RESEARCH IN MATHEMATICS AND COMPUTER SCIENCE AT ARGONNE:

January 1988 - August 1989

Gail W. Pieper

Mathematics and Computer Science Division

August 1989

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.
A major purpose of the Technical Information Center is to provide the broadest dissemination possible of information contained in DOE’s Research and Development Reports to business, industry, the academic community, and federal, state and local governments.

Although a small portion of this report is not reproducible, it is being made available to expedite the availability of information on the research discussed herein.
## Contents

1. MCS Division Highlights—1988-1989 .......................... 1

2. MCS Scientific Staff
   2.1 Permanent Staff ........................................ 3
   2.2 Wilkinson Fellow ....................................... 5
   2.3 Joint Appointments .................................. 5

3. User Facilities ............................................. 7

4. Analytical and Numerical Methods .................................. 9
   4.1 Nonlinear Differential Equations (H. Kaper, M. K. Kwong) .......... 9
   4.2 Spectral Theory and Boundary Value Problems (H. Kaper, G. Leaf) .... 10
   4.3 Inequalities (M. K. Kwong) .................................. 10
   4.4 Delay Equations (M. K. Kwong) .................................. 12
   4.5 Symbolic Computational Methods in Differential Equations (M. K. Kwong, G. K. Leaf) .................................. 13
   4.6 Optimization (A. Griewank, J. Moré) .................................. 14
   4.7 Bifurcation Phenomena in Combustion (H. Kaper, G. Leaf, M. Minkoff, A. Bayliss, B. Matkowsky) .................................. 16
   4.8 Computational Fluid Dynamics (G. Leaf) .................................. 17
   4.9 Numerical Methods for Partial Differential Equations (G. Leaf, M. Minkoff) .... 18
   4.10 Quadrature (J. Lyness) ........................................ 20
   4.11 Special Functions (W. J. Cody, P. T. P. Tang) ..................... 21

5. Advanced Computing Concepts ..................................... 22
   5.1 Numerical Linear Algebra (C. Bischof, J. Dongarra, D. Sorensen) .... 22
   5.2 Automated Reasoning (L. Was, W. McCune) .......................... 23
   5.3 Logic Programming (E. Lusk, R. Overbeek, R. Stevens, I. Foster, S. Winker) ........ 24
   5.4 Abstract Programming (J. Boyle, K. Dritz, T. Harmer) .......... 26
   5.5 Automated Program Transformation (J. Boyle, K. Dritz, T. Harmer) .... 28
   5.6 Automated Program Verification (K. Dritz) .......................... 28
   5.7 Parallel Programming Environments (J. Dongarra, D. Sorensen, E. Lusk, R. Overbeek, C. Bischof) .................................. 29
   5.8 Software Tools for Analyzing and Transforming Programs (W. Cowell) .... 30
6 Special Activities Supported by Non-DOE Funding

6.1 NSF Funding (C. Bischof, J. Dongarra, D. Sorensen, S. Wright) ................ 31
   6.1.1 Internship ........................................... 31
   6.1.2 LAPACK ................................................. 32

6.2 SDI Funding (J. Boyle, W. J. Cody, K. Dritz, P. T. P. Tung, T. Harmer) .......... 32
   6.2.1 Portability, Libraries, and Ada ................................... 32
   6.2.2 Parallel Algorithm Studies ..................................... 34
   6.2.3 Code Conversion and Development .............................. 34

7 Technology Transfer ........................................... 35

7.1 Technology Transfer with Industry .................................... 35
   7.1.1 Quintus Computer Systems, Inc. .................................. 35
   7.1.2 NAG .................................................... 35

7.2 Technology Transfer with Other Argonne Divisions ......................... 35
   7.2.1 Experimental Breeder Reactor-II Division ...................... 35
   7.2.2 Computing and Telecommunications Division ..................... 36

A Permanent Scientific Staff ..................................... 37

B Administrative and Technical Support Staff ................................ 38

C Professional Activities ......................................... 39

D Scientific Impact .................................................. 41

E Meetings, Workshops, and Classes ..................................... 50

F Special Term Appointments and Other Visitors .......................... 61

G Seminars ......................................................... 64
MATHEMATICS AND COMPUTER SCIENCE RESEARCH AT ARGONNE

January 1988 - August 1989

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 January 1988 - August 1989. The body of the report gives a brief look at the MCS staff and the research facilities, and discusses various projects carried out in two major areas of research: analytical and numerical methods and advanced computing concepts. Projects funded by non-DOE sources are also discussed, and new technology transfer activities are described. Further information on division staff, visitors, workshops, and seminars is found in the appendices.

1 MCS Division Highlights—1988-1989

Seven years ago, Argonne National Laboratory formed a new division— the Mathematics and Computer Science Division—to conduct basic research in mathematics and computer science. Today, the MCS Division enjoys international recognition for its work in such areas as automated reasoning, advanced computing, logic programming, and computational mathematics.

In this 1988-89 annual report, we highlight the activities of the scientific staff that have made this year a success for the Mathematics and Computer Science Division.

Five of our scientific staff—Jack Dongarra, Ewing Lusk, Jorge Moré, Ross Overbeek, and Dan Sorensen—were promoted to Senior Scientist. The promotions are particularly significant in that this is the first time Argonne has awarded the honor to a computer scientist.

Two other awards certainly deserve mention. Larry Wos received a University of Chicago Award for Distinguished Performance at Argonne in recognition of his scientific accomplishments and leadership role in the field of automated reasoning. Jim Cody received an Outstanding Contribution Certificate from the IEEE Computer Society for his work on the IEEE floating-point standards.

The year also saw major developments in our advanced computing research efforts. First, together with Rice University, the California Institute of Technology, and Los Alamos National Laboratory, we were awarded a grant from the National Science Foundation to establish a Center for Research in Parallel Computation. The CRPC will bring together computer scientists, mathematicians, and physical scientists to foster a science of parallel programming. Second, to ensure that we remain in the forefront of advanced computing research, we added four new machines to our Advanced Computing Research Facility: a massively parallel Connection Machine, an Ardent Titan graphics supercomputer, a BBN TC2000 (Butterfly II), and a BBN GP1000 (Butterfly I) sited at Michigan State University.
We are particularly proud to have been chosen to host the Ninth Annual Conference on Automated Deduction (CADE-9). Held on May 23-26, 1988, the conference commemorated the twenty-fifth anniversary of the discovery of the resolution principle at Argonne in 1963. More than 200 scientists attended the conference, which served as a forum for reporting on research in automated deduction and related issues—theorem proving, logic programming, and program verification, to name a few. Ewing Lusk and Ross Overbeek were program chairmen and edited the conference proceedings of 52 papers, published by Springer-Verlag as volume 310 in the series Lecture Notes in Computer Science.

Other 1988-89 high points included the development of an extremely accurate and portable method for testing the elementary functions, demonstration of the effectiveness of symbolic programming languages in mathematical analysis, and initiation of a study of the use of advanced computational technologies for biological sequencing.

In the remaining sections of this report, we highlight these and other activities in the Mathematics and Computer Science Division from January 1988 through August 1989. First we give a brief look at our scientific staff and at our research facilities. Then we review specific projects in two broad areas: analytical and numerical methods and advanced computing concepts. Included also is 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.
2 MCS Scientific Staff

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 the first Wilkinson Fellow 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 three new members: Christian Bischof, Man Kam Kwong, and Peter Tang. We also congratulate five staff members—Jack Dongarra, Ewing Lusk, Jorge Moré, Ross Overbeek, and Dan Sorensen—on their promotion to Senior status. This marks a singular occasion: the first time Argonne has promoted a staff member to Senior Computer Scientist.

Christian Bischof is our newest member; he initially joined our division as the first Wilkinson Fellow in September 1988. He has played an active role in the development of portable algorithms for the LAPACK project and in the development of a new blocking methodology for partitioning problems to achieve maximum performance. He is currently working on programming methodologies for massively parallel machines.

James M. Boyle has been active for several years in the area of program transformation. His efforts led to the development of TAMPR, which was used to convert LINPACK programs from single precision to double precision and from complex to real. His current interests include abstract programming and development of a parallel version of TAMPR.

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 is currently preparing another collection of portable special function routines and test programs, and is working on software support for the floating-point standards.

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.

Jack J. Dongarra* is principally interested in devising linear algebra algorithms and developing portable algorithms for high-performance computers. He has been involved in the LINPACK and EISPACK projects, is codesigner of the Level 2 and Level 3 BLAS, and is working on a linear algebra package designed to supersede LINPACK and EISPACK.

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 developed a technique for revealing speedup losses in parallel programs, and he has been working on an automated program verification capability for floating-point programs.

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, and
collaboration in the preparation of software for the inverse Laplace transform and Sturm–Liouville eigenvalue problems.

Andreas Griewank brings a strong background in computational mathematics and applied analysis. His research focuses on bifurcation phenomena, automatic differentiation, and optimization.

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

Man Kam Kwong joined the MCS Division after a long but informal association with our applied analysis group. 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.

Ewing L. Lusk was one of the implementors of the reasoning programs LMA and ITP and has been actively involved in automated reasoning since he joined the Division several years ago. He is one of the principal developers of a parallel programming approach that implements synchronization through monitors written as macros. He is currently part of the international Gigalips project, developing a system for parallel logic programming.

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 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 logic.

Michael Minkoff is one of the chief collaborators in the design of DISPL, and continues to investigate new techniques for solving multidimensional PDEs. He is also a principal investigator of problems involving bifurcation phenomena in combustion.

Jorge J. Moré 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, and is heading the BioLogic Project.

Danny C. Sorensen* has been concerned with the development of parallel algorithms for numerical linear algebra. Emphasis has been on restructuring algorithms for dense linear systems and on the development of divide-and-conquer algorithms for the symmetric eigenvalue problem and singular value decomposition. Toward this goal he and his colleagues have designed new algorithms and restructured existing ones for parallel computers. He is also a principal designer of the SCHEDULE facility for developing and analyzing parallel programs.
Ping Tak Peter Tang joined our staff as a permanent member after a year of working in our
division supported by special funding from the SDIO Systems/Phase 1 BM/C3 Technology Office.
His efforts have focused on the development of test programs for the elementary functions and the
design of a new fast algorithm for linear complex Chebyshev approximation.

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.

*Left the Laboratory during the reporting period.

2.2 Wilkinson Fellow

Christian Bischof joined the MCS Division in August 1988 as the first Wilkinson Fellow in Compu-
tational Mathematics. The fellowship was established in honor of the late James Wilkinson
and is awarded to a young scientist actively engaged in numerical linear algebra and parallel com-
puting research. Bischof’s work focused on the solution of rank-deficient least-squares problems
on high-performance architectures. Such problems arise, for example, in computer-aided design,
structural analysis, and geodesy. Bischof developed a new technique, called incremental condition
estimation, which enabled him to design an algorithm for distributed-memory machines that avoids
the synchronization overhead of the traditional approach. A preliminary implementation of the
algorithm runs up to twice as fast as the traditional method on the CRAY X-MP, CRAY Y-MP,
and CRAY-2.

2.3 Joint Appointments

Will Winsborough held a joint appointment from September 1988 through August 1989 with MCS
and the Computer Science Department of the University of Chicago. During this time he worked
on three projects. The first involved development of a technique for creating multiple versions of
Prolog procedures, each specialized for a different use, with a method for compile-time selection that
incurs minimal run-time overhead. The second project involved adapting, from Prolog to Strand,
techniques used to characterize terms to which variables can be bound during program execution.
The final project involved verifying the algorithms introduced by Bruynooghe and Jannsens for
the analysis of Prolog.

Marc Garbey holds a joint appointment with the Engineering Sciences and Applied Mathe-
matics Department of Northwestern University. His research in partial differential equations focuses on
three areas: the use of singular perturbation theory and bifurcation analysis, the use of asymptotic
analysis in a numerical computation, and the use of symbolic manipulation languages for asym-
ptotic analysis. During the summer of 1989, Garbey worked on singular perturbation governed by
conservation laws and improved a domain decomposition computation of such a problem based on
asymptotic analysis. He also obtained promising results using asymptotic analysis on a combustion problem involving self-propagating high-temperature synthesis. Currently, he is investigating the accuracy of domain decomposition based on an adaptive pseudospectral method in each subdomain for a nonlinear reaction-diffusion system; the objective is to design a parallel algorithm for the Alliant or Butterfly. He is also investigating the computation of conservation laws on SIMD machines.

Figure 1: Advanced Computing Research Facility
3 User Facilities

To conduct the experimentation vital to much of our mathematics and computer science research, the Division operates the Advanced Computing Research Facility (ACRF).

Currently, the ACRF includes eleven commercial multiprocessing computers:

- A Connection Machine - Model 2, with 16,384 nodes, 8 kilobytes of memory at each node. The CM-2 is capable of a performance in the gigaFLOPS range.
- An Active Memory Technology DAP 510/8 with 1024 one-bit processors and a total of 8 megabytes of memory.
- A BBN TC2000 (Butterfly II) with 62 processors and a total of 128 megabytes of memory. The Butterfly has a performance in excess of 765 MIPS.
- A BBN GP1000 (Butterfly I) with 96 processors. This machine was acquired as part of a joint agreement with Michigan State University and is sited at MSU.
- An Alliant FX/8 system with 8 vector processors sharing 32 megabytes of memory. The system is used primarily for research on numerical algorithms.
- An Encore Multimax system with 20 processors sharing 64 megabytes of memory. This system permits the future addition of "clusters" of processors which share memory directly and which are further connected to a large global memory.
- A Sequent b21 system with 24 processors sharing 24 megabytes of memory. This system is effective for programming methodology investigations and other non-numeric tasks.
- An Intel iPSC/d5 five-dimensional hypercube system with 32 nodes each having one-half megabyte of local memory. Because it has no shared memory, all communication and synchronization among tasks are done by message passing.
- An Intel iPSC-4X/d4 four-dimensional hypercube system with 16 vector processors, each with 1.5 megabytes of memory.
- A 4-processor Ardent Titan graphics supercomputer with a peak performance of 64 megaFLOPS. The Ardent enables scientists to explore the effect of visualization tools, which are becoming increasingly important for large-scale scientific computing.
- A 2-processor Sequent Symmetry machine, used primarily for netlib.

A Cydrome Cydra 5 minisupercomputer was also installed in the ACRF for about six months. This machine was donated by Cydrome but had to be removed when Cydrome closed in November 1988.

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.

To ensure optimal use of these machines, the ACRF has installed a 29-Gigabyte Epoch I fileserver. The new system uses a combination of RAM buffers, magnetic disks, and WORM optical disks to provide NFS service to the network. The system automatically migrates files to and from the optical disks to keep the magnetic disks free for caching the most frequently used disk blocks. The installation is part of a cooperative project with Epoch systems to determine the suitability of such servers for advanced scientific computing environments.

Our staff members access the ACRF computers through scientific workstations. Each permanent staff member is provided with a Sun workstation to provide a sophisticated and responsive environment for program development and to serve as a graphics output device.

Augmenting the Sun workstations is a newly installed network of 21 NeXT computers. The NeXT computers offer an advanced software development environment, the Mach operating system, and Display Postscript Imaging models. With the NextStep tools, users have constructed programs with professional user interfaces in a fraction of the time required with X windows or Sunview. We are also experimenting with the NeXT sound recording capability, Objective-C programming environment, Mathematica, and other NeXT bundled software. The NeXT-based workstations will ultimately be used for teaching the ACRF courses and other advanced courses on object-oriented programming and computer algebra.

To encourage outside use of the ACRF advanced computers, we have linked the ACRF to various national networks. Researchers may access the multiprocessors remotely through Tymnet, for example. Our main local area network has also been connected to the NSFnet via a dedicated line from Argonne to the University of Illinois at Urbana.

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:

- a two-week institute in advanced computing held in September 1988 (and scheduled again for September 1989) for graduate students and postdoctoral researchers from around the country (sponsored in part by DOE and in part by the National Science Foundation);

- an ACRF Industry Affiliate Program, established in early 1989 to encourage interactions with industry. The program is coordinated by ARCI, an organization established to encourage the transfer of research results from Argonne and the University of Chicago to private industry; and

- an ACRF University Affiliate Program, which provides a means whereby academic institutions may link their facilities with ours and give their students access to the ACRF machines for educational programs in parallel computing.
4 Analytical and Numerical Methods

Research in Analytical and Numerical Methods involves the analysis of mathematical equations, the application of analytical and numerical techniques to problems in the natural and engineering sciences, the design and analysis of numerical algorithms, the development of special techniques to measure algorithm reliability and efficiency, and the preparation of software based on broadly applicable computational methods. Uniting the diverse activities is a common goal: to gain knowledge about the behavior and properties of physical phenomena and to develop predictive tools for experiments.

Highlights for 1988-1989 include resolution of a long-standing open problem about the oscillation or nonoscillation of solutions of an Emden-Fowler equation and development of a convergence theory of trust region methods.

4.1 Nonlinear Differential Equations (H. Kaper, M. K. Kwong)

Current research focuses on the qualitative analysis of certain nonlinear Emden-Fowler equations. These equations arise in the study of semilinear reaction-diffusion equations, for example, when solutions with radial symmetry or similarity solutions are sought. During 1988, we were able to resolve a long-standing open problem about the oscillation or nonoscillation of solutions of an Emden-Fowler equation. This work is of fundamental importance for the problem of the existence of a ground state for the original semilinear elliptic equation.

We also improved on recent work by Friedman, Friedman, and McLeod and by Peletier and Troy. Our results are connected with the study of the profile of the solution to a semilinear parabolic differential equation that exhibits the phenomenon of extinction time. Our method relies on results from the classical Sturm comparison theory for linear differential equations.

Other investigations were concerned with the phenomena of blow-up and quenching at finite time for semilinear parabolic differential equations. In blow-up, the solution itself develops a singularity at some finite time; in quenching, it is one of its derivatives that becomes infinite. We identified a gap in the proof of a widely quoted result of Kawarada (Publ. RIMS, Kyoto University 10, pp. 729-736, 1975). The author claimed to prove that, for a rather simple equation, the time-derivative of the solution becomes infinite at the time of quenching. We found that the gap in the proof could not be corrected using Kawarada’s approach. We developed a new approach, which turned out to be applicable to a much wider class of equations. This work was presented in Nonlinear Analysis.

We completed a study of the uniqueness problem for the ground state of a semilinear elliptic equation with a reaction term that grows sublinearly for large values of the argument. Our results confirm what most workers in the subject believed to be true but were unable to prove with conventional techniques. We tackled the problem with the theory of differential inequalities in a way never used before in the study of the problem.

We successfully completed a study of the uniqueness problem for the ground state of the nonlinear reaction-diffusion equation \( \Delta u + f(u) = 0 \). The techniques that can be used on this problem depend strongly on the properties of the nonlinear function \( f \). This case is further complicated by the occurrence of a critical exponent \( p^* \) (when \( f \) is a monomial), whose value depends on the dimension \( n \) of the underlying space. A ground state exists if and only if \( p \) is strictly less than \( p^* \).
Coffman was the first to show that in three dimensions \((n = 3, p^* = 5)\), the ground state is unique when \(p = 3\). He attributed the basic principle underlying his proof to Kolodner. McLeod and Serrin subsequently extended this result to cover all \(p \leq 3\). However, the method of proof fails for the range \(3 < p < 5\). We were able to fill in this gap. Our result covers all dimensions \(n \geq 3\). The method used relies heavily on ordinary differential equations techniques and point set topology.

Also during late 1988, we investigated free boundary problems for the Emden-Fowler equation, which arise in the study of various physical phenomena, for example the study of Tokamak equilibria with magnetic islands in plasma physics. The differential equation that models such phenomena is one of Emden-Fowler type on the real line with a nonlinearity that either takes a negative value when the argument is zero or approaches zero sublinearly when the argument approaches zero. The objective of the study is to show the existence and uniqueness of a solution that vanishes with zero slope outside a bounded interval. Since the exact boundary of the interval is not prescribed, but rather dictated by the given conditions, the problem is known as a "free boundary" problem. We provided an existence proof that covers a very wide class of equations. Our approach can be modified to handle some equations with a singularity at the origin (where one of the coefficients of the equation blows up). Uniqueness can also be established in some particular cases.

4.2 Spectral Theory and Boundary Value Problems (H. Kaper, G. Leaf)

This year we completed the four-volume proceedings of our Focused Research Program in "Spectral Theory and Boundary Value Problems," held at Argonne in 1987.

Our efforts in spectral theory concluded with an investigation of the numerical calculation of eigenvalues for Sturm-Liouville operators with matrix coefficients. The main result of our investigation was the development of a procedure that can calculate any specified eigenvalue to a given order of accuracy without a priori knowledge of any other eigenvalues. The procedure applies to the class of operators with regular positive-definite weight matrix functions and separated boundary conditions. The algorithm is based on a theory developed by F. V. Atkinson of Pruefer-type transformations for Sturm-Liouville operators with matrix coefficients. In this theory one associates a group of unitary matrix functions with the operator. Determining a specified eigenvalue involves determining the winding number of a unitary matrix function which in turn requires a careful tracking of the eigenvalues of the unitary matrix function. Since these latter eigenvalues lie on the unit circle, a key feature of our procedure is that any errors in this determination are in the phase but not in the amplitude.

4.3 Inequalities (M. K. Kwong)

Our research program in inequalities focuses on norm inequalities of the Landau type for linear operators and other inequalities that arise in the study of differential equations. We are also interested in Hermitian matrix inequalities.

Landau's inequality states that for all twice differentiable functions \(f\) defined on the whole real line, \(\|f'\|^2 \leq K\|f''\|\|f\|\), where \(K\) is a constant independent of \(f\). Landau-type inequalities are generalizations of this inequality, which may involve operators other than the differentiation operator and function spaces other than \(L^2(R)\). The existence of such a constant \(K\) depends on the underlying function space(s), as well as on the operator. Finding the best possible constant \(K\)
and determining the associated extremal element, if it exists, are the primary objectives of research in this area.

An inequality similar to Landau's inequality plays an important role in the study of periodic solutions of ordinary differential equations. We have been seeking an $L^p$ extension and higher order analogues of this inequality. We have also initiated an investigation of Moser's inequality which arises in the study of an elliptic equation.

During 1988, we developed a proof of Landau-type inequalities involving the difference operator and three possibly distinct vector norms. Specifically, we reduced the discrete case to the continuous case by using a spline construction of Z. Ditzian.

We solved the problem of Landau's inequality for 2 by 2 matrices in the Hilbert space $l^2$, using the MAPLE software package. We found an explicit expression for the best possible constant in terms of the entries of the matrix. The surprising result was that the constant is given not by one single formula, but rather by three separate ones, each valid in a certain range of the parameters. One of these formulas gives a constant value over the entire range.
We generalized the classical Lyapunov theorem to solutions of the Hermitian matrix equation
\[ AAX + XAA + tAXA = P, \]
where \( A \) and \( P \) are positive Hermitian matrices and \( t \) a scalar value within a certain interval that depends on the dimension.

With the help of M. Dever, a student from Northern Illinois University, we determined some bounds on the interval for \( t \) in the above result for dimensions up to 8. In this effort we used the symbolic manipulation programs MAPLE and REDUCE.

We also investigated the notion of matrix monotone and matrix concave functions, first introduced by Loewner and Kraus, respectively. We were able to show that if a function \( f(t) \) is matrix monotone, then under almost all circumstances it is strictly concave while \( tf(t) \) and \( f(t)/t \) are strictly convex.

Finally, with the help of D. Cohoon, a student research participant in the summer of 1988, we were able to make some progress in the study of matrix Landau inequalities in the \( l^1 \) and \( l^\infty \) spaces. The numerical and graphical capabilities of the software MATLAB were used extensively in experimenting with various examples of matrices. From the results of these experiments, we conjectured and proved that given a matrix \( A \), the best constant is always achieved at a vector \( x \) when one of the components of \( x \) or \( Ax \) or \( A^2x \) is zero. This allows us at least in the case of 2 by 2 matrices to reduce the determination of the best constant to examining just a finite number of cases. Our next efforts will focus on obtaining a more general formula expressing the constant in terms of the elements of the given matrix \( A \).

4.4 Delay Equations (M. K. Kwong)

Physical problems involving processes that are affected by past history are modeled with differential equations containing a delay function. Solutions of delay equations can behave very differently from similar equations without delay. For instance, solutions of a first-order linear differential equation cannot have an infinite number of zeros, while some of those of the corresponding delay equation can. Such oscillatory solutions describe interesting natural phenomena. Mathematicians are therefore interested in obtaining criteria that guarantee the oscillation of all solutions of a delay equation.

To this end, we are using a technique based on Riccati equations to derive new criteria and to confirm a well-known conjecture by Hunt and York. Working with S. Du of the University of Southwestern Louisiana, we extended the Riccati equation technique to the study of second-order delay equations. We established a generalization of the classical Sturm comparison theorem for delay equations. This result differs from other comparison theorems in that more precise inequalities between the solutions of the equations are obtainable, instead of merely a prediction of oscillation or nonoscillation. As an application we extended a result of Moroney on uniqueness of solutions of a nonlinear boundary value problem to delay equations. The new inequalities are indispensable in the proof. Our approach also yielded oscillation criteria, improving earlier results of Mahfoud and Wong.
We have begun a new initiative in symbolic computational methods. Our initial efforts have focused on two areas: matrix inequalities and complex analysis.

A study of a Hermitian matrix inequality in high dimensions was carried out with the help of M. Dever (Northern Illinois University). Using topological techniques, we established the validity of the inequality for small values of a certain parameter. However, although the inequality is known to remain valid for larger values of the parameter, the topological approach fails to recover such values. An algebraic approach is theoretically feasible, but the amount of computation involved increases tremendously as the dimension increases. Manual calculation stopped at dimension four. Using MAPLE, we succeeded in going up to dimension eight.

We also used the MAPLE programs to study the matrix Landau inequality. Little progress has been made in this area since it was first introduced by Ljubic in 1960. There is a theoretically simple approach, but again the amount of computation has made it impractical. With MAPLE, we obtained a complete solution for the 2 by 2 case in the space $l^2$. Furthermore the work has inspired several results applicable to higher dimensions. Specifically, with the help of D. Cohoon (a student research participant), we have made progress on the problem in the spaces $l^1$ and $l^\infty$. Such results would not have easily been obtained without the insights provided by the use of the MAPLE programs.

In addition to studying matrix inequalities, we investigated the existence of multiple positive solutions for semilinear elliptic equations. Specifically, given a domain $D$ in $R^n$ and a nonlinearity $F(u)$, under what conditions does the Dirichlet problem $\Delta u + F(u) = 0$ in $D$, with $u = 0$ on the boundary of $D$ have multiple positive solutions? Such solutions can usually be characterized as states that minimize certain energy functionals and are therefore called ground states. The issue of multiple states is of importance since the solutions represent possible equilibrium states that might be attained by physical systems. When the domain is a sphere and $F(u) = u^p$, the problem is well understood; yet little is known about the perturbed nonlinearity $F(u) = u^p + f(u)$ for spherical domains. By scaling, we obtained the closely related nonlinear eigenvalue problem $\Delta u + u^p + \lambda f(u) = 0$ in the unit sphere, or between two fixed annuli. For particular choices of the perturbation term $f(u)$, we would like to study the global structure, especially regarding the presence or absence of bifurcations, in terms of the parameter $\lambda$ (or the radius of the spherical domain).

During 1988 we concentrated our effort on obtaining tools to conduct this project. We were already familiar with MAPLE and MACSYMA. To these, we decided to add the AUTO package of E. Doedel for numerically solving bifurcation systems of ordinary differential equations. We modified AUTO to interface to a local graphics library and run on our Sun workstations and Alliant FX/8. We also obtained a versatile package for solving boundary value problems for ordinary differential equations, COLSYS developed by U. Ascher, J. Christiansen, and R. D. Russell. We then employed these tools in studying some sample problems, involving the nonlinearity $F(u) = u^p \pm u$ and the domain between two concentric annuli. Our initial effort demonstrated that the computational tools are suitable for mathematical analysis.
4.6 Optimization (A. Griewank, J. Mora)

Optimization research focuses on the development of algorithms for optimization problems that are of importance in scientific computing. The aim of this work is to understand the optimization problem, design and analyze an algorithm, and implement the algorithm into software. Previous research in this program has led to significant progress in the use of trust region methods for the solution of unconstrained problems.

Specifically, we have developed a convergence theory for the trust region method for convex and linearly constrained problems. This work is important because it applies to practical algorithms for the solution of the trust region subproblem and does not make any assumptions about linear independence. The research was motivated by our interest in large-scale bound-constrained problems; these problems are of especial importance in scientific computing because they arise naturally as requirements that the parameters lie in a given range of values. A major thrust of our current work is thus the development of algorithms and software for bound-constrained optimization problems.

During 1988, we completed our investigation of the global and local convergence conditions of trust region methods for convex and linearly constrained problems. The motivation for this work is to show that algorithms based on trust region methods have stronger convergence properties than algorithms based on either line searches or sequential quadratic programming. Global convergence is established for general convex constraints while the local analysis is for linearly constrained problems. The main local result shows that if the sequence converges to a nondegenerate stationary point, then the active constraints at the solution are identified in a finite number of iterations. This result is important in the solution of large-scale problems. As a consequence of the identification properties, we develop rate of convergence results by assuming that the step is defined by a truncated Newton method. One of the advantages of our approach is that it is geometrical, and thus the theory can be developed without any assumptions of linear independence. Another advantage is that our assumptions concerning the trust region subproblem can be satisfied by practical algorithms; previous results made the unrealistic assumption that it was possible to obtain the global solution of the trust region subproblem.

We also completed a pilot project on the numerical performance of algorithms for the solution of quadratic programming problems subject to bounds. The aim of this work is to compare algorithms that use active set strategies with algorithms that use the gradient projection method.

One of the novel features of recent versions of the gradient projection method is that the search parameter $a_k$ is only required to satisfy a sufficient decrease condition. For a convex quadratic, this is illustrated in Figure 3. In this figure $\phi_k$ is a piecewise quadratic, piecewise convex function which measures the reduction in the function, while $\psi_k$ is a piecewise linear function which defines the sufficient decrease criterion. The search parameter must satisfy $\phi_k(a_k) \leq \psi_k(a_k)$.

Tests were carried out on a wide range of problems with 100 variables. The numerical results show that on mildly degenerate problems the gradient projection algorithm requires considerably fewer iterations and less time than the active set strategy. On nondegenerate problems the number of iterations typically decreases by a factor of at least 10. For strongly degenerate problems the performance of the gradient algorithm deteriorates, but it still performs better than the active set strategy. We expect that improvements will be even more dramatic on larger problems, and thus we will pursue this approach further.

Investigations continued on the identification properties of optimization algorithms, with the
aim of extending our results for linear and convex constraints to general constrained problems. Our colleague J. Burke (University of Washington) has characterized those algorithms that identify the optimal active constraints in a finite number of iterations. He has carefully examined the role of the linear independence condition and shown it to be required in a first-order theory of constraint identification. The results have been applied, in particular, to the sequential quadratic programming algorithm and the QL algorithm.

We also completed a joint project with S. Vavasis (Cornell University) on a concave knapsack problem. Motivation for this research derives from the observation that combinatorial optimization problems can be modeled by separable concave minimization problems subject to bounds constraints and one equality constraint. We have characterized strict minimizers of concave minimization problems subject to linear constraints. This result is unusual because in general it is not possible to characterize local minimizers. We have also used this characterization to develop an algorithm for the concave knapsack problem that determines a local minimizer in order $n \log n$ operations. This is the first algorithm guaranteed to find a local minimizer of a concave problem in a polynomial number of operations. Previous results assumed convexity of the problem.

Figure 3: Sufficient decrease condition for $\alpha_k$
We continued a project on nonlinear least squares problems. This research was motivated by inverse problems for certain operator equations, for example, the determination of diffusion coefficients of elliptic partial differential equations. In these problems the numerical solution of the discretized equations is typically quite costly, and thus difference approximations to all partial derivatives are likely to be expensive. In cooperation with L. Sheng (Del-Mar College) we developed a quasi-Gauss-Newton method related to the Broyden formula. This algorithm requires only one function evaluation per iteration and yields excellent results on our test problems. The underlying equation is solved only once per iteration, and this leads to significant savings in the number of arithmetic operations per step. Linear problems are solved in a finite number of steps, and on nonlinear problems the rate of convergence is usually comparable to that of the Gauss-Newton method. Because the search directions are not guaranteed to be downhill, the line search procedure requires particular attention. We completed work on the line-search with satisfactory results. As a spin-off we can prove global convergence for a univariate equation solver that does not enforce the customary, but occasionally hindering, interval reduction ratios.

In cooperation with R. Drake (Washington State University), we implemented and tested a new line search procedure which appears to recover superlinear convergence to some singular roots, where the unmodified Newton method converges only rather slowly. The method is thus particularly useful in numerical bifurcation and continuation codes, where (nearly) singular systems arise at (near) turning and bifurcation points. This assertion was numerically checked at and near the turning point of the Bratu problem in a cylinder discretized by bilinear finite elements. On the Argonne test set the new line-search achieved about the same performance as the considerably more complicated tensor method.

Finally, in cooperation with Th. Jongen (Technical University Aachen), Man Kwong, and R. Womersley (University of New South Wales), we have shown that a Lipschitzian objective function with a bounded level set is strictly convex if and only if its generalized gradient is injective. This equivalence is of considerable interest for the convergence analysis of variable metric methods.

4.7 Bifurcation Phenomena in Combustion (H. Kaper, G. Leaf, M. Minkoff, A. Bayliss, B. Motkowsky)

A long-term research effort is aimed at understanding the dynamics and structure of flames and combustion fronts in gaseous and solid-fuel combustion, respectively. Investigations focus on bifurcation phenomena as a potential mechanism for transition from laminar to turbulent combustion. Research involves efforts in modeling, analysis, and computation.

During 1988, we investigated problems related to the structure and stability of curved flames. In particular, we considered the two-dimensional problem of a flame stabilized on a point source of fuel. The circular flame is a solution of this problem, which we term the basic solution. We investigated the stability of this basic solution and found that there are two neutral stability boundaries that occur for Lewis numbers less than and greater than unity, respectively. Crossing the former boundary leads to the onset of cellular flames, whereas crossing the latter boundary leads to the onset of pulsating flames. We used these results as a starting point for the numerical computation of the cascade of bifurcations that occur after the uniform circular flame becomes unstable. We first found a transition to a cellular solution, by decreasing the Lewis number which was employed as the bifurcation parameter, with the source strength kept constant. This work is reported in *Applied Math. Letters*. We then found a transition to cellular solutions by increasing the
source strength, with the Lewis number kept constant. Upon further increasing the source strength, we found a transition from a stationary three-cell solution to a stationary four-cell solution, with an interval of bistability in which both the three-cell and the four-cell solutions coexist, each with its own domain of attraction. We then found a cascade of transitions from the stationary three-cell solution to the stationary four-cell solution to the stationary five-cell solution to the stationary six-cell solution.

We continued to explore improvements of our code for the numerical investigation of the dynamics of pulsating and cellular flames in gaseous mixtures. Specifically, we attempted to find optimal coordinate systems by considering alternative choices of functionals to measure the interpolation error and by developing optimization procedures to minimize the functionals employed. We successfully developed a new family of functionals $I_r(u)$, with the property that the maximum norm of the error is bounded by $I_r(u)/J^r$, where $r$ is an integer and $J$ is the degree of the polynomial approximation. These functionals are used in our adaptive procedure whereby the problem is dynamically transformed to minimize $I_r(u)$. The number of collocation points is then chosen to maintain a prescribed error bound. The new functionals enjoy significant advantages over the functionals previously employed.

We extended our study of solid-fuel combustion. We previously had found that a period-doubling secondary bifurcation occurred as we increased the bifurcation parameter. One might have expected to see a sequence of period-doubling secondary bifurcations leading to chaos, but our results for the parameter regime considered indicated that the system returns to the singly periodic branch of solutions. We note that the route to chaos via a sequence of period doubling bifurcations may yet occur for other parameter regimes.

4.8 Computational Fluid Dynamics (G. Leaf)

The objective of our fluid flow research is to develop an effective multigrid algorithm based on the use of a nonuniform adaptive spatial grid. Our investigations involve several stages. First we developed an efficient multigrid algorithm for the Navier-Stokes equations, which used a novel relaxation scheme. We then developed a multigrid algorithm based on nonuniform spatial grids. The prototype algorithm was limited to one set of nested refinements. The next stage features many sets of subrefinements based on a quad-tree refinement strategy. Finally, we will investigate an adaptive strategy for this quad-tree refinement and explore extensions for complex domains.

During 1988, we investigated the use of composite grids in conjunction with multigrid methods for the numerical solution of fluid flow problems. Typical flows exhibit small regions of rapid variation that require fine grids for a full resolution. Outside these regions the flow is smooth, so coarse grids are adequate. This investigation focused on the use of a sequence of overlaid uniform grids to generate a nonuniform approximation on the finest grid. Specifically, we developed a procedure based on the use of intermediate grids that do not extend over the entire domain. Several issues were considered, including the use of appropriate transfer functions between grids, the use of the governing equations near the interface between grids at different levels, and the development of an appropriate data structure to describe the various grids and associated transfer functions.

We developed a transfer technique that preserves mass balance during transfers, thus eliminating the need for rebalancing after grid transfer. We also devised a correct technique for applying
the governing equations near grid interfaces. We tested this concept in both a non-multigrid and multigrid composite-grid algorithm. The data structure for the current algorithm allows each grid to be an arbitrary rectangular subdomain of the next coarser grid. This data structure is limited but allows efficient resolution of a boundary layer or point singularity. We used this approach to solve cavity-driven flows with Reynolds number up to 1000. We calculated these flows with full multigrid efficiency, despite the corner singularity present in this problem.

4.9 Numerical Methods for Partial Differential Equations (G. Leaf, M. Minkoff)

This activity deals with the analysis of methods and software design for solving nonlinear systems of partial differential equations (PDEs). Currently our interest focuses on pseudo-spectral methods and preconditioned iterative methods.

In collaboration with A. Bayliss (Northwestern University) and D. Gottlieb (Brown University and ICASE), we analyzed functionals that measure the spectral interpolation error for the Chebyshev pseudo-spectral method. Our implementation of this method relies on the introduction of a family of coordinate transformations in which the solution can be more accurately approximated. Its effectiveness depends critically upon the accuracy of the measure of the error. We introduced and validated a new functional that more accurately monitors the spectral interpolation error in the pseudo-spectral method. We evaluated this functional in the context of our computational analysis of bifurcation phenomena in combustion. We also introduced a second-order accurate time discretization to compute higher dimensional solutions more effectively, and developed new optimization techniques to obtain optimal coordinate transformations.

In finite-difference discretizations of the system of PDEs of fluid flow, convective effects can generate indefinite nonsymmetric matrices that are regular, sparse, and of large order. Direct methods for solving such systems require pivoting, which generates fill-in and results in large storage requirements. Storage cost is indeed a major limitation in the solution of multidimensional PDEs. We are investigating the potential of preconditioned iterative methods for solving such linear systems. Specifically, we have focused our attention on two classes of preconditioners: nested factorization and approximate inverses.

Nested factorization is based on a recursive derivation of an approximate Cholesky-type factorization at each level of the block tridiagonal structure. The recursively defined factors form an approximate block LU factorization, which is stored for use in the solution of the linear system. This factorization procedure exploits the block tridiagonal structure as much as possible. We have developed and implemented two variants; the first preserves column sums, the second row sums. One motivation for using the first variant is that it preserves mass balances for systems arising in physical systems.

Another class of preconditioners can be developed on the basis of a direct approximation of the inverse of the matrix. This approach leads to matrix-vector multiplies in the iterative method, whereas the nested factorization approach leads to solving linear systems at each step. We developed a technique that uses approximate inverses to generate the lower-level factorizations while still using the overall approach of nested factorization. This technique leads to an algorithm that is significantly different from the variants based on column and row sum preservations. Specifically, at each stage where an inverse is needed, we select an approximation to the inverse that, within the class of matrices under consideration, is a best approximation in the sense of the Frobenius norm.
We designed and implemented preconditioners based on the use of the diagonal and tridiagonal approximate inverses.

To evaluate the usefulness of nested factorization and approximate inverses, we considered a finite-difference approximation of a three-dimensional elliptic equation involving diffusion, convection, and absorption terms. By varying the convective term, we controlled the indefiniteness and asymmetry of the coefficient matrix.

We concluded that for a variety of all Peclet numbers, the preconditioner based on the use of approximate tridiagonal inverses was the most robust and efficient procedure for this class of indefinite and nonsymmetric linear systems. We also concluded that this preconditioner has the most potential for parallel and vector exploitation.
4.10 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 new class of rules for high-dimensional quadrature.

Lattice rules are extensions of the number theoretic numerical quadrature rules associated with Koborov and Conroy. With I. H. Sloan (University of New South Wales, Australia) we continued our investigation into the group theoretical structure of these rules.

Our primary attention, however, has focused on the generator matrix approach to lattice theory. Working with collaborators in Bergen, Norway, and in Halifax, Nova Scotia, we developed a theory that describes the structure of the sublattices of sets of lattices. This theory exploits the Hermite normal form of an integer matrix to develop sequences of embedded quadrature rules.

We continued our collaboration with researchers at the University of Naples developing software
for the numerical inversion of the Laplace transform. Our software for implementing the Weeks method has now appeared in the *ACM Trans. on Math. Software* and is being incorporated in the Numerical Algorithms Group (NAG) and IMSL software libraries.

### 4.11 Special Functions (W. J. Cody, P. T. P. Tang)

Our research in special functions includes (1) work on approximation theory; (2) the development of numerical methods for approximating functions; (3) the development of approximations, algorithms, and testing methodology for elementary functions; and (4) the development of approximations, algorithms, and testing methodology for the special functions of mathematical physics.

The clean arithmetic environment specified by the new IEEE standard poses new challenges for this activity. For example, we believe it is now possible to design nearly perfectly accurate elementary function programs and to support the design with rigorous proof of the accuracy. As the algorithms and software for functions become more sophisticated, it is necessary to develop more discerning techniques for performance evaluation. These ideas carry over to the work on special functions, which continues to focus on algorithms and methods of performance evaluation. Research concentrates on algorithms that exploit the new IEEE architecture but remain tolerant of older, more flawed architectures.

As part of our research in approximation theory, we documented the generalization of the quadratically convergent Remez algorithm from real to complex in a paper entitled “A fast algorithm for linear complex Chebyshev approximations.” We also presented a shorter version of the paper, containing a new proof, at the Conference on Algorithms for Approximation held in July 1988 in Swindon, England.

In our work on elementary functions, we developed a methodology for portable implementation and verification of elementary function subroutines. This work was documented in the paper “Use of language features in the implementation and validation of a standardized generic package of elementary functions in Ada,” which we presented to the Sandia Workshop on Ada held in the fall of 1988.

Other work on elementary functions concentrated on algorithms tailored to IEEE arithmetic. We prepared implementation guides with detailed error analysis for the functions $e^x$, $e^x - 1$, $\log x$, and $\log(1 + x)$. We also developed portable and highly accurate testing methods for $e^x$ and $\log x$.

Work on testing procedures for special functions continued. We developed new accuracy-testing methods for Dawson’s integral, error functions, and certain Bessel functions. We added two new function programs (one for the exponential integrals $Ei(x)$ and $E_1(x)$, and one for Dawson’s integral), and six performance evaluation programs (for Dawson’s integral, error and complementary error functions, and the Bessel functions $K_\nu(x)$, $I_0(x)$, $I_1(x)$, and $I_\nu(x)$) to the SPECFUN package. Part of this work was conducted with the assistance of L. Stoltz, a resident student associate.

We developed algorithms for several of the ancillary functions from the appendices to the IEEE floating-point standards. This work was documented in the paper “ALGORITHM XXX: Functions to Support the IEEE Standard for Binary Floating-Point Arithmetic.”
5 Advanced Computing Concepts

Research in Advanced Computing Concepts involves the design of algorithms for numerical linear algebra, the formulation of new inference rules and strategies in automated reasoning, the development of parallel systems based on logic programming, the analysis of abstract programming methods and program transformation techniques, and the development of parallel programming tools.

Highlights for 1988-1989 include use of our powerful new automated reasoning program to obtain proofs of theorems from elementary set theory and development of new algorithms that resolve long-standing questions about orthogonal eigenvectors. We have also begun a new project in advanced computational methods for biology as an application of our work in logic programming.

5.1 Numerical Linear Algebra (C. Bischof, J. Dongarra, D. Sorensen)

Our efforts in numerical linear algebra involve recasting many of the standard algorithms for the solution of linear equations. In some cases, new algorithms must be developed; in other cases, existing algorithms must be refined (particularly for the eigenvalue problem). Of vital importance to the development of efficient yet portable algorithms is the use of the Level 1, 2, and 3 Basic Linear Algebra Subprograms (BLAS). The algorithms developed in this project will form the basis for a new software package, LAPACK, which we expect will become the standard software for linear algebra on advanced computers.

During 1988, the code for the symmetric eigenvalue problem was restructured in terms of the Level 3 BLAS and made independent of the SCHEDULE package. In collaboration with P. Tang, we developed new algorithms to more accurately compute the eigenvectors. The new techniques completely resolve any numerical questions concerning the computation of orthogonal eigenvalues in the face of finite-precision arithmetic. These new techniques are of two types: one uses simulation of extended-precision arithmetic, and the other is based on a new iterative method to solve the secular equation.

A version of the divide-and-conquer technique for the symmetric eigenvalue problem was developed for SIMD architectures. A code has been developed and run on the Connection Machine. This work, which is still in progress, is being done in collaboration with L. Johnsson at Thinking Machines Corporation. We presented our initial results at an international conference on vector and supercomputing in Tromso, Norway, June 1988; we are also preparing a manuscript for publication in the open literature.

A restricted pivoting strategy for the QR factorization was developed to circumvent data-dependent column exchanges associated with traditional column pivoting strategies. The strategy hinges on a new technique called incremental condition estimation to monitor the numerical reliability of triangular factorizations inexpensively. Experiments on hypercube machines show that restricted pivoting reduces communication overhead for pivoting by about 50 percent. The strategy also makes it possible to formulate a blocked algorithm for the QR factorization of a rank-deficient matrix.
5.2 Automated Reasoning (L. Wos, W. McCune)

Our long-term goal in automated reasoning is to provide software that is extremely effective for that aspect of problem solving focusing on reasoning, especially on logical reasoning. We attack questions—some of which are open—taken from various areas of science to measure our progress, to determine where next to put our effort, to demonstrate the usefulness of reasoning programs for diverse applications, and to show how one can use our programs for the needed research in automated reasoning itself.

During 1988-1989, our investigations in combinatory logic were extended to a fuller development of a systematic approach for constructing (where they exist) fixed point combinators when given some chosen set of combinators. To complement this method, we succeeded in proving theorems for identifying certain sets of combinators for which such a construction is impossible. The power of this new construction method, called the kernel method, was demonstrated by an experiment that found five combinators in 3 CPU seconds on a Sun workstation, compared to the 20 CPU hours required to find them originally. The kernel method is efficient in that it requires far less computer time than typical programs. It is also pleasing in that, unlike many computer-oriented approaches, it can be applied by hand. Finally, the method has the advantage that it can be applied to any set that one conjectures to have the strong or weak fixed-point property.

We have applied our new kernel method—and a complementary theory we just developed—to various open questions focusing on the presence and absence of fixed point properties for given sets of combinators. Those results were published in an Argonne report, which will provide the basis of a book that shows how research can be conducted successfully with an automated reasoning program.

We continued design and implementation of a new automated theorem-proving program, called OTTER, which is permitting us to explore research paths that have previously been inaccessible. The program, coded in C, is similar in scope and purpose to the AURA and LMA/ITP programs, with an emphasis on performance, portability, and ease of use. A preliminary version of OTTER has been distributed to about 80 researchers.

OTTER is also extendable to parallel form. Early work on a parallel version of OTTER has been encouraging; when we tested several theorems, we obtained nearly linear speedups with up to 20 processors.

Among other successes with OTTER, we were able to obtain the first computer proof of a theorem by Lukasiewicz, a theorem suggested as a challenge problem for automated theorem-proving programs. Of additional interest, the proof produced by OTTER is substantially different from that of Lukasiewicz. In a different area, we successfully used OTTER to obtain the first unaided proof of two deep theorems of equivalential calculus. Additionally, we developed a uniform approach for attacking problems of this type and, using this approach, found far more satisfactory proofs—shorter and less complex than previously known.

We also succeeded in producing a set of clauses for studying problems from topology. With these clauses, OTTER obtained proofs of some elementary theorems. In a related study, we obtained evidence suggesting that we can use OTTER for proving theorems from elementary set theory. Some of the results of the two studies have been published in the *Journal of Automated Reasoning*.

We formulated a new literal ordering strategy. Preliminary experiments with that strategy suggest that its use will materially reduce the execution time for applying inference rules such
as UR-resolution. This strategy should also prove valuable for decreasing the time required for nonunit subsumption and for the application of linked inference rules.

Our interactions with researchers have been rewarding. In particular, with R. Veroff (University of New Mexico), members of our group extended and formalized additional aspects of linked inference. The intention is to write an extensive document that details the various abstract and practical aspects of linked inference rules. We also held a workshop on advanced topics in automated reasoning, which led to new collaborations on topics such as program synthesis and program verification.

Work continued on a database of problems, lemmas, and theorems as tests for evaluating new ideas in automated reasoning. We transformed it to be compatible with OTTER and then verified a substantial part of the database. We plan to make the database accessible in 1989 via electronic mail throughout the world.

5.3 Logic Programming (E. Lusk, R. Overbeek, R. Stevens, I. Foster, S. Winker)

Our work is motivated by the desire to produce the fastest possible automated reasoning systems capable of exploiting state-of-the-art computers. We currently believe that the way to achieve this goal lies through parallel computing and logic programming. Thus the first phase of our work involved research into portable methods for programming parallel computers. The current phase involves the application of the general-purpose methods and software developed in the first phase to several research projects, including the development of a general parallel logic programming system and the parallelization of the automated reasoning algorithm we have used for years. This work has enabled us to begin investigating whether logic programming is an appropriate paradigm for programming parallel computers in a high-level language, and whether such a high-level language (like Prolog) can compete with more traditional procedural languages (like C) on a major application like automated reasoning. The next phase will involve the construction of high-performance automated reasoning systems based on the insights gained in the first two phases.

The Gigalips Project (a collaboration among Manchester University, the Swedish Institute of Computer Science, and Argonne) produced Aurora Prolog, an experimental OR-parallel logic programming system. In 1988 the system went from being minimally functional to being stable enough for extensive experimentation and tuning to take place, although considerable work remains before the system can be considered finished. Aurora represents one type of parallel logic programming system, based on the existing Prolog language. It is particularly appropriate for implementation on shared-memory multiprocessors.

We began work with a different type of parallel logic programming, called "committed choice" logic programming, which is particularly appropriate for, but not limited to, the distributed-memory model of computation. We acquired early releases of one such language, Strand, which was developed by A.I. Limited in Britain. The language is similar to Flat Parlog, which was developed by researchers in the Parlog group at Imperial College, with whom we have collaborated informally for several years. We developed a parallel performance monitoring tool for Strand based on the Gauge system developed by C. K. Selman (Aerospace Corp.). It allows a wide variety of data to be gathered about the parallel execution of programs written in Strand and other languages. This data can then be examined on both a per-processor and machine-wide basis using graphical displays. We also developed an asynchronous garbage collector for Strand that works efficiently on
message-passing multiprocessors.

We experimented with a very early release of Strand on a computational biology application. We found that it was possible to express parallel algorithms conveniently in Strand, and to attain performance very close to that attainable in systems written completely in a lower-level language like C. On one important algorithm (multiple-sequence alignment for RNA sequences), we attained speedups of 9 on 16 processors. We also determined that the performance is governed by the task size (see Table 1).

In other work involving logic programming in the context of genetic sequence analysis, we wrote programs that can be used to help locate secondary and tertiary structure within ribosomal RNA. We have already identified a previously unknown structure in 7s RNA; we are now using our programs to search for more such structure. We also began a collaboration with Amoco Technology Company on a number of topics ranging from protein structure prediction to sensitivity of similarity searches. In the joint work on protein structure prediction, we helped Amoco researchers use logic programming to recognize patterns within a small library known “motifs.” We are also using Prolog to analyze spectroscopy data relating to protein structure.

Finally, we developed a message-passing version of our classical theorem-proving algorithm. This research is a first step in exploiting both workstation networks and distributed-memory multiprocessors in automated reasoning applications.

Table 1: Effect of Task Size on Run-time: 11 processors

<table>
<thead>
<tr>
<th>No. of Sequences to Be Aligned</th>
<th>Length of First Sequence</th>
<th>No. of Tasks</th>
<th>Mean Task Time, sec.</th>
<th>Total Time, sec.</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>25</td>
<td>760</td>
<td>5.2</td>
<td>472</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>269</td>
<td>10.7</td>
<td>394</td>
</tr>
<tr>
<td>10</td>
<td>50</td>
<td>177</td>
<td>16.9</td>
<td>405</td>
</tr>
<tr>
<td>20</td>
<td>50</td>
<td>137</td>
<td>22.0</td>
<td>414</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>71</td>
<td>37.9</td>
<td>486</td>
</tr>
<tr>
<td>20</td>
<td>100</td>
<td>65</td>
<td>41.8</td>
<td>461</td>
</tr>
</tbody>
</table>
5.4 Abstract Programming (J. Boyle, K. Dritz, T. Harmer)

Ideally, abstract programs are written in terms of concepts and notations taken from the problem domain—physics, graph theory, partial differential equations, linear algebra, or other theories and notations appropriate to the problem at hand. This approach makes it easy to specify the program and to verify that the specification is correct. In this way, abstract programming reduces the cost and increases the reliability of computer programming.

The chief barrier to the practical adoption of abstract programming is that abstract programs cannot be executed on a computer in the form in which they are written. They require extensive rewriting to remove the abstractions and make them efficiently executable. Thus, one focus of our research in abstract programming is to develop techniques, using program transformers or “super compilers,” to derive efficiently executable code automatically from abstract programs.

This activity is concerned with making abstract programming practical. We have been conducting an extensive study of pure Lisp, which is equivalent to Church’s lambda calculus, as an abstract programming notation.

In 1988 we studied the use of pure Lisp to specify discrete-event simulations and investigated techniques to derive parallel implementations from these specifications. Maintaining the correct relative timing of events is the main challenge in parallel implementations of discrete-event simulations. Thus, the derivation must guarantee that the derived program does not simulate an event (irrevocably) until it is known that no other event can precede it, while still preserving the possibility of parallel execution and avoiding introducing prohibitively large overheads. Determining whether another event can precede a given one requires global knowledge, so that determination can itself become a bottleneck in parallel execution. These problems are compounded if asynchronous events such as power failures are permitted in the simulation (as they must to model real physical processes).

There are two principal approaches to implementing parallel discrete-event simulations: that of Chandy and Misra, and the Time-Warp of Jefferson. We began implementing program transformations to create both of these implementations from a single specification. We specified several simple simulations to serve as tests for the transformations.

We also studied numerical abstract programming by developing a sequence of pure Lisp specifications for the “auction” integer-programming algorithm developed by David Castanon of Alphatech Corporation. Each specification in the sequence was more parallel than its predecessor, and yielded an implementation with a greater speedup on shared-memory parallel processors. This work helped us explore how one can avoid unnecessary sequentialization in specifications.

We developed optimizations for the “cons” function in Lisp, which operate in conjunction with the well-known tail-recursion elimination optimization to produce a highly efficient Fortran implementation of the abstract specification. In addition, we developed a set of transformations that optimize the implementation of programs derived from abstract specifications that make extensive use of abstract data types. Together, these implementation techniques enable us to automatically derive a program for the auction algorithm that runs only a factor of two slower than the hand-written Fortran program. We are implementing additional optimizations that will increase its speed.

We also began to study the problem of constructing an SIMD implementation of pure Lisp. In this study we treated the vector processors of a CRAY X-MP computer as a 64-processor SIMD
machine. This study has given us an understanding of how the synchronization operations for shared-memory MIMD machines translate to the more synchronous SIMD environment. Preliminary experiments with this implementation have shown rather disappointing speedups, only a factor of two over the scalar speed of the X-MP. Apparently, the high (scalar) overhead of managing the process queues necessary to support the SIMD execution is the limiting factor. These results suggest that the challenging problem that must be solved for SIMD implementations is to make a program "look as SIMD as possible." We are investigating approaches to developing such transformations.

Figure 6: Language levels between pure Lisp and parallel Fortran
5.5 Automated Program Transformation (J. Boyle, K. Dritz, T. Harmer)

For the past several years, we have been investigating methods for specifying, implementing, and proving the correctness of program transformation systems. These investigations have led to the development of TAMPR, a general program transformation system.

Currently, we are working on making the TAMPR transformer transportable. During the past year we ported the existing TAMPR system to Sun workstations in order to take advantage of their greater computational power. This port included moving the TAMPR program and transformation recognizers, the large-stack version of Lisp used by the TAMPR transformer, the transformer itself, and the simple output formatter program. In addition, we ported the Lisp optimizer used with the Lisp version of the transformer. The Sun Unix system permits Lisp to use enough memory to transform the entire TAMPR transformer into Fortran; because of limited memory, we had had to omit the symbol-generation part of the transformer when we transformed it on the VAX.

We made a number of performance enhancements to the ported TAMPR components, including producing faster versions of the recognizers and formatter. We also produced a Sun version of TAMPR that uses a programming language grammar that incorporates the Fortran 77 language and, in addition, representations for much of the C programming language. This version of TAMPR makes it possible to generate an output program in any of the languages Fortran 66, Fortran 77, or C as the result of a TAMPR derivation. We developed transformations and support code that enable the C version of a program derived from a pure Lisp specification to use absolute pointers for stack references instead of the array subscripts used in the Fortran version. We compared the performance of this C version with that of the Fortran version for a typical test program, and found the absolute-pointer C version to be 25-35% faster.

In the theoretical area, we investigated proving that program transformations achieve a goal. We wrote a set of transformations that use lambda expressions in pure Lisp to describe a portion of a code generator for a RISC (Reduced Instruction Set Computer) architecture. We used transformation induction to prove that these transformations achieve the goal of placing a program in a sem canonical form characterized by the property that each variable in an expression is fetched exactly once, regardless of how many times it appears as an operand in the expression. Based on this proof, we discovered that the set of transformations we had written to solve this problem contained unneeded transformations—ones written to handle cases that the proof showed could not occur. Eliminating these unneeded transformations reduced the application time for the set of transformations by 10-40 percent.

5.6 Automated Program Verification (K. Dritz)

This work in automated program verification involves the development of models, axioms, and proof strategies for programs using floating-point arithmetic. The investigation was motivated by a study of plant control software which involved the use of floating-point computation. Results from our research will have far more general applicability, however, and indeed have been sought by numerical analysts for a long time.

Prior to 1988, we developed tentative models and axioms for floating-point arithmetic, based on the Brown model. We also found appropriate demodulators and workable proof strategies to allow the ITP automated theorem-proving program to attack part of a proof of correctness of a...
modified Newton iteration algorithm for SQRT. ITP found a flaw in the first step of the argument prescreening and reduction: some slightly negative arguments (very close to zero) escaped rejection by argument prescreening, and some slightly positive arguments (again, very close to zero) behaved like zero during argument reduction, producing an infinite loop. This unexpected result, which is attributable to a degree of nondeterminism in the modeling of floating-point comparisons of nearly similar quantities, revealed the need either to slightly weaken the claims that are made about floating-point computations or to adopt qualifications on the behavior of the hardware and software that go beyond, and are not rooted in, the model.

During 1988-1989, we investigated ways of generating post-conditions applicable to each branch following a comparison operation. This requires integration with the theorem prover's case-splitting machinery. Our next effort will be to repeat our experiments using our powerful new automated theorem prover OTTER.

5.7 Parallel Programming Environments (J. Dongarra, D. Sorensen, E. Lusk, R. Overbeek, C. Bischof)

The goal of our research in parallel programming environments is to assist in formulating correct parallel algorithms and to mechanize as much as possible the process of translating an abstract parallel algorithm, such as one that might be conceived by an algorithm researcher, to an efficient implementation on a specific parallel machine. Over the past five years we have developed approaches in the design of certain numerical algorithms that allow for both efficiency and portability. Our current efforts focus on parallel architectures, with particular emphasis on two areas: parallel programming methodologies and environments for algorithm development.

In the first area, we extended the Fortran version of our portable monitor-macro library to include message-passing primitives (for the distributed memory model). Work also began on integrating monitors and message-passing primitives into C via the C++ programming language.

In the second area, we made several extensions to the SCHEDULE package:

- The SCHEDULE package was ported to a number of new machines including the CRAY X-MP, CRAY Y-MP, and IBM 3090 systems.
- The package was improved to allow an unlimited number of processes in a given parallel program (the previous limit was 1000).
- A technique of recycling jobtags (process identifiers) was developed to allow the iteration of a graph. This has allowed powerful loop structures to be incorporated into SCHEDULE. (This work was done in collaboration with F. Hanson, the principal investigator.)

The SCHEDULE trace facility was also enhanced. A color version was developed, and several analysis features were also incorporated. These include a histogram of processor utilization, more distinction of the various phases that a process might be in, ability to exhibit the critical path in a computation, and general improvement of the performance of the animated display.

A Build facility was also developed. This represents the first step in providing a tool for the automatic generation of a parallel program from a graphical specification. Non-trivial programs have been developed graphically, run on a parallel machine, and then analyzed for performance.
using the trace facility. At present, any program that can be represented by a static graph with at most one level of dynamic spawning can be generated. Some details must be supplied or modified textually, but all of the specification of data dependencies and calls to SCHEDULE subroutines are generated automatically. Several outside users have successfully utilized the tool to generate programs.

A collaborative effort involving the integration of SCHEDULE with the CODE package being developed by J. C. Browne (University of Texas, Austin) was begun. During the summer of 1988 it was possible to have the CODE package generate SCHEDULE code. The CODE package is another effort to generate parallel programs from a visual representation. This collaboration was significant in that it provided a wide variety of machines for which CODE could generate programs and made the visual analysis of the SCHEDULE trace facility available to CODE users. It also provided a means to generate more sophisticated and more complete programs than can be generated with the Build facility.

5.8 Software Tools for Analyzing and Transforming Programs (W. Cowell)

We are constructing software tools to support Fortran programming on advanced-architecture machines. The unifying feature of these tools is their use of data-dependency analysis, which is an essential component of adapting a Fortran program to operate on a parallel architecture and to performing transformations that improve the efficiency of memory utilization by the program. The tools are integrated into the Toolpack environment. From a tool-writing standpoint, this means that they are written in Fortran utilizing Toolpack-provided, Fortran-callable access to the parse tree/symbol table representation of a Fortran program. From a user's standpoint, these tools are an extension of the widely used Toolpack support environment for Fortran.

Tools recently added to the collection include a call tree analyzer that produces information about the input/output status of arguments in subprogram calls. This information assists the user in responding to queries posed by the DO-loop parallelization tool in its data-dependency analysis. The call tree analyzer also looks for side effects that could cause errors when a program is executed in a multiprocessing environment, for example, updating of COMMON variables in a routine called from within a loop whose iterations are executed in parallel.

In an application of these tools, a program being developed for the Strategic Defense Initiative to match weapons to targets using a transportation optimization model was converted to parallel operation. The speedup was approximately linear, with an efficiency of 80%. Other tools completed during this period were a loop marker, an expression simplifier, a loop interchange tool, and a tool to promote local scalars to arrays in loops.
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 (C. Bischof, J. Dongarra, D. Sorensen, S. Wright)

The MCS Division receives funding from the National Science Foundation to support several projects. We already noted in the introductory highlights that together with Rice University, Caltech, and Los Alamos National Laboratory, we are a partner in the newly formed NSF Center for Research on Parallel Computation. Additionally, NSF supports an internship and has provided funding for the development of a portable linear algebra software package.

6.1.1 Internship

Under a Faculty Internship funded by the National Science Foundation, we have had two people work with MCS Division members for several months.

Stephen Wright (North Carolina State University) developed algorithms for advanced-computer architectures, both vector and parallel. For example, he designed an algorithm for solving general banded linear systems on multiprocessors. Unlike other methods, this algorithm can—in principle—produce a solution to any nonsingular system. A nested dissection strategy is used to partition the matrix into main diagonal blocks that are factorized separately, leading to a smaller, “reduced” system that can be solved on one processor. The method takes account of the possibility that rank deficiency may occur in the submatrices. When this occurs, the size of the reduced system may be increased slightly, but a solution can still be obtained. The algorithm has been implemented on the ACRF's Alliant, Sequent, and Encore multiprocessors with the aid of the SCHEDULE parallel programming package.

In a related area, Wright and his colleague V. Pereyra studied the solution of the nearly block-diagonal linear systems that arise from the finite difference discretization of two-point boundary value problems. The methods they developed are most suited to computers with vector-processing capabilities, such as the Cray machines. In contrast to traditional methods, the computations are arranged so that the length of the vectors is equal to the order of the chosen discretization, rather than the order of the underlying differential equation (which is usually smaller). The new methods have application to algorithms for seismic ray-tracing, which in turn are used in earthquake location and seismic prospecting.

Pat Eberlein (SUNY at Buffalo) spent two months at Argonne. During the first month, she implemented a general eigenvalue/vector program on the hypercube, using the 32-node machine and experimenting with the 16-node vector machine. For matrices of order 61 and 128, she was able to achieve speedups of 16.8 and 24, respectively, on the 32-node machine. During the second month at Argonne, Eberlein worked on getting a nonsymmetric version of the Jacobi algorithm running on the Alliant. Subsequently, she wrote codes also for the symmetric case. For a matrix having
eigenvalues that are symmetric about the origin and that lie between +2 and -2, the speedup obtained ranged from 4.6 (for \( n = 16 \)) to 7.2 (for \( n = 64 \)).

6.1.2 LAPACK

The National Science Foundation is supporting development of a linear algebra library based on EISPACK and LINPACK. The package, called LAPACK, involves the integration of the two sets of algorithms into a unified library, with a systematic design; incorporation of recent algorithmic improvements; and restructuring of the algorithms to make as much use as possible of the Basic Linear Algebra Subprograms (BLAS). Use of the BLAS is the basis of our approach to achieving efficiency.

Our objective is to make the library highly efficient, or at least tunable to high efficiency, on each machine; to create a user interface that is uniform across machines; and to make the programs widely available.

During the past year, all of the routines that will make up the linear system and linear least-squares portion of the LAPACK package were converted to the Level 2 BLAS, and most have been converted to the Level 3 BLAS as well. A preliminary version of the first part of LAPACK has been released to test sites throughout the country.

6.2 SDI Funding (J. Boyle, W. J. Cody, K. Dritz, P. T. P. Tang, T. Harmer)

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. Three main areas of investigation have been conducted during the past year.

6.2.1 Portability, Libraries, and Ada

The ACM SIGAda Numerics Working Group has completed its development of a proposed standard specification for a generic package of elementary functions for Ada, and the ISO-IEC/JTC1/SC22/WG has recommended the proposal for adoption as an ISO standard. The proposed elementary functions standard seeks to remove a major deterrent to the portability of scientific and engineering applications written in Ada, including those of interest to SDIO—namely, that the elementary functions, when available at all, are provided merely as vendor extensions to ANSI/MIL-STD-1815A-1983 (Ada). The proposal is noteworthy because it goes beyond merely standardizing the names and parameter profiles of the elementary functions; it also includes demanding (but realistic) accuracy requirements for the results of the functions.

Also receiving attention has been the committee's proposed standard specification for a generic package of primitive functions for Ada. This standard supports the construction of accurate, efficient, and portable mathematical software by providing low-level functions for precise manipulation of floating-point values in certain ways (such as scaling by a power of the radix). The standard calls for accuracy that cannot be guaranteed by the Ada model of floating-point arithmetic; each implementor will be expected to provide the required accuracy by techniques that depend on the particular implementation (such as manipulating the exponent and fraction part of floating-point values directly, as bit strings or integers, which simultaneously achieves the objective of efficiency).
Thus, while an implementation of this standard is not by itself portable, applications that use it can be. It will have immediate application in accurate, efficient, and portable implementations of the elementary functions standard, but its usefulness is not limited to that.

The committee is now considering specifications for real and complex vector and matrix packages. The packages will include the fundamental arithmetic operations on each data type (complex scalar, real/complex vectors, and real/complex matrices), as well as mixed-mode operations. They will also include selection, conversion, and composition operations for all types, and complete sets of vector-vector, matrix-vector, and matrix-matrix operations.

In parallel with our contributions to these standard specifications, we have begun to devise a conforming, portable implementation of the proposed elementary functions standard. Four functions have been completed so far: exp, log, log(1 + x), and argument reduction for trigonometric functions. This package is capable of handling instantiation from 6 digits through 33 digits.

The accuracy achieved by our implementation required the development of new methods for testing functions to an unprecedented resolution. The new testing methods have also been implemented portably in Ada. We have used them to test our implementations of log and exp and to compare ours with those supplied as vendor extensions; the results show ours to be more accurate.

\begin{table}[h]
\centering
\caption{Test results for EXP}
\begin{tabular}{|c|c|c|c|c|}
\hline
No. of Test & Single Precision & Double Precision \\
Arguments (×10^3) & Interval & VADS & Tang & VADS & Tang \\
\hline
100 & $I_1$ & 0.500 & 0.501 & 1.640 & 0.523 \\
200 & $I_2$ & 0.500 & 0.501 & 1.644 & 0.521 \\
150 & $I_3$ & 0.500 & 0.501 & 1.683 & 0.521 \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\caption{Test Results for EXP on the Sequent Balance}
\begin{tabular}{|c|c|c|c|c|}
\hline
No. of Test & Single Precision & Double Precision \\
Arguments (×10^3) & Interval & VADS & Tang & VADS & Tang \\
\hline
100 & $I_1$ & 0.500 & 0.501 & 0.501 & 0.511 \\
200 & $I_2$ & 0.500 & 0.501 & 0.730 & 0.511 \\
150 & $I_3$ & 0.500 & 0.501 & 0.780 & 0.510 \\
\hline
\end{tabular}
\end{table}

33
6.2.2 Parallel Algorithm Studies

Recently, we have been testing and timing three algorithms for the assignment problem on the DAP, Encore, Symmetry, and Alliant machines. The three algorithms are the Jonker-Volgenant algorithm, the unscaled Auction algorithm, and the scaled Auction algorithm. The three algorithms have been evaluated with large dense problems in which all assignments are feasible with variations in the maximum cost and size of the problem. Sequential algorithms were implemented on the Alliant and Encore with concurrency and vectorization selected by compiler options. The algorithms for the DAP were rewritten and reorganized by means of an array syntax. The results showed that the Jonker-Volgenant algorithm was superior to the other two algorithms on all machines when the maximum cost was 1000 or more. In general, the results show that the DAP is faster than the other machines (for example, 5 to 50 times faster than the 8-processor Symmetry, depending on the problem size and maximum cost) and in some cases considerably faster than the Alliant on unoptimized Alliant code. However, further experiments are in progress to optimize the Alliant code by means of the array syntax, and to run the same tests on the Connection Machine 2 in Fortran with the array notation.

6.2.3 Code Conversion and Development

An accessibility code implemented in Fortran was successfully run on a wide variety of MIMD machines. A high degree of portability was achieved by using the ACRF monitor macros to parallelize the code. Near linear speedup was achieved for the accessibility computation portion of the code (the initialization portion of the code was not parallelized); on the Balance a speed gain of 22.6 was achieved with 24 processors. The code is believed to be well suited for SIMD machines such as the Connection Machine, which has 16,000 processors (expandable to 64,000) and supports virtual processors well in excess of the number of physical processors. An Ardent Titan port of the code is under way to gain familiarity with this newest supercomputer entry to the ACRF.

A threat assessment code was converted from Pascal to C and parallelized with the ACRF monitor macros for the MIMD machines. Data dependencies precluded a high-level of parallel processing, and a maximum speed-up of about 3 was achieved. We plan to obtain an Ada version of the code for study in a parallel environment.
7 Technology Transfer

For several years Argonne National Laboratory has encouraged interactions between research divisions and industry. This year we have also actively participated in a project to encourage other divisions within Argonne to use advanced computing techniques and facilities for their research.

7.1 Technology Transfer with Industry

7.1.1 Quintus Computer Systems, Inc.

Tim Lindholm of Quintus worked at Argonne for a four-month period from July through October 1988. Quintus is one of the world’s foremost implementors of commercial Prolog systems. Lindholm’s objective in collaborating with ACRF staff was to develop a high-quality, high-performance commercial implementation of an OR-parallel Prolog system on a shared-memory multiprocessor. The engine he designed was extremely satisfactory; the new system surpassed the 15-35 percent improvement hoped for at the start of the project.

7.1.2 NAG

The software tools discussed in Section 5.8 form the core of a commercial software product being developed by the Numerical Algorithms Group, Inc. under a grant from the Illinois Department of Commerce and Community affairs, administered by the Argonne Technology Transfer Center. The tools are being further developed, tested, documented, provided with a friendly user interface, and packaged for commercial distribution. The MCS researcher responsible for the tools, W. R. Cowell, is acting as a consultant to the Numerical Algorithms Group, Inc. The resulting software product will be available in early 1990 under the name VecPar.77.

7.2 Technology Transfer with Other Argonne Divisions

7.2.1 Experimental Breeder Reactor-II Division.

Greg Chisholm of Argonne’s EBR-II has been collaborating with staff members on a joint project called the Full-Authority, Fault-Tolerant, Reactor Control System (FAFTRCS). The primary focus of this collaboration has been the specification and analysis of software, extending earlier work in the formal analysis of systems. Chisholm and his colleagues at MCS successfully generated proofs about claims for system behavior. In particular, they proved that a prototypical reactor control system (which is computer based) is tolerant of single failures. The analysis depends on abstraction of system behavior and subsequent decomposition through multiple levels down to implementation details.

An essential feature of this effort is compatibility with a program verification methodology being developed as part of a larger collaboration between the MCS Division, EBR, and Michigan State University. The methodology is based on Argonne’s ITP theorem prover and TAMPR transformational system, and uses the Floyd-Hoare induction assertion methods. The focus of this work has been to develop a general transformational capability and analysis strategy to aid in analyzing
programs written in a subset of the C language. The ultimate objective is to verify software for potential use in a reactor control system.

An additional aspect of the collaboration between EBR-II and MCS has centered on the parallelization of reactor simulation codes. An outgrowth of work with SDI, this research has included implementation of different models on the ACRF machines and application of Toolpack tools to automatically transform code for portability and analysis.

7.2.2 Computing and Telecommunications Division

M. Minkoff, W. R. Cowell, and J. Rowlan of the MCS Division have been working with Argonne’s Computing and Telecommunications Division to design or adapt techniques and programs to exploit advanced-computer architectures. Cowell’s efforts have focused on the development and use of software tools to analyze and to automatically transform Fortran programs. Minkoff has been coordinating efforts to adapt large applications codes to parallel architectures. Rowlan’s expertise is in the visualization of computational results on the ACRF Ardent workstation. Their work on a combustion modeling project illustrates how scientists can use high-performance computers and graphics supercomputers to analyze large-scale computations. The objective is to use the shared-memory parallel/vector architecture of the Ardent to generate the data from a combustion code, and then to use these data for rendering a sequence of three-dimensional images on the Ardent workstation. To maximize performance, the researchers are using Cowell’s collection of preprocessing tools to automatically exploit loop-based parallelism in the software.
A Permanent Scientific Staff

J. M. Boyle, *Ph.D.*, Northwestern University, 1970
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
J. J. Dongarra, *Ph.D.*, University of New Mexico, 1980
K. W. Dritz, *M.S.*, Massachusetts Institute of Technology, 1967
B. S. Garbow, *M.S.*, University of Chicago, 1952
A. Griewank, *Ph.D.*, Australia National University, 1980
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
M. Minkoff, *Ph.D.*, University of Wisconsin, 1973
J. J. Moré, *Ph.D.*, University of Maryland, 1970
D. C. Sorensen, *Ph.D.*, University of California, San Diego, 1977
L. T. Wos, *Ph.D.*, University of Illinois, 1957

* Left the Laboratory during the reporting period.
B Administrative and Technical Support Staff

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

* Left the Division during the reporting period.
C Professional Activities

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
Member, Working Group 2.5 for Mathematical Software
SIGAda Numerics Working Group

W. R. Cowell  President, W. R. Cowell Consulting, Inc.

J. J. Dongarra* Adjunct Professor, Northern Illinois University
Associate Editor, CACM
Associate Editor, Parallel and Distributed Computing
Associate Editor, ACM Transactions on Mathematical Software
Associate Editor, Supercomputer
Associate Editor, International Journal of Supercomputer Applications
Associate Editor, Journal of Supercomputing
Associate Editor, IMPACT of Computing in Scientific Applications
Chairman, SIAM-SIAG on Supercomputing
Member, Tech. Rev. Group, Office of Advanced Scientific Computing
Member, SIGNUM Board
Member, SIAM Council

K. W. Dritz  Member, ACM SIGAda Numerics Working Group
Member, WG9 (Ada) Numerics Rapporteur Group

B. S. Garbow  Consultant, C. Abaci

H. G. Kaper  Associate Editor, Integral Equations and Operator Theory
Associate Editor, Transport Theory and Statistical Physics
Adjunct Professor, Northern Illinois University

E. Lusk  Adjunct Professor, Northern Illinois University
Associate Editor, Journal of Automated Reasoning
Co-chair, 9th International Conference on Automated Deduction
Program co-chair, North American Conference on Logic Programming

J. N. Lyness  Associate Editor, Mathematics of Computation
Adjunct Professor, Northern Illinois University

M. Minkoff  Adjunct Professor, Northern Illinois University

J. J. Moré  Associate Editor, SIAM Journal on Numerical Analysis
Associate Editor, Numerische Mathematik
Member, SIGNUM Board

R. Overbeek  Co-chair, 9th International Conference on Automated Deduction
Program co-chair, North American Conference on Logic Programming
Visiting Professor, the University of Texas at Austin
D. C. Sorensen*  
Member, Parallel Computing Forum  
Associate Editor, SIAM Journal on Scientific & Statistical Computing  
Associate Editor, ACM Trans. on Math. Software  
Adjunct Professor, University of Illinois at Urbana  
Adjunct Professor, Rice University

R. Stevens  
Book Review Editor for the Journal of Automated Reasoning  
Sequent Users Group SIGPP committee  
Organizer of Encore Users Group

P. T. P. Tang  
Member, SIGAda Numerics Working Group

L. Wos  
Adjunct Professor, University of Illinois at Urbana  
Editor-in-Chief, Journal of Automated Reasoning  
President, Association for Automated Reasoning

* Left the Laboratory during the reporting period.
D Scientific Impact

The following list reflects articles published, reports distributed, and talks presented by Division members since January 1988. 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.

Publications in Books and Refereed Journals


W. R. Cowell and C. P. Thompson, "Tools to Aid in Discovering Parallelism and Localizing Arithmetic in Fortran Programs," Software Practice and Experience, to appear (also MCS-P6-1088)


K. W. Dritz, "Plugging the Holes in the Sieve of Eratosthenes," Ada Letters 9, no. 2 (March/April 1989) 72-77 (also MCS-P4-0888)


F. B. Hanson and A.-M. Wazwaz, "Singular and Turning Point Resonance: Small Perturbations


M. K. Kwong, "Uniqueness of Positive Solutions of $\Delta u - u + u^p = 0$ in $\mathbb{R}^n$," Arch. Rat. Mech. 105 (1989), 243-266


G. K. Leaf, M. Minkoff, and J. C. Diaz, "Nested Factorization Preconditioners for Convective-Diffusion Problems in Three Dimensions," Mathematics for Large-Scale Computing, J. C. Diaz,


Contributions to Conference Proceedings


J. J. Dongarra, D. C. Sorensen, and O. Brewer, “Tools and Methodology for Programming


Preprints (submitted for publication)

To enable researchers to have access to our research results before they appear in the open literature, we have instituted a preprint series. Once a manuscript is published, it is removed from this list and cited in the appropriate section (Journals or Conference Proceedings).


Man Kam Kwong, "Oscillation of First-Order Delay Equations," MCS-P3-0888

Man Kam Kwong and A. Zettl, “Determining the Best Constant for the $2 \times 2$ Matrix Landau Inequality Using MAPLE,” MCS-P7-0988


Man Kam Kwong and A. Zettl, “Norm Inequalities for the Powers of a Matrix,” MCS-P9-0988


W. J. Cody, “Performance Evaluation of Programs Related to the Real Gamma Function,” MCS-P12-0988


C. H. Bischof, “A Parallel QR Factorization Algorithm with Controlled Local Pivoting,” MCS-P21-1088

F. B. Hanson and D. C. Sorensen, “The SCHEDULE Parallel Programming Package with Recycling Job Queues and Iterated Dependency Graphs,” MCS-P22-0189

C. P. Gupta, “A Nonlinear Boundary Value Problem Associated with the Static Equilibrium of an Elastic Beam Supported by Sliding Clamps,” MCS-P24-1188


M. T. Chu, “A Continuous Jacobi-like Approach to the Simultaneous Reduction of Real Matrices,” MCS-P30-1288


S. J. Wright, “Algorithms for Minimization Subject to Bounds,” MCS-P32-1288
W. Winsborough, "Source-Level Transformations for Multiple Specialization of Horn Clauses," MCS-P33-1288

J. N. Lyness, T. Sorevik, and P. Keast, "Notes on Integration and Integer Sublattices," MCS-P34-1288


A. Griewank, "The Equivalence of Strict Convexity and Injectivity of the Gradient in Bounded Level Sets," MCS-P41-0189

J. V. Burke, "On the Identification of Active Constraints II: The Nonconvex Case," MCS-P43-0189

Man Kam Kwong, "On the Kolodner-Coffman Method for the Uniqueness Problem of Emden-Fowler BVP," MCS-P44-1089

S. J. Wright, "Implementing Proximal Point Methods for Linear Programming," MCS-P45-9189


A. Bayliss, B. J. Matkowsky, and M. Minkoff, "Bifurcation and Pattern Formation in Combustion," MCS-P50-0189


G. K. Leaf, M. Minkoff, and J. C. Diaz, "Nested Block Factorization Preconditioners for Convective-Diffusion Problems in Three Dimensions," MCS-P53-0289

P. T. P. Tang, "Table-Driven Implementation of the Logarithm Function in IEEE Floating-Point Arithmetic," MCS-P55-0289

P. Takac, "A Fast Diffusion Equation Which Generates a Monotone Local Semiflow I: Local Existence and Uniqueness," MCS-P57-0289


C. Gupta, "Existence and Uniqueness Theorems for a Fourth Order Boundary Value Problem of Sturm-Liouville Type," MCS-P59-0289

W. J. Cody and L. Stoltz, “The Use of Taylor Series to Test Accuracy of Function Programs,” MCS-P61-0289


J. M. Howie and R. B. McFadden, “Ideals Are Greater on the Left,” MCS-P63-0289

S. J. Wright, “A Parallel Algorithm for Banded Linear Systems,” MCS-P64-0289


O. Brewer, J. Dongarra, and D. Sorensen, “A Graphics Tool to Aid in the Generation of Parallel Fortran Programs,” MCS-P68-0389


M. T. Chu, “Least Squares Approximation by Real Normal Matrices with Specified Spectrum,” MCS-P70-0389

C. Bischof, “Computing the Singular Value Decomposition on a Distributed System of Vector Processors,” MCS-P71-0389

C. P. Gupta, “A Fourth-Order Nonlinear Boundary Value Problem and an Integral Type Sign Condition,” MCS-P72-0489

M. K. Kwong, “A Comparison Result and Elliptic Equations Involving Subcritical Exponents,” MCS-P73-0589

C. P. Gupta, “Nonlinear Second Order System of Neumann Boundary Problems at Resonance,” MCS-P74-0489


I. Foster and S. Taylor, “Strand: A Practical Parallel Programming Language,” MCS-P80-0689


S. Wright, “Convergence of an Inexact Algorithm for Composite Nonsmooth Optimization,” MCS-P83-0689


W. Winsborough, “Path-Dependent Reachability Analysis for Multiple Specialization,” MCS-P86-0689

W. McCune and L. Wos, “The Absence and the Presence of Fixed Point Combinators,” MCS-P87-0689


S. J. Wright, “Solution of Discrete-Time Optimal Control Problems on Parallel Computers,” MCS-P89-0789

W. J. Cody, “ALGORITHM XXX: Functions to Support the IEEE Standard for Binary Floating-Point Arithmetic,” MCS-P90-0789


C. A. Bischof, “A Block QR Factorization Algorithm Using Restricted Pivoting,” MCS-P95-0789

L. Wos, “The Problem of Guaranteeing the Absence of a Complete Set of Reductions,” MCS-P95-0789

C. P. Gupta, “Solvability of a Fourth-Order Fully Quasilinear Boundary Value Problem with Periodic Boundary Conditions,” MCS-P96-0889


I. Foster, “Copy Avoidance through Local Reuse,” MCS-P99-0989

Reports


Technical Memoranda


W. J. Cody, "ELEFUNT Test Results under FORTRAN-PLUS on the Active Memory Technology DAP 510-8," MCS-TM-125 (September 1988)


F. Hanson, "Computational Dynamic Programming for Stochastic Optimal Control on a Vector Multiprocessor," MCS-TM-113 (June 1988)


Oral Presentations

C. H. Bischof, "Incremental Condition Estimation and an Application in a Local Pivoting Scheme" Meeting of the American Mathematical Society, Lawrence, Kansas, October 27-29, 1988


C. H. Bischof, "A Parallel QR Factorization Algorithm Using Local Pivoting," Conf. on Hypercube and Concurrent Computers and Application, Monterey, California, March 6-8, 1989

C. H. Bischof, "Incremental Condition Estimation and Applications," 3rd SIAM Conf. on Optimization, Boston, April 3-5, 1989


J. M. Boyle, "Has Functional Programming Any Application to Numerical Linear Programming?" IFIP WG2.1 (Algor: Programming Languages and Calculi), Chamrousse, France, January 8-13, 1989


M. T. Chu, "Isospectral Flows and Abstract Matrix Factorizations," Colloquium, Western Illinois University, April 20, 1989

M. T. Chu, "Continuous Realization Methods and Their Application," Colloquium, Western Illinois University, April 21, 1989

M. T. Chu, "Continuous Realization Methods and Their Application," Colloquium, Michigan State University, April 26, 1989

54

J. J. Dongarra, "The ACRF and High-Performance Linear Algebra Libraries," Wilmington, Delaware, October 11, 1988

J. J. Dongarra, "Tools for Parallel Algorithm Development," Workshop on Parallel and Distributed Processing, Bonas, France, October 4, 1988


J. J. Dongarra and D. C. Sorensen, Participants in a Workshop on Vector and Parallel Processing, University of Umea, Umea, Sweden, August 29-September 2, 1988


F. B. Hanson, "The Alliant FX/8," University of Illinois at Chicago, UIC Workshop Program on Scientific Supercomputing, January 25, 1988

F. B. Hanson, "Numerical Considerations," University of Illinois, UIC Workshop Program on Scientific Supercomputing, February 17, 1988

F. B. Hanson, "Numerical Libraries," University of Illinois at Chicago, UIC Workshop Program on Scientific Supercomputing, February 19, 1988


F. B. Hanson, "Vectorization, Parallelization and Dynamic Programming," Bethesda, Maryland, Mathematics Research Branch, NIDDK, National Institutes of Health, March 24, 1988


H. G. Kaper, "Nonlinear Diffusion Equations," Department of Mathematics, University of Vienna, Vienna, Austria, June 12, 1989

H. G. Kaper, "Bifurcation Phenomena in Combustion," Department of Mathematics, University of Vienna, Vienna, Austria, June 13, 1989


H. G. Kaper, "Uniqueness of Non-Negative Solutions of a Class of Semilinear Elliptic Equations," Mathematics Department, Technical University, Budapest, Hungary, June 16, 1989


E. Lusk, "Parallel Programming in Logic," University of Maryland, March 1988

E. Lusk, "High Performance Automated Reasoning," University of Texas, April 1988


E. Lusk, "A Three-Day Course in Parallel Computation," Australian National University, July 1988

E. Lusk, "Parallel Programming in Logic," University of Sidney, July 1988

E. Lusk, "Parallel Programming in Logic," University of Wollongong, July 1988

E. Lusk, "Parallel Programming in Logic," Melbourne University, August 1988

E. Lusk, "More Oxen or a Bigger Ox: The Challenge of Parallel Computation in the 90's," Australian National University, August 1988


W. W. McCune, "Indexing and Unification of First-Order Formulas," Northwestern University, March 7, 1989


J. J. Moré, "Trust Region Methods: Motivation and New Results," University of Virginia, Department of Mathematical Sciences, February 1989.


G. W. Pieper, "Writing That Works," invited seminar, Division of Educational Programs, Argonne National Laboratory, August 1989.


R. Stevens, Guest lecture and two-week course on parallel computing, Australia National University, Canberra, Australia, December 1-17, 1988.

E Meetings, Workshops, and Classes

1. Energy Research Supercomputer Users Group Meeting and Associated Workshops — May 9-12, 1988

Argonne hosted the first meeting of the Energy Research Supercomputer Users Group (ERSUG). The objective of the meeting was to foster open discussion between ER supercomputer centers and ERSUG on the status of the computational resources of the centers and on goals for adding and improving resources and removing expensive capabilities no longer needed. Included in the agenda were presentations by representatives from Argonne, Florida State University, and the National Magnetic Fusion Energy Computer Center.


The annual two-day tutorial workshop on automated reasoning was held at Argonne. The workshop included an overview of automated reasoning, a discussion of the growing role of logic programming, and examples of applications in logic circuit design and power plant control.


Jack J. Dongarra organized this second international conference, held in Tromso, Norway. The conference focused on issues in applied research and development, and included presentations on such topics as domain decomposition, issues in reservoir modeling, computational fluid dynamics, and robotics.

4. IFIP TC2/WG2.5 — August 22-26, 1988

B. T. Smith served as co-chairman of the fifth working conference of IFIP, held at Stanford. Two dozen papers were presented by scientists from the Soviet Union, Great Britain, the United States, France, Japan, and Denmark. Topics included effective scientific applications; language design issues; and libraries, environments, and tools.

5. Summer Institute in Parallel Programming — September 6-16, 1988

We hosted a Summer Institute in Parallel Programming to familiarize university researchers with new techniques in high-performance computing. Twenty-five graduate students and post-doctoral researchers were selected out of 80 applicants to participate in the institute, which was sponsored in part by the National Science Foundation and in part by the Department of Energy.

6. ACRF Coming of Age Celebration — September 22, 1988

The Mathematics and Computer Science Division held a celebration and open house to celebrate the Coming of Age of the Advanced Computing Research Facility. Attending were persons who were involved in the early history of the facility, computer vendors who have worked with ACRF staff in the selection of advanced computers, and members of our affiliates program who help faculty and students interested in using the ACRF computers.

7. Midwest Society for Programming Languages and Systems — October 22, 1988

Argonne National Laboratory hosted a one-day workshop whose objective was to identify what languages and systems work is being done in the Midwest. Approximately 50 people attended the meeting, including representatives from universities and research centers.
8. Workshop on Advanced Computational Methods in Biological Sequencing – November 3–5, 1988

The MCS Division hosted a three-day workshop for seven biologists and seven computer scientists to discuss ways in which advanced computer technologies could benefit in genome sequencing projects. The participants agreed that model projects should be established to investigate the feasibility of using logic programming and parallel computers in molecular biology projects.


E. Lusk and R. Overbeek co-chaired the Ninth International CADE conference at Argonne. The conference commemorated the twenty-fifth anniversary of the discovery of the resolution principle, which took place at Argonne in the summer of 1963. For this ninth conference, 52 papers and abstracts of 21 implementations were accepted and published in the proceedings. Six papers were from our division.

10. Classes on Parallel Computing

Argonne conducted 18 classes on parallel computing from January 1988 through August 1989. The attendees, averaging 20-25 persons per class, represented universities, industry, and various research laboratories throughout the country. Until the beginning of 1989, the intent of the classes was to familiarize researchers with the ACRF environment, to offer hands-on experience on the parallel computer systems, and to apply parallel programming to each attendee's area of research. Fortran is the highlighted programming language in these classes. In March 1989, we began to offer an additional class specifically focusing on parallel computing in Ada. Because all validated Ada compilers include standardized tasking features, this new class devoted relatively more time to language issues and less to the variations in parallel architectures and to solutions to the portability problem posed by those variations. Most of the attendees at the four Ada classes conducted to date have been SDI contractor personnel.


The North American Conference on Logic Programming met at Argonne on Thursday and Friday, June 1–2. Following the meeting on Saturday, June 3, was an industrial workshop on the commercial use of logic programming. Among the questions considered were, “Does this technology offer any real benefits to the commercial data processing community?” “What would cause this technology to either be adopted or not be adopted by the DP community?” “What problems are addressed by logic programming technology, and what are the alternative solutions that exist in the marketplace?”

12. Workshop on Advanced Topics in Automated Reasoning and Automated Theorem Proving – August 1–4, 1989

Twenty-eight researchers from the United States and Europe participated in a four-day workshop on automated reasoning held by the MCS Division in early August. The purpose of the workshop was to consider open questions and problems amenable to attack with an automated reasoning program. The workshop was a distinct success: the participants were able to solve an open question in combinatory logic, refine conjectures in Robbins algebra, and outline strategies for extending linked inference rules. The workshop also led to collaborations on such areas as program synthesis and program verification.
Special Term Appointments

A. Bayliss, Northwestern University
B. J. Matkowsky, Northwestern University
B. T. Smith, University of New Mexico
A. Wojcik, Michigan State University

Other Visitors

E. Anderson, University of Illinois
P. Arbenz, ETH - Zurich
F. V. Atkinson, University of Toronto
J. Barlow, Pennsylvania State University
R. Butler, University of North Florida
T. Butler, University of North Florida
C. Y. Chan, University of Southwestern Louisiana
C. Chiu, Florida State University
M. Chu, North Carolina State University
T. Colin, Ecole Normale Superieure de Lyons
J. Crammond, Imperial College
C. Dawson, University of Chicago
S.-W. Du, University of Southwestern Louisiana
P. Eberlein, SUNY at Buffalo
S. Fineberg, Purdue University
I. Foster, Imperial College
C. Gupta, Northern Illinois University
M. Garbey, University of Valenciennes
S. Hammarling, NAG, Inc.
F. Hanson, University of Illinois at Chicago
R. Hempel, GMD, West Germany
L. Henschen, Northwestern University
S. Hoshen, Naperville H.S.
A. Hudson, University of Kentucky
A. Jindal, University of Illinois at Chicago
D. Joerg, Ill. Math. and Science Academy
A. Kadifa, ARCH Development
N. Karonis, Syracuse University
P. Keast, Dalhousie University
M. Knaap, University of Leiden
J. Kohl, Purdue University
T. Lindholm, Quintus Corp.
R. McFadden, Northern Illinois University
P. Mateti, Wright State University
M. D. Medley, Augusta College
M. Miller, David Lipscomb University
W. Newman, University of Wisconsin
L. Ni, Michigan State University
J. Northrup, North Carolina State University
R. Olson, University of Illinois
L. Peletier, University of Leiden
N. Pfugger, University of Chicago
A. Ruhe, Gothenburg University
D. Salane
A. Sameh, University of Illinois
C. P. Schuetz, Siemens AG
V. Senevirathne, Northern Illinois University
M. Shelley, University of Chicago
G. Shroff, Cornell University
P. Skler, Ecole Normale Superieure de Lyons
T. Sorevik, University of Bergen
J. Stephen, Northern Illinois University
D. Stevens, Northwestern University
L. Stoltz, South Dakota State University
P. Takac, Vanderbilt University
S. Tuecke, University of Illinois
C. Thompson, Bergen Scientific Centre
G. Toraldo, University of Naples
H. Tsutsui, Northern Illinois University
T. Weigert, Motorola, Inc.
R. M. Winfrey, ARCH Development
W. Winsborough, University of Chicago
R. Womersley, University of New South Wales
S. Wright, North Carolina State University
S. Wu, Third Institute of Ministry, China
A. Zettl, Northern Illinois University
H. Zhang, University of Texas at Austin

Student Participation Program

V. Belcher
J. Bloomstein
R. Evard
T. Henry
V. Herrate
T. Homulka
P. Lacroute
J. Levin
R. Lingevitch
R. Manning
D. Iller
M. Price
F. Spivey
C. Wirk

Science and Engineering Research

S. Dave
F. Ketabchi
G Seminars

During the period January 1988 through August 1989, 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.

Lionel Ni, Formal Verification of CMOS Circuits, Michigan State University, January 21, 1988

Miguel A. Herrero, Nonlinear Diffusion with Absorption, Complutense University of Madrid, Spain, March 15, 1988

Francisco Bernis, Nonlinear Boundary Value Problems, Complutense University of Madrid, Spain, March 29, 1988

Jeff McFadden, Crystal-Melt Interface Instabilities during Directional Solidification of a Binary Alloy, National Bureau of Standards, March 31, 1988

Alan Bundy, The Use of Proof Plans to Guide Inductive Proofs, University of Edinburgh, June 1, 1988

Davis Wise, Applicative Programming and Matrix Arithmetic, Indiana University, June 3, 1988

Alvin Bayliss, Traveling Cylindrical Waves in Gaseous Combustion, Northwestern University, July 12, 1988

Peter Takac, Equilibrium Solutions of Periodic Nonlinear Parabolic Equations, Vanderbilt University, July 14, 1988

Mariette Knaap, Quasilinear Elliptic Equations with Nearly Critical Growth, University of Leiden, Netherlands, July 19, 1988

Bernard Kawohl, A Family of Torsional Creep Problems, University of Heidelberg, July 21, 1988

L. A. Peletier, Moser's Inequality, University of Leiden, Netherlands, July 29, 1988

M. J. D. Powell, Multivariable Approximation Using Radial Basis Functions, University of Cambridge, England, August 11, 1988

M. Chu, Isospectral Flows and Abstract Matrix Factorizations, MCS Division, Argonne National Laboratory, September 20, 1988

Christopher Beattie, Finite Element Approaches to Bracketing Eigenvalues of Continuous Systems, Virginia Polytechnic Institute, September 23, 1988


M. Chu, Homotopy Method for General Lambda-Matrix Problems, MCS Division, Argonne National Laboratory, October 4, 1988

Peter Arbenz, Computing Eigenvalues of Banded Symmetric Toeplitz Matrices, ETH - Zurich, October 10, 1988

Y. Rodin, Reiman Problems in Soliton Theory, Northwestern University, October 28, 1988

64
Carl Ponder, Parallelism and Algebraic Manipulation Systems, University of California at Berkeley, November 7, 1988

Mike Shelley, Adaptive Pseudo-Spectral Methods in Computational Fluid Dynamics, University of Chicago, November 22, 1988

Ed Veling, Optimal Lower Bounds for the Spectrum of the Schrodinger Equation, National Institute for Public Health, Netherlands, December 14, 1988

Kevin McLeod, Entire Solutions of Semilinear Equations, University of Wisconsin, January 5, 1989

Tien-Yien Li, Homotopy Algorithms for the Symmetric Eigenvalue Problem, Michigan State University, January 17, 1989

S. M. Verduyn Lunel, A Functional Analytic Approach to Delay Equations, Georgia Institute of Technology, January 20, 1989

Vered Rom-Kedar, Transport and Mixing in Two-Dimensional Chaotic Flows, The James Franck Institute, University of Chicago, February 9, 1989

Wai Sun Don, Unsteady Compressible Wake Flow Past a Circular Cylinder with Spectral Methods, Brown University, February 15, 1989

Roderick Wong, Asymptotic Expansions—A Survey of Recent Results, University of Manitoba, February 23, 1989


Patrick J. Rabier, Bifurcation in Rotating Bodies, University of Pittsburgh, March 9, 1989


Frank McCabe, Denotational Graphics, Imperial College, April 5, 1989

Clinton Dawson, Numerical Methods for Flow in Porous Methods, University of Chicago, April 20, 1989

Homer Walker, Quasi-Newton Methods for Underdetermined Nonlinear Equations, University of Utah, April 28, 1989

Lothar Reichel, A Divide and Conquer Method for the Orthogonal Eigenproblem with Application to Signal Processing, Bergen Scientific Centre, May 1, 1989


Corrado Boehm Solving Fixed Point Equations without Fixed Point Operators, University of Rome, May 9, 1989

John G. Cleary A Distributed Backtracking Algorithm and a Proof of Its Completeness, University of Calgary, May 19, 1989


Mark Jones, *Using the Method of Lanczos to Solve the Generalized Eigenvalue Problem*, Duke University, June 8, 1989

Howard Allen Levine, *The Role of Critical Exponents in Blowup Theorems*, Iowa State University, June 8, 1989


Shoshana Kamin, *A Nonlinear Diffusion Equation with Unbounded Initial Data*, Tel Aviv University, August 1, 1989

Paul E. Plassmann, *The QR Factorization of Large Sparse Matrices on a Multiprocessor*, Cornell University, August 17, 1989

Julio Diaz, *Approximate Inverse Preconditionings for Sparse Linear Systems*, University of Tulsa, August 21, 1989


Stefan Fangmeier, *Supercomputers for Scientific Research Applications*, NCSA Scientific Visualization Program - University of Illinois, February 18, 1988

J. C. Browne, *Architectural and Language Independent Parallel Programming*, University of Texas, April 5, 1988


Iain Duff, *The Solution of Large-Scale Least-Squares Problems on Supercomputers*, Harwell Laboratory, May 20, 1988


Oliver A. McBryan, *Promise vs. Performance for Massively Parallel Computers*, University of Colorado, August 19, 1988

Peter Deuflhard, *Concepts of an Adaptive Hierarchical Finite Element Code*, Center for Information Technology - West Germany, September 15, 1988


Alan Huang, *Optical Digital Computers*, AT&T Bell Laboratories, October 6, 1989

Jon A. Solworth, *PARSEQ: A Parallel Programming Language with Sequential Semantics*, University of Illinois at Chicago, October 20, 1988

Craig Upson, *Co-Processing Environments for Interactive Visualization*, Stellar Computer Inc., October 27, 1988


Gerard Meurant, *Practical Use of the Conjugate Method on Vector and Parallel Supercomputers*, Centre d'Etudes de Limeil, November 1, 1988

Willi Schoenauer, *Why I Like Vector Computers*, University of Karlsruhe, West Germany, December 9, 1988


Tim Peierls, *Sparse Partial Pivoting on a Multiprocessor*, AT&T Bell Laboratories, February 10, 1989

David Gelernter, *Parallelism to the People*, Yale University, February 16, 1989


Ilse Ipsen, *Systolic Computation of Uniform Recurrence Equations in One Variable*, Yale University, August 31, 1989
Participants

F. V. Atkinson, University of Toronto, Canada
Alvin Bayliss, Northwestern University and Argonne National Laboratory
Michael Du, University of Southwestern Louisiana
Marc Garbey, Université de Valenciennes, France
Andreas Griewank, Argonne National Laboratory
Hans G. Kaper, Argonne National Laboratory
Maria Knaap, University of Leiden, Netherlands
Man Kam Kwong, Argonne National Laboratory
Michael Minkoff, Argonne National Laboratory
Bert Peletier, University of Leiden, Netherlands
Peter Takac, Vanderbilt University

Seminars

M. Minkoff, Traveling Cylindrical Waves in Gaseous Combustion, July 11
A. Bayliss, Traveling Cylindrical Waves in Gaseous Combustion, July 12
P. Takac, Equilibrium Solutions of Periodic Nonlinear Parabolic Equations, July 14
M. K. Kwong, Existence Results for a Free Boundary Value Problem in Plasma Physics, July 15
M. Knaap, Quasilinear Elliptic Equations with Nearly Critical Growth, July 19
B. Peletier, Moser's Inequality, July 29
Michael Du, A Sturm Comparison Theorem for Delay Equations, August 4
F. V. Atkinson, Asymptotics of Nonlinear Eigenvalue Problems for the Laplacian, August 9
M. Garbey, Nonlinear Stability Analysis for a Combustion Problem, August 10
Previous Reports about the Mathematics and Computer Science Division

*Applied Mathematical Sciences Research at Argonne:*

ANL-80-42 (1980)
ANL-81-40 (1981)
ANL-82-26 (1982)
ANL-83-31 (1983)
ANL-84-32 (1984)

*Research in Mathematics and Computer Science at Argonne:*

ANL-85-24 (1985)
ANL-86-35 (1986)
ANL-88-18 (1988)