JOURNAL OF COMPUTATIONAL PHYSICS 12, 422–428 (1973)

Note

Direct Methods for the Solution of Poisson's Equation on a Staggered Grid

I. Introduction

The Neumann problem for Poisson's equation on a rectangle is

$$\nabla^2 u = -u_{xx} - u_{yy} = f(x, y) \tag{1a}$$

on $R = \{(x, y) : 0 < x < 1, 0 < y < 1\}$ and

$$\partial u/\partial n = 0 \tag{1b}$$

on the boundary of R (n represents the normal derivative).

A frequent problem in fluid dynamics is to approximate (1) on a staggered grid

$$R_h = \{(x_i, y_j) : x_i = (i - \tfrac{1}{2})\, \Delta x,\ i = 0, 1, 2,..., M + 1,$$

and

$$y_j = (j - \tfrac{1}{2})\, \Delta y,\ j = 0, 1, 2,..., N + 1\},$$

where M, Δx, N, and Δy are suitably chosen so that

$$M\,\Delta x = N\,\Delta y = 1.$$

The approximation to (1a) is the usual five-point star and the derivatives in (1b) are approximated by central differences, e.g., $u_x(0, y_j) = 0$ is approximated by $(u(x_1, y_j) - u(x_0, y_j))/\Delta x = 0$. Letting the approximation to $u(x_i, y_j)$ be denoted by v_{ij}, we find that the vector v of unknowns satisfies

$$\begin{bmatrix} J-I & -I & & & 0 \\ -I & J & -I & & \\ & & \cdots & & \\ & & -I & J & -I \\ 0 & & & & J-I \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{N-1} \\ v_N \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{N-1} \\ f_N \end{bmatrix}, \tag{2}$$

where $J = 2I + (\Delta y/\Delta x)^2 L$ is an $M \times M$ matrix,

$$L = \begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & 0 \\ & & \cdots & & \\ 0 & & -1 & 2 & -1 \\ & & & -1 & 1 \end{bmatrix},$$

$$v_j = \begin{bmatrix} v_{1j} \\ v_{2j} \\ \vdots \\ v_{Mj} \end{bmatrix}, \qquad f_j = \begin{bmatrix} f_{1j} \\ f_{2j} \\ \vdots \\ f_{Mj} \end{bmatrix},$$

and

$$f_{i,j} = (\Delta y)^2 f(x_i, y_j).$$

(This same approximation is obtained with a standard grid, but using one-sided differences to approximate the derivatives instead of centered differences.)

The direct solution of this system by Gauss elimination has been discussed by Shintani [4]. We would like to discuss the solution by two faster methods: the fast Fourier transform (FFT) method and cyclic odd-even reduction with the Buneman modification.

II. Fast Fourier Transform Method

To apply the techniques of FFT we must know the eigenvalues λ_i and normalized eigenvectors w_i ($i = 1, 2,..., M$) of J. Gear [3] has computed the eigenvalues and eigenvectors of L from which we may determine that

$$\lambda_i = 2 + 4s^2 \sin^2((i-1)\pi/2M), \qquad s = \Delta y/\Delta x \tag{4}$$

and that $w_i^t = (w_{1i}, w_{2i},..., w_{Mi})$ where for $k = 1, 2,..., M$

$$w_{ki} = \begin{cases} (2/M)^{1/2} \cos[(2k-1)(i-1)\pi/2M], & i = 2, 3,..., M \\ (1/M)^{1/2}, & i = 1 \end{cases} \tag{5}$$

Hence, it is easy to see that we can apply the FFT method with considerable advantage if N is highly composite [1].

III. Buneman's Method

We now discuss the solution of Eq. (2) by cyclic reduction with the Buneman variation for numerical stability. (A complete description of this method can be found in [2].) For this development we must assume that $N = 2^{k+1} + 1$. By adding row $r - 1$ and row $r + 1$ and J times row r, half of the unknown vectors v_j are eliminated from (2) to give the new system

$$\begin{bmatrix} J(J-I)-I & -I & & & 0 \\ -I & J^2-2I & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & J^2-2I & -I \\ 0 & & & -I & J(J-I)-I \end{bmatrix} \begin{bmatrix} v_1 \\ v_3 \\ \vdots \\ v_{N-2} \\ v_N \end{bmatrix} = \begin{bmatrix} Jf_1 + f_2 \\ Jf_3 + f_2 + f_4 \\ \vdots \\ Jf_{N-2} + f_{N-3} + f_{N-1} \\ Jf_N + f_{N-1} \end{bmatrix}. \tag{6}$$

Since this coefficient matrix has the same structure as that of Eq. (2), we can repeat the reduction process and again eliminate half of the remaining unknowns. Setting

$$\begin{aligned} K^{(0)} &= J - I, \qquad & J^{(0)} &= J, \\ f_j^{(0)} &= f_j, & 1 &= 1, 2,\ldots, N, \end{aligned} \tag{7}$$

we can write the r-th step of the process as

$$\begin{bmatrix} K^{(r)} & -I & & & 0 \\ -I & J^{(r)} & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & J^{(r)} & -I \\ 0 & & & -I & K^{(r)} \end{bmatrix} \begin{bmatrix} v_1 \\ v_{1+2^r} \\ \vdots \\ v_{N-2^r} \\ v_N \end{bmatrix} = \begin{bmatrix} f_1^{(r)} \\ f_{1+2^r}^{(r)} \\ \vdots \\ f_{N-2^r}^{(r)} \\ f_N^{(r)} \end{bmatrix}, \tag{8}$$

where

$$K^{(r)} = J^{(r-1)}K^{(r-1)} - I \tag{9}$$

$$J^{(r)} = [J^{(r-1)}]^2 - 2I \tag{10}$$

and

$$f_{1+j\cdot 2^r}^{(r)} = J^{(r-1)} f_{1+j\cdot 2^r}^{(r-1)} + f_{1+j\cdot 2^{r-1}}^{(r-1)} + f_{1+3j\cdot 2^{r-1}}^{(r-1)}. \tag{11}$$

The last step is with $r = k$, after which we have only the remaining system

$$\begin{bmatrix} K^{(k)} & -I & 0 \\ -I & J^{(k)} & -I \\ 0 & -I & K^{(k)} \end{bmatrix} \begin{bmatrix} v_1 \\ v_{1+2^k} \\ v_N \end{bmatrix} = \begin{bmatrix} f_1^{(k)} \\ f_{1+2^k}^{(k)} \\ f_N^{(k)} \end{bmatrix} \tag{12}$$

from which we obtain the single linear system

$$Mv_{1+2^k} = \{K^{(k)}J^{(k)} - 2I\}\, v_{1+2^k} = K^{(k)}f^{(k)}_{1+2^k} + f_1^{(k)} + f_N^{(k)}. \tag{13}$$

To solve (13) we show that M can be factored as a product of 2^{k+1} linear factors of the form $J - \mu_i I$. To solve for the remaining unknowns we need to give the same sort of factorization for each $K^{(r)}$ and $J^{(r)}$. The factorization for $J^{(r)}$ is developed in [2]. It is

$$J^{(r)} = \prod_{j=1}^{2^r} (J - 2\cos[(2j-1)\pi/2^{r+1}]I). \tag{14}$$

From (7) and (9) we see that $K^{(r)}$ is a polynomial, say q_r, of degree 2^r in J. The polynomials q_r satisfy

$$\begin{aligned} q_0 &= x - 1 \\ q_{r+1} &= p_r q_r - 1, \qquad r = 0, 1, 2,..., k-1, \end{aligned} \tag{15}$$

where p_r is the polynomial in J which is equal to $J^{(r)}$. From [2] we know that if the substitution $x = 2\cos\theta$ is made, then

$$p_r(x) = 2\cos 2^r\theta. \tag{16}$$

Using this same substitution we find that

$$q_r = \cos(2^r + \tfrac{1}{2})\,\theta/\cos\tfrac{1}{2}\theta. \tag{17}$$

(Using (15) and (16), this statement may be easily verified by induction.) Hence, we have that

$$K^{(r)} = \prod_{j=1}^{2^r} (J - 2\cos[(2j-1)\pi/(2^{r+1}+1)]I). \tag{18}$$

The factorization of M requires finding the zeros of $q_k\, p_k - 2$. Using the representations (16) and (17) we get that

$$M = \Big[\prod_{j=1}^{2^k} (J - 2\cos(2j\pi/(2^{k+1}+1))\, I)\Big]\Big[\prod_{j=0}^{2^k-1} (J - 2\cos(j\pi/2^k)\, I)\Big]. \tag{19}$$

From (4) with $i = 1$ we note that (2) is singular. M is also singular, as can be seen from the factor corresponding to $j = 0$.)

The Buneman variant requires that the right sides $f_l^{(r)}$ defined by (11) not be computed but be defined implicitly by two auxiliary vectors $p_l^{(r)}$ and $q_l^{(r)}$ where

$$f_l^{(r)} = J^{(r)} p_l^{(r)} + q_l^{(r)}. \tag{20a}$$

In our case this factorization only applies when $l = 1 + 2^r, 1 + 2 \cdot 2^r, \dots, N - 2^r$. For the remaining ones we write

$$f_1^{(r)} = K^{(r)} p_1^{(r)} + q_1^{(r)} \tag{20b}$$

and

$$f_N^{(r)} = K^{(r)} p_N^{(r)} + q_N^{(r)}. \tag{20c}$$

Just as in [2], we can use the factorizations (20) and the defining Eq. (11) to develop implicit recurrence formulas for the $p_j^{(r)}$ and $q_j^{(r)}$. Initially they are

$$\begin{aligned}
K^{(0)} p_1^{(1)} &= f_1, & q_1^{(1)} &= f_2 + p_1^{(1)} \\
J^{(0)} p_j^{(1)} &= f_j, & q_j^{(1)} &= f_{j-1} + f_{j+1} + 2p_j^{(1)}, \qquad j = 3, 5, 7, \dots, N-2 \\
K^{(0)} p_N^{(1)} &= f_N, & q_N^{(1)} &= f_{N-1} + p_N^{(1)},
\end{aligned}$$

then for $r = 1, 2, \dots, k-1$

$$\begin{aligned}
K^{(r)}(p_1^{(r+1)} - p_1^{(r)}) &= q_1^{(r)} + p_{1+2^r}^{(r)} & q_1^{(r+1)} &= q_{1+2^r}^{(r)} + p_1^{(r+1)}, \\
J^{(r)}(p_j^{(r+1)} - p_j^{(r)}) &= p_{j-2^r}^{(r)} + p_{j+2^r}^{(r)} + q_j^{(r)} & q_j^{(r+1)} &= q_{j-2^r}^{(r)} + q_{j+2^r}^{(r)} + 2p_j^{(r+1)}, \\
& j = 1 + 2^{r+1}, 1 + 2 \cdot 2^{r+1}, 1 + 3 \cdot 2^{r+1}, \dots, N - 2^{r+1}, \\
K^{(r)}(p_N^{(r+1)} - p_N^{(r)}) &= q_N^{(r)} + p_{N-2^r}^{(r)} & q_N^{(r+1)} &= q_{N-2^r}^{(r)} + p_N^{(r+1)},
\end{aligned}$$

and finally $p_{1+2^k}^{(k+1)}$ and $q_{1+2^k}^{(k+1)}$ are obtained from above with $r = k$ and $j = 1 + 2^k$.

The solution of (2) is then obtained by solving the following equations:

$$\begin{aligned}
M(v_{1+2^k} - p_{1+2^k}^{(k+1)}) &= q_{1+2^k}^{(k+1)}, \\
K^{(k)}(v_j - p_j^{(k)}) &= q_j^{(k)} + v_{1+2^k}, \qquad j = 1, N;
\end{aligned} \tag{21}$$

then for $r = k-1, k-2, \dots, 0$

$$J^{(r)}(v_j - p_j^{(r)}) = q_j^{(r)} + v_{j-2^r} + v_{j+2^r},$$

for $j = 1 + 2^r, 1 + 3 \cdot 2^r, 1 + 5 \cdot 2^r,..., N - 2^r$, where

$$p^{(0)}_j = 0 \qquad \text{and} \quad q_j^{(0)} = f_j .$$

As we have seen, (21) is a singular system. The right side of (21) must be orthogonal to $(1, 1,..., 1)^t$. It can be proved that this condition is a consequence of the original orthogonality condition

$$\sum_{j=1}^{N} \sum_{i=1}^{M} f_{ij} = 0$$

which arises from the condition that the right side of (2) must be orthogonal to $(1, 1, 1,..., 1)^t$.

As in [2], we can reduce the storage requirements by half by not storing the vectors $p_i^{(r)}$, but using instead their definitions in terms of the vectors $q_i^{(r)}$.

IV. Comparison of the Two Algorithms

An operation count quickly shows that both algorithms require about $4N^2 \log_2 N$ (we have taken $N = M$) multiplications, so they appear to be about equal. In fact, numerical experiments on the NCAR CDC 6600 using $M = N = 64$ have shown that the Buneman algorithm takes about 710 msec to compute the solution while the FFT algorithm requires about 1870 msec. The FFT algorithm can be speeded up by eliminating the two reorderings required since they are inverses of each other. However, the eigenvalues then must be reordered so at most one reordering can be eliminated. Subtracting the time required for 64 reorderings in the above example still leaves a running time for the FFT routine of 1490 msec. So the Buneman algorithm is at least twice as fast as the FFT routine. This may be accounted for by realizing that the operation counts only agree in the highest order terms and a more careful count gives $4N^2 \log_2 N + \frac{1}{2}N^2$ for the Buneman algorithm, but $4N^2 \log_2 N + 14N^2$ for the FFT algorithm.

It appears that even though the Buneman algorithm is more complicated, it is superior to the FFT algorithm.

References

1. G. D. Bergland, A fast Fourier transform algorithm for real-valued series, *Comm. ACM* **11** (1968), 710.

2. B. L. Buzbee, G. H. Golub, and C. W. Nielson, "Direct methods for solving Poisson's equation, *SIAM J. Numer. Anal.* **7** (1970), 656.
3. C. W. Gear, A simple set of test matrices for eigenvalue problems, *Math. Comp.*, **23** (1969), 125.
4. H. Shintani, Direct solution of partial difference equations for a rectangle, *J. Sci. Hiroshima Univ. Ser. A-I Math.* **32** (1968), 53.

Received: July 11, 1972.

Roland A. Sweet

Natural and Physical Sciences Division,
University of Colorado,
Denver, Colorado, 80202
and
*National Center for Atmospheric Research**
Boulder, Colorado 80302

* The National Center for Atmospheric Research is sponsored by the National Science Foundation.