RESEARCH IN MATHEMATICS AND COMPUTER SCIENCE AT ARGONNE

April 1, 1985 - June 30, 1986

Mathematics and Computer Science Division

Paul C. Messina, Division Director

edited by

Gail W. Pieper

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.
This report reviews the research activities in the Mathematics and Computer Science Division at Argonne National Laboratory for the period April 1, 1985, through June 30, 1986. The body of the report gives a brief look at the MCS staff and the research facilities, and discusses various projects carried out in four major areas of research: advanced computing, applied analysis, computational mathematics, and software methodology. Information on division staff, visitors, workshops, and seminars is found in the appendixes.
## CONTENTS

### HIGHLIGHTS 1

### MCS STAFF 3

### USER FACILITIES 6

  A. Advanced Computing Research Facility 6  
  B. Research Computing Facility 7

### ADVANCED COMPUTING RESEARCH 8

  A. Algorithms and Software 8  
  B. Parallel Programming Methodologies 8  
  C. Programming Languages 9  
  D. Advanced Computer Architectures 10

### APPLIED ANALYSIS 12

  A. Qualitative Analysis 12
    1. Sturm-Liouville Operators 12  
    2. Bifurcation Phenomena 13  
    3. Functional Inequalities 14  
  B. Quantitative Analysis 14
    1. Fluid Flow Problems 14  
    2. Sturm-Liouville Eigenvalue Problems 16  
  C. Constructive Analysis 16

### COMPUTATIONAL MATHEMATICS 17

  A. Optimization 17
    1. Advanced Optimization Techniques 17  
    2. Optimization in Modeling 18
The past year has been a busy one for the Mathematics and Computer Science (MCS) Division. We have collaborated with over fifty visitors (at times more than doubling our staff), hosted ten workshops, and sponsored almost a hundred seminars on mathematics, computer science, and high-performance computing.

It has also been a rewarding time. Two of our projects — in mathematical software (specifically EISPACK and LINPACK) and in automated reasoning — were cited in the 1986 report of DOE Basic Energy Sciences accomplishments. Other highlights include the following:

1. The Warren Abstract Machine for logic programming was extended to exploit the availability of multiprocessing systems. Significant speedups were obtained on the HEP, the Sequent Balance 8000, and the Encore Multimax.

2. The pure Lisp specification of the TAMPR transformer was transformed to parallel Fortran and successfully implemented on several multiprocessors. Speedups were excellent: 13.6 on a Denelcor HEP, 11.5 using 16 processors on the Encore Multimax, and 6.6 using 8 processors on the Sequent Balance 8000.

3. A representation scheme based on Petri nets was used to obtain proof of fault tolerance for the basic transfer commands of the Draper Fault-Tolerant Processor. The scheme is now being extended to the register and system levels.

4. Superior performance on both sequential and parallel machines was obtained with a new algorithm for solving the symmetric tridiagonal eigenvalue problem. The new algorithm is based on a divide-and-conquer scheme modified to exploit parallelism.

5. Studies of lock contention on multiprocessors have yielded new insights about parallel programs. Based on these experiments, new strategies have been devised that sharply improve the performance of parallel programs.

6. A long-sought inference rule, called negative hyperparamodulation, was formulated for handling inequalities. Preliminary experiments indicate that the new rule yields proofs for various simple theorems quickly and easily.

7. A conjugate-gradient algorithm was developed that quickly identifies the binding set on non-degenerate problems. Thus, the algorithm provides an excellent means for solving large-scale quadratic programming problems subject to bound constraints.

8. A several-year investigation of systems of Sturm-Liouville equations culminated in a general
proof of the oscillatory properties of $n$-dimensional systems at infinity.

Many of these achievements reflect our increasing involvement in advanced computing. Under special DOE funding, we have expanded our Advanced Computing Research Facility to include four new multiprocessors, and we have been actively investigating innovative methods that enable these machines to be used effectively.

Complementing this research are strong programs in applied analysis, computational mathematics, and software methodology. Our approach involves the formulation of new analytical and numerical methods, the design of computational algorithms, their implementation in computer programs, and their application in practical problems.

Our investigations continue to emphasize a mix of theory and application. For example, we are utilizing automated reasoning techniques to model properties of a fault-tolerant computer system considered for use in one of Argonne's experimental reactors. Other activities that blend theory and application include modeling kinetics of fast electrode reactions, and developing computational methods for solving chemical equilibrium problems.

Many of these activities arise as collaborative projects with other Argonne divisions and with research institutions throughout the world. Recent investigations include study of fluid flow problems (with the Components Technology Division), design of a database for plant control (with the Hanford Engineering Development Laboratory), algorithm development for advanced computers (with the University of Illinois), and investigation of lattice rules for high-dimensional quadrature (with the University of New South Wales).

This report highlights the activities in the Mathematics and Computer Science Division from April 1, 1985, through June 30, 1986. First we give a brief look at our staff and at our research facilities. Then we review specific projects in four main areas: advanced computing research, applied analysis, computational mathematics, and software methodology. A complete list of Division members, their publications, and their professional activities is provided in the appendixes. Also included is a list of people who visited Argonne to conduct seminars, participate in workshops, or collaborate on special projects.
MCS STAFF

The success of our research is directly attributable to the energy and expertise of our scientific staff. Here we give a brief look at our permanent staff members—their past achievements and their current research.

James M. Boyle. Active for several years in the area of program transformation, Jim’s 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. In addition to working on a parallel version of TAMPR, Jim’s current interests include abstract programming and performance measurements of parallel programs.

William J. Cody, Jr. Jim was instrumental in establishing the IEEE standard for floating-point arithmetic, now implemented in the 8087 chip. He also developed much of FUNPACK, a collection of special function subroutines, and wrote a book entitled Software Manual for the Elementary Functions. Jim is currently preparing special function routines and test programs that combine high quality and transportability — features often lacking in today’s software.

Wayne R. Cowell. Wayne 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. Jack’s primary research interests are devising linear algebra algorithms and developing portable algorithms for high-performance computers. He has been involved in the LINPACK and EISPACK projects and is co-designer of the second-order BLAS, which exploit the special features of vector architectures.

Kenneth W. Dritz. Ken has assisted his colleagues in several different projects, including TAMPR and Toolpack. He has also contributed to the development of PL/I and now continues his interest in programming languages by monitoring developments in Ada. Recently, he launched a study of performance measurement of parallel programs that is providing significant insights about parallelism, lock contention, and potential speedups on multiprocessors.

John R. Gabriel. Our "human engineering" expert, John has long been concerned with factors affecting computer operations. He has studied the role of expert display systems in nuclear plant, and is currently exploring the use of PERT networks in critical path analysis.

Burton S. Garbow. Best known for his pioneering work on EISPACK, Burt also collaborated with his colleagues on the development of MINPACK, Toolpack, and the Toeplitz package. He is currently involved in two areas of research: programming environments and quadrature software.
Kenneth E. Hillstrom. An active participant in the development of MINPACK and JAKEF, Ken has recently become involved also in performance modelling of parallel computers. In particular, he has devised three-dimensional surface representations to illustrate speedup as a function of the number of tasks and processes.


Gary K. Leaf. Gary's work on the numerical solution of partial differential equations led to the development of DISPL, a package widely used by scientists and engineers throughout the United States. Gary's work continues to reflect a mix of theory and application; for example, he is studying a new technique for modeling electrode kinetics effects. He is also analyzing problems in fluid flow and materials science.

Ewing L. Lusk. One of the implementors of the reasoning programs LMA and ITP, Rusty has been actively involved in automated reasoning since he joined the Division several years ago. A recent accomplishment was the development of a new parallel programming approach that implements synchronization through monitors written as macros. At present he is working on parallel logic programming as the foundation for the next generation of automated reasoning systems.

James N. Lyness. James' research involves highly theoretical work on series acceleration and the use of lattice rules for high-dimensional quadrature. Author of a manual on quadrature software, he is also concerned with disseminating software for practical applications.

Paul C. Messina. As Division Director, Paul keeps track not only of our current research projects but also of proposed new areas of investigation. He is particularly concerned that high-performance computers be available to Argonne scientists, and thus is an advisor on the Energy Research Supercomputer Facility and a member of Argonne's Task Group for Supercomputer Acquisition Strategy. He is also chairman of the DOE Advanced Computing Committee Language Working Group.

Michael Minkoff. One of the chief collaborators in the design of DISPL, Mike continues to investigate new techniques for solving multidimensional PDEs. He is also the primary investigator for optimization problems involving special structure (e.g., chemical equilibrium problems), and has initiated a new study of performance modeling of parallel computers.

Jorge J. Moré. Jorge's work in optimization led to 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. Experienced in automated deduction systems, Ross helped develop the LMA/ITP system that is widely used in the United States today. He is currently involved in developing a parallel Warren Abstract Machine for logic programming, which promises to be useful for the design of extremely powerful new inference engines.
Gail W. Pieper. Our technical editor, Gail prepares our brochures and proposals, edits our manuscripts, and writes our public relations material. She also teaches technical editing at a local college and has given short courses on grammar and technical writing at Argonne and other research centers.

Eugene Rackow. Gene is responsible for maintaining the software and hardware for the ACRF computers and the Division's VAX 11/780. Working with the computer vendors, he resolves problems that may arise with the new systems. Gene also answers technical questions and helps set up new accounts for ACRF computer users.

Brian T. Smith. Brian's contributions have ranged from work on EISPACK to studies of Jordan algebra. His current efforts focus on the use of automated reasoning in modeling the properties of a fault-tolerant computer system, and language studies to ensure that programming languages (such as Ada and Fortran) meet the changing needs of their users.

Danny C. Sorensen. Danny's primary concern has been algorithms that achieve both portability and high performance. Toward this goal he and his colleagues have designed new algorithms and restructured existing ones for parallel computers. Danny is also working with researchers at the University of Illinois on applications such as circuit simulation.

Rick Stevens. The newest member of our staff, Rick oversees the daily operations of the Advanced Computing Research Facility. His particular concerns are documentation and systems utilities. Rick also conducts performance measurements of the new multiprocessors.

Lawrence T. Wos. Winner of the first American Mathematical Society prize for current achievements in automated theorem proving, Larry 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. Co-author of a book entitled Automated Reasoning: Introduction and Applications, he is currently writing a companion volume discussing open research problems in the field.
USER FACILITIES

To conduct the experimentation vital to much of our mathematics and computer science research, the Division has established two major facilities: the Advanced Computing Research Facility and the Research Computing Facility.

A. Advanced Computing Research Facility

James M. Boyle, Jack J. Dongarra, Ewing L. Lusk, Ross Overbeek, Eugene Rackow, Danny C. Sorensen, and Rick Stevens

The Advanced Computing Research Facility (ACRF) was established in 1984 in recognition of the role that parallel computer architectures will play in the future of scientific computing. The first computer acquired was a Denelcor HEP computer with one process execution module (PEM). A multiple-instruction multiple-data stream (MIMD) machine, the HEP proved effective for a wide variety of applications; moreover, it has stimulated additional research on other MIMD systems. We also installed an eight-processor Lemur computer, a locally designed and built parallel computer with eight processors sharing memory. The machine has run demonstration programs verifying full parallelism and synchronization.

More recently, we installed four commercial multiprocessing computers: an Alliant FX/8 system with 8 vector processors sharing 32 megabytes of memory, an Encore Multimax system with 20 processors sharing 20 megabytes of memory, a Sequent Balance 21000 system with 24 processors sharing 16 megabytes of memory, and an Intel iPSC five-dimensional hypercube system with 32 nodes each having one-half megabyte of memory. These experimental computers are linked to each other and to the dual-processor VAX 11/780 system in the Division. The VAX is a host on the Arpanet/Milnet and Tymnet networks and is connected by a local network to Argonne's central computing facility, which in turn is connected to the MFEnet.

By operating these computers, we will be able to address many open questions regarding the use of each type (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 the next.
B. Research Computing Facility

James M. Boyle, Eugene Rackow, and Rick Stevens

The Research Computing Facility (RCF) is based on a dual-processor VAX minicomputer. The facility is used by MCS Division staff to conduct the experimentation that is vital to much of our mathematics and computer science research. With the establishment of the Advanced Computing Research Facility, the RCF has taken on an additional role: that of front-ending the experimental computers in the ACRF.

Over the past year and a half, we have added new disk drives, a memory controller, additional memory, and the Berkeley 4.2BSD UNIX operating system. This new hardware and software have provided a more responsive computing environment.

More recently, we have begun to expand the RCF to include several Sun-3 workstations linked to each other and the VAX with an Ethernet local network. We also enhanced our existing Ridge-32 workstation with additional memory, network software, and improved compilers.
ADVANCED COMPUTING RESEARCH

The advanced computing research program, focusing on parallel architecture, aims to create portable algorithms, software, and programming techniques for both numeric and reasoning tasks. The following sections highlight the advanced scientific computing research in the ACRF, which is divided into four main areas: algorithms and software, parallel programming methodologies, programming languages, and advanced computer architectures.

A. Algorithms and Software

Jack J. Dongarra and Danny C. Sorensen

A major goal of our research is to create algorithms and software that achieve high performance and portability on advanced computer architectures. Part of our efforts have focused on designing new parallel algorithms. For example, one significant achievement during this past year was creating an algorithm for solving the symmetric tridiagonal eigenvalue problem in a parallel setting. Not only is this algorithm excellent on parallel architectures, but its performance is superior to the standard algorithm on sequential machines.

In addition to designing new parallel algorithms, we analyzed the most frequently used and time-consuming algorithms relating to eigenvalue problems and linear equations for dense matrices, then reorganized them to utilize matrix-vector operations. The results were impressive: we achieved super-vector speeds on a CRAY X-MP and an Alliant FX/1 and, with minimal reprogramming, have been able to port the algorithms to diverse parallel processors such as the Sequent Balance 8000, the Encore Multimax, and the Alliant FX/8.

We also restructured algorithms for dealing with banded systems of linear equations, then implemented them successfully on the CRAY X-MP and the Alliant FX/8.

B. Parallel Programming Methodologies

James M. Boyle, Jack J. Dongarra, Kenneth W. Dritz, Kenneth E. Hillstrom, Ewing L. Lusk, Michael Minkoff, Ross A. Overbeek, and Danny C. Sorensen

Closely associated with our work on algorithms and software is research on parallel programming methodologies.

We implemented the transformer component of the TAMPR automated program transformation system on several parallel computers by automatically transforming the pure Lisp specification to parallel Fortran. The parallel version of the transformer was highly successful—it achieved a speedup of 12.5 with 16 processes for a real application. We approached the problem by first defining a sequence of language levels, or models of computation, leading from pure Lisp to parallel Fortran. We chose "parallel Fortran with an unbounded number of processes" as the central level. Although having an unbounded number of processes is unrealistic in terms of real machines, this model is useful for proving that pure Lisp programs executed in parallel do not "deadlock." We then developed program transformations that convert the pure Lisp program to this level in several stages. Next, to get an executable parallel program, we wrote additional transformations that implement the unbounded-number-of-processes model in terms of a process queue and a finite, fixed number of "server" processes. We
applied these transformations to the TAMPR transformation interpreter, a large list-processing program in which the available parallelism, and hence the speedup, depends on the input data. The resulting Fortran program was run on a Denelcor HEP. For data having minimal parallelism, we obtained speedups of 2.4, while for data permitting high parallelism, we obtained speedups of 12.8. We also implemented the parallel version of the TAMPR transformer on the Encore Multimax and Sequent Balance 8000 computers. Only two transformations had to be modified to produce these implementations. Again, good speedups were achieved: 6.6 for 8 processors, 11.5 for 16 processors. We are currently investigating alternative implementations that permit larger-grained parallelism.

Another area investigated was the dynamic allocation of resources to a library on a parallel computer. In 1985, we developed a package (called SCHEDULE) of a Fortran-callable subroutine that aids in programming explicitly parallel algorithms in Fortran. One significant advantage of SCHEDULE is that no machine-dependent statements or extensions are required in the user's code. SCHEDULE has been successfully implemented on the HEP, VAX 11/780, Alliant FX/8, CRAY-2, and CRAY X-MP.

We also developed a monitors/macros approach that implements synchronization through monitors written as macros, which are in turn recoded for each machine. We demonstrated the portability of the approach by successfully moving to the Lemur a number of our algorithms developed on the HEP. During the past year, we implemented Fortran and C versions of the basic monitors/macros package for the Encore Multimax, the Sequent Balance 8000, and the Alliant FX/8 machines, as well as the CRAY-2.

In a new study of the behavior of parallel programs, we developed a measurement technique that requires only one run, instead of the usual two, to ascertain speedup. The technique has the additional virtue of reporting, in the same run, the speedup lost due to contention for locks and lack of parallel work. In a related study, we determined the percentage of time that various numbers of processes had work to do. The insight gained from these studies enabled us to appreciate better the influence of algorithm design on achievable parallelism and to devise strategies for reducing synchronization bottlenecks and improving speedup.

We also developed an approach using mathematical modeling, computer simulation, and graphical methods to analyze the performance of machines that achieve parallelism either with multiple processors or with an execution pipeline. We showed that for a certain set of independent tasks a finite-queue model adequately characterizes the performance of a Denelcor HEP and that an infinite-queue model describes the performance of an Alliant FX/8. We are now attempting to extend these results to more general problems and other computer architectures.

C. Programming Languages

Kenneth W. Dritz, Ewing L. Lusk, Paul C. Messina, Ross A. Overbeek, and Brian T. Smith

We designed and implemented a parallel version of the Warren Abstract Machine to study logic programming as a language suitable for parallel computers. We investigated several dialects, primarily through extended visits by researchers from the groups at the Weizmann Institute (Concurrent Prolog), the University of Lisbon (Delta-Prolog), and Imperial College, London (Parlog). As part of this effort, the MCS Division held a workshop on the Warren Abstract Machine, which brought together the primary implementors of high-performance logic-programming systems.
We also began investigating the use of logic programming as a language for controlling parallelism in numerical programs whose floating-point calculations are expressed in C. Preliminary experiments on the Encore Multimax indicate that this mixed-language approach is feasible; however, the large level of granularity this approach requires may be a problem.

Another language being analyzed for programming multiprocessor systems is Ada, which is available on the Sequent Balance 8000. Unlike most other programming languages, Ada has built-in, high-level features for synchronization that make it especially attractive for multiprocessing.

D. Advanced Computer Architectures

William J. Cody, Jr., Wayne R. Cowell, Jack J. Dongarra, Burton S. Garbow, and Danny C. Sorensen

We have been studying various advanced computers to evaluate their software environments and to gain an understanding of their performance potential.

One study analyzed the results of running MACHAR and the ELEFUNT suite of transportable Fortran test programs (Reference 1) on the Encore Multimax and the Alliant. Overall, the library proved to be accurate, but a weakness in error handling was evident. As a result of this work, Alliant Computer Systems Corporation and Encore Computer Corporation have agreed to modify their software libraries.

Another project focuses on developing a Fortran-oriented programming environment comprising Unix, Toolpack, and graphics software, running on a workstation with graphics capability, and accessing one or more advanced computers. The initial prototype involved a VAX connected to advanced-architecture machines; we are now working on a more realistic prototype using a Sun workstation connected to the machines in the ACRF.

In collaboration with scientists at the University of Illinois Center for Supercomputer Research and Development, we have been involved in the CEDAR project. Our main role in the CEDAR project, aimed at constructing a multiprocessor for large-scale scientific computation, is to devise numerical algorithms and mathematical software for this architecture. We have been studying various application programs for circuit simulation, computational fluid dynamics, and optimization in structural analysis.

We have also begun an examination of theoretical models for MIMD computer evaluation. One model views a computation as a set of interdependent demands for resources, which can be represented by a PERT chart. For example, if we wish to compile programs P1, P2, and P3 and link-edit the resulting object modules to an executable file E, then we have the following diagram:
Our intention is to use this model to evaluate parallel systems. Specifically, if the execution time for arcs can be calculated, we should be able to identify those resources (hardware and software) where changes could improve the system performance.

Finally, to assimilate the available information on commercial and experimental high-performance computer systems, we prepared an extensive survey of advanced-architecture machines. The report has been widely distributed throughout the United States and Europe.
APPLIED ANALYSIS

Applied analysis research at Argonne involves the application of analytical and numerical techniques to problems in the natural and engineering sciences. Major efforts focus on spectral analysis of differential operators and bifurcation and stability analysis of nonlinear phenomena in combustion. In addition, research continues on modeling and analysis of certain fluid flow and materials science problems.

A. Qualitative Analysis

Hans G. Kaper, Gary K. Leaf, Bernard J. Matkowsky*, Man Kam Kwong†, and Anton Zettl†

Qualitative Analysis refers to the analysis of classes of equations that share certain formal characteristics. The objective is to obtain information about the existence of solutions, their uniqueness, and their properties. This information, in turn, provides guidelines for the quantitative solution of a given problem on a computer. Emphasis is placed on spectral analysis of Sturm-Liouville operators, bifurcation phenomena in combustion, and functional inequalities.

1. Sturm-Liouville Operators

Hans G. Kaper, Man Kam Kwong†, Clausine van Winter,** and Anton Zettl†

The spectral analysis of Sturm-Liouville operators is important for the solution of boundary-value problems in physics and engineering. Our goal is to determine the spectral properties of such operators in terms of the coefficient and boundary conditions, and to study the oscillatory behavior of their eigenfunctions.

During the past year we completed our investigation of the oscillatory properties of solutions of systems of Sturm-Liouville equations. The object of this investigation was the differential equation $y''+Q(t)y=0$ for the $n$-dimensional vector-valued function $y$ on a semi-infinite domain; the function $Q$ is continuous, and its values are real symmetric matrices of order $n$. It had been known for some time that oscillatory behavior of the solution at infinity results if the largest eigenvalue of the once-integrated matrix $Q$ tends to infinity with $t$, provided the trace of $Q(t)$ satisfies a certain growth condition. The conjecture was that such a growth condition might not be necessary for oscillatory behavior at infinity. In 1985 we showed that this is indeed the case when the system is two dimensional. Subsequently we proved a more general result, namely, that $n$-dimensional systems are oscillatory at infinity if at least $n-1$ eigenvalues of the once-integrated matrix $Q$ tend to infinity with $t$. This year our investigations culminated in a proof of the statement that the solution of the $n$-dimensional system is oscillatory at infinity if the largest eigenvalue of the once-integrated matrix $Q$ is sufficiently large on a sufficiently large set of $t$-values. The statement not only confirms, but strengthens the original conjecture.

* Northwestern University
† Northern Illinois University
**University of Kentucky
We began several investigations focusing on the Titchmarsh-Weyl $m$-coefficient which, together with the spectral function, occupies a central place in the theory of singular Sturm-Liouville eigenvalue problems. The coefficient is an analytic function of the spectral parameter in the upper and lower half of the complex plane. Its asymptotic behavior provides information about the eigenvalues and eigenfunctions. It also relates to the theory of best constants in norm inequalities of Sobolev type for the differential operator.

Another investigation focused on the Kuramoto-Sivashinsky equation, a nonlinear partial differential equation that has been proposed to describe the evolution of flame fronts. Upon separation of variables, the equation defines an eigenvalue problem for a fourth-order differential expression; the linearized form of this expression is used to study flame front stability. In the original formulation, and in most subsequent generalizations, the differential expression is formally self-adjoint. However, a recent generalization by Sivashinsky, Law, and Joulin for flames in stagnation-point flow led to a fourth-order differential expression that was not formally self-adjoint, the non-selfadjoint part representing transverse convection in the physical problem. In collaboration with B. Schultz (University of Essen, W. Germany; temporarily at Northern Illinois University) we are studying the spectral properties of the differential operator defined by this fourth-order expression.

Additionally, we are investigating multiparticle Schroedinger operators with dilation-analytic interactions. Such operators give rise to a scattering theory with dilation-analytic wave and scattering operators. Preliminary results indicate that the dilation-analytic and customary scattering theories are congruent. The transformation relating them determines a family of projections onto invariant subspaces of the self-adjoint Schroedinger operator. These projections are not self-adjoint; they determine invariant parts of the Schroedinger operator that have simple spectra but are not normal. One of these parts provides fundamental insight into the optical model of nuclear physics.

2. Bifurcation Phenomena

*Hans G. Kaper, Gary K. Leaf, and B. J. Matkowsky*

The stability characteristics of flames offer challenging areas of investigation in applied analysis. Our research was motivated by work by Matkowsky and Sivashinsky. Several years ago, Matkowsky and Sivashinsky derived a model of adiabatic flame propagation under the assumption of weak, but non-zero, thermal expansion. The model incorporates a Boussinesq-like coupling between the hydrodynamic equations and the temperature and enthalpy equations through the gravitational forcing term. Matkowsky and Sivashinsky later extended the model to include the effects of volumetric heat loss and used it for a linear stability analysis of non-adiabatic plane flames. They found that heat loss has a decisive influence on stability: downward propagating planar flames that are adiabatically stable may become unstable when heat loss is taken into account.

During the past year we extended the work of Matkowsky and Sivashinsky and completed a nonlinear stability analysis of non-adiabatic plane flames. We found that non-adiabatic plane flames exist only if the heat loss is less than a critical value. We obtained a nonlinear evolution equation for the flame front and showed that the effects of volumetric heat loss can be summarized in a scale factor multiplying the gravitational constant. Thus, in the context of the present model, the stability and

*Northwestern University*
bifurcation properties of non-adiabatic plane flames can be found from those for adiabatic plane flames upon substitution of the appropriately scaled value of the gravitational constant.

We also collaborated with S. B. Margolis (Sandia Laboratories, Livermore) in a study of combustion phenomena in solid fuels. The propagation of combustion fronts in such fuels is generally accompanied by conductive and radiative heat losses. A common way to account for these heat losses is to allow for a volumetric heat loss in the temperature equation. We constructed a model of solid fuel combustion that included the effects of volumetric heat loss and analyzed its ability to support steady planar combustion fronts. We found that, as in the case of gaseous combustion, the burning velocity is a multivalued function of the heat loss parameter, with a reversed C-shaped response curve. The critical value of the heat loss parameter (above which a steady, planar, self-propagating reaction cannot sustain itself) corresponds to an extinction limit.

3. Functional Inequalities

Hans G. Kaper and A. Zettl†

During the past several years our investigations of norm inequalities have concentrated on the operator of differentiation. This year we extended our work to include difference operators. In collaboration with B. Spellman (undergraduate student, Yale University), we designed a procedure to find the best possible constants \( C(n,k) \) in norm inequalities of Sobolev type for the difference operator on the Banach space of bi-infinite sequences endowed with the uniform norm topology. The inequalities provide estimates of the \( k \)-th power of the operator in terms of its \( n \)-th power (\( k \) less than \( n \)). We computed the values of \( C(n,k) \) for all pairs \( (n,k) \), for \( n \) up to and including 6.

B. Quantitative Analysis

Hans G. Kaper, Gary K. Leaf, and Anton Zettl†

Quantitative Analysis refers to the development and analysis of methods for finding approximate solutions to scientific and engineering problems. Two areas are pursued: fluid flow and Sturm-Liouville problems.

1. Fluid Flow Problems

Gary K. Leaf

In collaboration with C. Thompson (Harwell, England) and S. P. Vanka (Components Technology Division), we investigated multigrid solution procedures for the steady-state Navier Stokes equations for the problem of natural convection in a two-dimensional square cavity subject to differential side heating. The equations are differenced with a hybrid finite-difference scheme that switches from central differences to donor cell differences, depending on the local cell Reynolds number. This hybrid procedure accounts for the buoyancy effect, which manifests itself through the coupling of the temperature field to one of the momentum equations. In addition, it accounts for the coupling between the velocity field and the energy equation. On the coarsest grid, the equations are solved with a full Newton-Raphson procedure, where a direct solution technique is used. We developed the numerical techniques, implemented the algorithms, and tested the resulting program. We are currently developing

†Northern Illinois University
contour graphics, such as the one shown below, to help us analyze the data.

We are also extending the investigation to focus on the capability and efficiency of our multigrid scheme for flows with high Rayleigh numbers. The numerical study is being conducted on the CRAY at the National Magnetic Fusion Energy Computing Center. Preliminary studies have demonstrated that we can achieve solutions for flows with Rayleigh numbers up to $10^6$. 
2. **Sturm-Liouville Eigenvalue Problems**  
  
  *Hans G. Kaper, Charles T. Fulton,* and Anton Zett†
  
  One of the most common problems of mathematical physics is the solution of the singular Sturm-Liouville boundary value problem on the half-line. Depending on the potential function, the problem may have some discrete negative eigenvalues together with a continuous spectrum, or a purely continuous spectrum. In 1984 we began a research effort aimed at developing a suite of algorithms capable of handling the majority of physically interesting problems of quantum mechanics, as far as computing the so-called spectral function is concerned. We have now developed a code for computing the spectral density function for a class of problems that are regular at the left endpoint and singular at the right endpoint. The basic algorithm can handle problems with standard boundary conditions, as well as problems where the eigenvalue parameter occurs linearly in the boundary condition. The main part of the code uses a shooting algorithm to refine the initial guesses that are provided by a finite element matrix method. For the higher eigenvalues, the code incorporates an automatic switch-over to asymptotic formulae for the eigenvalues and norms of the eigenfunctions. We made several successful test runs, including one for the Bessel equation of order zero. We also began preparing a users' guide to the code.

**C. Constructive Analysis**  
  
  *Gary Leaf*
  
  Constructive Analysis refers to the development of mathematical models for describing and analyzing physical phenomena. Current work is carried on in collaboration with N. Q. Lam of the Materials Science and Technology Division.

  We are developing a mathematical model that describes the dynamic redistribution of implanted solute atoms in a metal substrate during ion implantation at high temperatures. Our early model included the effects of segregation from differential diffusion. During 1985 we extended the mathematical model to account for competition between the rates of ion deposition and sputtering. The inclusion of this phenomenon led to the formulation of a free boundary problem. We immobilized the free boundary by an appropriate coordinate transformation. The resulting mathematical model consisted of a system of five coupled, nonlinear, parabolic partial differential equations, supplemented by a set of initial boundary conditions, as well as a global (integral type) constraint; the latter was recast as a nonlinear boundary condition for one of the species.

  We have begun extending this work, developing a new model for the creation of point defects in, for example, metal alloys under the influence of a medium-energy proton beam. To date, the governing equations for this model have been developed.

  *Florida Institute of Technology  
† Northern Illinois University
Computational mathematics research at Argonne involves 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. Efforts focus on large-scale and linearly constrained optimization, high-quality software and test programs for special functions, quadrature algorithms and techniques, and methods for solving partial differential equations.

A. Optimization

Burton S. Garbow, Kenneth E. Hillstrom, Michael Minkoff, Jorge J. Moré, and Danny C. Sorensen

Optimization plays a major role in research activities at Argonne. One part of our studies is concerned with the development of new algorithms for general optimization problems; these algorithms can then be implemented in our MINPACK collection of high-quality optimization software. Another part of our studies focuses on specially structured optimization problems that arise in specific energy applications problems.

1. Advanced Optimization Techniques

Burton S. Garbow, Kenneth E. Hillstrom, Jorge J. Moré, and Danny C. Sorensen

Optimization research currently focuses on large-scale optimization, linearly constrained optimization, and parallelism in optimization.

We began investigating the convergence properties of projected gradient methods for the solution of optimization problems subject to convex constraints. The initial work, done in cooperation with P. Calamai (Waterloo University), considered linear constraints, but other convex constraints (such as ellipsoidal constraints) are also of interest. At present, linearly constrained optimization problems are usually solved with active set strategies. These strategies generally restrict the change in the dimension of the working subspace by dropping or adding only one constraint at each iteration. Thus, for example, if there are \( k \) constraints active at the solution but the starting point is in the interior of the feasible set, then the method will require at least \( k \) iterations to converge. This shortcoming does not apply to methods based on projected gradients; they can add or drop many constraints at any iteration.

We have shown that the convergence results for a projected gradient method are quite strong. In particular, we have demonstrated that any limit point of the iterates is a stationary point of the problem and that the sequence of projected gradients converges to zero. Moreover, if the method is applied to a linearly constrained problem, then the binding set is identified in a finite number of iterations, provided the stationary point is nondegenerate.

We extended these results to an algorithm of the conjugate-gradient type. We have shown that if the function is quadratic, then the algorithm terminates in a finite number of iterations on nondegenerate problems. Numerical results obtained at the National Magnetic Energy Fusion Computing Center have also shown that on nondegenerate problems the binding set is identified quickly. Thus, an algorithm based on these ideas provides an excellent means for solving large-scale quadratic programming
problems subject to bound constraints.

We also investigated more efficient methods for solving large-scale systems of nonlinear equations. In these problems it is usually possible to compute bounds on the elements of the Jacobian matrix; this is certainly the case for problems that arise from discretizations of differential equations. We developed a quasi-Newton method that preserves bounds on the elements of the Jacobian matrix, and have shown that this update can be computed with the same amount of work as updates that do not take the bounds into account. In particular, we devised an algorithm that computes this update in time proportional to the number of nonzeros in the sparsity pattern of the Jacobian matrix. Finally, we showed that this update shares the local convergence properties of Broyden's and Schubert's updates.

In conjunction with the work on optimization algorithms, we made several enhancements to the codes for MINPACK-2 (a collection of optimization subroutines for solving systems of nonlinear equations, nonlinear least-squares problems, and unconstrained and linearly constrained minimization problems). In particular, the codes have been written in Fortran 77; subroutine names now reflect the precision of the computation; the codes correctly handle Jacobians with zero columns; the calculation of the step has been made more robust; and the initial choice of step bound has been improved.

We have also been working actively on a new book about optimization. Our current aim is to cover the important theoretical and practical developments in the field; for example, discussions of algorithms will be accompanied by comments on software. The background research needed for a first draft has been completed.

2. Optimization in Modeling

Michael Minkoff

Problems in energy systems analysis often involve the use of specialized numerical optimization techniques. We have been studying one such case — a chemical problem that requires the accurate determination of concentrations of chemical species at fixed temperature and pressure; gas, condensed, and chemical solution phases must be identified, and their species and concentrations calculated. The optimization problem can be formulated as an energy minimization problem subject to mass balance constraints (Gibbs free-energy minimization). By using an ideal model of a chemical solution, we can apply duality theory to obtain a computational approach using methods of geometric programming. Our initial work was designed to handle relatively small problems (25 species with 4 elements). More recently the approach was used in a variety of coal combustion chemistry studies and was extended to solve large-scale chemical equilibrium problems (300 species and 11 elements with concentrations ranging over 60 orders of magnitude). To our knowledge, problems of this size and complexity have not been previously solved.

Our next efforts in this area involve modifying the dual geometric programming approach for use in a workstation environment. We will also consider nonideal models of a chemical solution. By introducing artificial variables, we have been able to solve a limited class of nonideal models by imbedding the problem in a larger ideal solution model. This approach will be studied further to determine the class of nonideal models that can be treated in this manner.
B. Approximations and Software Basics

William J. Cody, Jr.

Work on approximations and software basics involves the production of software for special functions and programs exploiting new techniques for performance evaluation of the function software. Our major efforts focus on the development of SPECFUN, a collection of special function programs that are highly transportable, at a small cost in performance. Each of the function programs is accompanied by a transportable test program suitable for checking any software for that function; wherever possible, the test programs are self-contained, extending ideas first used in the ELEFENT package.

Deficiencies in the double-precision elementary function library supplied with 4.2 BSD UNIX initially hampered the work on SPECFUN. Testing showed that only the existing programs DATAN and DSQRT were accurate enough for our purposes. We therefore prepared replacement routines based on the Software Manual for the Elementary Functions for all of the other functions. The programs are available through the National Energy Software Center.

<table>
<thead>
<tr>
<th>Function</th>
<th>FUNPACK</th>
<th>SPECFUN</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_1(x), Ei(x), e^{-x}Ei(x)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>$\Gamma(x)$</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>lnGamma(x)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>psi(x)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>erf(x), erfc(x)</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>F(x)</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>J_0(x)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>J_1(x)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>J_1(x)</td>
<td></td>
<td>P</td>
</tr>
<tr>
<td>Y_0(x)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>I_0(x), e^{-x}I_0(x)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>I_1(x), e^{-x}I_1(x)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>I_1(x), e^{-x}I_1(x)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>K_0(x), e^{-x}K_0(x)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>K_1(x), e^{-x}K_1(x)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>K_1(x), e^{-x}K_1(x)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>K_0(x), e^{-x}K_0(x)</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>K_1(x), e^{-x}K_1(x)</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>funpack SPECFUN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Contents of FUNPACK and SPECFUN
During the past year we added to SPECFUN programs for the error and complementary error functions, the Bessel functions K of order 0 and 1, and a test program for the logarithm of the gamma function. We also worked on RKBESL, a program for the K Bessel functions of arbitrary real order and positive argument, and on the accompanying test program. RKBESL implements an algorithm of N. Temme and builds on a program prepared by J. B. Campbell. Improvements in RKBESL over Campbell's program include conversion to Fortran 77, better program structuring, extension to computation of both exponentially scaled and unscaled functions, elimination of underflow and overflow, and program portability. The accompanying test program exploits the multiplication theorem for Bessel functions as a means of checking accuracy.

We plan to finish the companion program RJBESL (J Bessel functions of arbitrary real argument and order) and its test program. We also expect to add programs for the Bessel functions Y of order 0 and 1, and to begin work on Bessel functions Y for arbitrary real order. The Y functions of orders 0 and 1 will require the generation of new rational minimax approximations because suitable ones do not now exist. We have the tools to generate these approximations, but the most efficient approximation forms must be determined.

C. Quadrature
James N. Lyness and Burton S. Garbow

The thrust of our quadrature research comprises both the design of quadrature techniques and algorithms and the construction of quadrature software.

We are investigating the use of lattice rules for high-dimensional quadrature (six to nine dimensions). Lattice rules are extensions of the number theoretic rules associated with Korborov and Conroy. In conjunction with I. H. Sloan of the University of New South Wales in Australia, we carried out some preliminary calculations involving transformations that take all space to a hypercube. We are now studying their possible use as a basis for extrapolation.

With colleagues at the University of Naples, we are also working on the Laplace transform problem. Our investigations focus on the Weeks and the Talbot-Green methods.

New projects include theoretical investigations into the nature of some advanced error functional expansions relating to N-dimensional integration of singular functions. In addition, we will begin a study of the use of series acceleration techniques in various areas; our starting point will be the evaluation of one-dimensional integrals whose integrand function has an infinite oscillating tail. We have already modified a method of Longman that employs the Euler transformation; in many cases this seems to be the best available method.

D. Numerical Solution of Partial Differential Equations
Gary K. Leaf and Michael Minkoff

In solving time-dependent systems of nonlinear partial differential equations (PDEs) (e.g., convection-diffusion problems), large systems of structured linear algebraic equations need to be
solved. In an effort to reduce the storage requirements, we are studying the role of preconditioned iterative methods for solving such linear systems. These methods involve two aspects: the selection of an iterative technique and the selection of a preconditioner.

In the area of iterative techniques, we are studying conjugate-gradient-like algorithms for nonsymmetric matrices. Two algorithms based on the conjugate residual method have been implemented — Orthomin and Orthodir. We have also implemented a third algorithm based on a generalization of ordinary conjugate gradients applied to nonsymmetric matrices; it differs from the other algorithms in the choice of the functional being minimized and the recursion relation for the search direction.

In the area of preconditioners, we are studying nested factorization and approximate inverses. During 1985 we developed and implemented two variants of nested factorization. The first preserves column sums, in the sense that the column sums of the original matrix are the same as those of the approximate matrix. (One motivation for using this variant is that in the case of a tridiagonal matrix, an exact factorization will be generated.) The second variant preserves row sums, which can be interpreted as preserving column sums for the transpose matrix.

During the last part of 1985, we began studying approximate inverse preconditioners. Our initial approach uses the recursive aspect of the nested factorization method but applies an approximate inverse at the inner levels of the recursion. In particular, an approximation to the inverse is developed which, within the class of matrices under consideration, is a best approximation in the sense of the Frobenius norm. Several preconditioners were developed based on the use of diagonal and tridiagonal approximate inverses.

We are now evaluating the usefulness of nested factorization and approximate inverses for treating convection-diffusion for PDEs. In particular, we are considering a finite-difference approximation of a three-dimensional elliptic equation involving diffusion, convection, and absorption terms. By varying the convective term, we can control the indefiniteness and asymmetry of the coefficient matrix. As a measure of the efficiency of the method, we are considering the number of iterations (the time and the operation counts) in the preconditioning and the iterative phases. In techniques of this type, the number of iterations is known to be equal to the number of distinct eigenvalues of the symmetric part of the preconditioned matrix. We plan to use computational and graphical tools to examine these eigenvalues and to relate the results to the iteration count.
SOFTWARE METHODOLOGY

Software Methodology research includes activities that range from theoretical work to applications of interest to the Department of Energy.

A major part of these activities involves research in automated reasoning. Studies include design of new inference rules and strategies, development of an abstract machine for logic programming that exploits multiprocessor architectures, and applications in such areas as power plant control and logic circuit design.

In addition, Argonne is working on programming environments for advanced-architecture machines, automated program transformation, abstract programming, and standards for Fortran and IEEE floating-point arithmetic.

A. Automated Reasoning

John R. Gabriel, Ewing L. Lusk, William W. McCune, Ross A. Overheek, Brian T. Smith, and Lawrence T. Wos

Automated reasoning has emerged as an exciting new research area. We are concentrating on three categories: theory, software, and applications.

1. Theory

William W. McCune and Lawrence T. Wos

Our research on new inference rules and new strategies proved extremely encouraging. Most significant was the discovery of a long-sought-after inference rule, negative hyperparamodulation. In its simplest form, negative hyperparamodulation considers a set of clauses, one of which is an inequality, one of which is a clause into which some substitutions will be attempted, and the remaining are equalities. Our early experiments, taken from problems in group theory and ring theory, suggest that negative hyperparamodulation yields proofs for simple theorems quickly and easily. This capability is one measure of the progress of a field such as automated reasoning.

We also formulated a new version of (positive) hyperparamodulation. The new version has the side effect of controlling to some extent the actions of paramodulation. This control resembles the use of a strategy that reorders the retained clauses on the set of support.

In another area we discovered a possible new use for the unit preference strategy, a strategy formulated in the early 1960s by Argonne researchers. It was found that certain puzzles are proved in one-twelfth the time that occurs with the current given/have-been-given algorithm. As a result, we are exploring the possibility of using alternative algorithms. We expect this effort to continue for several years, as we explore search algorithms that are term oriented rather than clause oriented. Such algorithms would be particularly useful for controlling the actions of paramodulation.

In February 1986, we held an informal workshop with R. Boyer (University of Texas) and M. Stickel (SRI). The object was to study the feasibility of using Gödel's finite axiomatization of set
theory with an automated reasoning program. We produced a set of clauses that has been proved to be equivalent to Gödel's axiomatization. This accomplishment makes virtually all of mathematics at least theoretically within reach of automated reasoning. Preliminary studies show how one can obtain complete and detailed proofs within this axiomatization of various theorems, including ones from group theory. We plan to continue this study for several years, for it is deep and difficult.

2. Software

_Ewing L. Lusk and Ross A. Overbeek_

To support our theoretical work, we continue to improve our automated reasoning software. Enhancements this year included a mechanism for handling reasoning by case analysis, which allowed the program to do several hard problems previously beyond its grasp; modification of the subsumption routine to test for identical clauses (thus reducing the run time on long problems by up to 35 percent); and development of a "hot list," which asks the user to specify certain equality units as being of special interest.

A major achievement this year was the implementation of an extended Warren Abstract Machine (WAM), an abstract machine suitable for implementations of inference engines. The technology included in our implementation of WAM was pioneered by David Warren. We extended Warren's design to allow those features required to exploit the availability of multiprocessor systems and to implement more general inference engines (these additions included instructions to support an "occurs check" and "instance of a formula" computations). Our implementation of WAM achieved substantial speedups on the Denelcor HEP, the Sequent Balance 8000, and the Encore Multimax. We expect that it will become the foundation for a new generation of inference systems. These systems will feature substantially higher performance through clause compilation and parallelism, ease of maintenance, and ease of customization.

We also investigated several dialects of parallel logic programming, primarily through extended visits by researchers from the groups at the Weizmann Institute (Concurrent Prolog), the University of Lisbon (Delta-Prolog), and Imperial College, London (Parlog). This work confirmed the wisdom of our decision to concentrate our efforts on the abstract machine and postpone commitment to any particular parallel logic-programming language. We have begun writing both a Prolog and a Parlog compiler (to generate the WAM machine code). This work will result in a portable, high-level language capable of exploiting existing multiprocessors. We intend to use the new capabilities to create a prototypical theorem prover for Horn sets (a restricted set of first-order problems). We will use the theorem prover to answer a variety of significant research questions: How much improvement in performance can be obtained through the technique of compilation of clauses (rather than the manipulation of clauses as data structures)? How much exploitable parallelism is there in most mathematical theorems? How suitable are Prolog and Parlog for constructing substantial software systems for use on multiprocessors? By the end of this year we expect to realize performance improvements of factors of 10 to 50 using the new technology.

3. Applications


Our major effort involves a long-term project in plant control. This work is being carried out in
collaboration with A. Wojcik of Michigan State University (formerly of the Illinois Institute of Technology) on issues of proof of properties of computer hardware, and entails a close working relationship with Draper Laboratories at the Massachusetts Institute of Technology. The goal of the project is to prove statements about properties of a fault-tolerant computer system under consideration for use in Argonne’s Experimental Breeder Reactor-II.

We began the project by exploring various representations for modeling the hardware for the Draper fault-tolerant processor (FTP). During 1985 we decided on a modeling and representation scheme based on Petri nets. Using this scheme, we obtained proofs of fault-tolerance of the basic data transfer commands of both an initial and revised design for the FTP. These proofs have shown the minimum fault-containment regions for each of the primary data transfer commands, illustrating how the Draper FTP can function properly in the presence of more than one fault. We also began searching for a common representation that models both hardware and software based on Petri net modeling. A generic model for an application program was devised and used to show that any program consistent with the model and performing an operation \( P \) on the available data performs \( P \) in the presence of a fault in one of the hardware fault-containment regions of the Draper FTP. In addition, we proved the 32 functions of the logic and arithmetic logic unit 74LS181. Currently, we are using the Petri net representation to model more of the Draper FTP at the register and system levels. In particular, the programmable-logic-array chips unique to the Draper FTP are being analyzed.
We began exploring analysis of digital logic circuits by recursive functions. In 1983 and 1984 H. Barrow published brief papers on proving the correctness of digital hardware designs, the theoretical foundations for which were published by Gordon in 1981. Barrow's work has since been proprietary to Fairchild, and thus details of implementation have not appeared in print. Our initial attempts in 1984 to write Prolog programs replicating the functionality reported by Barrow were only partly satisfactory; they did not deal with certain interesting problems not mentioned by Barrow. During 1985, however, we made some progress in understanding Gordon's work and its relevance to our research on fault tolerance. We have therefore begun to extend the programs written in 1984, in an effort to make them into useful tools.

In another area, we have begun investigating models of discovery in physics. We have formulated a paradigm where a physical theory is viewed as a computable function mapping causes into consequences, subsuming a collection of experimental results, and thought to be valid in some domain containing the experimental causes. Our objective is to use ITP to infer a theory from experiments. The task we have undertaken is to infer Newton's laws from Kepler's laws. So far, this has been done only for the law of acceleration. Further progress depends on whether ITP can be extended easily to provide an equivalent of lambda calculus through calls on the Prolog subsystem of ITP. Similar subsystems can be envisaged, such as the numerical solution of partial differential equations. This would allow the solution search mechanisms of theorem proving to be used in more general problem solving. Of course, owing to the possibility of logical fallacy in the subsystem, the "solution" would not have the rigor of a "proof." Nevertheless, such a hybrid seems to have potential as a "problem solver."

Finally, we completed several small applications projects, including the design of a database for plant control and maintenance for the Fast Flux Test Facility breeder reactor in Richland, Washington (in collaboration with S. Seeman, R. Colley, and D. Smith of the Hanford Engineering Development Laboratory), use of our automated reasoning system ITP to explore alternative approaches to finding efficient sequences for cars passed through an assembly line (in collaboration with General Motors), and design and implementation of an expert system for writing special research contracts and for the processing of small purchase orders (in collaboration with the Procurement Department of the DOE Chicago Office).

B. Program Development Aids and Automated Transformation

James M. Boyle, Wayne R. Cowell, Kenneth W. Dritz, Burton S. Garbow, and Brian T. Smith

Our work on software tools and techniques has three main concerns: analyzing programming environments oriented to the development and maintenance of Fortran programs, developing an abstract programming methodology to make programming easier and faster, and automatically creating reliable variants of software routines. The broad range of activities included under this area makes it possible for MCS scientists both to develop new software production tools and to test them in a working environment.

1. Software Tools and Programming Environments

Wayne R. Cowell and Burton S. Garbow

Using experience from our past work on the Toolpack project, we are conducting research on a Fortran-oriented programming environment for advanced-architecture machines. Initial efforts were
devoted to constructing tools that perform certain program transformations automatically. Analysis of the transformations, construction of the tools, and evaluation of the results were carried out in collaboration with Christopher P. Thompson (Atomic Energy Research Establishment, Harwell, U.K., and Numerical Algorithms Group, Inc., Downers Grove, Illinois).

For nests of Fortran DO loops that match certain patterns commonly found in numerical linear algebra software, the tools unroll the loops and simplify the resulting Fortran. The effect is to reduce the number of vector loads and stores—a major bottleneck to efficiency—when the transformed Fortran is compiled by a vectorizing compiler.

Automated transformation tools offer several benefits. In the case of the unrolling transformations, programs may be written and maintained in their relatively simple "rolled" form while the transformation tools produce the more complicated (from a human standpoint), but more efficient, "unrolled" form. Since the tools transform Fortran constructs that match certain paradigms exhibited by existing programs of interest, any other program that contains such constructs may then be transformed according to the same techniques. In this sense the tools function as an "expert" who applies specified techniques to problems that fit certain patterns. Moreover, the ability to easily generate versions of the program unrolled to specified depths facilitates experimentation to find the optimum unrolling depth for a given program on a given machine.

To determine the effectiveness of the automated transformations, we compared the performance of a collection of linear algebra routines that had been transformed by hand were transformed by the tools and timed on a CRAY-1 and a CRAY X-MP. The tool-transformed routines performed essentially the same as the hand-transformed routines.

The transformation tools were incorporated into a UNIX/Toolpack environment and executed on the "workstation" (modeled, in this case, by a VAX 11/780); the transformed programs were run on advanced-architecture machines.

Toolpack-based transformations among Fortran representations
2. Automated Program Transformation

James M. Boyle and Kenneth W. Dritz

We are investigating methods for specifying, implementing, and proving the correctness of program transformations. These investigations have led to the development of TAMPR, a general program transformation system, which is being used in a number of applications (e.g., converting programs written in Lisp into efficiently executable and portable Fortran programs for both sequential and parallel machines, changing multidimensional arrays to one-dimensional arrays to improve program efficiency) traditionally performed using various separate tools.

Our focus during the past two years has been on making the TAMPR transformer transportable. In the approach we are using, the transformer (written in Lisp) is taken as the abstract specification for the Fortran program that implements the transformer. A set of TAMPR transformations evolve the applicative Lisp program into a Fortran one. In 1985-1986 we enhanced the Lisp-to-Fortran transformations to handle Lisp programs that use some nonapplicative constructs. Such constructs are used in the portions of the transformer that perform input and output and in the trace facility that prints the result of each transformation step. These transformations were completed, tested, and applied to the appropriate sections of the transformer.

We also enhanced the portability of the Lisp F3 support routines for the transformer. These routines can now be ported to machines using either the "normal" method of packing characters into a word or the DEC "byte-swapped" approach, simply by changing one parameter setting. They have been ported to the Ridge-32 scientific workstation, the CRAY X-MP, the Denelcor HEP, the Sequent Balance 8000, and the Encore Multimax computer.

3. Abstract Programming

James M. Boyle and Kenneth W. Dritz

Abstract programming involves the use of both problem-related abstractions and abstractions of the programming process itself. By encapsulating certain details—isolating their interactions with other parts of the program—such abstractions make it easier to write clear, correct programs. Moreover, these features make an abstract program potentially more useful than a conventional one. The problem-oriented notation contains information about the intent of the program that is lost when it is written using low-level constructs, such as loops, arrays, and variables. Additionally, the abstract program is not cluttered with implementation decisions that are specific to a particular architecture. Thus, an abstract program is a good starting point for deriving programs that run efficiently on computers of widely differing architectures—conventional sequential machines, vector machines, parallel machines, hypercubes, and distributed networks.

One area studied involved a numerical abstract programming problem—automatically introducing temporary variables (which could be held in machine registers) to optimize loops that compute recurrence relations. In collaboration with R. Taylor (undergraduate co-op student), we wrote transformations to carry out this optimization and to automatically decide which elements of the recursion to place in temporary variables. Only a modest number of new transformations were needed because we were able to reuse transformations originally developed for other purposes. This ability to reuse transformations in different applications is one of the major advantages of a transformation-based programming methodology.
Together with M. N. Muralidharan (University of Kentucky), we also studied an optimization that can be performed during the Lisp-to-Fortran conversion—the conversion of tail-recursive functions to iterative ones. This optimization reduces the demand for space on the argument stack and speeds up the execution of the transformed program.

**C. Language and Arithmetic Systems Activities**

*William J. Cody, Jr., Paul C. Messina, and Brian T. Smith*

Language and arithmetic systems research studies address the problem of ensuring that programming languages and arithmetic systems meet the diversified and changing needs of their users. Argonne is participating in three projects: revisions of ANS Fortran, tracking of scientific programming languages for use in the Department of Energy, and development of IEEE standards for floating-point arithmetic. Our special concern in these projects is the suitability of the language and computer arithmetic for numerical computations.

1. **Fortran Standards Committee**

   *Brian T. Smith*

Since 1977, the X3J3 Fortran Standards Committee of the American National Standards Institute has been considering revisions of the standard. We have assisted in proposals for an environmental inquiry facility, a generalized precision facility, and a derived data-type facility. We have also taken an active role in editing the basis document, particularly the chapter on expression, which will become officially the draft proposed standard in late 1986.

2. **Language Working Group**

   *Paul C. Messina and Brian T. Smith*

Commercially available multiprocessor systems for scientific work have been installed at several of the DOE national laboratories. The Language Working Group (LWG) has been studying the language requirements needed to program such systems effectively. Two of its members, collaborating with P. O. Frederickson of Los Alamos National Laboratory, prepared a report that specifies language extensions to Fortran that should be used for such systems. The LWG also organized a workshop that explored the question of whether a mixed language approach might be a viable alternative to the creation of very complex programming languages. Additionally, as part of a study of compiler technology that may contribute to the effective use of the new computer architectures, the LWG heard presentations from Kuck Associates on their techniques.

3. **Floating-Point Arithmetic**

   *William J. Cody, Jr.*

During the past several years we have been actively involved in the standards work of the IEEE Computer Society.

The IEEE P754 committee completed a draft of its proposed standard for binary floating-point arithmetic in December 1982. The standard was approved in 1985, and this committee is now inactive.

The IEEE P854 committee began deliberations in January 1981 on an upward-compatible radix-
and wordlength-independent standard. The first draft of that proposed standard was published in August 1984. In December 1985, the Microprocessor Standards Committee approved P854 Draft 3.0 and passed it on to the Technical Committee on Microcomputers for further action. Review and approval by the committee should take about six months, barring objections on technical grounds. Following that approval, the draft standard will be submitted to the IEEE Standards board for a procedural review and acceptance as a standard. We expect final approval by late 1986.

Our current efforts involve preparation of a document that describes a function library supporting the special features of IEEE-style arithmetic.
Appendix A

PERMANENT 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
J. R. Gabriel, M.S., University of Otago, New Zealand, 1953
B. S. Garbow, M.S., University of Chicago, 1952
K. E. Hillstrom, M.S., Northwestern University, 1957
H. G. Kaper, Ph.D., Rijksuniversiteit, Groningen, 1965
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
P. C. Messina, Ph.D., University of Cincinnati, 1972
M. Minkoff, Ph.D., University of Wisconsin, 1973
J. J. More, Ph.D., University of Maryland, 1970
R. A. Overbeek, Ph.D., Pennsylvania State University, 1971
G. W. Pieper, Ph.D., University of Illinois, 1969
E. Rackow, A.A., Wisconsin School of Electronics, 1978
B. T. Smith, Ph.D., University of Toronto, 1969
D. C. Sorensen, Ph.D., University of California, San Diego, 1977
R. Stevens, B.S., Western Michigan University, 1985
L. T. Wos, Ph.D., University of Illinois, 1957
Appendix B

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
Working Group 2.5 for Mathematical Software
Chairman, IEEE Standards Subcommittee for Radix and Format
Independent Floating Point Standard
Consultant, C. Abaci

J. J. Dongarra
Associate Editor, CACM
Associate Editor, Parallel and Distributed Computing
Associate Editor, Frontiers in Applied Mathematics
Chairman, SIAM-SIAG on Supercomputing
Member, Technical Review Group for the Office of Advanced Scientific
Computing, NSF
Member, SIGNUM Board
Member, SIAM Council

J. R. Gabriel
Member, American Society of Mechanical Engineers
Process Control Computer Committee
Member, IEEE Working Group on AI in Computer Aided
Control Systems Design

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

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

P. C. Messina
Chairman, DOE Advanced Computing Committee Language
Working Group
Member, Energy Research Supercomputer Facility
Advisory Board
Co-chairman, Task Group for Supercomputer Acquisition Strategy
Member, Source Selection Board for High-Performance Computer Acquisition
M. Minkoff
Adjunct Professor, Northern Illinois University

J. J. Moré
Editor, SIAM Journal on Numerical Analysis
Associate Editor, Numerische Mathematik
Adjunct Professor, Northwestern University

G. W. Pieper
Assistant Professor, Illinois Benedictine College

B. T. Smith
Chairman, SIGNUM Fortran Committee
Member, DOE Advanced Computing Committee Language Working Group
Member, International Federation for Information Processing Working Group 2.5 for Mathematical Software
Member, X3J3 Fortran Standards Committee
Member, NASA/RTI Panel on "Validation Research Planning"

D. C. Sorensen
Associate Editor, SIAM Journal on Scientific and Statistical Computing
Associate Editor, ACM Trans. on Math. Software

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

PRESENTATIONS

The following list reflects articles published, reports distributed, and talks presented from April 1, 1985, through June 30, 1986. 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


J. J. Dongarra, "How Do the 'Minisupers' Stack Up?" Computer, 19, 3 (March 1986) 93, 100


J. J. Dongarra, L. Kaufman, and S. Hammarling, "Squeezing the Most Out of Eigenvalue Solvers on High-


Computing (Nice, France), June 1985


E. Lusk and R. Overbeek, "Reasoning about Equality," JAR 1, 2 (May 1985) 209-228


Portability, Complexity, and Efficiency in Multiprocessing Environments." Proc. of the Workshop on Parallel Processing Using the Heterogeneous Element Processor, Norman, Oklahoma, March 20-21, 1985, pp. 245-26


L. Wos, "Automating Reasoning," Abacus 2, 3 (Spring 1985) 6-21


Reports


Technical Memoranda


(August 1985)


Oral Presentations


J. M. Boyle, "The TAMPR Program Transformation System," Reading BCS/FACS Transformation Workshop,
Reading, England, April 2, 1985

J. M. Boyle, "Deriving Sequential and Parallel Programs from Pure LISP Specifications by Program Transformation," IFIP TC2 Working Conference on Program Specification and Transformation, Bad Toelz, West Germany, April 14-17, 1986

W. J. Cody, "SPECFUN—A Portable Special Function Package," USARO Workshop on Microcomputers in Large-Scale Computing, University of Delaware, May 20-22, 1985


H. G. Kaper, "Inequalities for Differential and Difference Operators," DOE-NASIG Meeting, Albuquerque, New Mexico, August 14, 1985


H. G. Kaper, "Asymptotics of the Titchmarsh-Weyl m-Coefficient for Integrable Potentials," International Conference on Differential Equations and Mathematical Physics, University of Alabama, Birmingham, Alabama, March 5, 1986


E. L. Lusk, "The Argonne Automated Reasoning System," Computer Science Department, Universitaet Kaiserlautern, June 1985
E. L. Lusk. "Automated Reasoning," Computer Science Department, DePaul University, September 1985


E. L. Lusk. "Portable Programs for Parallel Processors," First RIMSIG meeting, Lawrence Livermore National Laboratory, October 1985


Minkoff, M., "Mathematical Programming with Differential Equations Simulation Models," XII International Mathematical Programming Symposium, Boston, Massachusetts, August 5-9, 1985


J. J. Moré. "Projected Gradients and Conjugate Gradients for Large Scale Problems," Rice University, Department of Mathematical Sciences, February 1986


J. J. Moré. "Projected Gradients and Conjugate Gradients for Large Scale Problems," Waterloo University, Department of Optimization and Combinatorics, March 1986


D. C. Sorensen, "A Fast Algorithm for the Symmetric Eigenvalue Problem," NASIG meeting, Albuquerque, New Mexico, August 1985

D. C. Sorensen, "Linear Algebra on High-Performance Computers," International Symposium on Mathematical Programming, special session on parallel processing, Cambridge, Massachusetts, August 1985


D. C. Sorensen, "Linear Algebra on High-Performance Computers," invited talk, Parallel Computing 85, Berlin, West Germany, September 1985

D. C. Sorensen, "Performance of the Alliant FX/8 on Some Basic Linear Algebra Problems," RIMSIG meeting, Lawrence Livermore National Laboratory, October 1985


L. Wos, "Future of Automated Reasoning," NCC panel and lecture, July 18, 1985


L. Wos, Two lectures on Automated Reasoning, University of Wisconsin Parkside, April 18, 1986
Appendix D
TEMPORARY STAFF AND VISITORS

Faculty Research Participants

James Burke, University of Washington
Charles Fulton, Florida Institute of Technology (sabbatic leave)
Alan Genz, Washington State University
Floyd Hanson, University of Illinois, Chicago
Terrance Houlihan, Polytechnic University
Lothar Reichel, University of Kentucky
Clasine van Winter, University of Kentucky (sabbatic leave)

Visiting Scientists

Frederick Atkinson, University of Toronto
Julio Diaz, University of Oklahoma
Iain Duff, AERE Harwell, England
Robert McFadden, Northern Illinois University
Lennart Johnsson, Yale University
Robert McFadden, Northern Illinois University

Staff Term Appointments

Assistant Scientist

Terry Disz, University of Illinois, Chicago

Assistant Computer Scientists

Jose Cunha, University of Lisbon
Joseph Kijaich, Jr., Illinois Institute of Technology
Jacob Levy, Weizmann Institute, Israel
William W. McCune, Northwestern University
Jairo Panetta, Purdue University

Scientific Assistants

Andrew A. Anda
Nicholas Karonis, Northern Illinois University
Special Term Appointments

Man Kam Kwong, Northern Illinois University
Bernard Matkowsky, Northwestern University
Anthony Wojcik, Illinois Institute of Technology
Anton Zettl, Northern Illinois University

Guest Graduate Student

Steven Winker, University of Illinois, Chicago Circle

Research Associates

Ralph Butler, University of North Florida
Paul Calamai, University of Waterloo, Canada
Monagur Muralidharan, University of Kentucky

Resident Associates

K. A. Ariyawansa, Washington State University
Jeremy DuCroz, NAG Central Office
Lawrence Henschen, Northwestern University
Brad Lucier, Purdue University
Peter J. Mayes, NAG Ltd.
Steven G. Nash, Johns Hopkins
Ahmed H. Sameh, University of Illinois, Urbana
Jeff S. Scroggs, University of Illinois, Urbana

Student Resident Associates

Jennifer Arbanas, Western Michigan University
Jonathan Mills, Western Michigan University
Bruce Parrello, Northwestern University
Todd Peterson, Cornell University
Sushil V. Pillai, University of Illinois, Chicago
Vasanthi Swaminathan, University of Illinois, Chicago
Eric Van de Velde, Courant Institute
Student Research Participants

Richard Chapman, Wake Forest University
David W. Gillman, Yale University
Patrick Haley, Northwestern University
Tasso Kaper, College of the University of Chicago
William Kastak, Baldwin Wallace College
Scott King, University of Chicago
Christina Mihaly, Bowling Green State University
William G. Newman, Hastings College
Robert Olson, Lisle Senior High School
Beth Spellman, Yale University
David Tcheng, Illinois State University

Research Aides

James Kohl, Purdue University
Betsy Ann Knox, Illinois Benedictine College

Co-op Employees

Barbara Bonar, Queens University of Belfast
Robert Taylor, Queen’s University of Belfast
Appendix E

MEETINGS AND WORKSHOPS

1. Mixed Language Programming Workshop — April 30 - May 2, 1985

The Language Working Group held a workshop on mixed language programming. This workshop featured discussions from scientists from Europe and North America on the perceived needs for mixed language programming, the difficulties in implementing these environments, and the impact on language standardization. Languages covered included Fortran, PL/I, C, Ada, Pascal, Algol 68, Lisp, Prolog, and SISAL.

2. Language Working Group Meeting — May 2-3, 1985

The Language Working Group meeting was held in Chicago to examine proposed features of Fortran 8x and to review problems that are hampering the standardization process.


The goal of the Argonne workshop on the Warren Abstract Machine was to explore work in progress on high-performance inference engines, particularly those based on David Warren’s abstract machine. The topics included compilers, intrinsics, database and graphics interfaces, and multiprocessing.


The MCS Division sponsored a workshop on differential equations as part of the ongoing research program in applied analysis. The ten invited participants attended seminars and collaborated in various scientific activities.

5. International Conference on Vector and Parallel Computing — June 2-6, 1985

Approximately 250 people attended a week-long conference in Norway to discuss current research and development in supercomputing. Topics included the role of supercomputing in applied artificial intelligence, the development of numerical algorithms for vector and parallel computing, and supercomputing architectures. In parallel with the conference were workshops on industrial problems, logic programming, and reservoir modeling. Argonne played an active role in coordinating this conference.


Argonne sponsored a one-day workshop on advanced scientific computing. In the morning, guest speaker Jack Worlton presented an overview of high-performance computing. During the afternoon, discussions focused on user problems and applications in universities, government, and national laboratories.

A workshop was held at Argonne during the last two weeks of February to study Gödel's finite axiomatization of set theory. The participants were Robert Boyer from University of Texas and MCC, Mark Stickel from SRI, and Ewing Lusk, William McCune, Ross Overbeek, and Larry Wos from ANL/MCS. Boyer had brought to our attention the fact that almost all of mathematics could be at least theoretically attacked with an automated reasoning program, providing we used an approach similar to that based on Gödel’s finite axiomatization of set theory. The other approaches to set theory, unfortunately, do not lend themselves to a straightforward representation in first-order predicate calculus. McCune wrote a program in Prolog that took Gödel's axioms and produced a finite set of clauses from them. During the two weeks, we massaged those clauses to produce a logically equivalent set, but one that has properties more suitable to effective automated reasoning. We also produced clauses that lay the foundation for some studies in number theory and abstract algebra. Using those for algebra, McCune produced some lengthy proofs of very basic lemmas in group theory. His success demonstrates how the proofs might look, and gives us a basis for trying to find appropriate strategies and inference rules for producing such proofs efficiently with an automated reasoning program. These proofs were produced in part by hand, and in part by computer. The clauses that capture the essence of Gödel's axioms, and some of the proofs, have been published in the Journal of Automated Reasoning.


This Language Working Group meeting, held in Sante Fe, New Mexico, focused on two main topics: a review of progress on revisions to the Fortran Standard, and discussions on parallel programming issues with vendors and implementors. Additionally, there was discussion of several LWG-sponsored workshops, including one on Ada® and one on programming parallel computers.


Approximately 50 researchers from industry, universities, and other research facilities in the Chicago area attended a one-day symposium on advanced computing. Argonne staff and invited speakers presented lectures on the state of the art in advanced computing, including information about the various architectures available, problems that arise in using them, and research aimed at finding solutions.

10. Workshop on Automated Reasoning — June 24-25, 1986

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.

11. Programming Languages for Advanced Architecture Computers — June 30-July 1, 1986

Argonne hosted a workshop designed as an information exchange forum among commercial vendors of computers with parallel, vector, and other architectural features. Each participating organization prepared a brief, informal description of their strategy with regard to programming languages for their present and future computers. Ample time was also allowed for open discussions. Among the topics covered were the automatic detection of parallelism by "smart" compilers, enhancements of Fortran, extensions to other conventional programming languages, and new and specialized languages.
Appendix F

SEMINARS

During the period April 1, 1985 - June 30, 1986, 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, sponsored jointly by the MCS Division and Computing Services.

Ian Gladwell
*Portable Almost Block-Diagonal Solvers on Pipeline Processors*
University of Manchester, England
May 3, 1985

A. Faustini
*CHRONOS, A Language for Formal Description of Real Time Systems and Their Programming*
Arizona State University
May 14, 1985

Nicholas Gould
*The Accurate Determination of Search Directions for Simple Differentiable Penalty Functions*
University of Waterloo
May 16, 1985

George Bader
*Solutions of Nonlinear Collocation Equations*
Simon Fraser University
May 17, 1985

Mark Kon
*Sobolev Smoothing Properties of Analytic Functions of Elliptic Operators*
Boston University
May 23, 1985

Jacqueline Fleckinger
*Asymptotics of Eigenvalues for Indefinite Problems*
Université Paul Sabatier
June 4, 1985

Silvia Solmi
*Mathematical Aspects in the Adjustment of Astronomic Data and the Reconstruction of the Celestial Sphere*
Stanford University
July 2, 1985
Douglas R. Smith  
*Semiautomatic Algorithm Design*  
Kestrel Institute  
July 18, 1985

E. Mulbaney, R. Hausman, and P. Cannon  
*ST-100 Array Processor - Description and Usage*  
STAR Technologies, Inc.  
September 18, 1985

Russell R. Barton  
*Using Pseudo Functions for Testing Unconstrained Optimization Algorithms*  
RCA Research Laboratories  
September 19, 1985

Jurgen Batt  
*The Nonlinear Vlasov-Poisson System of Partial Differential Equations in Stellar Dynamics and Plasma Physics*  
Mathematisches Institut der Universität Muenchen, West Germany  
September 26, 1985

Jacob Levy  
*Concurrent Prolog and Alternative Dialects*  
Weizmann Institute of Science  
September 27, 1985

Florian Potra  
*Newton-Like Methods for Non-Linear Boundary Value Problems*  
University of Iowa  
October 22, 1985

Alexandru Nicolau  
*ESP: An Environment for Scientific Parallel-Programming*  
Cornell University  
October 24, 1985

George Byrne  
*Experiments in Numerical Methods for a Problem in Combustion Modeling*  
Exxon Research and Engineering Co.  
November 4, 1985
Steve Gregory
*The Sequential PARLOG Machine*
Imperial College of Science and Technology, University of London
November 22, 1985

Daniel Woods
*An Adaptive Strategy for Multi-Objective Optimization Problems*
Rice University
January 16, 1986

Richard Waldinger
*Deductive Program Synthesis Research*
SRI International
January 22, 1986

Charles Broyden
*Skew-Symmetric Matrices, Staircase Functions, and Theorems of the Alternative*
University of Essex, England, and University of Illinois, Chicago
February 6, 1986

Joseph W. Jerome
*Two Potentially Useful Algorithms for Nonlinear Systems*
Northwestern University
February 20, 1986

Ralph Carlson
*Some New Results in Solving Scattered Data Interpolation Problems*
Lawrence Livermore National Laboratory
April 17, 1986

John A. Trangenstein
*Higher Order Numerical Methods for Compositional Reservoir Simulation*
Exxon Production Research Company
May 16, 1986

Walter Gautschi
*Moment-Preserving Approximation by Spline Functions*
Purdue University
June 12, 1986

Paul Voda
*Negation in Logic Programming*
The University of British Columbia
June 27, 1986
Andreas Griewank  
*The Mathematical Analysis of Multiphase Equilibrium for Hydrocarbons and Other Mixed Fluids*  
Southern Methodist University  
July 3, 1986

**High-Performance Computing Seminars**

**John Gustafson**  
*Multicomputing with FPS Scientific Computers*  
Floating Point Systems, Inc.  
May 2, 1985

**Gerard Meurant**  
*Multitasking Experiments on the CRAY X-MP-48*  
Centre d'Etudes de Limeil, France  
July 15, 1985

**Lennart Johnson**  
*Band Matrix Solvers for Hypercube Architectures*  
Yale University  
July 30, 1985

**Eric Van de Velde**  
*Algorithms for Computational Fluid Dynamics on the Hypercube*  
Courant Institute  
August 20, 1985

**Julio Diaz**  
*Development of a Block Preconditioned Conjugate Gradient Iterative Solver for Sparse Linear Systems*  
University of Oklahoma  
August 22, 1985

**Gordon Bell**  
*The Encore Continuum: A Complete Distributed Workstation-Multiprocessor Computing Environment*  
Encore Computer  
August 27, 1985

**Herman te Riele**  
*Vector Research at the CWI*  
Centrum voor Wiskunde en Informatica, Amsterdam  
September 3, 1985
John Rollwagen
*CRAY Research Today and Tomorrow*
Cray Research Inc.
September 9, 1985

Anne Greenbaum
*Attempts at Parallelizing Multigrid-Type Methods*
Lawrence Livermore National Laboratory
September 18, 1985

Chris Anderson
*A Look at Domain Decomposition for Elliptic Partial Differential Equations*
Stanford University
October 3, 1985

Cleve Moler
*Matrix Computation on the Intel Hypercube*
Intel Scientific Computers
October 11, 1985

Joel H. Ferziger
*Computational Approaches to Turbulence Research*
Stanford University
October 16, 1985

Daniel A. Reed
*Stencils and Problem Partitionings: Their Influence on the Performance of Multiple Processor Systems*
University of Illinois
February 13, 1986

Eric Van de Velde
*Hypercube Algorithms and Implementations*
Courant Institute of Mathematical Sciences
February 14, 1986

Avi Lin
*On a Parallel Algorithm for Linear Recurrence Systems with Minimum Communication*
Technion Institute of Technology, Israel
February 26, 1986
John Bolstad
A Multigrid Continuation Method for Elliptic Problems with Folds
Lawrence Livermore National Laboratory
March 11, 1986

David Kamowitz
Theoretical and Computational Results for MGR [v]
Multigrid Methods
University of Wisconsin
March 17, 1986

Gerald Hedstrom
Numerical Methods for Multiple-Scale Problems
Lawrence Livermore National Laboratory
March 18, 1986

Patrick Worley
Minimal Information Algorithms and Parallel Computation
Stanford University
March 26, 1986

Ibrahim K. Abu-Shumays
Vectorization of Transport and Diffusion Computations on the CDC CYBER 205
Bettis Atomic Power Laboratory
March 27, 1986

Jeffrey M. Augenbaum
Implicit Methods for Numerical Weather Prediction
NASA/Goddard Space Flight Center
March 31, 1986

Steve Otto
Irregular Finite Elements on a Hypercube
California Institute of Technology
April 3, 1986

Tony F. Chan
Numerical Algorithms on Hypercube Multiprocessors
Yale University
April 11, 1986

Ridgeway Scott
What Do You Want to Say?
University of Michigan
May 15, 1986
David Klappholz
*The Refined Language Approach to Automatic Parallelism Detection*
Stevens Institute of Technology
June 26, 1986
Distribution for ANL-86-35

Internal:

J. M. Beumer (2)
K. L. Kliewer
A. B. Krisciunas
P. C. Messina
G. W. Pieper (84)
B. A. Simmons

ANL Patent Dept.
ANL Contract File
ANL Libraries
TIS Files (5)

External:

DOE-TIC, for distribution per UC-32 (168)
Manager, Chicago Operations Office, DOE
Mathematics and Computer Science Division Review Committee:
  J. L. Bona, Pennsylvania State University
  T. L. Brown, U. of Illinois, Urbana
  P. Concus, LBL
  S. Gerhart, MCC, Austin, Texas
  G. H. Golub, Stanford U.
  W. C. Lynch, Xerox Corp., Palo Alto
  J. A. Nohel, U. of Wisconsin, Madison
  D. Austin, ER-DOE
  G. Michael, LLL