithmic terms representing the (linear) constraints and by finding minimizers of each function in this sequence. Instead, we use the two-metric gradient projection algorithm to choose its search direction, and we perform the projection onto the feasible set, inexactly, by solving a convex quadratic programming subproblem with an interior-point algorithm. Analysis showed that this algorithm has desirable local and global convergence properties. In ongoing work we are using interior-point methodology to solve optimal control problems. The resulting algorithms can handle quite general constraints and can be implemented efficiently on parallel computers.

In collaboration with V. Pereyra of Weidlinger Associates (Los Altos, California), we continued work on algorithms for the geophysical inversion problem. At its topmost level, this problem is solved by adjusting a set of parameters so that predicted data from a finite model of a study area gives an optimal fit to the observed data, in some sense. We investigated minimization of $l_2$ (least-squares) and $l_1$ functions of the residual vector, allowing linear constraints to be imposed on the variables. Both algorithms proved successful in solving a model problem, and the $l_1$ norm minimization method proved to be particularly robust when "outliers," arising from inaccurate observations, were introduced into the data. This is a desirable feature, since outliers are almost always present in real data.

In collaboration with T. Coleman (Cornell University), we developed an efficient parallel implementation of the Levenberg-Marquardt algorithm for solving nonlinear least-squares problems. This implementation introduces several new parallel matrix factorization algorithms and demonstrates good efficiency for large-scale problems. We also extended this work to large sparse systems. A new parallel sparse orthogonal factorization algorithm was developed and implemented on the hypercube architecture. An advantage of this new algorithm is that the data structures used for the factorization can be reused in the trust-region problem solution. This common data structure greatly increases the efficiency of an implementation.

4.1.3 Automatic Differentiation and the Numerical Analysis of Nonlinear Systems (A. Griewank)

The computational solution of virtually all nonlinear problems requires the evaluation of first derivatives. Moreover, once sufficiently accurate solutions have been obtained, their partials with respect to certain input variables form the basis of a subsequent sensitivity analysis. In unconstrained optimization, a superlinear rate of convergence can be achieved only if full second-derivative information is explicitly available or suitably approximated. The location and structural analysis of transition states in parameter-dependent systems call for the estimation of second and higher derivatives, depending on the nature of the singularity. To date, derivatives are still obtained mostly by tedious and error-prone hand calculations or through inaccurate and expensive divided differences. Automatic differentiation constructs an algorithm that evaluates the full vector or matrix of partial derivatives simultaneously with the underlying function at a given argument. In other words, automatic differentiation differentiates programs rather than algebraic expressions.
We have taken a leadership role in increasing the optimization community’s understanding of automatic differentiation and related techniques by organizing and directing a Workshop on the Automatic Differentiation of Algorithms. This workshop was sponsored by the Society for Industrial and Applied Mathematics (SIAM) and held in Breckenridge, Colorado, on January 6-8, 1991 (see Section 6.3).

We had significant success in devising and implementing several new automatic differentiation techniques in our new software package ADOL-C. We use an overloading approach that enables user-defined structures and functions to be handled without any difficulty, even if the latter are recursive. Compared with other software, ADOL-C is mainly distinguished by two desirable features. First, derivative vectors of arbitrary order are obtained at a complexity that does not depend on the number of independent variables. Second, the potentially large set of auxiliary data is generated and accessed in a strictly sequentially fashion, so that it can be rapidly paged out to disk. This self-contained “tape” can be used for other computational purposes (e.g., the concurrent scheduling of function and derivative evaluations).

We conducted various experiments to exploit parallelism in automatic differentiation. A self-scheduling scheme was implemented for shared-memory machines, which yielded good speedups on the Sequent Symmetry for function evaluations and gradient evaluations. In collaboration with Chris Bischof, we implemented an optimized execution schedule that is approximately twice as efficient as our earlier scheme. We also developed tools for automatically compiling and statically scheduling code on parallel machines. Our next efforts will focus on augmenting ADOL-C to recognize subprograms and (in particular) vector instructions, to decrease the memory requirements of the serial version and the processing granularity of the parallel version.

4.1.4 Domain Decomposition Algorithms (W. Gropp, B. Smith)

Many scientific problems—for example, in fluid dynamics and combustion—give rise to partial differential equations (PDEs). Numerical methods that are based on dividing the physical (or computational) domain up into many subproblems are called domain decomposition methods. Such a decomposition provides a natural method for parallelizing the solution of a PDE.

Our work has concentrated on methods for solving the system of linear equations that arise from implicit differencing methods (either finite differences or finite elements) applied to PDEs in two and three spatial dimensions. Much current research in domain decomposition methods is concerned with finding good preconditioners for iterative methods. A key to these preconditioners is a careful handling of the interactions between both with neighboring and distant domains. However, any interaction with other domains reduces the parallel efficiency of the computation. In addition, the finite element and difference models of elliptic problems in three dimensions have qualitatively different behavior from their two dimensional equivalents. Thus the domain decomposition algorithms we use must have some fundamentally different structure from those used for problems in two dimensions. Hence, a major part of this work is involved with determining the tradeoffs between the quality of the preconditioner and the parallel efficiency of that preconditioner.

This is done in several ways. From a theoretical perspective, an analysis of the conditioning of the preconditioned system gives bounds on the number of iterations needed to “solve” the linear system. This can be combined with a parallel complexity analysis to give estimates of parallel efficiency. We have added a number of results of this nature in the past year that show that these
methods are optimal or nearly optimal.

For problems such as finite difference methods applied to multicomponent problems, empirical studies are needed. To this end, we have concentrated on a set of non-selfadjoint variable-coefficient two-dimensional problems, including multicomponent problems, and on somewhat simpler three-dimensional problems. We compared domain decomposition methods against a variety of popular alternative approaches, including sparse direct and preconditioned iterative methods. These studies showed that domain decomposition methods are often superior, particularly on harder problems.

We also studied the parallel performance of domain decomposition on a range of problems and obtained good results for a variety of parallel machines, including a shared-memory machine (BBN TC-2000) and a distributed-memory machine (64-node Intel iPSC/860). Problems with hundreds of thousands (in two-dimensional) or millions (in three-dimensional) of unknowns have been solved.

An additional feature of domain decomposition is that it provides a suitable framework for applying other techniques, such as local refinement or asymptotics-induced differencing schemes. Our two-dimensional domain decomposition code was applied to the parallel solution of the nonlinear streamfunction-vorticity equations for flow over a backstep, with local refinement used to resolve the recirculant flow.

Figure 2: Solution of a 3-D Finite Element Problem with 912,576 Unknowns on an iPSC/860 Parallel Computer (Total solution time in sec. is given beside each data point.)
4.1.5 Optimal Control and Boundary Value Problems (S. Wright)

We have begun a new investigation focusing on problems that arise in the solution of banded linear systems of equations, optimal control of ordinary differential equations (ODEs), and two-point boundary value ODEs. A common feature of these three problems is that the relationships between the unknowns can be represented by a graph in which the connectivity pattern is essentially linear or circular. The “best” one-processor algorithms are, in each case, inherently serial. Parallel algorithms are constructed with the help of a recursive strategy, which is based on the principles of domain decomposition. A major accomplishment of the past year was the development of stable algorithms for two-point boundary value ODEs, which can be implemented efficiently in a vector or parallel environment. The most popular algorithms for this problem (collocation, finite differencing, multiple shooting) have aspects that are trivially parallelizable, but it is necessary in all of them to solve a linear system of equations which has an “almost block diagonal” structure. This is usually the most time-consuming part of the overall process. Parallel solution of this system has been an active area of research, but previously proposed algorithms lacked the stability properties that are necessary for robustness. We refer to our new class of algorithms as “structured elimination” methods. Members of this class were shown to give significant speedups over the best serial codes on parallel and vector machines. Both analysis and practical experience confirm the stability of the method. With the help of Ian Foster, the core algorithm was successfully implemented in the parallel language PCN.

A further achievement was the design of parallel algorithms for discrete-time optimal control problems. The most successful approaches to this problem are based on the application of a Newton or quasi-Newton method (or, in the case of bound-constrained problems, a two-metric gradient projection algorithm) to produce a sequence of linear-quadratic auxiliary problems, which are then solved by using dynamic programming. The latter method has the important advantage that it computes solutions to linear-quadratic problems with $N$ variables in $O(N)$ time, rather than the $O(N^3)$ complexity usually associated with calculation of Newton steps. It does, however, have the disadvantage that it is inherently serial. We have developed a partitioned dynamic programming technique, variants of which can be efficiently implemented on vector processors and on shared-memory and hypercube computer architectures. A recursive domain-decomposition methodology is used. The optimal number of levels of recursion depends on the characteristics of the architecture on which the algorithm is implemented.

4.1.6 Function Approximation Techniques (W. J. Cody, P. Tang)

Our research in function approximation techniques seeks to improve the quality of the primitive, elementary, and special functions essential to scientific computing and to advance the use of function approximation techniques in other areas of scientific computing.

A major accomplishment this year was the completion of SPECFUN, a new package of highly transportable Fortran special function programs and test drivers. Special functions are fundamental to the solution of many scientific and engineering problems such as problems in spectroscopy, heat conduction, molecular structure, and the design of sonar and radar antennas. A unique feature of the package is the test drivers. These test programs include accuracy tests based on carefully selected identities, and tests of the response to common misuse of the function programs. Equally important, they permit scientists to determine the quality of similar function programs from other
libraries. (At least one major commercial library has been modified because of test results obtained with this package.) No other general test programs of this type exist for special functions.

Another major accomplishment was our generalization of our accurate test method to a large class of elementary functions: \( \log_b(x) \) for arbitrary base \( b \), \( \sin x \), \( \cos x \), \( \tan x \), the inverse functions \( \arctan, \arcsin \), and \( \arccos \), and the power function \( x^y \). Among all elementary functions, the power function is the most difficult to compute. Based on our table-driven approach for elementary function computation, we invented a new algorithm that computes \( x^y \) to full machine accuracy. This algorithm was adopted by Sun Microsystems and MIPS Computer Inc.

We successfully conducted a detailed error analysis of the orthogonality problem. This problem is related to the divide-and-conquer algorithm that was left unsolved for a number of years. The analysis resolved this problem by proving that a new equation solver that we developed delivers the accuracy needed to guarantee orthogonality always.

We modified our package of portable C programs from the appendixes to the IEEE floating-point standards to improve portability, and we augmented the package to include two additional recommended functions: \( \text{finite}(x) \) and \( \text{isnan}(x) \).

We also completed work on the development of various algorithms and test programs, including preparation of the CELEFUNT package of Fortran test programs for the complex elementary functions, development of a Fortran program for the normal probability function, and testing of the arithmetic and elementary function library on the Sequent Symmetry computer.

4.1.7 Quadrature (J. Lyness)

The thrust of our quadrature research includes the design of quadrature techniques, the construction and evaluation of quadrature software, and the development of the underlying theory. New work focuses on lattice rules, a class of number theoretic rules associated with Koborov and Conroy for high-dimensional quadrature.

We achieved several major advances in our research on the generator matrix approach to lattice theory. Working with T. Sorevik (Bergen, Norway), we developed a theory that tackles the formidable task of handling geometrically equivalent lattices in a cost-effective way. With P. Keast (Halifax, Nova Scotia), we established a connection between the invariants of a lattice rule and the Smith Normal Form of the generator matrix of its reciprocal lattice.

In collaboration with I. H. Sloan (University of New South Wales, Australia), we are investigating the properties of a new set of quadrature rules; these are based on lattice rules but do not require equal weights. Preliminary results are encouraging.

We are also investigating the role of extrapolation-based quadrature in the boundary element method for a class of partial differential equations. The vital asymptotic expansions on which this quadrature may be based were mainly developed at Argonne in the 1970s. With standard techniques of classical analysis, these expansions were extraordinarily difficult to establish. Responding to recent pressure for more information about them, we revised the underlying theory. As a result of our work, we have been able to present the proofs in a systematic way, making them easier to understand. At the same time, we have broadened the range of validity of some of the expansions.
4.2 Applied Mathematics

Applied mathematics is concerned with mathematical equations, properties of their solutions, and ways to find these solutions numerically. The equations generally describe classes of physical phenomena, such as shocks and boundary layers in fluid flow, fronts in reaction-diffusion systems, and patterns in global climate evolution. The methods we use for their analysis are analytical as well as numerical, with the assistance of symbolic manipulation software where appropriate.

4.2.1 Fluid Dynamics (I-L. Chern, W. Gropp, G. Leaf)

A highlight of our research in computational fluid dynamics was the implementation of a nonuniform adaptive mesh algorithm based on the use of multigrid procedures for the solution of steady, two-dimensional incompressible Navier-Stokes equations. The adaptive mesh procedure uses a local estimator for the truncation error, which is a byproduct of the solution procedure. The serial version of the algorithm was used to investigate cavity flows and expansion/contraction flows (see Figure 3).

![Figure 3: Steady Incompressible Flow over Backward-Facing Step (Re = 400)](image_url)

In collaboration with N. Zabusky and X. Yang (Rutgers University), we developed a second-order Godunov code to investigate dipolar-vortex and jet formation in shock-accelerated density-stratified layers. We showed that this formation is the fundamental mechanism of the Richtmyer-Meshkov instability in a density-stratified flow.

We also investigated parallel numerical algorithms for PDEs on the sphere. This effort was motivated by our work in climate modeling (see Section 5.2). We developed a control volume method that is linearly scalable on parallel machines. The well-known pole problem is avoided by the use of a three-dimensional Cartesian coordinate system to represent the equations and by the use of an icosahedral grid on the sphere.

In support of the global climate modeling effort (see Section 5.2), we successfully analyzed the nondivergent barotropic vorticity equation on the sphere, proving existence and uniqueness and establishing the regularity properties of the solution.
4.2.2 Conservation Laws (G.-Q. Chen, I-L. Chern)

Our research on the propagation of oscillations for nonlinear hyperbolic systems of conservation laws affirmed a long-standing conjecture. With the aid of geometric measure analysis, we established a rigorous framework for 2 by 2 systems with at least one linearly degenerate field. We proved that the initial oscillations propagate along the linearly degenerate field and cancel along the genuinely nonlinear field.

In earlier investigations of strictly hyperbolic viscous systems of conservation laws, we had shown that perturbations of a constant state evolve into diffusion waves over long periods of time. We generalized this result to nonstrictly hyperbolic viscous systems, where multiple-mode diffusion waves are formed, and we applied this result in a study of three-phase flow of oil reservoirs.

In a joint program with A. Rustichini (Northwestern University), we continued to study the formal relationship between systems of conservation laws and game theory. We established a general framework to solve global solutions to nonzero sum dynamic games by means of conservation laws. By carefully analyzing nonlinear wave behavior, we constructed global solutions to a system of conservation laws. We then used these results to solve some nonzero sum game problems.

In a joint project with E. Tadmor (Tel-Aviv University) and Qiang Du (University of Chicago), we developed a spectral method for discontinuous solutions to multidimensional conservation laws. We studied the stability, truncation errors, and convergence of this method for multidimensional scalar conservation laws. We also provided numerical evidence of the efficiency of this method.

Finally, we studied the long-time behavior of solutions of Lax-Friedrichs finite difference equations for hyperbolic conservation laws. We found that the solutions form several discrete diffusion waves propagating at constant characteristic speeds in the long time. This study quantifies error propagation in the Lax-Friedrichs scheme.

4.2.3 Combustion (H. Kaper, G. Leaf)

We investigated global solutions to the compressible Navier-Stokes equations for a reacting mixture describing dynamic combustion. Using the maximum principle and the method of energy estimates, we established existence theorems for global generalized solutions of the initial boundary value problems and the Cauchy problem. We also studied the asymptotic behavior of the generalized solutions; we identified the different asymptotic behavior of solutions with impermeable thermally insulated boundaries and with boundaries prescribed by the boundary data.

As part of our ongoing investigations of the structure and stability properties of of flames and fronts in low-Mach number combustion, we analyzed with M. Garbey (Université Claude Bernard, Lyon, France) the mechanism for the transfer of stability between spinning and standing combustion waves. The analysis in solid surface combustion involved the reduction of a system of nonlinear partial differential equations describing the processes of transport and combustion to a system of ordinary differential equations describing the dynamics of the combustion front. For the algebraic manipulations, we used the symbolic computation program MAPLE. The use of such a program was crucial because the manipulations needed in the higher-order asymptotic calculation would have been prohibitive if done by hand.

We also began an investigation of fluid/flame interactions. We developed a low-Mach number model describing tubular flames in a cylindrical burner. This model couples the fluid, thermal, and
chemical processes in a reactive flow. As a first step, we restricted our attention to radial flow in the symmetry plane of the burner. We implemented an algorithm describing this one-dimensional flame. The flow is decomposed into a potential component and an incompressible component. The potential flow is described by a Poisson equation coupled to thermal and chemical effects through the sources. The incompressible flow is described by an inhomogeneous, incompressible Navier-Stokes equation with a variable density. The energy, species, and equation of state complete the set of governing equations. Spectral collocation is used for spatial discretization combined with a semi-implicit time discretization. Dynamic regridding is used to resolve the flame.

4.2.4 Reaction-Diffusion Systems (H. Kaper, M. Kwong)

We continued to progress in our research on reaction-diffusion equations. Our investigations, which are concerned primarily with questions relating to bifurcation phenomena and chaotic dynamics, have been focused on positive solutions (ground states) of nonlinear differential equations of the type $u_t = \Delta u + f(\lambda, u)$. These equations arise typically in problems related to flow in porous media, combustion, and chemical reactors.

An interesting application of our results occurred recently in plasma physics, where an equation of the type above, with $f(\lambda, u) = u^p - \lambda u^q$ ($0 < q < p \leq 1$, $\lambda > 0$), provides a simple mathematical model of Tokamak equilibria with magnetic islands. The central question concerns the existence of a radially symmetric solution that does not change sign inside a bounded region and vanishes with zero slope on the boundary. The location of the boundary is unknown and is to be found as part of the solution. Preliminary numerical experiments were inconclusive. We were able to show that such a solution exists and is indeed unique, predicted the behavior of the boundary as $p$, $q$, and $\lambda$ are varied, and developed a reliable numerical procedure, which confirmed these theoretical results.

In general, our investigations have been concerned with nonlinearities of the type $f(\lambda, u) = u^p + \lambda u^q$ with $\lambda$ positive or negative. With this type of nonlinear source term, there is a delicate balance between the exponents $p$ and $q$, the sign of $\lambda$, and the dimension $N$ of the domain; and solutions of prescribed type may or may not exist in the various ranges of these parameters. Also, the methods needed in the proofs tend to vary from case to case. Among our recent accomplishments we mention a proof of a long-standing and well-known conjecture of Brezis and Nirenberg concerning the uniqueness of the ground state when $f(\lambda, u) = u^p + \lambda u$, where the exponent $p$ is critical or subcritical (i.e., $p \leq (N+2)/(N-2)$) and $\lambda$ is positive. Our approach involved a new application of the Lyapunov energy functional of the equation, use of the Sturm comparison principle, and an appropriate scaling of the solutions.

Much of this work is done in collaboration with academic researchers: L. Zhang (University of Minnesota), Y. Li (University of Chicago), C. Bandle (University of Basel), and others.

4.2.5 Superconductivity (H. Kaper, M. Kwong)

The complex Ginzburg-Landau (CGL) equations provide a simple mathematical model for the equilibrium thermodynamic properties of inhomogeneous superconductors. Their analysis is complicated because of the rich structure of the solution space and poses a formidable challenge in applied mathematics. Our goal is to study bifurcation phenomena and the dynamical behavior of
solutions of the CGL equations relevant to superconductors.

In collaboration with P. Takac (Vanderbilt University), we analyzed a time-dependent form of
the CGL equations (see Figure 4). (The same equation also arises in hydrodynamic stability theory,
for example in the description of Tollmien-Schlichting waves in plane Poiseuille flow, convection rolls
in the Rayleigh-Bénard problem, and Taylor vortices in the Couette flow between counterrotating
circular cylinders.) We established the existence of stable 2-tori for the equation in one spatial
dimension with quasi-periodic boundary conditions. The quasi-periodic boundary conditions imply
periodicity for the macroscopic quantities (superconducting electron densities and currents, field
strengths), but not necessarily for microscopic quantities (order parameter). Our results indicate
that the conclusions reported in the literature, indicating the existence of stable 2- and 3-tori under
periodic boundary conditions, are based on incomplete numerical evidence.

Figure 4: Density of Superconducting Electrons for Interaction between Two
Rotating Waves in the Ginsburg-Landau Equation

4.3 Computer Science

Computational science research addresses ways in which researchers can reduce the time required
to write programs, increase their adaptability to high-performance computers, and enhance their
clarity and correctness. For example, we are developing parallel-programming tools for transport-
ing programs to new computer architectures. We are also investigating abstract programming
techniques for writing programs more accurately and efficiently. In addition, we continue to work
with applications-oriented groups on projects such as the use of program transformation to develop
communications-oriented software.
4.3.1 Logic-based High-Performance Computing (I. Foster, E. Lusk, W. McCune, S. Winker)

The field of logic-based high-performance computing has seen tremendous expansion in the past several years. Our interest in this field is a direct outgrowth of our research in automated reasoning, logic programming, and parallel programming languages. The objective is to develop, implement, and use logic-based tools for the solution of scientific problems on high-performance computers. Current emphasis is on applications in parallel Prolog, graphics tools for logic transformations, parallel automated reasoning, and computational biology.

Parallel Prolog. Motivated by the desire to produce the fastest possible automated reasoning systems capable of exploiting state-of-the-art computers, we have been exploring parallel computing and logic programming. To carry on this research with world leaders, we formed a collaborative project, informally known as the Gigalips project, with the University of Bristol and the Swedish Institute of Computer Science. The project has resulted in Aurora Prolog, an experimental OR-parallel logic programming system based on the existing Prolog language. It is particularly appropriate for shared-memory multiprocessors. During the past eighteen months (in collaboration with S. Mudambi of Brandeis University), we adapted the Aurora parallel Prolog system for the BBN TC-2000 architecture. This system has thus become the fastest platform by far for Prolog programs.

Graphics Tools for Logic Transformation. In collaboration with a summer student T. Henry (University of Tennessee), we developed a window-based program for constructing, displaying, and managing first-order logic formulas (in OTTER language). The main motivation for constructing this program, called FormEd, was the desire to have formulas displayed in a readable, two-dimensional format. Users of FormEd can make two kinds of transformation on formulas: (1) logic transformations, such as negation normal form (NNF) translation, which preserve the meaning of a formula, and (2) edit transformations, which can be used to make arbitrary changes, such as adding a hypothesis to a subformula. Most transformations can be made to the entire displayed formula or to a selected subformula. The sequence of transformations leading to the current formula is saved so that any or all of them can be "undone" and "redone." Currently, FormEd can be used as a stand-alone program to help students learn formal logic, to study properties of normal-form conversion, and to help prepare input for a resolution/paramodulation theorem prover.

Parallel Automated Reasoning. Automated reasoning provides an unusually challenging arena for the study of fundamental parallel computing methods and tools, because of the extremely irregular nature of the computations involved. The available parallelism and grain size vary widely from problem to problem and even over time during the solution of a single problem. We are well positioned for this work, having the fastest sequential theorem prover and an active user group, together with tools for expressing both shared-memory and distributed-memory algorithms and tools for studying their behavior. During 1990, we constructed a parallel automated reasoning system based on our sequential program OTTER and the parallel closure algorithm developed last year. The resulting system, called ROO, was implemented using the parallel programming system p4 (see Section 4.3.2). Using ROO, we obtained substantial speedups (as high as 22 on a 24-processor Sequent Symmetry) on a wide range of problems.

Computational Biology. One major accomplishment was the use of our logic programming toolkit to design a tailored computer search of ribosomal RNA, focusing on the tetra-loops. Tetra-loops are tight hairpin loops within the secondary structure of an RNA molecule. Although the loops are easily located manually, their composition varies widely. Many questions of interest to
biologists regarding these loops require extensive computer searches for their answers. The detailed nature of a specific biological question can make it unsuitable for input to a general database inquiry program. We were able to identify tetra-loop composition in 16S rRNA. This work, conducted with C. R. Woese and R. R. Gutell (University of Illinois), marks a milestone in our investigations of logic programming: The research has been recognized as making a solid contribution to biology, rather than simply presenting a faster technique or a new algorithm.

We investigated the use of logic programming to find the most reliable signature positions of the ribosome in *E. coli*. "Signatures" are features within the ribosomal sequence that characterize certain groupings or phylogenetic subgroups of bacteria. We developed (with C. R. Woese) a comprehensive set of signatures for the Bacteria and the Archaea. Previously, researchers had only "anecdotal" knowledge of such signatures. The new technique obtains the frequency of exceptions to signature patterns. From this data, we then determined which signature positions are most reliable.

We also investigated the potential of parallel computing for large-scale numerical computations in molecular biology. This work led to the design and validation of a parallel algorithm that successfully aligns RNA sequences. Preliminary experiments indicate that our method is markedly superior to other techniques.

We generalized our covariance procedure (used last year to detect secondary structure in RNA previously unknown to biologists). The new generalized procedure in principle allows researchers to use their own covariance techniques while exploiting the advantages of our automated program based on logic programming.

Finally, in collaboration with T. Kazic (Washington University) we used logic programming techniques to develop a prototype database for reactions involved in anaerobic catabolism of glucose and pyruvate, including information on pathways, substrates, products, catalysts, class of reaction, kinetic and thermodynamic parameters, and the conditions for and consequences of a given reaction. The database integrates biochemical information with genetic data on gene location, structure, and orientation.

### 4.3.2 Software Tools for Parallel Computing (*W. Cowell, I. Foster, E. Lusk, R. Stevens*)

As the potential utility of parallel processing becomes increasingly apparent, the need for a software technology that allows a straightforward use of the next generation of parallel machines also increases. The focus of our research is the development of a software technology that allows effective use of highly parallel multiple-instruction multiple-data (MIMD) systems. The technology that we are developing is intended to simplify parallel application development at three levels:


2. A portable implementation of the high-level language PCN provides a more expressive notation that simplifies development and maintenance of complex programs. Interfaces to Fortran and C allow reuse of existing code in "bilingual" programs.

3. Libraries of algorithm templates allow users to develop parallel applications by reusing parallel program structures for which performance and scalability issues are well understood.
P4. We radically overhauled the old "Argonne macros" system for low-level portable programming of parallel computers. The C library was completely rewritten to enhance portability, simplicity, and efficiency. This work was in collaboration with R. Butler (University of North Florida). The new system is called p4, after the title of the book we wrote about the earlier system, *Portable Programs for Parallel Processors*. The system has now also been ported to a wide set of machines: Intel iPSC/860, Sequent Balance and Symmetry, Encore Multimax, BBN TC-2000 and GP-1000, Sun 3 and Sun 4, AT&T 3b2, and IBM RS 6000. A reference manual has been written, and a tutorial document is under way.

**Templates.** Our work on templates has two goals: to develop tools that simplify the task of generating new templates, and to identify key templates for interesting application areas. In 1990, we made progress in both areas. We constructed a prototype implementation of a template generation tool called the Argonne Program Synthesizer (APS), in Strand, and completed the design of a second-generation tool to be implemented in PCN during 1991. We used APS to implement a variety of templates, for problems as diverse as grid problems (numerical solution of partial differential equations), genetic algorithms (molecular dynamics), and load balancing (search problems).

**Validation of Tools.** A key part of our work is the validation of tools and techniques in both small test problems and large applications. Hence, we worked closely during 1990 with mathematicians and application scientists to develop several parallel application codes—for example, nuclear magnetic resonance analysis (using a parallel genetic algorithm, with scientists from Argonne's Chemistry Division); climate modeling (a variety of algorithms for solving shallow water equations, with mathematicians from the Mathematics and Computer Science Division involved in global climate modeling); weather modeling (parallelization of MM4, a mesoscale weather model, with scientists from Argonne's Environmental Assessment and Information Sciences Division); two-point boundary value problems (with a mathematician from the Mathematics and Computer Science Division). Our experience developing these codes has been positive and has resulted not only in greater understanding of our tools but also in the ability to develop sophisticated algorithms very quickly.

**Performance Visualization Tools.** The Gauge performance profiling tool, originally developed in the context of logic programming systems, was integrated into the PCN programming environment. Some of this work was done in collaboration with C. Kesselman (Aerospace Corporation).

We completed work on three interrelated tools that collectively make up Atrace, a graphical performance visualization tool. Atrace provides for the run-time creation and post-mortem analysis of execution traces. Three different tools display the logfiles in different ways: as a set of parallel time-lines, as an animated set of communicating processes, and as a set of statistical graphs. Atrace was used extensively in the tuning of ROO (see Section 4.3.1). It was also integrated with the Strand parallel programming system.

We also developed an X-based graphics tool called upshot. This tool enables one to view log files produced by parallel programs (see Figure 5), and thus provides insights about parallel program behavior. A users manual for upshot is being completed.
**Communication Optimizations.** We designed a framework for local refinement of communication structures in PCN programs. This allows commonly used PCN communication patterns, such as producer/consumer stream communication, to be replaced with low-level send/receive operations (e.g., p4 operations). These refinements can be performed with only local modifications to a program. The Gauge performance tool integrated into PCN allows the user to determine when communication optimizations of this sort need to be performed.

**Loop Analysis.** We used VecPar_77 to analyze an adaptive multigrid code for the solution of the Navier-Stokes equations for steady two-dimensional incompressible flows. VecPar_77 is a commercial product of the Numerical Algorithms Group, Inc., which was developed from prototype research tools emerging from earlier Argonne research. The tool helped to locate data dependencies that inhibit parallelism. We then parallelized the multigrid code using loop concurrency on a shared-memory multiprocessor. Loop concurrency is well suited to the multigrid code. The coarse grid in this code is grouped into a set of subdomains that serve as the basic unit involved in all operations; these operations are then implemented by loops over the subdomains in the refined grid. Preliminary performance tests on a Sequent Symmetry indicate that about 95% of the computational time (in serial) is spent in these loops. For a problem modeling a backward-facing-step flow, we obtained a speedup of about 13.6 when 20 processors were active.

4.3.3 **Automated Reasoning (E. Lusk, W. McCune, R. Stevens, S. Winker, L. Wos)**

The objective of the research in automated reasoning is to enhance our reasoning programs, with the goal of eventually producing a program that can function at the level of a colleague. Our approach is to develop theory and then implement the developments and test their value with a variety of difficult problems. In addition, we frequently attempt to answer open questions with the
aid of our reasoning programs, an action that has led to important contributions to mathematics and logic and to the formulation and introduction of new strategies and inference rules. Our current theorem prover, OTTER, provides a solid laboratory for studying refinements of higher-level automated reasoning ideas. OTTER itself is undergoing constant improvement, in order to ensure that it is as fast as possible and is able to handle very large problems.

A highlight of our research (conducted with R. Veroff, University of New Mexico) was the complete formulation of the linked inference principle, in which equality plays no essential role. Experiments with this new class of inference rules produced marked improvements (as much as a factor of 75 reduction) in the CPU time required to obtain proofs.

We also developed several new strategies for improving the effectiveness of proof finding. The "iterative strategy" blocks the use in the current run of formulas found to be absent from the set of proofs found in an earlier run; the "depth-first strategy" complements the breadth-first strategy already available in our automated reasoning program OTTER; the "proof checking strategy" uses the ancestors to determine which clauses to keep and which to purge by means of demodulation; and the "resonance strategy" seeks a proof by directing the search through the use of weight templates that mirror the steps of other proofs from the same field. We also formulated a long sought-after refinement of the set of support strategy; with the passive list, the refinement blocks the retention of clauses subsumed by a clause that is provably dependent on the axioms of the investigation.

In the spring of 1990, we succeeded in obtaining the first computer proof of the theorem known as Łukasiewicz's fifth conjecture, when phrased as a problem for condensed detachment. The result has added interest in that it was obtained with a combination of proof finding and proof checking techniques and, even more, because it provides a new benchmark problem for automated reasoning programs. Indeed, without substantial guidance, no computer proof has yet been obtained with a reasoning program; as a measure of the difficulty of the problem, our effort that generated more than 983,000,000 clauses did not succeed.

In August, we held a Theory Institute in Automated Reasoning focusing on the close coupling between representation, inference rule, and strategy. As a result of the institute, we formed a new collaboration with D. Scott (Carnegie Mellon University) on studies in sentential calculus. He suggested 68 theorems for consideration by our program OTTER, and we succeeded in proving all 68 in a single run. The studies culminated in the discovery of shorter proofs and a new axiom system.

In January 1991, we succeeded in greatly simplifying Kalman's axiomatization of the left group (LG) calculus. Kalman's axiomatization contains five axioms with a total of 92 symbols. Extensive use of OTTER resulted in the discovery of single-axiom systems. With OTTER, we obtained similar results with the right group (RG) calculus.

4.3.4 Abstract Programming (J. Boyle)

Research in abstract programming seeks to discover high-level, machine-independent ways to think about programming and to express programs. This approach to programming extends the idea of modularity used in ordinary programming to the point where virtually all aspects of a program are expressed independently of one another. Such extreme modularity means that the abstract program is more useful than an ordinary program, because it is easy to reuse parts of the abstract program in new applications; the abstract program is also more flexible, because the implementation of
one part can be changed without affecting other parts. Important to the practical application
of abstract programming is the ability to derive efficiently executable programs from abstract
specifications. Thus, the study of abstract programming entails the study of program derivations
and of techniques, such as program transformation, for carrying them out. Our long-term goal is
to reduce the time required to write programs, to increase their adaptability to novel hardware
architectures, and to enhance their clarity and correctness.

A major accomplishment was achieving “parity” on the CRAY X-MP computer. By parity,
we mean that the program derived from a functional specification executes as fast (or faster)
that a hand-written program for the same application. To achieve parity, we wrote a functional
specification in pure Lisp for a numerical algorithm that uses cellular automaton techniques to solve
one-dimensional hyperbolic partial differential equations (PDEs). We used the TAMPR program
transformation system, together with newly developed optimizations (some specific to the problem
and others tailored to the CRAY) to produce a Fortran program optimized for high-performance on
the CRAY X-MP. Based on this research, we began work on the two-dimensional problem, revising
the functional specification for the hyperbolic PDE solver to make it completely independent of
the dimension of the problem except at the lowest levels of abstraction.

In cooperation with T. Harmer (the Queen’s University of Belfast), we ported the TAMPR
program transformation system to the NeXT scientific workstation. In the process, we have sim-
plified porting it to other Unix environments by reducing the system’s dependence on the Allegro
Common Lisp system. In addition, we have developed a graphical user interface for preparing
transformations and applying them to programs. This interface was easy to develop using the
NeXT Interface Builder tool.

4.3.5 Automated Program Verification (K. Dritz)

This project seeks to develop models, axioms, and proof strategies for the automated verification of
programs that feature floating-point arithmetic—the one aspect of programming language seman-
tics usually ignored in theorem proving and automated program verification as being too difficult
to characterize, even when integer and nonnumerical operations are well understood. The problem
with floating point is twofold: its operations (unlike integer and nonnumerical operations) are only
approximations to their mathematical counterparts, and the approximations vary with the imple-
mentation (i.e., with the underlying hardware and/or the compiler). The basis of our approach is
to appeal to the Brown model of floating-point arithmetic for the necessary semantic definitions,
in terms of result intervals or error bounds, and to couple that with an appropriate means of ex-
pressing the expected behavior of a floating-point program or fragment in similar terms. While the
Brown model has influenced the design and formal specification of programming languages (such
as Ada), it has not yet been exploited in the field of automated program verification. Our goal is
to contribute to the underlying theory and practice of automated program verification as applied
to floating-point programs and to produce a practical means of proving their correctness (relative
to accuracy claims); at present, the only way to validate floating-point programs is to test them
sufficiently, and that approach amounts only to a failure to demonstrate the existence of bugs.

We changed our axioms, which had previously modeled floating-point values as intervals, to
a form in which floating-point values are modeled as numbers whose uncertainty is captured by
appropriate qualifications or constraints. This change increased the realism of our model by, for
example, clarifying that two different uses of the same variable (with no intervening assignment)
while both yielding effective values that might come from some relatively large interval—must necessarily both yield effective values that fall within the same very small interval situated somewhere within the larger interval. We demonstrated the suitability of our new axioms by obtaining automated proofs of Brown's "effective value lemma" and "small relative-error theorem," which will ultimately play the role of axioms (lemmas) in our approach to automated program verification of floating-point programs. Our proofs were obtained using only linked UR-resolution.
5 Research Supported by DOE – Office of Health and Environmental Research

Under special funding from the Office of Health and Environmental Research (OHER), we are addressing two Grand Challenge areas: molecular biology and parallel climate modeling.

5.1 Molecular Biology (I. Foster, R. Overbeek, S. Winker)

We are using the logic programming language Prolog to develop solutions to problems brought to our attention by biologists. Prolog permits rapid development and testing of algorithms; and it offers considerable computational power with relatively small programmer effort, both in logical inference and database development. Our major activities include the following.

Simulation of sequence reconstruction from oligo-hybridization data. This work (conducted with visitors from the Genetic Engineering Center of Belgrade, Yugoslavia, led by R. Drmanac) involves establishing an environment for simulating the reconstruction of genetic sequences from oligo-hybridization data. One sequence of especial interest is lambda (from GenBank), the largest known sequence. We have successfully developed algorithms in Prolog (in a few days) that took months to implement using more conventional technology. We are now conducting simulations to identify the critical parameters for the oligo-hybridization technology. Of particular concern are the accuracy assumptions under which the sequence can be constructed and the exact oligos required for partial sequencing.

Multiple sequence alignment and the related problem of inserting sequences into a large existing alignment. This project (conducted with C. Woese and G. Olsen, University of Illinois at Urbana) involves the evaluation of a number of multiple-sequence alignment algorithms against the alignment of the small ribosomal subunit maintained by researchers at the University of Illinois. This alignment now contains over 400 sequences (each over 1,400 characters in length) and will probably include 1,000 to 2,000 within two years. We have developed an algorithm of linear complexity that successfully partitions the alignment into relatively short segments; we are currently extending this work into a complete algorithm for aligning sets of RNA sequences.

Formation of contigs. This research (conducted with C. Woese) was carried out in the context of constructing physical maps from partial digest data. We are currently awaiting further data from the biologists.

5.2 Parallel Climate Modeling (I.-L. Chern, I. Foster, W. Gropp, R. Stevens)

We have been collaborating with the National Center for Atmospheric Research and with Oak Ridge National Laboratory in DOE's CHAMMP (Computer Hardware, Advanced Mathematics, and Model Physics) Modeling Program in support of parallel climate modeling. Efforts during the past year focused on four areas: scalability analysis, parallel algorithms and numerical methods, tools, and load distribution. We also held two workshops.

Historically, the scalability of parallel algorithms has been determined by trial and error. This approach is not possible in the case of the Advanced Climate Model (ACM), as the target computers do not yet exist. Nor is it justifiable, given the high cost of implementing a model. Fortunately,
scalability analysis can enable us to predict algorithm performance on future computers. This year we successfully completed a comprehensive analytic and empirical analysis of the parallel scalability of the spectral transform method, widely used in climate models. We also conducted an empirical study of parallel implementations of the method applied to the shallow water equations. The relevance of the results to full climate models was established. Our research introduces sorely needed facts to the debate about numerical methods for proposed teraop computers.

We developed solution techniques and parallel algorithms for the spectral transform method as well as for the composite mesh finite-difference method and the control volume method. Prototype parallel implementations have been constructed and are being used in empirical studies in support of the scalability analysis. We expect one of these methods to form the basis for the future advanced climate model. Completion of this work will provide the information required to decide which method is most appropriate.

The use of high-level programming tools is an important component of our strategy for developing a portable and maintainable ACM. We have evaluated two programming systems, Strand and PCN, and have found that they greatly simplify the task of developing and maintaining parallel codes. We are currently evaluating the performance of codes constructed using these tools in order to quantify overheads attributable to these tools. We also prototyped a graphics-based user interface for the ACM. Our prototype showed that it is possible to provide an environment that enables users to design and execute climate modeling experiments without becoming familiar with details of the underlying implementation.

Finally, we have focused on the issue of load distribution. The amount of physics computation performed at different grid points in a global climate model can differ significantly depending on the values of physical variables. Irregular distribution of computational requirements is a potential source of inefficiency in a parallel model. To quantify these load imbalances, we conducted an experimental study using an existing model, CCM1. Significant static and seasonal load imbalances were observed; diurnal effects are minimal. Load typically varies by a factor of two between grid cells. The impact of this variation on expected performance of a parallel model is being investigated. We have also begun investigating several techniques for distributing physics computations to the nodes of a parallel computer in order to correct load imbalances. These include simulated annealing and simpler “greedy” algorithms based on run-time data collected by our high-level programming tools.

![Figure 6: Predicted Throughput vs. Resolution on Parallel Computers](image)
6 Special Activities Supported by Non-DOE Funding

In addition to funding from the Department of Energy, the Mathematics and Computer Science Division receives a small amount of support from other agencies. The work performed generally complements the main research interests of the division. Here we highlight some of our accomplishments.

6.1 NSF Funding

The MCS Division receives funding from the National Science Foundation to support several projects, principally in association with colleges and universities.

6.1.1 CRPC

The National Science Foundation Science and Technology Center for Research in Parallel Computation (CRPC) provides funding to Argonne to conduct basic research in parallel computing and to provide access to the array of advanced computers in the Advanced Computing Research Facility (ACRF). During 1990, CRPC funds were used to upgrade the BBN TC-2000. Research projects included the following:

- PCN. One of our goals for 1990 was to explore the practicality of extending the concurrent programming language Strand, which we had previously studied intensively during 1989, to provide better support for scientific applications of interest to DOE. Working with researchers at Caltech and the Aerospace Corporation, we designed and constructed a prototype implementation of a language similar to Strand, called PCN, and evaluated the utility of this language in a number of applications.

- A pilot project aimed at achieving an efficient parallel implementation of the ADOL-C automatic differentiation package (see Section 4.1.3). The project concentrated on the reverse mode, using a dependency queue. On the Helmholtz energy function with 300 variables, we obtained a speedup of 11 on 15 processors of a Sequent Symmetry. This result shows that reasonable speedups can be obtained even without transformations to improve the granularity of the gradient evaluation.

- Development of a system for constructing a computational graph from ADOL-C’s computational trace, as well as two transformations for this graph, hoisting and splitting, which improve its computational granularity and reduce contention, respectively. Results obtained in a large shallow-water test problem showed a significant improvement in granularity.

- Study of the use of concurrency for function-gradient evaluations and Hessian-vector products of partially separable functions. Our main result shows that the use of parallelism in these computations usually leads to a synchronization problem but that it is possible to avoid this synchronization problem by a suitable partitioning of the element functions. This result is of interest because we also show that these partitions can be obtained automatically for any partially separable function.
6.1.2 Summer Institutes in Parallel Computing

Twenty-five graduate students and postdoctoral researchers participated in each of two two-week summer institutes in parallel programming at Argonne National Laboratory on September 5-15, 1989, and on September 4-14, 1990. The institutes are part of a continuing effort by our division to familiarize young researchers with new techniques in high-performance computing. Each day of the two-week institutes feature lectures by leading researchers in advanced computing on such topics as dataflow systems; the Tera computer and the Connection Machine; programming support environments; program composition; and pipeline architectures. The institutes are sponsored jointly by the National Science Foundation and the U.S. Department of Energy.

6.1.3 Minority Internship in Parallel Computing

Under a Minority Internship in Parallel Computing funded by the National Science Foundation, Amos Joel Carpenter (Butler University, Indiana) is spending a year in the Mathematics and Computer Science Division at Argonne. The internship is intended specifically for a candidate who is interested in acquiring knowledge and experience in the area of parallel processing. Since his arrival late this summer, Carpenter has familiarized himself with Argonne's p4 package and has been using that package to parallelize his codes in approximation theory.

6.1.4 Workshop on Asymptotic Analysis and Numerical Solution of PDEs

Forty mathematicians from the United States and Europe met at Argonne National Laboratory February 26-28, 1990, to participate in a workshop on Asymptotic Analysis and the Numerical Solution of Partial Differential Equations. The workshop was motivated by recent strides in asymptotic analysis, coupled with new results in numerical methods for the solution of partial differential equations (PDEs). The workshop consisted of two parts: a formal program of invited presentations and contributed papers, and informal working sessions. Among the applications discussed were physical oceanography, quantum chemistry, fluid dynamics, combustion, and reaction-diffusion systems. The workshop was supported by the National Science Foundation and the U.S. Department of Energy. Proceedings of the workshop have been published by Marcel Dekker as part of the series *Lecture Notes in Pure and Applied Mathematics*.

6.2 SDI Funding

The MCS Division receives funding from the Strategic Defense Initiative Office Systems/Phase 1 BM/C3 Technology Office to conduct research in parallel programming and algorithm development.

Together with Argonne’s Engineering Physics Division, we have been examining performance and other properties of parallel SDI algorithms. We completed work on the Connection Machine (CM-2) implementation of a Sparta prototype weapon-target accessibility algorithm; this work has been documented in an Argonne report. The work is significant in that we have developed optimized versions that run on a broad class of machines from modest size (8-24 processors) MIMD to massively parallel (16K to 64K processors) SIMD machine architectures with near-optimal speed gain.
We are also developing a parallel multisensor-multitarget algorithm front-end to the accessibility code. In the current prototype, a sensor node representing different ground and space-based sensors supplies signals (with Gaussian noise added) to tracking nodes. Each track is assigned a separate processor node to handle sensor fusion and correlation by using an extended Kalman filter approach. A master node gathers track information and graphically displays the estimated (processed) tracks superimposed on a depiction of the globe during the simulation. The system currently handles maneuvering, noncrossing tracks. The simulation is running in distributed-processing fashion on a network of Sun workstations to focus on internode communication issues. Our next task is to add accessibility and weapon-target assignment functions, and to map various nodes to appropriate parallel or sequential machine architectures.

As an outcome of the SDIO-sponsored conference at Tysons Corner in August 1989, we have been studying the feasibility of fault-tolerant parallel processing for SDI applications. We developed an experimental program focusing on the interactions between fault tolerance, parallel processing, and security. The goals are to determine the impact of these issues during design and verification, and to evaluate their effect on system performance. The program is based on formal methods that construct dependency lists (symbolic representations of the components) to demonstrate the claims of fault tolerance. Argonne's automated theorem-proving tools and program transformation system play a vital part in the formal analysis. The issue of parallelism is also of central interest. We are examining the effect of parallelism on performance and maintainability. The project is conducted with Alphatech (algorithm design), the Charles Stark Draper Laboratories (fault-tolerant, parallel-processing computer design), Michigan State University (formal methods), and the University of New Mexico (formal methods).

Another important part of our work with SDIO involves the drafting of standards for numerical libraries for Ada and the implementation of exemplary, conforming libraries and their test programs.
The standardization effort is carried out through participation on the SIGAda Numerics Working Group and the ISO WG9 Numerics Rapporteur Group. The proposed standard for the elementary functions was accepted by WG9 in October 1989, and a package of elementary functions that conform to the proposed standard was released in August 1990. Several industrial institutions including Westinghouse, Boeing, and IBM have requested and received copies of the software. We expect to continue refining this package. The Numerics Working Group also concluded its work on the primitive functions standard; we are writing the Rationale document for it. We also have constructed a highly accurate test program for exp, exponentiation, log, and arbitrary-based log; the program can be used as a validation tool for the elementary functions.

Finally, the MCS Division provides SDI contractors with access to the Advanced Computing Research Facility (ACRF) and consultation in developing parallel programs and executing them on the parallel machines in the ACRF. During the past year, Alphatech Corp. of Burlington, Mass., made extensive use of these machines in its work as subcontractor to TRW. In addition, by invitation of the SDIO, a one-day tutorial on programming in Ada was taught at the 8th National Conference on Ada Technology in Atlanta.

6.3 U.S. Army and Air Force

The U.S. Army and U.S. Air Force funded the 1991 SIAM Workshop on Automatic Differentiation of Algorithms: Theory, Implementation, and Application, January 1991 (see Section 4.1.3). This was the first scientific meeting devoted to this emerging area, and the response of the participants was extremely favorable, with many expressing pleasure at the breadth and quality of the work discussed. Topics included automatic differentiation implementations (precompilers for Fortran programs, integrated symbolic/numerical environments for PC's, and implementations by overloading in C++ and other advanced languages), large-scale applications (weather modeling, oceanography, beam tracing in optics, orbit analysis, and forest growth modeling), and the relationship between fully symbolic computer algebra and more numerical techniques. A review of the workshop has appeared in SIAM News, and the proceedings will be published later this year.

6.4 U.S. Army

The Mathematics and Computer Science Division at Argonne is collaborating with the Army High Performance Computing Research Center at the University of Minnesota to produce algorithms and optimization software for high-performance computers, and to develop a collection of significant optimization problems to serve as test problems for the MINPACK-2 package. During the first three months of this project we have made significant advances in both the development of algorithms and in the collection of optimization problems.

A major accomplishment is the development of a specialized version of the LMDER code for nonlinear least squares problems found in MINPACK-1. The new code contains an improved version of the QRSOLV subroutine to compute the Levenberg-Marquardt parameter. We have also replaced the subroutine that computes the QR factorization of the Jacobian matrix with code provided by Chris Bischof from the LAPACK project. Our results show that the new QRSOLV subroutine obtains excellent speedups even for small problems. For example, on a problem with \( n = 128 \), the speedup for 8 processors is nearly 7 on the Alliant FX/8. Comparisons with the old version of QRSOLV are also favorable; the new subroutine shows an improvement by a factor of more than 5.
on the Alliant FX/8. We have also compared the new LMDER code with the MINPACK-1 version. On a problem obtained from the National Institute of Standards and Technology with $n = 134$ variables and $m = 252$ observations, the new code shows an improvement by a factor of 6 over the MINPACK-1 version.

Another major accomplishment is the development of a Newton method for unconstrained minimization. This subroutine is based on the GQTPAR subroutine of Moré and Sorensen which implements a highly efficient algorithm for the solution of the trust region subproblem. GQTPAR has been in existence for several years and has been widely used, yet it has not been incorporated into a high-quality Newton method. Our preliminary results for this code show that for problems with $n = 512$ on an Alliant FX/8, the improvement is more than a factor of 10 over the MINPACK-1 code. This improvement is due to the exploitation of the vector and parallel architecture of the Alliant FX/8.

We have also completed a preliminary version of the test problem collection. The main requirement for inclusion in this collection is that each problem must come from a real application and be representative of other commonly encountered problems. Problems in the preliminary version of the collection come from such diverse fields as fluid dynamics, medicine, combustion, nondestructive testing, chemical kinetics, lubrication, optimal design, superconductivity, and mathematics.

7 Technology Transfer

The MCS Division has long encouraged interactions between research divisions and industry. Two projects deserve particular note this year.

7.1 Motorola

We are investigating the applicability of abstract programming and the TAMPR program transformation system in a software development project with the Motorola Corporation. In cooperation with F. Weil (Motorola Communications Research), T. Weigert (Research Institute in Symbolic Computation, Linz, Austria), and T. Harmer (The Queen's University of Belfast), we developed high-level specifications for a software module (several subroutines) used in a police radio system produced by Motorola's Communications Sector. We also developed the transformations needed to derive C code from these specifications. The transformations produce code that is as good as or better than the code written by hand. In performing this work, we uncovered programming practices used in hand coding that could easily lead to bugs in the final code. We devised program transformations that ensure that these practices are applied correctly in constructing the code. We prepared a manuscript describing this work and presented it at the Motorola AI Applications Symposium in Phoenix, Arizona, in November.

7.2 U.S. Air Force

The proposed standard for the elementary functions in Ada has led to a great demand from the international Ada community for a portable implementation of such a library. To address this demand, the U.S. Air Force is funding a joint project with Argonne and the Numerical Algorithms Group, Ltd., to develop two software libraries. The first comprises implementations of all twenty
functions specified by the standard; at present, more than half the functions have been completed. The second comprises highly accurate tests for these functions. An important aspect of the project is portability: the two libraries will be designed so that they can be compiled and run correctly without modification on a variety of hardware and software environments.

8 Educational Activities

Education has become an increasingly important responsibility of those involved in scientific computing. In addition to encouraging the transfer of research results to industry, the MCS Division conducts various educational activities designed to train future scientists and teachers of science and mathematics. We anticipate that our efforts in this area will contribute significantly to the new DOE High Performance Computing and Communications Program.

8.1 A-PRIME – Argonne Program in Mathematics Education

A-PRIME (Argonne PRogram In Mathematics Education) is a new project aimed at strengthening teachers' understanding of the concepts of mathematics. The project is being carried out jointly by the MCS Division and Argonne's Division of Educational Programs.

A central feature of the project is a Mathematics Visualization Laboratory, designed to train future and practicing teachers in mathematics. The new Laboratory features NeXT computers running Mathematica, a general-purpose automated mathematical tool that supports numerical computations.

Our initial target for A-PRIME is Chicago State University, which trains many of the future teachers for the Chicago-area elementary and high schools. Working with CSU faculty, we are developing a new curriculum that uses animated sequences of two- and three-dimensional images to illustrate core mathematical concepts. The new curriculum is intended to improve understanding and comprehension of abstract mathematical ideas among student teachers. The teachers in turn will be able to demonstrate mathematical concepts effectively and will impart knowledge and enthusiasm to their students.

8.2 Special Course in Supercomputing Offered to Minority Students

Eighteen college undergraduates from around the country met at Argonne National Laboratory in January 1991 to participate in a week-long course on supercomputing. The course was jointly sponsored by Argonne's Affirmative Action Office, Division of Educational Programs, and the MCS Division and was offered to undergraduate minority students.

Lectures included an introduction to supercomputers, techniques for transforming programs from scalar to vector and parallel, and methods for debugging code. The students vectorize a sample code, implement parallelism, and compare the results with scalar code. This "hands-on" experience was conducted in the MCS Division's classroom/labatory and involved use of our NeXT workstations for interactive work and the Alliant FX/8 multiprocessor for program runs.

We are now exploring the possibility of developing an expanded course that would allow students to continue using Argonne's computing facilities via national networks.
8.3 Visitors Program

The MCS Division supports an extensive visitors program throughout the year. Many university faculty, postdocs, graduate students, and undergraduate students, as well as staff members from other laboratories and industry, find it beneficial to spend periods of time in the division to collaborate with our regular staff. The visitors in turn represent a significant resource to the division's staff, providing a channel to the world outside, communicating new ideas and contributing new challenges to the division's research programs. A complete list of our visitors is given in Appendix F.
A Scientific Staff

C. H. Bischof, Ph.D., Cornell University, 1988
J. M. Boyle, Ph.D., Northwestern University, 1970
G.-Q. Chen, Ph.D. Academia Sinica, 1986
I-L. Chern, Ph.D. Courant Institute of Mathematical Sciences, 1983
W. J. Cody, Jr., M.A., University of Oklahoma, 1956; D.Sc. (Hon.), Elmhurst College, 1977
W. R. Cowell, Ph.D., University of Wisconsin, 1954
K. W. Dritz, M.S., Massachusetts Institute of Technology, 1967
I. T. Foster, Ph.D., Imperial College, University of London, 1988
B. S. Garbow, M.S., University of Chicago, 1952
A. O. Griewank, Ph.D., Australia National University, 1980
W. D. Gropp, Ph.D., Stanford University, 1982
M. T. Jones, Ph.D., Duke University, 1990
H. G. Kaper, Ph.D., Rijksuniversiteit, Groningen, 1965
M. K. Kwong, Ph.D., University of Chicago, 1973
G. K. Leaf, Ph.D., University of Illinois, 1961
E. L. Lusk, Ph.D., University of Maryland, 1970
J. N. Lyness, D. Phil., Oxford University, 1957
W. W. McCune, Ph.D., Northwestern University, 1984
J. J. Moré, Ph.D., University of Maryland, 1970
R. A. Overbeek, Ph.D., Pennsylvania State University, 1971
P. E. Plassmann, Ph.D., Cornell University, 1990
B. F. Smith, Ph.D., Courant Institute of Mathematical Sciences, 1990
P. T. P. Tang, Ph.D., University of California, Berkeley, 1987
S. K. Winker, Ph.D., University of Illinois at Chicago, 1984
L. T. Wos, Ph.D., University of Illinois, 1957
S. J. Wright, Ph.D., University of Queensland, 1984
B Administrative and Technical Support Staff

J. M. Beumer, Editorial Specialist
J. A. Griffin, Executive Secretary
M. W. Henderson, Scientific Assistant
T. M. Huml, Administrative Secretary
H. G. Kaper, Division Director
R. E. King, Assistant Division Director
D. M. Levine, Scientific Associate
C. L. McCabe, Administrative Secretary
G. W. Pieper, Technical Editor Senior
E. A. Rackow, Scientific Assistant
R. L. Stevens, Manager, MCS Computing Facilities
C  Professional Activities

C. H. Bischof  Adjunct Professor, Northern Illinois University
J. M. Boyle  Member, International Federation for Information Processing
              Working Group 2.1 for Algol and Abstracto
W. J. Cody  Member, International Federation for Information Processing
              Working Group 2.5 for Mathematical Software
W. R. Cowell  Member, SIGAda Numerics Working Group
              President, W. R. Cowell Consulting, Inc.
K. W. Dritz  Member, ACM SIGAda Numerics Working Group
              Member, WG9 (Ada) Numerics Rapporteur Group
I. T. Foster  Member, CRPC Technical Steering Committee
              Member, Program Committee NACL 89
B. S. Garbow  Consultant, C. Abaci
A. Griewank  Regional Editor, Optimization Methods and Software
              Chair, SIAM Workshop on the Automatic Differentiation
              of Algorithms
              Chairman, Conference on Bifurcation and Chaos
W. D. Gropp  Member, SIGNUM Board
M. T. Jones  Consultant, ICASE
H. G. Kaper  Corresponding member, Royal Netherlands Academy of Sciences
              Associate Editor, Integral Equations and Operator Theory
              Associate Editor, Transport Theory and Statistical Physics
              Adjunct Professor, Northern Illinois University
              Editor at large, Marcel Dekker, Inc.
              Member, Concurrent Supercomputing Consortium Policy board
              Member, CRPC Executive Committee
              Organizer, 26th Midwest PDE Conference
              Chairman, Workshop on Asymptotic Analysis and
              the Numerical Solution of PDEs
M. K. Kwong  Adjunct Professor, Northern Illinois University
              Organizer, Workshop on Asymptotic Analysis and
              the Numerical Solution of PDEs
E. L. Lusk  Adjunct Professor, Northern Illinois University
              Associate Editor, Journal of Automated Reasoning
J. N. Lyness  Associate Editor, Mathematics of Computation
              Adjunct Professor, Northern Illinois University
W. W. McCune  Member, Program Committee CADE-11
              Secretary, Association for Automated Reasoning
<table>
<thead>
<tr>
<th>Name</th>
<th>Positions and Responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>J. J. Moré</td>
<td>Co-chair, Organizing Committee, SIAM Conference on Optimization</td>
</tr>
<tr>
<td></td>
<td>Member, CRPC Technical Steering Committee</td>
</tr>
<tr>
<td></td>
<td>Associate Editor, SIAM Journal on Optimization</td>
</tr>
<tr>
<td></td>
<td>Associate Editor, Numerische Mathematik</td>
</tr>
<tr>
<td></td>
<td>Member, SIGNUM Board</td>
</tr>
<tr>
<td>R. A. Overbeek</td>
<td>Member, DOE Informatics Task Force (Human Genome)</td>
</tr>
<tr>
<td>G. W. Pieper</td>
<td>Managing Editor, Association for Automated Reasoning</td>
</tr>
<tr>
<td></td>
<td>Instructor, Illinois Benedictine College</td>
</tr>
<tr>
<td></td>
<td>Consultant, University of Tennessee</td>
</tr>
<tr>
<td>E. A. Rackow</td>
<td>Member, Steering Committee, Metro Chicago Sun Local Users Group</td>
</tr>
<tr>
<td></td>
<td>Member, CRPC Facilities Committee</td>
</tr>
<tr>
<td></td>
<td>Argonne Representative, Concurrent Supercomputing Consortium</td>
</tr>
<tr>
<td></td>
<td>Member, Sequent Users Group SIGPP Committee</td>
</tr>
<tr>
<td>P. T. P. Tang</td>
<td>Member, SIGAda Numerics Working Group</td>
</tr>
<tr>
<td>L. T. Wos</td>
<td>Adjunct Professor, University of Illinois at Urbana</td>
</tr>
<tr>
<td></td>
<td>Editor-in-Chief, Journal of Automated Reasoning</td>
</tr>
<tr>
<td></td>
<td>President, Association for Automated Reasoning</td>
</tr>
<tr>
<td>S. J. Wright</td>
<td>Editor, Optimization Methods and Software</td>
</tr>
</tbody>
</table>
D Scientific Impact

The following list reflects articles published, reports distributed, and talks presented by MCS Division members since September 1989. We continue to send out a periodic mailing of abstracts of our publications, through which we hope to keep scientists better informed of the work of the Mathematics and Computer Science Division and to encourage interactions with other research institutions.

Books


Publications in Books and Refereed Journals


M. T. Chu, "A Continuous Jacobi-like Approach to the Simultaneous Reduction of Real Matrices," Linear Algebra and Its Applications 147 (1991) 75-96 (also MCS-P30-1288)


L. Wos, "Meeting the Challenge of Fifty Years of Logic," J. Automated Reasoning 6 (1990) 213–232 (also MCS-P129-0290)


Contributions to Conference Proceedings


Preprints (submitted for publication)

C. H. Bischof and P. C. Hansen, "Structure-Preserving and Rank-Revealing QR-Factorizations," MCS-P100-0989


A. K. Singh and R. Overbeek, "Derivations of Efficient Parallel Programs: An Example from Genetic Sequence Analysis," MCS-P104-0989


R. L. Stevens, "The Interesting Future of Supercomputing," MCS-P110-1189


M. K. Kwong and L. Zhang, "Uniqueness of the Positive Solution of $\Delta u + f(u) = 0$," MCS-P117-1289

N. T. Karonis, "Timing Parallel Programs That Use Message Passing," MCS-P119-1090

D. Solow and R. S. Womersley, "A Piecewise Linear Concave Minimization Algorithm for the Linear Complementarity Problem," MCS-P120-0190


L. M. Ewerbring and F. T. Luk, "The HK Singular Value Decomposition of Rank-Deficient Matrix Triplets," MCS-P125-0190


J. V. Burke, "A Robust Trust Region Method for Constrained Nonlinear Programming Problems," MCS-P131-0190


D. Jacobs, A. Langen, and W. Winsborough, "Multiple Specialization of Logic Programs with Run-Time Tests," MCS-P135-0290

A. Mulkers, W. Winsborough, and M. Bruynooghe, "Analysis of Shared Data Structures for Compile-Time Garbage Collection in Logic Programs," MCS-P137-0290


P. T. P. Tang, “Table-driven Implementation of the Expm1 Function in IEEE Floating-Point Arithmetic,” MCS-P144-0390

H. Zhang, “Large-Time Behavior of the Maximal Solution of Equation $u_t = (m^{-1}u_x)_x$ with $-1 < m \leq 0$,” MCS-P146-0490


G.-Q. Chen, “Global Solutions to the Compressible Navier-Stokes Equations for a Reacting Mixture,” MCS-P155-0590

M. K. Kwong and Y. Li, “Uniqueness of Radial Solutions of Semilinear Elliptic Equations,” MCS-P156-0590

Per Christian Hansen, “Analysis of Discrete Ill-Posed Problems by Means of the L-Curve,” MCS-P157-0690


J. M. Boyle and T. J. Harmer, “A Practical Functional Program for the CRAY X-MP,” MCS-P159-0690


G.-Q. Chen and A. Rustichini, “Global Solutions to a System of Conservation Laws and Nonzero Sum Dynamic Games,” MCS-P161-0790

S. J. Wright, “An Interior-Point Algorithm for Linearly Constrained Optimization,” MCS-P162-0790

D. G. Szyld and M. T. Jones, “Two-Stage and Multi-Splitting Methods for the Parallel Solution of Linear Systems,” MCS-P165-0790


47
S. J. Wright, "Partitioned Dynamic Programming for Optimal Control," MCS-P173-0890
V. Pereyra and S. J. Wright, "Three-Dimensional Inversion of Travel Time Data for Structurally Complex Geology," MCS-P174-0890
S. Winker, "Absorption and Idempotency Criteria for a Problem in Near-Boolean Algebras," MCS-P177-0990
S. J. Wright, "Stable Parallel Algorithms for Two-Point Boundary Value Problems," MCS-P178-0990
W. J. Cody, "Algorithm XXX: SPECFUN—A Portable Package of Special Functions and Test Drivers," MCS-P179-0990
M. T. Jones and M. L. Patrick, "The Lanczos Algorithm for the Generalized Symmetric Eigenproblem on Shared-Memory Architectures," MCS-P182-1090
B. F. Smith, "A Domain Decomposition Algorithm for Elliptic Problems in Three Dimensions," MCS-P185-1090
Y. P. Hong and C.-T. Pan, "Rank-revealing QR Factorization and SVD," MCS-P188-1090
W. J. Cody, "The Normal Integral," MCS-P189-1090
W. McCune, "Experiments with Discrimination Tree Indexing and Path Indexing for Term Retrieval," invited paper, MCS-P191-1190
C. H. Bischof and N. J. Kelly, "XBibTeX and Friends," MCS-P193-1190
P. T. P. Tang, "Table-Lookup Algorithms for Elementary Functions and Their Error Analysis," MCS-P194-1190


M. T. Jones and P. E. Plassmann "Parallel Iterative Solution of Sparse Systems Using Orderings From Graph Coloring Heuristics," MCS-P198-1290

L. Wos and W. McCune, "The Application of Automated Reasoning to Questions in Mathematics and Logic," MCS-P199-1290


C. Bischof, A. Griewank, and D. Juedes, "Exploiting Parallelism in Automatic Differentiation," MCS-P204-0191


M. T. Jones and P. E. Plassmann "An Improved Incomplete Cholesky Factorization," MCS-P206-0191

R. Overbeek and I. Foster, "Aligning Multiple RNA Sequences," MCS-P207-0191

J. N. Lyness, "Extrapolation-based Boundary Element Quadrature," MCS-P208-0191


Reports

R. Butler and R. Overbeek, A Tutorial on the Construction of High-Performance Resolution/Paramodulation Systems, ANL-90/30 (September 1990)


W. J. Cody, CELEFUNT: A Portable Test Package for Complex Elementary Functions, ANL-91/1 (January 1991)


W. W. McCune, OTTER 2.0 Users Guide, ANL-90/9 (March 1990)


**Technical Memoranda**

W. J. Cody, "ELEFUNT Test Results under AST Fortran V1.8.0 on the Sequent Symmetry," MCS-TM-138 (July 1990)


**Oral Presentations**

C. Bischof, seminar, North Carolina State University, Raleigh, North Carolina, April 16-18, 1990


C. Bischof, "Orthogonal Factorization of Rank-Deficient Matrices on High-Performance Architectures," Supercomputing Research Center, Greenbelt, Maryland, May 22, 1990

C. Bischof, "Orthogonal Factorizations of Rank-Deficient Matrices on High-Performance Architectures," Dept. of Computer Science, Yale University, New Haven, Connecticut, May 24, 1990


C. Bischof, "Computing Rank-Revealing Triangular Factorizations on Parallel Machines," Department of Computer Science, Technical University, Lyngby, June 14, 1990


C. Bischof, "QR Factorizations of Rank-Deficient Matrices on High-Performance Architectures" (plenary talk), Householder Symposium XI on Numerical Linear Algebra, Tylosand, Halmstad, Sweden, June 18, 1990

C. Bischof, "Fundamental Linear Algebra Routines on High-Performance Architectures" (invited talk), Supercomputer’90 Workshop, Mannheim, Germany, June 22, 1990

C. Bischof, "Incremental Condition Estimation and Applications," Dept. of Applied Math, University of Bielefeld, Germany, June 25, 1990

C. Bischof, "LAPACK—Now and Beyond," KFA Juelich, Germany, June 26, 1990


C. Bischof, "Orthogonal Factorizations for Parallel and Vector Architectures," Department of Applied Mathematics, University of Heidelberg, Germany, Sept. 18, 1990

C. Bischof, "Linear Algebra Algorithms for Parallel Architectures," Department of Applied Mathematics, University of Frankfurt, Germany, Sept. 21, 1990

C. Bischof, "Efficient Orthogonal Factorizations of Singular Matrices," IBM Research Center in Heidelberg, Germany, Sept. 24, 1990

C. Bischof, "Updating Cholesky Factors on SIMD Architectures," Parallel Circus, Toronto, Canada, October 25, 1990


51


K. W. Dritz, “Parallel Programming in Ada,” full-day tutorial at Eighth Annual National Conference on Ada Technology, Atlanta, Georgia, March 5, 1990


I. Foster, “Developing Parallel Applications in Strand” and “Strand: A Practical Parallel Programming Tool,” tutorial at North American Conference on Logic Programming (NACL), October 1989

I. Foster, “Parallel Programming in Strand,” tutorial at Argonne, November 1989


I. Foster, “The PCN Run-Time System,” CRPC research symposium, Rice University, August 28, 1990

I. Foster, “Towards a Tera-op Climate Model,” Caltech, December 5, 1990

I. Foster, “A Template-Based Approach to Parallel Programming,” University of Texas at Austin, February 4, 1991


A. Griewank, "Sequential Evaluation of Adjoints and Higher Derivatives by Overloading and Reverse Accumulation," Iowa State University, July 21, 1990


W. Gropp, Copper Mountain Conf. on Iterative Methods, April 3-5, 1990


M. K. Kwong, "Uniqueness Results for a Nonlinear Boundary Value Problem," Northern Illinois University, October 17, 1989


M. K. Kwong, "Recent Results on the Uniqueness of Semilinear Elliptic Equations," University of Leiden, Leiden, the Netherlands, June 26-July 27, 1990


D. Levine, "Parallel and Supercomputer Architectures," lecture series in advanced computing, Division of Educational Programs, Argonne, September 1990

E. Lusk, "Programming Parallel Computers," Computer Science Department, Michigan State University, East Lansing, Michigan, February 6, 1990


E. Lusk (with W. McCune), "Tutorial on Automated Reasoning Systems," 10th Conference on Automated Deduction, Kaiserslautern, Germany, July 1, 1990


E. Lusk, "Visualizing Parallel Execution," Argonne Workshop on Performance Visualization, Argonne, August 3, 1990


E. Lusk, "Portable Parallel Computing and Applications," Technical University of Munich, November 17, 1990

E. Lusk, "High-Performance Computing Applications," Australian National University, December 15, 1990

J. Lyness, "An Introduction to Lattice Rules," University of Hobart, March 14, 1990; University of New Zealand at Dunedin, April 10, 1990; University of New Zealand at Wellington, April 11, 1990; and University of New Zealand at Auckland, April 20, 1990

J. Lyness, "Extrapolation-based Boundary Element Quadrature," Conf. on Numerical Methods in Applied Science and Industry, Turin, Italy, June 17-21, 1990; also at Stuttgart, Germany, June 28, 1990; and University of Darmstadt, Germany, June 29, 1990
W. McCune, "Automated Theorem Proving and Applications and Argonne National Laboratory," series of lectures, University of Catania, Catania, Italy, December 3-9, 1989

W. McCune, International Symposium on Artificial Intelligence and Mathematics, Fort Lauderdale, Florida, January 3-5, 1990

W. McCune, seminar, "Theorem Proving with OTTER," University of Texas at Austin, Austin, Texas, April 3, 1990


J. Moré, "The MINPACK-2 Project," CSC Virginia Technology Center, Falls Church, Virginia, August 16, 1990


R. Overbeek, "Gel Interpretation," Conf. of the American Chemical Society, Miami Beach, Florida, September 13, 1989

R. Overbeek, "Theorem Proving," ICOT, Tokyo, Japan, March 1, 1990

R. Overbeek, 3rd Workshop on Programming Languages and Compilers for Parallel Computing, University of California at Irvine, Irvine, Calif., August 1-3, 1990

G. W. Pieper (with C. Caruthers), "Writing at Work," seminar, Division of Educational Programs, Argonne National Laboratory, September 20, 1989

G. W. Pieper (with C. Caruthers), "Working at Writing," seminar, Division of Educational Programs, Argonne National Laboratory, February 27, 1990

G. W. Pieper, "Technical Writing Workshop," Division of Educational Programs, Argonne National Laboratory, February 21, 1991


R. Stevens, CRPC Annual Meeting, Los Angeles, Calif., October 16-17, 1989

R. Stevens, Demonstration of OTTER, ICOT, Tokyo, Japan, March 1, 1990
P. T. P. Tang, seminar, IBM Toronto, Toronto, Canada, October 2-3, 1990
P. T. P. Tang, seminar, Northern Illinois University, DeKalb, Illinois, October 12, 1990


S. Wright, “Interior-Point Algorithms for Linearly Constrained Optimization,” University of Queensland, Australia, June 1990


S. Wright, “Stable Parallel Algorithms for Two-Point Boundary Value ODEs,” Parallel Circus, Toronto, Canada, October 1990


E Meetings, Workshops, and Classes

In addition to the activities supported by NSF and SDI funding (see Section 6), the MCS Division organized and hosted numerous other meetings, workshops, and classes.

1. Classes on Parallel Computing

Argonne conducted 12 classes on parallel computing from September 1989 through February 1991. The attendees, averaging 20-25 persons per class, represented universities, industry, and various research laboratories throughout the country.

2. Midwest PDE Conference

The MCS Division hosted the 26th Midwest Partial Differential Equations (PDEs) Meeting at Argonne on November 4-5, 1989. The purpose was to communicate recent research results in PDEs, both through invited lectures and informal discussion. The meeting was attended by approximately 60 mathematicians, mostly university faculty and postdoctoral researchers. Several staff members in our division also attended; and G.-Q. Chen, who is the recipient of the first Argonne-University of Chicago Fellowship in Mathematics, presented a lecture on “Hyperbolic systems of conservation laws with degeneracy.” Of particular note is the fact that this was the first time the semi-annual event was hosted by a nonacademic institution.

3. First Affiliates Workshop

On March 22-23, 1990, the ACRF sponsored the first Affiliates Workshop. The two-day workshop was designed to give our industrial and academic affiliates an opportunity to share the results of their research using the ACRF computers and to present information about recent advances in high-performance computing. Ewing Lusk, ACRF scientific director, opened the workshop with a brief history of the ACRF. Rick Stevens, manager of the MCS Division computing facilities, followed with a more detailed review of the ACRF computing facilities. The remainder of the two-day workshop was devoted principally to presentations by our affiliates, with topics including a new framework for studying performance on shared-memory multiprocessors, the MasPar Programming Environment, the University of Illinois at Chicago Workshop Program on Scientific Supercomputing, collaborative projects with Northern Illinois University and Argonne, sensor fusion, the MaGIC program at the Australian National University, tools for programming analysis in a multiprocessing environment, tests of the memory bandwidth of the Butterfly switch, parallel algorithms for discrete event simulation on the Connection Machine, computational treatment of stochastic optimal control applications in continuous time, optical technology, object-based computing on the NeXT, and lattice gauge theory applied to quantum chromodynamics and quantum electrodynamics problems on the CM-2. Lusk concluded the workshop with an open discussion about how the ACRF can help its users.

4. Performance Visualization Workshop

The MCS Division hosted a three-day parallel program performance visualization workshop on July 18-20, 1990. The workshop was held in conjunction with Argonne's participation in the CHAMMP project. CHAMMP (Computer Hardware, Advanced Mathematics and Model Physics) is a DOE initiative to develop an advanced climate model that increases the capability of climate
models by a factor of 10,000 or more within ten years. The July workshop focused on performance visualization tools for collecting, presenting, and interpreting performance data on massively parallel machines (having from 1,000 to 10,000 processors). Of particular importance was the issue of scalability of the display technology and data collection schemes. Participants included researchers from Aerospace Corporation, Argonne National Laboratory, Iowa State University, Oak Ridge National Laboratory, Sandia National Laboratory, Tufts University, and the University of Illinois.

5. Theory Institute in Automated Reasoning

Nineteen of the world's best researchers in automated reasoning, logic, and mathematics met at Argonne on August 6-10, 1990, to address various obstacles confronting the automation of reasoning. The occasion was the Theory Institute in Automated Reasoning, organized by the Mathematics and Computer Science Division and supported by a special grant from Argonne's Physical Research Administration program. The researchers focused on the interrelationship of representation, inference rule, and strategy, each of which materially affects the performance of an automated reasoning program. Several challenging questions in mathematics and logic were answered, and many of the participants have begun using our automated reasoning program OTTER and our database of axioms, lemmas, and test problems to experiment with new ideas.

6. Workshop on Numerical Methods for PDEs on the Sphere

Representatives of the research groups involved in DOE's CHAMMP Modeling Program (Argonne, the National Center for Atmospheric Research, Los Alamos National Laboratory, and NASA) met at Argonne on August 13-15, 1990, to discuss the various equations relevant for global climate investigations. Topics included the impact of parallel computing on algorithms for PDEs on the sphere, development of models, the spectral transform method, semi-Lagrangian techniques, and parallel schemes for the shallow-water equations.

7. Workshop on the Intel Touchstone DELTA System

Argonne recently joined several leading U.S. research institutions to form a consortium that will have access to an Intel Touchstone DELTA system beginning in early spring 1991. To enable Argonne researchers to get an early start on preparing calculations for the new machine, the Mathematics and Computer Science Division conducted an intensive workshop on the new system. The workshop focused on the innovative architecture of the Touchstone system, parallel programming techniques for transporting code to the Touchstone, a procedure for predicting how well a code will scale from one processor to the multiprocessor DELTA machine, and methods for writing efficient code in parallel programming notation and in assembly language. Workshop participants were also given laboratory time to test examples on an Intel Gamma iPSC/860, which Argonne has installed in the Advanced Computing Research Facility as a staging platform for the DELTA. Access to the DELTA will enable Argonne scientists to tackle complex computations—including the Grand Challenges—that require enormous computational resources. Proposed projects include modeling contaminant transport phenomena for hazardous waste remediation problems and extending climate models to produce realistic regional-scale predictions.
F  Special Term Appointments and Other Visitors

Special Term Appointments

Brett Averick, University of Minnesota  
Alvin Bayliss, Northwestern University  
Richard Carter, University of Minnesota  
Toni Kazic, Washington University  
B. J. Matkowsky, Northwestern University  
Brian Smith, University of New Mexico  
Robert Veroff, University of New Mexico  
Anthony Wojcik, Michigan State University

Faculty Research Leave at Argonne

Lorenz Biegler, Carnegie Mellon University  
Peter Takac, Vanderbilt University

NSF Minority Internship in Parallel Computing

Amos Carpenter, Butler University

Other Visitors

K. A. Ariyawansa, Washington State University  
Frederick Atkinson, University of Toronto  
Sabash Balakrishna, Carnegie Mellon University  
Hamid Bellout, Northern Illinois University  
Karin Bennett, University of Kentucky  
Iauw-Bhieng, Carnegie Mellon University  
Oleg Bourdakov, USSR Academy of Sciences  
Massimo Bruschi, University of Milan  
Ralph Butler, University of North Florida  
Tracye Butler  
Thiery Colin, Ecole Normale Superieure de Lyon  
Clinton Dawson, University of Chicago  
LeRoy Drummond, University of Tulsa  
Michelle Evard, Andrews University  
Remy Evard, Andrews University  
Charles Fulton, Florida State University  
Marc Garbey, Ecole Normale Superieure de Lyon  
Ahmad Ghafarian, University of Chicago  
Per Christian Hansen, Technical University of Denmark  
Floyd Hanson, University of Illinois, Chicago  
Terence Harmer, The Queen's University of Belfast  
Graham Hodgson, NAG, Ltd.
Xiao Huang, Pennsylvania State University
Nicholas Karonis, Syracuse University
Carl Kesselman, Aerospace Corporation
James Kohl, Iowa University
Marco Lapegra, University of Napoli
Howard Levine
Frank McCabe, Imperial College
Vladimir Mazourik, USSR Academy of Sciences
Shyam Mudambi, Bradeis University
Dale Myers, University of Hawaii
I. M. Navon, Florida State University
Jorge Nocedal, Northwestern University
Robert O’Malley, Rensselaer Polytechnic Institute
Ching-Tsuan Pan, Northern Illinois University
Daniel Pierce, Boeing Computer Services
Morgan Price, University of Chicago
Avijit Purkayatha, Northern Illinois University
David Race, University of Surrey
Marcela Rosemblun, Rice University
Gautam Shroff, Rensselaer Polytechnic Institute
Tor Sorevik, University of Bergen
Joseph Stephen, Northern Illinois University
Deborah Stevens, Northwestern University
Mark Stickel, SRI
Peter Takac, Vanderbilt University
Ronald Taylor, Case Western Reserve University
Chris Thompson, IBM Scientific Centre
Rong dong Wang, Northern Illinois University
Shou-hong Wang, Indiana University
Thomas Weigert, Johannes Kepler University and Motorola
Liquan Zhang, University of Minnesota

Pre-College

Amy Crook, Illinois Math and Science Academy
Eugene Foss, Illinois Math and Science Academy
David Joerg, Illinois Math and Science Academy
Ben Sussman, Oak Park High School

Science and Engineering Research

Simon DeVriendt
Eric J. Fraser
Theodore Gaunt
Steven J. Hammond
Virginia Herrarte
James C. Hu
David Juedes  
Nickolas Kelly  
Kathy K. Kong  
James L. Mosley  
J. Robert Neely  
Peter J. Nigro  
Shawn J. Reese  
Michael J. Wassmer  

Summer Student Research Participation Program  

James Bordner  
James Garnett  
Tamara Henry  
Cheryl Hile  
Jeffrey Jackson  
Brad Karp  
Alkes Price  
Jay Srinivasan
G  Seminars

During the period September 1989 through February 1991, the Mathematics and Computer Science Division sponsored numerous seminars in mathematics and computer science. Many of these were part of a continuing series of seminars on high-performance computing.

  Eric F. Van de Velde, Concurrent Continuation with Adaptive Data Distribution, Caltech, September 14, 1989
  Alberto Pettorossi, Decidability Results and Characterization of Strategies for the Development of Logic Programs, University of Rome, September 18, 1989
  Juan E. Santos, Wave Propagation in Fluid-Saturated Porous Media, Yacimientos Petroliferos Fiscales S.E., Buenos Aires, September 21, 1989
  L. Magnus Ewerbring, A New Matrix Decomposition Algorithm, Argonne National Laboratory, September 28, 1989
  Carl Hauser, The Xerox Portable Common Runtime System, Xerox Palo Alto Research Center, October 16, 1989
  Renato DeLeone, Serial and Parallel Solution of Large Linear Complementarity Problems and Linear Problems, University of Wisconsin, October 24, 1989
  Michael Renardy, Ill-Posedness at the Boundary for Sliding Contact Problems, Virginia Polytechnic Institute, October 26, 1989
  Mark T. Jones, Parallel Solution of a Generalized Eigenvalue Problem, Duke University, October 27, 1989
  Christopher Thompson, Fully Adaptive Multigrid Methods for the Incompressible Navier Stokes Equations, IBM Bergen Scientific Centre, October 30, 1989
  Alastair Wood, On Eigenvalues with Exponentially Small Imaginary Parts and Stokes' Lines, Dublin City University, Ireland, October 31, 1989
  Gene Golub, Orthogonal Polynomials and Least Squares, Stanford University, November 3, 1989
  Edmund Christiansen, Using the LP Primal Affine Scaling Algorithm on a CRAY X-MP/48, University of Iowa, November 16, 1989
  Frederick Atkinson, Asymptotics of the Prescribed Mean Curvature Equation, University of Toronto, November 28, 1989
  Rudiger Seydel, Numerical Procedures for Continuation in the Presence of Singularities, University of Pittsburgh and Universität Würzburg, December 7, 1989
  Jesse Barlow, Solution of Linear Least Squares Problems: Constrained and Unconstrained, The Pennsylvania State University, December 8, 1989

Elaine Oran, *Chemical-Fluid Instabilities in Reactive Flows*, Naval Research Laboratory, December 21, 1989

Ken Kunen, *Partial Negation-as-Failure*, University of Wisconsin, January 8, 1990

A. G. Ramm, *Exact Inversion of the Scattering Data in Three-Dimensional Problems*, Kansas State University, January 11, 1990

Charles Amick, *Uniqueness Results for the Benjamin-Ono Equation*, University of Chicago, February 6, 1990

Gary Lindstrom, *Open Languages: Types as Semantic Interfaces*, University of Utah, February 14, 1990


Susan T. Dumais, *Using Singular Value Decomposition to Improve Information Retrieval*, Bell Communications, March 5, 1990

Yi Li, *Concentration Compactness Principle and Its Application in Scaler Field Equation*, University of Chicago, March 6, 1990


Ed DeLuca, *Turbulent Thermal Convection: The Physics, the Numerics, and the Movie*, University of Chicago, March 29, 1990

Germund Dahlquist, *Optimal One-Leg Methods for ODEs*, Royal Institute of Technology, Sweden, April 9, 1990

Paul E. Sacks, *Reconstruction Techniques for Classical Inverse Sturm-Liouville Problems*, Iowa State University, April 12, 1990


John Toland, *Dynamics of Indefinite Hamiltonian Systems*, University of Bath, U.K., April 17, 1990

Robert Donnelly, *Molecular Structure Determination by a New Global Optimization Technique*, Auburn University, April 26, 1990


Alex Pothen, *Ordering Algorithms for Parallel Sparse Matrix Factorizations*, University of Wisconsin, May 18, 1990

Hsiao Ling, *Evolutionary Equations with Dissipation*, Courant Institute of Mathematical Sciences, June 18, 1990


H. M. Glaz, *An Implicit-Explicit Godunov Scheme and Preliminary Results for Compressible Flow*, University of Maryland, July 6, 1990


L. M. Delves, *The ESPRIT Parallel Numerical Library Projects*, University of Liverpool, July 17, 1990


Liquan Zhang, *Some Results on Uniqueness of Positive Solutions of Semilinear Elliptic Equations*, University of Minnesota, September 15, 1990

John W. Dawson, Jr., *The American Career of Kurt Goedel*, The Pennsylvania State University, September 21, 1990 (sponsored jointly with the Physics Division)

M. J. D. Powell, *Interpolation to Functions of One Variable by Multiquadrics*, University of Cambridge, September 27, 1990

Shuo-hong Wang, *Attractors to the 2D Model of Large-Scale Atmospheric Dynamics*, Indiana University, October 4, 1990

Howard Levine, *A Fujita Type Global Existence—Global Nonexistence Theorem for a Weakly Coupled System of Reaction-Diffusion Equations*, Iowa State University, October 18, 1990


Ralph Carlson, *Recent Developments in Scattered Data Interpolation*, Lawrence Livermore National Laboratory, November 8, 1990

Philip Thomas Keenan, *Thermal Simulation of Pipeline Flow*, University of Chicago, November 29, 1990


James Burke, *Stable Perturbation of Nonsymmetric Matrices*, University of Washington, December 20, 1990

Malcolm Stocks, *Parallelization of Electronic Structure Algorithms*, Oak Ridge National Laboratory, January 24, 1991 (sponsored jointly with the Materials Science Division)


High-Performance Computing Seminars

Lennert Johnsson, *A Data Parallel Supercomputer*, Yale University and Thinking Machines, October 3, 1989


Dan Pierce and John Lewis, *Multifrontal Householder QR Factorization*, Boeing Computer Services, November 30, 1989


Lars Elden, *Iterative Solution of Non-Symmetric Systems Arising in Hyperbolic PDEs*, Linkoping University, Sweden, May 4, 1990

Dan C. Marinescu, *Communication and Control in SPMD Parallel Numerical Computations—The E/T Model*, Purdue University, May 10, 1990


Harry Scott Berryman, *Sparse Iterative Solvers on Scalable Parallel Computers*, NASA Langley Research Center and Yale University, July 26, 1990

M. W. Berry, *Multiprocessor Sparse SVD Algorithms and Applications*, University of Alabama, December 6, 1990


Amos Carpenter, *Numerical Results on Best Uniform Polynomial and Rational Approximations of |x| on [-1, +1]*, Argonne and Butler University, February 7, 1991