On the Solution of Poisson's Equation on a Regular Hexagonal Grid Using FFT Methods

W. M. PICKERING

Department of Applied and Computational Mathematics, The University, Sheffield S10 2TN, England

Received August 16, 1984; revised May 15, 1985

An FFT method for solving the discrete Poisson equation on a rectangle using a regular hexagonal grid is described and the results obtained for a model Dirichlet problem are compared with those obtained on a rectangular grid. For a given grid size the results demonstrate that the hexagonal method is more accurate, but rather less efficient, than the usual 5-point method, whereas for comparable accuracy to be achieved by both methods, the hexagonal method was found to be approximately 20 to 30 times faster than the 5-point method for the model problem. © 1986 Academic Press, Inc.

1. INTRODUCTION

Fast, direct methods for solving a certain class of linear partial differential equations are well established and there is a considerable literature on the subject. For example, Hockney [1, 2], Buzbee et al. [3], Swartztrauber [4], and Temperton [5, 6] describe the methods of Fourier analysis and cyclic reduction as applied to the discrete (two-dimensional) Poisson equation on a rectangle, Le Bail [7] gives a detailed classification of equations which may be solved by FFT methods, and Houstis and Papatheodorou [8] derived a 9-point FFT method for the Helmholtz equation and provide detailed comparisons with the results obtained from other algorithms. Wilhelmson and Ericksen [9] consider FFT methods for the three-dimensional Poisson equation.

The main purpose of the present paper is to describe an FFT method which may be used to solve the discrete Poisson equation on a rectangle with a grid consisting of regular hexagons, each of which is divided into six equilateral triangles. Computational results are presented for a model Dirichlet problem for various grid sizes and these results are compared with those obtained on rectangular grids.

2. THE FINITE DIFFERENCE EQUATIONS

With indices i and j defined as indicated in Fig. 1, the usual finite difference approximation (see, for example, Allen [10]) to Poisson's equation

$$\nabla^2\Phi(x, y) = q(x, y), \tag{1}$$

0021-9991/86 $3.00

Copyright © 1986 by Academic Press, Inc.
All rights of reproduction in any form reserved.

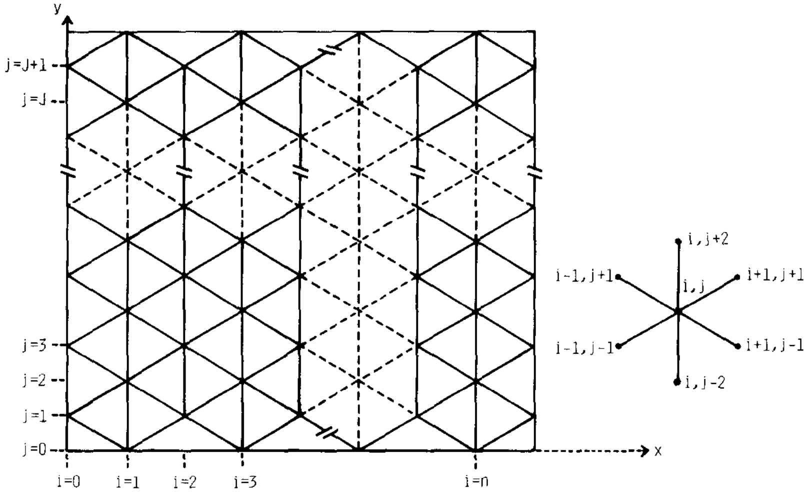

FIG. 1. The region of integration and finite difference molecule.

on a mesh consisting of equilateral triangles of side h, may be written

$$\begin{gathered} -6\Phi_{i,j}+\Phi_{i,j-2}+\Phi_{i,j+2}+\Phi_{i+1,j+1}+\Phi_{i-1,j+1} \\ +\Phi_{i+1,j-1}+\Phi_{i-1,j-1}=Q_{i,j}+O(h^6) \\ (i=1,2,\ldots,n;\ j=1,2,\ldots,J), \end{gathered} \qquad (2)$$

where

$$Q_{i,j}=\frac{3h^2}{2}q_{i,j}+\frac{3h^4}{32}\nabla^2 q_{i,j}, \qquad (3)$$

and we shall assume that n is odd and J is even.

For j even and odd we define, respectively, the vectors $\boldsymbol{\phi}_j$ and $\boldsymbol{\psi}_j$ by

$$\boldsymbol{\phi}_j=(\Phi_{1,j},\Phi_{3,j},\ldots,\Phi_{n,j})^T$$

and (4)

$$\boldsymbol{\psi}_j=(\Phi_{2,j},\Phi_{4,j},\ldots,\Phi_{n-1,j})^T,$$

where $\boldsymbol{\phi}_j$ has $m=(n+1)/2$ components, and $\boldsymbol{\psi}_j$ has $m-1$ components. Hence, for j even, we find that

$$-6\boldsymbol{\phi}_j+\boldsymbol{\phi}_{j-2}+\boldsymbol{\phi}_{j+2}+B(\boldsymbol{\psi}_{j-1}+\boldsymbol{\psi}_{j+1})=\mathbf{r}_j^*, \qquad (5)$$

where

$$B=\begin{bmatrix} 1 & 0 & \cdot & \cdot & \cdot & 0 \\ 1 & 1 & 0 & \cdot & \cdot & 0 \\ 0 & 1 & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & 0 & 1 & 1 \\ 0 & \cdot & \cdot & \cdot & 0 & 1 \end{bmatrix}_{m\times(m-1)} \tag{6}$$

$$\mathbf{r}_j^* = \mathbf{r}_j - (\Phi_{0,j-1} + \Phi_{0,j+1}, 0,...0, \Phi_{n+1,j-1} + \Phi_{n+1,j+1})^T, \tag{7}$$

$$\mathbf{r}_j = (Q_{1,j}, Q_{3,j},..., Q_{n,j})^T, \tag{8}$$

and, for j odd,

$$-6\boldsymbol{\psi}_j + \boldsymbol{\psi}_{j-2} + \boldsymbol{\psi}_{j+2} + B^T(\boldsymbol{\phi}_{j-1} + \boldsymbol{\phi}_{j+1}) = \mathbf{s}_j, \tag{9}$$

where

$$\mathbf{s}_j = (Q_{2,j}, Q_{4,j},..., Q_{n-1,j})^T. \tag{10}$$

The terms $\Phi_{0,j\pm 1}$, $\Phi_{n+1,j\pm 1}$ which appear in Eq. (7) are known boundary values.

We may eliminate the vectors $\boldsymbol{\psi}_j$, $\boldsymbol{\psi}_{j\pm 1}$, $\boldsymbol{\psi}_{j\pm 2}$ from Eqs. (5) and (9) as follows. For j odd we have from (5) that

$$-6\boldsymbol{\phi}_{j-1} + \boldsymbol{\phi}_{j-3} + \boldsymbol{\phi}_{j+1} + B(\boldsymbol{\psi}_{j-2} + \boldsymbol{\psi}_j) = \mathbf{r}_{j-1}^* \tag{11}$$

$$-6\boldsymbol{\phi}_{j+1} + \boldsymbol{\phi}_{j-1} + \boldsymbol{\phi}_{j+3} + B(\boldsymbol{\psi}_j + \boldsymbol{\psi}_{j+2}) = \mathbf{r}_{j+1}^*, \tag{12}$$

and hence $\boldsymbol{\psi}_{j\pm 2}$ may be eliminated by multiplying (9) by B and subtracting (11) and (12), giving

$$-8B\boldsymbol{\psi}_j - \boldsymbol{\phi}_{j-3} - \boldsymbol{\phi}_{j+3} + (BB^T + 5I_m)(\boldsymbol{\phi}_{j+1} + \boldsymbol{\phi}_{j-1}) = \mathbf{R}_j, \tag{13}$$

where I_m is the unit matrix of order m and

$$\mathbf{R}_j = B\mathbf{s}_j - \mathbf{r}_{j-1}^* - \mathbf{r}_{j+1}^*. \tag{14}$$

Relation (13) may thus be employed in (5) to produce, for j even ($\neq 2, J$), the equation

$$2(H_m - 15I_m)\,\boldsymbol{\phi}_j + (H_m + 16I_m)(\boldsymbol{\phi}_{j-2} + \boldsymbol{\phi}_{j+2}) - \boldsymbol{\phi}_{j-4} - \boldsymbol{\phi}_{j+4} = \mathbf{R}_j^*, \tag{15}$$

where

$$\mathbf{R}_j^* = 8\mathbf{r}_j^* + \mathbf{R}_{j-1} + \mathbf{R}_{j+1}, \tag{16}$$

and

$$H_m = BB^T - 4I_m = \begin{bmatrix} -3 & 1 & 0 & \cdot & \cdot & 0 \\ 1 & -2 & 1 & 0 & \cdot & 0 \\ 0 & 1 & -2 & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & 0 & 1 & -2 & 1 \\ 0 & \cdot & \cdot & 0 & 1 & -3 \end{bmatrix}. \tag{17}$$

For $j=4$ Eq. (15) contains ϕ_0, a vector of known boundary values, which may be taken to the right-hand side, and similarly for $j=J-2$ the term ϕ_{J+2} is known. For $j=2, J$ a special procedure was employed to derive finite difference equations which have the same order truncation error as (15) and incorporate the boundary conditions on $j=0, J+2$. This procedure is explained for the particular case $j=2$. Essentially, ψ_1, ψ_3, and ψ_5 may be eliminated by using Eq. (5) for $j=2, 4$ and Eq. (9) for $j=1, 3$, but we note that, for $j=1$, Eq. (9) involves the "fictitious" ψ_{-1} which may be obtained as follows.

Figure 2 shows the grid near the boundary for i even. Using Taylor series it is easy to derive the relation

$$\Phi_{i,-1} = -\Phi_{i,1} + 2\Phi_{i,0} + \frac{h^2}{4}\left(\frac{\partial^2 \Phi}{\partial y^2}\right)_{i,0} + \frac{h^4}{192}\left(\frac{\partial^4 \Phi}{\partial y^4}\right)_{i,0} + O(h^6), \tag{18}$$

where $\Phi_{i,-1}$ denotes a fictitious value of Φ and $\Phi_{i,0}$ denotes a known boundary value of Φ. Thus, assuming that (1) is satisfied at $(i, 0)$ and using the relation $\nabla^4\Phi = \nabla^2 q$, we may show that to $O(h^6)$

$$\psi_{-1} = -\psi_1 + \psi^{(0)}, \tag{19}$$

where $\psi^{(0)}$ is a known vector whose components are given by

$$\psi_i^{(0)} = 2\Phi_{i,0} + \frac{h^2}{4}\left(q - \frac{\partial^2 \Phi}{\partial x^2}\right)_{i,0} + \frac{h^4}{192}\left(2\frac{\partial^2 q}{\partial y^2} - \nabla^2 q + \frac{\partial^4 \Phi}{\partial x^4}\right)_{i,0}. \tag{20}$$

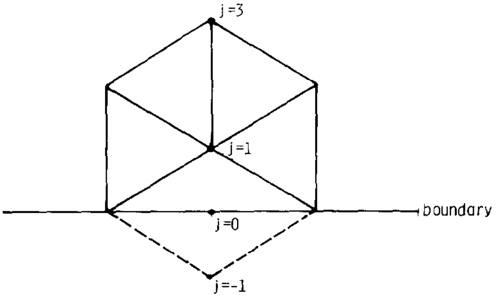

FIG. 2. The grid near the boundary $j=0$.

It is clear that the case $j=J$ may be treated similarly and for the problem considered in Section 4, $\psi^{(0)}$ and $\psi^{(J+2)}$ may be calculated from analytic formulae. For other problems it may be necessary to evaluate these vectors approximately and the configuration of the grid allows the use of suitable difference formulae for the derivatives in (20) to enable $\psi^{(0)}$ and similarly $\psi^{(J+2)}$ to be evaluated with leading error term $O(h^6)$.

Thus the complete system of equations may be written as

$$
\begin{aligned}
&(2H_m-29I_m)\,\phi_2+(H_m+16I_m)\,\phi_4-\phi_6\\
&\qquad=6(\mathbf{r}_2^*-\phi_0)-\mathbf{r}_4^*+B(\mathbf{s}_1+\mathbf{s}_3-B^T\phi_0-\psi^{(0)})\\
&(H_m+16I_m)\,\phi_2+2(H_m-15I_m)\,\phi_4+(H_m+16I_m)\,\phi_6-\phi_8\\
&\qquad=\mathbf{R}_4^*+\phi_0\\
&-\phi_{j-4}+(H_m+16I_m)\,\phi_{j-2}+2(H_m-15I_m)\,\phi_j+(H_m+16I_m)\,\phi_{j+2}-\phi_{j+4}\\
&\qquad=\mathbf{R}_j^*\qquad(j=6,8,\ldots,J-4),\\
&-\phi_{J-6}+(H_m+16I_m)\,\phi_{J-4}+2(H_m-15I_m)\,\phi_{J-2}+(H_m+16I_m)\,\phi_J\\
&\qquad=\mathbf{R}_{J-2}^*+\phi_{J+2},\\
&-\phi_{J-4}+(H_m+16I_m)\,\phi_{J-2}+(2H_m-29I_m)\,\phi_J\\
&\qquad=6(\mathbf{r}_J^*-\phi_{J+2})-\mathbf{r}_{J-2}^*+B(\mathbf{s}_{J+1}+\mathbf{s}_{J-1}-B^T\phi_{J+2}-\psi^{(J+2)}).
\end{aligned}
\tag{21}
$$

Equations (21) have a quindiagonal block structure of $J/2$ blocks in which each non-null mth-order block is either $-I_m$ or a linear function of the tridiagonal matrix H_m. It is perhaps worth remarking here that if the ϕ_j's rather than the ψ_j's are eliminated, the resulting equations corresponding to (21) have a similar block structure but the tridiagonal blocks are linear functions of the matrix $\bar{H}_{m-1}=B^TB-4I_{m-1}$, where

$$
\bar{H}_{m-1}=\begin{bmatrix}
-2 & 1 & 0 & \cdot & \cdot & \cdot & 0\\
1 & -2 & 1 & 0 & \cdot & \cdot & 0\\
0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0\\
0 & \cdot & \cdot & 0 & 1 & -2 & 1\\
0 & \cdot & \cdot & \cdot & 0 & 1 & -2
\end{bmatrix}. \tag{22}
$$

As is well known, this matrix arises naturally in the FFT method for the Dirichlet problem for the 5-point discrete Poisson equation on a rectangular grid, whereas the matrix (17) differs from those usually encountered in problems which may be solved by the FFT method. As there appears to be no particular advantage in either formulation, we present here the details for the FFT method based on Eq. (21).

The eigenvalues and corresponding eigenvectors of H_m are easily shown to be given by

$$\lambda_s = -2 + 2\cos\alpha_s,$$

and

$$\mathbf{x}_s = \left(\sin\frac{\pi}{2}\alpha_s, \sin\frac{3\pi}{2}\alpha_s, \ldots, \sin\left(v+\frac{1}{2}\right)\alpha_s, \ldots, \sin\left(m-\frac{1}{2}\right)\alpha_s\right)^T \qquad (s = 1, 2, \ldots, m), \tag{23}$$

respectively, where $\alpha_s = s\pi/m$, and the eigenvectors satisfy the relations

$$\begin{aligned} \mathbf{x}_r^T\mathbf{x}_s &= 0, && r \neq s, \\ &= m/2, && r = s \neq m, \\ &= m, && r = s = m. \end{aligned} \tag{24}$$

Thus defining

$$\begin{aligned} &\mathbf{R}_2^{**} = 6(\mathbf{r}_2^* - \boldsymbol{\phi}_0) - \mathbf{r}_4^* + B(\mathbf{s}_1 + \mathbf{s}_3 - B^T\boldsymbol{\phi}_0 - \boldsymbol{\psi}^{(0)}) \\ &\mathbf{R}_4^{**} = \mathbf{R}_4^* + \boldsymbol{\phi}_0 \\ &\mathbf{R}_j^{**} = \mathbf{R}_j^* \qquad (j = 6, 8, \ldots, J-4) \\ &\mathbf{R}_{J-2}^{**} = \mathbf{R}_{J-2}^* + \boldsymbol{\phi}_{J+2} \\ &\mathbf{R}_J^{**} = 6(\mathbf{r}_J^* - \boldsymbol{\phi}_{J+2}) - \mathbf{r}_{J-2}^* + B(\mathbf{s}_{J+1} + \mathbf{s}_{J-1} - B^T\boldsymbol{\phi}_{J+2} - \boldsymbol{\psi}^{(J+2)}), \end{aligned} \tag{25}$$

and setting

$$\boldsymbol{\phi}_j = \sum_{s=1}^{m} c_{s,j}\mathbf{x}_s$$

and (26)

$$\mathbf{R}_j^{**} = \sum_{s=1}^{m} d_{s,j}\mathbf{x}_s$$

so that $d_{s,j} = \mathbf{x}_s^T \mathbf{R}_j^{**}/|\mathbf{x}_s|^2$, the systems of equations for the coefficients $c_{s,j}$ take the quindiagonal form

$$\begin{aligned}
&(2\lambda_s - 29)\, c_{s,2} + (\lambda_s + 16)\, c_{s,4} - c_{s,6} &&= d_{s,2}\\
&(\lambda_s + 16)\, c_{s,2} + 2(\lambda_s - 15)\, c_{s,4} + (\lambda_s + 16)\, c_{s,6} - c_{s,8} &&= d_{s,4}\\
&-c_{s,j-4} + (\lambda_s + 16)\, c_{s,j-2} + 2(\lambda_s - 15)\, c_{s,j} + (\lambda_s + 16)\, c_{s,j+2} - c_{s,j+4} &&= d_{s,j}\\
&&&(j = 6, 8,\dots, J-4) \qquad (27)\\
&\quad -c_{s,J-6} + (\lambda_s + 16)\, c_{s,J-4} + 2(\lambda_s - 15)\, c_{s,J-2} + (\lambda_s + 16)\, c_{s,J} &&= d_{s,J-2}\\
&\qquad -c_{s,J-4} + (\lambda_s + 16)\, c_{s,J-2} + (2\lambda_s - 29)\, c_{s,J} &&= d_{s,J},
\end{aligned}$$

where $s = 1, 2,\dots, m$. In the usual manner, once the coefficients have been determined from (27), $\boldsymbol{\phi}_j$ may be synthesised using (26) and an algorithm which efficiently implements the necessary Fourier analysis and synthesis is given in the Appendix. The solution for $j = 1, 3, 5,\dots, J+1$ may be recovered, if required, by solving the equations

$$\begin{aligned}
-7\boldsymbol{\psi}_1 + \boldsymbol{\psi}_3 &= \mathbf{s}_1 - B^T(\boldsymbol{\phi}_0 + \boldsymbol{\phi}_2) - \boldsymbol{\psi}^{(0)}\\
\boldsymbol{\psi}_{j-2} - 6\boldsymbol{\psi}_j + \boldsymbol{\psi}_{j+2} &= \mathbf{s}_j - B^T(\boldsymbol{\phi}_{j-1} + \boldsymbol{\phi}_{j+1}) \qquad (j = 3, 5,\dots, J-1) \qquad (28)\\
\boldsymbol{\psi}_{J-1} - 7\boldsymbol{\psi}_{J+1} &= \mathbf{s}_{J+1} - B^T(\boldsymbol{\phi}_J + \boldsymbol{\phi}_{J+2}) - \boldsymbol{\psi}^{(J+2)},
\end{aligned}$$

which immediately decouple into $m-1$ tridiagonal systems each of order $J/2+1$, for the components $\psi_j^{(k)}$, $k = 1, 2,\dots, m-1$, $j = 1, 3, 5,\dots, J+1$.

Before describing the computational procedure some further comments are relevant. In principle it is possible to perform cyclic reduction on Eqs. (21) but the eliminated equations have a block tridiagonal structure which require the use of the FFT method or further cyclic reductions for their solution. It is likely that a method consisting of a combination of quindiagonal and tridiagonal cyclic reduction procedures and FFT would produce an optimised algorithm for solving Eqs. (21), in the manner of the usual FACR(l) method for the 5-point approximation (Swartztrauber [4], Temperton [5, 6]). The reduction in execution time achieved by using this algorithm compared with using Fourier analysis or cyclic reduction alone is typically 10–60%. For the hexagonal grid method this approach requires some further study, particularly as cyclic reduction may be used for both the quindiagonal systems and the eliminated tridiagonal equations. In the present paper we concentrate on solving Eqs. (21) by the direct application of the FFT method as already described, although we note also that the method can be readily adapted to problems for which the domain is periodic in the j-direction. For this case the equations corresponding to (21) may be written in the form

$$\begin{aligned}
&U\boldsymbol{\phi}_0 + V\boldsymbol{\phi}_2 - \boldsymbol{\phi}_4 && -\boldsymbol{\phi}_{J-2} + V\boldsymbol{\phi}_J = \mathbf{R}_0^* \\
&V\boldsymbol{\phi}_0 + U\boldsymbol{\phi}_2 + V\boldsymbol{\phi}_4 - \boldsymbol{\phi}_6 && -\boldsymbol{\phi}_J = \mathbf{R}_2^* \\
&-\boldsymbol{\phi}_{j-2} + V\boldsymbol{\phi}_{j-2} + U\boldsymbol{\phi}_j + V\boldsymbol{\phi}_{j+2} - \boldsymbol{\phi}_{j+4} && = \mathbf{R}_j^* \\
& && (j = 4, 6, \ldots, J-4) \qquad (29) \\
&-\boldsymbol{\phi}_0 && -\boldsymbol{\phi}_{J-6} + V\boldsymbol{\phi}_{J-4} + U\boldsymbol{\phi}_{J-2} + V\boldsymbol{\phi}_J = \mathbf{R}_{J-2}^* \\
&V\boldsymbol{\phi}_0 - \boldsymbol{\phi}_2 && -\boldsymbol{\phi}_{J-4} + V\boldsymbol{\phi}_{J-2} + U\boldsymbol{\phi}_J = \mathbf{R}_J^*,
\end{aligned}$$

where $U = 2(H_m - 15I_m)$ and $V = H_m + 16I_m$.

3. Computational Procedure

We consider solving Eq. (1) over the region shown in Fig. 1 with $0 \leqslant x \leqslant 1$, $0 \leqslant y \leqslant 1/\sqrt{3}$ for a sequence of grid sizes defined by the parameters

$$n = 2^{\gamma} - 1, \qquad (30)$$

where γ is an integer ($\geqslant 4$),

$$\begin{aligned}
m &= \tfrac{1}{2}(n+1) \\
J &= n - 1
\end{aligned}$$

and $\qquad (31)$

$$h = 1/m\sqrt{3}.$$

It is convenient to work with two arrays $QE(i_0, j_e)$ and $Q0(i_e, j_0)$ for which the corresponding grid point coordinates are defined by

$$\left.\begin{aligned}
x_0 &= (i_0 - \tfrac{1}{2})\, h\sqrt{3} && (i_0 = 1, 2, \ldots, m) \\
y_e &= j_e h && (j_e = 0, 1, 2, \ldots, J/2 + 1),
\end{aligned}\right. \qquad (32)$$

and

$$\left.\begin{aligned}
x_e &= i_e h\sqrt{3} && (i_e = 0, 1, 2, \ldots, m) \\
y_0 &= (j_0 - \tfrac{1}{2})\, h && (j_0 = 1, 2, \ldots, J/2 + 1)
\end{aligned}\right. \qquad (33)$$

and the overall computational procedure is generally similar to that for the usual FFT method on a rectangular grid with one level of cyclic reduction.

The vectors $\boldsymbol{\phi}_0, \boldsymbol{\phi}_{J+2}$ of known boundary values are stored in appropriate elements of the array QE and the right-hand sides of (21) are calculated and stored in QE. The array $Q0$ is used to store the vectors $\mathbf{s}_1 - \boldsymbol{\psi}^{(0)}$, $\mathbf{s}_j$, $j = 3, \ldots, J-1$, and

$\mathbf{s}_{J+1}-\boldsymbol{\psi}^{(J+2)}$ which appear in (28). The right-hand sides of (27) are calculated using Fourier analysis and the m sets of quindiagonal systems (27) are solved using a Gaussian elimination algorithm (see, for example, von Rosenberg [11]) and the final solution, for j even, is obtained in the array QE by Fourier synthesis using (26). The solution for j odd may then be obtained in the array $Q0$ by solving Eqs. (28).

4. Result and Discussion

For the actual computations we chose

$$q(x, y)=(x^2+y^2)\,e^{xy} \tag{34}$$

for which the analytical solution to (1) is

$$\Phi(x, y)=e^{xy} \tag{35}$$

and appropriate boundary values were computed using (35). Values of the maximum modulus error and RMS errors (obtained by comparison with (35)) are shown in Table I for values of γ in the range $4 \leqslant \gamma \leqslant 7$, together with the corresponding execution times. These times were calculated by running the program typically 100–200 times for $4 \leqslant \gamma \leqslant 6$ and 20–50 times for $\gamma=7$ and include the time taken to solve (28). The program (Program I) was written in Fortran 77 and run on a Prime 9950 machine with approximately 13 decimal digit precision.

For the purposes of comparison a second program (Program II) was written to solve the same equation (using the FFT method) over the rectangular region $0 \leqslant x \leqslant 1$, $0 \leqslant y \leqslant 1/\sqrt{3}$ using the usual 5-point approximation with step-lengths $h_x=1/(N+1)$, $h_y=h_x/\sqrt{3}$, where $N=2^{\gamma-1}-1$. This choice of grid implies that, for a given value of γ, the work involved in Fourier analysis and synthesis is approximately the same for this method as for the hexagonal grid calculation since the usual form of sine transform was used (Cooley et al. [12]). Furthermore, both Programs I and II used a real periodic FFT algorithm based on the Cooley–Tukey

TABLE I

Values of Maximum Modulus Error, RMS Error, and Execution Times for the Solution of Poisson's Equation on a Regular Hexagonal Grid

γ	m	$J/2$	Maximum modulus error	RMS error	Execution time (sec)
4	8	7	3.1, −8	1.8, −8	0.057
5	16	15	1.9, −9	1.0, −9	0.25
6	32	31	1.2, −10	6.1, −11	0.75
7	64	63	1.5, −11	7.9, −12	2.77

TABLE II

Values of Maximum Modulus Error, RMS Error, and Execution Times for the Solution of Poisson's Equation Using the Usual 5-Point Approximation

γ	N	Maximum modulus error	RMS error	Execution time (sec)
4	7	5.1, −6	2.7, −6	0.037
5	15	1.4, −6	6.6, −7	0.13
6	31	3.4, −7	1.6, −7	0.46
7	63	8.6, −8	3.9, −8	1.74

algorithm [13] and Table II shows the results obtained for the 5-point approximation.

In order to gain some insight into the efficiency of these programs compared with those of other authors a Fortran version of POT1 (Christiansen and Hockney [14]) was run on the Prime 9950 under the same operating conditions. The values for the maximum modulus error and RMS error using POT1 were found to be identical with those shown in Table II and the execution times are given in Table III. That these times are generally similar to those given in Table II for corresponding values of N is not surprising since POT1 requires approximately half the number of Fourier transforms of Program II but takes about twice as long as Program II to evaluate the sine transform. Overall the results indicate that POT1 is rather more efficient than Program II.

It is clear that the execution times shown in Tables II and III are smaller than the corresponding values shown in Table I, although as γ increases the relative difference between corresponding times decreases. These results are probably a consequence of the complexity of the calculation of the right-hand sides of (21) and the fact that these equations are quindiagonal rather than tridiagonal. Furthermore, as γ increases we would expect the major part of the execution time to be spent on the Fourier analysis and synthesis, which, for both Programs I and II, is roughly com-

TABLE III

Execution Times For POT1 on the Prime 9950[a]

N	Execution time (sec)
7	0.026
15	0.093
31	0.35
63	1.35

[a] The values of maximum modulus error and RMS error are as given in Table II.

parable for the same value of γ, and our results tend to confirm that this is the case. As indicated earlier, the times shown in Table I include the solution for j odd (Eqs. (28)) and if this were omitted it was found that, for example, for $\gamma = 7$ the execution time was reduced by approximately 11% of the value shown.

The values of the maximum modulus error and RMS error shown in Table I are all substantially less than the corresponding values shown in Table II. Moreover, the fact that the hexagonal grid approximation is $O(h^4)$ is clearly demonstrated by the reduction of errors by a factor of approximately 16 between consecutive values of γ for $4 \leqslant \gamma \leqslant 6$ (in the case $\gamma = 7$ the solution accuracy is near machine precision) whereas Table II demonstrates the $O(h^2)$ nature of the 5-point approximation. Rather than comparing error values in Tables I and II for the same value of γ, it is perhaps more realistic to compare entries in Table I for $\gamma = \gamma_0$ with entries in Table II for $\gamma = \gamma_0 + 1$, since then $h_x \simeq 0.9h$. Thus we find that, for example, for $\gamma_0 = 4$ the RMS error for the hexagonal grid calculation is approximately 37 times smaller than that for the 5-point method, whereas, for $\gamma_0 = 6$, the corresponding factor is approximately 640. It is worth pointing out here that all the values of the maximum modulus error and RMS error given in Table II are larger than those in Table I, irrespective of the value of γ. Thus, for example, comparing the case $\gamma = 4$ in Table I with $\gamma = 7$ in Table II, we see that for our model problem, roughly comparable accuracy is achieved by both methods and that, on this basis, the hexagonal method is faster than the 5-point method (Program II) by a factor of approximately 30. A similar comparison using the POT1 times for the 5-point calculations gives a factor of almost 24 and thus it is clear that the hexagonal grid method can offer a considerable advantage in terms of speed for a given accuracy over the usual 5-point method.

This type of advantage for higher-order methods has been confirmed for more general elliptic equations by, for example, Houstis et al. [8, 15] and Manohar and Stephenson [16]. In particular, the method considered in [8] is a fourth-order (cartesian grid) FFT method for the Helmholtz equation. These authors considered eight test problems and compared their results with those obtained from several fast direct methods and concluded that their method is much superior to second-order methods on the basis of speed for comparable accuracy. Averaging over seven of the test problems chosen they found their 9-point method to be a factor of 51 times faster than the best 5-point methods with the individual factors for three of the problems being in the range 19 to 27, approximately, and a further three in the range 70–108. A factor of 7 was obtained for a problem which has a singularity in the third derivative.

5. Conclusions

We have shown that it is possible to solve Poisson's equation on a regular hexagonal grid using FFT techniques. The results for the test problem considered here demonstrate that the method is more accurate than the 5-point method for a given step size and also that, for the two methods to give solutions of comparable accuracy, the hexagonal solver is faster than methods based on the 5-point formula by a factor of between 20 and 30 approximately.

These results are in general accord with those found by other workers using high-order cartesian grid methods.

Appendix

We describe here an efficient procedure for the evaluation of the Fourier analysis and synthesis required in Eqs. (26). We consider the inverse discrete Fourier transform (IDFT)

$$\mathbf{z}=\sum_{s=1}^{m} \beta_s \mathbf{x}_s, \tag{A1}$$

where $\mathbf{x}_s$ denotes an eigenvector given by (23) and $\mathbf{z}$ is an m-component vector. Denoting by z_k the kth component of $\mathbf{z}$ we have that

$$z_k=\sum_{s=1}^{m} \beta_s \sin{(2k-1)}\frac{\pi s}{2m} \qquad (k=1, 2,..., m) \tag{A2}$$

and using (24), the corresponding discrete Fourier transform (DFT) is given by

$$\beta_s=\frac{2}{m}\sum_{k=1}^{m} z_k \sin\frac{s\pi(2k-1)}{2m} \qquad (s=1, 2,..., m-1) \tag{A3}$$

and

$$\beta_m=\frac{1}{m}\sum_{k=1}^{m} z_k(-1)^{k+1}.$$

By defining

$$\begin{aligned} \hat{\beta}_s &= m\beta_s/2 \qquad (s=1, 2,..., m-1) \\ \hat{\beta}_m &= m\beta_m \end{aligned} \tag{A4}$$

Eqs. (A3) may be written in the form

$$\hat{\beta}_s=\sum_{k=1}^{m} z_k \sin\frac{s\pi(2k-1)}{2m} \qquad (s=1, 2,..., m) \tag{A5}$$

and we note that this relation is the same as that given by Swartztrauber [4] (his Eq. (3.17)) for the IDFT which is required in the solution of the Dirichlet–Neumann problem for Poisson's equation in a rectangle using the usual 5-point approximation.

Thus, following Swartztrauber's procedure, we may deduce that

$$\begin{aligned}\hat{g}_s &= (\hat{\beta}_s + \hat{\beta}_{m-s}) \cos\frac{s\pi}{2m} + (\hat{\beta}_s - \hat{\beta}_{m-s}) \sin\frac{s\pi}{2m} \\ &= z_1 + \sum_{k=1}^{m/2-1} \left[(z_{2k+1} - z_{2k}) \cos\frac{2s\pi k}{m} + (z_{2k+1} + z_{2k}) \sin\frac{2s\pi k}{m} \right] \\ &\quad + (-1)^{s+1} z_m, \end{aligned} \tag{A6}$$

and clearly this relation has the form of the real periodic inverse transform, that is, Eq. (A6) is of the form

$$\begin{aligned}\hat{g}_s &= \frac{1}{2} g_1 + \sum_{k=1}^{m/2-1} \left(g_{2k} \cos\frac{2s\pi k}{m} + g_{2k+1} \sin\frac{2s\pi k}{m} \right) \\ &\quad + \frac{1}{2} g_m (-1)^s \qquad (s = 1, 2, ..., m), \end{aligned} \tag{A7}$$

where

$$\left.\begin{aligned} g_1 &= 2z_1 \\ g_{2k} &= z_{2k+1} - z_{2k} \\ g_{2k+1} &= z_{2k+1} + z_{2k} \\ g_m &= -2z_m. \end{aligned}\right. \qquad k = 1, 2, ..., m/2 - 1 \tag{A8}$$

Thus, given the values z_k, $k = 1, 2, ..., m$, we use (A8) to construct g_k, $k = 1, 2, ..., m$, and perform the real inverse transform (A7) (of length m) to obtain $\hat{g}_s$, $s = 1, 2, ..., m$. From (A6) we may show that $\hat{\beta}_s$ may be determined from

$$\hat{\beta}_s = \frac{1}{2} (\hat{g}_s + \hat{g}_{m-s}) \sin\frac{s\pi}{2m} + \frac{1}{2} (\hat{g}_s - \hat{g}_{m-s}) \cos\frac{s\pi}{2m} \tag{A9}$$

and the required DFT (the values of β_s) is obtained from relations (A4). The IDFT (Eqs. (A2)) may be calculated essentially by reversing the above procedure. Using relations (A4) one may calculate $\hat{g}_s$ from the first of relations (A6) and hence perform a real periodic DFT on this data to produce g_1, $g_2, ..., g_m$. Equations (A8) are then easily solved for the required z_k, $k = 1, 2, ..., m$.

Acknowledgment

I am pleased to thank Mrs. L. C. Wilkinson for programming assistance.

References

1. R. W. Hockney, *J. Assoc. Comput. Mach.* **12**, 95 (1965).
2. R. W. Hockney, *Methods in Computational Physics* (Academic Press, New York/London, 1970), Vol. 9, p. 136.
3. B. L. Buzbee, G. H. Golub, and C. W. Nielson, *SIAM J. Numer. Anal.* **7** 627 (1970).
4. P. Swartztrauber, *SIAM Rev.* **19**, 490 (1977).
5. C. Temperton, *J. Comput. Phys.* **31**, 1 (1979).
6. C. Temperton, *J. Comput. Phys.* **34**, 314 (1980).
7. R. C. Le Bail, *J. Comput. Phys.* **9**, 440 (1972).
8. E. N. Houstis and T. S. Papatheodorou, *Advances in Computer Methods for Partial Differential Equations II*, edited by R. Vichnevetsky (IMACS, New Brunswick, NJ 1977), p. 46.
9. R. B. Wilhelmson and J. H. Ericksen, *J. Comput. Phys.* **25**, 319 (1977).
10. D. N. de G. Allen, *Relaxation Methods in Engineering and Science* (McGraw–Hill, New York/London, 1954), p. 147.
11. D. U. von Rosenberg, *Methods for the Numerical Solution of Partial Differential Equations* (Elsevier, New York, 1969), p. 113.
12. J. W. Cooley, P. A. W. Lewis, and P. D. Welch, *J. Sound Vibration* **12**, 315 (1970).
13. J. W. Cooley and J. W. Tukey, *Math. Comput.* **19**, 297 (1965).
14. J. P. Christiansen and R. W. Hockney, *Comput. Phys. Commun.* **2**, 139 (1971).
15. E. N. Houstis, R. E. Lynch, J. R. Rice, and T. S. Papatheodorou, *J. Comput. Phys.* **27**, 323 (1978).
16. R. P. Manohar and J. W. Stephenson, *J. Comput. Phys.* **51**, 444 (1983).