ACTIVITIES AND OPERATIONS OF ARGONNE'S ADVANCED COMPUTING RESEARCH FACILITY:
February 1990 through April 1991

Gail W. Pieper
Mathematics and Computer Science Division

May 1991

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

Abstract 1

1 Advanced Computing Research Facility: Highlights 1

2 Expansion of the ACRF 2
  2.1 New Machines ................................................. 2
  2.2 New Consortium .............................................. 4
  2.3 New Deputy Scientific Director .............................. 4

3 Research in Parallel Computation 4
  3.1 Algorithm Design ............................................. 4
  3.2 Schemes for Efficient Parallelization ....................... 6
  3.3 Programming Languages for High-Performance Computers .... 7
  3.4 Software Packages ........................................... 7
  3.5 Parallel Programming Systems .............................. 8
  3.6 Climate Modeling ........................................... 8

4 Collaborative Projects with Other Research Centers 10
  4.1 Center for Research in Parallel Computation ................ 10
  4.2 Army High Performance Computing Research Center ........... 10

5 Publications, Preprints, and Presentations 11

6 The ACRF—A User Facility 10
  6.1 Classes ..................................................... 19
  6.2 Proposals ................................................... 19
  6.3 Minority Internship in Parallel Computing .................... 20
  6.4 Visitors ..................................................... 20
  6.5 Seminars .................................................... 23

7 Affiliates Programs 24
  7.1 Academic Affiliates ......................................... 24
  7.2 Industrial Affiliates ......................................... 25
  7.3 Affiliates Workshops ....................................... 26
ACTIVITIES AND OPERATIONS OF ARGONNE'S ADVANCED COMPUTING RESEARCH FACILITY: February 1990 through April 1991

by

Gail W. Pieper

Abstract

This report reviews the activities and operations of the Advanced Computing Research Facility (ACRF) from February 1990 through April 1991. The ACRF is operated by the Mathematics and Computer Science Division at Argonne National Laboratory. The facility's principal objective is to foster research in parallel computing. Toward this objective, the ACRF operates experimental advanced computers, supports investigations in parallel computing, and sponsors technology transfer efforts to industry and academia.

1 Advanced Computing Research Facility: Highlights

The Advanced Computing Research Facility (ACRF) at Argonne National Laboratory has been a leader in high-performance computing research for more than six years. During the reporting period of February 1990 through April 1991, the ACRF continued to play a major role in advancing the solution of scientific problems. Further, the ACRF promoted the transfer of its expertise to scientists, engineers, educators, and students in industry and academia. Among the highlights for this period, we note the following:

- Awarded Argonne’s first Minority Internship in Computing
- Hosted two workshops—one on performance visualization tools and one on numerical methods—as part of the Department of Energy program to develop an advanced climate model on a massively parallel machine
- Sponsored two ACRF Affiliates Workshops for industrial and university affiliates
- Helped organized A-PRIME (Argonne PRogram In Mathematics Education) for training future mathematics teachers

In this report, we review these and other activities that have made the ACRF one of the world's principal centers for research in parallel computing.

We are especially proud of our role in promoting Argonne's participation in the Concurrent Supercomputing Consortium. Membership will provide Argonne computational scientists access
to the Intel Touchstone DELTA System, the world's most powerful computer, capable of a peak performance of 32 gigaflops.

Another notable event was the visit of Admiral James D. Watkins, DOE Secretary. The Admiral inspected the ACRF computers and then discussed with our undergraduate students their projects in the ACRF.

All of these events underscore the efforts of the ACRF in making an impact on scientific problem solving. We intend to continue our mission of operating the ACRF as a research facility both for current scientists and for future researchers or teachers.

2 Expansion of the ACRF

2.1 New Machines

We have continued to select and operate computers with innovative designs that are likely to be effective for a wide range of scientific computing tasks. For example, during 1990 we acquired an IBM RS/6000 series machine and worked with IBM in evaluating the hardware and software of this new workstation. In addition, we acquired a Solbourne Enterprise server with 256 megabytes of memory, and we began collaborative experiments with Solbourne designed to identify critical needs for gigabyte workstation architectures over the next decade.

We have also added to the ACRF two new machines:

The **Sequent Symmetry** has 26 processors sharing 32 megabytes of memory. This machine replaces the smaller Sequent Balance, providing researchers with a larger, more powerful, and more advanced shared-memory architecture.

The **Intel Touchstone Gamma** machine has 8 nodes, each with 16 megabytes of memory. This distributed-memory parallel computer is based on the i860 chip, one of the fastest available microprocessors. The machine will enable potential users of the Intel DELTA to immediately begin porting code and familiarize themselves with the characteristics of the architecture.

The ACRF also features six other advanced computers:

- **Active Memory Technology DAP 510**, with 1,024 one-bit processors arranged in a 32 by 32 array, with a total of 8 megabytes of memory.
- **Alliant FX/8**, with 8 vector processors sharing 32 megabytes of memory.
- **BBN TC2000**, a 45-node, shared-memory parallel computer with 192 megabytes of memory.
- **Connection Machine Model 2**, manufactured by Thinking Machines, with 16,384 one-bit processors and a total of 128 megabytes of memory.
- **Encore Multimax**, with 20 processors sharing 64 megabytes of memory.
- **Stardent Titan**, with 4 processors and 32 megabytes of memory.

The ACRF machines are linked to each other and are connected to the Internet. Thus, access to the ACRF machines is readily available throughout the world.
Fig. 1: ACRF
2.2 New Consortium

In late 1990, Argonne joined several leading U.S. research institutions to form the Concurrent Supercomputing Consortium. Through the consortium, Argonne computational scientists will have access to the Intel Touchstone DELTA System.

The DELTA has 528 high-speed microprocessors which work together through a new "mesh interconnect" scheme. The computer has a peak speed of 32 gigaflops, classifying it as the world's fastest computer.

The DELTA will be located at Caltech in the spring of 1991. Argonne users will access the system through Argonne's ESnet computer network node. The Mathematics and Computer Science Division will serve as the liaison between Caltech and Argonne users.

To enable Argonne researchers to get an early start on preparing calculations for the new machine, the Mathematics and Computer Science Division conducted an intensive workshop on the new system in late 1990. The workshop focused on the innovative architecture of the Touchstone system, parallel programming techniques for transporting code to the Touchstone, a procedure for predicting how well a code will scale from one processor to the multiprocessor DELTA machine, and methods for writing efficient code in parallel programming notation and in assembly language. Workshop participants were also given laboratory time to test examples on the Intel iPSC/860, which will be used in the ACRF as a staging platform for the DELTA.

2.3 New Deputy Scientific Director

William Gropp has been appointed deputy scientific director of the ACRF. In his new position, Gropp will assist the scientific director in overseeing the ACRF research activities. Gropp joined the Mathematics and Computer Science Division at Argonne in 1989. His principal research interest is domain-decomposed methods for partial differential equations, with particular attention to their performance on parallel computers. He is also studying the scalability of numerical methods for global climate models.

3 Research in Parallel Computation

Since its establishment in 1984, the ACRF has encouraged research in algorithm design, parallel programming languages, and new techniques and tools to make parallel computing easier and more efficient. Here we highlight recent achievements that promise to make an impact on scientific problem solving. For reports of previous work, see [1]-[5].

3.1 Algorithm Design

Chris Bischof, Peter Tang, and C.-T. Pan developed a stable algorithm for maintaining Cholesky factors of symmetric positive definite matrices under arbitrary rank-1 changes. Unlike traditional downdating algorithms, the new algorithm has the ability to overlap different updates; it is also more stable than the standard variants of the Givens algorithm and hyperbolic Householder
transformations. The new algorithm has the added advantage that it is well suited for implementation on systolic arrays and single-instruction stream multiple-data stream (SIMD) architectures. Implementation results on the 1,204-processor DAP-510 emphasize the simplicity and practicality of the new algorithm.

Magnus Ewerbring, our 1989-90 Wilkinson Fellow, devised a new method for finding the rotations for a product of three matrices. Using backward error analysis, he proved that each updated matrix is upper triangular relative to the machine precision times its norm. Further, the product is diagonal in the sense that no off-diagonal element is larger than the relative machine precision times the product of the norms of the data matrices. The new method has given excellent numerical results on the Connection Machine and appears to be well suited for parallel computation.

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

William Gropp and D. E. Keyes investigated the effectiveness of domain-decomposed preconditioned Krylov methods for PDEs with adaptive refinement. With careful consideration of the preconditioner, cumulative speedups are obtainable out to at least medium-scale granularity (up to 64 processors in the tests reported). The largest problems they tested involve about $O(10^5)$ unknowns partitioned into $O(10^3)$ subdomains and converge in tens of iterations.

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

Ewing Lusk and J. K. Slaney devised a parallel algorithm for computing the closure of a set under an operation. This type of computation has been used in automated theorem proving, abstract algebra, and formal logic. The algorithm is particularly suited for shared-memory parallel computers.

Jorge Moré and T. Coleman developed an efficient parallel implementation of the Levenberg-Marquardt algorithm for solving nonlinear least-squares problems. This implementation introduces several new parallel matrix factorization algorithms and demonstrates good efficiency for large-scale problems. They also extended this work to large sparse systems. A feature of the new parallel sparse orthogonal factorization algorithm is that the data structures used for the factorization can be reused in the trust-region problem solution. This common data structure greatly increases the efficiency of an implementation.

Barry Smith, our 1990-91 Wilkinson Fellow, formulated a domain decomposition algorithm for elliptic problems in three dimensions. He showed that the condition number of the resulting preconditioned problem is bounded independently of both the number of the subdomains and the jumps in the coefficients of the differential equation between subdomains. The new algorithm also has more potential parallelism than the iterative substructuring methods previously proposed for problems in three dimensions.
Stephen Wright developed parallel algorithms for the solution of linear-quadratic optimal control problems. The algorithms are based on a straightforward decomposition of the problem domain and are related to multiple shooting methods for two-point boundary value problems. Their arithmetic cost is approximately twice that of the serial dynamic programming approach; however, the new algorithms have the advantage that they can be efficiently implemented on a wide variety of parallel architectures.

3.2 Schemes for Efficient Parallelization

Chris Bischof and B. Karp developed a system for increasing the granularity of parallelism and reducing contention in automatic differentiation. The new system builds on the automatic differentiation package ADOL-C developed at Argonne. ADOL-C produces a tape, or trace, of the computational function. Once the computational graph has been generated, hoisting is used to eliminate chain constructs and improve the granularity and make the graph more suitable for parallel processing. Next, splitting (breaking a node into several nodes) is used to reduce the likelihood of contention. The value of hoisting and splitting in solving scientific problems was demonstrated on the shallow-water equations, which are important in climate modeling. The graph produced from ADOL-C's tape was 108,686 nodes; after hoisting and splitting, the graph contained only 70,896 nodes. Clearly, one-node “hot spots” of contention had been eliminated. The new technique, while problem dependent, is likely to provide a useful base for the construction of an efficient parallel implementation of automatic differentiation.

Wayne Cowell helped develop VecPar.77, a precompiler tool that analyzes and transforms Fortran 77 programs intended for execution on machines with vector or shared-memory multitasking capabilities. In particular, the tool provides support for the task of modifying existing programs, written originally for serial execution, so that they perform well on advanced-architecture machines. Recently VecPar.77 was used to help locate data dependencies in an adaptive multigrid code for the solution of the Navier-Stokes equations for steady two-dimensional incompressible flows. A direct descendant of Toolpack, VecPar.77 was created by a project jointly supported by the Numerical Algorithms Group, Inc., and the Department of Commerce and Community Affairs of the State of Illinois.

Ian Foster developed techniques for the automatic generation of self-scheduling parallel programs. A high-level language is used to express both concurrent components of applications and scheduling algorithms. Partitioning and data dependency information are expressed by simple control statements. An automatic source-to-source transformation takes application code, control statements, and scheduling routines and generates a new program that can schedule its own execution of a parallel computer. The approach has several important advantages compared to previous proposals. It generates programs that are portable over a wide range of parallel computers. There is no need to embed special control structures in application programs. Finally, the use of a high-level language facilitates the development, modification, and reuse of parallel programs.

Foster also developed an asynchronous garbage collector for message-passing multiprocessors. This combines Weighted Reference Counting interprocessor collection and tracing intraprocessor collection to permit individual processors to reclaim local storage independently. A novel feature is the integration of Weighted Reference Counting collection and the communication algorithms required to support a global address space in a single assignment language. This significantly reduces communication overhead and space requirements attributable to garbage collection.
Ewing Lusk and R. Butler radically overhauled the “Argonne macros” system for low-level portable programming of parallel computers. The C library was completely rewritten to enhance portability, simplicity, and efficiency. The new system, called p4, has been ported to a wide set of machines, including the Intel iPSC/860, the Sequent Symmetry, the Encore Multimax, the BBN TC2000, Sun workstations, the AT&T 3b2, and the IBM RISC workstation. A reference manual has been written, and a tutorial document is under way.

Ewing Lusk and Ian Foster completed work on a variety of performance visualization tools. Atrace—which comprises tools for displaying logfiles as a set of parallel timelines, as an animated set of communicating processes, and as a set of statistical graphs—was integrated with the Strand parallel programming system and used to tune the parallel automated theorem prover ROO. Foster also integrated the Gauge performance profiling tool (originally developed in the context of logic programming systems) into the PCN programming environment.

3.3 Programming Languages for High-Performance Computers

James Boyle and T. Harmer explored the usefulness of functional programming for specifying numerical computations. Usually, procedural languages are used for such computations. The researchers demonstrated, however, that efficient programs can be implemented from high-level functional specifications—and implemented automatically. As a test case, they prepared a pure functional specification for an algorithm that solves one-dimensional hyperbolic partial differential equations. Since algorithms such as this are designed to solve large problems, including the “Grand Challenges,” slow execution cannot be tolerated. Moreover, in many cases the scientist wishes to use such algorithms to exploit novel advanced-computer architectures. The Argonne researchers demonstrated that, by using automated program transformations, they were able to derive a Fortran program that executes faster on a CRAY X-MP than does the handwritten Fortran implementation of the same one-dimensional cellular automaton algorithm.

Ian Foster and Ross Overbeek demonstrated the use of a high-level concurrent programming language (such as Strand) to construct parallel programs from (possibly pre-existing) sequential components. In an applications study, they developed new codes and parallel versions of existing codes in computational biology, weather modeling, and automated reasoning. Their results show that the bilingual approach encourages the development of parallel programs that perform well, are portable, and are easy to maintain.

Ewing Lusk and his colleagues developed a system called Aurora, as part of an informal research collaboration known as the Gigalips Project. The new system was constructed by adapting Sistus Prolog (a fast, portable, sequential Prolog system) for or-parallel operation. The techniques for constructing a portable multiprocessor version follow those pioneered in a predecessor system, ANL-WAM. For a range of benchmarks, Aurora on a 20-processor Sequent Symmetry is 4 to 7 times faster than Quintus Prolog on a Sun 3/75. Good performance is also reported on some large-scale Prolog applications.

3.4 Software Packages

Chris Bischof studied fundamental linear algebra computations on high-performance computers as part of his work on the LAPACK library. This library provides a uniformly designed set
of subroutines for solving systems of simultaneous linear equations, least-squares problems, and
eigenvalue problems for dense and banded matrices. To make the library efficient on today’s high-
performance computers, the designers are using block algorithms and the Basic Linear Algebra
Subprograms (BLAS). Performance results show that the LAPACK approach is suitable for vector
uniprocessors and shared-memory multiprocessors.

Jim Cody tested the floating-point arithmetic and the elementary function libraries under AST
Fortran on a 24-processor Sequent Symmetry computer. The programs MACHAR and PARANOIA
were used to check the quality of arithmetic, and the ELEFUNT suite of programs from the book
Software Manual for the Elementary Functions by Cody and Waite was used to check function
performance. Two complete sets of tests were run, one for each type of floating-point processor,
Intel 80387 and Weitek 1167.

Mark Jones and M. L. Patrick designed and implemented a package called LANZ for solving
the large sparse symmetric generalized eigenproblem. The package uses a version of the Lanczos
method and is based on recent research into solving the generalized eigenproblem. It has been
successfully tested on four different architectures: Convex 200, CRAY Y-MP, Sun-3, and Sun-4.

3.5 Parallel Programming Systems

Ewing Lusk constructed a parallel automated reasoning system based on Argonne’s sequential
automated reasoning system OTTER and the parallel closure algorithm developed last year. The
resulting system, called ROO, was implemented using the parallel programming system p4. Using
ROO, Lusk obtained substantial speedups (as high as 22 on a 24-processor Sequent Symmetry) on
a wide range of problems.

Ross Overbeek and Ian Foster, together with their colleagues R. Butler and A. Jindal,
implemented a parallel theorem proving system that supports hyperresolution, subsumption, and
the set of support strategy. The upper level of the program, written in the concurrent program-
ming language Strand, implements the general algorithm and describes the interconnectivity of
processes that perform the theorem-proving tasks such as resolution and subsumption. These com-
putationally intensive tasks are performed by procedures implemented in C. Speedups have been
demonstrated on such test problems as SAM’s lemma.

3.6 Climate Modeling

Climate modeling is one of the Grand Challenges identified by the Department of Energy. The
goal of DOE-supported research is to provide a feasible climate modeling environment based on
advanced computer technology and state-of-the-art numerical techniques. Argonne researchers
have been involved specifically in the Computer Hardware, Advanced Mathematics, and Model
Physics (CHAMMP) project in several ways and have achieved notable successes:

- I-Liang Chern investigated parallel numerical algorithms for PDEs on the sphere. He
developed a control volume method that is linearly scalable on parallel machines. The well-
known pole problem is avoided by the use of a three-dimensional Cartesian coordinate system
to represent the equations and by the use of an icosahedral grid on the sphere.
• Ian Foster and F. McCabe constructed several versions of a Fast Fourier Transform—one in Strand, another in C, and two in a mixture of both Strand and C. For the bilingual programs, the C portions use destructive assignment; copying is used between sections to ensure adequate separation. With this strategy, it proved very easy to generate correct programs.

• Ian Foster and Rick Stevens studied the issue of load distribution. The amount of physics computation at different grid points in a global climate model can differ significantly depending on the values of physical variables. Irregular distribution of computational requirements is a potential source of inefficiency in a parallel model. To quantify these load imbalances, they conducted an experimental study using an existing model, CCM1. Significant static and seasonal load imbalances were observed (typically a factor of two variation between grid cells). They also began investigating several techniques for distributing physics computations to the nodes of a parallel computer in order to correct load imbalances. These include simulated annealing and simpler "greedy" algorithms based on run-time data collected by their high-level programming tools.

• Nicholas Karonis worked on the meso-scale weather model MM4, parallelizing Fortran modules of existing weather codes. The parallelization is done by developing a bilingual programming environment in which PCN code "rests" over the Fortran code. PCN is used to orchestrate—at a coarse level—and parallelize the algorithm. The modeling and computationally intensive portions of the program are executed by the Fortran modules.

• James Kohl helped develop versions of various climate modeling algorithms for execution on parallel computer architectures. He began with a (500-line) parallel version of the barotropic vorticity equations, using the high-level parallel language Strand in conjunction with existing Fortran source code. To reduce the number of interprocessor communications, Kohl and his colleagues developed a new structure that improved performance by about 10%. Next, Kohl parallelized the (5000-line) Shallow Water Equations (SWE) code, developed a graphics display of the data output, and wrote a collection of shell scripts for collecting performance analysis data from the various versions of SWE for different problem sizes.

• Rick Stevens, Ian Foster, and D. Joerg converted the MM4 Fortran code to work on machines other than the Cray. They achieved a relatively portable code which has been run successfully on the Symmetry. The NeXT was used, with Mathematica, to interpret and display output data in various plot forms.

Several of our visitors also worked on the CHAMMP project:

• Amy Crook and Bill Crosbie explored techniques for visualizing data from the Cray and the shallow water equations. They used Doré on the Stardent Titan to create an interactive demonstration package that displays the load-balance data as mapped over a globe. They also made a video of the data being animated through several timesteps.

• Ben Sussman and John Michalakes conducted a preliminary analysis of the balance of calculation across multiple processors (hypothetically running the Cray Climate Model (CCM)). In addition, they used Mathematica to produce relevant graphs, charts, and animations of the data, to be used by other researchers planning concrete load-balancing strategies.
Jeffrey Jackson worked on parallelizing the Composite Mesh Climate Model. In the model, the northern and southern hemispheres are projected onto planes, each tangent at a pole. This plane projection alleviates the problem of infinitely small regions as latitudes approach the poles. The planes are broken into multiple charts and distributed to separate processors. The parallel distribution and communication is handled in Strand, while the numerical methods and algorithms are written in C.

4 Collaborative Projects with Other Research Centers

The MCS Division receives special funding from the National Science Foundation and from the U.S. Army to conduct collaborative research in parallel computing.

4.1 Center for Research in Parallel Computation

Argonne continues to participate in the National Science Foundation Science and Technology Center for Research in Parallel Computation (CRPC). During the reporting period, CRPC funds were used to upgrade some of the ACRF equipment and to support basic research in parallel computing, including the following:

- PCN. Working with researchers at Caltech and the Aerospace Corporation, we designed and constructed a prototype implementation of the concurrent programming language PCN. We evaluated the utility of this language in a number of applications, including global climate modeling and computational biology.

- Parallel ADOL-C. We conducted a pilot project aimed at using the reverse mode to achieve an efficient parallel implementation of the ADOL-C automatic differentiation package. On the Helmholtz energy function with 300 variables, we obtained a speedup of 11 on 15 processors of a Sequent Symmetry. This result shows that reasonable speedups can be obtained even without transformations to improve the granularity of the gradient evaluation.

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

Staff from the MCS Division also serve on the CRPC Technical Committee, the Facilities Committee, and the Executive Committee to help guide the center in its study and development of efficient programming methods for state-of-the-art parallel computers.

4.2 Army High Performance Computing Research Center

The MCS Division is also collaborating with the Army High Performance Computing Research Center at the University of Minnesota to produce algorithms and optimization software for high-performance computers and to develop a collection of significant optimization problems that will
serve as test problems for the MINPACK-2 package. During the first three months of this project, we have made two major accomplishments:

- Design of a specialized version of the LMDER code for nonlinear least squares problems found in MINPACK-1. The new code contains an improved version of the QRSOLV subroutine to compute the Levenberg-Marquardt parameter. Excellent speedups are obtained even for small problems. For example, on a problem with \( n = 128 \), the speedup for 8 processors is nearly 7 on the Alliant FX/8. Comparisons with the old version of QRSOLV are also favorable; the new subroutine shows an improvement by a factor of more than 5 on the Alliant FX/8.

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

5 Publications, Preprints, and Presentations

One important measure of our scientific impact is the publication of papers and the presentation of our results to the research community. Listed below are publications, preprints, and presentations relating to advanced computing research conducted using the ACRF machines during 1990.

Publications in Journals and Conference Proceedings


I. Foster and R. Overbeek, "Experiences with Bilingual Parallel Programming," Proc. 5th Conf. on Distributed Memory Computers and Applications, IEEE, 1990, pp. 1137-1146


F. B. Hanson and D. C. Sorensen, "The SCHEDULE Parallel Programming Package with Recycling Job Queues and Iterated Dependency Graphs," Concurrency: Practice and Experience 2, no. 1 (March 1990) 33-54


on the Numerical Solution of Markov Chains, North Carolina State University, Raleigh, 1990


S. J. Wright, "Implementing Proximal Point Methods for Linear Programming," *J. Optimization Theory and Appl.* 65, no. 3 (June 1990) 531

S. J. Wright, "Solution of Discrete-Time Optimal Control Problems on Parallel Computers," *Parallel Computing* 16 (1990) 221-238 (also MCS-P89-0789)

A. C. Yu and B. W. Wah, "Design and Evaluation of a Pre-Compiler for Multiprocessing of Lisp Programs," *Proc. Triennial Congress on Cybernetics and Systems*, June 1990

Preprints

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


I. Foster, "Automatic Generation of Self-Scheduling Programs," MCS-P143-0390


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

I. Foster and R. Stevens, "Bilingual Parallel Programming," MCS-P163-0790

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

S. J. Wright, "Partitioned Dynamic Programming for Optimal Control," MCS-P173-0890

S. J. Wright, "Stable Parallel Algorithms for Two-Point Boundary Value Problems," MCS-P178-0990

M. T. Jones and M. L. Patrick, "The Lanczos Algorithm for the Generalized Symmetric Eigenproblem on Shared-Memory Architectures," MCS-P182-1090


B. F. Smith, "A Domain Decomposition Algorithm for Elliptic Problems in Three Dimensions," MCS-P185-1090


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


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


Reports


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


Oral Presentations

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


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

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


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


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

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

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

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


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

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

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

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


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


I. Foster, "The PCN Run-Time System," CRPC Research Symposium, Rice University, Houston, Texas, August 28, 1990

I. Foster, "Parallel Programming in PCN," Oak Ridge National Laboratory, Oak Ridge, Tenn., October 26, 1990

I. Foster, "Towards a Tera-op Climate Model," Caltech, Pasadena, Calif., December 5, 1990

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


W. Gropp, "The Impact of Parallel Computing on Algorithms for PDEs on the Sphere," Ar-


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

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


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


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


E. Lusk, "Portable Parallel Computing and Applications," Technical University of Munich, November 17, 1990
E. Lusk, "High-Performance Computing Applications," Australian National University, December 15, 1990


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


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


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


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


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


P. Wohl, "SIMD Neural Net Mapping on MIMD Architectures," International Conference on Parallel Processing, August 1990

Bulletin

For our ACRF users, we continue to produce a periodic bulletin called Adventures in Parallelism. From February 1990 through April 1991, we mailed out four issues, ranging from eight to twelve pages. Our objective is to let users know of new projects, equipment, and research results in parallel computing associated with our ACRF.
6 The ACRF—A User Facility

The ACRF was established to be used for advanced computing research by researchers throughout the world. To encourage use of the ACRF's specialized resources, the Mathematics and Computer Science Division conducts classes; supports an active visitors program; and sponsors a variety of seminars, workshops, and institutes.

6.1 Classes

Argonne conducted bimonthly classes on parallel computing during the past year. The attendees, averaging 20-25 persons per class, represented universities, industry, and various research laboratories throughout the world. The classes familiarize the attendees with the ACRF environment, offer ample hands-on experience on the parallel computer systems, and help the attendees apply parallel programming to their particular area of research. Session topics include parallelizing compilers, the p4 package for portable parallel programming, introduction to the LAPACK project, the Strand programming language, and the VecPar.77 interactive parallelization tool.

In addition to these general classes in parallel programming, the ACRF staff offered several special classes, including writing parallel programs in Strand, programming the Connection Machine, programming the DAP, using Mathematica, and converting code for the Intel Touchstone architecture.

6.2 Proposals

One of the objectives of the ACRF classes is to encourage researchers to use our advanced computers. Frequently, after attending a class, researchers wish to continue using the ACRF machines. During 1990, approximately 400 proposed projects were approved by the ACRF staff. The two pie charts below depict the type of project approved and the broad affiliation of the users.
6.3 Minority Internship in Parallel Computing

Under a Minority Internship in Parallel Computing funded by the National Science Foundation, Amos Joel Carpenter (Butler University, Indiana) is spending a year in the Mathematics and Computer Science Division at Argonne. The internship is intended specifically for a candidate who is interested in acquiring knowledge and experience in the area of parallel processing.

Since his arrival late this summer, Carpenter has familiarized himself with Argonne’s p4 package and has been using that package to parallelize his codes on the Alliant and the Encore.

6.4 Visitors

During the past year and a half, we have had numerous visitors here working on projects involving computer graphics, high-performance algorithms, and software tools for advanced-architecture computers. These visitors, who include scientists from industry and academia, stay for periods ranging from two weeks to several months.

Special Term Appointments

Brett Averick
University of Minnesota
MINPACK-2

Richard Carter
University of Minnesota
MINPACK-2

NSF Minority Internship in Parallel Computing

Amos Carpenter
Butler University
uniform polynomial approximations

Other Visitors

K. Ariyanwansa
Washington State University
algorithms and software for stochastic linear programs with recourse

James Bordner
University of Illinois at Urbana
parallel program for solving partially separable functions

Amy Crook
Illinois Mathematics and Science Academy
visualizations of load-balancing data
LeRoy Drummond  
University of Tulsa  
message-passing tools on the BBN TC2000 system

Michele Evard  
Andrews University  
graphical user interface prototype for community climate models

Remy Evard  
Andrews University  
parallel climate modeling

Eugene Foss  
Illinois Mathematics and Science Academy  
visualization of global climate data

Eric Fraser  
The Evergreen State College  
simulated annealing experiments  
dynamic and static load-balancing methods for GCMs

Marc Garbey  
Lyon, France  
hypercube implementation of domain decomposition techniques

James Garnett  
University of Texas at Austin  
constructing a library of manager/worker schedulers for the Strand compiler

Theodore Gaunt  
University of Michigan, Ann Arbor  
parallel implementations of algorithmic differentiation techniques

Steven Hammond  
University of Maine, Orono  
programmable transformation tool for parallel programs

Floyd Hanson  
University of Illinois, Chicago  
computational stochastic dynamics

Terence Harmer  
The Queen's University of Belfast  
abstract programming and efficient implementations on the CRAY

Virginia Herrarte  
Blackburn College  
performance visualization tools for parallel processors
James Hu
Kansas State University
Generation of Computational Graphs from ADOL-C Software

Jeffrey Jackson
Southwest Baptist University,
parallelizing the Composite Mesh Climate Model

David Joerg
Illinois Mathematics and Science Academy
conversion of weather model for the Symmetry

Nicholas Karousis
Syracuse University
visualization of distributed algorithms
parallelization of weather modeling modules

Brad Karp
Yale University
graph transformations of automatic differentiation on parallel machines

Carl Kesselman
Aerospace Corp.
load-balancing strategies for a parallel general circulation model

James Arthur Kohl
University of Iowa
development of climate modeling algorithms

Kathy Kong
California State Polytechnic University, Pomona
performance analysis of the Intel Touchstone systems

Marco Lapenga
University of Napoli
mathematical software for multidimensional quadrature

Frank McCabe
Imperial College
mixed programming language environments

James L. Mosley
Indiana University/Purdue at Fort Wayne
parallelization of the Community Climate Model
J. Robert Neeley  
State University of New York at Plattsburgh  
parallel model physics algorithms, parallel algorithms for spectral transform methods

Peter Nigro  
SUNY at Stony Brook  
measurement of message-passing performance using PCN and Strand

Ching-Tsuan Pan  
Northern Illinois University  
updating algorithms for Cholesky factors of symmetric positive definite matrices

Avijit Purkayatha  
Northern Illinois University  
parallel algorithm for Sylvester observer equation

Deborah Stevens  
Northwestern University  
visualization of climate model codes

Ben Sussman  
Oak Park High School  
load-balancing strategies and data visualization

Michael Wassmer  
LaFayette College  
parallelization of the composite mesh code

6.5 Seminars

The MCS Division continues to sponsor a series of seminars on high-performance computing. Below are listed the names of our speakers from February 1990 through April 1991, including their affiliation and the title of their talk.


Lars Elden, *Iterative Solution of Non-Symmetric Systems Arising in Hyperbolic PDEs*, Linkoping University, Sweden, May 4, 1990
Dan C. Marinescu, *Communication and Control in SPMD Parallel Numerical Computations—The E/T Model*, Purdue University, May 10, 1990


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

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


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


L. V. Kale, *The Chare Kernel Parallel Programming System*, University of Illinois at Urbana, March 4, 1991

Prasanna Job, *Parallelization of Probabilistic Sequential Search Algorithms*, DePaul University, April 4, 1991

Amos Carpenter, *Asymptotics for the Zeros and Poles of the Normalized Pade to e^z*, Butler University, April 24, 1991

Bret Michael, *Formalization of Policy Systems*, George Mason University, April 26, 1991


7 Affiliates Programs

We have an active affiliates program for both universities and industrial organizations interested in using the ACRF multiprocessing systems.

7.1 Academic Affiliates

The Academic Affiliates Program is extended to a limited number of colleges and universities interested in high-performance computing. Currently, 28 colleges and universities are enrolled.
| Australian National University | State University of New York at Albany |
| University of Alabama at Birmingham | Texas Christian University |
| Andrews University | University of Chicago |
| California Institute of Technology | University of Illinois - Chicago |
| Cornell University | University of Illinois - Urbana-Champaign |
| Drexel University | University of Iowa |
| John Jay College of Criminal Justice | University of Kentucky |
| Marquette University | University of Michigan |
| Memorial University, Newfoundland | University of Tennessee |
| Michigan State University | University of Tulsa |
| Northern Illinois University | University of Virginia |
| Northwestern University | Vanderbilt University |
| Oregon State University | Western Michigan University |
| | Youngstown State University |

### College and University Courses

Several of our university affiliates used the ACRF machines for courses in advanced computing. In particular, we note the following:

- **Duke University** – Prof. A. Lastra used the Connection Machine for a graduate-level programming class
- **University of Chicago** – Prof. C. Owens used the Connection Machine for a graduate course entitled “Topics in AI”
- **University of Illinois at Chicago** – Prof. F. Hanson used the Alliant for a graduate course entitled “Introduction to Supercomputing”
- **University of Illinois at Chicago** – Prof. S. Roy used the Alliant for a Workshop Program on Scientific Supercomputing
- **University of Nebraska at Lincoln** – Prof. Samal used the Connection Machine and the BBN TC2000 for a course in operating systems
- **University of Tennessee** – Prof. J. Dongarra used the BBN TC2000 for a graduate course on parallel computation
- **University of Washington** – Prof. S. Eggers used the Sequent and Encore for a graduate seminar in advanced compiler optimization
- **Vanderbilt University** – Prof. C. Fischer used the Alliant for an undergraduate course on parallel and concurrent Fortran for engineers

### 7.2 Industrial Affiliates

To complement our Academic Affiliates Program, we have established an Industrial Affiliates Program. The purpose of this program is to provide industry with access to a variety of experimental machines and to bring together users and manufacturers of parallel processors.
To date, we have 15 associate and affiliate members:

<table>
<thead>
<tr>
<th>Active Memory Technology</th>
<th>MasPar Computer Corporation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alliant Computer Systems</td>
<td>Mercury Computer Systems</td>
</tr>
<tr>
<td>BBN Advanced Computer Systems</td>
<td>NeXT, Inc.</td>
</tr>
<tr>
<td>Convex Computer Corporation</td>
<td>Phillips Petroleum</td>
</tr>
<tr>
<td>Encore Computer Systems</td>
<td>Sequent Computer Systems</td>
</tr>
<tr>
<td>Epoch Systems</td>
<td>Stardent Computer Corporation</td>
</tr>
<tr>
<td>Intel Scientific Computers</td>
<td>Thinking Machines Corporation</td>
</tr>
<tr>
<td>Litton Data Systems</td>
<td></td>
</tr>
</tbody>
</table>

7.3 Affiliates Workshops

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

The success of this workshop encouraged us to make such a meeting an annual event. Accordingly, the second affiliates workshop was held on April 25–26, 1991. Nineteen of our affiliates attended, from as close as Chicago and as far away as Australia. Presentations focused on the Intel Touchstone DELTA System project, parallel computational control, software packages for heterogeneous network environments, stable updating and downdating Cholesky algorithms, the Epoch-1 InfiniteStorage Server, schemes for array processors, object-oriented applications programming, parallelization tools for the BBN, MasPar's new programming environment, tools for the Solbourne, classroom use of the Alliant, and orthogonal spline collocation. Rick Stevens concluded this second workshop with an overview of software and hardware trends and a glimpse at Argonne's plans for expanding the ACRF.

8 High-Performance Computing Activities

In addition to the Affiliates Workshops, the ACRF sponsored several other activities that focused on high-performance computing.
8.1 Summer Institute

For the fourth time, we held a **Summer Institute in Parallel Programming**. Twenty-five graduate students and postdoctoral researchers spent two weeks at Argonne, from September 4 to 14, 1990, learning about recent advances in parallel computing and experimenting with new ideas on state-of-the-art machines.

A central feature of the Institute was the invited lectures by leading researchers in advanced computing: Arvind (MIT), Jay Batson (BBN Advanced Computers, Inc.), Mani Chandy (Caltech), Jack Dongarra (Univ. of Tenn./ORNL), Josh Fisher (Hewlett Packard Laboratories), David Gelernter (Yale University), Paul Huray (University of South Carolina), Ken Kennedy (Rice University), Justin Rattner (Intel Scientific Computing), and Guy Steele (Thinking Machines Corp.). Each invited speaker conducted a two-and-a-half-hour session, covering such topics as instruction-level parallelism, Nonuniform Memory Architectures (NUMA), implicitly parallel programming, dataflow architectures, programming support environments the evolution of linear algebra software, the Linda programming language, and federal directions in high-performance computing.

ACRF staff supplemented the invited presentations with classes on parallelizing compilers, program transformations, and the programming languages Strand and PCN. The staff also helped researchers in developing parallel programs and in using programming tools to improve program performance on the ACRF machines.

The Institute in Parallel Programming was sponsored in part by the National Science Foundation and in part by the U.S. Department of Energy. The two organizations have agreed to support a similar conference in the summer of 1991.

8.2 CHAMMP-related Meetings

One of the Grand Challenges facing researchers is climate modeling. To address this challenge, Argonne is participating in the CHAMMP project. Two workshops were held during 1990.

The **Performance Visualization Workshop**, held in July, focused on performance visualization techniques for large-scale parallel computers. Groups currently working in the area of performance visualization met to survey the state of the art and to discuss techniques that would scale up to massively parallel machines. Topics included multimedia presentations, three-dimensional modeling, scaling to 100-K processors and beyond, hardware support, application-specific visualization, hardware performance vs software performance, performance prediction, and architecture independence. A report summarizing the state of the art in parallel performance visualization, and the views of the attendees on future directions, is being prepared.

The second workshop, held during August, was concerned with **Numerical Methods for PDEs on the Sphere**. Participants were from a variety of disciplines, including climate modeling, reactor modeling, applied mathematics, and parallel computing. The program included thirteen talks, covering both implementations and numerical methods, and generated much discussion concerning future directions for parallel climate modeling.