SYMPOSIUM ON COMPLEXITY OF SEQUENTIAL AND PARALLEL NUMERICAL ALGORITHMS

PROGRAM AND ABSTRACTS

Department of Computer Science
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

May 16-18, 1973

The Symposium is sponsored by the Office of Naval Research under Contract N00014-67-A-0314-0010, NR 044-422.

The Symposium on Complexity of Sequential and Parallel Algorithms was organized to provide a forum for the presentation and discussion of recent results and current work on topics such as sequential and parallel algorithms, algebraic and analytic computational complexity, the influence of machine organization on algorithms, and the influence of specific problems on machine organization.

The program of the Symposium appears here, together with the abstracts of contributed papers, alphabetically by author. Complete texts of invited papers will appear in the conference Proceedings to be published by Academic Press in Fall 1973.

SYMPOSIUM ON COMPLEXITY OF SEQUENTIAL
AND PARALLEL NUMERICAL ALGORITHMS

May 16 - 18, 1973

Invited paper presentations will be 50 minutes in length with 10 minutes for questions. Contributed paper presentations will be 15 minutes in length with 5 minutes for questions. To permit participants to come to specific talks we will adhere strictly to the published schedule.

PROGRAM

Tuesday, May 15

7:00 PM - 10:00 PM Cash Bar at Mudge House

Wednesday, May 16

Invited Paper Session - Room 7500 Science Hall

Session Chairman - D. Young, University of Texas at Austin

9:00 AM - 10:00 AM H. S. Stone, Stanford University
"Problems of Parallel Computation"

10:00 AM - 11:00 AM D. Kuck, University of Illinois
"Parallelism in Ordinary Programs"

11:00 AM - 11:30 AM Coffee

11:30 AM - 12:30 AM J. F. Traub, Carnegie-Mellon University
"Iterative Solution of Tridiagonal Systems on Parallel or Vector Computers"

12:30 PM - 1:30 PM Lunch at Skibo Cafeteria or Snack Bar

Contributed Paper Sessions

Room 7316

Session Chairman - J. Savage, Brown University

1:30 J. van Leeuwen
University of Calif., Berkeley
"On the Efficient Validation of Execution Sequences of Simple Recursive and Parallel Program Schemata"

Room 7500

Session Chairman - B. Chartres, University of Virginia

H. T. Kung and J. F. Traub
Carnegie-Mellon University
"On the Efficiency of Parallel Iterative Algorithms for Non-Linear Equations"

Wednesday, May 16 (continued)

Contributed Paper Sessions (continued)

	Room 7316	Room 7500
1:50	S. Schindler Technische Universität Berlin "The Complexity of Scheduling Algorithms for Multiprocessor Systems"	T. A. Straeter NASA/LRC "A Parallel Variable Metric Optimization Algorithm"
2:10	G. E. Collins Stanford University "Efficient Quantifier Elimination for Elementary Algebra"	M. W. Green Stanford Research Institute "A Highly Parallel Algorithm Algorithm for Solving Polynomial Equations"
2:30	J. Perl Technische Universität Berlin "Problem-Classes and Characteristic Algorithms"	D. Stevenson and J. F. Traub Carnegie-Mellon University "Iterative Solution of Block Tri-Diagonal Systems on Parallel or Vector Computers"
2:50	F. G. Carty Goodyear Aerospace Corporation "Gauss Elimination on the STARAN"	D. Heller Carnegie-Mellon University "A Determinant Theorem with Applications to Parallel Algorithms"
3:10		W. Miller IBM T. J. Watson Research Center "The Complexity of Roundoff Analysis"

Invited Paper Session - Room 7500 Science Hall

Session Chairman - R. Gregory, University of Texas at Austin

3:30 PM - 4:30 PM	R. Brent, Australian National University "The Parallel Evaluation of Arithmetic Expressions in Logarithmic Time"
4:30 PM - 5:30 PM	W. M. Gentleman, University of Waterloo "The Relevance of Various Cost Models of Complexity"

6:00 PM - 7:00 PM	Cash Bar in Faculty Lounge, Skibo
7:00 PM - 8:00 PM	Dinner in Faculty Dining Room, Skibo
8:00 PM -	Cash Bar in Faculty Lounge, Skibo

Thursday, May 17

Invited Paper Session - Room 7500 Science Hall

Session Chairman - M. Shaw, Carnegie-Mellon University

9:00 AM - 10:00 AM	W. P. Jones, NASA Ames Research Center "Data Mapping for Solving Poisson's Equations with Fast Transformations on Illiac IV"
10:00 AM - 11:00 AM	J. L. Owens, Lawrence Livermore Laboratory "The Influence of Machine Organization on Algorithms"
11:00 AM - 11:30 AM	Coffee
11:30 AM - 12:30 AM	D. R. Reddy, Carnegie-Mellon University "Some Numerical Problems in Artificial Intelligence: complexity and implications for multiprocessor architecture"
12:30 PM - 1:30 PM	Lunch at Skibo Cafeteria or Snack Bar

Contributed Paper Sessions

	Room 7316	Room 7500
	Session Chairman - T. Pavlidis, Princeton University	Session Chairman - L. Hageman, Westinghouse-BAPL
1:30	G. Pitts Central Texas College "A New Differencing Algorithm for Solving Partial Differential Equations"	D. J. Rose and R. Bank Harvard University "An $O(n^2)$ Method for Solving Constant Coefficient Boundary Value Problems in Two Dimensions"
1:50	M. T. McClellan University of Maryland "Computing Times for the Integral Solution of Linear Equations"	D. A. Calahan University of Michigan "Parallel Solution of Sparse Simultaneous Equations"
2:10	B. Brosowski Universität Göttingen "Best Approximation of Functions by Computational Schemes with a Given Number of Operations"	A. T. Berztiss University of Pittsburgh "Expected Densities in Operations on Sparse Matrices"

Thursday, May 17 (continued)

Invited Paper Session - Room 7500 Science Hall

Session Chairman - G. Collins, Stanford University

2:30 PM - 3:30 PM	A. Borodin, University of Toronto "On the Number of Arithmetics Required to Compute Certain Functions - Circa May 1973"
3:30 PM - 4:00 PM	Coffee
4:00 PM - 5:00 PM	S. Winograd, IBM Research Center "Some Remarks on Fast Multiplication of Polynomials"
5:00 PM - 7:00 PM	Happy Hour in Computer Science Lounge, Science Hall 4219, and Tour of Computer Science Department Research Facilities
7:00 PM - 8:00 PM	Dinner in Faculty Dining Room, Skibo

Friday, May 18

Invited Paper Session - Room 7500 Science Hall

Session Chairman - G. W. Stewart, Carnegie-Mellon University

9:00 AM - 10:00 AM	J. R. Bunch, Cornell University "Complexity of Sparse Elimination"
10:00 AM - 11:00 AM	G. Birkhoff, Harvard University "Elimination by Nested Dissection"
11:00 AM - 11:30 AM	Coffee

Contributed Paper Sessions

	Room 7316	Room 7500
	Session Chairman - F. Fritsch, Lawrence Livermore Laboratory	Session Chairman - E. M. Reingold, University of Illinois
11:30	E. I. Field and H. Stralberg Universal Analytics, Inc. "Solution of Large Sets of Linear Equations on Illiac IV"	D. Dobkin Harvard University "Lower Bounds on Matrix Multiplication"
11:50	B. P. Shay Naval Res. Lab. and Univ. of Md. "A Microprogrammed Implementation of Parallel Program Schemata"	J. F. Savage Brown University "Matrix Multiplication and Polynomial Evaluation with Known Matrices and Polynomials"

Friday, May 18 (continued)
Contributed

Contributed Paper Sessions (continued)

Room 7316

12:10 T. G. Rauscher and B. P. Shay
Naval Research Lab.
"The Influence of Computation Schemata Representations of Signal Processing Algorithms on the Architecture of the AN/UYK-17 Computer"

Room 7500

Z. M. Kedem
Columbia University
"On the Number of Multiplications Required to Compute Certain Functions"

12:30 PM - 1:30 PM Lunch at Skibo Cafeteria or Snack Bar

Contributed Paper Sessions

Room 7316

Session Chairman - F. Fritsch, Lawrence Livermore Laboratory

1:30 L. Lamport
Massachusetts Computer Assoc.
"The Parallel Execution of FORTRAN DO Loops"

1:50 T. Pavlidis
Princeton University
"Efficient Implementation of Functional Approximation Algorithms for Picture Processing"

2:10 G. W. Cobb
Texas Instruments Incorporated
"What a Vector Computer Can Do for a Meteorological Model"

Room 7500

Session Chairman - E. M. Reingold, University of Illinois

1:30 R. L. Probert
University of Waterloo
"On the Complexity of Algorithms for Symmetric Computations"

1:50 M. Shaw and J. F. Traub
Carnegie-Mellon University
"Analysis of a Family of Algorithms for the Evaluation of a Polynomial and Its Derivatives"

2:10 E. C. Horvath and J. E. Ullman
Princeton University
"A Stable Sorting Algorithm Utilizing Variable Extra Space"

Friday, Hay 18 (continued)

Invited Paper Session - Room 7500 Science Hall

Session Chairman - D. Rose, Harvard University

2:30 PM - 3:30 PM	M. H. Schultz, Yale University "The Complexity of Partial Differential Equations"
3:30 PM - 4:30 PM	A. Schtfnhage, UniversitHt TUbingen "Fast Schmidt Orthogonalization and Unitary Transformations of Large Matrices"

BEST APPROXIMATION OF FUNCTIONS BY COMPUTATIONAL SCHEMES WITH A GIVEN NUMBER OF OPERATIONS

B. Brosowski and J. SpieB

UniversitSt GSttingen and GWD G6ttingen

We consider best approximation of functions by the set $V_{n,k,j}$. of expressions computable with n additions/subtractions, k multiplications and j divisions.
For some values of n, k and j the sets $V_{n,k,j}$, are explicitely determined and some of the problems connected with the best approximation by elements of $V_{n,K,j}$ are discussed. We show that in the cases connsidered best approximation exists but in general is not unique. Further we show that the sets ^ j [illegible] Kolmogorov-sets especially they are not convex.

Expected Densities in Operations on Sparse Matrices

A. T. Berztiss,
Department of Computer Science, University of Pittsburgh.

The ability to predict the density of the result of a matrix operation is of practical importance in the design of efficient storage representations for sparse matrices in applications packages for the manipulation of sparse matrices. Also, such results can be used to design the proper strategy for selecting the order in which elements of large sparse matrices are to be looked at when operations are performed on the matrices.

Given two sparse matrices, an n x k matrix A of density a and a k x m matrix B of density b, the exact expected value of the density d of their product is derived under the assumption that the distribution of nonzero elements in A and B is random. An approximation $d \simeq 1 - (1 - ab)^k$ has been found to be adequate for most practical purposes. The approximation tends to the exact value as k increases, and both values then tend to unity. Work is in progress on the effect deviations from randomness have on the expected density of the product, and on the expected density of the inverse of a matrix.

ABSTRACT

"Parallel Solution of Sparse Simultaneous Equations"

D. A. Calahan
Department of Electrical and Computer Engineering
The University of Michigan
Ann Arbor, Michigan 48104

The ordering of sparse equations has been used in the past to reduce computation and storage requirements. In this paper, ordering is examined as a mechanism to permit simultaneous (parallel) operations on many rows and columns. This allows a block partitioning of the matrix, where each block can be considered a single entity for the purposes of the division and multiplication operations involved in the solution process.

To accomplish this ordering, a combinatorial algorithm that has algebraic growth with N(the matrix size) or a faster heuristic procedure may be used. The latter is adopted, and applied to a number of randomly-generated examples. This procedure is also applied to a sparse matrix arising from a mechanical engineering simulation problem. In all cases it is shown that the fill produced by parallelization is not excessive.

The pros and cons of the method are considered in view of contemporary machine architectures.

F. G. Carty
Goodyear Aerospace Corporation
1210 Massillon Road
Akron, Ohio 44315

216 - 794-2574

GAUSS ELIMINATION ON THE STARAN

ABSTRACT

Techniques are described which can be used to implement Gauss elimination and Gauss-Jordan reduction for n simultaneous linear equations on a parallel processor such as the Goodyear Aerospace STARAN. One technique reduces the number of arithmetic operations from $(4n^3+9n^2-7n)16$ needed by Gauss elimination on a sequential computer to as few as 3n parallel arithmetic operations on a parallel processor. The reduction in the number of arithmetic operations increases the significance of data movement operations which will also be discussed. A preliminary comparison with parallel Gauss elimination such as suggested by Katz in [1] shows some new techniques to be faster for moderate n.

[1] L. C. Hobbs et al, Parallel Processor Systems, Technologies and Applications, Spartan, New York, 1970.

TITLE: WHAT A VECTOR COMPUTER CAN DO FOR A METEOROLOGICAL MODEL

AUTHOR: DR. GARY W. COBB

AFFILIATION: TEXAS INSTRUMENTS INCORPORATED

ABSTRACT

In this paper, three meteorological models are analyzed from the point of view of coding techniques for the Advanced Scientific Computer (ASC). A textbook example of a BUSHBY - WHITELAM baroclinic model was coded and analyzed for its' vector organization. Results show that a one-clock-per-arithmetic-operation rate can be achieved on a one-pipe ASC on over 90 percent of the model code. A barotropic primitive equation model was coded directly from the difference equations exploiting the ASC Fortran subarray statement. A comparison with an equivalent Fortran IV coding shows the power of the ASC optimizing Fortran compiler. Recoding of a very pointwise algorithm for calculating the adiabatic adjustments from a sophisticated global baroclinic model has been done to yield a more parallel code. Comparisons are made between these two algorithms from a coding standpoint.

EFFICIENT QUANTIFIER ELIMINATION FOR ELEMENTARY ALGEBRA*

George E. Collins
Stanford University and University of Wisconsin

A <u>cylindrical algebraic decomposition</u> (c.a.d.) of R (the reals) is a connected partition $S=(S_1,S_2,\ldots,S_{2k+1})$ of R where k=0 and $S_1=R$ or, for k real algebraic numbers $a_1< a_2< \ldots< a_k$, $S_{2i}=\{a_i\}$, $S_{2i+1}=(a_1,a_{i+1})$, an open interval, $S_1=(-\infty,a_1)$ and $S_{2k+1}=(a_k,\infty)$. A c.a.d. $S=(S_1,\ldots,S_{2k+1})$ of the Cartesian product R^{r-1}, together with continuous real-valued algebraic functions $f_{i,1}< f_{i,2}< \ldots< f_{i,\nu_i}$ on S_i, $1\leq i\leq 2k+1$, determines a c.a.d. $(S_{1,1},\ldots,S_{1,2\nu_1+1},S_{2k+1,1},\ldots,S_{2k+1,2\nu_k+1})$ of R^r, and $(S_{i,1},\ldots,S_{i,2\nu_i+1})$ is a connected partition of the cylinder $S_i \times R$. $b=(b_1,\ldots, b_{2k+1})$ is an <u>algebraic sample</u> of the c.a.d. $S=(S_1,\ldots,S_{2k+1})$ in case $b_i \in S_i$ and the coordinates b_i are algebraic for all i. If $\mathcal{A}$ is a set of real polynomials in r variables, the c.a.d. S of R^r is $\mathcal{A}$-<u>invariant</u> in case each $A \in \mathcal{A}$ is invariant in sign in each cell of S.

An algorithm D is described which, given a finite set $\mathcal{A}$ of integral polynomials in r variables, computes an $\mathcal{A}$-invariant c.a.d. S of R^r, an algebraic sample b of S and for each cell S_i of S a formula ψ_i of elementary algebra which defines S_i.

From D we obtain a quantifier elimination algorithm E. Given a formula $\phi(x_1,\ldots,x_r)$, E applies D to the set of polynomials occurring in ϕ and evaluates ϕ at all sample points. Let t(r,m,n,d) be the maximum computing time of E for formulas ϕ in r variables containing at most m polynomials, with degrees bounded by n, coefficient lengths by d. For each fixed r, t(r,m,n,d) is dominated by a polynomial $P_r(m,n,d)$.

*Research supported by the National Science Foundation (GJ-30125X), the Wisconsin Alumni Research Foundation, and (in part) by the Advanced Research Projects Agency of the Office of the Secretary of Defense (SD-183).

Lower Bounds on Matrix Multiplication : Abstract

David Dobkin
Division of Engineering and Applied Physics
Harvard University
Cambridge, Massachusetts

An algebraic structure is presented through which complexity problems concerning polynomial multiplication and matrix multiplication can be studied. Within this structure, elementary methods are presented for comparing the relative difficulties of such problems. This format explains the gap between achievable upper bounds and predicted lower bounds for the complexity of matrix multiplication. Further study of these assumptions yields insight into methods of extending present lower bounds on matrix multiplication. Using this formulation, it can be shown that multiplying nxn matrices is at least half as hard as multiplying an arbitrary polynomial of degree n-1 by a set of n polynomials of the same degree. Methods of relating matrix multiplication computations to computations of inner products of n-tuples of polynomials are also studied.

Supported by the U.S. Office of Naval Research under the Joint Services Electronics Program by Contract N00014-67-A-0298-0006.

SOLUTION OF LARGE SETS OF LINEAR EQUATIONS ON ILLIAC IV

Dr. Eric I. Field and Mr. Halstein Stralberg

Universal Analytics, Inc.

The solution of a large set of linear equations, characterized by a sparse, symmetric matrix of coefficients, is frequently required in many application areas such as structural mechanics, heat transfer, and circuit analysis. Efficient solutions of such equation systems require a technique that takes advantage of the sparseness of the coefficient matrix.

A method is demonstrated to solve such equation systems efficiently on the ILLIAC IV parallel processor. ILLIAC IV consists of 64 processing elements, each with an associated memory of 2048 64-bit words. These processing elements, which are supervised by a common control unit, are able to operate in parallel. That is, they each perform the same operations but on different data.

The method takes advantage of the sparseness of the coefficient matrix by using a combination of the bandwidth and wavefront technique. Rows and columns of the matrix are assumed to be organized such that the bulk of the nonzero coefficients lie within a band around the diagonal, while a limited number of "special columns" with nonzero coefficients are allowed outside that band.

Greatest efficiency is achieved when the bandwidth as well as the number of "special columns" active in any one row, are both multiples of 64. In applying the algorithm, the bandwidth is chosen so as to include the dense portion of the matrix around the diagonal, while scattered nonzero coefficients outside the dense portion are included in "special columns." If there are no "special columns," the algorithm reduces to the standard bandwidth method. If all nonzero coefficients are represented in "special columns," the algorithm reduces to the wavefront approach to equation solving. Efficient utilization of the ILLIAC IV parallel processing capability is achieved by storing the coefficient matrix such that all entries belonging to a given matrix column are stored in the same processing element.

For a problem with 4,000 unknowns, bandwidth of 320, and the number of active "special columns" limited to 64 in any one row, the solution time is estimated to be 20 seconds.

A HIGHLY PARALLEL ALGORITHM FOR SOLVING POLYNOMIAL EQUATIONS

by

Milton W. Green
Stanford Research Institute
Menlo Park, California

Abstract

Let $P(z) = a_n z^n + \ldots a_0$ be an arbitrary polynomial of degree n where the coefficients a_i may be real or complex, and the roots $\lambda_1^* \ldots \lambda_n^*$ need not be distinct.

We have investigated a zero-finding algorithm that consists of successive relaxation of the system of assignments

$$\lambda_i \leftarrow \lambda_i - P(\lambda_i) / \prod_{i \neq j} (\lambda_i - \lambda_j), \qquad (i,j = 1,2 \ldots n)$$

where the initial values of λ_i are chosen to be unequal and not all real but are otherwise quite arbitrary. Although the above transformation was known to Weierstrass in 1903, its remarkable global convergence properties seem to have gone unnoticed. What is actually observed (in hundreds of trial cases) is an initially rather chaotic behavior of the values of λ_i approximating the zeros of P(z), followed by a final phase in which the convergence toward all isolated roots is quadratic. Multiple zeros cause no difficulties, however convergence in the neighborhood of such roots is linear (a la Newton's method) and limited by machine precision in the expected way.

Since all zeros of P(z) are found "simultaneously" by this method, any available number of processors may be used either synchronously or asynchronously to compute individual approximants in the system of assignments that determine the new λ_i. Other opportunities for parallelism exist in the computation of the polynomial $P(\lambda_i)$ and the product $\pi(\lambda_i - \lambda_j)$. Furthermore, convergence is not prevented by occasional use of old values rather than new ones in the computation of any given λ_i, allowing memory-fetch conflicts to be ignored.

The above considerations lead to a very simple task scheduling method that can employ any number of processors ≥ 1 advantageously in solving a polynomial. When the number of available processors is substantially larger than the degree n the time T_n to find all of the roots of P(z) can be reduced to o(n).

A DETERMINANT THEOREM WITH
APPLICATIONS TO
PARALLEL ALGORITHMS

Don Heller
Dept. of Computer Science
Carnegie-Mellon University

We state and prove an expansion theorem for the determinant of any Hessenberg matrix. The expansion is expressed as a vector-matrix-vector product which can be efficiently evaluated on a parallel machine. We consider the computation of the first N terms of the sequence defined by the general linear recurrence

y(i) = Z]lo ^^^) y(3) + H(i), i * o.

On a sequential machine this problem is 0(N), with N processors it is 0(N), and with 0(1^) processors it is 0(log N) using our expansion. Other applications include locating roots of analytic functions and proving doubling formulas for linear recurrences with constant coefficients.

This work was supported in part by the National Science Foundation under Grant GJ-3211 and by the Office of Naval Research under Contract N00014-67-A-0314-0010, NR044-422.

A STABLE SORTING ALGORITHM

UTILIZING VARIABLE EXTRA SPACE

E.C. Horvath and J.D. Ullman
(to be presented by Horvath)

Princeton University

ABSTRACT

A stable sorting algorithm is defined as one which does not permute the order of records bearing eaual keys. Some examples of stable sorts are the List Merge Sort and the Bubble Sort, which require $O(n \log n)$ and $O(n^2)$ time, respectively, and $O(n)$ and $O(1)$ extra space, respectively, to sort n records.

In this paper we propose a stable sorting algorithm which uses extra space $S(n)$, $2 \le S(n) \le n$, and requires $O(n^2 \log n/\log S(n))$ time in the worst case.

ON THE NUMBER OF MULTIPLICATIONS REQUIRED

TO COMPUTE CERTAIN FUNCTIONS

Zvi M. Kedem
Dept. of Mathematical Statistics
Columbia University
New York, N.Y. 10027

ABSTRACT

Let F be an infinite field and let $x, a_1, \ldots, a_m$ be indeterminates. Let Φ and φ be a matrix and a vector of elements in $F(x)$. A method to study the number of multiplications and divisions required to compute $\Phi\underline{a} + \varphi$ ($\underline{a}$ si the column vector $(a_1, \ldots, a_m)$) is described. It can be applied to analysis of various problems, including computation of the sequence $a_1x, a_2x^2, \ldots, a_mx^m$ and computation of a polynomial $\sum_{k=1}^{m} a_k x^{\alpha(k)}$ where $0 \leq \alpha(1) < \alpha(2) < \ldots < \alpha(m)$. The method is based on suitable substitutions of elements of $F(x)$ for the indeterminates $a_1, \ldots, a_m$.

ON THE EFFICIENCY OF PARALLEL ITERATIVE ALGORITHMS FOR NON-LINEAR EQUATIONS

H.T. Kung and J.F. Traub
Carnegie-Mellon University

ABSTRACT

A parallel iterative root-finding algorithm $\Sigma(m) = (\sigma, \Lambda(m,\sigma))$ consists of an iteration function σ which defines iterates approximating a root of a non-linear equation and a procedure $\Lambda(m,\sigma)$ which computes the iteration function σ by using m processors. In this paper we define an efficiency measure for $\Sigma(m)$ and use this efficiency measure to study the gain in speed by using m processors. We also analyze various known parallel methods from the efficiency point of view.

This work was supported in part by the National Science Foundation under Grant GJ-3211 and by the Office of Naval Research under Contract N00014-67-A-0314-0010, NR044-422.

The Parallel Execution of FORTRAN DO Loops

Leslie Lamport

Massachusetts Computer Associates, Inc.

Two methods are described for compiling a nest of DO loops for execution on a parallel computer. The hyperplane method obtains parallel execution along linear subspaces of the index set. It is suitable for both asynchronous and synchronous multiprocessor computers. The coordinate method obtains parallel execution along one or more loop indices. If necessary, it will change the order of execution of the various parts of the loop body. It is suitable for any synchronous parallel computer, such as an array or pipelined vector machine.

These methods will yield parallel execution for a large class of loops. This has implications for the design of future computers and their compilers.

COMPUTING TIMES FOR THE INTEGRAL SOLUTION OF LINEAR EQUATIONS

by

Michael T. McClellan
Computer Science Center
University of Maryland

Abstract

Algorithms for computing the greatest common divisor (gcd) of n integers are reviewed. Recently derived computing time functions for the case of two integers, obtained to within codominance, are applied to obtain similar functions for the case of n integers. Related algorithms for the integer solution of systems of linear equations are then considered. Explicit functions of system size and coefficient bounds are obtained, which dominate the computing times of these algorithms.

The Complexity of Roundoff Analysis

Webb Miller

IBM T. J. Watson Research Center

The complexity of Wilkinson-style roundoff analysis is discussed. Very powerful, yet simple, heuristic computer techniques are described.

EFFICIENT IMPLEMENTATION OF FUNCTIONAL APPROXIMATION ALGORITHMS FOR PICTURE PROCESSING

T. Pavlidis
Princeton University

ABSTRACT

Algorithms based on piecewise functional approximation can be very useful in many applications of pattern recognition and picture processing[1-4].However their usual implementations tend to be slower than those of simple heurestic algorithms because their time complexity tends to be of order N^2 where N is the size of the picture(for a kxk picture $N=k^2$).We will investigate the problem of increasing their speed.

In general such algorithms are iterative.If T_0 is the initialization time,T_i the time per iteration and I the number of iterations,the total time will be $T_0+I\cdot T_i$.If m is the order of approximating polynomials and R the number of regions then T_0 is of order mN for both uniform approximation(using linear programming) and integral square error approximation(using closed formulas with integrals over the picture domain).If the approximation is reevaluated on each iteration then T_i is also of order mN.However it is possible to device schemes where only updating of the approximations is necessary so that T_i is of order mR.The use of a split-and - merge algorithm which not only varies region boundaries but also merges or splits regions guarantees that the number of iterations is at most of order N/R[4].Therefore the total time is of order mN and the functional approximation algorithms become competitive in speed with the simple heuristic algorithms.

Practical experience has shown that the introduction of heuristics in the initial choice of the regions can reduce the number of iterations sufficiently so that the initialization term becomes dominant.

In the past efforts to improve the speed of picture processing algorithms have involved the use of arrays of complex processors with all the problems of synchronization associated with them.The following seems to be a better use of parallelism:Since most of the time is spent either in additions (evaluating integrals for integral square error approximations) or in locating maxima(linear programming schemes for uniform approximation) it is advisable to use arrays of very simple processors which will perform these operations,possibly without necessating the transfer of data from memory except for the final results.Then the initialization would become of order R rather than N.

References:

1.T.Pavlidis,in Proc.of 1972 Annual ACM Conference,pp.631-636.
2.T.Pavlidis,in Computer Graphics and Image Processing(in press).
3.T.Pavlidis,in IEEE Trans.on Computers(in press).
4.T.Pavlidis and S.L.Horowitz,"Piecewise Approximation of Plane Curves"(submitted for publication).

PROBLEM-CLASSES AND CHARACTERISTIC ALGORITHMS

Dr. Jürgen Perl
Technische Universität Berlin

ABSTRACT

Algorithms to find out shortest ways in graphs can be applied in an analogous way to generalized problems (f.e. see Hu, Visotschnig, Pair, etc.). Because of the good efficiency of these graph-algorithms it is reasonable and usefull to know problems as much as possible which can be solved by these algorithms.

This method in a generalized form can be used for an economic search for algorithms to solve practical problems:

Problems which have similar characteristic structure relative to best-solving algorithms are collected to classes. Each problem-class then corresponds with a characteristic best-solving algorithm.

To find a solving algorithm for a problem then means:

> Analyse the structure of the problem,
> find out the problem-class the problem belongs to,
> and take the characteristic algorithm.

This principle shall be illustrated by the example of covering-problems and graph-algorithms:

It will be shown that the so-called linear covering-problems can be solved with the above-mentioned graph-algorithms.
On the contrary the general covering-problem defines an own problem-class with other characteristic algorithms.

A NEW DIFFERENCING ALGORITHM FOR SOLVING PARTIAL DIFFERENTIAL EQUATIONS

by

DR. GERALD N. PITTS

CENTRAL TEXAS COLLEGE

and

DR. BARRY L. BATEMAN

TEXAS TECH UNIVERSITY

ABSTRACT

Many physical systems in engineering can be represented by partial differential equations. These differential equations are normally solved through the use of numerical approximation, i.e. finite differences. This representation of the system entails a system of simultaneous finite difference equations for each partial equation utilized in the model.

There are several well known techniques for solving these systems of simultaneous difference equations and they all involve repetitive differencing of absolute magnitudes which can cause both a loss of numerical accuracy and an increase in solution time.

The purpose of this paper is to describe a new differencing scheme which will better the accuracy of past methods as well as reduce solution time. This new method utilizes the concept of operating on the differences themselves instead of differencing absolute magnitudes on each iteration to solution.

Tables of comparison between the new method and previously used methods will be provided for illustrating the increased accuarcy and reduced solution time.

ON THE COMPLEXITY OF ALGORITHMS

FOR SYMMETRIC COMPUTATIONS

Robert L. Probert
Dept. of Applied Analysis & Computer Science
University of Waterloo

ABSTRACT

An equivalence between bilinear chains and matrix multiplication algorithms which do not exploit commutativity of multiplication is demonstrated. From this equivalence and a decomposition theorem of Fiduccia, we derive a characterization of matrix multiplication algorithms as a tensor matrix-vector product which can be decomposed into the product of three elementary matrices by a vector. This characterization provides a facility for proving the main Symmetry Theorem, namely that algorithms to compute any of the six matrix products of the forms (m*n)(nxp), (nxm)(mxp), (pxm)(mxn), (mxp)(pxn), (nxp)(pxm), (pxn)(nxm) require the same number of multiplication steps. In addition, we exhibit a straightforward method of obtaining equicomplex algorithms for nmp and pmn products from one for mnp products.

As a simple application of the result we modify a theorem of D. Kirkpatrick to obtain a lower bound of $m_x(d_< + d_> - 1)$ multiplications for matrix multiplication problems of the form (mxn)(nxp) where

m_x = max {m,n,p} , and d_g, d_j have the same values

as the two lesser dimensions.

The results imply that the order of the dimensions in an mnp product is of no consequence to the multiplicative complexity of the corresponding computation. In fact, the results provide considerable evidence that the lower bound is a function of the relative sizes of the dimensions; thus, the more symmetric m,n, and p are (with product mnp fixed), the lower the multiplicative complexity of the corresponding matrix multiplication problem.

THE INFLUENCE OF COMPUTATION SCHEMATA REPRESENTATIONS
OF SIGNAL PROCESSING ALGORITHMS
ON THE ARCHITECTURE OF THE AN/UYK-17 COMPUTER

Tomlinson G. Rauscher
Barry P. Shay

Code 5^90
Information Processing Systems Branch
Communications Sciences Division
Naval Research Laboratory
Washington, D. C. 20375

In modern signal processing systems, tasks such as wave form generation, filtering, and spectrum analysis may be performed by sampling analog signals and transfonning the sampled digital data. For systems like radar and sonar, required sampling rates and hence computation speeds were so high that special-purpose hardwired devices were required to perform the desired computations. The development of LSI circuit technology and high speed memories now make the construction of programmable high speed digital signal processors feasible.

The capability to effect the second order recursive filter and the FFT butterfly is fundamental in the computation of signal processing algorithms applicable to our interests. Analysis of the computation schemata for the filter a#d FFT processes reveal a high degree of parallelism which should be exploited to perform the processes quickly and efficiently. The similarities between the sequencing and structure of the filter and FFT schemata suggest the design of a computer which can manage both processes.

The architecture of the Signal Processing Arithmetic Unit (SPAU) of the AN/UYK-17 computer reflects the computation schemata of the second order filter and the FFT butterfly. The arithmetic section contains four multipliers and four adders which operate in parallel with each other and with transfers to and from high speed memories. The provision of internal registers facilitates pipelining to further increase system throughput. The SPAU provides flexibility in the development of signal processing algorithms through user horizontal microprogrammed control of the parallel internal resources. The cycle time of the SPAU is 150 nanoseconds; a ***102k*** point FFT can be computed in 1.5 milliseconds.

To allow the SPAU to concentrate on the transformation of signal information, the AN/UYK-17 contains a separate Microprogrammed Control Unit (MCU) which accepts and organizes the data which the SPAU accesses from high speed buffers. Hence, the SPAU need not consider the problems of i/o processing.

AN $O(n^2)$ METHOD FOR SOLVING CONSTANT COEFFICIENT BOUNDARY VALUE PROBLEMS IN TWO DIMENSIONS

D. J. Rose and R. Bank
Harvard University

ABSTRACT

Let M be an $n^2 \times n^2$ matrix of block tridiagonal form

$$M = \begin{bmatrix} T & -I & & & \\ -I & T & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & T & -I \\ & & & -I & T \end{bmatrix}$$

where T is an $n \times n$ tridiagonal matrix, and I is the $n \times n$ identity.

We show that the solution x to $Mx = k$ can be obtained in $0\ (n^2)$ arithmetic operations, (and $0\ (n^2)$ storage). This is asymptotically faster than previously studied methods.

In practice stability appears to be of some concern. Some modifications are discussed in the context of numerical examples. The method generalizes in a straightforward fashion to constant coefficient elliptic boundary value problems in two and higher dimensions.

Matrix Multiplication and Polynomial Evaluation

with Known Matrices and Polynomials

by

J. E. Savage

Center for Computer and Information Sciences

and

Division of Engineering
Brown University
Providence, R.I. 02912

Abstract

In this talk, we consider algorithms for the computation of $m \times n$ matrix-vector products and the evaluation of m polynomials of degree n in one variable when the matrix entries and polynomial coefficients are known before the algorithms are constructed. When the entries and coefficients assume at most s distinct values, $m, n >> s$, products can be computed and polynomials evaluated with numbers of operations which are on the order of $mn/\log_s(m)$ and $mn/\log_s(mn)$, respectively. Furthermore, these upper bounds can be improved upon by at most constant factors for the worst case matrix and worst case set of polynomials.

The algorithms given in this paper require a search of matrix entries and polynomial coefficients after which the bounds mentioned above apply. Since the setup costs can be amortized over many matrix multiplications and polynomial evaluations, they offer a potential improvement over preconditioning when the matrix entries and polynomial coefficients are bounded.

* The research reported here was completed primarily at Brown University and in part at the Jet Propulsion Laboratory, Pasadena, California. The Brown portion was supported in part by a grant from the National Science Foundation, Grant No. GJ-32270, and in part by a grant from the Advanced Research Projects Agency of the Department of Defense which was monitored by U.S. Army Research Office - Durham, Grant No. DA-ARO-D-31-124-73-G65. The JPL portion was supported by a grant from the National Aeronautics and Space Administration.

THE COMPLEXITY OF SCHEDULING ALGORITHMS

FOR MULTIPROCESSOR SYSTEMS

Dr. Sigram Schindler

Technische Universität Berlin

ABSTRACT

The problem of scheduling N tasks - the operational precedence structure of which is represented as a finite, weighted forest or anti-forest G - on a multiprocessor system consisting of M identical processors was investigated in [1] - [4]. The weight W_I of node I, $1 \leq I \leq N$, is regarded as the processing-time of the task represented by node I, and all tasks are to be processed completely within total processing time CT. It is assumed that preemptions of all tasks are allowed (except in [1]).

This paper is concerned with the complexity of the scheduling algorithms developped in [1]-[4]. For this purpose they are reformulated in pseudo-Algol allowing parallel execution and exhibiting the 'channel load' they may impose the computer system (depending on the seize of primary memory). So the different aspects of complexity of scheduling algorithms and schedules can be stated explicitly. Bounds for the number of preemptions are given; pathological cases are constructed, showing that scedules with few preemptions might be less suitable than schedules with many preemptions.

The extension of the investigations to the case treated in [5] and [6], where G is assumed to be a finite, directed, acyclic, weighted graph (and M=2), is discussed shortly.

References

[1] HU,T.C.: Parallel Sequencing and Assembly Line Problems, Operations Research 9, No. 6 (1961).

[2] MUNTZ, R.R. and COFFMAN, E.G.: Preemptive Scheduling for Real-Time Tasks on Multiprocessor Systems, JACM, Vol. 17, No. 2, April 1970.

[3] SCHINDLER, S.:Classes of Optimal Schedules for Multiprocessor Systems, 2. Jahrestagung der Gesellschaft für Informatik, Karlsruhe, Oct. 2-4, 1972.

[4] ____________ Scheduling a Multiprocessor on Anti-Forests, SIAM/SIGNUM Fall Meeting, Austin, Oct. 16-18, 1972.

[5] ____________ and SIMONSMEIER, W.: The Class of all Optimal Schedules for Two-Processor Systems, 7. Annual Princeton Conference on Information Sciences and Systems, March 22-23, 1973.

[6] COFFMAN, E.G. and GRAHAM, R.L.: Optimal Scheduling for Two-Processor Systems, Acta Informatica 2 (1972).

ANALYSIS OF A FAMILY OF ALGORITHMS
FOR THE EVALUATION OF A POLYNOMIAL
AND ITS DERIVATIVES

Mary Shaw and J. F. Traub

Carnegie-Mellon University

ABSTRACT

We have previously presented a one-parameter family of algorithms for the evaluation of a polynomial and some of its derivatives [1]. We point out certain members of the family that correspond to well-known techniques. An analysis of the way the number of multiplications depends on the parameter is given, and a technique for selecting the best value of the parameter, given the degree of the polynomial and the number of derivatives desired, is indicated.

[1] Shaw, Mary and J. F. Traub, "On the Number of Multiplications for the Evaluation of a Polynomial and Some of Its Derivatives," Computer Science Department Report, Carnegie-Mellon University, August, 1972.

This research was supported in part by the National Science Foundation under Grant GJ-32111 and the Office of Naval Research under Contract N00014-67-A-0314-0010, NR 044-422.

A MICROPROGRAMMED IMPLEMENTATION OF PARALLEL PROGRAM SCHEMATA

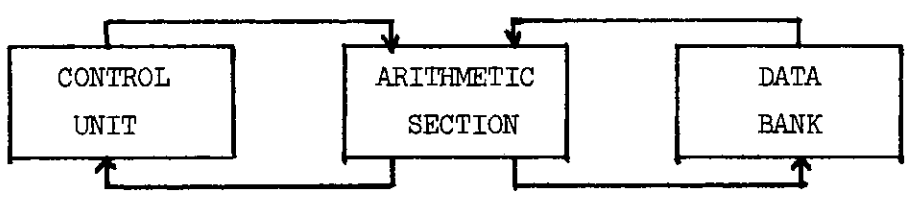
Barry P. Shay

Naval Research Laboratory
Washington, D. C. 20375
and
University of Maryland
College Park, Maryland

Many problems of particular interest are parallel in nature and require some degree of parallel processing to be solved rapidly. This is especially true for digital signal processing algorithms. These algorithms are highly structured and they may be decomposed into computing kernels of varying degrees of complexity. In their decomposed forms, the algorithms display potential parallelism at many levels. This potential must be realized by any envisioned computing structure on which they will be executed.

A model was chosen which can represent parallelism in a definitive way. This model, called parallel program schemata or computation schemata, is a means of representing algorithms (programs) to be executed by a machine. The model basically consists of two parts: a data flow graph (DFG) and a precedence or control graph (PRG). The usefulness of the model is that all valid execution sequences are apparent from the partial ordering relationships among operator occurrences of the PRG.

A machine model has been developed which implements computation schemata. The machine consists of three parts as indicated below:



The arithmetic section contains operators, registers, selectors and decision elements. The data bank represents both a source and sink for data which is continually being transformed by the arithmetic section. The control unit configures and sequences the arithmetic section.

Since the machine model has been developed as an implementation of schemata, the schemata description of the algorithm and the description of the machine are represented in a unified manner. That is, the representation makes no distinction between software (i.e., the algorithm) and hardware (i.e., the machine resources).

The model developed can be considered a generalization of a horizontally microprogrammed machine. As such, the problem of reducing the number of words in the micro-program while making maximum utilization of hardware arises. This problem may be attacked in a variety of ways including:

1. partitioning an input schema modulo the machine (assignment of registers and operators),

2. a single assignment type description of the schema input,

3. formal state minimization of finite automata,

4. combining elementary operations into a single operation by matrix multiplication.

ITERATIVE SOLUTION OF BLOCK TRI-DIAGONAL SYSTEMS
ON PARALLEL OR VECTOR COMPUTERS

D. Stevenson and J.F. Traub
Carnegie-Mellon University

ABSTRACT

The Parallel Gauss Algorithm introduced in (1) for tridiagonal matrices is generalized to block tridiagonal matrices. The convergence properties for block tridiagonal matrices are analogous to those proved for tridiagonal matrices.

The Parallel Gauss and Jacob's algorithms are compared on a model problem. The trade-off between the two algorithms depends on the size of the blocks and on the diagonal dominaces of the matrix.

(1) J.F. Traub, Iterative Solution of Tridiagonal Systems on Parallel or Vector Computers. Proceedings of Symposium on Complexity of Sequential and Parallel Numerical Algorithms.

This research was supported in part by the National Science Foundation under Grant GJ-32111 and the Office of Naval Research under Contract N00014-67-A-0314-0010, NR 044-422.

A PARALLEL, VARIABLE METRIC OPTIMIZATION ALGORITHM

by Terry A. Straeter

NASA, Langley Research Center
Hampton, Virginia

Abstract

This paper introduces an algorithm designed to exploit the stream, parallel, or pipeline computing capabilities of the next generation of computers (ILLIAC and STAR). If p is the degree of parallelism, then one cycle of the parallel variable metric algorithm is defined as follows: first, the function and its gradient are computed in parallel at p locations; then the metric is modified by p rank one corrections; and finally a single univariant minimization is carried out in the Newton-like direction. Several properties of this algorithm are established in the paper. In addition, the convergence of the iterates to the solution is proved for a quadratic functional on a real separable Hilbert Space; in fact, for a finite dimensional space the convergence is in one cycle. Results of numerical experiments indicate that the new algorithm will exploit the stream, parallel, or pipeline computing capabilities of the new computers to effect faster convergence than serial techniques currently in use. In fact, the experiments indicate that even when the computations are done serially, the new algorithm is very competitive with the widely used Davidon–Fletcher–Powell technique.

ON THE EFFICIENT VALIDATION OF EXECUTION SEQUENCES OF SIMPLE RECURSIVE AND PARALLEL PROGRAM SCHEMATA

J. van Leeuwen*

Department of Computer Science
University of California at Berkeley
Berkeley, California 94720

Mathematics Department
State University at Utrecht
Utrecht, Netherlands

In general, (non-deterministic) recursive procedures with non-nested calls are not flow-chartable (Strong [2]). If the recursion depth is known, e.g. expressed by a formal parameter, and the scheme is of recurrent type (see e.g. Herman [1]), one can use this value to pre-set the stack once and for all, creating the required space for activation records or pointers and giving way for an efficient, iterative simulation. The same technique works for simple independent parallel processes. The machine implementation leads to the model of non-deterministic preset pushdown automata. We will identify a (recursive) procedure with the collection of concrete execution sequences it gives rise to. Some results are:

Theorem. Algorithms verifiable by locally finite pre-set pda with the finite return property are exactly those described by recursive procedures of recurrent type. (For explanation of undefined terms see [3], [4].)

Theorem. Algorithms recognizable for l.f. preset pda with f.r.p. are recognizable on a deterministic non-erasing stack machine.

Theorem. Algorithms recognizable for l.f. preset pda with f.r.p. are deterministically recognizable in space n (on a Turing machine).

A detailed analysis of the pre-set pushdown technique is given in [3], [4].

* * * * *

[1] Herman, G.T., et al., Description of languages using recurrence systems, Tech. Rept., Dept. of Comp. Sci., SUNY at Buffalo.

[2] Strong, H.R., Translating recursion equations into flow-charts, JCSS 5 (1971) 254-285.

[3] van Leeuwen, J., Pre-set pushdown automata and oL-grammars, Tech. Rept. 10, Dept. of Comp. Sci., UC Berkeley.

[4] van Leeuwen, J., Pre-set pushdown automata and their languages (in preparation).

*Research supported by the Dutch Organization for the Advancement of Pure Research (ZWO).