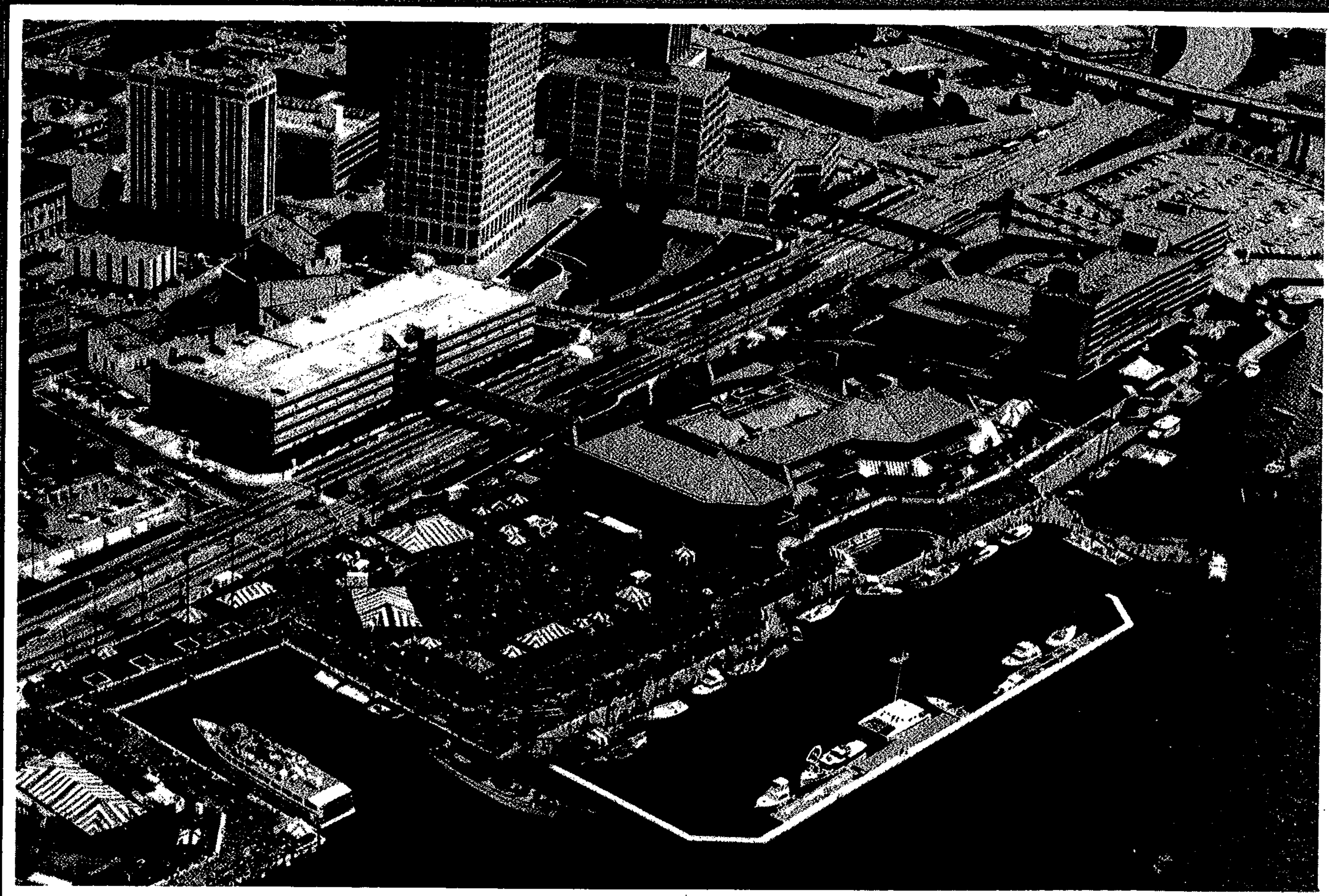
Second SIAM Conference on Parallel Processing for Scientific Computing

Norfolk, Virginia



Sponsored by the SIAM Activity Group on Supercomputing

TABLE OF CONTENTS

ORGANIZING COMMITTEE

Robert G. Voigt (Chairman), ICASE, NASA Langley Research Center

Billy L. Buzbee, Los Alamos National Laboratory

Dennis B. Gannon, Indiana University

Garry H. Rodrigue, Lawrence Livermore National Laboratory

FUNDING AGENCIES

SIAM is conducting this conference with the partial support of the Air Force Office of Scientific Research, the Army Research Office, and the National Science Foundation.

Photographs graciously provided by the Norfolk Convention and Visitors Bureau.

MEETING HIGHLIGHTS

Tutorials

Monday, November 18, 9:00 AM
Tutorial 1
SUPERCOMPUTERS: PAST, PRESENT AND FUTURE

Three driving forces are at work in supercomputer architecture—commitments, advances in technology, and research. The dominant commitment is to vector processing, and this commitment will continue with the design of multiple vector processor systems such as the Cray X-MP, Cray-2, and the ETA-10. Near term advances in supercomputer architecture will include designs using a small number (up to 16) of very fast (supercomputer speed) processing elements (PEs). Designs such as the Denelcor HEP-2 are providing a basis for exploring the part of the speed-parallelism space using somewhat slower PEs but more of them. In the research area, a province primarily of the universities, designs incorporating even slower PEs and even larger numbers of them are being studied as ways of creating economical yet high-performance supercomputers.

The speaker will review the evolution of supercomputer architecture, describe a taxonomy of current architectures, and discuss the prospects for a science of parallel computation. A rationale for parallel computation, along with some of its key issues and limitations will be presented.

Jack Worlton
Los Alamos National Laboratory
Los Alamos, NM

Monday, November 18, 10:30 AM
Tutorial 2
VECTOR COMPUTER ARCHITECTURES AND THEIR IMPACT ON COMPUTATION

The recognition that conventional von Neumann computer architectures are nearing their practical limit in computational speed has led to the development of vector (and parallel) processing computers for large, scientific applications. A vector computer relies on designed overlap and replication within its architecture to achieve peak computational rates that are orders of magnitude higher than the conventional computers of today. However, the performance realized for a program on a vector computer depends on how well the scientist and programmer have matched the algorithm and data structure to the computer architecture. The speaker will discuss the architectures of several vector computers and examine their influence on computational efficiency.

Jay Lambiotte, Jr.
NASA Langley Research Center
Hampton, VA

Monday, November 18, 11:15 AM
Tutorial 3
MULTIPROCESSOR ENVIRONMENTS: MAPPING ALGORITHMS TO ARCHITECTURES

Parallel computing systems with a large number of processors are regarded by some to be the approach to take in developing large-scale high-performance computing systems. One of the major difficulties with such systems, however, is the mapping of the algorithms onto the system so that the resources are used as effectively as possible. In particular, it is desirable to minimize communication and synchronization costs while balancing the computational load so that all processors are performing useful work as much of the time as is possible. Proper choice of algorithms can be crucial and appropriate programming language features can lead to assistance from compilers.

The speaker will review the important characteristics of parallel computing systems consisting of a potentially large number of processors, using specific examples to demonstrate the critical points.

John R. Van Rosendale
Department of Computer Science
University of Utah
Salt Lake City, UT

Monday, November 18, 1:30 PM
Tutorial 4
PARALLEL AND VECTOR ALGORITHMS—AN INTRODUCTION

For traditional computers of past years, the computational performance of an algorithm was relatively independent of the computer. Now, however, with the advent of vector and parallel computers, to achieve optimum performance, the design of an algorithm has become dependent on the computer architecture.

The speaker will review a variety of numerical algorithms now available for vector and parallel computers. The primary problem areas covered will be the direct and iterative solution of systems of linear equations, with emphasis on problems arising from the discretization of partial differential equations. Techniques to be discussed include factorization methods, SOR and related methods, and preconditioned conjugate gradient methods. The similarities and the differences in the design of algorithms for vector and parallel computation will be emphasized.

James M. Ortega
University of Virginia
Charlottesville, VA

Monday, November 18, 3:00 PM
Tutorial 5
FACTORIZATION METHODS IN LINEAR ALGEBRA

Modern techniques in computational linear algebra are based on a decompositional approach. The general idea is the following: Given a problem involving a matrix, one factors or decomposes the matrix into a product of simpler matrices from which the problem can easily be solved. The approach to matrix computations through decompositions has turned out to be quite fruitful. However, such algorithms as implemented on conventional machines may not be well suited to providing peak performance on advanced architectures.

Architectures of future machines promise to offer a profusion of computing environments. In implementing the various decomposition algorithms on new architectures, it is desirable to minimize the effort involved while maintaining certain qualities in the algorithm, such as portability and high levels of performance. It seems to be an unnecessary waste of effort to recast these algorithms with only one computer in mind.

The speaker will examine some common factorization methods used to solve problems arising in linear algebra and discuss how these methods can be implemented in a portable fashion on a wide range of high-performance computers without sacrificing efficiency.

Jack J. Dongarra
Argonne National Laboratory
Argonne, IL

Monday, November 18, 3:45 PM
Tutorial 6
ITERATIVE METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS

The discretization of elliptic partial differential equations often leads to a system of linear equations that must be solved numerically. The speaker will examine the iterative solution of such a system of equations on both vector computers and multiple processor systems. The iterative methods to be discussed are point and block Jacobi and multi-color SOR and preconditioned conjugate gradient methods. She will also show how the techniques for the parallel implementation of such methods apply to the parallel solution of initial boundary value problems using explicit methods and implicit methods such as ADI.

Loyce Adams
Department of Computer Science
University of Washington
Seattle, WA

Monday, November 18, 4:30 PM
Special Lecture
THE NSF SUPERCOMPUTER INITIATIVE

In a concentrated effort to keep academic research competitive with the rest of the world, NSF has launched a major initiative in providing supercomputer access to university scientists and engineers.

The ultimate goal is to create a nationwide network of supercomputer centers, which will be provided with state-of-the-art computational equipment. Access time will be allocated to researchers on the basis of scientific merit.

Recently, NSF took the first step toward achieving this goal by selecting the first four sites for these centers—San Diego, Illinois, Cornell and Princeton. Two hundred million dollars have been committed to support these centers over the next five years. The NSF initiative is expected to have a major impact on: the solution of basic research problems not possible on current academic computers; the training of students and new researchers in the use of advanced computational instrumentations; and the stimulation of the computer industry to develop even better supercomputers in the future.

John W. D. Connolly
Director
Office of Advanced Scientific Computing
National Science Foundation
Washington, DC

Monday, November 18, 5:00 PM
Special Lecture
NCSA: AN INTERDISCIPLINARY RESEARCH CENTER BUILT ON AN INTEGRATED SUPERCOMPUTING ENVIRONMENT

The University of Illinois National Center for Supercomputing Applications is creating a comprehensive computing environment. The high speed network will connect a Cray X-MP with large memory Solid State Disk, trillion bit on-line mass store, high volume central output facilities, and over one hundred PCs and workstations. The latter will be on every desk in a new kind of scientific research center. Here some 40 top science and engineering users of the computational system will interact with 20 support computer professionals and staff researchers. Our goal is to identify and remove common bottlenecks with today hinder progress in scientific computing. Our center will work closely with the Center for Supercomputing Research and Development, Directed by David Kuck, on code optimization and new algorithms for multiprocessors.

Larry L. Smarr
Director
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign
Urbana, IL

Research Presentations

Tuesday, November 19, 8:30 AM
Research Presentation 1
THE "ULTRACOMPUTER"—AN EXPERIMENTAL SHARED MEMORY MIMD COMPUTER

The NYU Ultracomputer is based on an experimental architecture featuring thousands of processors sharing a multigigabyte common memory so as to cooperate on solving problems too large for today's computers.

Extensive simulation studies indicate that an "Ultracomputer" having the component count of present day large machines, but constructed using 1990 technology, would offer orders of magnitude more performance than current vector supercomputers and would be much more readily applicable to a wide range of important computational problems.

To date, a reliable eight processor prototype has been constructed and a highly parallel operating system based on Unix* has been implemented. VLSI switches have been designed and fabricated for use in a processor-memory interconnection network; enhanced switches have been designed that will enable the network to combine simultaneous memory references to the same location.

Allan Gottlieb
Courant Institute of Mathematical Sciences
New York University
New York, NY

Tuesday, November 19, 9:15 AM
Research Presentation 2
A TESTBED FOR PARALLEL ALGORITHM DEVELOPMENT

A hardware and software system has been designed to permit easy implementation and comparison of parallel algorithms. The hardware, the "ZMOB", consists of a set of 64 parallel processors that appear to the user to be fully interconnected. The software includes pre- and post-processors to load data and unload answers, a multiprogramming system for each processor, and a snapshotting system that can stop the processors at any time to measure progress and bottlenecks. The software is designed to be transportable to other parallel environments.

The system has been used in the design and evaluation of parallel matrix algorithms on a variety of architecture (rings, 2-dimensional mesh connected arrays, etc.) with a variety of assumptions on the cost of communications. The speaker will describe the hardware and software system and its use, including experiments evaluating various algorithms, task assignments, and task schedulings.

Dianne P. O'Leary
University of Maryland
College Park, MD

Tuesday, November 19, 1:30 PM
Research Presentation 3
COMPUTATIONAL PHYSICS ON PARALLEL MACHINES

The advent of large parallel computers has stimulated much activity among those concerned with actually solving problems on such machines. The easiest route to follow in the near term is to uncover parallelism in existing collections of algorithms and recode (or perhaps, just recompile) existing codes. This approach certainly can work for some problems if the number of processing units is not too large (less than 8, for example).

A better approach is to reformulate the expressions of physical processes so as to expose the parallelism provided by nature. An example of this approach is the Independent Time Step Method wherein different regions of a problem advance asynchronously at their own rates. Code written with this method cannot be characterized by the categories explicit or implicit. With existing parallel computers, a primary concern for this method is to restrict the parallelism to levels at which the hardware can function optimally.

Peter G. Eltgroth
Lawrence Livermore National Laboratory
Livermore, CA

Tuesday, November 19, 2:15 PM
Research Presentation 4
PARALLEL ALGORITHMS FOR PARTIAL DIFFERENTIAL EQUATIONS—CHARACTERISTICS AND DEVELOPMENT TECHNIQUES

Many of the physical problems which are important enough to spur the development of highly parallel supercomputers require partial differential equations in their mathematical models. In fact, we can expect that a significant portion of the running time on these machines will be spent on the numerical solution of systems of partial differential equations. It follows that we will want to use algorithms which construct numerical solutions as efficiently as possible on the parallel architecture in use.

Over the past twenty years, performance gains due to the development of faster algorithms for the solution of partial differential equations have kept pace with performance gains due to the introduction of faster hardware. Will we be able to move these fast algorithms onto the new parallel architectures? Or, will we see the development of new algorithms that are even more efficient?

Paul O. Frederickson
Los Alamos National Laboratory
Los Alamos, NM

Wednesday, November 20, 8:30 AM
Research Presentation 5
ARCHITECTURE AND PROGRAMMING OF THE COSMIC CUBE

The cosmic cube is an experimental concurrent computer that consists of 64 small "node" computers connected by queued communications channels in the plan of a binary 6-cube. Its design was based on a message-passing computational model that respects the communications limitations of high-complexity microelectronics. The programming style used for the cosmic cube—an adaptation of object-oriented programming—reflects this same concern for locality. A computation is formulated as a collection of communicating processes (objects) that are distributed through the nodes. The message primitives provided by the cosmic cube's resident operating system employ a weak synchronization with message order preservation, and have proved to be both convenient and powerful. The presentation will include a discussion of the computer and examples of concurrent formulations and programs for several scientific applications.

Charles L. Seitz
California Institute of Technology
Pasadena, CA

Wednesday, November 20, 9:15 AM
Research Presentation 6
SUPRENUM—THE GERMAN MULTIGRID SUPERCOMPUTER PROJECT

The goal of the Suprenum project is to develop a high-performance computer for large-scale scientific computation ("numericher superrechner"). It is agreed that the corresponding architecture should be characterized by the following essentials: make substantial use of parallel processes, reflect the regular ("grid") structure of a large class of numerical applications, and support the multilevel ("multigrid") principle.

Plans for the first phase of the project (1985–1988) are to build a Suprenum prototype, the hardware of which will be MIMD-multiprocessor-system consisting of 256 processor-knots with local memory. Each 16 knots (one "cluster") will be connected by a standard industrial bus, and all clusters are to be connected by a 3-dimensional structure of buses. Considering it as a combined hardware-software system, Suprenum is to be equipped with suitable MIMD software based on parallel multilevel algorithms. The idea is to obtain full performance according to the formula: Suprenum-performance = multi-processor-performance (hardware) × multi-level-performance (software).

Ulrich Trottenberg
Gesellschaft fur Mathematik und Datenverarbeitung and University of Cologne
Federal Republic of Germany

Thursday, November 21, 8:30 AM
Research Presentation 7
PARALLEL ALGORITHMS FOR LEAST SQUARES PROBLEMS

The central question in the parallel matrix computation area is: How should a given matrix computation be apportioned among an array of processors and how best can that array be configured? A useful method that is ideal for exploring this question is the block Jacobi method for computing the singular value decomposition. The speaker will discuss how this algorithm can be used to solve various constrained and unconstrained least squares problems and report on this experience with the technique on a "prototype" of the Cornell supercomputer.

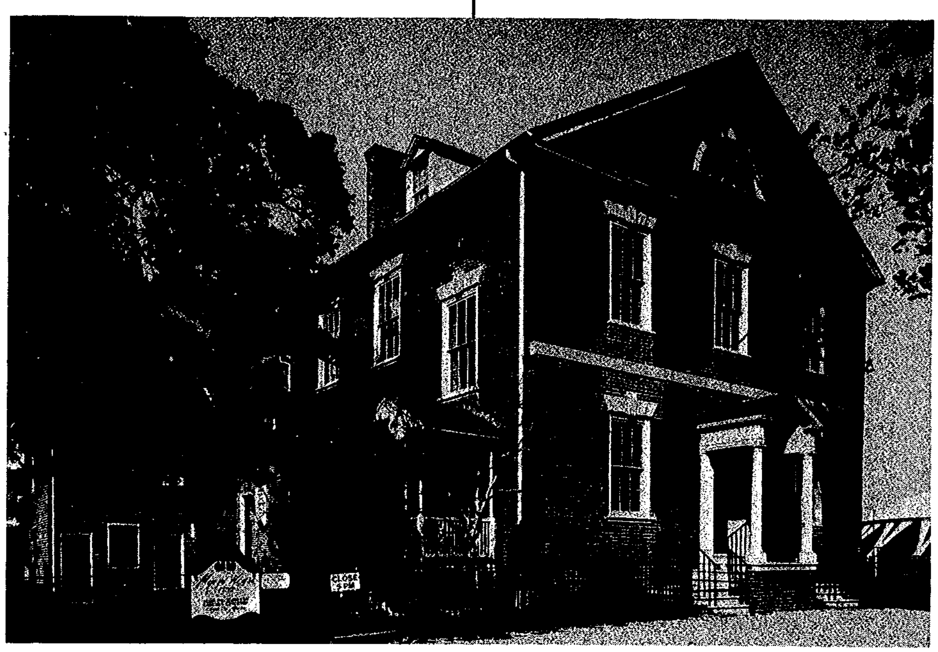
Charles Van Loan
Cornell University
Ithaca, NY

Thursday, November 21, 9:15 AM
Research Presentation 8
ALGORITHMS AND EXPERIMENTS FOR PARALLEL LINEAR SYSTEMS SOLVERS

With the advent of multiprocessors such as the Denelcor HEP and the Cray-XMP, it has become evident that reorganization of many sequential, direct and iterative linear system solvers is essential for efficient utilization of such machines. For several important problems, new algorithms (which may not be efficient on a uniprocessor) yield superior performance on multiprocessors.

The speaker will describe direct and iterative schemes for solving block tridiagonal linear systems on a tightly-coupled cluster of vector processors and discuss the performance of these algorithms on such a cluster. He will also propose algorithms suitable for a machine consisting of several clusters, i.e., algorithms suitable for multiprocessors possessing three levels of parallelism.

Ahmed H. Sameh
University of Illinois
Urbana, IL



Selected Papers

1. **Parallelism with FPS-164 Array Processors and IBM Hosts: Hardwares and Applications to Chemistry And Physics**
   Enrico Clementi, IBM Corporation, Kingston, NY
2. **Parallel Processing of the F_N Method for Radiative Transfer Problems**
   R. C. Y. Chin and G. W. Hedstrom, Lawrence Livermore National Laboratory, Livermore, CA; and C. E. Siewert, North Carolina State University, Raleigh, NC
3. **The Massively Parallel Processor—Architecture, Programming and Applications**
   H. K. Ramapriyan, J. P. Strong and J. C. Tilton, NASA Goddard Space Flight Center, Greenbelt, MD
4. **Interpreting Parallel Processor Performance Measurements**
   Harry F. Jordan, University of Colorado, Boulder, CO
5. **Parallelization and Performance Prediction of the Cooley-Tukey FFT Algorithm for MIMD Shared-Memory Architectures**
   Alan Norton, IBM Thomas J. Watson Research Center, Yorktown Heights, NY and Allan J. Silberger, Cleveland State University, Cleveland, OH
6. **Using a Single-Program-Multiple-Data Computational Model for Parallel Execution of Scientific Applications**
   F. Darema-Rogers, V. A. Norton and G. F. Pfister, IBM T. J. Watson Research Center, Yorktown Heights, NY
7. **Methods for Reducing the Effects of Communication Delays in Multiprocessor Systems**
   Joel H. Saltz and Vijay K. Naik, ICASE, NASA Langley Research Center, Hampton, VA
8. **Chaotic, Asynchronous Relaxations for the Numerical Solution of Differential Equations by Parallel Processors**
   Debasis Mitra, AT&T Bell Laboratories, Murray Hill, NJ
9. **A Fast Algorithm for the Symmetric Eigenvalue Problem**
   D. C. Sorensen and J. J. Dongarra, Argonne National Laboratory, Argonne, IL
10. **A Non-Systolic Matrix Product Algorithm**
    Philip A. Nelson, University of Washington, Seattle, WA
11. **Problem Partitioning Issues in the Parallel Solutions of Large Toeplitz Systems**
    Jean-Marc Delsome and Ilse C. F. Ipsen, Yale University, New Haven, CT
12. **Solving Dense Linear Systems on a Hypercube Automaton**
    Peter R. Cappello, University of California, Santa Barbara, CA

13. **Solving Elliptic Partial Differential Equations on a Hypercube Multiprocessor**
T. F. Chan, Y. Saad and M. H. Schultz, Yale University, New Haven, CT

14. **Algorithms and Architectures: PDE Solution on Parallel Computers**
Oliver A. McBryan and Eric F. Van DeVelde, Courant Institute, New York University, New York, NY

15. **A Comparison of Domain Decomposition Techniques for Elliptic Partial Differential Equations**
David E. Keyes and William D. Gropp, Yale University, New Haven, CT

16. **On Compiling Loop Algorithm onto Systolic Arrays**
Keith R. Allen and Roy P. Pargas, Clemson University, Clemson, SC

17. **Parallel Algorithms for Network Optimization**
Robert R. Meyer, University of Wisconsin-Madison, Madison, WI

18. **Parallel Debugging: A Preliminary Proposal**
James H. Griffin, Computer Research and Applications Group, and Los Alamos National Laboratory; and Harvey J. Wasserman, Computer User Services Group, Los Alamos National Laboratory, Los Alamos, NM

19. **Solving Equations of Motion on a Virtual Tree Machine**
W. W. Armstrong, T. A. Marsland, M. Olafsson, and J. Schaeffer, University of Alberta, Edmonton, Alberta, Canada

20. **Adapting a Navier-Stokes Code to the ICL-DAP**
Chester E. Grosch, Old Dominion University, and British Maritime Technology, Teddington, England

21. **Implementation of a Navier-Stokes Algorithm on a Local Memory Parallel Processor**
Steven E. Krist, Joint Institute for Advancement of Flight Sciences, NASA Langley Research Center, Hampton, VA

22. **Transformation of Sequential Programs for Multiprocessor Execution: The Blaze Environment**
Piyush Mehrotra, Purdue University, West Lafayette, IN; and Dennis Gannon, Indiana University, Bloomington, IN; and John Van Rosendale, University of Utah, Salt Lake City, UT

23. **A Development Environment for Scientific Parallel Programs**
Alexandru Nicolau, Cornell University, Ithaca, NY

24. **Processor Self-Scheduling in Large Multiprocessor Systems**
Peiyi Tang, Pen-Chung Yew, and Chuan-Qi Zhu, Center for Supercomputing Research and Development, University of Illinois at Urbana-Champaign, Urbana, IL

Hardware/Software Exhibits

Exhibits will be open Monday, November 18 through Thursday, November 21 from 10:00 AM to 5:00 PM.

1. **ALLIANT COMPUTER SYSTEMS CORPORATION**
David L. Rome
Director of Marketing
Alliant Computer Systems Corporation
Acton, MA

2. **AMETEK/COMPUTER RESEARCH DIVISION**
E. Floyd Sherman
Director of Marketing
AMETEK/Computer Research Division
Arcadia, CA

3. **ANALOGIC CORPORATION**
Bruce R. Mackie
Director of Sales and Marketing
Computing Systems Group
Analogic Corporation
Wakefield, MA

4. **INTEL SCIENTIFIC COMPUTERS**
Christopher H. Wain
Account Manager
Intel Scientific Computers
Beaverton, OR

5. **NCUBE CORPORATION**
John F. Palmer
Chairman
NCUBE Corporation
Tempe, AZ

6. **SEQUENT COMPUTER SYSTEMS, INC.**
R. L. Gimbel
Manager of Product Marketing
Sequent Computer Systems, Inc.
Portland, OR

7. **MACNEAL SCHWENDLER CORPORATION**
Bill Moffitt and Louis Komzsik
MacNeal Schwendler Corporation
Los Angeles, CA

8. **SPERRY CORPORATION**
David J. Deak
Manager of Supercomputer Marketing
Sperry Corporation
Blue Bell, PA

9. **LORAL INSTRUMENTATION**
Paul J. Friedman
Product Manager, Computer Systems
Loral Instrumentation
San Diego, CA

10. **APPLIED DYNAMICS INTERNATIONAL**
Peter W. Barhydt
Assistant Sales Manager
Applied Dynamics International
Ann Arbor, MI

11. **CONVEX COMPUTER CORPORATION**
Steve Wallach
Convex Computer Corporation
Richardson, TX

12. **ELXSI**
J. Robert Hedges
Director, Market Planning
ELXSI
San Jose, CA

13. **INMOS CORPORATION**
Steve Roe
Microcomputer Marketing Engineer
INMOS Corporation
Colorado Springs, CO

14. **FLOATING POINT SYSTEMS, INC.**
Ed Kushner and Steve Oslon
Floating Point Systems, Inc.
Portland, OR

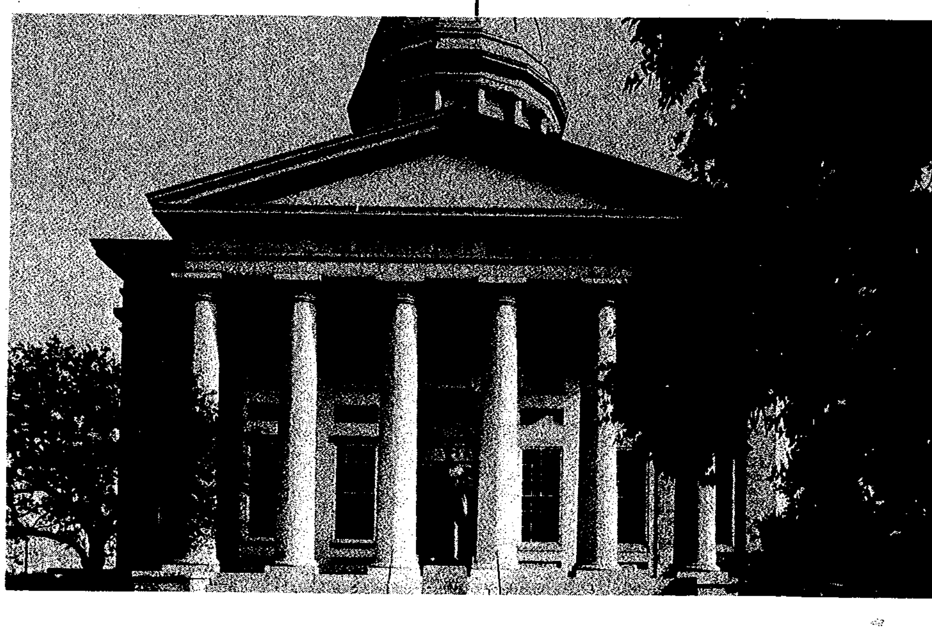
Special Functions

Welcoming Reception
Sunday, November 17, 8:00 PM
Promenade, Third Floor

Wine and Cheese Party
Monday, November 18, 6:00 PM
Promenade, Third Floor

Cash Bar and Poster Session
Tuesday, November 19, 6:00 PM
Promenade, Third Floor

Free Beer and Hardware/Software Exhibits
Wednesday, November 20, 6:00 PM
Promenade and York Hall, Third Floor



PROGRAM-AT-A-GLANCE

Sunday, November 17/PM

5:00 PM
Registration Opens
Promenade, Third Floor

8:00 PM
Welcoming Reception
Promenade, Third Floor

10:00 PM
Registration Closes

Monday, November 18/AM

7:30 AM
Registration Opens
Promenade, Third Floor

8:30 AM/Poplar Hall
Opening Remarks
Robert G. Voigt, Associate Director
ICASE
Hampton, VA

9:00 AM/Poplar Hall
Tutorials 1, 2 and 3
Chairman: Billy Buzbee
Los Alamos National Laboratory
Los Alamos, NM

SUPERCOMPUTERS: PAST, PRESENT AND FUTURE
Jack Worlton
Los Alamos National Laboratory
Los Alamos, NM

10:00 AM/**Coffee**

10:30 AM/Poplar Hall
VECTOR COMPUTER ARCHITECTURES AND THEIR IMPACT ON COMPUTATION
Jay Lambiotte, Jr.
NASA Langley Research Center
Hampton, VA

11:15 AM/Poplar Hall
MULTIPROCESSOR ENVIRONMENTS: MAPPING ALGORITHMS TO ARCHITECTURES
John R. Van Rosendale
University of Utah
Salt Lake City, UT

12:00 Noon/**Lunch**

Monday, November 18/PM

1:30 PM/Poplar Hall
Tutorials 4, 5 and 6
Chairman: Robert G. Voigt
ICASE
Hampton, VA

PARALLEL AND VECTOR ALGORITHMS—AN INTRODUCTION
James M. Ortega
University of Virginia
Charlottesville, VA

2:30 PM/**Coffee**

3:00 PM/Poplar Hall
FACTORIZATION METHODS IN LINEAR ALGEBRA
Jack J. Dongarra
Argonne National Laboratory
Argonne, IL

3:45 PM/Poplar Hall
ITERATIVE METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS
Loyce Adams
University of Washington
Seattle, WA

4:30 PM/Poplar Hall
Special Lecture
THE NSF SUPERCOMPUTER INITIATIVE
John W. D. Connolly
National Science Foundation
Washington, DC

5:00 PM/Poplar Hall
Special Lecture
NCSA: An Interdisciplinary Research Center Built On An Integrated Supercomputing Environment
Larry L. Smarr
University of Illinois
Urbana, IL

6:00 PM/Promenade
Wine and Cheese Party

Tuesday, November 19/AM

8:30 AM/Poplar Hall
Research Presentations 1 and 2
Chairman: Dennis Gannon
Indiana University
Bloomington, IN

THE "ULTRACOMPUTER"—AN EXPERIMENTAL SHARED MEMORY MIMD COMPUTER
Allan Gottlieb
Courant Institute of Mathematical Sciences
New York University
New York, NY

A TESTBED FOR PARALLEL ALGORITHM DEVELOPMENT
Dianne P. O'Leary
University of Maryland
College Park, MD

10:00 AM/**Coffee**

10:30 AM/CONCURRENT SESSIONS

Selected Papers 1A/Poplar Hall
Applications
Chairman: Peter Eltgroth
Lawrence Livermore National Laboratory
Livermore, CA

Selected Papers 1B/Providence Hall
Performance
Chairman: Dianne O'Leary
University of Maryland
College Park, MD

12:00 Noon/**Lunch**

Tuesday, November 19/PM

1:30 PM/Poplar Hall
Research Presentations 3 and 4
Chairman: Garry Rodrigue
Lawrence Livermore National Laboratory
Livermore, CA

COMPUTATIONAL PHYSICS ON PARALLEL MACHINES
Peter G. Eltgroth
Lawrence Livermore National Laboratory
Livermore, CA

PARALLEL ALGORITHMS FOR PARTIAL DIFFERENTIAL EQUATIONS—CHARACTERISTICS AND DEVELOPMENT TECHNIQUES
Paul O. Frederickson
Los Alamos National Laboratory
Los Alamos, NM

3:00 PM/**Coffee**

3:30 PM/CONCURRENT SESSIONS

Contributed Papers 1/Poplar Hall
Applications 1
Chairman: Steve Davis
Naval Surface Weapons Center
Silver Spring, MD

Contributed Papers 2/Claremont Room
Architecture 1
Chairman: Youcef Saad
Yale University
New Haven, CT

Contributed Papers 3/Providence Hall
Linear Algebra 1
Chairman: Ahmed Sameh
University of Illinois
Urbana, IL

Contributed Papers 4/Brandon Room
Software 1
Chairman, Daniel Reed
University of Illinois
Urbana, IL

6:00 PM/Promenade
POSTER SESSION
Cash Bar

Wednesday, November 20/AM

8:30 AM/Poplar Hall
Research Presentations 5 and 6
Chairman: John Van Rosendale
University of Utah
Salt Lake City, UT

ARCHITECTURE AND PROGRAMMING OF THE COSMIC CUBE
Charles L. Seitz
California Institute of Technology
Pasadena, CA

SUPRENUM—THE GERMAN MULTIGRID SUPERCOMPUTER PROJECT
Ulrich Trottenberg
Gesellschaft fur Mathematik und Datenverarbeitung and University of Cologne
Federal Republic of Germany

10:00 AM/**Coffee**

10:30 AM/CONCURRENT SESSIONS

Selected Papers 2A/Poplar Hall
Algorithms
Chairman: Joseph Oliger
Stanford University
Stanford, CA

Selected Papers 2B/Providence Hall
Linear Algebra
Chairman: Michael Heath
Oak Ridge National Laboratory
Oak Ridge, TN

12:00 Noon – 12:30 PM/Providence Hall
Discussion of the Extended Basic Linear Algebra Subprograms
Group Leader: Jack Dongarra
Argonne National Laboratory
Argonne, IL

12:00 Noon/**Lunch**

Wednesday, November 20/PM

1:30 PM/CONCURRENT SESSIONS

Selected Papers 3A/Poplar Hall
Elliptic Equations
Chairman: Loyce Adams
University of Washington
Seattle, WA

Selected Papers 3B/Providence Hall
Software
Chairman: Piyush Mehrotra
Purdue University
West Lafayette, IN

3:00 PM/**Coffee**

3:30 PM/CONCURRENT SESSIONS

Contributed Papers 5/Poplar Hall
Applications 2
Chairman: Marsha Berger
Courant Institute of Mathematical Sciences,
New York University
New York, NY

Contributed Papers 6/Claremont Room
Architecture 2
Chairman: Donald Heller
Shell Development Company
Houston, TX

Contributed Papers 7/Providence Hall
Linear Algebra 2
Chairman: Charles Van Loan
Cornell University
Ithaca, NY

Contributed Papers 8/Brandon Room
Software 2
Chairman: Shahid Bokhari
ICASE, Hampton, VA

6:00 PM – 10:00 PM/York Hall
Hardware/Software Exhibits
Free Beer on the Promenade

Thursday, November 21/AM

8:30 AM/Poplar Hall
Research Presentations 7 and 8
Chairman: Jack Dongarra
Argonne National Laboratory
Argonne, IL

PARALLEL ALGORITHMS FOR LEAST SQUARES PROBLEMS
Charles Van Loan
Cornell University
Ithaca, NY

ALGORITHMS AND EXPERIMENTS FOR PARALLEL LINEAR SYSTEMS SOLVERS
Ahmed H. Sameh
University of Illinois
Urbana, IL

10:00 AM/**Coffee**

10:30 AM/CONCURRENT SESSIONS

Selected Papers 4A/Poplar Hall
Applications
Chairman: Oliver McBryan
Courant Institute of Mathematical Sciences,
New York University
New York, NY

Selected Papers 4B/Providence Hall
Environments
Chairman: Harry Jordan
University of Colorado
Boulder, CO

12:00 Noon/**Lunch**

Thursday, November 21/PM

1:30 PM/CONCURRENT SESSIONS

Contributed Papers 9/Poplar Hall
Applications 3
Chairman: Jay Lambiotte
NASA Langley Research Center
Hampton, VA

Contributed Papers 10/Claremont Room
Architecture 3
Chairman: Chester Grosch
Old Dominion University
Norfolk, VA

Contributed Papers 11/Providence Hall
Linear Algebra 3
Chairman: Danny Sorensen
Argonne National Laboratory
Argonne, IL

Contributed Papers 12/Brandon Room
Software 4
Chairman: Joel Saltz
ICASE, Hampton, VA

4:00 PM/Program Adjourns

Hardware/Software Exhibits

Exhibits will be open Monday, November 18 through Thursday, November 21 from 10:00 AM to 5:00 PM.

ALLIANT COMPUTER SYSTEMS CORPORATION

Alliant will exhibit its recently announced family of parallel processing entry-level-computers. These systems are based on an architecture which interconnects multiple high performance vector processors with each other through dedicated parallel processing control hardware and with a common global memory at over 370 MB per second. Available for hands-on demonstrations will be an optimizing compiler that identifies opportunities for parallel and vector execution in existing programs and an operating system which schedules, transparently to the user, multiple processors as a single resource during single program execution.

David L. Rome
Director of Marketing
Alliant Computer Systems Corporation
Acton, MA

AMETEK/COMPUTER RESEARCH DIVISION

The Ametek/Computer Research Division will exhibit its first generation Concurrent Processing Hardware and Software. Based on a hypercube topology, the Ametek system includes microprocessor based hardware, concurrent operating system and messaging system, programming tools, and applications libraries. Considerable effort has been devoted to developing a programming environment which simplifies the task of concurrent programming and makes it accessible to the scientific programmer.

E. Floyd Sherman
Director of Marketing
AMETEK/Computer Research Division
Arcadia, CA

ANALOGIC CORPORATION

A high performance computer workstation based upon a Hewlett-Packard 9000 Desktop Computer and an Analogic AP500 Array Processor with appropriate storage and display systems. We will be running computationally-intensive complex imaging tasks as well as mathematically intensive signal processing and statistical tasks.

Bruce R. Mackie
Director of Sales and Marketing
Computing Systems Group
Analogic Corporation
Wakefield, MA

INTEL SCIENTIFIC COMPUTERS

The iPSC family of concurrent computing systems puts supercomputer performance in the hands of the individual researcher.

The iPSC is a multiprocessor system that combines a proven parallel processing architecture with Intel's strength in VLSI technology. The concurrent operation of as many as 128 independent processing units ensures high performance, while the use of standard VLSI microcomputers guarantees low cost and high reliability. These processing units, or nodes are connected using a hypercube architecture.

Three iPSC models are available, consisting of 32, 64, or 128 processing nodes.

Christopher H. Wain
Account Manager
Intel Scientific Computers
Beaverton, OR

NCUBE CORPORATION

We will exhibit a high performance parallel computer system based on the hypercube interconnection network. It will be a standalone computer with its own operating system, compilers and libraries. We will have demonstration programs that exhibit the power of parallel processing. The computer has a very high bandwidth Input/Output system which will also be demonstrated.

John F. Palmer
Chairman
NCUBE Corporation
Tempe, AZ

SEQUENT COMPUTER SYSTEMS, INC.

Sequent is demonstrating its Balance (TM) 8000 System, a high performance, general purpose, parallel computer that includes from two to twelve, tightly coupled, 32-bit microprocessors running a single re-entrant copy of DYNIX (TM), an enhanced UNIX (TM) 4.2 bsd operating system. Features supporting parallel programming include fast hardware locks, shared memory, and a library of parallel programming support routines.

R. L. Gimbel
Manager of Product Marketing
Sequent Computer Systems, Inc.
Portland, OR

MACNEAL SCHWENDLER CORPORATION

MSC, a leader in the supercomputer software industry, will demonstrate how MSC/NASTRAN uses vectorization and parallel processing to dramatically increase performance on modern supercomputers. In addition, MSC will show its new matrix equation solving program for the IBM PC, MSC/mate.

Bill Moffitt and Louis Komzsik
MacNeal Schwendler Corporation
Los Angeles, CA

SPERRY CORPORATION

We will be displaying a console mock-up of the Sperry Integrated Scientific Processor (ISP) System. Designed around the highly stable and successful Sperry 1100/90 System, this completely integrated system efficiently executes scientific vector-oriented Fortan code. With this system, you get the outstanding performance of a supercomputer coupled with the program development and support environment of a large mainframe.

David J. Deak
Manager of Supercomputer Marketing
Sperry Corporation
Blue Bell, PA

LORAL INSTRUMENTATION

Loral Instrumentation, a division of Loral Corporation, will present the LORAL DATAFLO™ LDF 100 mini-super computer. The LDF 100 is a parallel processing computer, employing data flow, a fifth generation computer architecture. Loral Instrumentation has delivered hundreds of its first generation data flow systems for high speed data acquisition and processing. The LDF 100, Loral's second generation data flow computer, is a modular large grain data flow system, incrementally configurable from 5 to 256 high performance Node Processors. The LDF 100 is user programmable in standard high level languages and includes a highly integrated I/O system.

Paul J. Friedman
Product Manager, Computer Systems
Loral Instrumentation
San Diego, CA

APPLIED DYNAMICS INTERNATIONAL

Applied Dynamics International is a developer and manufacturer of high-speed real-time hardware and software for the scientific simulation community. ADI will present a display and literature showing state-of-the-art parallel processing for scientific computing.

Peter Barhydt
Applied Dynamics International
Ann Arbor, MI

CONVEX COMPUTER CORPORATION

Convex Computer Corporation will demonstrate the Convex C-1 affordable supercomputer running the Convex UNIX virtual memory operating system (based on 4.2 bsd UNIX) and the Convex optimizing and vectorizing FORTRAN '77 compiler. This general purpose system executes in excess of 60 million operations per second, making it ideal for such computation intensive applications as linear algebra, theoretical and scientific computational research, mathematical modeling, and statistical analysis.

Steve Wallach
Convex Computer Corporation
Richardson, TX

ELXSI

The ELXSI System 6400 multiprocessor will demonstrate parallel performance in production codes and in timesharing/multi-processing environments. Applications include HSPICE and AIDSSIM in the CAD/CAE field and NISA and DYNA-3D in the mechanical/structural design area. ELXSI will show the importance of architectural balance in parallel processing; the ease of parallel processing in FORTRAN, C, and Pascal; and the importance of a symmetric organization and massive memory to efficient algorithm development in production codes.

J. Robert Hedges
Director, Market Planning
ELXSI
San Jose, CA

INMOS CORPORATION

INMOS manufactures high performance VLSI components and is now introducing the Transputer family. The first devices are very fast 32-bit microprocessors with onchip interprocessor communications which allow concurrent systems of any size to be built. Transputers are optimized for HLLs with special support for occam, a descendant of CSP, allowing efficient implementation of parallel systems. We will be demonstrating transputers in a multiprocessor system alongside our occam-based Transputer Development System.

Steve Roe
Microcomputer Marketing Engineer
INMOS Corporation
Colorado Springs, CO

FLOATING POINT SYSTEMS, INC.

FPS will illustrate how its 64 Series of Scientific Computers, with a true 64-bit parallel architecture and large directly-addressable memory, provides supercomputing performance for large-scale scientific and engineering applications.

Using parallel processing techniques, including multiple functional units, parallel instruction words, and concurrent hardware accelerators, the FPS 64 Series offers performance up to 341 MFLOPS. The FPS optimizing Fortran compiler and extensive libraries facilitate efficient use of parallel features.

Another level of parallelism, multiprocessor configurations, offers an incremental approach to even greater levels of performance, flexibility, and large problem capacity.

Ed Kushner and Steve Oslon
Floating Point Systems, Inc.
Portland, OR

LIST OF SPEAKERS*

A

Adams, L., Mon. PM
Agarwal, R. C., Thu. PM
Agrawal, D. P., Tue. PM
Aharonian, G., Thu. PM (2)
Alexandridis, N. A., Tue. PM
Allen, K., Wed. PM
Allen, F. D., Thu. PM
Andriessen, J., Tue. PM
Andriessen, J., Thu. PM
Armstrong, W. W., Thu. AM
Axelsson, O., Thu. PM

B

Baden, S. B., Thu. PM
Bassett, M. E., Wed. PM
Benner, R. E., Thu. PM (2)
Berkovich, S. Y., Wed. PM
Bhavsar, V. C., Wed. PM
Bhavsar, V. C., Thu. PM
Bostic, S. W., Tue. PM
Briggs, W. L., Thu. PM
Brooks, E. D., Tue. PM
Bykat, A., Wed. PM

C

Calahan, D. A., Tue. PM
Calahan, D. A., Wed. PM
Cappello, P. R., Wed. AM
Carlson, D. A., Tue. PM
Celis, M. R., Tue. PM
Chiang, Y. F., Thu. PM
Chen, M. C., Wed. PM
Chen, Y. M., Tue. PM
Chiarulli, D. M., Wed. PM
Chin, R. C. Y., Tue. AM
Clark, P., Wed. PM
Clementi, E., Tue. AM
Connolly, J., Mon. PM
Conroy, J., Tue. PM
Cosnard, M., Wed. PM
Craig, J. N., Wed. PM
Cybenko, G., Wed. PM

D

Darema-Rogers, F., Tue. AM
Datta, K., Thu. PM
de Doncker, E., Thu. PM
Dietz, H., Wed. & Thu. PM
Dongarra, J. J., Mon. PM
Dorband, J. E., Thu. PM
Duff, I. S., Wed. PM

E

Eddy, W. F., Thu. PM
Edwards, O. D., Tue. PM
Eijkhout, V., Tue. PM
Eltgroth, P. G., Tue. PM
Evans, D. J., Wed. PM

F

Fielland, G. N., Tue. PM
Fisher, D. C., Wed. PM
Frederickson, P. O., Tue. PM
Fulton, R. E., Tue. PM

G

Gary, J., Thu. PM
Gerdts, A., Wed. PM
Ghafoor, A., Tue. PM
Ginsberg, M., Tue. PM
Gomez, G., Tue. PM
Gottlieb, A., Tue. AM
Gropp, W. D., Wed. PM
Grosch, C. E., Thu. AM
Gustavson, F. G., Wed. PM
Gustavson, F. G., Thu. PM

H

Hamilton, H., Thu. PM
Horvath, J. C., Wed. PM
Hu, Y-H., Tue. PM

I

Ipsen, I.C.F., Wed. AM & PM

J

Jessup, E., Thu. PM
Johnsson, L., Tue. PM
Jones, R., Tue. PM
Jordan, H. F., Tue. AM
Josin, G. M., Tue. PM

K

Kalogerakis, M. A., Wed. PM
Kamgnia, E. R., Tue. PM
Kapenga, J. A., Wed. PM
Kaplan, I., Thu. PM (2)
Karakashian, O., Wed. PM
Karp, A. H., Tue. PM
Keyes, D. E., Wed. PM
Kirtane, J., Tue. PM
Khalaf, S., Tue. PM
Klappholz, D., Tue. PM
Konno, C., Tue. PM
Koniges, A. E., Thu. PM
Krist, S. E., Thu. AM
Kumar, V. K. P., Tue. PM
Kumar, S. P., Tue. PM
Kushner, E., Tue. PM
Kwasowiec, W., Tue. PM

L

Lambiotte, J., Mon. AM
Lambrakos, S. G., Wed. PM
Leuze, M. R., Thu. PM
Lucier, B., Wed. PM
Luk, F. T., Tue & Wed.. PM

M

MacGregor, D. M., Tue. PM
McAulay, A. D., Thu. PM
McBryan, O. A., Wed. PM
McDowell, C. E., Tue. PM
McGrath, J. F., Thu. PM
Mehrotra, P., Thu. AM
Melhem, R. G., Wed. PM
Meyer, R. R., Wed. PM
Minkoff, M., Tue. PM
Mitra, D., Wed. AM
Modi, J. J., Tue. PM
Moler, C., Wed. PM
Munshi, A. A., Thu. PM

N

Nelson, P. A., Wed. AM
Nicolau, A., Tue. PM
Nicolau, A., Thu. AM
Norton, A., Tue. AM
Nour-Omid, B., Thu. PM

O

O'Leary, D. P., Tue. AM
Ortega, J. M., Mon. PM

P

Palecek, L., Thu. PM
Palmer, J. F., Tue. PM
Papadopoulou, E., Thu. PM
Parks, T., Tue. PM
Perez, J-C., Wed. PM
Philippe, B., Thu. PM
Pierce, D., Tue. PM
Place, J. P., Wed. PM
Plemmons, R. J., Tue. PM
Prakash, A., Thu. PM
Proulx, V. K., Tue. PM
Purcell, C. J., Wed. PM

R

Ramapriyan, H. K., Tue. AM
Ransom, J. B., Wed. PM
Reed, D. A., Wed. PM
Resasco, D. C., Thu. PM
Rice, J. R., Wed. PM

S

Saad, Y., Wed. PM
Saltz, J. H., Wed. AM
Sameh, A. H., Thu. AM
Saridakis, Y. G. Tue. PM
Schnabel, R., Tue. PM
Seitz, C. L., Wed. AM
Shen, J. P., Tue. PM
Simon, H. D., Thu. PM (2)
Sorensen, D. C., Wed. AM

T

Tang, P., Thu. AM
Tang, W. P., Tue. PM
Thacker, W. I., Thu. PM
Tolsma, L. D., Tue. PM
Treadway, A. H., Thu. PM
Trottenberg, U., Wed. AM

V

Vandevender, W. H., Thu. PM
van de Geijn, R. A., Thu. PM
Van Loan, C., Thu. AM
Van Rosendale, J. R., Mon. AM
Varman, P. J., Wed. PM
Vogel, C., Tue. PM

W

Wasserman, H. J., Wed. PM
Weigand, G. G., Thu. PM
White, R. E., Tue. PM
Wilson, D. G., Thu. PM
Wolf, G., Wed. PM
Woods, D. J., Tue. PM
Worlton, J., Mon. AM
Wunderlich, M., Thu. PM

Y

Young, D. P., Tue, PM

ADDITIONAL SPEAKERS

Park, H., Wed. PM
Smarr, L., Mon. PM
Wilson, A., Tue. PM
Wilson, P., Wed. PM

*Speakers' names are followed by the day, and whether morning or afternoon, when they are scheduled to present their papers. The number in parentheses indicates how many times a speaker will make a presentation on the same day.

TIMETABLE

Sunday, November 17/PM

5:00 PM
Registration Opens
Promenade, Third Floor

8:00 PM
Welcoming Reception
Promenade, Third Floor

10:00 PM
Registration Closes

Monday, November 18/AM

7:30 AM
Registration Opens
Promenade, Third Floor

8:30 AM/Poplar Hall
Opening Remarks
Robert G. Voigt, Associate Director
ICASE
Hampton, VA

9:00 AM/Poplar Hall
Tutorials 1, 2 and 3
Chairman: Billy Buzbee
Los Alamos National Laboratory
Los Alamos, NM

SUPERCOMPUTERS: PAST, PRESENT AND FUTURE
Jack Worlton
Los Alamos National Laboratory
Los Alamos, NM

10:00 AM/**Coffee**

10:30 AM/Poplar Hall
VECTOR COMPUTER ARCHITECTURES AND THEIR IMPACT ON COMPUTATION
Jay Lambiotte, Jr.
NASA Langley Research Center
Hampton, VA

11:15 AM/Poplar Hall
MULTIPROCESSOR ENVIRONMENTS: MAPPING ALGORITHMS TO ARCHITECTURES
John R. Van Rosendale
University of Utah
Salt Lake City, UT

12:00 Noon/**Lunch**

Monday, November 18/PM

1:30 PM/Poplar Hall
Tutorials 4, 5 and 6
Chairman: Robert G. Voigt
ICASE
Hampton, VA

PARALLEL AND VECTOR ALGORITHMS—AN INTRODUCTION
James M. Ortega
University of Virginia
Charlottesville, VA

2:30 PM/**Coffee**

3:00 PM/Poplar Hall
FACTORIZATION METHODS IN LINEAR ALGEBRA
Jack J. Dongarra
Argonne National Laboratory
Argonne, IL

3:45 PM/Poplar Hall
ITERATIVE METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS
Loyce Adams
University of Washington
Seattle, WA

4:30 PM/Poplar Hall
Special Lecture
THE NSF SUPERCOMPUTER INITIATIVE
John W. D. Connolly
National Science Foundation
Washington, DC

5:00 PM/Poplar Hall
Special Lecture
NCSA: An Interdisciplinary Research Center Built On An Integrated Supercomputing Environment
Larry L. Smarr
University of Illinois
Urbana, IL

6:00 PM/Promenade
Wine and Cheese Party

Tuesday, November 19/AM

8:30 AM/Poplar Hall
Research Presentations 1 and 2
Chairman: Dennis Gannon
Indiana University
Bloomington, IN

THE "ULTRACOMPUTER"—AN EXPERIMENTAL SHARED MEMORY MIMD COMPUTER
Allan Gottlieb
Courant Institute of Mathematical Sciences
New York University
New York, NY

A TESTBED FOR PARALLEL ALGORITHM DEVELOPMENT
Dianne P. O'Leary
University of Maryland
College Park, MD

10:00 AM/**Coffee**

10:30 AM/CONCURRENT SESSIONS

Tuesday, November 19/*10:30 AM–12:00 Noon*
Selected Papers 1A/Poplar Hall
APPLICATIONS

Chairman: Peter Eltgroth, Lawrence Livermore National Laboratory, Livermore, CA

10:30
Parallelism with *FPS-164* Array Processors and *IBM* Hosts: Hardwares and Applications to Chemistry and Physics
We take up the problem of realistic computer simulation of complex systems. We discuss chemical complexity and the concomitant progression through a set of steps based on submodels. We present a few examples of this approach applied to biophysical and chemical problems of importance in biology.

Solution of such problems motivates our experiments in parallel systems. Our hardware consists of two IBM-4341 and one IBM-4381 host computers and ten FPS-164 attached array processors. Hardware enhancements to this system are discussed. Our successes with this system suggest it represents a pragmatic, economic and efficient answer to many computer intensive applications.

Enrico Clementi
IBM Corporaton
Kingston, NY

11:00
Parallel Processing of the F_N Method for Radiative Transfer Problems
The equation of radiative transfer is an integrodifferential equation which involves first-order partial derivatives with respect to time and the spatial variables and an integral over the momentum variables. For such large problems we need methods which incorporate any analytic information we might have about the solution. Analysis may also be used to identify nearly independent processes which may be computed in parallel.

We have implemented the F_N method on a serial machine and are transferring it to a 4-processor VAX.

R. C. Y. Chin and G. W. Hedstrom, Lawrence Livermore National Laboratory, Livermore, CA
C. E. Siewert, North Carolina State University, Raleigh, NC

11:30
The Massively Parallel Processor–Architecture, Programming and Applications
The Massively Parallel Processor (MPP) is a unique computer capable of orders of magnitude higher computational speeds than conventional computers.

The MPP was conceptualized at the Goddard Space Flight Center (GSFC) in 1977 and developed, under contract, by the Goodyear Aerospace Corporation (GAC). It was delivered to GSFC in May 1983. Since then, GSFC has been active in integrating it with a VAX 11/780 host computer and developing systems and applications software. This paper describes the architecture of the MPP, the facilities available for developing applications software on it and the use of its high-speed capabilities for NASA's applications.

H. K. Ramapriyan, J. P. Strong and J. C. Tilton
Space Data and Computing Division
NASA Goddard Space Flight Center
Greenbelt, MD

Tuesday, November 19/*10:30 AM–12:00 Noon*
Selected Papers 1B/Providence Hall
PERFORMANCE

Chairman: Dianne O'Leary, University of Maryland, College Park, MD

10:30
Interpreting Parallel Processor Performance Measurements
This work demonstrates some models which give good insight into the behavior of measured execution times for parallel algorithms on parallel processors as the degrees of physical, program and algorithmic parallelism are varied. The models can be used to predict performance and optimize program design choices. They can also be used after the fact to obtain execution profile type information about an existing parallel program. The predicted features of execution time are very evident in measurements made on pipelined parallel processors which have low hardware scheduling overhead.
Harry F. Jordan
University of Colorado
Boulder, CO

11:00
Parallelization and Performance Prediction of the Cooley-Tukey FFT Algorithm for MIMD Shared-memory Architectures
We study the parallelization of the Cooley-Tukey radix two FFT algorithm for MIMD (nonvector) architectures. Parallel algorithms are presented for one- and multidimensional Fourier transforms, and coded in FORTRAN. Precise instruction counts were measured from traces in order to determine exactly the instructions and synchronizations to be executed by each processor in the system, as a function of the problem size. We thereby can predict execution time on various shared-memory architectures, and we present such estimates for the RP3 parallel prototype. These techniques enable one to estimate the value of algorithmic optimizations, and to better understand the hardware improvements suitable for FFT computation.
Alan Norton
IBM Thomas J. Watson Research Center
Yorktown Heights, NY
Allan J. Silberger
Cleveland State University
Cleveland, OH

11:30
Using A Single-Program-Multiple-Data Computational Model for Parallel Execution of Scientific Applications
We present our experience with our Single-Program-Multiple-Data computational model which we have used to parallelize a number of scientific applications. The computational model assumes a shared memory organization and is based on the scheme that all processes executing a program in parallel remain in existence for the entire execution; however, the tasks to be executed by each process are determined dynamically during execution by the use of appropriate synchronizing constructs that are imbedded in the code. We will discuss parallelization features of applications and performance issues such as overhead, speedup, efficiency.
F. Darema-Rogers, V. A. Norton, and G. F. Pfister
Computer Science Department
IBM T. J. Watson Research Center
Yorktown Heights, NY

12:00 Noon/Lunch

Tuesday, November 19/PM

1:30 PM/Poplar Hall
Research Presentations 3 and 4
Chairman: Garry Rodrigue
Lawrence Livermore National Laboratory
Livermore, CA

COMPUTATIONAL PHYSICS ON PARALLEL MACHINES
Peter G. Eltgroth
Lawrence Livermore National Laboratory
Livermore, CA

PARALLEL ALGORITHMS FOR PARTIAL DIFFERENTIAL EQUATIONS—CHARACTERISTICS AND DEVELOPMENT TECHNIQUES
Paul O. Frederickson
Los Alamos National Laboratory
Los Alamos, NM

3:00 PM/**Coffee**

3:30 PM/CONCURRENT SESSIONS

Tuesday, November 19/*3:30–6:00 PM*
Contributed Papers 1/Poplar Hall
APPLICATIONS 1

Chairman: Steve Davis, Naval Surface Weapons Center, Silver Spring, MD

3:30/112/A25
Parallel Processing Opportunities for Structural Finite Element Methods
Robert E. Fulton, George Washington University and Institute for Computer Applications to Science and Engineering, NASA Langley Research Center, Hampton, VA

3:45/12/A3
Some Parallel Algorithms for Matrix Structural Analysis
Robert J. Plemmons, Departments of Computer Science and Mathematics, North Carolina State University, Raleigh, NC

4:00/60/A14
Finite Element Computation on the Distributed Array Processor (DAP) and on Transputer Machine
J. J. Modi, Department of Engineering, Cambridge University, Cambridge, U.K.

4:15/85/A19
Architectural Considerations in Program Design: A Comparison of Two MIMD Computers
Susan W. Bostic and Jonathan B. Ransom, Structures and Dynamics Division, NASA Langley Research Center, Hampton, VA; Thomas W. Crockett, PRC Systems Services, Aerospace Technologies Division, Hampton, VA

4:30/113/A25
Concurrent Numerical Optimization in a Distributed Computing Environment
Robert B. Schnabel, Department of Computer Science, University of Colorado, Boulder, CO

4:45/3/A1
Schwarz Splitting and Parallel Computation
Wei Pai Tang, Department of Computer Science, Stanford University, Stanford, CA

5:00/2/A1
High Level Parallelism in Hierarchy of GPST Inversion Algorithm
Y. M. Chen, Department of Applied Mathematics and Statistics, State University of New York, Stony Brook, NY

5:15/37/A9
Numerical Solution of an Inverse Scattering Problem on a Supercomputer
Curt Vogel, Applied Mathematical Sciences, Iowa State University, Ames, IA; and Gerhard Kristensson, Division of Electromagnetic Theory, Royal Institute of Technology, Stockholm, Sweden

5:30/124/A28
Error, Convergence and Stability Analysis of Multi-rate Methods for Real Time Simulation
Diego Bricio Hernández, Depto. de Ingenieria de Procesos e Hidráulica, Universidad Autónoma Metropolitana Unidad Ixtapalapa, México, D. F., Mexico; and Francisco J. Carrión and Gustavo Rodriguez Gómez, Departamento de Simulación, Instituto de Investigaciones Eléctricas, Mexico

5:45/46/A11
Parallel Finite-Difference Migration
Stewart A. Levin, Department of Geophysics, Stanford University, Stanford, CA; and Terry Parks, Algorithms Department, Guiltech Research Company, Sunnyvale, CA

Tuesday, November 19/*3:30–5:30 PM*
Contributed Papers 2/Claremont Room
ARCHITECTURE 1

Chairman: Youcef Saad, Yale University, New Haven, CT

3:30/95/A22
Delft Parallel Processor 1984
J. H. M. Andriessen, Department of Mathematics and Informatics, Delft University of Technology, Julianalaan, The Netherlands

3:45/114/A26
ROPE: A Statically-scheduled Supercomputer Architecture
Kevin Karplus and Alexandru Nicolau, Department of Computer Science, Cornell University, Ithaca, NY

4:00/83/A19
Parallel Numerical Computations on a VLSI-Based Multiprocessor
V. K. Prasanna Kumar, P.S. Tseng, and K. Hwang, Computer Research Institute, University of Southern California, Los Angeles, CA

4:15/84/A19
A Mesh-Connected Computer Architecture for Solving Elliptic Pde's
Jayant Kirtane, Department of Computer Science, Pennsylvania State University, University Park, PA

4:30/66/A15
Parallel Architectures for a Family of Iterative Schemes
Yiannis G. Saridakis, Department of Mathematics and Computer Science, Clarkson University, Potsdam, NY

4:45/24/A6
Hypercubes: Architecture and Algorithms
John F. Palmer, NCUBE Corporation, Tempe, AZ

Note: The numbers that follow the individual times for each paper represent the numerical order and page number of the abstract. For papers

5:00/25/A6
The Shared Memory Hypercube
Eugene D. Brooks III, Lawrence Livermore National Laboratory, Livermore, CA

5:15/122/A28
B-HIVE Project: Present and Future
Dharma P. Agrawal, Winser E. Alexander, Edward F. Gehringer, Ravi Mehrotra, Department of Electrical and Computer Engineering; and Jon Mauney, Department of Computer Science; North Carolina State University, Raleigh, NC

Tuesday, November 19/*3:30–5:30 PM*
Contributed Papers 3/Providence Hall
LINEAR ALGEBRA 1

Chairman: Ahmed Sameh, University of Illinois, Urbana, IL

3:30/29/A7
Solving Tridiagonal Linear System of Equations on the CYBER 205 Computer
Swarn P. Kumar, Department of Computer Science, Colorado State University, Fort Collins, CO; and Janusz S. Kowalik, Boeing Computer Services Company, Advanced Technology Applications Division, Seattle, WA

3:45/69/A16
Data Permutations and Basic Linear Algebra Computations on Ensemble Architectures
Lennart Johnsson, Department of Computer Science, Yale University, New Haven, CT

4:00/71/A16
Vectorization of Scalar Recurrences
O. Axelsson and V. Eijkhout, Department of Mathematics, Catholic University, Toernooiveld, Nijmegen, The Netherlands

4:15/99/A23
CORDIC Implementation of Rotation-Based Matrix Algorithms
Yu-Hen Hu, Department of Electrical Engineering, Southern Methodist University, Dallas, TX

4:30/54/A12
Block Angular Parallel Least Squares Computations
D. Pierce, Department of Mathematics; and R. J. Plemmons, Departments of Computer Science and Mathematics, North Carolina State University, Raleigh, NC; and A. H. Sameh, Department of Computer Science and The Center for Advanced Computation, University of Illinois, Urbana, IL

4:45/9/A2
The Parallel Solution of a Recursive Least Squares Problem in Signal Processing
Franklin T. Luk, School of Electrical Engineering; and Sanzheng Qiao, Center for Applied Mathematics, Cornell University, Ithaca, NY

5:00/41/A9
Adaptable Supercomputers in Image Processing
N. A. Alexandridis, P. D. Tsanakas, S. Ziavras, Department of Electrical and Computer Engineering, Ohio University, Athens, OH

5:15/123/A28
Performance Comparisons of Numerical Kernels Executed on Supercomputers, Mainframes, Minis, and Micros
Myron Ginsberg, Department of Computer Science, General Motors Research Laboratories, Warren, MI

Tuesday, November 19/*3:30–5:45 PM*
Contributed Papers 4/Brandon Room
SOFTWARE 1

Chairman: Daniel Reed, University of Illinois, Urbana, IL

3:30/63/A15
Folding Microcode for Processing Arrays of Data
Ralph Jones, Analogic Corporation, Computing Systems Group, Wakefield, MA and College of Engineering, Boston University, Boston, MA

3:45/30/A7
Parallel Programming on a General Purpose Multiprocessor
Gary N. Fielland, Sequent Computer Systems, Inc., Portland, OR

4:00/93/A21
An Expert Tool for Programming Multiprocessors
Henry Dietz, Department of Electrical Engineering and Computer Science, Polytechnic Institute of New York, Farmingdale, NY; and David Klappholz, Center for Distributed Processing, Department of Computer Science, Stevens Institute of Technology, Hoboken, NJ

4:15/96/A22
Programming Language Equivalent to Petri Nets
Wlodzimierz Kwasowiec, Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

4:30/115/A26
Interconnection Networks for Multiprocessor System
Arif Ghafoor, Department of Electrical and Computer Engineering, Syracuse University, Syracuse, NY

4:45/7/A2
Interconnection Topology for a Completely Distributed MIMD Computer
Viera K. Proulx, The MITRE Corporation, Signal Processing and Electronic Warfare Experimentation and Data Analysis Division, Bedford, MA

5:00/116/A26
Automatic Compilation of Interprocessor Communication for Multiple Processor Systems
Ronald P. Bianchini, Jr. and John Paul Shen, Department of Electrical and Computer Engineering, Carnegie-Mellon University, Pittsburgh, PA

5:15/117/A26
Solving Linear Recurrence Systems on Mesh-Connected Computers with Multiple Global Buses
David A. Carlson, Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA

5:30/125/A28
Evaluation of DEQSOL (Differential EQuation SOlver Language) Through its Application to Practical Problems
Chisato Konno, Michiru Tsuji, and Yukio Umetani, Central Research Laboratory, Hitachi Ltd., Kokubunji, Tokyo, Japan; and Hiroyuki Hirayama, Hitachi VLSI Engineering Ltd., Kodaira, Tokyo, Japan

---

Tuesday, November 19/*6:00 PM*
POSTER SESSION /Promenade

Cash Bar

22/A5
Efficient Parallel Vectorized Algorithms for Successive Over Relaxation
Oliver D. Edwards, Department of Mathematics, Carnegie-Mellon University, Pittsburgh, PA

28/A6
Fuzzy Clustering on a Few Processors
J. V. Dave and Alan H. Karp, IBM Scientific Center, Palo Alto, CA

32/A7
Parallel One-Way Dissection for the Solution of Sparse Positive Definite Systems
John M. Conroy, Applied Mathematics Program, University of Maryland, College Park, MD

36/A8
A Parallel Architecture for Pattern Recognition
Saleh Khalaf, Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI

47/A11
A Failsafe Associate Memory Circuit
Gary M. Josin, International Computhought Unlimited, Vancouver, British Columbia, Canada

62/A14
Performance Evaluation Methods for Parallel Computing
Roger L. Crane, RCA Laboratories, Princeton, NJ; and Michael Minkoff and Kenneth E. Hillstrom, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL

78/A18
Anomaly Reporting—A Tool for Debugging and Developing Parallel Numerical Algorithms
William F. Appelbe, Department of Electrical Engineering and Computer Science, University of California, San Diego, La Jolla, CA; and Charles E. McDowell, Department of Computer and Information Sciences, University of California, Santa Cruz, CA

80/A18
A Comparison of Small-grain Sparse Equation Solving on the Cray X-MP-4 and the CRAY-2
D. A. Calahan, Department and Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI

90/A21
Parallel Algorithms for Variational Inequalities
Robert E. White, Department of Mathematics, North Carolina State University, Raleigh, NC

Note: The numbers that follow the individual times for each paper represent the numerical order and page number of the abstract. For papers with more than one author, an underline is used to denote the author who will present the paper.

92/A21
Solving Large Sets of Coupled Equations Iteratively by Vector Processing
L. D. Tolsma, Department of Physics, Eindhoven University of Technology, Eindhoven, The Netherlands

101/A23
A Numerical Confotmal Mapping Method for Simply-Connected Domains
Emmanuel R. Kamgnia, Center for Supercomputing Research and Development, Urbana, IL

105/A24
Using the FPS-164/MAX to Solve Linear Equations: A Comparison of Dot Product and SAXPY Formulations
Ed Kushner and Steve Oslon, Floating Point Systems, Inc., Portland, OR

130/A30
Optimal Polynomial Acceleration of Fluid Dynamics Codes Using Krylov Subspace Methods
David Paul Young, Boeing Computer Services, Tukwila, WA; Lawrence B. Wigton and Neng J. Yu, Boeing Commercial Airplane Co., Seattle, WA

132/A30
On the Convergence of the Nelder-Mead Simplex Algorithm
J. E. Dennis, Jr. and Daniel J. Woods, Department of Mathematical Sciences, Rice University, Houston, TX

133/A30
A Trust Region Strategy for Nonlinear Equality Constrained Optimization
Maria Rosa Celis, John E. Dennis, Jr., and Richard A. Tapia, Department of Mathematical Sciences, Rice University, Houston, TX

Wednesday, November 20/AM

8:30 AM/Poplar Hall
Research Presentations 5 and 6
Chairman: John Van Rosendale
University of Utah
Salt Lake City, UT

ARCHITECTURE AND PROGRAMMING OF THE COSMIC CUBE
Charles L. Seitz
California Institute of Technology
Pasadena, CA

SUPRENUM—THE GERMAN MULTIGRID SUPERCOMPUTER PROJECT
Ulrich Trottenberg
Gesellschaft fur Mathematik und Datenverarbeitung and University of Cologne
Federal Republic of Germany

10:00 AM/Coffee

10:30 AM/CONCURRENT SESSIONS

Wednesday, November 20/*10:30 AM–12:00 Noon*
Selected Papers 2A/Poplar Hall
ALGORITHMS

Chairman: Joseph Oliger, Stanford University, Stanford, CA

10:30
Methods for Reducing the Effects of Communication Delays in Multiprocessor Systems
Methods are proposed for efficient computation of numerical algorithms on a wide variety of MIMD machines. These techniques reorganize the data dependency patterns so that the processor utilization is improved.

The model problem examined finds the time-accurate solution to a parabolic partial differential equation discretized in space and implicity marched forward in time. The algorithms investigated are extensions of block Jacobi and SOR.

The methods suggested here increase the degree to which work can be performed while data is communicated between processors. Two methods explored here are: 1) Performing iterations over a window of several timesteps, which allow efficient overlap of computation with communication; 2) Treating subsets of mesh points in each processor as independently schedulable subtasks. The effect of the window size, of subtask scheduling, and of domain partitioning on the system performance is examined both analytically and experimentally by implementing the algorithm on a Flex/32 multiprocessor computer system.

Joel H. Saltz and Vijay K. Naik
ICASE
NASA Langley Research Center
Hampton, VA

11:00
Chaotic, Asynchronous Relaxations for the Numerical Solution of Differential Equations by Parallel Processors
We consider chaotic, asynchronous relaxations for the numerical solution by parallel processors of initial value problems involving a class of ordinary differential equations. The differential equations may be non-linear, non-autonomous and non-homogeneous. The main attraction of the algorithm is that it is essentially asynchronous, requiring negligible coordination between tasks, and the idle times of processors are negligible. The form of the given equations is as a partitioned system which closely corresponds to the composition of physical systems from subsystems. The algorithm iterates asynchronously on functions. We prove that the computed functions uniformly converge at a geometric rate to the unique solution of the given equations. The assumptions on the differential equations are dominance conditions. For linear autonomous equations the assumption is that a certain matrix is a M-matrix.

Debasis Mitra
Mathematical Sciences Research Center
AT&T Bell Laboratories
Murray Hill, NJ

11:30
A Fast Algorithm for the Symmetric Eigenvalue Problem
The symmetric eigenvalue problem is one of the most fundamental problems of computational mathematics. It arises in many applications, and therefore represents an important area for algorithmic research. While it is unlikely that dramatic improvements in execution times for this problem will be obtained with new serial algorithms, it is reasonable to expect that eigenvalue calculations might be accelerated through the use of parallel algorithms. Such an algorithm will be presented in this talk. The algorithm is able to exploit parallelism at all levels of the computation and is well suited to a variety of architectures. However, a pleasant and unexpected bonus of this method is that the parallel algorithm, even when run in serial mode, is significantly faster than the best sequential algorithm on large problems, and is effective on moderate size (order > 30) problems when run in serial mode. Computational experience on a variety of computer architectures will be presented.

D. C. Sorensen and J. J. Dongarra
Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL

Wednesday, November 20/*10:30 AM–12:00 Noon*
Selected Papers 2B/Providence Hall
LINEAR ALGEBRA

Chairman: Michael Heath, Oak Ridge National Laboratory, Oak Ridge, TN

10:30
A Non-systolic Matrix Product Algorithm
Most parallel matrix-matrix product algorithms are systolic. To use these algorithms on a general purpose machine such as the CHiP or cube, the processors will already contain the matrices, so the data must be circulated as if it was being fed in from an external source. We present a non-systolic matrix product algorithm in which the data movement is not the circulation pattern of the adapted systolic algorithms. A technique similar to Strassen's algorithm is used. The running time is O(n) using n^2 processors. We compare this algorithm to a systolic algorithm and give experimental results.

Philip A. Nelson
University of Washington
Seattle, Washington

11:00
Problem Partitioning Issues in the Parallel Solution of Large Toeplitz Systems
Algorithms for the parallel solution of large systems of linear equations with symmetric positive definite Toeplitz coefficient matrices are considered when the number of available processors is smaller than the problem size. Large Toeplitz systems occur upon discretization of integral equations in the study of stationary time series, for boundary value problems in stress analysis and inverse problems in seismic exploration.

Combinations of the following algorithms for the solution of dense Toeplitz systems are considered: generalized Levinson, doubling algorithm and Toeplitz Hyperbolic Cholesky Solver, a highly pipelineable algorithm based on the Schur algorithm. The problem is partitioned into three levels, and given the number of processors and communication costs, the most efficient combination of algorithms is determined as well as the optimal size of the problems to be solved in each processor. We consider the implementation of these algorithms on a linear processor array and on the Hypercube. Numerical examples will be presented and the treatment of non-definite problems will be discussed.

Jean-Marc Delosme
Dept. of Electrical Engineering, and
Ilse C. F. Ipsen
Dept. of Computer Science
Yale University
New Haven, CT

Note: The numbers that follow the individual times for each paper represent the numerical order and page number of the abstract. For papers with more than one author, an underline is used to denote the author who will present the paper.

11:30
Solving Dense Linear Systems on a Hypercube Automaton
An automaton is given for solving a dense linear system, $Ax = b$. With respect to communication, the automaton's cells are organized as a hypercube. The rate of communication is O(1), modeling existing hypercube computers.

The automaton performs Gaussian elimination with pivoting followed by back substitution.

Large linear systems can be solved on a small hypercube. A square linear system of n equations may be solved efficiently on a hypercube of $H \leq n^2/\log n$ cells. Such an automaton uses $O(n^2/H)$ memory/cell and solves the system in time $O(n^3/H + n\log H)$. For this range of H, the automaton's cell-time product and total memory are equal to those needed to perform Gaussian elimination with pivoting followed by back substitution on a single processor. It thus can be configured in a range of efficient processor-time tradeoffs. For $H = n^2/\log n$, the automaton's time is $O(n \log n)$.

Peter R. Cappello
University of California
Santa Barbara, CA

*12:00 Noon–12:30 PM/*Providence Hall
Discussion of the Extended Basic Linear Algebra Subprograms
Group Leader: Jack Dongarra
Argonne National Laboratory
Argonne, IL

*12:00 Noon/*Lunch

Wednesday, November 20/PM

1:30 PM/CONCURRENT SESSIONS

Wednesday, November 20/*1:30–3:00 PM*
Selected Papers 3A/Poplar Hall
ELLIPTIC EQUATIONS

Chairman: Loyce Adams, University of Washington, Seattle, WA

1:30
Solving Elliptic Partial Differential Equations on a Hypercube Multiprocessor
We discuss the implementation of several classical methods for solving elliptic partial differential equations on a loosely coupled parallel processor based on the hypercube topology. The methods considered are a banded Gaussian elimination algorithm, the Alternating Direction method, and multigrid algorithms. A simple model is used for estimating the total execution time and evaluating the performances of each algorithm. The model takes into account both communication and arithmetic delays and the resulting complexity analysis shows that high overall efficiencies can be achieved if the data is carefully assigned into the processors. Another important aspect which is fully discussed for the Alternating Direction algorithm is the importance of selecting the optimal algorithm according to the relative size of the problem and to the architecture. Thus, three different implementations of the Alternating Direction method are compared and it is concluded that they may all perform well for different problem sizes. The complexity analysis of the different algorithms can help to determine the best method to use given the paramaters of the architectures and the problem size.

T. F. Chan, Y. Saad and M. H. Schultz
Yale University
New-Haven, CT

2:00
Algorithms and Architectures: PDE Solution on Parallel Computers
Algorithms for the numerical solution of elliptic equations will be presented, along with implementations on several parallel computers. Particular emphasis will be placed on comparison of algorithms on different architectures including both shared-memory and message-based systems.

Numerical methods to be discussed will include parallel implementations of grid refinement schemes, of multigrid methods and of pre-conditioned conjugate gradient techniques. Architectures involved include the Denelcor HEP, the NYU/IBM Ultracomputer and various Hypercubes.

Oliver A. McBryan, and
Eric F. Van DeVelde
Courant Institute, New York University
New York, NY

2:30
A Comparison of Domain Decomposition Techniques for Elliptic Partial Differential Equations
A number of algorithms based on domain decomposition have been proposed in recent years for the solution of elliptic pde's in two dimensions. These are even more interesting as parallel methods since much of the computational work can be broken into independent pieces and overlapped. We compare the performance of several domain decomposition methods and some conventional global domain methods on a common set of model problems, and examine the parallelizability of each. We point out the unity under certain conditions of methods which have been presented independently, and also identify some problem characteristics which tend to favor certain methods over others in the context of parallelism. The decomposition topologies include simple interfaces (with and without overlap) and cross-points. We consider four communication topologies: shared-memory, a ring, a two-dimensional mesh, and an n-cube.

David E. Keyes and William D. Gropp
Research Center for Scientific Computation
Yale University
New Haven, CT

Wednesday, November 20/*1:30–3:00 PM*
Selected Papers 3B/Providence Hall
SOFTWARE

Chairman: Piyush Mehrotra, Purdue University, West Lafayette, IN

1:30
On Compiling Loop Algorithms onto Systolic Arrays
This paper gives a detailed example of automatic generation of systolic arrays given loop algorithms expressed in a high-level language. One problem examined is that of convolution: four different algorithms are presented, each shown to generate three systolic arrays. An analysis of the relationship of data dependencies in an algorithm and data flow in the corresponding systolic array is presented. This study has direct application in computer-aided design of systolic arrays. Under certain conditions, the designer can specify an algorithm (typically a nested loop) in a high-level language and is automatically presented with several distinct systolic array implementations, each with different I/O and data flow properties.

Keith R. Allen and
Roy P. Pargas
Clemson University
Clemson, SC

2:00
Parallel Algorithms for Network Optimization
New parallel algorithms will be discussed for two types of network optimization problems: nonlinear, traffic assignment problems and generalized networks. These algorithms have very large granularity in that the units of parallel computation are optimization subproblems on subnetworks. Such methods are well-suited for the Crystal multicomputer of the Computer Sciences Department at the University of Wisconsin-Madison. Crystal is a network currently consisting of twenty VAX 11/750's that can efficiently exchange messages via a 10 megabit/second token ring. Implementations of the network algorithms on Crystal will be described.

Robert R. Meyer
University of Wisconsin-Madison
Madison, WI

2:30
Parallel Debugging: A Preliminary Proposal
Research into parallel processing applications has been an area of intense interest at the Los Alamos National Laboratory for several years. We are thus acutely aware of the unique programming problems associated with parallel computing, such as task synchronization errors, hidden data dependencies, and irreproducibility of task execution sequences. Currently available debuggers are woefully inadequate for detecting and isolating these bugs and, as a consequence, the Laboratory has funded a research project to develop a unified approach to error diagnosis in parallel computing. Our preliminary investigations have indicated that an interactive debugger designed for parallel processes should possess such capabilities as control monitoring, data monitoring, dynamic program modification, static program displays, real time program displays, and assertions. We intend to discuss in this paper extensions that will be particularly useful for debugging parallel programs.

James H. Griffin
Computer Research and Applications Group, and
Harvey J. Wasserman
Computer User Services Group
Los Alamos National Laboratory
Los Alamos, NM

*3:00 PM/*Coffee

3:30 PM/CONCURRENT SESSIONS

Wednesday, November 20/*3:30–5:30 PM*
Contributed Papers 5/Poplar Hall
APPLICATIONS 2

Chairman: Marsha Berger, Courant Institute of Mathematical Sciences
New York University, New York, NY

3:30/111/A25
Stencils and Problem Partitionings: Their Influence on Parallel Architectures
Daniel A. Reed, Department of Computer Science, University of Illinois, Urbana, IL; and Loyce M. Adams, Applied Mathematics Group, University of Washington, Seattle, WA; and Merrell L. Patrick, Department of Computer Science, Duke University, Durham, NC

3:45/89/A20
Large-Memory CYBER 205 Simulations of Euler Flows Using a Vector Algorithm
Arthur Rizzi, Aeronautical Research Institute and Royal Institute of Technology, Bromma, Sweden; Charles J. Purcell, ETA Systems, Inc., St. Paul, MN

4:00/97/A22
Parallel Adaptive Numerical Schemes for Hyperbolic Systems of Conservation Laws
Bradley Lucier, Department of Mathematics, Purdue University, West Lafayette, IN; and Ross Overbeek, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL

4:15/77/A18
Adaptive Mesh Refinement for PDEs in 3-D on Parallel Processors
William D. Gropp, Department of Computer Science, Yale University, New Haven, CT

4:30/40/A9
Parallel Algorithms for Parabolic and Hyperbolic Partial Differential Equations
Ohannes Karakashian, Department of Mathematics, University of Tennessee, Knoxville, TN

4:45/81/A18
Concurrent Implementation of the Crank-Nicolson Method for Heat Transfer Analysis
Jonathan B. Ransom, Structures and Dynamics Division, NASA Langley Research Center, Hampton, VA; and Robert E. Fulton, George Washington University, NASA Langley Research Center, Hampton, VA

5:00/34/A8
Development of Parallel Monte Carlo Algorithms: Some Techniques and Their Applications
V. C. Bhavsar, School of Computer Science, University of New Brunswick, Fredericton, N. B., Canada

5:15/15/A4
An Implementation of Multigrid on a Concurrent Processor
Mark E. Bassett, Ametek, Inc., Computer Research Division, Pasadena, CA

Wednesday, November 20/*3:30–5:45 PM*
Contributed Papers 6/Claremont Room
ARCHITECTURE 2

Chairman: Donald Heller, Shell Development Company, Houston, TX

3:30/86/A20
Assessment of the GAPP Systolic Chip for Use in Computationally Intensive Signal Processing
Joseph N. Craig and John Sadowsky, Systems Engineering and Development Corporation, Columbia, MD

3:45/11/A3
A Computing System for Modeling of Large Scale Markov Chains
Simon Ya. Berkovich and Yousry S. El-Gamal, Department of Electrical Engineering and Computer Science, School of Engineering and Applied Science, The George Washington University, Washington, DC

4:00/17/A4
A Study of Data Interlock in VLSI Computational Networks for Sparse Matrix Manipulation
Rami G. Melhem, Department of Mathematics and Statistics, University of Pittsburgh, Pittsburgh, PA

4:15/118/A26
Sparse Matrix Operations in an Interleaved Array Processing Architecture
G. Wolf and J. R. Jump, Department of Electrical and Computer Engineering, Rice University, Houston, TX

4:30/119/A27
A VLSI Architecture for Matrix Multiplication
P. J. Varman, Department of Electrical and Computer Engineering, Rice University, Houston, TX; and I. V. Ramakrishnan, Department of Computer Science, University of Maryland, College Park, MD

4:45/26/A6
System Data Structure Contention Overhead in von Neumann Multiprocessor Configurations
Jerry P. Place, Department of Computer Science, University of Missouri, Kansas City, MO

5:00/120/A27
RISC CPU Design for MIMDs
Henry Dietz, Department of Electrical Engineering and Computer Science, Polytechnic Institute of New York, Farmingdale, NY; and David Klappholz, Department of Computer Science, Center for Distributed Processing, Stevens Institute of Technology, Hoboken, NJ

5:15/109/A24
A Hierarchical Condition Code Structure for Parallel Architectures
Donald M. Chiarulli, Walter G. Rudd, and Duncan A. Buell, Louisiana State University, Baton Rouge, LA

5:30/131/A30
OCCAM, A Concurrent Systems Programming Language
Pete Wilson, INMOS Corporation, Colorado Springs, CO

Wednesday, November 20/*3:30–5:45 PM*
Contributed Papers 7/Providence Hall
LINEAR ALGEBRA 2

Chairman: Charles Van Loan, Cornell University, Ithaca, NY

3:30/65/A15
The Parallel Solution of Toeplitz Systems By the Symmetric Q. I. F. Method
David J. Evans, Department of Computer Studies, University of Technology, Loughborough, England

3:45/82/A19
Complexity Results for Parallel Linear Systems Solvers
M. Cosnard and Y. Robert, Laboratoire TIM3—IMAG, France

4:00/79/A18
Equation-solving on the CRAY-2
D. A. Calahan, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI

4:15/76/A17
New Vector Linear Algebra Algorithms for Computers with a Cache Memory
Fred G. Gustavson, Gordon Slishman, and Bryant Tuckerman, IBM T. J. Watson Research Center, Yorktown Heights, NY

4:30/8/A2
Algorithm-based Fault Tolerance for Parallel Matrix Equation Solvers
Franklin T. Luk, School of Electrical Engineering, Cornell University, Ithaca, NY; and Haesun Park, Department of Computer Science, Cornell University, Ithaca, NY

4:45/20/A5
A New Parallel Algorithm for Banded Linear Systems
Michael A. Kalogerakis, Department of Mathematics and Computer Science, Clarkson University, Potsdam, NY

5:00/108/A24
A Recursive Bisection Method for the Solution of Band Matrix Linear Systems of Equations with Parallel Processors
Alex Bykat, Center for Computer Applications, University of Tennessee, Chattanooga, TN

5:15/87/A20
Matrix Computation on the Intel Hypercube
Cleve Moler, Intel Scientific Computers, Beaverton, OR

5:30/107/A24
Parallel Implementation of Multifrontal Schemes
Iain S. Duff, AERE Harwell, Didcot, Oxon, England

Note: The numbers that follow the individual times for each paper represent the numerical order and page number of the abstract. For papers with more than one author, an underline is used to denote the author who will present the paper.

Wednesday, November 20/*3:30-5:45 PM*
Contributed Papers 8/Brandon Room
SOFTWARE 2

Chairman: Shahid Bokhari, ICASE, Hampton, VA

3:30/33/A8
Interactive Graphics Tools for Monitoring and Evaluating Parallel Computations
George Cybenko, Department of Computer Science, Tufts University, Medford, MA and Statistics Center, Massachusetts Institute of Technology, Cambridge, MA; and Alva Couch, Department of Mathematics, Tufts University, Medford, MA

3:45/91/A21
Your Favorite Parallel Algorithm Might Not Be as Fast as You Think
David C. Fisher, Department of Mathematics, Harvey Mudd College, Claremont, CA

4:00/38/A9
A System-Supported Workload Balancing Scheme for Cooperating Reduction Machines
Claudia Schmittgen, André Gerdts, Jochen Haumann, Werner Kluge, Michael Woitass, Gesellschaft für Mathematik und Datenverarbeitung, Sankt Augustin, Germany and University of Bonn, Bonn, Germany

4:15/39/A9
Performance Tuning Tricks on a VAX Multiprocessor
Paul Clark, Digital Equipment Corporation, Littleton, MA

4:30/10/A3
Performance Evaluation Models for Distributed Computing
Catherine E. Houstis, Department of Electrical Engineering; and Elias N. Houstis and John R. Rice, Department of Computer Sciences, Purdue University, West Lafayette, IN

4:45/44/A10
A Highly Parallel Near Neighbors Algorithm of Order N Based on a Monotonic Logical Grid
S. G. Lambrakos and J. P. Boris, Laboratory for Computational Physics, Naval Research Laboratory, Washington, DC

5:00/1/A1
Synthesizing Parallel Algorithms by Transforming Recursion Equations
Marina C. Chen, Department of Computer Science, Yale University, New Haven, CT

5:15/61/A14
Embedding Binary Trees into a Hypercube
William D. Gropp and Ilse C. F. Ipsen, Department of Computer Science, Yale University, New Haven, CT

5:30/53/A12
An Algorithm for Parallel Computation of Approximate Empirical Distributions
John A. Kapenga, Department of Computer Science, Western Michigan University, Kalamazoo, MI

6:00 PM-10:00 PM/York Hall
Hardware/Software Exhibits
Free Beer on the Promenade

Thursday, November 21/AM

8:30 AM/Poplar Hall
Research Presentations 7 and 8
Chairman: Jack Dongarra
Argonne National Laboratory
Argonne, IL

PARALLEL ALGORITHMS FOR LEAST SQUARES PROBLEMS
Charles Van Loan
Cornell University
Ithaca, NY

ALGORITHMS AND EXPERIMENTS FOR PARALLEL LINEAR SYSTEMS SOLVERS
Ahmed H. Sameh
University of Illinois
Urbana, IL

10:00 AM/Coffee

10:30 AM/*CONCURRENT SESSIONS*

Thursday, November 21/*10:30 AM-12 NOON*
Selected Papers 4A/Poplar Hall
APPLICATIONS

Chairman: Oliver McBryan, Courant Institute of Mathematical Sciences, New York University, New York, NY

10:30
Solving Equations of Motion on a Virtual Tree Machine
The equations of motion for rigid links connected at hinges to form a tree-like structure are computationally expensive to compute, but are readily amendable to a parallel implementation. A parallel algorithm for such a solution could provide a real-time simulation of the dynamic behavior of robots or animated objects. However, the design of a parallel algorithm poses problems not encountered in the sequential case. Difficulties such as synchronizing processes, process to processor allocation, and communication overhead must be addressed. The facility used to investigate these problems is a Virtual Tree Machine multi-computer, implemented on a network of autonomous VAX-11/780's and SUN-2 processors each running the UNIX operating system.
W. W. Armstrong, T. A. Marsland, M. Olafsson, and J. Schaeffer
University of Alberta
Edmonton, Alberta
Canada

11:00
Adapting a Navier-Stokes Code to the ICL-DAP
In this paper I discuss the adaptation of a Navier-Stokes code, developed on a serial computer, to the SIMD Distributed Array Processor. The architecture of the DAP, the parallel language, DAP Fortran, as well as the algorithm will be briefly described. The modification of the algorithm necessary to fit the DAP architecture will be presented. Key sections of the implementation and performance results will be given. Finally, some implications of the restructuring will be discussed. In particular, the restructuring has revealed a segmented structure within the subroutines, suggesting how the algorithm can be adapted to MIMD architectures.
Chester E. Grosch
Old Dominion University
Norfolk, VA and
British Maritime Technology
Teddington, England.

11:30
Implementation of a Navier-Stokes Algorithm on a Local Memory Parallel Processor
The operation of several candidate algorithms for simulating turbulent flow on the special purpose "Navier-Stokes Computer" under development at Princeton University is considered. The Navier-Stokes Computer is a parallel processor consisting of multiple concurrent processors arranged in a multi-dimensional lattice, in which a simple "nearest neighbor" interconnect architecture is employed for communication between processors. The algorithms considered involve the iterative solution of a system of Poisson and Helmholtz equations, which are developed from a time discretization of the incompressible Navier-Stokes equations. The performance of the algorithms, with emphasis on the computation rate for a complete time step, will be considered.
Steven E. Krist
Joint Institute for Advancement of Flight Sciences
NASA Langley Research Center
Hampton, VA

Thursday, November 21/*10:30 AM-12:00 NOON*
Selected Papers 4B/Providence Hall
ENVIRONMENTS

Chairman: Harry Jordan, University of Colorado, Boulder, CO

10:30
Transformation of Sequential Programs for Multiprocessor Execution: The Blaze Environment
Multiprocessor architectures can be programmed using a CSP-like language, which requires explicit specification of tasks, or one can use a sequential language, leaving it to the compiler to extract parallelism. One example of this latter approach is the BLAZE language and program transformation system. We focus here on two typical numerical kernel procedures, a Gaussian elimination procedure and a Jacobi iteration procedure.
We show how these procedures would be described in both BLAZE and CSP-like languages and show how the BLAZE program transformation system would restructure the BLAZE procedures into a collection of communicating tasks.
Piyush Mehrotra
Purdue University
West Lafayette, IN
Dennis Gannon
Indiana University
Bloomington, IN
John Van Rosendale
University of Utah
Salt Lake City, UT

11:00
A Development Environment for Scientific Parallel Programs
This paper describes a development environment for parallel scientific code. The environment uses Percolation Scheduling, a transformational system for parallelism extraction, and an interactive profiling system to give the user control over the parallelization process while reducing the burdensome details of architecture, correctness-preservation and synchronization. Through a graphical interface the user suggests what should be done in

Note: The numbers that follow the individual times for each paper represent the numerical order and page number of the abstract. For papers parallel, while the system performs the actual changes using semantics-preserving transformations. If a request cannot be satisfied, the system reports the problem causing the failure. The user may then help eliminate the problem by supplying guidance or information not explicit in the code.

Alexandru Nicolau
Cornell University
Ithaca, NY

11:30
Processor Self-Scheduling in Large Multiprocessor Systems
This paper presents two processor self-scheduling schemes for large multiprocessor systems.

The first scheme is for a single parallel program, which is represented by a sequence of parallel Do-loops when outer serial loops are unrolled.

The second is a multiprogrammed processor self-scheduling scheme, in which each parallel program can be represented by a precedence graph of parallel Do-loops.

In both schemes, by using synchronization instructions to shared variables, the processors can schedule for themselves and execute parallel programs coordinately without calling the operating system kernel.

At the end of the paper, an implementation of parallel queue is presented.

Peiyi Tang, Pen-Chung Yew, and Chuan-Qi Zhu
Center for Supercomputing Research and Development
University of Illinois at Urbana-Champaign
Urbana, IL

12:00 Noon/Lunch

Thursday, November 21/PM

1:30 PM/CONCURRENT SESSIONS

Thursday, November 21/*1:30–4:15 PM*
Contributed Papers 9/Poplar Hall
APPLICATIONS 3

Chairman: Jay Lambiotte, NASA Langley Research Center, Hampton, VA

1:30/21/A5
Vectorization of Multidimensional Finite Difference Schemes With Application to Stefan-Type Problems
D. G. Wilson, Mathematical Sciences Section, Engineering Physics and Mathematics Division, Oak Ridge, TN

1:45/18/A4
An Evaluation of the Architecture and Performance of a Ten-Processor ELXSI System 6400
Melvin R. Scott, Applied Computer Graphics Division; and Walter H. Vandevender, Applied Mathematics Division, Sandia National Laboratories, Albuquerque, NM

2:00/27/A6
Block Multitasking of a Model, Lagrangian, Wave Code on the Denelcor HEP-1 ELXSI 6400, and CRAY XMP/48
D. L. Hicks, Department of Mathematical and Computer Sciences, Michigan Technological University, Houghton, MI; and M. R. Scott, Applied Computer Graphics Division and A. H. Treadway, Applied Mathematics Division, Sandia National Laboratory, Albuquerque, NM

2:15/49/A11
Concurrent Algorithm Development for Shared Memory Environments
G. G. Weigand and R. E. Benner, Fluid and Thermal Sciences Department, Sandia National Laboratories, Albuquerque, NM

2:30/51/A12
Concurrent Algorithms for Convectively Dominated Fluid Flows
G. R. Montry, G. G. Weigand, and M. R. Baer, Fluid and Thermal Sciences Department, Sandia National Laboratories, Albuquerque, NM

2:45/42/A10
Hydrocodes on the HEP, Part II
Darrell L. Hicks, Department of Mathematical and Computer Sciences, Michigan Technological University, Houghton, MI; and Joseph F. McGrath, KMS Fusion, Inc., Target Design & Analysis, Ann Arbor, MI

3:00/52/A12
Adaptive Two-Dimensional Integration on the HEP
Elise de Doncker and John A. Kapenga, Department of Computer Science, Western Michigan University, Kalamazoo, MI

3:15/16/A4
Parallel Solutions of ODE's by Multi-block Methods
Moody T. Chu and Hans Hamilton, Department of Mathematics, North Carolina State University, Raleigh, NC

3:30/57/A13
FFT Methods for an MIMD Machine
William L. Briggs, Department of Mathematics, University of Colorado, Denver, CO

3:45/74/A17
An Efficient Vector Formulation of the Mixed-radix FFT Algorithm
Ramesh C. Agarwal and James W. Cooley, IBM T. J. Watson Research Center, Yorktown Heights, NY

4:00/129/A30
A Multiprocessor Implementation of a Two Dimensional Vortex Method
Scott B. Baden, Computer Science Division, University of California, Berkeley, CA

Thursday, November 21/*1:30–4:00 PM*
Contributed Papers 10/Claremont Room
ARCHITECTURE 3

Chairman: Chester Grosch, Old Dominion University, Norfolk, VA

1:30/43/A10
Efficient Solutions to the Parallel Prefix Computation Problem
Atul Prakash and Yih-farn Chen, Computer Science Division, University of California, Berkeley, CA

1:45/64/A15
A New Algorithm for Parallel Carry—Lookahead Addition
Elena Papadopoulou, Department of Mathematics and Computer Science, Clarkson University, Potsdam, NY

2:00/75/A17
Vector Elementary Functions
Ramesh C. Agarwal, James W. Cooley, Fred G. Gustavson, and Bryant Tuckerman, IBM T. J. Watson Research Center, Yorktown Heights, NY

2:15/94/A21
Several Parallel Algorithms for Polynomials
J. H. M. Andriessen, Department of Mathematics and Informatics, Delft University of Technology, Julianalaan, The Netherlands

2:30/100/A23
Approximating Zeros in a Message-Based Multi-Processor Environment
Michael R. Leuze and Paul M. Sullins, Department of Computer Science, Vanderbilt University, Nashville, TN

2:45/70/A16
Efficient Parallel Algorithms for Controllability, Root-Separation and Cauchy Index Problems for Polynomials
B. N. Datta and Karabi Datta, Department of Mathematical Sciences, Northern Illinois University, DeKalb, IL

3:00/31/A7
A Parallel Version of the Continued Fraction Integer Factoring Algorithm
M. Wunderlich, Department of Mathematics, Northern Illinois University, DeKalb, IL; and H. C. Williams, Department of Computer Science, University of Manitoba, Winnipeg, Manitoba, Canada

3:15/98/A22
Inherent Parallelism in the Determination of Prime Implicants
Frank David Allen, Hughes Aircraft Company, Radar Systems Group, Los Angeles, CA

3:30/35/A8
On Pseudorandom Number Generators for Parallel Computers
V. C. Bhavsar, School of Computer Science, University of New Brunswick, Fredericton, N. B., Canada

3:45/127/A29
Conjugate Gradients on Optical Crossbar Interconnected Multiprocessor
Alastair D. McAulay, Central Research Laboratories, Texas Instruments Incorporated, Dallas, TX

Thursday, November 21/*1:30–4:15 PM*
Contributed Papers 11/Providence Hall
LINEAR ALGEBRA 3

Chairman: Danny Sorensen, Argonne National Laboratory, Argonne, IL

1:30/72/A16
Analysis of Incomplete Matrix Factorization Methods for Vector and Parallel Computers
Owe Axelsson, Department of Mathematics, Catholic University, Toernooiveld, Nijmegen, The Netherlands

1:45/50/A11
Applications of Multifrontal Methods in Conjugate Gradient Algorithms for Concurrent Finite Element Analysis
R. E. Benner and G. R. Montry, Fluid and Thermal Sciences Department, Sandia National Laboratories, Albuquerque, NM

2:00/6/A2
Approximate Inverses: A Family of Naturally Vectorizing Preconditioners
Horst D. Simon, Boeing Computer Services, Engineering Technology Applications Division, Tukwila, WA

Note: The numbers that follow the individual times for each paper represent the numerical order and page number of the abstract. For papers with more than one author, an underline is used to denote the author who will present the paper.

2:15/45/A10
Parallel Processing of a Preconditioned Biconjugate-Gradient Algorithm on Cray Supercomputers
Alice E. Koniges, National Magnetic Fusion Energy Computer Center, Lawrence Livermore National Laboratory, Livermore, CA

2:30/5/A2
The Solution of Large Dense Generalized Eigenvalue Problems on the Cray X-MP/24 with SSD
Roger Grimes, John Lewis, and Horst Simon, Boeing Computer Services, Engineering Technology Applications Division, Tukwila, WA; and Henry Krakauer and Su-Huai Wei, Department of Physics, The College of William and Mary, Williamsburg, VA

2:45/14/A4
A Parallel Shifted QR Algorithm
Robert A. van de Geijn, Department of Computer Science, University of Maryland, College Park, MD

3:00/59/A14
A Proof of a Vector Oriented Algorithm for Finding Eigenvalues of Real Symmetric Matrices
William I. Thacker, Department of Computer Science, Winthrop College, Rock Hill, SC; and Layne T. Watson, Department of Computer Science, Virginia Tech Institute, Blacksburg, VA

3:15/48/A11
Solving the Symmetric Tridiagonal Eigenvalue Problem on the Hypercube
Ilse C. F. Ipsen and Elizabeth Jessup, Department of Computer Science, Yale University, New Haven, CT

3:30/104/A23
A Parallel Algorithm for the Symmetric Tridiagonal Eigenvalue Problem
John G. Lewis and Horst D. Simon, Boeing Computer Services, Tukwila, WA; and Bahram Nour-Omid, Center for Pure and Applied Mathematics, University of California, Berkeley, CA

3:45/102/A23
A Multiprocessor Algorithm for the Tridiagonal Eigenvalue Problem
Sy-Shin Lo, Bernard Philippe, and Ahmed H. Sameh, Center for Supercomputing Research and Development, University of Illinois, Urbana, IL

4:00/128/A29
A Domain Decomposed Fast Poisson Solver for Multiprocessors
Tony F. Chan and Diana C. Resasco, Department of Computer Science, Yale University, New Haven, CT

Thursday, November 21/*1:30–4:15 PM*
Contributed Papers 12/Brandon Room
SOFTWARE 3

Chairman: Joel Saltz, ICASE, Hampton, VA

1:30/4/A1
An Optimizing Precompiler for Finite-Difference Computations on the Cyber 205
John Gary, Scientific Computing Division, National Bureau of Standards, Boulder, CO; and Lloyd Fosdick, Department of Computer Science, University of Colorado, Boulder, CO

1:45/88/A20
Massively Parallel Communication
John E. Dorband, Department of Computer Science, Pennsylvania State University, University Park, PA; Space Data and Computing Division, NASA Goddard Space Flight Center, Greenbelt, MD

2:00/73/A17
Fortran to Fortran: Automatic Conversion of a Software Library for an Attached Processor
Anthony A. Billings and William F. Eddy, Department of Statistics, Carnegie-Mellon University, Pittsburgh, PA

2:15/58/A13
Writing a Specialized Map Library for a Vector Computer
Lowell Palecek, (EML) Programming Group, Sperry Corporation, St. Paul, MN

2:30/67/A15
The Loral DATAFLO System
John Van Zandt, Gale Williamson and Ian Kaplan, Loral Instrumentation, San Diego, CA

2:45/68/A16
The Loral Data Graph Language
Ian Kaplan, Loral Instrumentation, San Diego, CA

3:00/121/A27
Sequential Languages for Programming Highly-Parallel Computers
Henry Dietz, Department of Electrical Engineering and Computer Science, Polytechnic Institute of New York, Farmingdale, NY; and Ken Stein and David Klappholz, Department of Computer Science, Center for Distributed Processing, Stevens Institute of Technology, Hoboken, NJ

3:15/55/A13
Multi-Language Code Generation for Sequential Architectures
Gregory Aharonian, Source Translation & Optimization, Belmont, MA

3:30/56/A13
Multi-Language Code Generation for Non-Sequential Architectures
Gregory Aharonian, Source Translation & Optimization, Belmont, MA

3:45/19/A5
A Modified Remes Algorithm Which Converges
Yi-ling F. Chiang, Department of Computer and Information Sciences, New Jersey Institute of Technology, Newark, NJ

4:00/126/A29
Minimizing Main Memory/Register Traffic
Ashfaq A. Munshi, IBM San Jose Research Center, San Jose, CA; and Karl M. Schimpf, Board of Studies in Computer and Information Sciences, University of California, Santa Cruz, CA

4:00 PM/Program Adjourns

Note: The numbers that follow the individual times for each paper represent the numerical order and page number of the abstract. For papers with more than one author, an underline is used to denote the author who will [illegible]

ABSTRACTS: CONTRIBUTED PAPERS AND POSTER PRESENTATIONS*

#1
Synthesizing Parallel Algorithms by Transforming Recursion Equations

A high level of data interdependency always occurs in any interesting computation. Parallel processing is concerned with distributing computations to a large assemblage of parallel processes in such a way that the processing of interdependent data is constrained to local interactions, thus allowing large scale parallelism to spread throughout the assemblage, and collectively, these local interactions accomplish the target task. The problem of devising correct distribution schemes can be tackled in a fundamental, systematic manner by establishing models of parallel processing systems, devising a formal and rigorous language for describing parallel computations, and developing synthesis methods upon such foundations. The speaker will present a formal framework for parallel processing, and a two-stage synthesis method consisting of **algorithm transformation** and **space-time mapping**. The method has been successful in generating systolic designs that supercede existing designs in performance, and parallel algorithms on various network topologies, such as square meshes, hexagonal meshes, hypercubes, and meshes of trees.

Marina C. Chen
Department of Computer Science
Yale University
P.O.Box 2158, Yale Station
New Haven, CT06520

#2
High Level Parallelism in Hierarchy of GPST Inversion Algorithm

The Generalized Pulse-Spectrum Technique (GPST) is a versatile and efficient iterative numerical algorithm for solving multi-parameter inverse problems (to determine the unknown coefficients, geometries of the space domain, initial-boundary values and sources from the measured data in the space - time domain or the space - complex frequency domain) of a system of nonlinear partial differential equations. It can be made super efficient by not only implementing the parallel algorithms in solving the corresponding direct problems of partial differential equations at the low level in the hierarchy of GPST inversion algorithm but also implementing parallelism and methods of domain-decomposition at the high level in the hierarchy of GPST inversion algorithm.

Y. M. Chen
Department of Applied Mathematics and Statistics
State University of New York at Stony Brook
Stony Brook, NY 11794

#3
Schwarz Splitting and Parallel Computation

The only promising architectures for providing the needed hundredfold increase in performance comprise the "massively" parallel devices embedded in some smart network topology. The numerical analyst must therefore study new ways to decompose large scale scientific problems for concurrent processors. We present here some new optimal algorithms for the solution of elliptic P.D.E's using domain decomposition, for which the convergence rates are independent of the mesh size. In particular the computational work required for achieving a fixed accuracy is proportional to the number of discrete unknowns.

Wei Pai Tang
Computer Science Departement
Stanford University
Stanford, CA. 94305

#4
An Optimizing Precompiler for Finite-Difference Computations on the Cyber 205

The precompiler is designed to generate efficient code for finite-difference computations. It accepts a program written in Fortran 77 extended to include Fortran 8X array operations, and translates it into Cyber 205 Fortran, or Fortran 77 for debugging. Particular attention is given to generating gather/scatter operations and bit vector operations in an optimal way so typical algorithms -- explicit difference, conjugate gradient, implicit splitting -- are efficiently realized. Optimizations also include regrouping factors in vector-scalar expressions to minimize vector operations. The precompiler is written in Fortran 77 and is running on a Masscomp and a Cyber 750.

John Gary
Scientific Computing Division
National Bureau of Standards
Boulder, CO 80303

Lloyd Fosdick
Department of Computer Science
University of Colorado
Boulder, CO 80309

* All contributed abstracts on pages A1 to A30 are numbered in the order that they were received at the SIAM office.

Abstracts: Contributed Papers and Poster Presentations

#5

The Solution of Large Dense Generalized Eigenvalue Problems on the Cray X-MP/24 with SSD

Problems in physics and material sciences often require the numerical solution of large full generalized eigenvalue problems. The problems are symmetric and definite, and in most cases only a few eigenvalues and vectors are required (typically about 10%). However, with a problem size ranging between 1000 and 2000 an in-core solution of these problems using standard eigenvalue routines from EISPACK is memory-limited even on today's supercomputers.

Here we demonstrate how a block shifted and inverted Lanczos routine, originally developed for sparse eigenvalue problems in dynamic analysis, can be used to solve these large dense problems. The Lanczos routine goes easily out-of-core, and permits the use of the Solid State Storage device on the Cray X-MP. We present some numerical results for problems of size up to 1500 arising in quantum mechanical band structure calculations.

Roger Grimes[1)], Henry Krakauer[2)], John Lewis[1)]

Horst Simon[1)], and Su-Huai Wei[2)]

1) Boeing Computer Services M/S 9C-01
Engineering Technology Applications Division
565 Andover Park West
Tukwila, WA 98188

2) Department of Physics
The College of William and Mary
Williamsburg, VA 23185

#6

Approximate Inverses: A Family of Naturally Vectorizing Preconditioners

A new type of preconditioners is proposed for the acceleration of conjugate-gradient-type methods for the solution of sparse nonsymmetric linear equations. This new family of preconditioners is based on approximate inverses, which can be computed with about the same effort as an incomplete LU factorization of the coefficient matrix. We present two algorithms for computing approximate inverses with band structure, because the band structure allows a natural vectorization of the preconditioning phase. Numerical results obtained on a CRAY X-MP demonstrate speed and efficiency of this new class of preconditioners.

Horst D. Simon
Boeing Computer Services
565 Andover Park West, M/S 9C-01
Tukwila, WA 98188

#7

Interconnection Topology for a Completely Distributed MIMD Computer

Future Generations Computer Architectures project at MITRE is designing a completely distributed fault-tolerant architecture for a MIMD computer system. It is to contain a large number of processors with private memory (smart memory cells) interconnected in such way that the route between any two processors is quite short.

This paper presents a design of the topology of the connections between the processors that can be used when wire crossings are permitted. The graph representing this topology is transitive and highly symmetrical. This results in a simple fault-tolerant routing algorithm and a uniform traffic distribution among processors. We present the routing algorithm and the results of simulations used to evaluate this design.

Viera K. Proulx
The MITRE Corporation
Burlington Road
Bedford, Massachusetts 01730
Department D-82
Signal Processing and Electronic Warfare
Experimentation and Data Analysis Division

#8

Algorithm-based Fault Tolerance for Parallel Matrix Equation Solvers

A weighted checksum scheme has been proposed by Jou and Abraham for computing an LU-factorization on a multiprocessor array. Their method is very efficient for detecting a transient error, but quite impractical for correcting it since the computations must be rolled back. In this paper, we show how to avoid the roll back, and develop a checksum scheme for solving the resultant triangular systems of linear equations. Finally, we discuss the numerical problems associated with weighted checksums and suggest possible remedies.

Franklin T. Luk, School of Electrical Engineering; Haesun Park, Department of Computer Science
Cornell University, Ithaca, NY 14853

#9

The Parallel Solution of a Recursive Least Squares Problem in Signal Processing

We discuss the parallel solution of a direction finding problem in signal processing. The problem requires least squares minimization with a single equality constraint, and is challenging in several aspects. First, we desire only the most recent element of the residual vector, and not the solution vector. Second, we need to solve variant problems that have different constraints but the same data matrix. Third, we want to update the residuals recursively as new rows of data come in.

Franklin T. Luk
School of Electrical Engineering
Cornell University
Ithaca, New York 14853

Sanzheng Qiao
Center for Applied Mathematics
Cornell University
Ithaca, New York 14853

#10

Performance Evaluation Models for Distributed Computing

This paper presents two models for the performance predicting for pairs of distributed algorithms and architectures. One model is for shared memory based multiprocessors and the other for non-shared memory MIMD architectures. The algorithm/architecture performance is determined in terms of the mapping of the algorithm to the architecture. The mappings considered try to optimize various performance objective functions such as communication cost and processing load under various software/hardware requirements. This study is focused on numerical algorithms for elliptic partial differential equations, and several existing multiprocessor machines.

Catherine E. Houstis
Department of Electrical Engineering

Elias N. Houstis
John R. Rice
Department of Computer Sciences
Purdue University
W. Lafayette, IN 47907

#11

A Computing System for Modeling of Large Scale Markov Chains

Discrete Markov chain representation of the random walk scheme is proved to be useful in simulation of physical problems, operations research, and information processing. This method of simulation lends itself directly to parallel processing by virtue of its uniform sequence of operations, small intermediate storage requirements, low accuracy, and simple control.
A highly parallel systolic architecture is developed for modeling of large-scale Markov chains. The proposed architecture is mainly composed of a number of identical and independent cells arranged as a pipeline. The computation process is organized as a continous data flow. This eliminates the need for addressing, pointers, broadcasting of data, scheduling, and control. The simplicity of the cell structure combined with the modularity of the system make it well suited to VLSI and the potential WSI implementation.

Simon Ya. Berkovich
Yousry S. El-Gamal
Department of Electrical Engineering and Computer Science
School of Engineering and Applied Science
The George Washington University
Washington, D.C. 20052

#12

Some Parallel Algorithms for Matrix Structural Analysis

The fundamental problem of linear elastic analysis is that of finding vectors of stresses and strains, given a finite element model of a structure and a set of external applied loads. This leads to a constrained minimization problem of considerable magnitude for large, complex structures. In this talk some alternatives to the displacement equations (normal equations in least squares terminology) in structural analysis are considered. Parallel implementations of algorithms based upon (1) null space methods, (2) weighted least squares methods, and (3) block cyclic SOR methods are discussed and some computational experience by graduate students at N.C. State University in implementing these algorithms on a Denelcor HEP and on a CRAY X-MP is described for some practical structural data.

Robert J. Plemmons
Departments of Computer Science and Mathematics
North Carolina State University
Raleigh, North Carolina 27695-8205

#13

Parallel Solution of Sparse Line ions
Using the Inverse Matrix

In repeated solutions linear equations Ax=b, seq s use the factors of the ma

WITHDRAWN

Using the inve gives $x=A^{-1}b$, and all the mul e done in parallel. After opt partitioning, the factor ally 2 to 5 percent full. An i uter would achieve a spee ound 90, but in practice memory e limits the performance. Simulat lts show the feasibility of two designs: a preprogrammed crossbar network of processors and memory banks, and a data-flow architecture with a pipelined addition processor.

D. M. MacGregor, M. K. Enns
Electrocon International, Inc.
611 Church Street
Ann Arbor, Michigan 48104

F. L. Alvarado
Elect. & Compt. Engr. Department
The University of Wisconsin
1425 Johnson Drive
Madison, WI 53706

#14

A Parallel Shifted QR Algorithm

This paper describes a parallel QR algorithm scheme to solve the algebraic eigenvalue problem. The computation is organized in such a way that there is no need to wait for an iteration to finish completely before starting the next. The result is an algorithm which takes 3n+c "time units" per iteration on $n^2/2$ processors, where n is the dimension of the matrix, and c is a constant. A shift and/or deflation scheme can be added at a cost independent of n. The algorithm will be implemented on ZMOB, an experimental parallel processor, using the DOMINO operating system.

Robert A. van de Geijn
Computer Science Department
University of Maryland
College Park, MD 20742

#15

An Implementation of Multigrid on a Concurrent Processor

A multigrid program has been written for a concurrent processor. The unique machine architecture required several modifications of the algorithm used. These were in the smoothing operator, and the method of solution on the coarsest grid. Various remedies to each of these problems were examined theoretically and numerically. The best approaches employed ILU for the smoother, and an iterative solution on the coarsest grid.

The resulting program was run on a variety of test problems on a 64 node machine based on the hypercube architecture. The results show that multigrid compared favorably with the other methods tested.

Mark E. Bassett
Ametek, Inc./Computer Research Division
80 South Lake Avenue, Suite 550
Pasadena, California 91101

#16

Parallel Solutions of ODE's by Multi-block Methods

We generalize the notion of linear multi-step methods for solving ordinary differential equations to a class of multi-block methods in which step values are all obtained together in a single block advance accomplished by allocating the parallel tasks on separate processors. We describe the basic formulation and give examples to demonstrate the existence of such a scheme. We also consider the possibility of forming a predictor-corrector type combination and study the resulting stability problem. Actual implementation of one of these methods has been experimented on the Denelcor HEP machine.

Moody T. Chu

Hans Hamilton

Department of Mathematics
North Carolina State University
Raleigh, NC 27695-8205

#17

A Study of Data Interlock in VLSI Computational Networks for Sparse Matrix Manipulation

The general question addressed in this study is: Are regular VLSI networks suitable for sparse matrix computations?. More specifically, we consider a special purpose self-timed network that is designed for certain specific dense matrix computation. We add to each cell in the network the capability of recognizing and skipping operations that involve zero operands, and then ask how efficient this resulting network is for sparse matrix computation?.

In order to answer this question, we generalize the concept of computation wavefronts to include irregular flow of data. The propagation properties of such fronts provides a means for the study of the effect of data interlock on the performance of self-timed networks.

Rami G. Melhem
Department of Mathematics and Statistics
University of Pittsburgh
Pittsburgh, PA 15260

#18

An Evaluation of the Architecture and Performance of a Ten-Processor ELXSI System 6400

The ELXSI System 6400 is a modular multiple processor computer whose architecture accommodates up to ten central processing units interfaced to a high-speed central bus. Sandia National Laboratories recently acquired a ten-processor ELXSI System 6400 for evaluation and for research and development of software and parallel algorithms. Some results of multi-tasking several types of computer codes including some large scientific applications codes will be presented.

Melvin R. Scott
Applied Computer Graphics Division, 2644

Walter H. Vandevender
Applied Mathematics Division, 2646

Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185

#19

A Modified Remes Algorithm which Converges

For given n and m, the Remes algorithm used to find a best approximation $R^*_{n,m}(X)$ to a continuous function in a finite interval may not converge. At every stage of the iterative process, the algorithm requires that at n+m+2 points the error function has equal magnitude with alternating signs. However, in a degenerate case, the optimal error function $E^*_{n,m}(X)$ does not have n+m+2 extremes. The requirement may not be fulfilled, and the algorithm fails to converge. In this paper, an initial approximation with property y is introduced. A modified Remes algorithm combined with this special initial approximation guarantees that the error function at every stage of the iterative process oscillates at least n+m+1 times. This assures the convergence of the algorithm. The least square approximation has property y as well as the generalized Pade approximation. Numerical examples are also included. In these examples, the best approximations obtained by using the modified algorithm are given in the standard, nonstandard, and degenerate forms as defined.

Yi-ling F. Chiang

Computer and Information Science Department

New Jersey Institute of Technology

Newark, New Jersey 07102

#20

A New Parallel Algorithm for Banded Linear Systems

A new parallel algorithm for the solution of banded systems of linear equations is presented. This algorithm, which is more effective for small bandwidth, is most suitable when the number of available processors is only a fraction of the order of the system. Numerical results obtained for some typical examples, and a comparison with existing parallel algorithms is also presented.

Michael A. Kalogerakis
Department of Mathematics and Computer Science
Clarkson University
Potsdam, NY 13676

#21

Vectorization of Multidimensional Finite Difference Schemes With Application to Stefan-Type Problems

Vectorization tactics for multidimensional finite difference equations and the result of their application to an explicit scheme for an enthalpy formulation of three dimensional Stefan-type problems are presented. The vectorization tactics consist of: (1) unfolding a three dimensional array of node points into one long vector, (2) factoring multidimensional calculations into several one-dimensional updates that can be performed simultaneously, and (3) using boolean variables as multipliers in place of logical tests within loops. A special feature of Stefan-type problems, which is easily accommodated, is a discontinuous change of material properties as the phase change proceeds.

D. G. Wilson
Mathematical Sciences Section
Engineering Physics and Mathematics Division
Building 9207A, Mail Stop 2
P.O. Box Y
Oak Ridge, Tennessee 37831

#22

Efficient Parallel Vectorized Algorithms for Successive Over Relaxation

This presentation describes algorithms for efficient successive over relaxation (SOR) codes on an important new class of computers-MIMD parallel vector machines. SOR is a fast iterative method particularly useful for solving linear systems resulting from the discretization of elliptic partial differential equations. The speaker will discuss how three characteristics of a parallel vectorized SOR algorithm - ordering, synchronization, and communication - control its efficiency. Experimental results will be presented which demonstrate that optimal speedup is obtained when using maximal length vectors, asychronous operation, and chaotic communication.

Oliver D. Edwards

Carnegie - Mellon University

Mathematics Dept.

Pittsburgh PA 15213

Abstracts: Contributed Papers and Poster Presentations

#24

Hypercubes: Architecture and Algorithms

The architecture of a parallel computer system and algorithms for its use are presented. The architecture emphasizes reliability and performance. A very high bandwidth Input/Output system is incorporated in the design. The algorithms are optimized for hypercube systems and cover linear equations and signal processing.

John F. Palmer
NCUBE Corporation
915 E. LaVieve Lane
Tempe, AZ 85284

#25

The Shared Memory Hypercube

We investigate the use of a hypercube packet switching network as a shared memory server for large N multiprocessors. Using a high performance switch node [1], we develop a packet switched memory server capable of providing high bandwidth vector access to a shared memory. The network exhibits adaptive behavior, absorbing conflicts as a vector operation proceeds, and delivers full bandwidth to all processors simultaneously. The adaptive behavior of the network is confirmed for a variety of addressing patterns using simulation. The high vector performance of the network, in combination with its ability to provide for local memory in the same hardware, makes in a very promising choice for large N shared memory multiprocessors.

[1] E.D. Brooks, "The Shared Memory Hypercube," LLNL, Livermore, UCRL 92479, 1985.

Eugene D. Brooks III
Parallel Processing Project
Lawrence Livermore National Laboratory
Livermore, CA 94550

#26

System-Data Structure Contention Overhead in von Neumann Multiprocessor Configurations

Non-von Neumann architecture is a significant research topic for those interested in the performance of algorithms in a multiprocessor configuration[1]. It is recognized that non-von Neumann architectures offer great promise, nevertheless, the more traditional architectural approaches to multiprocessor configurations should not be ignored.

The basic theme here is that contention for system data structures is a major performance impediment in existing von Neumann based multiprocessor configurations, and, techniques exist to substantially reduce this contention – therefore, the amount of processor power that may be applied to solving user problems will be increased.

This paper will present simulation studies of the contention for system data structures in a multiprocessor configuration and will examine two techniques to reduce this contention.

[1] Arvind and R.A. Innucci. A Critique of Multiprocessing von Neumann Style, *Proc. 10th Ann. Symp Comp Arch*, June 1983, pp. 426-436.

Jerry P. Place
Department of Computer Science
16 Cockefair Hall
University of Missouri - Kansas City
Kansas City, MO 64110

#27

Block Multitasking of a Model, Lagrangian, Wave Code on the Denelcor HEP-1, ELXSI 6400, and CRAY XMP/48

A multitasking method is described whereby the setup and solution of the PDE's in the model, Lagrangian, wave code are uniformly distributed on the processors (or processes) available. Preliminary results of this multitasking method are presented for the Denelcor HEP-1, ELXSI 6400, and the CRAY XMP/48. In addition, some results for cache management on the ELXSI 6400 and microtasking on the CRAY XMP/48 are discussed.

D. L. Hicks
Dept. of Mathematical & Computer Sciences
Michigan Technological University
Houghton, MI 49931

M. R. Scott
Applied Computer Graphics, Division, 2644
Sandia National Laboratory
Albuquerque, NM 87185

A. H. Treadway
Applied Mathematics Division, 2646
Sandia National Laboratory
Albuquerque, NM 87185

#28

Fuzzy Clustering on a Few Processors

An important step in analyzing images is dividing the pixels into classes. It has been shown that one non-parametric, unsupervised classification scheme that works well is fuzzy clustering. The algorithm, a modification of the fuzzy c-means algorithm of Bezdek (Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, 1981), consists of iteratively assigning each pixel a grade-of-membership in each of a predefined set of candidate clusters. On each iteration the cluster centers are recomputed until convergence is reached. Subsequently, each cluster can be further subdivided using the same algorithm.

Fuzzy clustering is computationally intensive but well suited to parallel processing. The parallel algorithm depends on the type of processors being used. We describe here an approach suitable for a few processors. The algorithm is divided into two parts. In the membership phase we parallelize on pixels - each processor examines a set of pixels for membership in all clusters. In the refinement phase we parallelize on clusters to find the new cluster centers - each processor examines all pixels for a set of clusters. Measured speed-up factors on a 2-way IBM 8081 and simulations of other systems will be presented.

J. V. Dave and Alan H. Karp
IBM Scientific Center
1530 Page Mill Road
Palo Alto, CA 94304

#29

Solving Tridiagonal Linear System of Equations on the CYBER 205 Computer

A partition method for solving a single tridiagonal system of linear equations is adapted to run on vector processors such as CRAY-1 or CYBER 205. Experimental results and comparisons with the method of cyclic reduction on the CDC CYBER 205 are included. These results show that initially the processing time decreases as the number of partitions P grows, but beyond certain number of partitions the algorithm slows down. This threshold value of P is found to be proportional to the square root of the number of equations, i.e.

$$p = \alpha\sqrt{n}$$

where $1 \leqq \alpha \leqq 3$

for $100 \leqq n \leqq 10{,}000$

Swarn P. Kumar
Department of Computer Science
Colorado State University
Fort Collins, CO 80523

and

Janusz S. Kowalik
Boeing Computer Services Company
Advanced Technology Applications Division
P. O. Box 24346 M/S 7A-03
Seattle, WA 98124

#30

Parallel Programming on a General Purpose Multiprocessor

VLSI-based microprocessor technology advances at approximately twice the rate of proprietary architecture, discrete-implementation processors. Thus it is natural to build parallel microcomputer systems which further amplify the performance and cost-effectiveness of this dynamic base.

However, parallel computing is itself a new technology in which software development is the key to exploiting hardware advances. General purpose multiprocessors benefit from their ability to bring existing software into the parallel computing world. Parallel programming on a general purpose microprocessor-based multiprocessor is described, including a discussion of relevant techniques, and applications which benefit from each.

Gary N. Fielland
Sequent Computer Systems, Inc.
14360 N.W. Science Park Drive
Portland, OR 97229

#31

A Paralell Version of the Continued Fraction Integer Factoring Algorithm

This is a description of the continued fraction integer factoring algorithm which is being implemented on the massively parallel processor (MPP). This machine, designed and built by Goodyear Aerospace and installed at the Goddard Space Flight Center in Greenbelt, MD, has 16384 SIMD processors, each of which has a memory of 1024 bits.

Under masking each processor executes the same program in lock step on its particular data. The present implementation will employ the large prime variation and the early abort strategy, and, when fully operational, should be competitive with the quadratic sieve implementation presently operating at Sandia National Laboratories.

M. Wunderlich
Dept. of Mathematics
Northern Illinois University
(On leave)
3021 Red Lion Lane
Silver Spring, MD 20904
U.S.A.

H.C. Williams
Dept. of Computer Science
University of Manitoba
Winnipeg, Manitoba
CANADA R3T 2N2

#32

Parallel One-Way Dissection for the Solution of Sparse Positive Definite Systems

The method of one-way dissection lends itself to parallel implementation. Given a sparse matrix, A, whose graph is a m by l grid with m less that sqrt (l) we can form the Cholesky factorization of A in O(m sqrt(l)) time using $m^2 l^{1/2}$ processors. The algorithm consists of several well known systolic and data-flow matrix algorithms which are connected together so as to keep communication costs at a minimum. This algorithm is currently being implemented on the ZMOB at the University of Maryland.

John M. Conroy
Applied Mathematics Program
University of Maryland
College Park, MD 20740

#33
Interactive Graphics Tools for Monitoring and Evaluating Parallel Computations

Many distinct factors influence the performance of parallel algorithms and computing systems. Loadbalancing, communications latency, synchronization overhead and data buffering must be monitored and interpreted to improve the design of complex scientific parallel programs. We believe that it is important to actually visualize a parallel computation because the complexity of these factors is too great to comprehend analytically. This talk will present the experiences, modeling and implementation of a graphics system for off-line evaluation and monitoring of parallel scientific program execution. If possible, a demonstration video tape will be shown.

George Cybenko
Department of Computer Science
Tufts University
Medford, MA 02155

and

Statistics Center
MIT
Cambridge, MA 02139

Alva Couch
Department of Mathematics
Tufts University
Medford, MA 02155

#34
Development of Parallel Monte Carlo Algorithms: Some Techniques and Their Applications

Two basic static and dynamic computation assignment schemes are proposed for assigning the primary estimate computations to processors in a parallel computer. The time complexity analyses of parallel algorithms based on these schemes are carried out using order statistics, renewal and queuing theories. It is shown that for a smaller number of processors, linear speedup can be achieved with static schemes and the speedup almost equal to the number of processors can be achieved with dynamic schemes. Applications of these schemes to solving linear equations and MORSE Monte Carlo radiation transport code system will be discussed.

V. C. Bhavsar
School of Computer Science
University of New Brunswick
P.O. Box 4400
FREDERICTON, N.B. E3B 5A3
Canada

#35
On Pseudorandom Number Generators for Parallel Computers

The execution of parallel Monte Carlo Algorithms usually involves parallel operation of a large number (10^3 - 10^5) of pseudorandom numbers (PRN) generators. The requirements and problems in PRN generation in such an environment will be discussed. We show that for a set of (sub-) sequences from linear congruential PRN generator(s) pooled together may possess higher discrepancy compared to a single PRN sequence of equivalent length used in a sequential computer. The statistical properties of Cebysev mixing transformations suggest that these may be promising for PRN generation in parallel computers. Computational results of simulation studies of PRN sequences generated using linear congruential methods and Cebysev mixing transformations in parallel computers will be presented.

V.C. Bhavsar
School of Computer Science
University of New Brunswick
P.O. Box 4400
FREDERICTON, N.B. E3B 5A3
Canada

#36
A Parallel Architecture for Pattern Recognition

In statistical pattern recognition, many time-consuming computations are performed. These computations have certain characteristics which make them easy to be parallelly implemented by hardware achitectures. This paper will focus in the design of a hardware trainer achitecture which computes the statistics and generates covariance matrix. A architecture for pattern classification is also suggested. Possible VLSI architectures of the hardware trainer and classifier are suggested. The efficiency of the proposed architectures is analyzed.

Saleh Khalaf
23300 Providence Dr. A-1006
Southfield, MI 48075

Department of Electrical and Computer Engineering
Wayne State University
Detroit, MI 48202

#37

Numerical Solution of an Inverse Scattering Problem on a Supercomputer

We consider computational aspects of the following inverse scattering problem: determine the location and shape of an obstacle imbedded in a homogeneous medium from scattering data in the frequency domain. We apply an optimal control approach using the T-matrix method for calculating the scattering data. The resulting optimization problem is characterized by expensive function evaluations and small, ill-conditioned linear systems. Several specialized numerical methods are presented and their implementation on a supercomputer is discussed.

Curt Vogel
Applied Mathematical Sciences
Ames Laboratory
Iowa State University
Ames, Iowa 50011

Gerhard Kristensson
Division of Electromagnetic Theory
Royal Institute of Technology
S-100 44 Stockholm
Sweden

#38

A System-Supported Workload Balancing Scheme for Cooperating Reduction Mashines

This paper reports on the conceptual design and the emulation of a multiprocessor system which is capable of executing applicative programs by division of labor using a dynamic, system-controlled workload balancing scheme. Program construction is confined to a formal specification of the problem, whereas the program execution is entirely organized by the system. The concept is based on Church's lambda-calculus as the universal operational mechanism.
The paper elaborates how the process hierarchies that may dynamically expand and collapse during program execution can be distributed within an arrangement of processing sites.
The correctness of this scheme has been validated by means of an emulation on conventional mini-computers that includes a sophisticated process monitoring system. Some experiences with the system performance will be detailed.

Claudia Schmittgen, André Gerdts, Jochen Haumann, Werner Kluge, Michael Woitass

Gesellschaft für Mathematik und Datenverarbeitung mbH
5205 Sankt Augustin 1, Germany, 02241 142818
and
University of Bonn, 5300 Bonn, Germany

#39

Performance Tuning Tricks on a VAX Multiprocessor

A simulated multi-VAX system, with shared memory and coherent, local caches, shows near linear speedup on several numerical problems. Image processing algorithms demonstrate the sensitivity of the architecture to seemingly minor changes in code or data layout. For certain non-linear transforms, a two instruction sequence using a "normally taken" conditional branch is slower, on average, than a three instruction sequence using a conditional branch that is "normally not taken". For certain data structures, a random map from virtual addresses to physical addresses degrades performance noticeably, compared with a controlled map. Sometimes it pays to synchronize more often than algorithmic consistency demands.

Paul Clark
Digital Equipment Corporation
LTN 2-2/C05
305 Foster St.
PO Box 1450
Littleton, MA 01460

#40

Parallel Algorithms for Parabolic and Hyperbolic Partial Differential Equations

Implicit Runge-Kutta methods are used to generate time-stepping schemes for parabolic and hyperbolic equations. It is shown that for a specific class of Runge-Kutta methods, the intermediate stages decouple and can be solved in parallel, thus reducing the overall execution time to that of a low order method.

Ohannes Karakashian
Department of Mathematics
University of Tennessee, Knoxville
Knoxville, TN 37996-1300

#41

ADAPTABLE SUPERCOMPUTERS IN IMAGE PROCESSING

This work is concerned with the use of adaptable architectures in image processing applications. The best way to increase the throughput of a real-time image processing system is to introduce modularity in the architecture leading to adaptable real-time super-systems that accomodate both space and time domain expansions and support a complete dynamic redistribution of the available hardware resources. The major types of adaptable systems are the reconfigurable and the dynamic ones.

Adaptations should be performed to the solution method, to the degree of parallelism or to the physical structure of the problem under consideration. Special emphasis is given to the computational requirements imposed by standard hierarchical image processing algorithms.

N.A. Alexandridis
P.D. Tsanakas
S. Ziavras
Dept. of Electr. & Computer Engineering
Ohio University
Athens, OH 45701

#42

Hydrocodes on the HEP, Part II

The parallel algorithms for the explicit Lagrangian calculations in one spatial dimension have been extended for implicit and Eulerian simulations. A decoupling of the equations divides the zones into blocks which are integrated independently to achieve the unconditional stability of an implicit scheme. A rezoning method converts the explicit or implicit Lagrangian calculations into explicit or implicit Eulerian.

A block-by-block algorithm on the HEP-UPX operating system achieves 1% serial code and speed-up factors of 8 to 9 on a single PEM system for all four calculations: explicit and implicit, Lagrangian and Eulerian. An experience with the much greater memory latency characteristic of a multi-PEM system shows speed-up factors of 17 to 18. Work on extensions to multiphase, multimaterial, and multidimensional continuum dynamics codes is in progress.

Darrell L. Hicks
Michigan Technological University
Mathematical and Computer Sciences Department
Houghton, Michigan 49931

Joseph F. McGrath
KMS Fusion, Inc.
Target Design & Analysis
3621 South State Road
P.O. Box 1567
Ann Arbor, Michigan 48106-1567

#43

Efficient Solutions to the Parallel Prefix Computation Problem

In this paper, we discuss efficient solutions to the parallel prefix computation problem. Parallel prefix computation problem is applicable to the design of fast carry lookahead circuits, generating parallel algorithms for solutions to linear and non-linear recurrences, and in the design of very fast VLSI adders. The algorithms presented in the paper are shown to be more general and optimal than existing solutions to the problem. They can be used to generate a range of solutions having different space-time requirements. A heuristic of "end-pruning" is suggested for optimizing the solutions further.

Atul Prakash and Yih-farn Chen
Computer Science Division
573 Evans Hall
University of California
Berkeley, CA 94720

#44

A Highly Parallel Near Neighbors Algorithm of Order N Based on a Monotonic Logical Grid

This paper describes a new, efficient 3D near neighbors algorithm whose cost scales as N and which vectorizes easily using data from contiguous memory locations.

A Monotonic Logical Grid (MLG) for storing the object data is defined dynamically so that objects which are adjacent in real space automatically have close address indices in the compact MLG data arrays.

The data "swapping" required to maintain the MLG as objects pass each other in space can be partitioned and vectorized using contiguous memory locations. The algorithms will execute effectively in array processors and partitions to take advantage of asynchronous multi-procesor architectures.

by Drs. S.G. Lambrakos and J.P. Boris
Laboratory for Computational Physics
Naval Research Laboratory, Wash. DC.

#45

Parallel Processing of a Preconditioned Biconjugate-Gradient Algorithm on Cray Supercomputers

The biconjugate-gradient method (BCG) provides an attractive alternative to the standard conjugate-gradient algorithms for the solution of sparse systems of linear equations with nonsymmetric and indefinite matrix operators. A form of the BCG algorithm together with an incomplete-LU decomposition preconditioning is used which divides the major tasks (back-solving and matrix multiplies) into separate but equal-sized pieces which can be computed independently. The algorithm is used to consider matrices arising from 9-point operators using a natural ordering. Performance data for a multitasking version of the code are given for Cray supercomputers.

Alice E. Koniges
National Magnetic Fusion Energy Computer Center
Lawrence Livermore National Laboratory
P.O. Box 5509, L-561
Livermore, CA 94550

#46
Parallel Finite-Difference Migration

Migration is an essential step in the proper imaging of seismic data for oil and gas exploration. A typical migration requires on the order of 10^8 to 10^9 floating point operations. We show how to exploit two types of parallelism available to finite-difference migration for rapid generation of seismic images. In particular we have an implementation for the SAXPY 1-M parallel computer that will perform migrations in about 10 seconds. This makes interactive migration a practical tool for the working geophysical interpreter.

Stewart A. Levin
Dept. of Geophysics, Stanford University
Stanford, CA. 94305

Terry Parks
Algorithms Dept.
Guiltech Research Company
255 San Geronimo Way
Sunnyvale, CA. 94086

#47
A Failsafe Associative Memory Circuit

An electrical circuit configuration is presented that simulates a theory of neural nets. This circuit is analogous to an associative net, which means it retrieves memories based upon their content rather than their location. This associative memory circuit also adapts to failure of interconnections between components as well as to breakdown of individual components, making it failsafe. This circuit, which is self-maintained, is stable against propagation delays, jitter and noise. It will be utilized in normal, noisy, or hazardous environments where access to information is immediate and imperative.

Gary M. Josin
International Computhought Unlimited
3535 West 39th Avenue
Vancouver, British Columbia
V6N 3A4

#48
Solving the Symmetric Tridiagonal Eigenvalue Problem on the Hypercube

A parallel implementation for the computation of all eigenvalues and eigenvectors of a symmetric tridiagonal matrix on the Hypercube is presented. The Hypercube is a loosely-coupled multiprocessor system where each of the 2^m processors is directly connected to m others. The order of the matrix is assumed to exceed the number of available processors.

The recursive structure of the Hypercube is exploited in a divide-and-conquer strategy coupled with a rank-one updating method (Cuppen 1981); the rank-one modification is obtained by determining the roots of a rational function. The suitability of this method for serial as well as parallel implementation on a shared-memory architecture has recently been shown (Dongarra and Sorensen 1985). Since the processors of the Hypercube share no common memory, sorting of the eigenvalues is used to ensure an even distribution of the roots over all processors. This in turn makes possible the recognition of multiple eigenvalues as well as high processor utilisation. It is shown that, inspite of the use of the asymptotically fastest method, sorting consumes the major share of the total communication cost.

Ilse C.F. Ipsen
Elizabeth Jessup
Dept. of Computer Science
Yale University
P.O. Box 2158 Yale Station
New Haven, CT 06520

#49
Concurrent Algorithm Development for Shared Memory Environments*

At Sandia National Laboratories the Advanced Computer Science Project was initiated within the Engineering Sciences Organization to develop highly concurrent algorithms/methods for finite element and finite difference engineering analysis. Algorithm development has been performed on a ten processor ELXSI 6400 system, with limited testing on the CRAY-XMP 24 and 48. The ELXSI computer is a bus architecture machine; the synchronous bus clocks 64 bits of data every 25ns and each CPU has a clock cycle of 50ns. Floating point is IEEE standard. Memory is shared except for a 16kb cache on each CPU. We will present the performance of two specific concurrent algorithms on the ELXSI. A finite element frontal solution of a confined oscillating drop and a finite difference Lagrangian solution of 1-D shock wave propagation will be presented. We will also discuss issues related to cache and main memory management at each time step.

*This work performed at Sandia National Laboratories supported by the U.S. Department of Energy under Contract No. DE-AC04-76DP00789.

G. G. Weigand and R. E. Benner
Fluid and Thermal Sciences Department
Sandia National Laboratories
Albuquerque, NM 87185

#50
Applications of Multifrontal Methods in Conjugate Gradient Algorithms for Concurrent Finite Element Analysis*

Frontal methods are an efficient means of Gauss elimination of sparse matrix equations arising in finite element analysis. Concurrent frontal methods have been developed by means of nested mesh dissection (Benner et al. 1985) and more general multifrontal strategies (Duff 1985). We are implementing concurrent frontal ideas in computation-intensive portions of preconditioned conjugate gradient (PCG) algorithms, such as assembly of finite element equations and the construction and application of preconditioners (e.g., incomplete Cholesky or element-by-element). We present results of concurrent PCG solution of symmetric positive definite systems on a ten processor ELXSI 6400 computer. We also discuss extension of the algorithms to non-positive definite symmetric systems and unsymmetric matrix problems.

*This work performed at Sandia National Laboratories supported by the U.S. Department of Energy under Contract No. DE-AC04-76DP00789.

R. E. Benner and G. R. Montry
Fluid & Thermal Sciences Department
Sandia National Laboratories
P. O. Box 5800
Albuquerque, NM 87185

#51
Concurrent Algorithms for Convectively Dominated Fluid Flows*

Flux-Corrected Transport is a powerful method for the solution of the system of partial differential equations that arise in convectively dominated systems. A two-dimensional explicit Flux-Corrected Transport module has been implemented in a multitasking environment on a ten processor ELXSI 6400 system and a Cray XMP 24. A high level of concurrency has been achieved by restructuring the original algorithm so it automatically accommodates any number of available processing elements and "critical" path calculations are minimized. We provide the results of the new concurrent algorithm's performance for one through ten processing elements.

*This work performed at Sandia National Laboratories supported by the U.S. Department of Energy under Contract No. DE-AC04-76DP00789.

G. R. Montry, G. G. Weigand and M. R. Baer
Fluid and Thermal Sciences Department
Sandia National Laboratories
Albuquerque, NM 87185

#52
Adaptive Two-Dimensional Integration on the HEP.

An algorithm for numerical quadrature over a triangle has been implemented on the Denelcor HEP. The integration procedure is adaptive and uses an extrapolation technique to allow for various types of singular integrand behaviour. The implementation was facilitated by using the Argonne macro package. Some simple high level macros were added to obtain an efficient distribution of the workload. Because of the portable nature of the Argonne macro package our implementation should be readily portable to other MIMD machines.
The approach taken in the parallelization of this algorithm is applicable in any adaptive procedure with similar work partitioning characteristics.

Elise de Doncker and John A.Kapenga(*)
Department of Computer Science
Western Michigan University
Kalamazoo, MI 49008
(*) on leave at:
Delft University of Technology,
Electrotecnn. Div., Networking,
c/o P. Dewilde
Mekelweg 4
Delft 2600GA
The Netherlands

#53
An Algorithm for Parallel Computation of Approximate Empirical Distributions.

An implementation of an algorithm for the calculation of Approximate Empirical Distribution Functions (AEDFs) has been completed for the Denelcor HEP using the Argonne macro package.
The algorithm is iterative, related to distribution sorting, linear in time and has a good global convergence. It can also be used as a parallel sorting method.
An AEDF which is close to an Empirical Distribution Function (EDF) can be used to calculate parametric, non-parametric and distribution tests and estimates as effectively as the EDF itself.
The speed and global convergence of the algorithm allows very large data sets and real time processing.

John A. Kapenga (*)
Department of Computer Science
Western Michigan University
Kalamazoo, MI 49008
(*) on leave at:
Delft University of Technology,
Electrotechn. Div., Networking,
c/o P. Dewilde
Mekelweg 4
Delft 2600GA
The Netherlands

#54
Block Angular Parallel Least Squares Computations

We consider least squares problems where the observation matrix is assembled into the usual block angular form, such as in geodetic computations. A parallel algorithm for the solution of such least squares problems based upon a block orthogonal factorization scheme of Golub and Plemmons is presented. The algorithm exploits the block structure of the matrices in both the orthogonal factorization and back substitution phases. Some numerical experience with large-scale least squares computations using the scheme on the Denelcor HEP multiprocessor is reported.

D. Pierce
Department of Mathematics
R. J. Plemmons
Departments of Computer Science and Mathematics
Box 8205
North Carolina State University
Raleigh, North Carolina 27695-8205

A. H. Sameh
Department of Computer Science and
The Center for Advanced Computation
University of Illinois
Urbana, Illinois 61801

#55
Multi-Language Code Generation for Sequential Architectures

STO has developed a series of tools for automatically generating source code for most sequential languages, in particular Fortran, Pascal, C, and ADA. A review of programming language design faults is presented to show why languages after Fortran 66 will not be useful for software sharing and development by scientists and engineers. These faults deal with multi-dimensional arrays, complex math, functions as parameters, strong typing, and function naming. How these problems can affect software development for non-sequential architectures will be discussed, as well as efforts for a multi-lingual BLAS.

Gregory Aharonian
Source Translation & Optimization
P.O. Box 404
Belmont, Mass 02178

#56
Multi-Language Code Generation for Non-Sequential Architectures

Nine languages for nonsequential architectures will be reviewed: Multi-Fortran, Array-Fortran, Fortran 88 Concurrent Fortran/C, ADA, Actus(Cray), APL(Analogic), Cybil(CDC), OCCAM(Inmos). The review will cover unique features, core subroutine libraries(BLAS), and automatic code generation. A discussion of the social aspects of this software will follow, including software sharing, the pros and cons on having many languages, and migration of problems from the sequential world. This will include information on the programming language needs for parallel automated reasoning.

Gregory Aharonian
Source Translation & Optimization
P.O. Box 404
Belmont, Mass 02178

#57
FFT Methods for an MIMD Machine

The Denelcor HEP-1 is used as a representative shared memory MIMD computer to develop single and multiple FFT algorithms. The results of these experiments are then used to design several in-place FFT-based Poisson solvers among which the superior method is determined.

William L. Briggs
Mathematics Department
University of Colorado at Denver
Denver, Colorado 80202

#58
Writing a Specialized Map Library for a Vector Computer

The Extended Math Library (EML) for the Sperry Integrated Scientific Processor contains subroutines such as FFT's, convolutions, and interpolations. The talk describes our experiences in developing the library, particularly in using the proposed Fortran 8X array syntax, and in assembler coding efforts to make optimal use of the Sperry ISP Architecture.

Lowell Palecek
Supervisor of Extended Math Library
(EML) Programming Group
c/o Sperry Corporation
P.O. Box 64942
St. Paul, MN 55164-0942

#59

A Proof of a Vector Oriented Algorithm for Finding Eigenvalues of Real Symmetric Matrices

An algorithm for finding the eigenvalues of real symmetric matrices for vector and matrix processors has been developed by M. Clint. The algorithm iterates using similarity transformations to reduce the matrix to a diagonal matrix, much like Jacobi's method. Empirical evidence indicates that the algorithm monotonically reduces the sum of squares of the off-diagonal elements. An actual proof of the convergence of the algorithm is the topic of this paper.

William I. Thacker
Winthrop College
Computer Science
Rock Hill, S.C. 29733

L. T. Watson
Virginia Tech
Computer Science
Blacksburg, Va. 24061

#60

Finite Element Computation on the Distributed Array Processor (DAP) and the Transputer Machine.

The execution of Finite Element (FE) forms a significant part of the work load on many large scale computer installations. The FE process may be divided into three steps, which are often repeated cyclically: (i) specification of the mesh configuration, (ii) mesh-generation - the region divided into elements, and (iii) analysis-basis functions are chosen and are integrated over each element, these combinations being assembled in a single system of linear equations, and solved approximately.

In this paper we discuss the applicability of parallel computation to the operations associated with (ii) and especially (iii). We describe the programming on the DAP, and also indicate its implementation on the transputer machine - a projected architecture of up to 128 transputers.

J.J. Modi
Cambridge University Engineering Department
Trumpington Street
Cambridge
CB2 1PZ
U.K.

#61

Embedding Binary Trees into a Hypercube

Algorithms for the embedding of binary trees into a Hypercube are discussed. A Hypercube is a parallel architecture where each of the 2^n processors is directly connected to n others. "Good" embeddings of trees are important not only for the parallel execution of adaptive computations, such as quadrature or bisection, but also for mesh refinement in Finite Difference and Finite Element methods. The objective is to schedule the subproblems so that processor utilization is high and the time for information exchange among processors is kept to a minimum.

We improve on a previous method of Y. Saad for embedding a complete binary tree (cbt) of 2^n nodes into a Hypercube of 2^{2n} nodes by giving an algorithm for embedding two disjoint cbt's with 2^n nodes each into a Hypercube of 2^{n+1} processors, so that each edge in the tree corresponds to one edge in the Hypercube. The algorithm is non-recursive. We will show how to embed a cbt with 2^n-1 nodes into a Hypercube with 2^n nodes so that each edge of the tree corresponds to "short" paths in the Hypercube (most of length one). Extensions to incomplete trees and "dynamic" embeddings will be discussed.

William D. Gropp
Ilse C.F. Ipsen
Dept. of Computer Science
Yale University
P.O. Box 2158 Yale Station
New Haven, CT 06520

#62

Performance Evaluation Methods for Parallel Computing

We investigate the performance of parallel processors using an idealized analytic model, a simulation model, and experiments run on pipelined MIMD machines, e.g. the Denelcor HEP. The experiments involve the solution of an initial value ordinary differential equations problem. Performance results are first presented for a varying number of problems of identical time length invoking a range of active processes. These results are then extended to problems of statistically varying time length. The goal of our work is the development of testing methods for evaluating parallel performance on a specific class of problems.

Roger L. Crane
RCA Laboratories
Princeton, NJ 08540

Michael Minkoff and Kenneth E. Hillstrom,
Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL 60439

#63

Folding Microcode for Processing Arrays of Data

A synchronous machine whose hardware is optimized for arrays of data needs to have a microcode that is structured to reflect both the parallelism of the machine and the pipeline structure. The linear execution of usual FORTRAN code goes through a folding operation to convert it into optimal code for a parallel processor. Automation of this folding operation is considered and several examples are used to illustrate its application.

Ralph Jones
Analogic Corporation
Computing Systems Group
Audubon Road
Wakefield, MA 01880

and

Boston University
College of Engineering
110 Cummington Street
Boston, MA 02215

#64

A New Algorithm for Parallel Carry - Lookahead Addition

A fast n-bit parallel carry-lookahead adder will be introduced.This adder implements efficiently a new algorithm for addition,by using the cascade method for the evaluation of the carries.
An expandable layout as well as the complexity estimates of area and time will be given.Comparisons with other existing designs,to prove superior speed properties,will also be included.

Elena Papadopoulou
Department of Mathematics and Computer Sc.
Clarkson University
Potsdam,NY 13676

#65

THE PARALLEL SOLUTION OF TOEPLITZ SYSTEMS BY THE SYMMETRIC Q.I.F. METHOD

The symmetric quadrant interlocking factorisation (Q.I.F.) method is proposed for the parallel solution and inversion of Toeplitz systems. The method is shown to exhibit a degree of inherent parallelism which is superior to that which exists in the well known solution strategies of LU factorisation and matrix bordering. The algorithm is shown to be suitable for SIMD computers. The paper also covers briefly topics such as storage requirements, error analysis and the computational complexity of the proposed methods.

David J. Evans
Department of Computer Studies
University of Technology
Loughborough,
Leics. LE11 3TU
England.

#66

Parallel Architectures for a Family of Iterative Schemes

The design of parallel architectures for a new family of iterative schemes is the main concern of this work. The fact that the above family possess two levels of parallelism (physical and inherent) naturally leads to the design of a two-level parallel architecture : a STAR like in the first level while for the second, implementation of systolic arrays is our choice. As an example, of importance to scientific computation, we consider the parallel iterative solution of the linear system arising from the discretization of certain BVP's(Boundary Value Problems) using Hermite cubics and the collocation method.

Yiannis G. Saridakis
Departemt of Mathematics and Computer Sc.
Clarkson University
Potsdam, NY 13676

#67

The Loral DATAFLO System

This paper will discuss the architecture of the Loral DATAFLO(tm) System, a commercially viable fifth generation parallel processor. The system is designed to execute real time and computationally intensive applications. The DATAFLO System is logically composed of three parts, a UNIX(tm) software development environment, the dataflow engine and an integral I/O sub-system. The processing power of the dataflow engine can be incrementally expanded from a system with approximately 10 dataflow processors to a system with more than 100 dataflow processors. Each dataflow processor is composed of two National Semiconductor NS32000 microprocessors. The Loral DATAFLO System uses a large-grain dataflow model of computation. This model avoids the massive communication overhead present in small grain dataflow systems and allows a topology independent network of high speed busses to be used for communication.

John Van Zandt, Gale Williamson and Ian Kaplan
Loral Instrumentation
8401 Aero Drive
San Diego, CA 92123

#68
The Loral Data Graph Language

The difficulty of creating software to compile and execute programs on parallel processors continues to be one of the major challenges in the design of a parallel processor. The Loral Data Graph Language is the framework that supports the development of applications programs on the Loral DATAFLO(tm) System. The DATAFLO System uses a large grain data flow model to support the execution of a single application of a large number of parallel processors. A program for this system is composed of two parts: a description of a data flow graph in the Data Graph Language and a set of graph node implementations in a standard language, such as C. FORTRAN or Ada(tm). The Data Graph Language allows a large class of graphs to be described, including graphs that contain conditional flow (IF-THEN-ELSE) and loops (WHILE and REPEAT). Non-deterministic program behavior has been a problem on many parallel processors. By constructing a graph from subgraphs that are known to be deterministic, the Data Graph Language allows the programmer to guarantee deterministic program behavior.

Ian Kaplan
Loral Instrumentation
8401 Aero Drive
San Diego, CA 92123

#69
Data Permutations and Basic Linear Algebra Computations on Ensemble Architectures

Ensemble Architectures are interesting candidates for future high performance computing systems. The ensemble configurations discussed here are linear arrays, 2-dimensional arrays, binary trees, shuffle-exchange networks, boolean cubes and cube connected cycles. We discuss a few algorithms for arbitrary data permutations, and some particular data permutation and distribution algorithms used in standard matrix computations. Special attention is given to data routing. Distributed routing algorithms in which elements with distinct origin and distinct destinations do not traverse the same conmmunications link make possible a maximum degree of pipelined communications. The linear algebra computations discussed are: matrix transposition, matrix multiplication, dense and general banded systems solvers, linear recurrence solvers, tridiagonal system solvers, fast Poisson solvers, and very briefly, interative methods.

Lennart Johnsson
Department of Computer Science
Yale University
P.O. Box 2158 Yale Station
New Haven, CT 06520-2158

#70
Efficient Parallel Algorithms for Controllability, Root-Seperation and Cauchy Index Problems for Polynomials

Using the recently developed parallel algorithms (Brent and Luk, a systolic array for the linear-time solution of Toeplitz systems of equations, J. VLSI and comp. System (1984) 1-22 and, S.Y. Kung and Y.H. Hu, fast and parallel algorityms for solving Toeplitz Systems, Proc. Internat. Symp. on Mini and Micro-Computers in control and Measurement, San Francisco, 1981), we propose efficient parallel algorithms for controllability, stability and root-location and the Cauchy-index problems involving polynomials. The algorithms require only $O(n)$ parallel steps and $O(n)$ processors for implementation.

B.N. Datta and Karabi Datta
Department of Mathematical Sciences
Northern Illinois University
DeKalb, IL 60115

#71
Vectorization of scalar recurrences

Scalar linear recurrences appear in e.g. polynomial evaluation using Horner's rule, and in linear systems solution. In both cases evaluating n terms takes 2n operations, which is optimal. While for Horner's scheme parallel algorithms exist that, given k processors, nearly reach the optimal complexity of $2n/k$, the standard Cyclic Reduction method for general recurrences takes $5n/k$ operations. We compare this to a method based on a product expansion of the inverse of the bidiagonal matrix associated with the recurrence. Although having an asymptotic complexity of $3n(\log n)/k$, this method may be more efficient for certain values of n, if the processors are pipelined. In addition to this the method has implementational advantages.
We also consider a truncated form of the product expansion method used as preconditioning for an iterative method.

O. Axelsson and V. Eijkhout
Department of Mathematics, Catholic University
Toernooiveld, 6525 ED Nijmegen, The Netherlands

#72
Analysis of incomplete matrix factorization methods for vector and parallel computers

In recent time much effort has been devoted to the construction of robust and effective preconditioners for elliptic difference equations based on the approximate factorization of matrices partioned into blocks, where inverses of diagonal block matrices are approximated by sparse matrices.

There exists two algorithms for this, one on classical and one on division free form. The latter variant needs no matrix solves during the forward- and back-solves and is hence fully parallel on the local block level. To get it fully parallel also on the global level between blocks we study methods based on truncated Euler product expansions. For computers such as CRAYX-M 2/4, Control Data ETA and CYBERPLUS, it is important to have parallelism on the top level to get little datatraffic between the processors and to get the full gain from multitasking and vectorization. In the talk an analytic study of the above methods based on limit matrices, is presented. It is found that the new method is particularly interesting for difference equations in three-dimensions.

Owe Axelsson
Department of Mathematics
Catholic University
Toernooiveld
6525 ED Nijmegen, The Netherlands

#73

Fortran to Fortran: Automatic Conversion of a Software Library for an Attached Processor

We will describe the development of an automated procedure for conversion of a mainframe Fortran subroutine library to an attached processor. The attached processor has its own Fortran compiler but the conversion must be transparent to subroutine library users. The major obstacles in the conversion are (1) communication of data between the processors and (2) inconsistencies in the language features supported by the two compilers. A major unresolved problem is the difference in floating-point arithmetic on the two processors.

Anthony A. Billings and William F. Eddy
Department of Statistics
Carnegie-Mellon University
Pittsburgh, PA 15213

#74

An Efficient Vector Formulation of the Mixed-radix FFT Algorithm

An efficient vector formulation of the mixed-radix fast Fourier transform (FFT) algorithm is presented. The algorithm is generalized to compute multiple transforms and multi-dimensional transforms. Its applicability to compute the Winograd Fourier transform algorithm (WFTA) is also briefly discussed. Important features of the algorithm are: data is always accessed with a small stride, vector lengths are largest possible, and only standard vector operations are used. The algorithm does not require 'gather-scatter' operations and the indirect addressing is used only for the initial 'bit-reversal'.

Ramesh C. Agarwal
James W. Cooley
IBM T.J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

#75

Vector Elementary Functions

Algorithms have been developed to compute 22 vector elementary functions: SIN, COS, TAN, COTAN, LN, LOG, EXP, POWER, SQRT, ATAN, ATAN2 and |z|. These algorithms are essentially table-based algorithms. An important feature we considered was the production of identical results for both scalar and vector algorithms. Of these, five functions are perfect, i.e., the computed result is always the correctly rounded version of the infinite precision result. For the rest of the functions, the error is no more than one ULP (unit in the last place). This implies exact computed results if they are machine representable.

Ramesh C. Agarwal
James W. Cooley
Fred G. Gustavson
Bryant Tuckerman
IBM T. J. Watson Research Center
P. O. Box 218
Yorktown Heights, NY 10598

#76

New Vector Linear Algebra Algorithms for Computers with a Cache Memory

This paper describes vector algorithms for matrix addition, multiplication, and solving dense or banded linear systems for which the matrices are general or positive definite. The algorithms are related to the JKI algorithm described in "Implementing Linear Algebra Algorithms for Dense Matrices on a Vector Pipeline Machine", by Dongarra, Gustavson, and Karp, in SIAM Review, Jan. 1984. The new algorithms consider the data traffic between memory and the vector CPU.

Fred G. Gustavson
Gordon Slishman
Bryant Tuckerman
IBM T.J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

#77

Adaptive Mesh Refinement for PDEs in 3-D on Parallel Processors

Adaptive techniques for partial differential equations have become very valuable in scientific computing. However, even with the fastest of the current generation of supercomputers, large multi-dimensional problems are beyond the our capabilities. For any significant increase in computer power, parallelism must be exploited. However, most work on using parallel computers has concentrated on either small numbers of processors or very regular, non-adaptive algorithms. We discuss a fully adaptive algorithm for time dependent and time independent PDEs in three space dimensions on large numbers of processors. This algorithm is designed for message-passing parallel processors, and is designed to both minimize internode communication and maintain a reasonably well balanced load on all processors. We will discuss the complexity analysis which leads to the design. Experiments will be presented on a variety of problems and processors, including a 128-node Intel Hypercube.

William D. Gropp
Dept. of Computer Science
Yale University
P.O. Box 2158 Yale Station
New Haven, CT 06520

#78

Anomaly Reporting - a Tool for Debugging and Developing Parallel Numerical Algorithms

One approach to exploiting the parallelism of new multiprocessor supercomputers is extending existing high level languages with primitives for task creation, synchronization, and communication. Experience has shown that such primitives can make program testing and debugging very difficult due to the introduction of subtle synchronization and parallel access errors. We present a technique for static analysis and debugging of parallel programs. The technique is based upon an algorithm for computing the reachable states of parallel programs written using a set of high level concurrency primitives. A prototype implementation of the technique for extended Fortran has been developed at UCSD.

William F. Appelbe
Electrical Engineering and Computer Science
Department, C-014
University of California, San Diego
La Jolla, CA 92093

Charles E. McDowell
Computer and Information Sciences
University of California, Santa Cruz
Santa Cruz, CA 95064

#79

Equation-solving on the CRAY-2

The CRAY-2 multiprocessor features a large, relatively slow common memory and fast local memories for each processor. Partitioning procedures for exploiting local memory in full-equation-solving are considered. Performance figures are given for a block-oriented solver, with and without pivoting, for a uniprocessor. Timing projections are made for a multiprocessor solution.

D. A. Calahan
Dept. of Electrical Engineering &
Computer Science
University of Michigan
Ann Arbor, MI, 48109-1109

#80

A Comparison of Small-grain Sparse Equation Solving on the CRAY X-MP-4 and the CRAY-2

The CRAY X-MP-4 and the CRAY-2 have hardware instructions which are intended to speed up operations on sparse data formats visa-vis the CRAY-1. In this paper, the speeds of sparse equation-solving kernels on both bit-mapped and compacted data are examined for both processors. Performance of vectorized sparse solvers is presented for both machines using compacted data format and matrices from problems in electronic circuits and oil reservoir simulation. Observations will be made on MP solution.

D. A. Calahan
Dept. of Electrical Engineering &
Computer Science
University of Michigan
Ann Arbor, MI, 48109-1109

#81

Concurrent Implementation of the Crank-Nicolson Method for Heat Transfer Analysis

To exploit the significant gains in computing speed provided by Multiple Instruction Multiple Data (MIMD) computers, concurrent methods for practical problems need to be investigated and test problems implemented on actual hardware. One such problem class is heat transfer analysis which is important in many aerospace applications. This presentation compares the efficiency of two alternate implementations of heat transfer analysis on an experimental MIMD computer called the Finite Element Machine (FEM). The implicit Crank-Nicolson method is used to solve concurrently the heat transfer equations by both iterative and direct methods. Comparison of actual timing results achieved for the two methods and their significance relative to more complex problems will also be discussed.

Jonathan B. Ransom
Structures and Dynamics Division
NASA Langley Research Center
Mail Stop 244
Hampton, VA 23665

Robert E. Fulton
George Washington University
NASA Langley Research Center
Mail Stop 269
Hampton, VA 23665

#82
Complexity Results for Parallel Linear Systems Solvers

We compare the parallelization of different direct methods to solve dense linear systems of algebraic equations.
We concentrate on both SIMD and MIMD models of computation, and we give new asymptotic complexity results, as well as some numerical experiments.
More specifically, we prove that greedy type algorithms are always optimal
- for the Gauss-Jordan diagonalization method
- for the QR decomposition method using Givens rotation

Then we derive complexity results, and we show that the predicted performance for the number of steps is close to its actual value.

M. Cosnard and Y. Robert
Laboratoire TIM3 - IMAG
BP 68 - 38402 Saint Martin d'Hères Cedex
FRANCE

#83
Parallel Numerical Computations on a VLSI-Based Multiprocessor

A simplified shared-memory multiprocessor organization is proposed for VLSI implementation. Efficient implementations of basic algorithms for linear recurrence evaluation, fast fourier transform and sorting are first demonstrated on the proposed architecture. Orthogonal triangularization for least square solution of an overdetermined linear system and simplex method for linear programming problems are then mapped into proposed architecture. $O(n)$ speedup over a uniprocessor system for the above benchmarks is achieved, where n is the number of processors in the system. The proposed system can be constructed using a limited VLSI chip set and provides an economic solution to the need for high performance multiprocessor systems in large scale scientific computations.

V.K. Prasanna Kumar, P.S. Tseng and K. Hwang
Computer Research Institute
University of Southern California
Los Angeles, CA 90089

#85
Architectural Considerations in Program Design: A Comparison of Two MIMD Computers

Development of efficient programs for parallel computer systems is heavily influenced by the architectures of those systems. This influence is studied by comparing the implementation of three sample problems on two different MIMD computers, NASA's Finite Element Machine and Flexible Computer Corporation's Flex/32. The three sample problems include a parallel matrix multiplier, eigenvalue solver, and structural analysis application. The hardware/software architectures of the two systems are reviewed, and major differences in the implementations of the sample problems for the two systems are discussed. Relative performance of the two architectures is compared in terms of speedup and parallel efficiency.

Susan W. Bostic
Jonathan B. Ransom
Structures and Dynamics Division
Mail Stop 244
NASA Langley Research Center
Hampton, VA 23665

Thomas W. Crockett
PRC Systems Services
Aerospace Technologies Division
3211 N. Armistead Avenue
Hampton, VA 23666

#84
A Mesh-Connected Computer Architecture for Solving Elliptic Pde's

For solving elliptic pde's, a special-purpose SIMD two-dimensional processor array is proposed, for realization in present-day VLSI technology. The unit processor, connected to nearest four neighbors, has a reduced instruction set, floating-point adder and multiplier, and small memory. The single-mesh array supports the standard iterative multigrid solvers for elliptic problems. The necessary procedures are encoded in hardware and are called by the runtime SIMD instruction sequence. Topics presented include: design details, performance estimates, optimal speedup, average-speed slowdown due to interprocessor communication and progress in VLSI implementation.

Jayant Kirtane
Department of Computer Science
The Pennsylvania State University
University Park, PA 16802

#86

Assessment of the GAPP Systolic Chip for Use in Computationally Intensive Signal Processing

The Martin Marietta Geometric Arithmetic Parallel Processor (GAPP) consists of 72 bit serial processors rectangularly interconnected on a chip to be used as a basic building block for a SIMD architecture. The design of the chip has many features which are ideal for image processing (Sobels, spoke filters, etc.). The question of how well suited the architecture is for signal processing, in particular, cross ambiguity functions, Wigner distribution functions, multi-dimensional Fourier transforms, etc., has been investigated. The speaker will present preliminary findings from research into this question using a GAPP emulator.

Dr. Joseph N. Craig
Dr. John Sadowsky

System Engineering and Development Corporation
9150 Rumsey Road
Columbia, Maryland 21045

#87

MATRIX COMPUTATION ON THE INTEL HYPERCUBE

The Intel iPSC is a new, large scale scientific computer consisting of 32, 64 or 128 independent single board processors with a hypercube interconnection topology. The machine has a capacity of several million floating point operations per second and a total of up to 64 megabytes of main memory. In this talk, we first describe the iPSC hardware, then discuss our approach to programming LINPACK-style matrix computations. Some timing results and performance figures will be given.

Cleve Moler
Intel Scientific Computers
15201 N.W. Greenbrier Parkway
Beaverton, OR 97006

#88

Massively Parallel Communication

Most people view a mesh-connected computer such as the Massively Parallel Processor (MPP) as a tool to perform massive numbers of calculations, but fail to see that it also has the potential to perform massive numbers of communication operations. This paper presents the "sorting" operation as a communication/computational primitive for mesh connected processor arrays, which can easily be adapted to binary n-cube connected processor arrays. Sorting is presented as a three fold operation: organizational, aggregational, and distributional. For example, it can use the parallel architecture to organize data, perform histograms of data, or to perform table lookups. These three key capablities, will allow a mesh connected computer to efficiently perform lexical analysis and parsing for a compiler, which may lead to techniques necessary for the development of a large scale "inference engine", used in many AI applications, which will run on a mesh connected computer such as the MPP.

John E. Dorband
Computer Science Department
Pennsylvania State University
University Park, PA

Current mailing address:

Space Data and Computing Division/635
NASA Goddard Space Flight Center
Greenbelt, MD 20771

#89

Large-Memory CYBER 205 Simulations of Euler Flows Using a Vector Algorithm

With the proper structure given to the data by the grid transformation each coordinate direction can be differenced by the algorithm throughout the entire grid in one vector operation. Boundary conditions must be interleaved which tends to inhibit concurrency of the overall scheme, but a stategy of no data motion together with only inner-loop vectorization is a good compromise. Computed examples of transonic vortex flow separating from the leading edge of a delta wing demonstrates the processing performance of the procedure. With a grid of over one million points a rate of 125 megaflops is sustained over the 2-hour course of the computation which involves 10^{12} floating point operations.

Arthur Rizzi
Aeronautical Research Institute & Royal Institute of Technology
Box 11021
S-16111 Bromma, SWEDEN

Charles J. Purcell
ETA Systems, Inc.
1450 Energy Park Dr.
St. Paul, MN 55108 USA

#90

Parallel Algorithms for Variational Inequalities

Several parallel algorithms for the numerical solution of variational inequalities associated with symmetric positive definite matrices will be discussed. Both the overlapping and nonoverlapping block methods will be described and compared. Applications to free boundary problems, and to optimal control of systems governed by an elliptic partial differential equation will be given.

Robert E. White
Department of Mathematics
Box 8205
North Carolina State University
Raleigh, North Carolina 27695-8205

#91

Your Favorite Parallel Algorithm Might Not Be as Fast as You Think

Suppose a problem has I inputs, U outputs, and T computations. Assume transmission speed is finite and that only one copy of each input exists at the start of computation. Then the time for the problem on a parallel processing machine situated in d-dimensional space is $\geq O(\max((I+U)^{1/d}, T^{1/(d+1)}))$.

For example, Gaussian elimination on a $k x k$ matrix has $I=k^2+k$, $U=k$ and $T=O(k^3)$. Then time $\geq O(\max(k^{2/d}, k^{3/(d+1)}))$. So in 1-dimension, time $\geq O(k^2)$; in 2-dimensions, time $\geq O(k)$; and in 3-dimensions, time $\geq O(k^{2/3})$.

David C. Fisher
Department of Mathematics
Harvey Mudd College
Claremont, CA 91711

#92

Solving Large Sets of Coupled Equations Iteratively by Vector Processing

The set of coupled linear second-order differential equations which has to be solved for the quantum-mechanical description of inelastic scattering of atomic and nuclear particles can be rewritten as an equivalent set of coupled integral equations. When some type of functions is used as piecewise analytic reference solutions, the integrals that arise in this set can be evaluated analytically. The set of integral equations can be solved iteratively. For the results mentioned an inward-outward iteration scheme has been applied. A concept of vectorization of coupled-channel Fortran programs, based upon this integral method, is presented for the use on the Cyber 205 computer. It turns out that, for two heavy ion nuclear scattering test cases, this vector algorithm gives in part a speed-up of a factor 4 to 8, resulting in an overall speed-up of about a factor of 2 to 3 compared to a highly optimized scalar algorithm for a one vector pipeline computer.

Tolsma L.D.
Department of Physics
Eindhoven University of Technology
Eindhoven
The Netherlands

#93

An Expert Tool for Programming Multiprocessors

Refined languages are languages which closely resemble conventional, sequential, programming languages such as **ForTran**, except in that they permit user specification of very precise *data access* rights. If these constructs are properly used, a flow-analyzing compiler can translate the sequential programs into consistently good, highly-parallel, race-free, code for virtually any multiprocessor architecture. This paper discusses a program development environment that will enable the user to ignore parallelism without sacrificing parallel execution, but will also provide an expert tool to aid the user in refining the specification of *data access* rights in his programs so that "better" parallelism can be obtained.

Henry Dietz
CS Division, EE/CS Department
Polytechnic Institute of New York
Route 110
Farmingdale, New York 11735

David Klappholz
Center for Distributed Processing, CS Department
Stevens Institute of Technology
Castle Point Station
Hoboken, New Jersey 07030

#94

Several Parallel Algorithms for Polynomials

Algorithms for polynomials are attractive for parallel processing because of their complexity and intrinsic parallelism. There are many different types of parallel processors. Since an algorithm is processor dependent, several algorithms will be presented.

1. Pipeline algorithm. The Horner scheme involves identical, simple operators.
2. Parallel pipeline algorithm. The pipeline is cut into parallel subpipelines.
3. Parallel algorithm. Recursive application of the second algorithm yields a full parallel algorithm.

4. Task scheduling algorithm. Tasks of an algorithm are allocated to some processors. These algorithms will be evaluated on a.o.: number of tasks and processors, processing time and efficiency.

J.H.M. Andriessen
Delft University of Technology
Dept. of Mathematics and Informatics
Julianalaan 132
2628 BL Delft
The Netherlands

#95
Delft Parallel Processor 1984

The Delft Parallel Processor 1984 (DPP84) is a general-purpose discrete-time parallel processor of the MIMD-type, with 16 Processing Elements (PE). Every PE is programmed individually and contains 3 processors, a data memory, a program memory, buffered I/O-ports and an 8-bit internal bus. The interconnection of PE's is fully parallel.
A host-processor initialises the PE's, loads the parallel program and controls the processing. For process monitoring x-t-, x-y-plots and tables are available.
The DPP84 is applied for education and research (parallel algorithms, interactive simulation, solution of the interconnection problem); it will expand to a system with some thousends of PE's.

J.H.M. Andriessen
Delft University of Technology
Dept. of Mathematics and Informatics
Julianalaan 132
2628 BL Delft
The Netherlands

#96
Programming Language Equivalent to Petri Nets

A simple language which on the one hand constitutes a basis for concurrent programming languages and on the other is equivalent to a class of Petri nets is introduced. The language consists of sequential and parallel compositions, Dijkstra's guarded operations in which only (simple) synchronizing guards are allowed and an operation of waiting.
Labelled Petri nets are defined and a "regular class" of these nets is considered. The main result states that these nets are equivalent to our language for concurrent processing. The equivalency means that the sets of concurrent computations of considered instruction and net are isomorphic.

Włodzimierz KWASOWIEC
Institute of Computer Science
Polish Academy of Sciences
P.O.Box 22, PKiN
00-901 Warsaw
Poland

#97
Parallel Adaptive Numerical Schemes for Hyperbolic Systems of Conservation Laws.

Presented here is an adaptive grid selection algorithm for upwind finite-difference approximations to hyperbolic systems of conservation laws in one space dimension. This heuristic algorithm, motivated by a previous method for scalar equations, can be proved effective for linear, constant coefficient problems, and reduces to the scalar method when applied to a single equation. A parallel implementation of an adaptive finite-difference scheme based upon this grid selection algorithm has been developed for shared-memory, multiple-instruction, multiple-data (MIMD) multiprocessor computers. This implementation, which uses a machine independent macro package for the synchronization code, incorporates new algorithms for load balancing and problem partitioning when using trees as data structures. Numerical results are presented for a shock tube problem with the Euler equations run on the Denelcor HEP.

Bradley Lucier
Department of Mathematics
Purdue University
West Lafayette IN 47907

Ross Overbeek
Mathematics and Computer Science Division
Argonne National Laboratory
Argonne IL 60439

#98
Inherent Parallelism in the Determination of Prime Implicants

This paper gives a formulation for determining the number of implicants for n variables that can be mapped into an ordering of implicants for Quine-McClusky (Q-M) prime implicant generation. This ordering takes the form of sums of products of binomial coefficients from order zero up through order n. When this ordering is combined with a minimal set of interprocess dependencies between implicants, a global view of the Q-M algorithm devoid of the usual procedurally imposed constraints is provided, which will allow implementation procedures to accomodate alternative resource configurations - especially those that employ concurrent processors.

Frank David Allen PhD.
Hughes Aircraft Company
Radar Systems Group
2807 Tilden Avenue
Los Angeles, CA 90064

#99

CORDIC Implementation of Rotation-Based Matrix Algorithms

This paper discusses the implementation of rotation-based matrix algorithms with pipelined linear array of CORDIC processors. A doubly pipelined computing strategy will be applied to achieve maximum utilization of hardware and speed. A discussion of numerical properties of the CORDIC algorithm for both fix point and floating point cases will also be included. Finally, a prototype VLSI architecture will be reported.

Yu-Hen Hu
Department of Electrical Engineering
Southern Methodist University
Dallas, Texas 75275

#100

Approximating Zeros in a Message-Based Multi-Processor Environment

An important numerical problem is the approximation of a single zero of a continuous function when given an interval containing the zero. In this work, we develop asynchronous algorithms to approximate such zeros for a number of message-based multiprocessor architectures which utilize only private local memories. These algorithms are then compared using a simulator capable of modeling a wide variety of multiprocessor configurations.

Michael R. Leuze
Computer Science Department
Vanderbilt University
Box 1679, Station B
Nashville, TN 37235

Paul M. Sullins

#101

A Numerical Conformal Mapping Method for Simply-Connected Domains

We present a numerical method for approximating the conformal mapping of a simply connected region onto the unit disk. The method is based on the finite difference approximation of the Cauchy-Riemann equations in polar coordinates. We show how to generate such mapping more efficiently than previously suggested. Applications include (i) generating curvilinear boundary-fitted coordinate systems in the region, and (ii) using such coordinate systems to solve the Laplace and the Helmholtz equations on some selected regions with smooth curved boundaries. The scheme is quite suitable for multiprocessors such as the Cray X-MP/48.

Emmanuel R. Kamgnia
Center for Supercomputing
Research and Development
305 Talbot Laboratory
104 South Wright Street
Urbana, IL 61801-2932

#102

A Multiprocessor Algorithm for the Tridiagonal Eigenvalue Problem

A multiprocessor algorithm for finding selected (or all) eigenvalues and corresponding eigenvectors of a symmetric tridiagonal matrix is presented. It is a pipelined variation of EISPACK routines - BISECT and TINVIT which consists of the three steps: isolation, extraction - inverse iteration, and partial orthogonalization. Multisections are performed for isolating eigenvalues in a given interval, while bisections are performed to extract these isolated eigenvalues. After the corresponding eigenvectors have been computed by inverse iteration, the modified Gram-Schmidt method is used to orthogonalize certain groups of these vectors. In a sequential mode, this algorithm is less than 20% slower than BISECT and TINVIT. Experiments on the Alliant multiprocessor and Cray X-MP/48 will be presented.

Sy-Shin Lo
Bernard Philippe
Ahmed H. Sameh

Center for Supercomputing Research and Development
University of Illinois
Urbana, IL 61801

#104

A Parallel Algorithm for the Symmetric Tridiagonal Eigenvalue Problem

The traditional QL algorithm for the symmetric tridiagonal eigenvalue problem does neither vertorize nor is it suitable for a parallel implementation. Here we investigate an alternative to the QL algorithm based on the spectrum slicing technique. The recurrence for the bottom pivot of the LDL^T factorization of the tridiagonal matrix, which also delivers an inertia count, can be evaluated for several points in parallel (or for all entries of vector without impeding vectorization). This parallel implementation of the function evaluation is used together with a combination of bisection and secant methods related to Brent's ZEROIN algorithm, to compute all eigenvalues of the tridiagonal matrix in parallel. Some numerical tests on the Cray X-MP are presented.

John G. Lewis
Boeing Computer Services
565 Andover Park West
Tukwila, WA 98188

Bahram Nour-Omid
Center for Pure and Applied Mathematics
University of California
Berkeley, CA 94720

Horst D. Simon
Boeing Computer Services
565 Andover Park West
Tukwila, WA 98188

#105
Using the FPS-164/MAX to Solve Linear Equations: A Comparison of Dot Product and SAXPY Formulations

The Matrix Algebra Accelerator (MAX) is a hardware option for the FPS-164 Scientific Computer that allows for parallel execution on operations in matrix algebra. Operations that are appropriate for this architecture allow it to be used to solve linear equations, among other things. This work describes two implementations for factoring and solving real, nonsymmetric systems of linear equations that require partial pivoting; in the first, the MAX is used to generate parallel dot products, while in the second the MAX computes parallel SAXPY's. The size regime and configuration where each method is appropriate will be identified.

Ed Kushner, Steve Oslon
Floating Point Systems, Inc.
P. O. Box 23489
Portland, OR 97223

#107
Parallel Implementation of Multifrontal Schemes

We consider the implementation of techniques for the solution of sparse linear equations in a parallel processing environment. In particular, we look at algorithms based on a multifrontal approach (Duff and Reid, 1982, 1983 and 1984) that uses a partial ordering strategy determined by representing the factorization by an elimination tree.

We define the concept of an elimination trees and show how they can be used to determine a scheduling algorithm for the efficient use of multiple processes. This requires several algorithmic changes from the work of Duff and Reid. We discuss these changes in some detail.

We study the performance of this approach on the Denelcor HEP, the Cray X-MP, and the INTEL hypercube.

Iain S. Duff
Bldg 8.9
AERE Harwell
Didcot
Oxon OX11 ORA
England

#108
A Recursive Bisection Method for the Solution of Band Matrix Linear Systems of Equations with Parallel Processors

A recursive bisection method for the solution of sparse linear systems of equations arising from Finite Element Method will be presented. This method is simple to implement and particularly suitable for solution with parallel processors. The method will be exemplified by results from its application to solution of banded systems of equations. Analysis of such applications shows that the use of O(sn) processors requires only O(slogn) time steps, resulting in speedup of O(sn/log n) for systems with bandwidth b=2s+1.

Alex Bykat
Center for Computer Applications
University of Tennessee
Chattanooga, TN 37402

#109
A Hierarchical Condition Code Structure For Parallel Architectures

A hierarchical condition code structure, generating boolean decision criteria at the global controller level, is adaptable to a wide variety of parallel architectures. In the reconfigurable extended-word-length design of the DRAFT architecture, a two level hierarchy is implemented. At the local level status bits from the local segment are multiplexed into a single bit local status using bits in the segment level control word. At the global level the status bits for each segment are combined with a global mask to generate a global boolean condition and a segment level condition code. This condition code provides a feedback mechanism for incorporating global information into local conditions.

Donald M. Chiarulli
Walter G. Rudd
Duncan A. Buell
Louisiana State University
Department of Computer Science
Baton Rouge, LA 70803

#110

Hypercube Implementation of a Monte Carlo Rocket Plume Simulation

The direct-simulation Monte Carlo method for rarified gas analysis has been implem—nted on the Caltech Hypercube 32-node para— —roc-essor. This method simulates — —ent rarified 2-D gas flows by — —ision histories of computatio— — computational "mole— —y classes of physi— —thod is well-suite— —ion because of the la— interactions that — —y and independent-ly — domain to the Hyper-cub — —sed for a rocket-plume anal — —ance of two decompositions - a map — spatial region per processor vs. a map — a fixed set of molecules per proc-essor - is discussed.

WITHDRAWN

Joan C. Horvath
Carl S. Guernsey
Propulsion Systems Analysis Section
Jet Propulsion Laboratory, 125/224
4800 Oak Grove Dr.
Pasadena, CA 91109

#111

Stencils and Problem Partitionings: Their Influence on Parallel Architectures

Given a discretization stencil, partitioning the problem domain is an important first step for the efficient solution of partial differential equations on multiple processor systems. We derive partitions that minimize interprocessor communication when the number of processors is known a priori and each domain partition is assigned to a different processor. Our partitioning algorithm uses the stencil structure to select appropriate partition shapes. For square problem domains, we show that non-standard partitions (e.g., hexagons) are preferable to the standard square partitions for a variety of commonly used stencils. We conclude with a formalization of the relationship between partition shape and stencil structure, allowing selection of optimal partitions for a variety of parallel architectures.

Daniel A. Reed
Department of Computer Science
University of Illinois
Urbana, Illinois 61801

Loyce M. Adams
Applied Mathematics Group FS-20
University of Washington
Seattle, Washington 98195

Merrell L. Patrick
Department of Computer Science
Duke University
Durham, North Carolina 27706

#112

PARALLEL PROCESSING OPPORTUNITIES FOR STRUCTURAL FINITE ELEMENT METHODS

The design of complex structures for aerospace, transport and energy applications require continually increased levels of detail for supporting analyses. Such design activities require large order finite element models and excessive computation demands. Advances in parallel processing MIMD computers provide the opportunity for significant increases in computational speed to support large scale computing in such areas as nonlinear stress analysis, dynamics responses, vibration analysis, three dimensional elastic behavior and optimum structural design. The key is the development of parallel algorithms appropriate for structural applications. This paper provides an overview of the finite element method for solid mechanics, summarizes several major computationally bound problems, identifies some of the major computing bottlenecks, and evaluates candidate algorithms for implementation on MIMD computers. The goal of the paper, based on the author's experience over eight years with the NASA Finite Element Machine project, is to provide guidance for needed algorithm development to aid solution of structural finite element calculations on evolving MIMD computers.

Dr. Robert E. Fulton
George Washington University
and
Visiting Summer Scientist
Institute for Computer Applications to Science and Engineering
NASA Langley Research Center
Hampton, Virginia

#113

Concurrent Numerical Optimization in a Distributed Computing Environment

A large project at the University of Colorado is exploring the design and implementation of concurrent numerical algorithms in a distributed computing environment. Such an environment is well suited to running concurrent algorithms whose interprocess communication requirements are small relative to the computation costs on each computer. This talk first provides some rationale for the Colorado project. Then we illustrate our algorithmic approach with a new concurrent approach to global optimization that is well suited to a loosely coupled multiprocessing environment. Finally we discuss our experience so far in implementing numerical algorithms on a network of workstations. This also includes the implementation of a chaotic relaxation algorithm, and the design and implementation of a set of protocols, built upon Berkeley Unix 4.2, that facilitate the use of remote processes and shared or broadcast variables.

Robert B. Schnabel
Department of Computer Science
University of Colorado
Boulder, Colorado 80309

Abstracts: Contributed Papers and Poster Presentations

#114

ROPE: a Statically-scheduled Supercomputer Architecture

Supercomputer architectures are not as fast as logic technology allows because memories are slower than the CPU, conditional jumps limit the usefulness of pipelining and prefetching mechanisms, and functional-unit parallelism is limited by the speed of hardware scheduling.

We propose a supercomputer architecture called Ring of Pre-fetch Elements (ROPE) that attempts to solve the problems of memory latency and conditional jumps without hardware scheduling. ROPE consists of a pipelined CPU or very-large-instruction-word data path with a new instruction pre-fetching mechanism that supports general multiway conditional jumps.

An optimizing compiler based on a *global* code transformation technique (Percolation Scheduling or PS) gives high performance without scheduling hardware.

Kevin Karplus
and
Alexandru Nicolau
Computer Science Department
Upson Hall
Cornell University
Ithaca, NY 14853

#115

Interconnection Networks for Multiprocessor System

This paper surveys different interconnection networks for multiple processor systems employed both for SIMD and MIMD machine architecture. A comparactive study of these networks in terms of important design parameters such as bus load, interbus distance, routing and adding complexity etc is given. Furthermore a detailed survey of different analytical modes for performance evaluation of these networks are given. Parameters such as system throughput, bandwidth, average time delay, permutation capability and fault tolerance etc. are considered.
In addition various systemwise algorithms are explored and compared. Finally many interesting research problems are identified in the areas of fault-tolerance, network partitioning and dynamic reconfigurability, etc.

Arif Ghafoor
Department of Electrical & Computer Engineering
Syracuse University
Syracuse, New York 13210

#116

Automatic Compilation of Interprocessor Communication for Multiple Processor Systems

A methodology for the automatic design and implementation of deterministic interprocessor communication in real-time and mission-oriented signal processing systems is presented. The application specification, or the target architecture, can be automatically mapped or compiled unto a physical multiple processor system, or the host architecture, using a network traffic scheduler. Such a scheduler has been designed and implemented and is capable of generating optimal solutions for realistic systems. With this methodology and the associated software tool, the effort required to design and implement special purpose signal processing systems can be drastically reduced.

Ronald P. Bianchini, Jr.
John Paul Shen

Carnegie-Mellon University
Dept. of Electrical and Computer Engineering
Schenley Park, Pittsburgh PA 15213

#117

Solving Linear Recurrence Systems on Mesh-Connected Computers With Multiple Global Buses

In this paper, we present parallel algorithms for solving linear recurrence systems, which arise in many application areas. The algorithms are designed to run on mesh-connected parallel computers modified by the addition of multiple global buses. Underlying their structure is a graph representation that can be reorganized to efficiently match the target parallel architecture. Analysis of the algorithms shows that linear requirements of the parallel time recurrences can be solved using asymptotically less time when multiple global buses are added to a mesh-connected computer.

David A. Carlson
Department of Electrical & Computer Engineering
University of Massachusetts
Amherst, MA 01003

#118

Sparse Matrix Operations in an Interleaved Array Processing Architecture

This paper describes the applicability of an Interleaved Array Processor (IAP) for operating on sparse matrices. More specifically, it shows how the widely used sparse matrix data structure like the row pointer/column index scheme can be directly used in some matrix applications without the need of indirect addressing in the innermost loop. This is achieved by using programmable functional units (PFU) in the array processor which are capable of executing relatively complex operations containing several arithmetic operations and branching. It is shown that sparse matrix-vector multiplication on the PFU-based architecture is almost 3.5 times as fast as it is on the one based on pipelined ALU's.

G. Wolf and **J.R. Jump**

Department of Electrical and Computer Engineering
Rice University
Houston, Texas 77251-1892

#119

A VLSI Architecture for Matrix Multiplication

Multiplication of two nXn matrices requires O(n) time and $O(n^2)$ processors on 2-D (two-dimensional) systolic arrays. 2-D arrays are not as well-suited to fault-tolerant VLSI implementation as linear arrays. However, due to their limited connectivity, linear arrays require $O(n^2)$ time to multiply two nXn matrices. In this paper we propose a novel VLSI architecture for matrix computations that has the desirable implementation and fault-tolerance characteristics of a linear array, but has a significantly better time performance. Multiplication of two nXn matrices on our architecture requires $O(\max\{n^2/k, n^{1.5}\})$ time using $\min\{nk, n^{1.5}\}$ processors where k is the available I/O bandwidth. These algorithms are optimal in their time and processor requirements.

P.J. Varman
Department of Electrical and Computer Engineering
Rice University
Houston, TX-77001

I.V. Ramakrishnan
Department of Computer Science
University of Maryland
College Park, MD-20742

#120

RISC CPU Design for MIMDs

In SISD computers, RISC (Reduced Instruction Set Computer) CPU designs have demonstrated performance superior to that of most conventional CPU designs and with only a fraction of their circuit complexity. Unfortunately, the SISD RISC CPUs are not well matched to dynamically-scheduled MIMD supercomputers (such as RFM-MIMD or Ultracomputer), mainly due to the memory interconnection network.

This paper discusses fundamental adaptations of RISC concepts including *delayed load*, subroutine parameter-passing, and incorporation of floating-point hardware. Several new MIMD-specific RISC concepts are also formulated. All discussions involve both hardware structures and the compiler techniques needed to use them.

Henry Dietz
CS Division, EE/CS Department
Polytechnic Institute of New York
Route 110
Farmingdale, New York 11735

David Klappholz
Center for Distributed Processing, CS Department
Stevens Institute of Technology
Castle Point Station
Hoboken, New Jersey 07030

#121

Sequential Languages for Programming Highly-Parallel Computers

This paper presents a methodology which permits *any* conventional, sequential, language (ForTran, for example) to be modified so that:

1. Users can write HLL code which differs from conventional code only in that *data access* rights are more precisely specified

2. Compilers, using well-known flow-analysis techniques, can generate consistently good, highly-parallel, race-free, code for virtually any machine architecture.

The goal of this methodology is not merely to find parallelism where none was envisioned by the programmer, but to provide a more general way of expressing algorithms for parallel computers without imposing a different programming style.

Henry Dietz
CS Division, EE/CS Department
Polytechnic Institute of New York
Route 110
Farmingdale, New York 11735

Ken Stein & David Klappholz
Center for Distributed Processing, CS Department
Stevens Institute of Technology
Castle Point Station
Hoboken, New Jersey 07030

Abstracts: Contributed Papers and Poster Presentations

#122

B-HIVE Project: Present and Future

B-Hive is a multiprocessor currently under development at North Carolina State University. Based on the ALPHA interconnection structure, it achieves short interprocessor communication paths without requiring a large number of connections between nodes. Each node in the 24-node prototype will have its own communication processor, freeing the main, or application, processor for virtually uninterrupted work on an application program. Initially, the operating system will concentrate on providing rapid interprocess communication facilities. With further enhancements to be made as the need arises. Off-the-shelf compilers will be modified to support interprocessor communication and eventually, automatic detection of parallelism.

Dharma P. Agrawal
Winser E. Alexander
Edward F. Gehringer
Ravi Mehrotra

Department of Electrical and Computer Engineering
North Carolina State University
Raleigh, NC 27695-7911

Jon Mauney

Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206

#123

Performance Comparisons of Numerical Kernels Executed on Supercomputers, Mainframes, Minis, and Micros

This presentation focuses on performance variations (primarily speed differences and some numerical fluctuations) across a wide spectrum of small, medium, and large scalar, vector, and parallel machines executing program segments commonly encountered in mathematical problems. Tests include sequences of floating-point operations run in single and double precisions (in Fortran), segments from scientific and engineering applications as well as efficiency and numerical consistency checks of compiler-generated code. The results can aid in determining which kinds of calculations are appropriate for each type of computer and provide estimates of speed relationships amongst machine classes. Examples are given along with a description of the testing techniques and performance metrics used to obtain the reported results.

Myron Ginsberg
Computer Science Department
General Motors Research Laboratories
Warren, Michigan 48090-9057

#124

ERROR, CONVERGENCE AND STABILITY ANALYSIS OF MULTIRATE METHODS FOR REAL TIME SIMULATION.

The use of parallel processors is a suitable alternative for development of real time, full scope dynamic simulators. In this context, two main problems must be addressed: The partitioning of the system and the selection of an appropiate numerical scheme to solve the resulting sets of differential equations.

Multirate integration methods can be a convenient option for solving the model in muliprocessor environments, specially when two or more different dynamics are present. This paper presents the error, convergence and stability analysis for a class of multirate integration methods.

Diego Bricio Hernández*
Francisco J. Carrión
Gustavo Rodríguez Gómez

Instituto de Investigaciones Eléctricas
Departamento de Simulación
Apartado Postal 475.
MEXICO

*Universidad Autónoma Metropolitana
Unidad Ixtapalapa
Depto. de Ingeniería de Procesos e Hidráulica
Apartado Postal 55-539
09340 México, D.F.
MEXICO

#125

Evaluation of DEQSOL (Differential EQuation SOlver Language) through Its Application to Practical Problems

DEQSOL is a programming language specially designed to describe PDE problems in a way quite natural for numerical analyses. This language has two design targets. One is to enhance programming productivity by establishing a new architecture-independent language interface between numerical analysts and vector/parallel processors. The other is to automatically generate highly vectorizable FORTRAN codes from DEQSOL descriptions, thus realizing efficient execution.
Application of DEQSOL to several practical problems has shown that these targets were successfully attained. Productivity, when measured by the required source lines-of-code, has been improved almost by an order of magnitude over FORTRAN programming. Also

most of generated FORTRAN codes have marked extremely high vectorization ratios(91% - 98%) on the Hitachi S-810 vector processor.

Chisato Konno,Michiru Tsuji,Yukio Umetani
8-th Dept.,Central Research Laboratory,
Hitachi Ltd.,
Kokubunji, Tokyo 185, Japan

Hiroyuki Hirayama
Hitachi VLSI Engineering Ltd.,
Kodaira, Tokyo 187, Japan

#126

Minimizing Main Memory/Register Traffic

It is well known that accesses to main memory degrade the performance of many supercomputers due to the fact that intermediate results have to be stored and retrieved from a slower main memory. Therefore, to speed up such computations we address the question of how vector expressions can be computed so that traffic to main memory is minimized. By modelling the problem as a pebble game on graphs we show that the problem is NP-complete. We present a polynomial time approximation algorithm based on dominators to solve the problem, and empirical evidence that the algorithm performs almost optimally on graphs such as the FFT, grids, and trees, Our method also points out how it may be possible to overlap main memory accesses with computation in order to achieve further speed up. Our technique is especially applicable to machines such as the Fujitsu VP-200 where the number of vector registers is dynamically controllable; this enables us to determine how many vector registers are optimal for the expression being evaluated.

Ashfaq A. Munshi

IBM San Jose Research
5600 Cottle Road
San Jose, CA 95193

Karl M. Schimpf

Board of Studies in Computer & Information Sciences
University of California at Santa Cruz
Santa Cruz, CA 95064

#127

Conjugate gradients on optical crossbar interconnected multiprocessor

Conjugate gradients are used for solving large ill conditioned equations, e.g. in reservoir modeling and seismic inversion (McAulay, Geophys. Jan. 85). Optical spatial light modulators, (SLM), under development (McAulay, Opt. Eng. Jan. 86) permit large fast crossbar switches not otherwise economic. This permits rapid reconfiguration to match an algorithm, (McAulay, SPIE, 85). A conjugate gradient algorithm graph is modified to provide additional parallelism. The operators are mapped to elementary processors and the links are mapped to the settings of an interconnecting SLM switch. An iteration of the algorithm should take approx. 100 microsec. for an n by n matrix, n <256, and 512 processing nodes.

Dr. Alastair D. McAulay
Texas Instruments Incorporated
Central Research Laboratories
P.O. Box 226015, MS 238
Dallas, Texas 75266
(214) 995-0345

#128

A Domain Decomposed Fast Poisson Solver for multiprocessors

We present a new domain decomposed fast Poisson solver on a rectangle divided into strips. The method first performs uncoupled fast solves on each subdomain, and then the interface variables are computed by fast Fourier transform, without computing or inverting the capacitance matrix explicitly. Finally, the solution on the interior of the strips can be computed by one more fast solve on each subdomain.

This method, as opposed to others, does not involve any iteration in the solution of the system for the interface variables. It is especially suited for parallel implementation, since the independent problems in the subdomains can be solved in parallel, and the communication involves the interface variables only.

An extension of the algorithm is proposed for the case of a rectangle subdivided into boxes.

Tony F. Chan

Diana C. Resasco

Yale University - Department of Computer Science
New Haven - CT 06520

Abstracts: Contributed Papers and Poster Presentations

#129

A Multiprocessor Implementation of a Two Dimensional Vortex Method

We consider a typical grid-free method, Anderson's Method of Local Corrections, for computational fluid dynamics. Since the calculation is compute bound and cannot be statically decomposed across parallel processors, it must employ some sort of dynamic load balancing. We propose certain primitives to accomplish that as well as other operations important for parallel computations. We compare variants of these primitive operations and measure their efficacy on the Intel Personal Scientific Computer, a message passing hypercube multiprocessor with up to 128 processor nodes. The primitives are applicable to diverse problems and to diverse machines, for instance shared memory architectures, without entailing massive reprogramming.

Scott B. Baden
University of California
Computer Science Division
Berkeley, CA 94720

#130

Optimal Polynomial Acceleration of Fluid Dynamics Codes Using Krylov Subspace Methods

We have developed a general procedure to accelerate a nonlinear iterative method $u^{n+1} = M(u^n)$ with Krylov subspace methods such as GMRES. GMRES is a Krylov subspace method developed recently by Saad and Schultz (Yale University Research Report YALEU/DCS/RR-254). Given a linear system $Ax = b$ and an initial estimate of the solution x_0, GMRES finds, after k steps, an approximation x_k that minimizes the L_2 norm of the residual over the Krylov subspace $K = [r_0, Ar_0, A^2r_0, \ldots, A^{k-1}r_0]$. GMRES converges for any invertible (possibly non-symmetric) matrix. Newton's method is used and the linearized problem is solved with GMRES(k). This can be done without modifying the original nonlinear function box M if the linearized operator is multiplied by vectors using finite differences. We present results for various fluid dynamics codes including Jameson's Euler multigrid code FLO53P showing substantial acceleration at almost no increased cost. Our formulation is general and usually does not require modification of the underlying software. The code vectorizes with vector length equal to the number of unknowns in the problem. The only additional requirement is extra storage, which can be easily accomodated by devices such as the CRAY SSD.

David Paul Young
Boeing Computer Services
565 Andover Park West, MS 9C-01
Tukwila, Washington 98188

Lawrence B. Wigton and Neng J. Yu
Boeing Commercial Airplane Co.
P. O. Box 3707
Seattle, Washington 98124

#131

OCCAM, A Concurrent Systems Programming Language

OCCAM is an imperative concurrent programming language developed by INMOS to ease the design and implementation of highly concurrent systems. Although it was designed alongside the INMOS Transputer family, its semantics and production quality compilers also encourage its use in systems built with other processors. The paper introduces the language, describing rationale, semantics, implementation and use, including the development tools.

Pete Wilson
Microcomputer Applications Manager
INMOS Corporation
P. O. Box 16000
Colorado Springs, CO 80935-6000

#132

On the Convergence of the Nelder-Mead Simplex Algorithm

The Nelder-Mead simplex algorithm for function minimization is an often used algorithm that uses only function value information. Although the algorithm was first presented in 1964, no convergence results have been established. In this talk we will review the algorithm and present a convergence theorem. We will then show the limitations of the theorem and provide examples to show that the convergence results can not be extended for the present formulation of the algorithm.

J. E. Dennis Jr.
Daniel J. Woods

Department of Mathematical Sciences
Rice University
P.O. Box 1892
Houston, Texas 77251

#133

A TRUST REGION STRATEGY FOR NONLINEAR EQUALITY CONSTRAINED OPTIMIZATION

Many current algorithms for nonlinear constrained optimization problems determine a search direction by solving a quadratic programming subproblem. The global convergence properties are addressed by using a line search technique and a merit function to modify the length of the step obtained from the quadratic program.

In unconstrained optimization, trust region strategies have been very successful. We present a new approach for equality constrained optimization problems based on a trust region strategy. The direction selected is not necessarily the solution of the standard quadratic programming subproblem.

Maria Rosa Celis
John E. Dennis Jr.
Richard A. Tapia

Department of Mathematical Sciences
Rice University
P.O. Box 1892
Houston, Texas 77251

GENERAL INFORMATION

Book Exhibits

The book exhibits are in the Promenade (Third Floor), Omni Hotel. The exhibit times are 9:00 AM to 5:00 PM, Monday, Tuesday and Wednesday; 9:00 AM to 3:00 PM, Thursday.

Special Notice To:

Contributed Paper Authors and Chairmen of Contributed Paper Sessions

Chairmen of contributed paper sessions are requested to adhere to the following scheduled times:

Regular Contributed Papers

Fifteen minutes are allowed for each contributed paper. Presenters are requested to spend a maximum of twelve (12) minutes for presentation of their paper, and three (3) minutes for questions and answers.

Selected Papers

Thirty minutes are allowed for each selected paper. Presenters are requested to spend a maximum of twenty-five (25) minutes for presentation of their paper and five (5) minutes for questions and answers.

Special Notice To:

All Conference Participants

SIAM requests participants to refrain from smoking in the session rooms during lectures. Thank you.

Special Notice To:

All Authors

The SIAM Journal on Scientific and Statistical Computing (SISSC) will devote a special issue containing papers from the 1985 SIAM Conference on Parallel Processing for Scientific Computing. To be considered for the special issue, which will be published in early 1987, you must send three copies of the paper to the SIAM office by November 21, or submit them at the SIAM registration desk during the conference. After November 21, submissions will be considered for a possible second special issue of SISSC, or for publications in other appropriate SIAM journals. A copy of the current editorial policy is available at the registration desk.

UPCOMING SIAM CONFERENCES

May 14–16, 1986
Third SIAM Conference on Discrete Mathematics
Clemson University
Clemson, South Carolina

July 21–25, 1986
SIAM 1986 National Meeting
Boston Park Plaza Hotel
Boston, MA

REGISTRATION INFORMATION

The SIAM registration desk will be located on the Promenade (third floor) of the hotel and will be open as listed below:

Sunday, November 17: 5:00 pm–10:00 pm
Monday, November 18–Wednesday, November 20: 7:30 am–6:00 pm
Thursday, November 21: 7:30 am–4:00 pm

		SIAG/SC* MEMBER	SIAM MEMBER	NON-MEMBER	FULL-TIME STUDENT
Registration Fees	Advance	$ 80	$ 85	$110	$15
	On Site	$120	$125	$150	$15

Wine and Cheese Party
Monday, November 18, 6:00 pm
$10.00

*Members of the SIAM Activity Group on Supercomputing (SIAG/SC) are entitled to a $5 discount from the SIAM member fee. SIAM members are encouraged to join the SIAG/SC. Please contact the SIAM Customer Services Department for details.

Non-member registrants are encouraged to join SIAM in order to obtain the member rate for conference registration and enjoy all the other benefits of SIAM membership.

Special Note:
There will be no prorated fees. There will be no refunds after the conference starts. SIAM does not accept credit cards.

Telephone Messages:
The telephone number at the Omni is 1-804-622-6664. They will either connect you to the SIAM desk or forward a message.

PUBLISH WITH SIAM

SIAM publishes eight academic journals and the bimonthly SIAM News as well as a variety of books in such series as SIAM Studies and Frontiers in Applied Mathematics.

We welcome the opportunity to discuss new writing projects of all kinds with SIAM members and other potential authors.

SIAM
117 South 17th Street
14th Floor
Philadelphia, PA 19103-5052
U. S. A.

SIAM Meetings & Conferences

November 18-21, 1985

Second SIAM Conference on Parallel Processing for Scientific Computing
Omni Hotel, Norfolk, VA

The first day of the conference will be a full-day tutorial on parallel processing. It will focus on the influence of computer architecture and give a review of parallel numerical algorithms. The main conference program will address experience with operational parallel machines and developments in numerical algorithms. *Program chairman:* Robert G. Voigt, ICASE-NASA Langley Research Center.

April 14-18, 1986

Joint IMA/SIAM Conference on the State of the Art in Numerical Analysis
University of Birmingham, United Kingdom

May 14-16, 1986

Third SIAM Conference on the Applications of Discrete Mathematics
Clemson University, Clemson, SC

A conference to encourage interaction between the developers and users of discrete mathematics. Topics will include: discrete optimization; graph theory; discrete mathematics in biological, computational, engineering, medical, physical, and social sciences; cryptography; operations research; algebraic network reliability; domination numbers; and partially ordered sets. *Program co-chairmen:* Richard D. Ringeisen (Clemson University) and Fred S. Roberts (Rutgers University).
Abstract deadline: December 9, 1985

June 2-4, 1986

SIAM Workshop on Multiphase Flow
Xerox Training Center, Leesburg, VA

Workshop Chairman: George C. Papanicolaou, Courant Institute of the Mathematical Sciences.

July 21-25, 1986

SIAM 1986 National Meeting
Boston Park Plaza Hotel, Boston, MA

Plus a one-day short course on Scientific Computing on Vector Machines (July 20)

Short Course Organizer: Francis Sullivan, National Bureau of Standards.

Topics include: parallel computation; multi-grid methods; compressible flow calculations; surface approximation and computer-aided design; bubbly fluids or fluidized beds; geophysical modeling; computer tomography; crystal growth in materials; porous media; and iterated mappings and chaos. *Program chairman:* Robert O'Malley, Rensselaer Polytechnic Institute.
Abstract deadling: February 21, 1986

August 11-14, 1986

SIAM Conference on Linear Algebra in Signals, Systems, and Control and a Short Course on Image Processing
57 Park Plaza Hotel, Boston, MA

Applications of linear algebra to linear and nonlinear problems in the following areas: signal processing such as those arising in speech and vision; large-scale systems; robust, stochastic, and adaptive control; statistical estimation and filtering; and geometric theory of multivariable control. Theoretical foundations, computational algorithms, symbolic manipulation, and expert software will be emphasized.
Abstract deadline: March 19, 1986

Program Chairman: Biswa N. Datta, Northern Illinois University, Dekalb IL.

November 17-20, 1986

SIAM/NSB Short Course on Scientific Software for Supercomputing National Bureau of Standards
Washington, DC

Short Course Organizer: Francis Sullivan, National Bureau of Standards.

PAPERS AND POSTERS

Contributed papers and poster presentations at the SIAM national and fall meetings may treat any aspect of applied mathematics, although contributions consistent with the themes of the plenary symposia are especially desired. At SIAM conferences, papers and poster presentations must be related to conference topics.

MINISYMPOSIA

Proposals for minisymposia are invited for the national and fall meetings. Minisymposia that focus on a topic consistent with the themes of the plenary symposia are especially desires.

Organizers of minisymposia must be SIAM members.

STANDARD FORM

Abstracts of contributed papers and poster presentations, as well as proposals for minisymposia, must be submitted on standard SIAM forms. These may be obtained by contacting SIAM Conference Coordinator, Suite 1400, Architects Building, 117 South 17th Street, Philadelphia, PA 19103-5052, (215) 564-2929.

REPLY FORM

Detach and mail to: SIAM Conference Coordinator, Suite 1400, Architects Building, 117 S. 17th Street, Philadelphia, PA 19103-5052, or phone (215) 564-2929.

Please send me information on the following SIAM meetings and conferences:

CONFERENCE/MEETING	PLEASE CHECK
Second SIAM Conference on Parallel Processing for Scientific Computing November 18-21, 1985 Omni Hotel, Norfolk, VA	❑ Program/Registration material
Joint IMA/SIAM Conference on the State of the Art in Numerical Analysis April 14-18, 1986 University of Birmingham, UK	❑ Program/Registration material
Third SIAM Conference on the Application of Discrete Mathematics May 14-16, 1986 Clemson University, Clemson, SC	❑ Paper/poster abstract form ❑ Minisymposia proposal form ❑ Program/Registration material
SIAM Workshop on Multiphase Flow June 2-4, 1986 Xerox Training Center Leesburg, VA	❑ Program/Registration material
SIAM 1986 National Meeting July 21-25, 1986 Boston Park Plaza Hotel, Boston, MA	❑ Paper/poster abstract form ❑ Minisymposia proposal form ❑ Program/Registration material
SIAM Conference on Linear Algebra in Signals, Systems, and Control and a Short Course on Image Processing August 11-14, 1986 57 Park Plaza Hotel, Boston, MA	❑ Paper/poster abstract form ❑ Minisymposia proposal form ❑ Program/Registration material
SIAM/NBS Workshop on Scientific Software for Supercomputing November 17-20, 1986 National Bureau of Standards Washington, DC	❑ Program/Registration material

❑ I am a SIAM member. ❑ Please send me information on SIAM membership.

1185

NAME

TITLE DEPARTMENT

ORGANIZATION AREA CODE/PHONE NUMBER

ADDRESS

CITY STATE ZIP CODE

SIAM

117 South 17th Street 14th Floor Philadelphia, PA 19103-5052 U. S. A.