RIACS

Research Institute for Advanced Computer Science
NASA Ames Research Center

TOTALLY PARALLEL MULTILEVEL ALGORITHMS FOR SPARSE ELLIPTIC SYSTEMS

Paul O. Frederickson

March, 1989

Research Institute for Advanced Computer Science
NASA Ames Research Center

RIACS Technical Report 89.10

NASA Cooperative Agreement Number NCC 2-387

TOTALLY PARALLEL MULTILEVEL ALGORITHMS FOR SPARSE ELLIPTIC SYSTEMS

Paul O. Frederickson

Research Institute for Advanced Computer Science
NASA Ames Research Center

RIACS Technical Report 89.10
March, 1989

The fastest known algorithms for the solution of a large elliptic boundary value problem on a massively parallel hypercube all require O(log(n)) floating point operations and O(log(n)) distance-1 communications, if we define massively parallel to mean a number of processors proportional to the size n of the problem. The algorithm TPMA (for Totally Parallel Multilevel Algorithm) that we describe below has, as special cases, four of these fast algorithms. These four algorithms are PSMG (Parallel Superconvergent Multigrid) of Frederickson and McBryan, Robust Multigrid of Hackbusch, the FFT based Spectral Algorithm, and Parallel Cyclic Reduction. The algorithm TPMA, when described recursively, has four steps:

(1) Project to a collection of interlaced, coarser problems at the next lower level.

(2) Apply TPMA, recursively, to each of these lower level problem, solving directly at the lowest level.

(3) Interpolate these approximate solutions to the finer grid, and a verage them to form an approximate solution on this grid.

(4) Refine this approximate solution with a defect-correction step, using a local approximate inverse.

Choice of the projection operator P, the interpolation operator Q, and the smoother S determines the class of problems on which TPMA is most effective. There are special cases in which the first three steps produce an exact solution, and the smoother is not needed (e.g. constant coefficient oeprators).

Key Words: Multilevel algorithm; Multigrid; Cyclic reduction; Spectral method .

Work reported herein was supported in part by Cooperative Agreement NCC2-387 between the National Aeronautics and Space Administration (NASA) and the Universities Space Research Association (USRA).

TOTALLY PARALLEL MULTILEVEL ALGORITHMS FOR SPARSE ELLIPTIC SYSTEMS

Paul O. Frederickson

Research Institute for Advanced Computer Science
NASA Ames Research Center

1: Introduction.

The fastest known algorithms for the solution of an elliptic boundary value problem

$$Au = v \qquad (1.1)$$

on a *massively parallel* hypercube, by which we mean a hypercube with a number p of processors proportional to the size n of the problem, are all very closely related in structure. It is immediately apparent that all proceed in O(log(n)) stages (or levels) consisting of $O(n)$ floating point operations executed in parallel and $O(p)$ parallel communications with a nearest neighbor. On closer examination one observes that in the k^{th} level of any of these algorithms the problem

$$A^k u^k = v^k \qquad (1.2)$$

is, in effect, being solved, and that this problem actually consists in a number of independent and interleaved subproblems. It is this observation that we wish to clarify in the following sections.

The algorithm TPMA (for Totally Parallel Multilevel Algorithm) that we describe below has, as special cases, four of these fast algorithms. These four algorithms are PSMG (Parallel Superconvergent Multigrid) of Frederickson and McBryan, Robust Multigrid of Hackbusch, the FFT based Spectral Algorithm, and Parallel Cyclic Reduction. Choice of the projection operator P^k, the interpolation operator Q^k, and the smoothing operator S^k in the algorithm TPMA determines which of these particular algorithms is represented. Which algorithm one wishes to use depends on many things, among them the characteristics of the problem (1.1). What new algorithms fill the space between these known ones is yet to be determined.

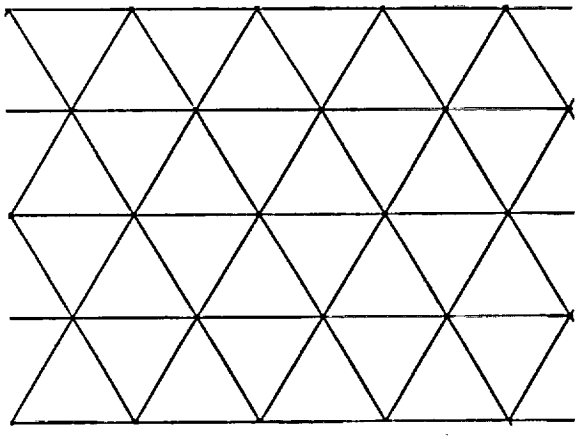

Fig. 1

It is useful, when attempting to understand the algorithm TPMA, to see clearly the intertwined subgrids at each of the multiple levels of the algorithm. These are more easily visualized in the case of plane triangulations, and hence we begin with these.

2: Graph Colorings and Subgrids.

The nodes of an arbitrary planar graph can be colored using at most four colors with no two adjacent nodes having the same color. Fewer colors may suffice. For example, the graph in Fig. 1 corresponding to an equilateral triangulation of the plane requires only three colors.

If we connect the nodes of the same color that are a graph distance two apart we form three interlaced subgrids, each larger by a factor $\sqrt{3}$ and rotated, as shown in Fig. 2. Observe that each subgrid corresponds to a subgraph of the square of the original graph. This is equivalent to the statement that nodes connected by an edge in the subgrid connect points that are separated by two edges of the original graph.

Similarly, the graph of the familiar five point Laplacian requires only two colors. The two subgrids that are formed by connecting nodes of like color which are a graph distance two apart in the original grid are larger by the factor $\sqrt{2}$. Use of this grid offers some advantage over the triangular grid when computing on a binary hypercube, for communication distance along each grid axis remains a power of two as the algorithm descends through the levels. Hypercube communication distance is therefore bounded by four.

Work reported herein was supported in part by Cooperative Agreement NCC2-387 between the National Aeronautics and Space Administration (NASA) and the Universities Space Research Association (USRA)

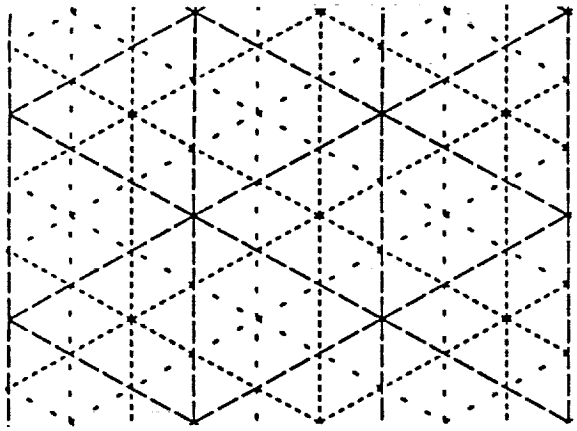

Fig. 2

When using TPMA to solve an elliptic problem discretized onto an unstructured triangular grid we are able to generate interlaced subgrids using an imperfect three-coloring, in which the number of adjacent nodes of the same color is minimized, or an imperfect two-coloring. This construction of interlaced subgrids is repeated recursively: each of these subgrids is a triangulation in its own right.

Finally, we observe that the graph of the nine-point Laplacian, although not planar, is colorable in four colors. The four interleaved subgrids, each larger by the factor $2 = \sqrt{4}$, lead to a very high performance implementation of TPMA on highly parallel hypercubes, and the CM2 in particular.

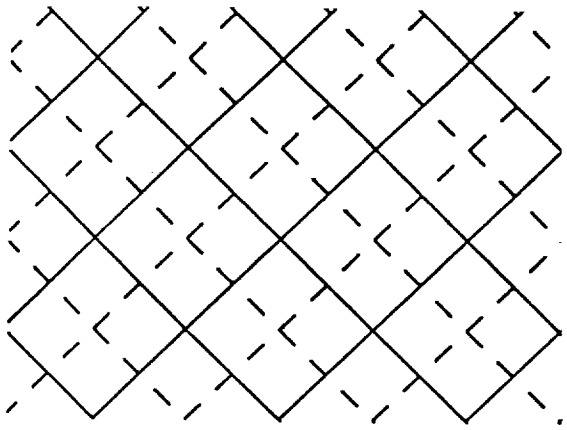

Fig. 3

3: TPMA: Totally Parallel Multilevel Algorithm.

Definition of the algorithm TPMA requires specification of the two operators Q^k and P^k for every level $0 < k \leq m$, and the operators A^k and S^k for $0 \leq k \leq m$. Consider first the projection operator $P^k\ y^k \rightarrow y^{k-1}$ which uses the data v^k in the equation

$$A^k u^k = v^k \tag{3.1}$$

to construct the data $v^{k-1} = P^k v^k$ of the corresponding equation at level $k-1$. The easiest example of a projection operator, $P^k = I$, is used in some versions of the algorithm TPMA. In this case the first step in projecting equation (3.1) to the interleaved subgrids at level $k-1$ is particularly easy.

Suppose that we have been able, somehow, to construct an approximate solution u^{k-1} in X^{k-1} to the equation $A^{k-1}u^{k-1} = v^{k-1}$. Then we will use the interpolation operator Q^k to map it into an approximate solution $u^k = Q^k u^{k-1}$ of equation (3.1).

The effect of Q^k is to combine the approximate solutions from all of the interleaved subgrids of a given grid into one approximate solution on that grid. Except at the highest level this grid will, in turn, be one of several interleaved subgrids of a grid at a yet higher level. In many cases Q^k is best described as an averaging operator, while in other cases it will simply be the identity.

The convergence theory is particularly easy to state when the two operators Q^k and P^k are adjoint or dual to each other. This is the situation, for example, when they are constructed using the Raleigh-Ritz-Galerkin procedure.

In most cases the operator A^{k-1} at level $k-1$ is defined recursively using

$$A^{k-1} = P^k A^k Q^k, \tag{3.2}$$

or, equivalently, via the commutative diagram

$$\begin{array}{ccc} X^k & \xrightarrow{A^k} & y^k \\ \big\downarrow Q^k & & \big\downarrow P^k \\ X^{k-1} & \xrightarrow{A^{k-1}} & y^{k-1} \end{array} \tag{3.3}$$

The task now is to solve the system (3.1) at every level k. At level $k = 0$ this is easier than at any other level, for the original system has been reduced to as many independent systems as possible. In many cases each of these system is only a scalar equation, and solution requires only a division. In any case we will denote the solution operator by S^0.

In general the approximate solution $u^k = Q^k u^{k-1}$ is not an exact solution to eqn. (3.1), and it is advantageous to use a local approximate inverse P^k to the operator A^k in a defect- correction step

$$u^k \leftarrow u^k + P^k(\ v^k - A^k u^k\ ) \tag{3.4}$$

after interpolation. In some cases the Jacobi operator (the reciprocal of the diagonal of the operator A^k) is adequate as a defect correction operator S^k, but in many cases it is worthwhile to take into account more of the structure of A^k when constructing S^k.

The algorithm TPMA that we have described implicitly above may now be defined explicitly to be a representation of the operator T^k given recursively by

$$T^k = S^k + (I - S^k A^k) Q^k T^{k-1} S^k \qquad (3.7)$$

with $T^0 = S^0$ as initial condition. When the four operators satisfy certain inequalities (see [4] for details) one can prove that T^k is an approximate inverse to A^k.

We will see that there are versions of TPMA for which $u^k = Q^k u^{k-1}$ is an exact solution to equation (3.1), rather than just an approximate solution. In this case the local approximate inverse S^k is not needed, or may be taken to be 0, except at the bottom level $k = 0$ where we assume that $S^0 A^0 = I$. Then the recursive definition of T^k reduces to $T^k = Q^k T^{k-1} P^k$, or the commutative diagram

$$\begin{array}{ccc} X^k & \xleftarrow{T^k} & Y^k \\ \downarrow Q^k & & \downarrow P^k \\ X^{k-1} & \xleftarrow{T^{k-1}} & Y^{k-1} \end{array} \qquad (3.6)$$

4: PSMG: Parallel Superconvergent Multigrid.

When solving an elliptic problem discretized onto a given grid (or graph) it is useful to have an approximate solution on a coarser grid, for this may be interpolated onto the given grid to serve as the initial approximation of a defect correction algorithm. This idea leads one to the classic multigrid algorithm, perhaps the most obvious example of a multilevel algorithm. In multigrid, recursively coarser grids are used, with the lowest level grid having so few points that the corresponding approximate elliptic problem can be solved directly. These algorithms are multilevel, but not totally parallel. In fact, almost all of the processors are standing idle almost all of the time. Isn't there something useful that these idle processors could do to further the solution?

This is the question that Oliver McBryan and the author asked themselves, and that led them to develop the totally parallel multilevel algorithm PSMG [4,5]. The essential idea of PSMG is to project the given problem onto all of the available coarser grids, forming enough lower level problems to keep all of the processors busy all of the time. This is done recursively, as in ordinary multigrid. The payoff is much faster convergence at no added computational cost. In fact, the program is somewhat simpler on the Connection Machine, for it is no longer necessary to turn more of the processors off at every step.

5: The FFT Based Spectral Method.

The classical spectral algorithm for the solution of a constant coefficient elliptic problem on a rectangular domain has three steps: take the FFT of the problem data v, divide this by the transform of the differential operator, and transform back. We may, however, equally well describe it as an example of the algorithm TPMA in which the operator P^k is given by

$$P^k{}_0 = \frac{1}{\sqrt{2}} (\; 0 \quad 1 \quad \omega_k{}^{i_0} \;) \qquad (5.1)$$

when $0 \le (\; i \;\; mod \;\; 2^{m-k+1} \;) < 2^{m-k}$, and

$$P^k{}_1 = \frac{1}{\sqrt{2}} (\; 1 \quad -\omega_k{}^{i_0} \quad 0 \;) \qquad (5.2)$$

otherwise. This three point operator works on points a distance 2^{m-k} apart, which are the points of the subgrid at that level. This is the operation of P^k in the first dimension, and is followed by a pair of operators transverse to these in each of the other dimensions. Q^k is the adjoint of P^k, and $S^k = 0$ except at the lowest level, where S^0 is the reciprocal of the fourier transform of the differential operator.

An important observation is that the Fourier multipliers $\omega_k{}^i$ need be computed only once, on the highest level. Those coefficients needed at the next lower level are a Hamming distance at most two away, and are moved in at the start of the computation at that level. For a more detailed discussion of FFT implementations on highly parallel computers see the recent papers of Kamin and Adams [1] and Schwarztrauber et. al. [9].

6: Parallel Cyclic Reduction.

The cyclic reduction (odd-even reduction) algorithm of Buneman [2] and Hockney [6] for solving a tridiagonal, or block tridiagonal, system of equations of the form

$$(A \; u)_i = a_{i,i-1} u_{i-1} + u_i + a_{i,i+1} u_{i+1} = v_i \qquad (6.1)$$

is another example of the algorithm TPMA. Here the projection operator P^k is defined by

$$P^k \; v_i = -a^k{}_{i,i-j} v_{i-j} + v_i - a^k{}_{i,i+j} v_{i+j}, \qquad (6.2)$$

where j denotes 2^{m-k}, and the lower-level operators A^k are defined, recursively, by

$$A^{k-1} = P^k \; A^k \qquad (6.3)$$

In the standard version of this algorithm, optimal on a sequential computer, the number of equations reduces by a factor of two at every step. After log(n) steps,

but only O(n) operations, a single system remains, and this is solved directly. Log(n) stages of back substitution remain to be done before the solution is known over the whole array.

The totally parallel version of this algorithm (Hockney and Jessope [7]) projects at every stage onto all nodes of the grid using the same projection operator (6.2). The advantage is that the solution is known at all nodes after the first log(n) stages. It is just twice as fast, therefore, as ordinary cyclic reduction on a sufficiently parallel computer. This is the version of cyclic reduction that is a special case of TPMA. To see this, observe that eqn. (6.3) is equivalent to eqn. (3.2) with $Q^k = I$, which is what parallel cyclic reduction requires. Because Interpolation is exact, we take the TPMA operator $S^k = 0$ in this case. For a fuller discussion of parallel cyclic reduction and related direct multilevel algorithms the recent paper of Swartztrauber and Sweet [8] is recommended.

Summary.

We describe the algorithm TPMA (Totally Parallel Multilevel Algorithm) and demonstrate that three of the fastest known algorithms for solving an elliptic boundary value problem on a highly parallel hypercube, such as the Connection Machine of Thinking Machines Corporation, are special cases of TPMA. These are the FFT based spectral algorithm, parallel cyclic reduction, and PSMG. Each of these appears to be optimal in certain situations. Since all are special cases of the same algorithm TPMA it may be possible to combine the advantages, and form hybrid algorithms that are better than any of these in particular problem domains.

Acknowledgements

Most of the work reported here is the result of a long and fruitful collaboration with Oliver McBryan. The research benefited greatly from discussions with Walter Tichy, Creon Levit and Eric Raible on hypercube communication schemes, with Roland Sweet, Tom Jordan and Youcef Saad on cyclic reduction algorithms, and with David Bailey and Paul Swartztrauber on parallel implementations of the FFT algorithm. Finally, the work of Craig Douglas and Willard Miranker, even though it moves in a somewhat different direction, is another motivation for the present research.

REFERENCES

[1] Ray A. Kamin III and George B. Adams III, *Fast Fourier Transform Algorithm Design and Tradoffs on the CM-2*, to appear, International Journal of High Speed Computing.

[2] O. Buneman, *A compact non-iterative Poisson solver*, Report 294, Stanford University Institute for Plasma Research,

[3] Paul O. Frederickson, *Totally Parallel Multilevel Algorithms*, RIACS Technical Report 88.34.

[4] Paul O. Frederickson and Oliver A. McBryan, *Parallel superconvergent multigrid*, in *Multigrid Methods*, S. F. McCormick (editor), Marcel Dekker, New York and Basel, 1988.

[5] Paul O. Frederickson and Oliver A. McBryan, *Intrinsically Parallel Multiscale Algorithms for Hypercubes*, in *The Third Conference on Hypercube Concurrent Computers and Applications*, Geoffry Fox (editor), ACM Press, New York, 1988.

[6] Roger Hockney, *A fast direct solution of Poisson's equation using Fourier analysis*, J. Assoc. Comput. Mach. 12 (1965) 95–113.

[7] Roger Hockney and Chris Jessope, *Parallel Computers: Architecture, Programming and Algorithms*, Adam Hilger, Bristol, 1981.

[8] Paul N. Swartztrauber and Roland A. Sweet, *Vector and parallel methods for the direct solution of Poisson's equation*, to appear, Journal of Applied and Computational Mathematics.

[9]Paul N. Swarztrrauber, Roland A Sweet, William L. Briggs, Van Emden Henson, and James Otto, *Bluestein's FFT for Arbitrary N on the Hypercube*, submitted, Parallel Computing.

[13] R. Kumar. A fast algorithm for solving a Toeplitz system of equations. IEEE Trans. Acoust. Speech and Signal Processing 33 (1985), pp. 254-267.

[14] N. Levinson. The Weiner RMS (root mean square) error criterion in filter design and prediction. J. Math. Phys. 25 (1947), pp. 261-278.

[15] Victor Pan and Robert Schreiber. An improved Newton iteration for the generalized inverse of a matrix, with applications. Submitted to J. ACM.

[16] Gilbert Strang. A proposal for Toeplitz matrix calculations. Studies in Applied Mathematics 74 (1986), pp. 171-176.

[17] Gilbert Strang and Alan Edelman. The Toeplitz-circulant eigenvalue problem $Ax = \lambda Cx$. Oakland Conference on PDEs, L. Bragg and J. Dettmans, eds., Longmans, London, 1987.

[18] W.F. Trench. An Algorithm for Inversion of Finite Toeplitz Matrices. SIAM J. 12:3 (1964), pp. 515-522.