A Fast Direct Solution of Poisson's Equation Using Fourier Analysis

R. W. HOCKNEY

Stanford University, Stanford, California*

Abstract. The demand for rapid procedures to solve Poisson's equation has led to the development of a direct method of solution involving Fourier analysis which can solve Poisson's equation in a square region covered by a 48 × 48 mesh in 0.9 seconds on the IBM 7090. This compares favorably with the best iterative methods which would require about 10 seconds to solve the same problem.

The method is applicable to rectangular regions with simple boundary conditions and the maximum observed error in the potential for several random charge distributions is 5×10^{-7} of the maximum potential change in the region.

1. *Introduction*

In many engineering problems concerning plasmas, electron tubes and ion guns, it is desired to follow the motion of numerous electrostatically interacting charged particles in two dimensions. If the region involved is divided into a large number of cells, and the velocity and position of each charged particle is recorded, then this simulation of space charge flow may be performed stepwise in time as follows:

1. *Charge Distribution.* At the beginning of each time step the position of each particle is examined and the charge of each particle is associated with the center of the cell in which the particle resides.

2. *Potential.* The charge distribution found in step 1 is used as the source term or right-hand side of Poisson's equation, the solution of which gives the electrostatic potential in the region.

3. *Acceleration.* The potential distribution found in step 2 is differenced to give an approximation to the electrostatic field acting on each particle. This field is then allowed to accelerate each particle individually for a short time interval. The new position and velocity of each particle is recorded and the cycle repeats at step 1. The description is thus analogous to the projection of a motion picture.

For such a simulation to be useful it is necessary to follow several thousand particles through several hundred time-steps and this means that the overall cycle time must be reduced to a few seconds or less.

The acceleration of all the particles is a simple calculation and can be performed in about a second on the IBM 7090.[1] The solution of Poisson's partial differential equation in step 2 is more difficult but it is clear that the solution must be obtained in about the same time if the simulation is to be useful.

Hitherto the tendency has been to use iterative methods to solve such an el-

* Computation Center.

[1] Computation times given in this paper will be for this machine except where specified.

Journal of the Association for Computing Machinery, Vol. 12, No. 1 (January, 1965), pp. 95–113

liptic equation. Theoretical estimates of the computing time for the best iterative methods, namely the two-line cyclic Chebyshev (2LCC) and Alternating Direction Implicite (ADI) methods, have been made which compare well with the experimental results of Hageman [1] and Price and Varga [2].

These lead to solution times of 10, 30 and 60 seconds respectively for ADI, 2LCC and SOR methods when applied to a 48 × 48 square mesh and an error reduction of 10^{-6}.

These solution times are thus roughly 10 times too slow for this application.

The iterative methods of solution named above are very general and can be used to solve Poisson's equation in systems with complicated electrode shapes and boundary conditions. In plasma applications however, where the behavior of the space charge distribution is of primary importance, it is often permissible to simplify the boundary conditions in order to obtain a faster solution.

In this paper we describe an alternative direct method of solution which takes advantage of this simplification, is applicable to a certain class of important problems, and is 10 times faster than the best iterative methods so far reported.

2. *Motivation and Discussion*

The problem to be discussed here is the solution of Poisson's equation in a rectangular domain where the boundary conditions are given on the perimeter of the domain only. The boundary conditions may be Dirichlet, Neumann or periodic (combination being permitted provided that the same type of condition pertains along the total length of any side). The method shows to best advantage in (x, y) cordinates. Thus we consider this case, taking the boundary conditions to be zero potential around the perimeter.

We have

$$\frac{\partial^2\phi(x,y)}{\partial x^2} + \frac{\partial^2\phi(x,y)}{\partial y^2} = \rho(x,y) \qquad \begin{cases} 0 \leqq x \leqq l \\ 0 \leqq y \leqq m \end{cases} \tag{1}$$

where $\phi(x, y) = 0$ for $x = 0, l$ or $y = 0, m$.

2.1 *Fourier Analysis.* The boundary conditions allow $\phi(x, y)$ to be expanded in a Fourier series in either the x-direction, y-direction or as a double Fourier series in both directions. A double Fourier expansion was suggested as long ago as 1952 by Hyman [7] and is essentially the method of Tensor Products, reported recently by Lynch et al [8]. However the determination of Fourier coefficients is a time consuming job on a computer and we have found that the fastest computer program is obtained if we expand in only one direction and choose this to be shortest. Let this be the x-direction then the expansion is

$$\phi(x,y) = \sum_k \bar{\phi}^k(y) \sin\frac{\pi k x}{l} \tag{2}$$

and similarly for $\rho(x, y)$ where $\bar{\phi}^k(y)$ is the Fourier amplitude of the kth harmonic.

On substituting (2) into the partial differential equation (1) and using the

orthogonal properties of the sine functions we obtain a set of ordinary differential equations relating the Fourier amplitudes of $\phi(x, y)$ and $\rho(x, y)$

$$\frac{d^2\bar{\phi}^k(y)}{dy^2} - \left(\frac{\pi k}{I_x}\right)^2 \bar{\phi}^k(y) = \bar{\rho}^k(y). \tag{3}$$

A single Fourier analysis has also been considered by Bickley and McNamee [12].

In the continuous case an infinite number of harmonic amplitudes are required in the representation of $\phi(x, y)$. However when performing the finite analogue of the expansion (2) to express the value of $\phi(x, y)$ at a discrete number of mesh points only, we find the number of harmonics required for the *exact* representation of the mesh function is equal to the number of mesh points (see for example Jeffreys and Jeffreys [6, par. 14.01]).

Due to the fact that the sine functions satisfy the boundary conditions and are the eigenfunctions of the differential operator in equation (1), the ordinary differential equations (3) for each harmonic are independent of each other. This change of a partial differential equation into a set of independent ordinary differential equations is the first crucial simplification of the method. It can only be carried out in certain simple geometrical situations when, for example, the external boundaries are parallel to the coordinate axes and the boundary conditions are of the type mentioned above. The presence of any internal conductors for example immediately couples the harmonics in equation (3) and makes the method as it stands impractical. However a modification of this direct method is being investigated which will allow the inclusion of interior boundaries and is suitable for cases where Poisson's equation is to be solved repeatedly for different space charge distributions but with fixed interior electrode surfaces.

2.2 *Tridiagonal Systems.* The ordinary differential equations (3), which in the finite analogue become a tridiagonal matrix equation, can be solved in a variety of ways. Our experience has been that the best technique depends on the boundary conditions imposed.

In the case that the potential and therefore $\bar{\phi}^k(y)$ has prescribed values at $y = 0, m$ the method of Gauss elimination in the neat form as given by Varga [9] and others is suitable and may be used for any number of mesh points. Gauss elimination is an inefficient method to use if the boundary conditions are periodic, and a new technique of "recursive cyclic reduction" has been developed for this case which is particularly neat if the number of mesh points is of the form 2^p or 3×2^p (see Section 6). This does not seem to be a severe restriction considering the resulting increase in computing speed. Indeed recursive cyclic reduction may be applied to the Dirichlet boundary conditions for these special numbers of mesh points and has the advantage over Gauss elimination in that it does not require the precomputation and storage of the auxiliary vector ω (see Varga [9, p. 195]).

An interesting and quick method of solution has been suggested by O. Buneman [11] for the case that $\bar{\phi}^k(0)$ is given and we have an open ended Neumann condition that

$$\frac{d\bar{\phi}^k}{dy} = \bar{\phi}^k = 0 \qquad (y = \infty). \quad (4)$$

In this case the equations (3) or their finite difference analogue may be factorized as follows:

$$\left\{\frac{d}{dy} - \left(\frac{\pi k}{L}\right)\right\}\left\{\frac{d}{dy} + \left(\frac{\pi k}{L}\right)\right\}\bar{\phi}^k(y) = \bar{\rho}^k(y). \quad (5)$$

Introducing the auxiliary function $\bar{\psi}^k(y)$ defined by

$$\left\{\frac{d}{dy} - \left(\frac{\pi k}{L}\right)\right\}\bar{\psi}^k(y) = \bar{\rho}^k(y), \quad (6)$$

we have

$$\left\{\frac{d}{dy} + \left(\frac{\pi k}{L}\right)\right\}\bar{\phi}^k(y) = \bar{\psi}^k(y). \quad (7)$$

Applying the condition (4) to (7) we see that $\bar{\psi}^k(\infty) = 0$. Integrating equation equation (6) inwards from infinity we see that $\bar{\psi}^k(y) = 0$ until the first charge is encountered, at say $y = \hat{y}$. In practice therefore (6) is integrated only from $y = \hat{y}$ to $y = 0$ yielding $\bar{\psi}^k(\hat{y})$ to $\bar{\psi}^k(0)$. Knowing the right-hand side, equation (7) may be integrated from $y = 0$ to $y = \hat{y}$ starting with the known value of $\bar{\phi}^k(0)$. This technique is known as the *marching method* and if the march is performed in the directions given with the factorization shown, there is no build-up of error due to the homogenous solutions of equations (6) and (7).

Having obtained $\bar{\phi}^k(y)$ as the solution of equation (3) the potential $\phi(y)$ is obtained by Fourier synthesis from equation (2).

Due to the reciprocity of the finite Fourier analysis and synthesis the program for Fourier synthesis will have much in common if not all in common with the program for Fourier analysis.

Summarizing we see that the solution is obtained in three stages:

1. Fourier analysis of the charge distribution $\rho(x, y) \rightarrow \bar{\rho}^k(y)$.
2. Solution of k independent sets of ordinary differential equations or the corresponding tridiagonal matrix equations $\bar{\rho}^k(y) \rightarrow \bar{\phi}^k(y)$.
3. Fourier synthesis of the potential distribution $\bar{\phi}^k(y) \rightarrow \phi(x, y)$.

2.3 *Computer Time.* On examining the number of computer operations[2] required to perform this calculation the method does not, at first sight, seem particularly attractive. This is mainly due to the time required to perform the analysis and synthesis, as may be seen if we consider the domain of the solution to be spanned by an $(n \times n)$ mesh. For stage 1, on each of the n lines of constant y, we must compute n Fourier components each of which require n operations giving a total of n^3 operations for the whole mesh. The solution of the n equations for one harmonic in stage 2 may be completed in the order of n operations giving a total of approximately n^2 operations for stage 2. Stage 3 of course also takes n^3 operations.

[2] Here the intended meaning is a multiplication and the addition that usually accompanies it.

As the conventional iterative methods will require of the order of n^2 operations per iterations it seems that the Fourier technique will only pay off if the number of iterations required is considerably larger than n. In a step by step simulation, when a good guess for the potential is available from the last time step, it seems quite likely that satisfactory convergence can be obtained in less than n iterations (n is typically 50 to 100). In this case no advantage has been obtained by the Fourier transformation and we have unnecessarily restricted ourselves to certain simple types of boundary conditions.

2.4 *Simplifications.* Two further simplifications are, however, available in the Fourier method which completely reverse the above assessment. In the first place if a suitable number is chosen for n (such as 12, 24, 48) the symmetry in the sine functions may be used to reduce the computing time for analysis and synthesis to about a tenth of the original estimate (see sect. 9(b)). Furthermore the two-cyclic nature of the finite difference equations allows one to replace the original n^2 equations involving all the points in the mesh to a set of $n^2/2$ slightly more complex equations involving only the points on the even lines of the mesh. This process known as *cyclic-reduction* may be done at the start and fortunately gives a set of revised equations which may also be solved by the Fourier method. The Fourier analysis and synthesis is then performed on only half the number of lines and computing time is reduced. The solution is completed by solving for the potential on the odd lines of the mesh directly from the known solution on the even lines. We have called this process *odd/even reduction* (see Section 4).

2.5 *Operations and Storage.* The Fourier method as described above applied to (x, y) geometry can solve Poisson's equation on a (48×48) mesh in 0.9 second with an error of about 10^{-6}. This time corresponds to about 10 computer operations per mesh point and if it is estimated that an iterative method will require at least 2 operations per point per iteration, we can see that an iterative method would have to converge in 5 iterations or less for it to be faster. It is hardly credible that any iterative method can achieve this.

Throughout the calculation new results may overwrite old and the storage required is very little more than the original mesh at n^2 points. With the aid of the results of Section 9 we can extend the comparison made by Lynch in [8] of the total number of arithmetic operations required to solve Poisson's equation on an $(N \times N)$ mesh:

SOR	*Tensor Product*	*ADI*	*Fourier*
$14N^3 \log N$	$4N^3$	$40N^2 \log^2 N$	$N^3/18 + 15.5N^2$

On the basis of these estimates the Fourier method when applicable is always superior to SOR and the Tensor product methods and is superior to ADI for $N < 10{,}000$ which includes all practical cases that can be solved on present day machines.

2.6 *Other Geometries.* (x, y) geometry is not always very realistic as it implies the existence of an infinite system in the z-dimension. For many applications axially symmetric geometry on (r, z) coordinates is more appropriate. The

Fourier method may be applied in these coordinates as described above if the Fourier analysis is performed in the z-direction and the only change is that the tridiagonal system of equations in stage 2 now has variable coefficients. The cyclic reduction method is not suitable for such equations but the Gauss elimination method is as efficient in radial coordinates as in the x-coordinates. Thus if the z-direction is the shortest there is no change in computing speed due to the change in the coordinate system. However if the z-direction is the largest, as it frequently will be in electron tube work, the computing time will be increased and the alternative of performing a Bessel analysis and synthesis in the shorter r-direction must be considered. The Fourier method with Bessel analysis proceeds in 3 stages as before, however there is no symmetry in the Bessel analysis and the reduction of the number of operations by a factor of about 10 cannot be achieved as it could in the case of Fourier analysis. The odd/even reduction, however, may be performed as before. For z long enough a Bessel analysis in the shorter r-direction will be beneficial. Preliminary estimates suggest that Bessel analysis should only be performed if (z/r) is greater than about 10.

2.7 *Generalization.* The basic principle of the Fourier method is the expansion of the solution in terms of the eigenfunctions of the Laplace operator for the problem. This principle can be applied in general to block matrices with commuting blocks and shows to its best advantage if the eigenfunctions of the blocks are sines and cosines. This situation does arise in the finite difference form of certain common linear partial differential equations and these are discussed in Section 11.

The Fourier method has also been successfully used in the transient study of the Magnetron by Yu, Kooyers and Buneman [10] using a (48×96) mesh and by Buneman and Wadhwa [11] in an ion gun problem using a (24×100) mesh.

3. *The* (48×48) *Plasma Problem*

Considering now, in detail, the application of the Fourier method to a particular situation arising in a plasma study which uses a (48×48) mesh, we report on the measured speed and accuracy of the solution. The boundary conditions being periodic are slightly different from the problem discussed in Section 2 but the principle of the method is unchanged.

Consider a square region in (x, y) geometry covered by a square 48×48 mesh, with the boundary condition that the solution be periodically repeated in both the x- and y-directions.[3]

Using the usual 5-point difference approximation, Poisson's equation may be written in finite difference form as

$$\phi_{i-1,j} + \phi_{i+1,j} + \phi_{i,j-1} + \phi_{i,j+1} - 4\phi_{i,j} = q_{i,j} \quad (i, j = 0, 1, \cdots, 47) \quad (8)$$

where ϕ_{ij} is the potential at the (i, j) node of the mesh and $q_{i,j}$ is the charge

[3] In order for the potential to be doubly periodic it is necessary for the total charge in the repeat square to be zero. We assume this to be the case.

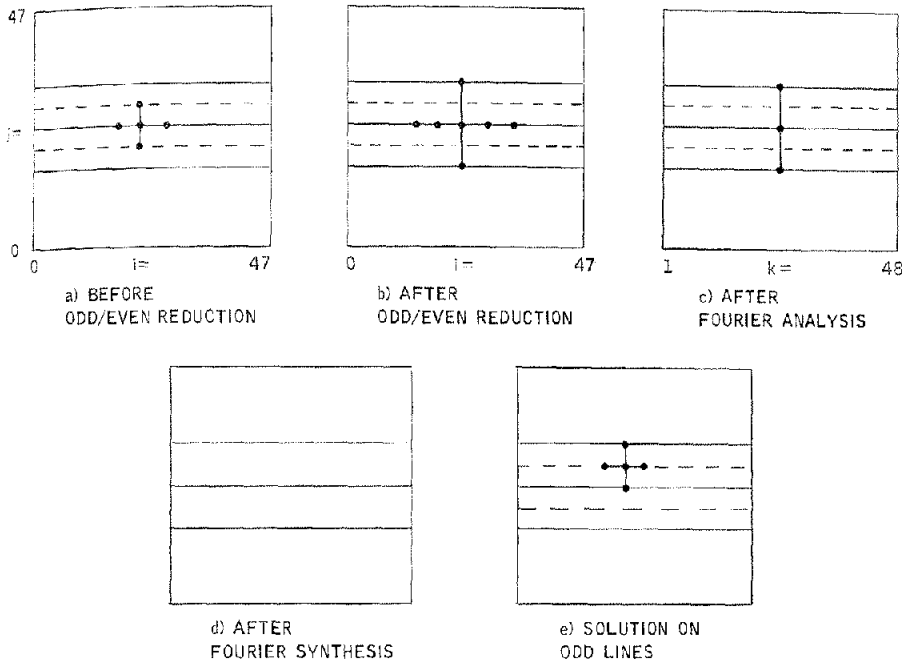

FIG. 1

associated with the (i, j) node of the mesh. The mesh numbering and interaction module for this approximation is shown in Figure 1a.

The boundary conditions are

$$\phi_{i+48k,j+48k} = \phi_{i,j}, \qquad q_{i+48k,j+48k} = q_{i,j} \tag{9}$$

where k is any integer.

A convenient way of including these boundary conditions is to state that all indices are to be interpreted modulo 48, and this is assumed throughout the rest of this paper.

The equations (8) with boundary conditions (9) may be written in block matrix form as follows:

$$B\phi = \begin{pmatrix} A & I & 0 & \cdots & 0 & I \\ I & A & I & & & \vdots \\ \vdots & & \ddots & & & 0 \\ 0 & & & I & A & I \\ I & 0 & \cdots & 0 & I & A \end{pmatrix} \begin{pmatrix} \phi_0 \\ \phi_1 \\ \phi_{(47)} \end{pmatrix} = \begin{pmatrix} \mathbf{q}_0 \\ \mathbf{q}_1 \\ \mathbf{q}_{(47)} \end{pmatrix} \tag{10}$$

where

$$\phi_j = \begin{pmatrix} \phi_{0j} \\ \phi_{1j} \\ \vdots \\ \phi_{47,j} \end{pmatrix}, \qquad \mathbf{q}_j = \begin{pmatrix} q_{0j} \\ q_{1j} \\ \vdots \\ q_{47,j} \end{pmatrix} \tag{11}$$

and

$$A = \begin{pmatrix} -4 & 1 & 0 & \cdots & 0 & 1 \\ 1 & -4 & 1 & & & 0 \\ 0 & & \ddots & & & \vdots \\ \vdots & & & \ddots & & 0 \\ 0 & & & 1 & -4 & 1 \\ 1 & 0 & \cdots & 0 & 1 & -4 \end{pmatrix}. \tag{12}$$

4. *Odd/Even Reduction*

The first step in the solution of equation (10) for the unknown potential on 48 lines of the mesh is the reduction of the problem to the solution of 24 more complicated equations for the unknown potential on the even numbered lines of the mesh only. After solving for the potential on the even lines the potential on the odd lines is obtained by exact interpolation as described in Section 8.

Consider three neighboring equations from the matrix equation (10):

$$\begin{aligned} \boldsymbol{\phi}_{j-2} + A\,\boldsymbol{\phi}_{j-1} + \boldsymbol{\phi}_j &= \mathbf{q}_{j-1} \\ \boldsymbol{\phi}_{j-1} + A\,\boldsymbol{\phi}_j + \boldsymbol{\phi}_{j+1} &= \mathbf{q}_j \\ \boldsymbol{\phi}_j + A\,\boldsymbol{\phi}_{j+1} + \boldsymbol{\phi}_{j+2} &= \mathbf{q}_{j+1} \end{aligned} \tag{13}$$

for $j = 0, 2, 4, \cdots, 46$ with the indices interpreted modulo 48. By multiplying the second equation on the left by $-A$ and adding we obtain

$$\boldsymbol{\phi}_{j-2} + (2I - A^2)\boldsymbol{\phi}_j + \boldsymbol{\phi}_{j+2} = \mathbf{q}_{j-1} - A\mathbf{q}_j + \mathbf{q}_{j+1} \qquad (j = 0, 2, \cdots, 46). \tag{14}$$

The equations (14) are 24 equations for the even lines with a 7 point interaction module as shown in Figure 1b. In expanded form they are

$$\phi_{i,j-2} - \phi_{i-2,j} + 8\phi_{i-1,j} - 16\phi_{i,j} + 8\phi_{i+1,j} - \phi_{i+2,j} + \phi_{i,j+2} = q_{i,j-1} - q_{i-1,j} + 4q_{i,j} - q_{i+1,j} + q_{i,j+1}.$$

5. *Fourier Analysis*

To solve equation (14) we first form a modified charge distribution on the even lines defined by

$$\mathbf{q}_j^* = \mathbf{q}_{j-1} - A\mathbf{q}_j + \mathbf{q}_{j+1} \qquad (j = 0, 2, \cdots, 46), \tag{15}$$

which in expanded form is

$$q^*_{i,j} = q_{i,j-1} - q_{i-1,j} + 4q_{i,j} - q_{i+1,j} + q_{i,j+1} \qquad \begin{cases} i = 0, 1, \cdots, 47, \\ j = 0, 2, \cdots, 46. \end{cases} \tag{16}$$

From the point of view of machine storage the modified charge density on the even lines may overwrite the original charge density as it is formed.

Next the potential and charge distribution are expanded in Fourier components

as follows:

$$\phi_{i,j} = \frac{1}{2}\bar{\phi}^{c}_{0,j} + \frac{1}{2}\bar{\phi}^{c}_{24,j}(-1)^{i} + \sum_{k=1}^{23}\left\{\bar{\phi}^{c}_{k,j}\cos\frac{2\pi ki}{48} + \bar{\phi}^{s}_{k,j}\sin\frac{2\pi ki}{48}\right\} \tag{17}$$

where

$$\bar{\phi}^{c}_{k,j} = \frac{2}{48}\sum_{i=0}^{47}\phi_{i,j}\cos\frac{2\pi ki}{48}, \qquad \bar{\phi}^{s}_{k,j} = \frac{2}{48}\sum_{i=0}^{47}\phi_{i,j}\sin\frac{2\pi ki}{48} \tag{18}$$

with analogous expressions for $q^{*}_{i,j}$, $\bar{q}^{c*}_{k,j}$ and $\bar{q}^{s*}_{k,j}$.

The sine and cosine functions satisfy the orthogonality relations:

$$\begin{aligned}
&\sum_{i=0}^{47}\cos\frac{2\pi ki}{48}\cos\frac{2\pi li}{48} = \delta_{kl}\frac{48}{2} && (k, l = 1, 2, \cdots, 23)\\
&\sum_{i=0}^{47}\cos\frac{2\pi ki}{48}\cos\frac{2\pi li}{48} = \delta_{kl}48 && (k = l = 0 \text{ or } 24)\\
&\sum_{i=0}^{47}\sin\frac{2\pi ki}{48}\sin\frac{2\pi li}{48} = \delta_{kl}\frac{48}{2} && (k, l = 1, 2, \cdots, 23)\\
&\sum_{i=0}^{47}\sin\frac{2\pi ki}{48}\cos\frac{2\pi li}{48} = 0 && \begin{cases} k = 1, 2, \cdots, 23\\ l = 0, 1, \cdots, 24. \end{cases}
\end{aligned} \tag{19}$$

Substituting the expansion (17) into (14) and using the orthogonality relations (19) we get the finite Fourier transform of equation (14)

$$\bar{\phi}_{k,j-2} + \lambda_k\bar{\phi}_{k,j} + \bar{\phi}_{k,j+2} = \bar{q}^{*}_{k,j} \tag{20}$$

where $\bar{\phi}$ and $\bar{q}^{*}$ refer to either the sine or cosine harmonic and

$$\lambda_k = -2\left(8 - 8\cos\frac{2\pi k}{48} + \cos\frac{4\pi k}{48}\right). \tag{21}$$

Noting that, because the chosen sines and cosines are the eigenfunctions of the matrix A, the equations (20) are 48 independent sets of 24 equations, one set for each of the 48 harmonic amplitudes.

The Fourier transform of the modified charge distribution on the even lines, $\bar{q}^{*}_{k,j}$, may overwrite the modified charge density on these lines. The storage layout and resulting interaction module is shown in Figure 1c.

6. *Recursive Cyclic Reduction*

The set of 24 equations for any of the 48 harmonic amplitudes may be written

$$\phi_{j-2} + \lambda\phi_j + \phi_{j+2} = q_j \qquad (j = 0, 2, \cdots, 46) \tag{22}$$

where the bar, star and constant subscript k have been dropped for brevity. These equations form a tridiagonal system with periodic boundary conditions and a particularly efficient method of solution has been devised in collaboration with Dr. G. Golub. This involves the recursive application of the process of cyclic reduction which follows.

Equation (22) is identical in form to equation (10) except that the matrix A is replaced by the scalar λ and the subscript advances in steps of 2 instead of 1.

The process of reducing the number of equations by half as described in equations (13) and (14) may now be similarly applied, leading to 12 equations linking every fourth line, namely:

$$\phi_{j-2^2} + \lambda^{(2)}\phi_j + \phi_{j+2^2} = q_j^{(2)} \qquad (j = 0, 4, \cdots, 44) \quad (23)$$

where

$$\lambda^{(2)} = 2 - \lambda^2, \qquad q_j^{(2)} = q_{j-2} - \lambda q_j + q_{j+2}\,. \quad (24)$$

The 12 equations (23) are of identical form to the equations (22) but with a modified right-hand side, q_j, and central coefficient, λ, as given by equation (24). The quantity $q_j^{(2)}$ may, for storage economy, overwrite the q_0, q_4, $q_8 \cdots q_{44}$, while q_2, q_6, $q_{10} \cdots q_{46}$ are kept unchanged in their location.

The process of reduction may therefore be carried out recursively until a small number of equations are obtained which are solved directly.

If we let t be the depth of the recursion the recurrence formulas become

$$\phi_{j-2^t} + \lambda^{(t)}\phi_j + \phi_{j+2^t} = q_j^{(t)} \qquad (j = 0, 2^t, \cdots, (48 - 2^t)) \quad (25)$$

where

$$\begin{aligned} \lambda^{(t+1)} &= 2 - (\lambda^{(t)})^2 \\ q_j^{(t+1)} &= q_{j-2^t}^{(t)} - \lambda^{(t)} q_j^{(t)} + q_{j+2^t}^{(t)} \end{aligned} \quad (26)$$

with

$$\left.\begin{aligned} \lambda^{(1)} &= \lambda \\ q_j^{(1)} &= \bar{q}_j^{*} \end{aligned}\right\}. \quad (27)$$

Three applications of the reduction process leave us with 3 equations for ϕ_0, ϕ_{16} and ϕ_{32} which cannot be further reduced, namely:

$$C\boldsymbol{\phi} = \begin{pmatrix} \lambda^{(4)} & 1 & 1 \\ 1 & \lambda^{(4)} & 1 \\ 1 & 1 & \lambda^{(4)} \end{pmatrix} \begin{pmatrix} \phi_0 \\ \phi_{16} \\ \phi_{32} \end{pmatrix} = \begin{pmatrix} q_0^{(4)} \\ q_{16}^{(4)} \\ q_{32}^{(4)} \end{pmatrix} = \mathbf{q}. \quad (28)$$

The eigenvalues, μ_i, and vectors, u_i of the matrix C are known:

$$\begin{aligned} \mathbf{u}_1' &= (1, 1, 1,), & \mu_1 &= 2 + \lambda^{(4)} \\ \mathbf{u}_2' &= (1, -2, 1), & \mu_2 &= \lambda^{(4)} - 1 \\ \mathbf{u}_3' &= (1, 0, -1), & \mu_3 &= \lambda^{(4)} - 1 \end{aligned} \quad (29)$$

where the prime denotes transpose.

Expanding the solution in terms of the eigenvectors,

$$\boldsymbol{\phi} = \alpha_1\mathbf{u}_1 + \alpha_2\mathbf{u}_2 + \alpha_3\mathbf{u}_3 \quad (30)$$

then

$$C\phi = \mathbf{q} = \alpha_1\mu_1\mathbf{u}_1 + \alpha_2\mu_2\mathbf{u}_2 + \alpha_3\mu_3\mathbf{u}_3 \tag{31}$$

and

$$\begin{aligned}
\alpha_1 &= \frac{\mathbf{u}_1' \cdot \mathbf{q}}{\mathbf{u}_1' \cdot \mathbf{u}_1} = \frac{q_0^{(4)} + q_{16}^{(4)} + q_{32}^{(4)}}{3(\lambda^{(4)} + 2)} \\
\alpha_2 &= \frac{\mathbf{u}_2' \cdot \mathbf{q}}{\mathbf{u}_2' \cdot \mathbf{u}_2} = \frac{q_0^{(4)} - 2q_{16}^{(4)} + q_{32}^{(4)}}{6(\lambda^{(4)} - 1)} \\
\alpha_3 &= \frac{\mathbf{u}_3' \cdot \mathbf{q}}{\mathbf{u}_3' \cdot \mathbf{u}_3} = \frac{q_0^{(4)} - q_{32}^{(4)}}{2(\lambda^{(4)} - 1)} .
\end{aligned} \tag{32}$$

Substituting equations (32) into (30) we get the solution

$$\begin{aligned}
\phi_0 &= \alpha_1 + \alpha_2 + \alpha_3 \\
\phi_{16} &= \alpha_1 - 2\alpha_2 + \alpha_3 \\
\phi_{32} &= \alpha_1 - \alpha_3 .
\end{aligned} \tag{33}$$

In order to find the other values of ϕ we calculate intermediate values recursively. First determining ϕ_8, ϕ_{24}, ϕ_{40} then ϕ_4, ϕ_{12}, ϕ_{20}, ϕ_{28}, ϕ_{36}, ϕ_{44}, etc. from the relation

$$\phi_j = \frac{1}{\lambda^{(t)}} \{q_j^{(t)} - \phi_{j-2^t} - \phi_{j+2^t}\} \tag{34}$$

for $t = 3, 2, 1$ and for $j = 2^t$ step 2^{t+1} until $(48 - 2^t)$ where all the quantities on the right-hand side of equation (34) are known.

The process of cyclic reduction described here is essentially a floating-point algorithm due to the fact that the magnitude of $\lambda^{(t)}$ can grow very quickly particularly for the higher harmonics. Consider for example the harmonic with $k = 24$, when

$$\begin{aligned}
\lambda_{24}^{(1)} &= -34 \\
\lambda_{24}^{(2)} &= -1.154 \\
\lambda_{24}^{(3)} &= -1.33 \times 10^6 \\
\lambda_{24}^{(4)} &= -1.77 \times 10^{12}.
\end{aligned} \tag{35}$$

This might be thought of as a disadvantage, bringing as it does the danger of machine overflow. In fact the phenomonon may be turned to advantage on a floating-point machine by noticing that if, at any level of the reduction, $\lambda^{(t)} > 10^n$ and we are only interested in computing with a precision of 1 part in 10^n, then equation (25) may be written

$$\lambda^{(t)}\phi_j = q_j^{(t)} \tag{36}$$

for $j = 0$ step 2^t until $(48 - 2^t)$ where the first and third terms of the left-hand side have been neglected in comparison with the second.

Thus the solution ϕ_j at the tth level can be determined by simple division from equation (37)

$$\phi_j = \frac{q_j^{(t)}}{\lambda^{(t)}} \tag{37}$$

and calculation of intermediate values started immediately.

An alternative scaling of the cycle reduction method can be made in which numbers decrease in magnitude and which is therefore suitable for a fixed-point machine. However it appears that an extra multiplication is introduced.

7. *Solution on the Even Lines*

The solution of the equations (22) by the technique of recursive cyclic reduction has determined the values of all 48 harmonic amplitudes on the 24 even lines of the mesh. The solution on the even lines is found by the process of Fourier synthesis using equation (17), and the stage indicated by Figure 1d is reached.

8. *Solution on the Odd Lines*

The solution for the potential on the odd lines can be found from equation (13)

$$A\phi_j = \mathbf{q}_j - \phi_{j-1} - \phi_{j+1} \tag{38}$$

for $j = 1$ step 2 until 47 where the potential vectors on the right-hand side are the known values on the even lines.

The equation (38) is a tridiagonal system with periodic boundary conditions and again is most conveniently solved by recursive cyclic reduction, starting from the expanded form of equation (38)

$$\phi_{i+1,j} - 4\phi_{i,j} + \phi_{i+1,j} = q_{i,j} - \phi_{i,j-1} - \phi_{i,j+1} \tag{39}$$

for $j = 1$ step 2 until 47, for $i = 0$ step 1 until 47.

9. *Operation Count and Speed*

In order to get more general formulas for the number of operations we consider an $(n \times m)$ mesh where the Fourier analysis is performed in the n-direction. The number of operations for the different stages of the calculation are as follows.

(a) *Form modified charge density on even lines.* According to equation (16) this takes 5 additions per point.[4] There are n points per line and $m/2$ lines therefore a total of

[4] The multiplication by 4 is an addition into the exponent of a floating-point number.

$$n \times \frac{m}{2} \times 5 = 2\frac{1}{2}nm \quad \text{additions}$$
$$0 \quad \text{multiplications} \tag{40}$$

(b) *Fourier analysis of the modified charge on even lines.* According to equation (18) Fourier analysis would require n multiplications and n additions per harmonic per line. There are n harmonics and $m/2$ lines therefore without any simplification we get a total of

$$n \times n \times \frac{m}{2} = \frac{n^2}{2}m \quad \text{additions}$$
$$\frac{n^2}{2}m \quad \text{multiplications} \tag{41}$$

If however we make use of the symmetry of the sines and cosines, grouping and adding together all terms multiplied by the same factor, before performing the multiplication, the number of operations can be drastically reduced. See for example Whittaker and Robinson [3] who give a formulation for $n = 12$ and 24. These techniques can be extended generally to $n = 12 \times 2^q$, where q is an integer equal to or greater than 1, and an ALGOL program has been written and tested for this case.

The number of operations in this program has been counted as a function of n and fitted empirically as follows:

$$\frac{n^2}{36} + 5.7n \quad \text{additions}$$
$$\frac{n^2}{36} + 0.6n \quad \text{multiplications.}$$

These formulas are asymptotically correct for large n, are correct to within 3 percent for $q \geqq 3$ and overestimate the total number of operations by 24 percent and 8 percent for $q = 1$ and 2, respectively.

Using these results we obtain for step (b)

$$\frac{n^2m}{72} + 2.85nm \quad \text{additions}$$
$$\frac{n^2m}{72} + 0.3nm \quad \text{multiplications.} \tag{42}$$

(c) *Solution of harmonic amplitudes of potential on even lines.* For a line of points 48 long, equations (27) and (34) show the operational counts for the process of cyclic reduction to be 2×95 additions and 94 multiplications.

In general we may say for a line q points long, cyclic reduction takes

$$4 \times q \text{ additions and } 2 \times q \text{ multiplications.} \tag{43}$$

In the determination of the harmonic amplitudes at this stage there are n tridiagonal systems to be solved each $m/2$ long. The total count is therefore

$$\begin{aligned} 4 \times n \times \frac{m}{2} &= 2nm \qquad \text{additions} \\ 2 \times n \times \frac{m}{2} &= nm \qquad \text{multiplications.} \end{aligned} \tag{44}$$

(d) *Fourier synthesis.* The Fourier synthesis required to obtain the potential from the harmonic amplitudes of potential via equation (17) can be simplified by grouping of terms to the same number of operations as for Fourier analysis in step (b) giving a further

$$\begin{aligned} \frac{n^2 m}{72} + 2.85nm \qquad &\text{additions} \\ \frac{n^2 m}{72} + 0.3nm \qquad &\text{multiplications.} \end{aligned} \tag{45}$$

(e) *Solution on odd lines.* First we form the right-hand side of equation (38) for all points on the odd lines. There are $n \times m/2$ such points giving $2 \times (nm)/2 = nm$ additions.

Next the tridiagonal system of equation (38) is solved by recursive cyclic reduction. There are $m/2$ such systems each n equation long. Using the results of (c) we have for the solution of these equations

$$\begin{aligned} 4 \times n \times \frac{m}{2} &= 2nm \qquad \text{additions} \\ 2 \times n \times \frac{m}{2} &= nm \qquad \text{multiplications.} \end{aligned} \tag{46}$$

The total number of operations for stage (e) is therefore

$$\begin{aligned} 3nm \qquad &\text{additions} \\ nm \qquad &\text{multiplications.} \end{aligned} \tag{47}$$

Total operations and storage. The number of operations for the solution of Poisson's equation given a right-hand side is therefore

$$\begin{aligned} &\frac{n^2 m}{36} + 13.2nm \qquad \text{additions} \\ \text{and} \quad &\frac{n^2 m}{36} + 2.3nm \qquad \text{multiplications} \\ \text{or in total} \quad &\frac{n^2 m}{18} + 15.5nm \qquad \text{arithmetic operations} \end{aligned} \tag{48}$$

Throughout the process new results may overwrite old and we need basically only one mesh of $(n \times m)$ storage locations. These originally contain the charge

TABLE 1. IBM 7090

Stage	*Additions*	*Estimated time (sec)*	*Multiplications*	*Estimated time (sec)*	*Total estimated time*	*Measured time*
a	$2\frac{1}{2}nm$	0.086	0	0	0.086	0.319
b	$n^2m/72$ $+2.85nm$	0.121	$n^2m/72$ $+0.3nm$	0.056	0.177	
c	$2nm$	0.069	nm	0.058	0.127	0.168
d	$n^2m/72$ $+2.85nm$	0.121	$n^2m/72$ $+0.3nm$	0.056	0.177	0.230
e	$3nm$	0.103	nm	0.058	0.161	0.189
Solution of Poisson's equation on 48 × 48 mesh =					0.728	0.906

TABLE 2. IBM 7094

Stage	*Additions*	*Estimated time*	*Multiplications*	*Estimated time*	*Total estimated time*	*Time*
a	$2\frac{1}{2}nm$	0.035	0	0	0.035	
b	$n^2m/72$ $+2.85nm$	0.049	$nm^2/72$ $+0.3nm$	0.022	0.071	
c	$2nm$	0.028	nm	0.023	0.051	
d	$n^2m/72$ $+2.85nm$	0.049	$nm^2/72$ $+0.3nm$	0.022	0.071	
e	$3nm$	0.041	nm	0.023	0.064	
Solution of Poisson's equation on (48 × 48) mesh =					0.292	0.362

TABLE 3

Machine	*Addition*	*Multiplication*
IBM 7090	15 μsec	25 μsec
IBM 7094	6 μsec	10 μsec

distribution which is overwritten by the Fourier transform of the charge, which is overwritten by the Fourier transform of the potential, which is finally overwritten by the potential solution.

The only other storage required is for the Fourier harmonics themselves. In general there would be $(n \times n)$ numbers describing the shape of the n harmonics however due to the symmetry of the sines and cosines only $n/4$ distinct numbers occur. The total date storage is therefore $(n \times m + n/4)$.

Tables 1 and 2 show the estimated times for each stage in the process using the operation counts above for the IBM 7090 and 7094, together with the measured time on the 7090.

For the purpose of estimation we have used the speeds in Table 3 for the floating-point operations. The measured time is taken from a floating-point symbolic FAP program. Due to the large number of additions some increase in speed could be obtained by programming in fixed point.

The difference between the measured and estimated times of about 24% i accounted for by computer housekeeping operations. Using this factor on th 7094 estimated figure we obtain 0.362 seconds as a realistic estimate for the tim of solution on the 7094. It is interesting to note from Table 1 that two frequentl repeated generalizations are untrue. It is not even approximately true, for ex ample, that additions may be neglected compared with multiplications, becaus in each stage of the process the time spent on additions is in fact greater tha the time spent on multiplications. It is also untrue that it is satisfactory t consider only the highest power of n for in this case the time spent computin operations proportional to n^2m is less than the time spent on stages with oper tions proportional to nm.

10. *Accuracy*

The accuracy of the method has been examined by testing its ability to r produce a given random distribution of potential.

We start by generating a random distribution of potential, ϕ^*, on the poin of the mesh. Next the charge distribution, q, which corresponds exactly to ϕ^* computed from equation (8) namely

$$q_{i,j} = \phi^*_{i-1,j} + \phi^*_{i+1,j} + \phi^*_{i,j-1} + \phi^*_{i,j+1} - 4\phi^*_{i,j}\,. \tag{49}$$

The Fourier technique was then used to derive a potential distribution, from the charge distribution q_{ij}, and the exact distribution ϕ^* and the solutio ϕ were examined.

The random distribution generated varied between $-\frac{1}{2}$ and $+\frac{1}{2}$ and the large value of the error, $(\phi^* - \phi)$, obtained with seven different distributions w 3.3×10^{-7}.

11. *Solution of Other Differential Equations*

Consider the general matrix equations

$$B\boldsymbol{\phi} = \mathbf{q} \tag{50}$$

where B is partitioned into $(m \times m)$ square blocks B_{ij} of size $(n \times m)$. $\boldsymbol{\phi}$ a $\mathbf{q}$ are partitioned into $(m \times 1)$ vectors $\boldsymbol{\phi}_j$ of length $(n \times 1)$.

$$B = \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1m} \\ B_{21} & B_{22} & & \cdot \\ \cdot & & \cdot & \cdot \\ \cdot & & \cdot & \cdot \\ B_{m1} & \cdots & & B_{mm} \end{pmatrix}, \quad \boldsymbol{\phi} = \begin{pmatrix} \boldsymbol{\phi}_1 \\ \boldsymbol{\phi}_2 \\ \cdot \\ \cdot \\ \boldsymbol{\phi}_m \end{pmatrix}, \quad \mathbf{q} = \begin{pmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \cdot \\ \cdot \\ \mathbf{q}_m \end{pmatrix}. \tag{5}$$

Such a system of partitioned equations naturally arises in the finite differen form of a two-dimensional partial differential equation, when the mesh is co fined to a rectangular region with m lines each containing n mesh points.

The Fourier technique in general can be applied to the solution of equatio (51) provided each submatrix B_{ij} has the same eigenvectors. In this case t

Fourier analysis and synthesis stages are performed using these eigenvectors. If equation (51) is to be solved only once the necessary determination of the eigenvectors will probably make the method unprofitable. If, however, equation (51) is to be solved repeatedly for different right-hand sides, as may occur in a transient problem, the solution for the eigenvectors need be done once only at the beginning. Even so, the method will not show to its full advantage unless the eigenvectors are sines and cosines when the special techniques for Fourier analysis discussed in Section 9b can be used.

Fortunately a large class of linear differential equations have this property and we now restrict the discussion to these.

The finite difference form of a linear partial differential equation at the point (i, j) on a mesh can be written in general as

$$\sum_{l,m=-p}^{+p} a_{l,m}\phi_{i+l,j+m} = \rho_{i,j} \tag{52}$$

where the coefficients $a_{l,m}$ are those appearing in the interaction module and in general are functions of i and j.

The equations will be of the desired form if:

1. The coefficients $a_{l,m}$ are independent of one index, say i which runs from 0 to n.

2. The coefficients have reflective symmetry with respect to this index, i.e., $a_{l,m} = a_{-l,m}$.

3. The boundary conditions can be expressed by defining $\phi_{i,j}$ if $i < 0$ as follows:

$$\begin{aligned} &\phi_{-i,j} = -\phi_{i,j} && \text{The zero value case} \\ \text{or}\quad &\phi_{-i,j} = +\phi_{i,j} && \text{The zero slope case} \\ \text{or}\quad &\phi_{-i,j} = \phi_{n-i,j} && \text{The periodic case} \end{aligned} \tag{53}$$

and similarly at the other boundary when $i > n$.

In terms of the original partial differential these conditions correspond to:

1. The coefficients are independent of one variable, say x.

2. The x derivatives appear only to an even order.

3. The boundary conditions are specified on a rectangle $0 \leqq x \leqq l_1$, $0 \leqq y \leqq l_2$ and are of the form

$$\begin{aligned} &\frac{\partial\phi}{\partial x} = 0 \quad \text{or} \quad \phi = 0 \qquad \text{along} \quad x = 0 \\ &\frac{\partial\phi}{\partial x} = 0 \quad \text{or} \quad \phi = 0 \qquad \text{along} \quad x = l_1 \end{aligned} \tag{54}$$

or the periodic condition $\phi(x, y) = \phi(x + l_1, y)$.

4. In the y-direction there is more freedom:

$$\begin{aligned} &a_1(x)\frac{\partial\phi}{\partial y} + b_1(x) = c_1(x) \qquad \text{along} \quad y = 0 \\ &a_2(x)\frac{\partial\phi}{\partial y} + b_2(x) = c_2(x) \qquad \text{along} \quad y = l_2 \end{aligned} \tag{55}$$

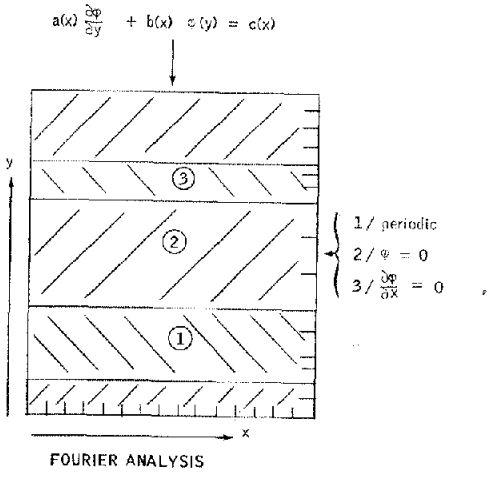

FIG. 2. Problems most suitable for Fourier method: (a) Rectangular region, (b) sla type property changes ①, ② etc., (c) simple boundary conditions, (d) uniform mesh a least in analysis direction, (e) typical equation $\nabla D(y)\nabla\phi(x, y) + k^2(y)\phi(x, y) = S(x, y)$

or the periodic condition $\phi(x, y) = \phi(x + l_2, y)$.

In this case a uniform mesh would be used in the x-direction and the Fourie method would be applied by performing the analysis and synthesis also in thi direction.

An example of an equation satisfying these conditions is the anisotropic di fusion equation in (x, y) or (r, z) coordinates, with Fourier analysis in the x and z-directions respectively:

$$D_x(y)\frac{\partial^2\phi}{\partial x^2} + \frac{\partial}{\partial y}D_y(y)\frac{\partial\phi}{\partial y} + k^2(y)\phi = s(x, y),$$

$$D_z(r)\frac{\partial^2\phi}{\partial z^2} + \frac{1}{r}\frac{\partial}{\partial r}\left\{D_r(r)\frac{\partial\phi}{\partial r}\right\} + k^2(r)\phi = s(r, z). \qquad (56$$

Laplace's, Poisson's and Helmholtz equations are special cases of this equatio

Both the five- and nine-point interaction modules for the Laplacian and th 25-point module [4] for the biharmonic equation $\nabla^4\phi = 0$ have the required r flective symmetry of the interaction coefficients and may also be solved by th Fourier method. Figure 2 shows diagrammatically the types of problem mos suitable for solution by the Fourier method.

The Fourier method, providing as it does a fast method for the solution o Poisson's equation over a rectangle, can be used as the basis for various bloc iterative schemes for more complex regions that can be divided up into rectangle One could consider for example a block 48-line iteration analogous to the bloc 1-line and 2-line methods [5].

12. *Conclusion*

For the special problems involving (x, y) geometry in the rectangle for whic the Fourier method is well suited there seems little doubt that it is a faste method of calculation than any direct or iterative method so far suggested.

For other problems where the method can be applied but is not well suited the position is less clear and we will have to await the results of practical numerical experiments before the fastest method can be chosen.

Acknowledgments. The author wishes to express his thanks to Professor O. Buneman of the Stanford Electronics Laboratories for suggesting this problem and for the original idea of using Fourier analysis. Thanks are also due to Professor G. Golub of the Stanford Computation Center for many stimulating discussions and ideas particularly regarding the solution of the tridiagonal equations.

This work was conducted under Triservice contract Nonr 225(24). The computations were made at the Stanford University Computation Center with the assistance of the National Science Foundation grant, NSF GP-948.

RECEIVED APRIL, 1964; REVISED AUGUST, 1964

REFERENCES

1. HAGEMAN, L. A. Block iterative methods for two cyclic matrix equations. WAPD-TM-327, Westinghouse Bettis Atomic Power Labs., Pittsburgh, Pa., April, 1962.
2. PRICE, H. S., AND VARGA, R. S. Recent numerical experiments comparing SOR and ADI methods. Rep. 91, Gulf Research and Development Co., Sept. 1962.
3. WHITTAKER AND ROBINSON. *Calculus of Observations.* Blackie & Son, London, 1944, 260–283.
4. MILNE. *Numerical Solution of Differential Equations.* Wiley, New York, 1960, 225.
5. PARTER, S. V. On two-line iterative methods for the Laplace and Biharmonic difference equations. *Num. Math. 1*, 208, 240–252.
6. JEFFREYS, H., AND JEFFREYS, B. S. *Mathematical Physics.* Cambridge U. Press, 1962.
7. HYMAN, M. A. Non iterative numerical solution of boundary value problems. *Appl. Sci. Res., B., 2* (1951), 325–351.
8. LYNCH, R. E., RICE, J. R., AND THOMAS, D. H. Direct Solution of Partial Difference equations by Tensor Product Methods. *Num. Math. 6* (1964), 185–199.
9. VARGA, R. S. *Matrix Iterative Analysis.* Prentice Hall, New York, 1962.
10. YU, S. P., KOOYERS, G. P., BUNEMAN, O. Simulation of distributed-emission crossed-field devices. Proc. 5th Internat. Cong. Microwave Tubes, Paris, 1964.
11. BUNEMAN, O., AND WADHWA, R. Solution of Poisson's equation and space charge neutralisation in ion gun systems. Private Communication.
12. BICKLEY, W. G., MCNAMEE, J. Matrix and other direct methods for the solution of systems of linear difference equations. *Phil. Trans. Roy. Soc. series A, 252* (1960), 69.