JOURNAL OF COMPUTATIONAL PHYSICS 33, 204–216 (1979)

Point Cyclic Reductions for Elliptic Boundary-Value Problems. I. The Constant-Coefficient Case

E. DETYNA

Computer Science Department, The University of Reading, Whiteknights Park, Reading RG6 2AX England

Received August 22, 1978; revised December 28, 1978

A new fast "N^2-algorithm" has been developed for solving two-dimensional elliptical partial differential equations. It employs point cylic reductions which reduce the number of equations by a factor of 4 at each step. In this first part the constant-coefficient case with Dirichlet boundary conditions is considered.

1. INTRODUCTION

We shall consider in this paper the numerical solution of constant-coefficient elliptic boundary-value problems with Dirichlet boundary conditions on square domains. Without loss of generality we restrict attention to the equation

$$(\nabla^2 + u)\phi(x, y) = \rho(x, y), \tag{1.1}$$

where u is a constant. Discretizing (1.1), using uniform finite differences, leads to a sparse linear system of equations of the form $Ma = b$, where M is $N^2 \times N^2$ matrix with only a few nonvanishing diagonals. The sparsity of the matrix M has been exploited in various ways to produce a fast solution $a = M^{-1}b$. These are known as "fast elliptical solvers." The earliest methods used either the fast Fourier transforms as in Hockney [1] or global cyclic reductions [2]. These algorithms required $O(N^2 \log_2 N)$ arithmetic operations to solve (1.1) on $N \times N$ mesh. Later these two algorithms were combined [3, 4] into an algorithm known as FACR(l) which requires only $O(N^2 \log_2 \log_2 N)$ arithmetic operations. Quite a different approach was used by Lorenz [5], who developed a marching algorithm, which although unstable gives very fast results. The operational count is $(10 + (4\log_2 N + 10)/D)\, N^2$, where D is the number of decimal digits lost in calculations. In certain computers with a very long word length, D may be allowed to be very large, thus resulting in the so-called "N^2-algorithm." An excellent review by Hockney of all fast elliptical solvers is available [6].

The aim of this paper is to present a point cyclic reduction method which is an "N^2-algorithm." In Sections 2 and 3, the basic idea of PCR is first presented and in Section 4, it is shown how it can be applied to Eq. (1.1).

0021-9991/79/110204–13$02.00/0

It is shown that the number of operations needed may be as little as $9.5N^2$ for $N \times N$ grid.

2. Point Cyclic Reduction

2.1. *Statement of the Problem*

Let $G^{(0)}$ be a square grid:

$$G^{(0)} = \{(x_i, y_j): x_i = ih, y_j = jh; i, j = 0, 1, 2,..., N = 2^M\}.$$

Let there be three functions ϕ, ρ^+, and $\rho^\times$, defined on that grid, such that they satisfy the following two sets of finite-difference equations on every point P inside the boundary:

$$\phi_{i-1,j} + \phi_{i+1,j} + \phi_{i,j-1} + \phi_{i,j+1} + u^+\phi_{i,j} = \rho^+_{ij}, \tag{2.1a}$$

$$\phi_{i-1,j-1} + \phi_{i+1,j-1} + \phi_{i-1,j+1} + \phi_{i+1,j+1} + u^\times\phi_{i,j} = \rho^\times_{ij}, \tag{2.1b}$$

where $f_{ij} \equiv f(P) \equiv f(x_i, y_j)$; u^+ and $u^\times$ are constants. The value of ϕ on the boundary is zero.

It should be stressed that functions ρ^+, $\rho^\times$ and constants u^+, $u^\times$ are such that the solutions of either sets of Eqs. (2.1a) *or* (2.1b) *gives identical value for ϕ.*

In other words we have twice as many equations as unknowns; therefore instead of solving one of the sets of Eqs. (2.1a) or (2.1b) one may choose some of the equations from set (2.1a) and some from (2.1b) and obtain the solution ϕ. That, in fact, is the basis of the PCR method.

2.2. *Notation*

Let $G^{(1)}$, $G^{(2)}$,..., $G^{(M-1)}$ be a series of subgrids of the grid $G^{(0)}$:

$$G^{(n)} = \{(x_i, y_j): x_i = ih, y_j = jh; i, j = 0, 2^n, 2 \cdot 2^n,..., 2^M\}.$$

The last subgrid in this series, $G^{(M-1)}$, consists of eight points on the boundary and one point in the center only.

Let us now define two difference operators, S^{n+} and $S^{n\times}$, which will simplify the manipulation of Eqs. (2.1):

$$\begin{aligned} S^{n+} &= E_x^{\,n} + E_y^{\,n} + E_x^{-n} + E_y^{-n}, \\ S^{n\times} &= (E_x^{\,n} + E_x^{-n}) \cdot (E_y^{\,n} + E_y^{-n}), \end{aligned} \tag{2.2}$$

where E_x and E_y are the shift operators in the x- and y-directions, respectively,

$$E_x^{\,n}E_y^{\,m}f(x, y) = f(x + nh, y + mh),$$

for any function $f(x, y)$. These operators have a simple visual representation: $S^{n+}f(P)$ or $S^{n\times}f(P)$ is the sum over "$+$ star" or "$\times$ star" of values of f around point P removed by n grid points from P.

The squares of these operators are again expressible in terms of the same operators:

$$\begin{aligned}(S^{n+})^2 &= S^{2n+} + 2S^{n\times} + 4,\\ (S^{n\times})^2 &= S^{2n\times} + 2S^{2n+} + 4.\end{aligned} \tag{2.3}$$

2.3. *One Step Reduction*

With the aid of the S operators, the original Eqs. (2.1) may be written as

$$\begin{aligned}(S^{+} + u^{+})\,\phi(P) &= \rho^{+}(P),\\ (S^{\times} + u^{\times})\,\phi(P) &= \rho^{\times}(P).\end{aligned} \tag{2.4}$$

Multiplying them by $S^{+} - u^{+}$ and $S^{\times} - u^{\times}$, respectively, gives

$$\begin{aligned}(S^{2+} + u^{2+})\phi &= \rho^{2+},\\ (S^{2\times} + u^{2\times})\phi &= \rho^{2\times},\end{aligned} \tag{2.5}$$

where

$$\begin{aligned}u^{2+} &= 4 - 2u^{\times} - (u^{+})^2,\\ u^{2\times} &= 4 - 2u^{2+} - (u^{\times})^2,\end{aligned} \tag{2.6}$$

and

$$\begin{aligned}\rho^{2+} &= (S^{+} - u^{+})\,\rho^{+} - 2\rho^{\times},\\ \rho^{2\times} &= (S^{\times} - u^{\times})\,\rho^{\times} - 2\rho^{2+}.\end{aligned} \tag{2.7}$$

It is seen that the structure of Eqs. (2.5) is identical to that of the original Eqs. (2.4), except that now they are on the coarser subgrid $G^{(1)}$, since the operators S^2 instead of S act on ϕ.

The R.H.S. of Eqs. (2.5), the reduced densities $\rho^{2+}(P)$ and $\rho^{2\times}(P)$, are linear combinations of ten values of ρ^{+} and $\rho^{\times}$ surrounding point P.

The new parameters u^{2+} and $u^{2\times}$, Eqs. (2.6), are also constants and play the same role as u^{+} and $u^{\times}$ in Eqs. (2.4).

Hence the new Eq. (2.5) have the same structure *and* give the same results as Eqs. (2.4) but on the coarser grid $G^{(1)}$, which consists of about one-fourth the number of points of the $G^{(0)}$ grid.

It is easy to see that this cyclic reduction can be repeated again leading to equations on the still coarser grid $G^{(2)}$. Hence after performing PCR k times one gets the following equations:

$$(S^{2^k+} + u^{2^k+})\,\phi(P) = \rho^{2^k+}(P), \tag{2.8a}$$

$$(S^{2^k\times} + u^{2^k\times})\,\phi(P) = \rho^{2^k\times}(P), \tag{2.8b}$$

where $P \in G^{(k)}$, constants u and functions ρ are defined recursively:

$$u^{2^k+} = 4 - 2u^{2^{k-1}\times} - (u^{2^{k-1}+})^2,$$
$$u^{2^k\times} = 4 - 2u^{2^k+} - (u^{2^{k-1}\times})^2, \tag{2.9}$$

and

$$\rho^{2^k+} = (S^{2^{k-1}+} - u^{2^{k-1}+})\,\rho^{2^{k-1}+} - 2\rho^{2^{k-1}\times},$$
$$\rho^{2^k\times} = (S^{2^{k-1}\times} - u^{2^{k-1}\times})\,\rho^{2^{k-1}\times} - 2\rho^{2^k+}, \tag{2.10}$$

for $k = 1, 2, \ldots, M-1$.

Thus now we have a separate set of Eqs. (2.8) on each grid $G^{(l)}$, for $l = 0, 1, 2, \ldots, M-1$, each giving the same solution, $\phi(P)$.

This completes the point cyclic reductions of Eqs. (2.1).

3. The Solution

3.1. *The Central Point Solution*

It was noted earlier that the subgrid $G^{(M-1)}$ consists of only one point C in the center and boundary points. It was also assumed that ϕ is zero on the boundary, hence

$$S^{2^{M-1}+,\times}\phi(C) = 0 \tag{3.1}$$

since this S operator sums the values of ϕ on the boundary alone.

With the aid of this result it is seen that either of Eqs. (2.8), for $k = M-1$, gives the value $\phi(C)$ at the center of the grid.

Therefore we may take a linear combination of these equations to give a solution:

$$\phi(C) = \frac{\alpha\rho^{2^{M-1}+}(C) + (1-\alpha)\,\rho^{2^{M-1}\times}(C)}{\alpha u^{2^{M-1}+} + (1-\alpha)\,u^{2^{M-1}\times}}, \tag{3.2}$$

where α is an arbitrary constant.

The result $\phi(C)$ does not depend on the value of α.

3.2. *Solution of the Reduced Equation*

Let us now turn our attention to the subgrid $G^{(M-2)}$ and Eqs. (2.8) on that grid ($k = M-2$). It is seen that $S^{2^{M-2}\times}\phi(P^\times)$ (see Fig. 1 for definition of $P^\times$ points) involves only values of $\phi(C)$ which is also known. Hence the solution for ϕ at $P^\times$ points on $G^{(M-2)}$ subgrid is

$$\phi(P^\times) = (\rho^{K\times}(P^\times) - S^{K\times}\phi(P^\times))/u^{K\times}, \tag{3.3}$$

where $K = 2^{M-2}$.

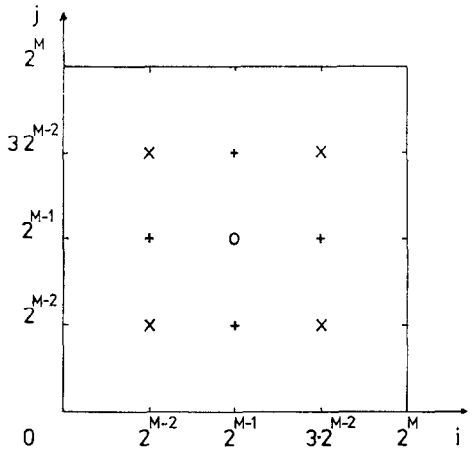

FIG. 1. The positions of $P^{\times}$ (marked $\times$), P^{+} (marked $+$) points and the centre point C (marked 0) on the subgrid $G^{(M-1)}$.

Having done this the solution on the remaining P^{+} points on the $G^{(M-2)}$ subgrid can be obtained from Eq. (2.8a):

$$\phi(P^{+}) = (\rho^{K+}(P^{+}) - S^{K+}\phi(P^{+}))/u^{K+}, \tag{3.4}$$

where, again, $K = 2^{M-2}$. This is possible since $S^{2^{M-2}+}\phi(P^{+})$ is a sum over known values of ϕ at the boundary, $P^{\times}$ points, or the central point C only, (see Fig. 1).

This completes the calculation of ϕ on the subgrid $G^{(M-1)}$.

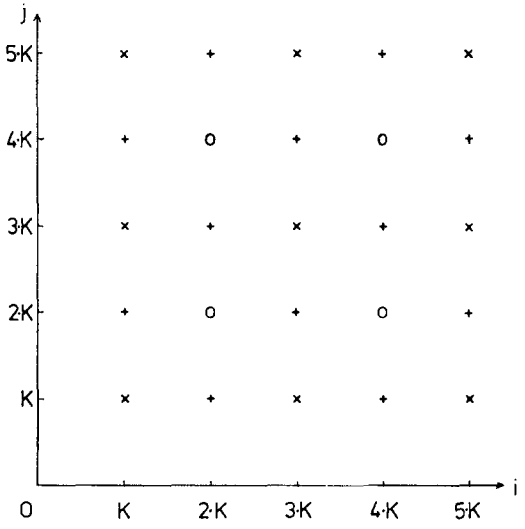

FIG. 2. The positions of $P^{\times}$ and P^{+} points on the $G^{(l)}$ subgrids; $K = 2^{l}$. The values of ϕ on the $G^{(l+1)}$ (marked 0) are known.

The next step is to calculate the solution ϕ on the finer grid $G^{(M-3)}$: using Eq. (3.3) for $K = 2^{M-3}$, the values of $\phi(P^{\times})$ are calculated first and then using Eq. (3.4), $\phi(P^{+})$ are found.

This process is repeated for all subgrids (see Fig. 2 for arbitrary K) until all values of ϕ are found on the original grid $G^{(0)}$.

In the last step, with $K = 2^0 = 1$, the original unreduced Eqs. (2.4) are used to obtain the solution.

3.2.1. *Extension to the Rectangular Region*

Let the grid $G^{(0)}$ be rectangular:

$$G^{(0)} = \{(x_i, y_j): x_i = ih, y_j = jh; i = 0, 1, 2,..., N_x = 2^{M_x}, j = 0, 1, 2,..., N_y = 2^{M_y}\},$$

where, say, $N_x > N_y$.

The series of subgrids $G^{(1)},..., G^{(M_y-1)}$ can be constructed in the same manner as for the $N_x = N_y$ case, but now the last subgrid consists of boundary points and $2^{M_x-M_y}$ central points $C_l = (2^{M_y-1}lh, 2^{M_y-1}h)$.

After performing $k = M_y - 1$ PCR's we get Eqs. (2.8) on the $G^{(M_y-1)}$ subgrid. Taking a linear combination of (2.8a) and (2.8b) gives

$$\phi(C_{l-1}) + \lambda\phi(C_l) + \phi(C_{l+1}) = \rho^{\lambda}(C_l), \tag{3.5}$$

where

$$\lambda = \alpha u^{K+} + (1 - \alpha)\, u^{K\times},$$

$$\rho^{\lambda}(C_l) = \alpha\rho^{K+}(C_l) + (1 - \alpha)\, \rho^{K\times}(C_l),$$

$K = 2^{M_y-1}$ and α is an arbitrary constant.

It is seen that Eq. (3.5) is a three-point one-dimensional finite-difference equation and can be solved by Gaussian elimination, cyclic reduction, or any number of other suitable methods.

Once the solution at central points C_l is obtained, the rest of the reduced Eqs. (2.8) can be solved in the same manner as for the square grid.

3.2.2. *The Inhomogeneous Boundary Conditions*

If the boundary values of ϕ are not zero, the L.H.S. of Eq. (3.1) is a nonvanishing constant, hence solution (3.2) should be modified in an obvious way. Similarly, the R.H.S. of Eq. (3.5) should be modified: the ρ^K's are replaced by $\rho^K - S^K\phi$.

3.3. *Operational Count*

In order to obtain the solution of Eqs. (2.4), first the series of reduced densities ρ^{2^l+} and $\rho^{2^l\times}$ are calculated on subgrids $G^{(l)}$ and then solutions are found using Eq. (3.3) or (3.4).

It is seen from Eqs. (2.10) that one requires five additions and two multiplications in order to calculate the reduced density at one point. It is also noted that in order to calculate $\rho^{n\times}$ reduced density, first the reduced ρ^{n+} density is required. On the other hand, it is seen from Fig. 2, that $\rho^{n\times}$ reduced densities are required on every second point of a subgrid. Therefore, the number of points at which the ρ^{n+} and $\rho^{n\times}$ reduced densities are calculated at each subgrid $G^{(l)}$ is

$$m_l^{+} = (2^{M-l} - 1)^2$$

and

$$m_l^{\times} = \tfrac{1}{2}(1 + (2^{M-l} - 1)^2),$$

respectively. Summing them over all the subgrids gives

$$\sum_{l=1}^{M-1} (m_l^{+} + m_l^{\times}) = \tfrac{1}{2}(N^2 - 6N + 4) + 2\log_2 N \lesssim \tfrac{1}{2}N^2.$$

Since the calculation of each reduced density requires the same number of operations, the PCR is completed in $2.5N^2$ additions and N^2 multiplications.

The solution stage, as can be seen from Eqs. (3.3) and (3.4) requires four additions and one division for each point on the grid.

To sum it up: in order to calculate a solution of Eqs. (2.1) on the square grid $N \times N$, the PCR method requires $6.5\,N^2$ additions and $2N^2$ multiplications/divisions.

Therefore, PCR can be called an "N^2-algorithm" with $8.5N^2$ floating point operations.

3.4. *The Stability*

The above-described solution of Eqs. (2.1) is exact, but if calculated on a computer, the round-off error may accumulate.

If we denote the round-off error in ρ's at the kth level of PCR by $\delta\rho$, the *maximum* error in ρ's at the $(k+1)$st level may be estimated from Eq. (2.10):

$$\begin{bmatrix} \delta\rho^{2^{k+1}+} \\ \delta\rho^{2^{k+1}\times} \end{bmatrix} = \mathbf{A}^{(k)} \begin{bmatrix} \delta\rho^{2^{k}+} \\ \delta\rho^{2^{k}\times} \end{bmatrix}, \tag{3.6}$$

where

$$\mathbf{A}^{(k)} = \begin{vmatrix} 4 + |u^{2^{k-1}+}|, & -2 \\ -8 - 2|u^{2^{k-1}+}|, & 8 + |u^{2^{k-1}\times}| \end{vmatrix}.$$

Therefore the round-off error in ρ's at the kth level is proportional to the product

$$\epsilon^{(k)} = \prod_{l=0}^{k-1} \mathbf{A}^{(l)}.$$

The solution, on the other hand, is proportional to u^{2^k} at the kth level, see Eqs. (3.3)

and (3.4). Therefore the maximum error of ϕ is calculated at the kth level as

$$(u^{2^k+})^{-1}(\epsilon_{11}^{(k)}\delta\rho^+ + \epsilon_{12}^{(k)}\delta\rho^\times)$$

or (3.7)

$$(u^{2^k\times})^{-1}(\epsilon_{21}^{(k)}\delta\rho^+ + \epsilon_{22}^{(k)}\delta\rho^\times)$$

depending on whether the solution is calculated at P^+ or $P^\times$ points.

In a special case, $u^+ = u^\times = -4$, the round-off error can be calculated exactly, since from Eqs. (2.9) we see that $u^{2^k+} = u^{2^k\times} = -4$ for any k. Therefore the matrix $\mathbf{A}^{(k)}$ is independent of k and equals

$$\mathbf{A} = \mathbf{A}^{(k)} = \begin{vmatrix} 8, & -2 \\ -16, & 12 \end{vmatrix}.$$

On solving $\det | \mathbf{A} - \lambda\mathbf{I} | = 0$ one gets the largest eigenvalue $\lambda = 16$. Since the solution in that case does not grow, the error is proportional to $\lambda^{M-1} = (\frac{1}{4}N^2)^2$, or the square of the number of grid points.

In order to establish the round-off error, in practice, the following procedure was adopted: uniformly random numbers were chosen for ϕ_{ij}'s and then ρ_{ij}^+'s and $\rho_{ij}^\times$'s were calculated from Eqs. (2.1). Using these ρ's the solution ϕ_{ij}^* was obtained on the CDC 7600 computer with the aid of the PCR algorithm. Since ϕ_{ij}'s are random numbers, the difference $\delta\phi_{ij} = \phi_{ij} - \phi_{ij}^*$ is due to round-off error alone. The results are shown in Table I. It may be estimated from the first column that $\lambda = 13.17$ rather than 16 as calculated from error analysis. It is also seen that for $u^+ < -4$, the error growth is very slow indeed. This is due to the rapid growth of u^{2^k} as a function of k: for $u^+ \ll -4$, $u^{2^k} \approx -\frac{1}{2}(u^k)^2$. Since the error in Eqs. (3.7) is divided by u^k, its growth must be inhibited.

TABLE I

The average round-off error ϵ as a function of mesh size and constant u^+.[a]

	$u^+ = -4$		$u^+ = -4.1$		$u^+ = -4.2$	
Mesh size	ϵ	D	ϵ	D	ϵ	D
8×8	1.7×10^{-14}	0	1.5×10^{-14}	0	1.1×10^{-14}	0
16×16	1.4×10^{-13}	1	5.0×10^{-14}	1	4.6×10^{-14}	1
32×32	1.8×10^{-12}	2	1.5×10^{-13}	1	8.8×10^{-14}	1
64×64	2.5×10^{-11}	4	2.5×10^{-13}	2	1.1×10^{-13}	1
128×128	3.2×10^{-10}	5	3.6×10^{-13}	2	1.3×10^{-13}	1

[a] In the second column the number of digits D lost in calculation is shown.

4. Applications of PCR to Elliptical Equations

In this section, we shall show how the above-described PCR method can be applied to the problem of numerical solutions of constant-coefficient elliptic boundary-value problems.

We restrict our attention to the elliptical equation (1.1):

$$(\nabla^2 + u)\phi(x, y) = \rho(x, y)$$

on a rectangular domain with Dirichlet boundary conditions.

There are several ways of discretizing Eq. (1.1) on a regular grid [1], the simplest two being the "five-point" formula and the "rotated five-point" formula.

These formulas can be written symbolically as an approximation to the ∇^2 operator:

$$\nabla^2 \cong (1/h^2)(S^+ - 4), \tag{4.1}$$

$$\nabla^2 \cong (1/h^2)(S^\times - 4)/(1 + \tfrac{1}{4}S^+), \tag{4.2}$$

where the inverse of an operator means that the R.H.S. of an equation should be multiplied by it; h is the spacing of the grid.

Unfortunately, although a formula similar to that of (4.1) exists for $h_x \neq h_y$, there is no equivalent to the "rotated five-point" formula. Therefore, this method is limited to discretization with identical spacing in x- and y-directions, $h_x = h_y = h$ only.

After some algebra, Eq. (1.1) can be discretised with the aid of (4.1) and (4.2) giving

$$(S^+ + u^+)\phi = \rho^+,$$

$$(S^\times + u^\times)\phi = \rho^\times,$$

where

$$\begin{aligned} u^+ &= -4 + uh^2, \\ u^\times &= -\tfrac{1}{4}(u^+)^2, \end{aligned} \tag{4.3}$$

and

$$\begin{aligned} \rho^+ &= h^2\rho, \\ \rho^\times &= \tfrac{1}{4}(S^+ - u^+)\,\rho^+. \end{aligned} \tag{4.4}$$

Therefore, we can now proceed to solve them using the PCR method, thus obtaining obtaining an approximate solution of Eq. (1.1).

Let us now calculate the operational count of this procedure. First we note that $\rho^\times$ needs to be calculated on $\frac{1}{2}N^2$ points of the grid, each calculation requiring six floating point operations.

Second, it is seen from Eqs. (2.7) and (4.4) that now

$$\rho^{2+} = 2\rho^{\times},$$

hence the first + cyclic reduction need not be calculated. For the same reason the first × cyclic reduction requires only five operations.

The following table recapitulates the whole operational count of floating point calculations:

PCR on all subgrids (2.10)	$3\,5N^2$
Solution on all subgrids (3.3), (3.4)	$5N^2$
Calculation of $\rho^{\times}$ (4.4)	$3N^2$
Less calculations omitted at the first PCR	$\frac{7}{4}N^2 + \frac{8}{2}N^2$
Total number of F.P.O.'s	$9.5N^2$

This number does not include the data preparations (multiplication of ρ by h^2) and calculation of u^{2*}, the latter being $O(\log_2 N)$.

Storage requirements. A Fortran program incorporating the PCR method requires $\frac{9}{8}N^2$ words of memory. At first an array R_{ij} stores ρ_{ij}^{+}, $\rho_{ij}^{\times}$ and $\rho_{ij}^{2+} = 2\rho_{ij}^{\times}$ at mesh points marked +, ×, and 0 respectively on Fig. 2 ($K = 1$). In order to execute higher PCR's an additional array of the length $\frac{1}{8}N^2$ is needed to store $\rho_{ij}^{2\times}$ for $i, j = 2, 6, 10,...$ and $i, j = 4, 8, 12,...$

During the solution stage, the array R_{ij} is replaced by the solution ϕ_{ij}.

Numerical results. A Fortran program incorporating the PCR method was written and tested on the CDC 7600 computer with FTN 4.6 compiler, OPT = 2.

Typical results of accuracy of the solution and CPU times are shown in Table II. The function ϕ used in Eq. (1.1) was $\phi = 0.5x(x-1)y(y-1) + 0.5x^2(x^2-1)y^2(y^2-1)$ with $\rho(x, y) = \nabla^2\phi + u\phi$. A solution was obtained on a rectangle domain $(0, 1) \times (0, N_y/N_x)$, where $N_y \leqslant N_x = 2^M1$, with grid spacing $h = 1/N_x$.

It is seen from Table II that CPU time is indeed proportional to the number of grid points. The reason why the rectangular grids are calculated slightly faster than the square grids with the same number of points is because they have fewer points inside the boundary.

4.1. *Error Analysis*

It was shown in Section 3.5 how the round-off error affects a solution on the computer. Apart from the discretization error in (4.1) and (4.2), which is $O(h^2)$, the solution is subjected to an error due to *the difference* in the two methods of discretization.

It can be shown with the aid of Eqs. (2.4) that ρ^{+} and $\rho^{\times}$ should satisfy the following identity:

$$(S^{+} + u^{+})\,\rho^{\times} = (S^{\times} + u^{\times})\,\rho^{+}. \tag{4.5}$$

TABLE II

Relative errors in approximating the solution of Eq. (1.1) and CDC 7600 CPU times as a function of grid size and constant u

Grid size	Maximum[a] error	Average[a] error	CPU time (sec)
8×8	1.3×10^{-3} 9.6×10^{-4}	4.3×10^{-4} 3.2×10^{-4}	0.00024
16×16	1.3×10^{-3} 9.6×10^{-4}	2.5×10^{-4} 2.0×10^{-4}	0.00062
32×32	1.1×10^{-3} 9.1×10^{-4}	2.1×10^{-4} 1.5×10^{-4}	0.0022
64×16	9.8×10^{-6} 9.6×10^{-6}	7.2×10^{-7} 5.2×10^{-7}	0.0020
64×64	1.1×10^{-3} 9.1×10^{-4}	1.8×10^{-4} 1.1×10^{-4}	0.0083
256×16	1.6×10^{-8} 1.6×10^{-8}	3.5×10^{-9} 3.5×10^{-9}	0.0075
128×128	9.6×10^{-4} 8.8×10^{-4}	1.3×10^{-4} 9.8×10^{-5}	0.033
1024×16	1.6×10^{-10} 1.6×10^{-10}	3.5×10^{-11} 3.5×10^{-11}	0.030

[a] Upper figures in each line refer to $u = 0$ and lower to $u = -5$.

On the other hand, the approximations (4.4) do not, in general, satisfy (4.5). Therefore, the two discretization errors in (4.4), $\delta\rho_D{}^+$ and $\delta\rho_D{}^\times$ will grow in the same manner as round-off errors described in Section 3.5, and their analysis is not repreated here. This error is responsible for the lower accuracy for large $N \times N$ grids in Table II than can be expected from discretization error $O(h^2) = O(1/N^2)$ alone.

In order to reduce this error, a more accurate approximation than (4.4) may be adopted, but this is outside the scope of this paper.

An alternative approach is to perform k PCR's; the resulting equations on the subgrid $G^{(k)}$ can then be solved by some other stable fast method. Let us for example consider the global reduction method [2] with operational count

$$6N^2 \log_2 N$$

or Hockney's POT1 [1] method with operational count

$$2.5N^2(\log_2 N + 2.4)$$

for the $N \times N$ grid.

If these methods are combined with PCR they will result in algorithms with operational counts equal:

$$N^2\left(9.5 + \frac{6\log_2 N - 6k - 8.5}{4^k}\right),$$

$$N^2\left(9.5 + \frac{2.5\log_2 N - 2.5k - 1.5}{4^k}\right),$$

respectively.

Assuming $N = 128$, the following table of operational counts per grid point can be compiled

k		0	1	2	⋯	7
Op. count	k PCR + GCR	43	16.4	10.8	⋯	9.5
	k PCR + POT1	23.5	12.9	10.2	⋯	9.5

The first column ($k = 0$) refers to the operational count of GCR or POT1 alone, whereas the last ($k = 7$) is for the PCR method on its own. But it can be seen that even if only one or two PCR's are calculated first, the operational count is dramatically lower than for GCR or POT1 methods on their own. It is also worth pointing out that if we first perform, say, three PCR's and then use an $O(N^2 \log_2 N)$ method, the resulting algorithm has roughly the same operational count

$$N^2\left(9.5 + \frac{O(\log_2 N - 3) - 8.5}{4^3}\right) \cong 10N^2$$

regardless of the method used.

If an $O(N^2 \log_2 N)$ method is numerically stable, then the resulting algorithm should give more accurate results than the PCR method on its own since it was pointed out in Section 3.4 that the error grows with the number of PCR's performed.

As an example, Hockney's POT1 program was combined with the PCR method to solve $\nabla^2\phi = \rho$ equation with the same ϕ as used in preparing Table II.

The results are shown in Table III. It is seen that performing two PCR's results in the maximum error about an order of magnitude worse than the error in the exact Hockney's solution, but an order of magnitude better than the PCR method alone.

TABLE III

Maximum errors in approximating the solution of Eq. (1.1) as a function of grid size and number of PCR's performed before Hockney's POT1 program is used.

Grid size	POT 1	1 PCR + POT 1	2 PCR's + POT 1
16×16	9.8×10^{-5}	8.0×10^{-4}	—
32×32	2.5×10^{-5}	1.9×10^{-4}	4.4×10^{-4}
64×64	6.1×10^{-6}	4.6×10^{-5}	1.0×10^{-4}
128×128	1.5×10^{-6}	1.1×10^{-5}	2.4×10^{-5}

The errors in Table III could be further reduced by using the "nine-point" formula in Hockney's method, thus incorporating both $\rho_{ij}^{2k\times}$ and ρ_{ij}^{2k+} to calculate the final solution.

ACKNOWLEDGMENT

I wish to express my appreciation for the encouragement and helpful comments as well as their patience with S^+ and S^+ operators to Professor R. W. Hockney, Dr. J. W. Eastwood, Dr. R. Berman, Dr. C. Jesshope and Dr. D. R. K. Brownrigg. I also wish to thank my wife Liz and Ms Jill Dickinson for correcting spelling errors and immaculate typing respectively. Financial support was provided by SRC.

REFERENCES

1. R. W. HOCKNEY, "Methods in Computational Physics," pp. 131–211, Academic Press, New York, 1969.
2. B. L. BUZBEE *et al.*, *SIAM J. Numer. Anal.* **7** (1970), 627–656.
3. P. N. SWARZTRAUBER, *SIAM Rev.* **19** (1977), 490–501.
4. C. TEMPERTON, Report 14, European Centre for Medium Range Weather Forecast, Bracknell, Berks., U.K., 1977.
5. E. N. LORENZ, *Monthly Weather Rev.* **104** (1976), 961–966.
6. R. W. HOCKNEY, Computer Science Departmental Report RCS 94, Reading University, Reading, U.K., 1978.