27
7-19-77
25 DWTIS

UCID- 17515

Lawrence Livermore Laboratory

OPERATOR SPLITTING ON THE STAR WITHOUT TRANSPOSING

Paul F. Dubois and Garry H. Rodrigue

June 8, 1977

This is an informal report intended primarily for internal or limited external distribution. The opinions and conclusions stated are those of the author and may or may not be those of the laboratory.

Prepared for U.S. Energy Research & Development Administration under contract No. W-7405-Eng-48.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Operator Splitting on the STAR without Transposing

Paul F. Dubois
Garry H. Rodrigue

SUMMARY

A combination of parallel elimination on one sweep and cyclic reduction on the other sweep is shown to be the fastest known method on the STAR for solution of the equations arising in five point operator splitting or ADI methods for solving the diffusion equation. Cyclic reduction is explained and the results of implementing this method in BBC are given. A factor of three speed up is obtained for the matrix solution phase, since all transposes are eliminated.

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

84

1. Preliminaries

We consider the problem of numerically solving the time-dependent partial differential equation

$$(1.1) \quad \frac{d\phi}{dt} = \nabla \cdot (D \nabla \phi) - \eta \phi + S$$

where

$$\nabla \cdot (D \nabla \phi) = \frac{\partial}{\partial x} D \frac{\partial \phi}{\partial x} + \frac{\partial}{\partial y} D \frac{\partial \phi}{\partial y}.$$

Denoting the time dependence by a superscript, i.e., $\phi^n = \phi(t_n)$, and letting $\Delta t = t_{n+1} - t_n$, (1.1) can be approximated by

$$(1.2) \quad \frac{\phi^{n+1} - \phi^n}{\Delta t} = (\nabla \cdot D \nabla - \eta)(\alpha \phi^{n+1} + \beta \phi^n) + S$$

where $0 \leq \alpha \leq 1$ and $\beta = 1 - \alpha$. Let

$$(1.3) \quad M_x \phi = - \frac{\partial}{\partial x} D \frac{\partial \phi}{\partial x} + \frac{\eta}{2} \phi$$

$$(1.4) \quad M_y \phi = - \frac{\partial}{\partial y} D \frac{\partial \phi}{\partial y} + \frac{\eta}{2} \phi.$$

The method of fractional steps replaces the two-dimensional system (1.2) with two one-dimensional systems. One such way is the "splitting" method ($\alpha = \beta = 1/2$), [8],

$$(1.5) \quad \frac{\phi^{n+1/2} - \phi^n}{\Delta t} = - \frac{1}{2} M_x (\phi^{n+1/2} + \phi^n) + S/2,$$

$$(1.6) \quad \frac{\phi^{n+1} - \phi^{n+1/2}}{\Delta t} = - \frac{1}{2} M_y (\phi^{n+1} + \phi^{n+1/2}) + S/2,$$

and another is the "alternating direction" method ($\alpha = \beta = 1/2$)

$$\frac{\phi^{n+1/2} - \phi^n}{\Delta t} = - \frac{1}{2} (M_x \phi^{n+1/2} - M_y \phi^n) + S/2,$$

$$\frac{\phi^{n+1} - \phi^{n+1/2}}{\Delta t} = -\frac{1}{2} \left(M_x \phi^{n+1/2} + M_y \phi^{n+1} \right) + S/2 .$$

(The notation $\phi^{n+1/2}$ denotes an intermediate result, not an approximation at $(t_n + t_{n+1})/2$.)

In this paper, we take (1.5) - (1.6) to be the representative method as the analysis and computational timings will also apply to the alternating direction method.

We now turn our attention to the approximations of M_x and M_y . To do so, one first introduces a spatial region R on which (1.1) is to be solved. We assume $R = [a, b] \times [c, d]$. Then, let

$$a = x_0 < x_1 < \dots < x_K = b ,$$

$$c = y_0 < y_1 < \dots < y_L = d$$

so that the lines $x = x_k$ and $y = y_\ell$ define an orthogonal grid G on R . Let

$$\Delta x_k = x_k - x_{k-1} , \quad 1 \leq k \leq K$$

$$\Delta y_\ell = y_\ell - y_{\ell-1} , \quad 1 \leq \ell \leq L .$$

We use the notation for the approximation of ϕ at the zone centers

$$\phi_{k+1/2, \ell+1/2}^n = \phi(x_k + 1/2\Delta x_{k+1}, y_\ell + 1/2\Delta y_{\ell+1}, t_n) .$$

Suppressing the $\ell+1/2$ indices, $M_x \phi$ can be approximated in the interior of the region R by

$$M_x^n \phi_{k+1/2} = \frac{D_{k+1}}{\Delta x_{k+1}} \left(\phi_{k+3/2}^n - \phi_{k+1/2}^n \right) - \frac{D_k}{\Delta x_k} \left(\phi_{k+1/2}^n - \phi_{k-1/2}^n \right) .$$

Similar estimates can be obtained for those approximations of ϕ on the boundary of R .

Suppose

$$(1.7) \quad \bar{\phi}^n = \begin{bmatrix} \phi_{1/2,1/2}^n \\ \phi_{1/2,3/2}^n \\ \phi_{1/2,5/2}^n \\ \vdots \\ \phi_{1/2,L-1/2}^n \\ \phi_{3/2,1/2}^n \\ \vdots \\ \phi_{3/2,L-1/2}^n \\ \vdots \\ \phi_{K-1/2,L-1/2}^n \end{bmatrix}$$

Then (1.5) becomes

$$(1.8) \quad A_1 \bar{\phi}^{n+1/2} = B_1 \bar{\phi}^n + \bar{S}/2$$

where A_1 and B_1 are $KL \times KL$ matrices of the form

$$(1.9) \quad K \left\{ \begin{bmatrix} x & 0 & \dots & 0 & x & x & & 0 \\ 0 & x & x & & & & & \\ \vdots & & \ddots & & & & & \\ 0 & & & x & x & 0 & & \\ x & x & & & 0 & & x & \\ & & & & & x & 0 & \dots & 0 \\ & & & & 0 & & & & x \end{bmatrix} \right.$$

The matrices A_1 and B_1 are stored in the STAR by diagonals; that is, each diagonal of the matrix is defined as a vector. In this way, the calculation of the right side of (1.8) is simple (cf. [1]). Let $d_1 = B_1 \bar{\phi}^n + \bar{S}/2$. Then the remaining problem in (1.8) is to solve the system

$$(1.10) \quad A_1 \bar{\phi}^{n+1/2} = d_1.$$

Let

$$(1.11) \quad A_1 = \begin{bmatrix} & & & & 0 \\ & & & & c \\ & & & b & 0 \\ & & a & 0 & \\ 0 & & & & \end{bmatrix}$$

where a, b, c, d_1 are the vectors that define the problem (1.10). To see how one solves the system (1.10) on the STAR it is helpful to view the matrix A_1 in the following form

$$A_1 = \begin{bmatrix} S_1 & T_1 & & & \\ R_2 & S_2 & T_2 & & \\ & & & & \\ & & R_{L-1} & S_{L-1} & T_{L-1} \\ & & & R_L & S_L \end{bmatrix}$$

where S_ℓ, R_ℓ , and T_ℓ are $K \times K$ diagonal matrices. Let

$$d_1 = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_L \end{bmatrix}$$

where δ_ℓ are K-dimensional vectors. Then the solution to (1.10) is given by parallel backward-forward recursion (BF):

$$(1.12) \left\{ \begin{array}{l} W_1 = T_1 / S_1 \\ W_i = T_i / (S_i - R_i W_{i-1}) \quad , \quad 2 \leq i \leq L \quad , \\ G_1 = \delta_1 S_1 \\ G_i = (\delta_i - R_i G_{i-1}) / (S_i - R_i W_{i-1}) \quad , \quad 2 \leq i \leq L \quad , \\ Z_L = G_L \\ Z_i = G_i - W_i Z_{i+1} \quad , \quad 1 \leq i \leq L-1 \end{array} \right.$$

and

$$\bar{\phi}^{n+1/2} = \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_L \end{bmatrix}$$

Note that $\bar{\phi}^{n+1/2}$ is stored as a vector in the form (1.7). The computation of (1.12) on the STAR can be accomplished by suitable offsetting of the vectors a, b, c of (1.11). Each vector operand in (1.12) has length K. If we assume $K = L$, then a theoretical estimate of the time in STAR minor cycles given by [5] is

$$(1.12.1) \quad 8.5 K^2 + 1062 K \quad .$$

We now turn to the calculation of (1.6); i.e., the "sweep" in the y-direction. Suppressing $k+1/2$ indices, $M_y \phi$ can be approximated, in a similar manner as before

$$(1.13) \quad M_y^n \phi_{\ell+1/2} = \frac{D_{\ell+1}}{\Delta y_{\ell+1}} \left(\phi_{\ell+3/2}^n - \phi_{\ell+1/2}^n \right) - \frac{D_\ell}{\Delta y_\ell} \left(\phi_{\ell+1/2}^n - \phi_{\ell-1/2}^n \right) \quad .$$

How one calculates (1.13) on the STAR depends upon the order that the elements of $\bar{\phi}^{n+1/2}$ are stored. If they are stored in the same order as $\bar{\phi}^n$ in (1.7), then (1.6) becomes

$$(1.14) \quad A_2 \bar{\phi}^{n+1} = B_2 \bar{\phi}^{n+1/2} + \bar{S}/2$$

where A_2 and B_2 are $KL \times KL$ matrices of the form

$$(1.15) \quad K \left[\begin{array}{cccccccc} x & x & & & & & & \\ x & x & x & & & & & \\ & x & x & x & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & x & & & & \\ & & & x & x & 0 & & \\ & & & & 0 & x & x & \\ & & & & x & & \ddots & \\ & & & & & x & x & 0 \\ & & & & & & \ddots & \\ & & & & & & & 0 & x & x \\ & & & & & & & & x & \\ & & & & & & & & & 0 & x \end{array} \right]$$

If, however, the elements in $\bar{\phi}^{n+1/2}$ are rearranged; i.e., "transposed," into the following form

$$\bar{\phi}^{n+1/2} = \begin{bmatrix} \phi_{1/2,1/2}^{n+1/2} \\ \phi_{3/2,1/2}^{n+1/2} \\ \phi_{5/2,1/2}^{n+1/2} \\ \vdots \\ \phi_{K-1/2,1/2}^{n+1/2} \\ \phi_{1/2,3/2}^{n+1/2} \\ \vdots \\ \phi_{K-1/2,3/2}^{n+1/2} \\ \vdots \\ \phi_{K-1/2,L-1/2}^{n+1/2} \end{bmatrix}$$

then (1.6) becomes

$$(1.16) \quad \bar{A}_2 \bar{\phi}^{n+1} = \bar{B}_2 \bar{\phi}^{n+1/2} + \bar{S}/2$$

where $\bar{A}_2$ and $\bar{B}_2$ are $KL \times KL$ matrices having the same structure as (1.9). Hence, the solution of (1.16) can be calculated by the (BF) algorithm (1.12). However, on completion of the (BF) algorithm, the solution $\bar{\phi}^{n+1}$ must rearrange its elements to the form (1.7) so that it can be used for the calculation of the x-sweep during the next cycle of the calculation. "Transposes" of the kind mentioned above are implemented on the STAR by a call to the BLIB routine IVTRANS or on the 7600 by a call to the STACKLIB routine IVTRANS. These transposes are very expensive on the STAR as the primary vector instruction used is the Transmit Index List. If one decides not to transpose, then one must solve the system (1.14) - (1.15). This

can be accomplished effectively on the STAR by using the cyclic reduction (CR) algorithm (detailed in the next section) instead of the usual backward-forward recursion given in [2]. A theoretical estimate for the time used in performing (CR) is (assuming $K = L$)

$$(1.17) \quad 36 K^2 + 6782 \log_2 K - 601 \quad .$$

2. Cyclic Reduction

We first illustrate the algorithm for a simple 8×8 tridiagonal system; i.e., let $Tx = y$ where

$$T = \begin{bmatrix} x & x & & & & & & \\ & x & x & x & & & & \\ & & x & x & x & & & \\ & & & x & x & x & & \\ & & & & x & x & x & \\ & & & & & x & x & x \\ & & & & & & x & x & x \\ & & & & & & & x & x \end{bmatrix}$$

The first step of the algorithm is to add a multiple of an odd row to the even row below it so as to achieve the following effect:

$$\begin{array}{lcl} \text{(odd)} & x & x & x \\ \text{(even)} & \curvearrowright & x & x & x \end{array} \longrightarrow \begin{array}{lcl} & x & x & x \\ & x & 0 & x & x \end{array}$$

This is done in parallel to all of the equations. The second step is to add a multiple of an odd row to the even row above it so as to achieve the following effect:

$$\begin{array}{lcl} \text{(even)} & x & 0 & x & x \\ \text{(odd)} & \curvearrowright & x & x & x \end{array} \longrightarrow \begin{array}{lcl} & x & 0 & x & 0 & x \\ & & x & x & x \end{array}$$

The resulting linear system has the form

$$\begin{bmatrix}
 x & x & & & & & \\
 0 & x & 0 & x & & & \\
 & x & x & x & & & \\
 & x & 0 & x & 0 & x & \\
 & & x & x & x & & \\
 & & x & 0 & x & 0 & x \\
 & & & x & x & x & \\
 & & & x & 0 & x &
 \end{bmatrix}
 \begin{matrix}
 \leftarrow (1) \\
 \leftarrow (2) \\
 \leftarrow (3) \\
 \leftarrow (4)
 \end{matrix}$$

Now, in this system, if the even-numbered unknowns are determined, the odd-numbered unknowns can be easily calculated. The even unknowns can be determined by solving the smaller system of equations determined by the rows indicated by the arrows; i.e., all of the even unknowns have been coupled to each other in another tridiagonal system half the size of the original system. The above procedure is then repeated on this smaller system. This process continues until one reaches a single equation in one unknown. However, if the original tridiagonal system is given by (1.15) where the matrix is composed of $L \times K \times K$ tridiagonal matrices, then it is easily observed that cyclic reduction applied to (1.15) is actually cyclic reduction applied to the smaller $K \times K$ systems at the same time. Hence, one achieves a two-fold parallelism. Also, for this system, one does not have to carry cyclic reduction to a single equation for the smaller $K \times K$ systems. That is, on applying cyclic reduction to (1.15), one continually reduces the size of the system until the resulting system is an $L \times L$ system, and this $L \times L$ system is, in fact, a diagonal matrix which is easily inverted. For a description of how the algorithm is implemented on the STAR in a more efficient manner than described above, one should consult any of the reports [3], [4], [5] or the authors.

3. Timings

Using (1.12.1) and (1.17), Table I gives a list of the estimated times for the (BF) and (CR) algorithms where we assume $K = L$. In addition, a list of estimated times for the combined algorithms (BF)-(BF) and (BF)-(CR) for solving (1.8) - (1.16) and (1.8) - (1.14), respectively, is shown.

T A B L E I

STAR Timings in Microseconds (Theoretical)

<u>K</u>	<u>(BF)</u>	<u>(CR)</u>	<u>(BF)-(CR)</u>	<u>(BF)-(BF)</u>
50	2974	4341	7315	5958
60	3773	5961	9734	7546
70	4640	7863	12503	9280
80	5574	10050	15624	11148
90	6577	12520	19097	13154
100	7648	15277	22925	15296

The (BF) and (CR) algorithms were coded in LRLTRAN and compiled by the latest FENIX compiler (using the REGFILE and OPTIMIZE options). The transposes were executed by calls to IVTRANS. The timings for these algorithms are contained in Table II (again $K = L$).

T A B L E I I

STAR Timings in Microseconds (Actual)

<u>K</u>	<u>(BF)</u>	<u>(CR)</u>	<u>Time of 1 Transpose</u>
50	4305	7319	4346
60	5375	9798	5844
70	6528	13127	7504
80	7708	16504	9590
90	8910	20498	11550
100	10350	24748	14210

As can be seen from Table II, using (BF) for (1.8) and (CR) for (1.14) is equivalent in time to using (BF) for both (1.8) and (1.16) and a transpose. Thus, if one has to transpose more than once during a cycle of a calculation, then one is better off using the (BF)-(CR) combination.

4. Experiments with BBC

In the current version of the BBC production code, operator splitting, as described in Section 1, is currently being used in the solution of the diffusion equation, cf. [6], [7]. The splitting calculations are achieved in each cycle of the code by calls to the subroutines, which for purposes of clarity we will call SUB1, SUB2, and SUB3. Specifically, the executive routine makes the following calls:

- 1) CALL SUB1
 - computes the coefficients of the system (1.8) for the x-sweep;
- 2) CALL SUB2
 - solves the system (1.8) using the (BF) algorithm described in (1.12);
- 3) CALL SUB3
 - transpose 4 vectors of length $K \cdot L$ to prepare for y-sweep;
- 4) CALL SUB1
 - computes coefficients of the system (1.16) for the y-sweep;
- 5) CALL SUB2
 - solves system (1.16) using the (BF) algorithm described in (1.12);
- 6) CALL SUB3
 - transpose 2 vectors of length $K \cdot L$ to prepare for next cycle.

BBC was executed on the STAR to solve a "real" problem furnished by William Sutcliffe of T-Division and halted at cycle 2857. The code was then executed for 14 more cycles. During this period the execution of the above subroutines consumed the following time: (In this problem $K = 44$ and $L = 278$.)

SUB1	605993 (microseconds)
SUB2	624509
SUB3	1811212

The code was then modified so that the SUB3 calls were deleted and the system arising for the y-system is of the form (1.14) and solved by the (CR) algorithm in the SUB2 routine. Some minor changes to SUB1 were necessary. No loss of accuracy in the answers was observed. The following timings were noted for the same 14 cycles:

SUB1	583593 (microseconds)
SUB2	791704
SUB3	0

This amounts to a 300% increase in speed for the SUB2-SUB3 portion of the code. Using cyclic reduction for the y-sweep thus provides a savings of 119030 μ sec/cycle. Hence, if the code were executed for 10,000 cycles, then a savings of approximately 20 minutes on the STAR would be observed.

5. Acknowledgments

The authors would like to acknowledge helpful conversations with William Sutcliffe, Ron Litterst, Jim Kohn, and Richard Belles.

REFERENCES

- [1] Madsen, N.K., Rodrigue, G.H., Karush, J., "Matrix Multiplication by Diagonals on a Vector/Parallel Processor," Information Processing Letters, June, 1976, Vol. 5, No. 2.
- [2] Richtmyer, R., Morton, K., Difference Methods for Initial Value Problems.
- [3] Madsen, N.K., Rodrigue, G.H., "A Comparison of Direct Methods for Tri-diagonal Systems on the CDC-STAR-100," UCRL-76993, Rev. 1, May 1976.
- [4] Rodrigue, G.H., Madsen, N., Karush, J. "Odd-Even Reduction for Banded Linear Equations," UCRL-78652, January 1976.
- [5] Lambiotte, J., Voigt, R., "The Solution of Tridiagonal Systems on the CDC-STAR-100 Computer," ICASE Report, July 19, 1974.
- [6] Sutcliffe, W., "Equations for a Two-Dimensional Diffusion Routine," UCID-17012, March 1974.
- [7] Sutcliffe, W., "BBC Hydrodynamics," UCID-17013, February 1974.
- [8] Yanenko, N., "The Method of Fractional Steps," New York: Springer Verlag, 1971.

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research & Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately-owned rights.

NOTICE

Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Energy Research & Development Administration to the exclusion of others that may be suitable.

Printed in the United States of America

Available from

National Technical Information Service

U.S. Department of Commerce

5285 Port Royal Road

Springfield, VA 22161

Price: Printed Copy \$: Microfiche \$3.00

Page Range	Domestic Price	Page Range	Domestic Price
001-025	\$ 3.50	326-350	10.00
026-050	4.00	351-375	10.50
051-075	4.50	376-400	10.75
076-100	5.00	401-425	11.00
101-125	5.50	426-450	11.75
126-150	6.00	451-475	12.00
151-175	6.75	476-500	12.50
176-200	7.50	501-525	12.75
201-225	7.75	526-550	13.00
226-250	8.00	551-575	13.50
251-275	9.00	576-600	13.75
276-300	9.25	601-up	*
301-325	9.75		

* Add \$2.50 for each additional 100 page increment from 601 to 1,000 pages;
add \$4.50 for each additional 100 page increment over 1,000 pages.

Technical Information Department

LAWRENCE LIVERMORE LABORATORY

University of California | Livermore, California | 94550