JOURNAL OF COMPUTATONAL PHYSICS **88**, 62–85 (1990)

On a Direct Method for Solving Helmholtz's Type Equations in 3-D Rectangular Regions

ARISTIDES TH. MARINOS

3 Agion Apostolon Street, Athens 113 62, Greece

Received February 5, 1988; revised May 30, 1989

A direct solution method for solving elliptic pde's of the type

$$k_x(z)\cdot\partial^2\varphi/\partial x^2 + k_y(z)\cdot\partial^2\varphi/\partial y^2 + k_z\cdot\partial^2\varphi/\partial z^2 + \sigma(z)\cdot\varphi = f(x, y, z)$$

in 3D parallelepipeds with $k_z = \text{const}$ and $k_x(z), k_y(z), \sigma(z)$ continuous functions of z, is presented. The spatial derivatives are approximated using the Hermite approach (Mehrstellenverfahren) with $O(h^6)$ truncation error for Dirichlet boundary conditions or for periodic solutions of the problem. For Neumann conditions, it seems that in order to retain the direct character of the numerical algorithm employed, one should approximate the first spatial derivatives on the boundary by means of conventional schemes having a truncation error of $O(h^3)$ type rather than $O(h^6)$ which accordingly reduce the overall accuracy of the results. Despite the substantial reduction of the overall accuracy for Neumann conditions, this case has not been exluded, because the structure of the difference equations remains invariant for problems in which instead of known values of first-order normal derivatives at the boundaries, these very boundaries constitute symmetry planes of the solution. This feature allows a direct solution method to be used for such a problem, whereas the $O(h^6)$ truncation error of the difference schemes employed is retained. The given pde is discretised on a three-dimensional grid and the set of difference equations is formulated as a linear system of matrix equations whose solution is found by a suitable decomposition of unknows based on knowledge of the eigenvalues and eigenvectors of simple tridiagonal matrices. A hint for extending the applicability of the method—by means of a coordinate transformation—in cylindrical domains with an annular cross section, is also given.

INTRODUCTION

Interest in solving elliptic pdes in 3D rectangular parallelepipeds—at least from the point of view of applied sciences—is due among other reasons to models which simulate a number of processes in steady state heat conduction or in electrical potential problems of box-shaped regions, in incompressible fluid dynamics [9], in some natural processes in the atmosphere [13], as well as in certain problems of semi-bounded plasmas [6].

However, even for elliptic pdes in three-dimensional domains more complicated than rectangular parallelepipeds, a class of solution methods known as "capacitance matrix methods" uses as an intermediate step the solution of the original pde in a 3D rectangular parallelepiped containing the domain [10].

0021-9991/90 $3.00

Since the computed solution must be a good approximation of the exact one, it is obvious that for economy reasons one should use the highest possible order of a difference scheme.

For the classical 5-point approximation related to $O(h^4)$ truncation error, a critical survey of a number of direct methods for the solution of the Poisson equation in a plane is contained in [12]. Direct methods for solving Poisson equation in 3D rectangular parallelepipeds—using 7-point approximation schemes—are presented in [14].

Another approach in solving the system of difference equations related to Poisson pde in 3D boxes, would be to reduce the problem to a number of Helmholtz's equations in two dimensions, e.g., considering the Fourier transform of the original equation with respect to z. The next step would be to use a direct method—see e.g., [11] in which a method with a fourth-order accuracy is discussed—for solving numerically the system of difference equations related to each of the "transformed" Helmholtz's equations in two variables and then employ an additional Fourier transform to find the desired solution.

The method presented in this paper for the direct solution of Helmholtz's pdes (with coefficients dependent on z) in 3D rectangular parallelepipeds is based on the formulation of the set of difference equations as a linear system of matrix equations having as unknowns a number of two-dimensional matrices. The structure of this system is such that for Dirichlet boundary conditions or for periodic solutions of the problem, the local truncation error is of the $O(h^6)$ type, whereas for Neumann conditions is of the $O(h^3)$ type.

For solving the system of matrix equations in question, a series of trigonometric transforms is applied on the unknown matrices as well as on the data thus leading to an ensemble of "transformed" tridiagonal systems. Finally, another series of trigonometric transforms applied to the solution of the above tridiagonal systems results in the discrete solution of the original pde.

Structure of the Difference Equations

The second spatial derivatives of

$$k_x(z) \cdot \partial^2\varphi/\partial x^2 + k_y(z) \cdot \partial^2\varphi/\partial y^2 + k_z \cdot \partial^2\varphi/\partial z^2 + \sigma(z) \cdot \varphi = f(x, y, z) \qquad (1)$$

can be approximated at a point (x, y, z) either as a linear combination of φ-values (conventional method) or as a linear combination of both φ-values and φ-derivatives at neighbouring points (Hermite-type method, Mehrstellenverfahren).

For discretising (1), a three-dimensional grid with spacings h_x, h_y, h_z which covers the rectangular region $a \times b \times c$ is established and an interior node (i, j, k) is considered whereas the total number of interior points relative to directions x, y, and z is equal to l, m, and n, respectively.

For second derivatives approximation at the interior grid point (i, j, k), Hermite-

type schemes (Mehrstellenverfahren) which are related to $O(h^6)$ truncation error, are employed.

Based on the approach presented in [9] and after making a more or less obvious extension of the scope of the analysis, one concludes that there exists a linear relation between the values of $k_x(z)\cdot\partial^2\varphi/\partial x^2+k_y(z)\cdot\partial^2\varphi/\partial y^2+k_z\cdot\partial^2\varphi/\partial z^2$ and φ at the 27 grid nodes around (i, j, k). However, utilising the pde (1) itself, it is seen that $k_x(z)\cdot\partial^2\varphi/\partial x^2+k_y(z)\cdot\partial^2\varphi/\partial y^2+k_z\cdot\partial^2\varphi/\partial z^2$ is in effect equal to $f(x, y, z)-\sigma(z)\cdot\varphi$

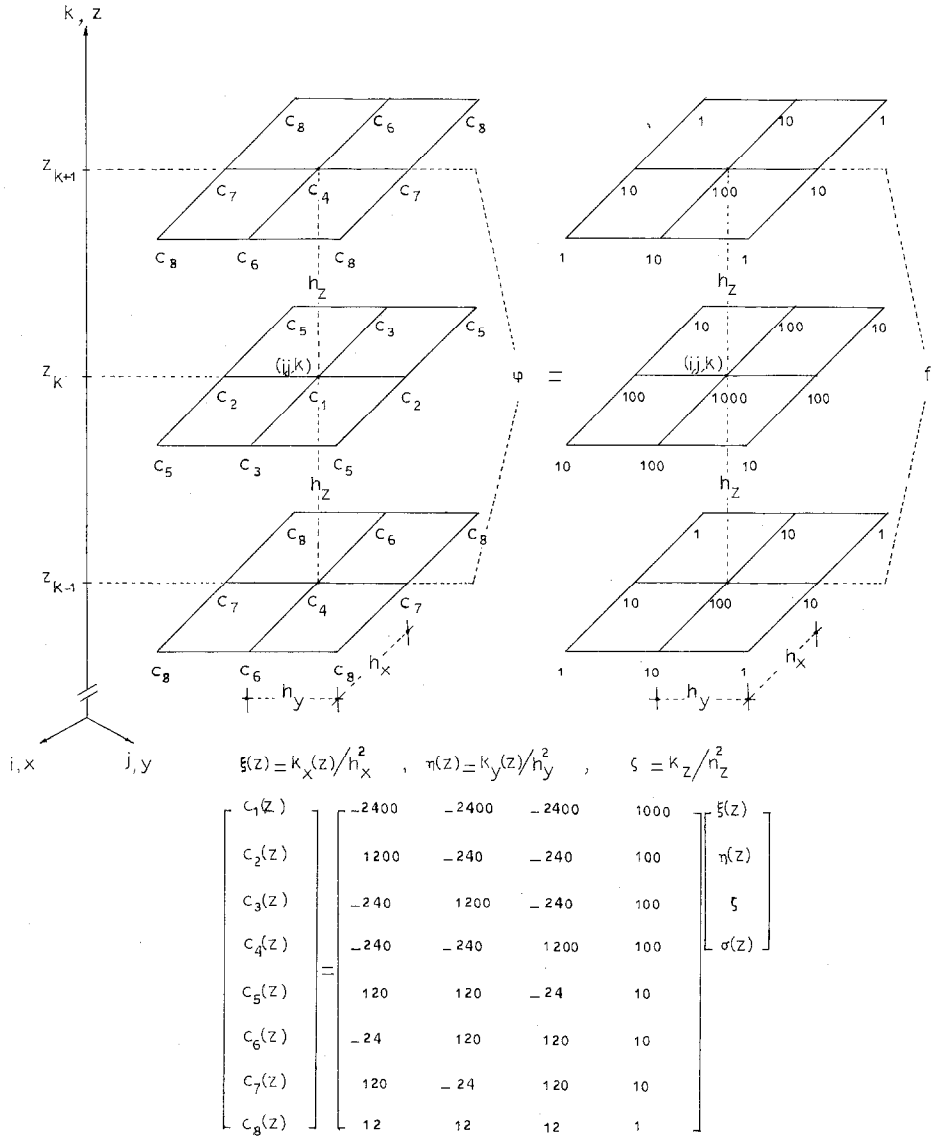

FIGURE 1

and consequently the previous linear relation between φ and $k_x(z)\cdot\partial^2\varphi/\partial x^2 + k_y(z)\cdot\partial^2\varphi/\partial y^2 + k_z\cdot\partial^2\varphi/\partial z^2$ is finally transformed to a linear relation between φ- and f-values at the 27 nodes around (i, j, k). This latest relation is expressed in schematic form by Fig. 1.

The formulation of the ensemble of these difference equations as a compact linear system of matrix equations with two-dimensional matrices as unknows is next presented in some detail for Dirichlet boundary conditions. This case, apart from constituting a familiar type of an elliptic boundary value problem, also exhibits all the essential features of the method.

However, the modifications required for problems with periodic solutions or Neumann conditions will also be indicated.

Compact Formulation of the Difference Equations

In all cases of forming the difference equations of the problem as a system of matrix equations, it is assumed that a 3D grid with steps h_x, h_y, h_z and with l, m, n interior nodes concerning the directions x, y, and z, respectively, has been established. The compact formulation emerges after the difference equation of the node ijk (see Fig. 1) is written in such a way, that the introduction of suitable tridiagonal or almost tridiagonal matrices for expressing the ensemble of these equations is in effect suggested by this form of writing.

More specifically, one can see that the linear relation connecting the 27 node values around ijk (see Fig. 1) can be written

$$
\begin{aligned}
&c_6(z_{k-1})\,\varphi_{i-1jk-1} + c_4(z_{k-1})\,\varphi_{ijk-1}/2 + c_6(z_{k-1})\,\varphi_{i+1jk-1}\\
&\quad + c_8(z_{k-1})[\varphi_{i-1j-1k-1} + \varphi_{i+1j-1k-1}]\\
&\quad + c_8(z_{k-1})[\varphi_{i-1j+1k-1} + \varphi_{i+1j+1k-1}]\\
&\quad + \varphi_{ij-1k-1}\cdot c_7(z_{k-1}) + \varphi_{ijk-1}\cdot c_4(z_{k-1})/2 + \varphi_{ij+1k-1}\cdot c_7(z_{k-1})\\
&\quad + c_3(z_k)\,\varphi_{i-1jk} + c_1(z_k)\,\varphi_{ijk}/2 + c_3(z_k)\,\varphi_{i+1jk}\\
&\quad + c_5(z_k)[\varphi_{i-1j-1k} + \varphi_{i+1j-1k}] + c_5(z_k)[\varphi_{i-1j+1k} + \varphi_{i+1j+1k}]\\
&\quad + \varphi_{ij-1k}\cdot c_2(z_k) + \varphi_{ijk}\cdot c_1(z_k)/2 + \varphi_{ij+1k}\cdot c_2(z_k)\\
&\quad + c_6(z_{k+1})\,\varphi_{i-1jk+1} + c_4(z_{k+1})\,\varphi_{ijk+1}/2 + c_6(z_{k+1})\,\varphi_{i+1jk+1}\\
&\quad + c_8(z_{k+1})[\varphi_{i-1j-1k+1} + \varphi_{i+1j-1k+1}]\\
&\quad + c_8(z_{k+1})[\varphi_{i-1j+1k+1} + \varphi_{i+1j+1k+1}\\
&\quad + \varphi_{ij-1k+1}\cdot c_7(z_{k+1}) + \varphi_{ijk+1}\cdot c_4(z_{k+1})/2 + \varphi_{ij+1k+1}\cdot c_7(z_{k+1})\\
&= \text{Linear combination of } f\text{-values around } ijk. \qquad (2)
\end{aligned}
$$

The structure of the already mentioned tridiagonal matrices depends on the boundary conditions and is given next for Dirichlet, periodic, and Neumann conditions.

Dirichlet Boundary Conditions

The $l \times m \times n$ unknowns of the problem are arranged in n rectangular matrices ϕ_k $(k = 1, 2, ..., n)$ with l rows and m columns, where ϕ_k contains the values of φ's at the "level" z_k $(z_k = (k-1) \cdot h_z)$. For Dirichlet boundary conditions at the faces of the parallelepiped the matrices ϕ_0 and ϕ_{n+1} contain the known φ-values at the boundaries $z = z_0$ and $z = z_0 + c$. The linear combinations of the f-values at each of the $l \times m$ nodes of "level" z_k have been arranged in a $l \times m$ matrix RHS_k $(k = 1, 2, ..., n)$, where the first and last rows as well as the first and last columns have "incorporated" the boundary values of φ at the faces which are perpendicular to x and y, respectively.

Using the explicit form (2) of the typical difference equation for the ijk node, one can verify that the $l \times m \times n$ equations (2) associated with the "level" z_k can be written as a matrix equation of the type,

$$\begin{aligned} & A_l[c_4(z_{k-1})/2,\ c_6(z_{k-1})] \cdot \phi_{k-1} \\ & \quad + c_8(z_{k-1}) \cdot A_l[0, 1] \cdot \phi_{k-1} \cdot A_m[0, 1] \\ & \quad + \phi_{k-1} \cdot A_m[c_4(z_{k-1})/2,\ c_7(z_{k-1})] \\ & \quad + A_l[c_1(z_k)/2,\ c_3(z_k)] \cdot \phi_k \\ & \quad + c_5(z_k) \cdot A_l[0, 1] \cdot \phi_k \cdot A_m[0, 1] \\ & \quad + \phi_k \cdot A_m[c_1(z_k)/2,\ c_2(z_k)] \\ & \quad + A_l[c_4(z_{k+1})/2,\ c_6(z_{k+1})] \cdot \phi_{k+1} \\ & \quad + c_8(z_{k+1}) \cdot A_l[0, 1] \cdot \phi_{k+1} \cdot A_m[0, 1] \\ & \quad + \phi_{k+1} \cdot A_m[c_4(z_{k+1})/2,\ c_7(z_{k+1})] = \mathrm{RHS}_k. \end{aligned} \tag{3}$$

In the above relation the symbols $A_l[\ ,\ ]$ and $A_m[\ ,\ ]$ denote symmetric tridiagonal matrices of order l and m, respectively, which are defined as follows:

$A_v[p, q]$ = a symmetric tridiagonal matrix of order v with all the diagonal elements equal to p and all the non-zero off-diagonal ones equal to q.

For $k = 1, 2, ..., n$ the above matric equations constitute a matrix system with n two-dimensional matrices $\phi_1, \phi_2, ..., \phi_n$ as unknowns, whose solution will be examined in the next section. Such an approach of compact formulation of a set of difference equations as a matrix equation has been used for Poisson's equation in a rectangle [1]. In this context an equation of the type $A_l\phi + \phi A_m = \mathrm{RHS}$ is proved to be equivalent to the set of difference equations which approximate the original pde using the conventional 5-point difference scheme with $O(h^4)$ truncation error.

Periodic Boundary Conditions

For the case of periodic solutions, e.g., with respect to x-direction and Dirichlet conditions with respect to y and z, a fictitious set of nodes at the plane $x = x_0 - h_x$ is considered and the basic difference equation (2) is written for the nodes $0jk$ which in effect constitute the intersection of the plane $x = x_0$ with the "level" z_k. In the present case, due to the periodicity of the solution with respect to x-direction, the role of φ_{-1jk}'s is played by the φ_{ljk}'s which lie in the intersection of the plane $x = x_0 + a - h_x$ and the "level" z_k. The matrix of unknows ϕ_k at the "level" z_k still has m columns but the number of its rows equals now $l+1$ and the same holds for the matrices RHS_k which contain the values of f. The appearance of an additional line in matrices ϕ_k and RHS_k in conjunction with the fact that all the φ_{-1jk}-values in (2) should be replaced by φ_{ljk}-values suggests that the square matrices of the type $A_l[\ ,\ ]$ in (3) should be replaced by square matrices of the type $B_{l+1}[\ ,\ ]$ whose structure is defined as follows:

$B_v[p, q]$ = a matrix of order v differing from the already introduced symmetric matrix $A_v[p, q]$ in that its uppermost NE and its lowest SW corner elements equal q instead of being zero.

The form of the matrix equation (3) for the above special type of periodic conditions emerges after the matrices $A_l[\ ,\ ]$ have been replaced by $B_{l+1}[\ ,\ ]$ matrices provided, of course, that both ϕ_k and RHS are now $(l+1)\times m$ matrices.

Exactly the same kind of arguments can lead to a compact matrix formulation of the difference equations for periodic solutions with respect to y only, where instead of premultiplications of ϕ's (which are now $l\times(m+1)$ matrices) by B-matrices and postmultiplications by A-matrices, one has exactly the reverse. For periodic solutions with respect to both the x and y directions, the ϕ's are pre- and postmultiplied by B-matrices only. Finally the periodicity of the solution with respect to z, results in a structure modification of the first and last matrix equations of the system. The first one now relates the matrices ϕ_0, ϕ_1, and ϕ_n, while the last one the matrices ϕ_n, ϕ_{n-1}, and ϕ_0.

Neumann Boundary Conditions

When Neumann conditions, e.g., at the plane $x = x_0$ occur, whereas at all the other faces of the parallelepiped the conditions are of the Dirichlet type, a fictitious set of nodes at the plane $x = x_0 - h_x$ is again considered and the basic difference equation (2) is written for the nodes $0jk$ common to the plane $x = x_0$ and the "level" z_k. If now a conventional approximation scheme of the type

$$\varphi_{-1jk} = \varphi_{1jk} - 2h_x \partial\varphi_{0jk}/\partial x + O(h^3) \tag{4}$$

is applied at the node $0jk$, it is seen that the role of φ_{-1jk} in (2) can be played by φ_{1jk} with the additional restriction that the value of the right-hand side associated with $0jk$ will be modified due to the presence of the quantities $2h_x\partial\varphi_{0jk}/\partial x$. The consequences of the above changes in (2) will be to double the coefficients of the

close-to-boundary nodes $1jk$, to make both the matrices ϕ_k and RHS_k have $(l+1)$ rows and m columns, and finally to modify the elements of the first row of RHS_k due to the presence of $2h_x \cdot \partial\varphi_{0jk}/\partial x$. The above doubling of coefficients of the φ_{1jk}-values for the difference equations which are associated with the $0jk$ nodes suggests that the square matrix $A_l[\ ,\ ]$ in (3) should be now replaced by square matrices of the type $C^{\sqcup}_{l+1}[\ ,\ ]$ whose structure is defined as follows:

$C^{\sqcup}_{\nu}[p, q]$ = a tridiagonal matrix of order ν differing from $A_\nu[p, q]$ only in the value of its uppermost off-diagonal element in the first row which now equals $2q$ instead of q.

For Neumann conditions at $x = x_0 + a$, the ϕ's are now premultiplied by matrices of the type $C^{L}_{l+1}[\ ,\ ]$ which are defined as follows:

$C^{L}_{\nu}[p, q]$ = a tridiagonal matrix of order ν differing from $A_\nu[p, q]$ only in the value of its lowest off-diagonal element in the last row which equals $2q$ instead of q.

Finally, for the case of Neumann conditions at faces normal to the y-direction the ϕ's will be postmultiplied by transpose C-type matrices instead of being premultiplied by such matrices, while for Neumann conditions at faces normal to z, the first and last matrix equation of the system will be suitably modified.

Although the formulation of the ensemble of difference equations as a system of matrix equations with $O(h^3)$ truncation error makes possible the use of a direct method for the numerical solution of problems with Neumann conditions, it is known that the overall accuracy of the results is poor [11]. However, for problems in which instead of known values of first-order derivatives at the boundaries these are in effect symmetry planes of the solution—e.g., when $\varphi_{-1jk} = \varphi_{1jk}$—the structure of the difference equations is identical to that referred to homogeneous Neumann conditions at the boundaries in question. This feature allows the direct solution algorithm presented in the next section to be used for such problems whose discrete analog is expressed by approximation schemes with $O(h^6)$ rather than $O(h^3)$ truncation error.

Before closing this section, some comments regarding the above compact matrix formulation (CMF) and the tensor product formulation (TPF) [7] seem to be necessary.

For k_x, k_y, and σ constants and when a 7-point difference approximation scheme is used, the direct solution algorithms related to CMF which are introduced in the next section, can be transferred to TPF. In addition, for the case considered—i.e., for 7-point difference approximation schemes—TPF leads to coefficient matrices which have a symmetric block tridiagonal structure with off-diagonal "elements" equal to unit matrices. This feature permits the efficient use of cyclic reduction methods for the solution of the difference equations and even more to the combination of such methods with fast Fourier transforms to speed up the solution process [5].

For the Helmholtz's type pde's considered in this paper when k_x, k_y, and σ are constants, a TPF of the difference equations for the 27-point approximation schemes used is still possible. The coefficient matrix is now expressed apart from terms, e.g., of the type $I_m \otimes A_l + A_m \otimes I_l$ also by terms of the form $A_m \otimes A_l$. Actually a TPF for the set of difference equations for $\nabla^2 \varphi = f$ and a HODIE approximation scheme with $h_x = h_y = h_z$ already exists [2, 8]. It is to be noted that the employment of a direct solution algorithm based on the existence of such a formulation resulted in considerable computational savings when the solution of a Poisson equation with a 27-point HODIE approximation scheme in the unit cube was considered [8].

However, when k_x and k_y depend on z while σ is not constant but is dependent on z, it seems difficult to find a TPF for the ensemble of the difference equations of (1) corresponding to CMF of the present section and related to high-order Hermite type approximation schemes. Even if some kind of TPF for this set of the difference equations is possible, it is not certain that this TPF will be of such a character that one of the tensor product forms introduced in [7] could be employed and a corresponding solution algorithm could be constructed. On the other hand, the CMF for the difference equations referring to (1) is related in a rather unambiguous way to the direct solution algorithm constructed for their solution as it is explained in the next section.

Solution of the System of Matrix Equations

The discussion in the last section shows that the system of matrix equations for the various boundary conditions has a block tridiagonal structure where the kth order equation is of the form

$$R_{k-1}\phi_{k-1} + P_{k-1}\phi_{k-1}Q_{k-1} + \phi_{k-1}S_{k-1} + R_k\phi_k + P_k\phi_k Q_k + \phi_k S_k + R_{k+1}\phi_{k+1} + P_{k+1}\phi_{k+1}Q_{k+1} + \phi_{k+1}S_{k+1} = \mathrm{RHS}_k \tag{5}$$

while the first and last equation usually contain two terms of the type $R\phi + P\phi Q + \phi S$. The matrices R, P, Q and S will be of the A, B, or C type according to the kind of boundary conditions prevailing at the faces of the parallelepiped. For example, for Dirichlet boundary conditions at all faces, the above equation (5) is identical to (3).

The direct solution algorithm for solving the system of matrix equations, is based on the fact that *all* the matrices which premultiply the unknown ϕ's have a matrix U of right eigenvectors in common and *all* the matrices postmultiplying ϕ's have some other matrix V as their common matrix of right eigenvectors.

The algorithm in question, in essence, is a rather evident extension of the so-called "irrational method" related to the solution of the matrix equation $R\phi + \phi S = \mathrm{RHS}$ [1]. This "irrational method" has already been employed for solving this latest type of matrix equation which constitutes the compact form of the difference equations when approximating Poisson pdes in rectangles by conventional 5-point schemes [1].

However, before the method can be applied to (5), one must define a number of additional diagonal matrices which in effect are the matrices of eigenvalues of R, P, Q, and S.

These definitions run as follows:

$$\begin{aligned}
\Lambda_k^R &= \text{diagonal matrix of the eigenvalues } d_{(k)i}^R \text{ of } R_k \text{ which premultiplies } \phi_k \\
\Lambda_k^P &= \text{diagonal matrix of the eigenvalues } d_{(k)i}^P \text{ of } P_k \text{ which premultiplies } \phi_k \\
M_k^Q &= \text{diagonal matrix of the eigenvalues } g_{(k)i}^Q \text{ of } Q_k \text{ which postmultiplies } \phi_k \\
M_k^S &= \text{diagonal matrix of the eigenvalues } g_{(k)i}^S \text{ of } S_k \text{ which postmultiplies } \phi_k.
\end{aligned} \tag{6}$$

In the above definitions the number of different values which k can assume varies from n to $n+2$ according to the type of boundary conditions.

The basic step in solving the ensemble of matrix equations (5), is to premultiply each equation by U^{-1} and then postmultiply it by V.

After these multiplications have taken place, the modified form of (5) is

$$\begin{aligned}
&U^{-1}R_{k-1}\phi_{k-1}V + U^{-1}P_{k-1}\phi_{k-1}Q_{k-1}V + U^{-1}\phi_{k-1}S_{k-1}V + U^{-1}R_k\phi_k V \\
&\quad + U^{-1}P_k\phi_k Q_k V + U^{-1}\phi_k S_k V + U^{-1}R_{k+1}\phi_{k+1}V \\
&\quad + U^{-1}P_{k+1}\phi_{k+1}Q_{k+1}V + U^{-1}\phi_{k+1}S_{k+1}V = U^{-1}\,\mathrm{RHS}_k\, V.
\end{aligned} \tag{6a}$$

This latest equation can be written in an equivalent form as

$$\begin{aligned}
&(U^{-1}R_{k-1}U)(U^{-1}\phi_{k-1}V) + (U^{-1}P_{k-1}U)(U^{-1}\phi_{k-1}V)(V^{-1}Q_{k-1}V) \\
&\quad + (U^{-1}\phi_{k-1}V)(V^{-1}S_{k-1}V) + (U^{-1}R_kU)(U^{-1}\phi_kV) \\
&\quad + (U^{-1}P_kU)(U^{-1}\phi_kV)(V^{-1}Q_kV) + (U^{-1}\phi_kV)(V^{-1}S_kV) \\
&\quad + (U^{-1}R_{k+1}U)(U^{-1}\phi_{k+1}V) + (U^{-1}P_{k+1}U)(U^{-1}\phi_{k+1}V)(V^{-1}Q_{k+1}V) \\
&\quad + (U^{-1}\phi_{k+1}V)(V^{-1}S_{k+1}V) = U^{-1}\,\mathrm{RHS}_k\, V.
\end{aligned} \tag{7}$$

However, since all the matrices R and P have a common matrix U of right eigenvectors and all the matrices Q and S have also some other matric V of right eigenvectors in common, it follows that in (7) all matrices R and P as well as Q and S have been transformed by similarity transformations to diagonal ones.

In other words, Eq. (7) is equivalent to

$$\begin{aligned}
&\Lambda_{k-1}^R\Psi_{k-1} + \Lambda_{k-1}^P\Psi_{k-1}M_{k-1}^Q + \Psi_{k-1}M_{k-1}^S + \Lambda_k^R\Psi_k + \Lambda_k^P\Psi_kM_k^Q \\
&\quad + \Psi_kM_k^S + \Lambda_{k+1}^R\Psi_{k+1} + \Lambda_{k+1}^P\Psi_{k+1}M_k^Q + \Psi_{k+1}M_{k+1}^S = U^{-1}\,\mathrm{RHS}_k\, V,
\end{aligned} \tag{8}$$

where the definition $\Psi_k = U^{-1}\phi_k V$ has already been used.

One considers next the equality of the two ijth elements of both the left- and right-hand members of (8). Due to the diagonal form of the A and M matrices, this equality—by means of the definitions (6)—leads finally to a relation of the following type:

$$\begin{aligned}
&[d^R_{(k-1)i} + d^P_{(k-1)i} \cdot g^Q_{(k-1)j} + g^S_{(k-1)j}] \cdot (\Psi_{k-1})_{ij} \\
&\quad + [d^R_{(k)i} + d^P_{(k)i} \cdot g^Q_{(k)j} + g^S_{(k)j}] \cdot (\Psi_k)_{ij} \\
&\quad + [d^R_{(k+1)i} + d^P_{(k+1)i} \cdot g^Q_{(k+1)j} + g^S_{(k+1)j}] \cdot (\Psi_{k+1})_{ij} \\
&= (U^{-1}\,\mathrm{RHS}_k\, V)_{ij}. \qquad (9)
\end{aligned}$$

The set of the above equations (9), for all the admissible values of k, constitutes in effect a conventional tridiagonal system relating the scalars $(\Psi_k)_{ij}$.

The solution of a number of such conventional tridiagonal systems by well-known methods for all the ordered pairs of indices (i, j) related to unknown ψ-values, finally results in calculating the elements of all the Ψ_k matrices. For the process in question, the calculation of all the elements of each matrix Ψ_k does not take place in the same computational stage but each tridiagonal system is used for calculating a definite element ij for all the Ψ matrices. After all the matrices $\Psi_k = U^{-1}\phi_k V$ have been determined, the elements of ϕ's are computed by means of the relation $\phi_k = U\Psi_k V^{-1}$.

Obviously the already sketched algorithm would be of limited practical value if one could not easily compute the elements of U and V as well as the eigenvalues of all the matrices R, P, Q, and S. Fortunately for the A, B, and C matrices, both their eigenvalues and their right eigenvectors are given by simple formulae dependent on the matrix order as well as on the values of its non-zero elements.

A method for obtaining a formula for the eigenvalues and eigenvectors of matrices whose "internal" rows—i.e., those except the first and last—are composed of a diagonal element p and two equal off-diagonal ones q, reduces this problem to the solution of a three-term recurrence relation. The two constants of the solution are next determined in a manner consistent to the structure of the first and last rows.

A list of eigenvalues and eigenvectors for the A, B, and C-type matrices is given in the Appendix.

Some Preliminary Numerical Experiments

In this section the results of a series of numerical experiments are presented. These results indicate the main characteristics of the algorithm concerning both its processing times versus the number of equations, as well as its accuracy.

The overall accuracy characteristics of the method are more or less known, since the basic difference equations which substitute the original pde have a $O(h^6)$ trun-

cation error, thus leading to fourth-order accuracy for problems with Dirichlet boundary conditions or with periodic solutions [9].

For Neumann boundary conditions the overall accuracy is not acceptable since a number of difference equations with $O(h^3)$ truncation error participate also in forming the discrete analog of the problem. However, the algorithm can still be of use, retaining its fourth order accuracy, for problems in which a number of boundaries are in effect symmetry planes of the solution since in that case no difference equations with $O(h^3)$ truncation error come into play.

Due to the fact that the accuracy will be of the same order for all cases apart from the case of Neumann boundary conditions—for which the algorithm is unsuitable anyway—it seems reasonable to choose test problems characterised by Dirichlet boundary conditions everywhere since, apart from exhibiting the acceptable accuracy of the method, they also constitute more or less familiar problems of elliptic pdes.

Although the method has been designed for problems with k_x, k_y, and σ dependent on z, test problems with constants k_x, k_y, and σ will be examined. This is done because the coefficients of a tridiagonal system of type (9) needed to calculate a definite ij element for all the Ψ matrices, vary for the several ij-pairs even when k_x, k_y, and σ are constants. For the case of constant values for k_x, k_y, and σ, the systems in question are symmetric and of such a character that "symmetric Gaussian elimination" can be used for their solution [4]. However, in both cases, i.e., when k_x, k_y, and σ are functions of z or when k_x, k_y, and σ are constants, these coefficients are computed either by means of very simple formulae dependent on precalculated values of $c_1(z), ..., c_8(z)$ at the various z_k "levels," or by using precalculated values of the scalars d and g themselves. Actually the time needed to form the coefficients of the tridiagonal systems (9) is a very small fraction of the total time related to the solution of the difference equations after the RHS matrices have been formed irrespectively of whether k_x, k_y, and σ are constants or they depend on z.

Due to the negligible influence of the time necessary to form the coefficients of the tridiagonal systems in question on the total solution time, the processing times reported refer to the period starting immediately after all the RHS matrices have been formed and ending just before editing the final solution ϕ. These processing times depend strongly on the computer used and the efficiency of the program but also on the type of boundary conditions, since this affects the transformation steps of RHS matrices to $U^{-1} \cdot \text{RHS} \cdot V$ matrices, especially when FFT-type algorithms are employed for such transformations [3]. As a matter of fact, the variety of the FFT-type algorithms related to the several combinations of boundary conditions and the associated increasing complexity of the program as a whole dictated, in a sense, the selection of problems with Dirichlet boundary conditions everywhere as the only representatives of the method's fourth-order accuracy.

For problems in 3D rectangular parallelepipeds with Dirichlet boundary conditions everywhere, the matrices U and V are of such a character—see also Appendix—that sine transforms of RHS matrices are introduced. These sine trans-

forms are performed either using precalculated values of the elements of U and V (Program Version I), or employing FFT algorithms when each of the indices $l+1$ and $m+1$ is equal to a power of 2 (Program Version II).

Keeping in mind the basic computational characteristics of the method as these have been described so far, it seems more or less justified for the test problems to be of the following type:

Solve $\nabla^2\varphi + \sigma \cdot \varphi = f(x, y, z)$ for Dirichlet boundary conditions on the faces of a unit cube ($a=b=c=1$) having its center at the origin and edges parallel to coordinate axes, while the parameter σ is constant, taking, e.g., only one of the values 1, 0, -1.

The source term $f(x, y, z)$ is generated through the relation $f(x, y, z) = \varphi_1(x/a) \cdot \varphi_2(y/b) \cdot \varphi_3(z/c)$, where the functions φ_j ($j=1, 2, 3$) are given as follows:

$$\varphi_1(x/a) = \cos(\pi p_1 \cdot x/a) \quad \text{or} \quad \varphi_1(x/a) = \cosh(q_1 \cdot x/a)$$

$$\varphi_2(y/b) = \cos(\pi p_2 \cdot y/b) \quad \text{or} \quad \varphi_2(y/b) = \cosh(q_2 \cdot y/b)$$

$$\varphi_3(z/c) = \cos(\pi p_3 \cdot z/c) \quad \text{or} \quad \varphi_3(z/c) = \cosh(q_3 \cdot z/c)$$

$$(p_1, p_2, p_3 \quad \text{and} \quad q_1, q_2, q_3 \text{ integers}).$$

The choice of a product of three symmetric functions as the known solution of the problem aims at testing the symmetry of the discrete solution constituting a reliability test of the algorithm, as well as at reducing the workload when forming the RHS matrices or checking the accuracy of the computed solution. The numerical experiments have been performed on an AMSTRAD 1512 personal computer using a FORTRAN 77 compiler with a single precision accuracy of order 10^{-6} and the results are summarised in Tables I, II, and III.

The results for each case consist of the maximum absolute discrepancy from the known solution, while for selected groups of cases mean processing times for both the Program Versions I (no FFT algorithms used) and II (FFT algorithms used) are also recorded.

Aiming at some additional information concerning the dependence of processing times on the total number of difference equations, the test problem

$$\nabla^2\varphi = -(\pi^2/a^2 + \pi^2/b^2 + \pi^2/c^2) \cdot \cos(\pi x/a)\cos(\pi y/b)\cos(\pi z/c)$$

in 3D rectangular parallelepipeds $a \times b \times c$ for varying ratios $a:b:c$ and for several discretization steps h_x, h_y, h_z is considered. The boundary conditions on every face of the parallelepiped are as usual of the Dirichlet type, whereas the results of the computations—consisting again of the maximum absolute discrepancy from the known solution as well as the processing times for Program Versions I and II—are summarised in Table IV.

Based on the results of all the above numerical experiments, one may conclude that the method is indeed characterised by a fourth-order accuracy and that the

TABLE I

$\nabla^2\varphi = \cdots (a=b=c=1)$

	$h_x=h_y=h_z$		
	1/4	1/8	1/16
Test function φ	max(ε)	max(ε)	max(ε)
$\cosh(2x/a)\cosh(2y/b)\cosh(2z/c)$	0.22(−3)	0.14(−4)	0.37(−5)
$\cosh(2x/a)\cos(\pi y/b)\cosh(2z/c)$	0.76(−3)	0.46(−4)	0.14(−5)
$\cosh(x/a)\cos(2\pi y/b)\cosh(2z/c)$	0.34(−1)	0.20(−2)	0.12(−3)
$\cos(\pi x/a)\cos(2\pi y/b)\cosh(2z/c)$	0.28(−1)	0.16(−2)	0.99(−4)
$\cos(\pi x/a)\cos(\pi y/b)\cosh(z/c)$	0.16(−2)	0.99(−4)	0.50(−4)
$\cosh(2x/a)\cos(\pi y/b)\cos(\pi z/c)$	0.13(−2)	0.81(−4)	0.36(−5)
$\cosh(2x/a)\cos(2\pi y/b)\cos(\pi z/c)$	0.28(−1)	0.16(−2)	0.99(−5)
$\cos(\pi x/a)\cos(2\pi y/b)\cos(nz/c)$	0.23(−1)	0.14(−2)	0.83(−5)
$\cos(2\pi x/a)\cos(2\pi y/b)\cos(2\pi z/c)$	0.32(−1)	0.19(−2)	0.11(−3)
$\cosh(2x/a)\cos(\pi y/b)\cos(2\pi z/c)$	0.28(−1)	0.16(−2)	0.99(−4)
$\cosh(2x/a)\cos(2\pi y/b)\cos(2\pi z/c)$	0.36(−1)	0.20(−2)	0.13(−3)
$\cos(\pi x/a)\cos(2\pi y/b)\cos(2\pi z/c)$	0.30(−1)	0.18(−2)	0.11(−3)
Mean processing times (seconds)			
Version I	0.48 ± .02	6.63 ± .03	110.30 ± .23
Version II	0.64 ± .02	7.18 ± .03	78.36 ± .19

TABLE II

$\nabla^2\varphi + \varphi = \cdots \quad (a=b=c=1)$

	$h_x=h_y=h_z$		
Test function φ	1/4 max(ε)	1/8 max(ε)	1/16 max(ε)
$\cosh(2x/a)\cosh(2y/b)\cosh(2z/c)$	0.22(−3)	0.14(−4)	0.21(−5)
$\cosh(2x/a)\cos(\pi y/b)\cosh(2z/c)$	0.79(−3)	0.48(−4)	0.47(−5)
$\cosh(2x/a)\cos(2\pi y/b)\cosh(2z/c)$	0.35(−1)	0.21(−2)	0.13(−3)
$\cos(\pi x/a)\cos(2\pi y/b)\cosh(2z/c)$	0.29(−1)	0.17(−2)	0.10(−3)
$\cos(\pi x/a)\cos(\pi y/b)\cosh(z/c)$	0.17(−2)	0.10(−3)	0.73(−5)
$\cosh(2x/a)\cos(\pi y/b)\cos(\pi z/c)$	0.14(−2)	0.84(−4)	0.64(−5)
$\cosh(2x/a)\cos(2\pi y/b)\cos(\pi z/c)$	0.29(−1)	0.17(−2)	0.10(−3)
$\cos(\pi x/a)\cos(2\pi y/b)\cos(\pi z/c)$	0.24(−1)	0.14(−2)	0.86(−4)
$\cos(2\pi x/a)\cos(2\pi y/b)\cos(2\pi z/c)$	0.33(−1)	0.19(−2)	0.12(−3)
$\cosh(2x/a)\cos(\pi y/b)\cos(2\pi z/c)$	0.29(−1)	0.17(−2)	0.10(−3)
$\cosh(2x/a)\cos(2\pi y/b)\cos(2\pi z/c)$	0.37(−1)	0.21(−2)	0.13(−3)
$\cos(\pi x/a)\cos(2\pi y/b)\cos(2\pi z/c)$	0.31(−1)	0.18(−2)	0.11(−3)
Mean processing times (seconds)			
Version I	0.49 ± .00	6.63 ± .03	110.27 ± .24
Version II	0.64 ± .02	7.16 ± .04	78.38 ± .20

TABLE III

$\nabla^2\varphi - \varphi = \cdots \quad (a = b = c = 1)$

Test function φ	$h_x = h_y = h_z$ 1/4 max(ε)	1/8 max(ε)	1/16 max(ε)
$\cosh(2x/a)\cosh(2y/b)\cosh(2z/c)$	0.21(−3)	0.13(−4)	0.19(−5)
$\cosh(2x/a)\cos(\pi y/b)\cosh(2z/c)$	0.73(−3)	0.45(−4)	0.25(−5)
$\cosh(2x/a)\cos(2\pi y/b)\cosh(2z/c)$	0.33(−1)	0.19(−2)	0.12(−3)
$\cos(\pi x/a)\cos(2\pi y/b)\cosh(2z/c)$	0.27(−1)	0.16(−2)	0.97(−4)
$\cos(\pi x/a)\cos(\pi y/b)\cosh(z/c)$	0.16(−2)	0.97(−4)	0.58(−5)
$\cosh(2x/a)\cos(\pi y/b)\cos(\pi z/c)$	0.13(−2)	0.79(−4)	0.44(−5)
$\cosh(2x/a)\cos(2\pi y/b)\cos(\pi z/c)$	0.27(−1)	0.16(−2)	0.97(−4)
$\cos(\pi x/a)\cos(2\pi y/b)\cos(\pi z/c)$	0.23(−1)	0.13(−2)	0.81(−4)
$\cos(2\pi x/a)\cos(2\pi y/b)\cos(2\pi z/c)$	0.32(−1)	0.18(−2)	0.11(−3)
$\cosh(2x/a)\cos(\pi y/b)\cos(2\pi z/c)$	0.27(−1)	0.16(−2)	0.97(−4)
$\cosh(2x/a)\cos(2\pi y/b)\cos(2\pi z/c)$	0.35(−1)	0.20(−2)	0.12(−3)
$\cos(\pi x/a)\cos(2\pi y/b)\cos(2\pi z/c)$	0.29(−1)	0.17(−2)	0.11(−3)
Mean processing times (seconds)			
Version I	0.49 ± .00	6.63 ± .03	110.28 ± .27
Version II	0.65 ± .01	7.18 ± .03	78.35 ± .19

TABLE IV

$\nabla^2\varphi = \cdots$ (Test function: $\varphi = \cos(\pi x/a)\cos(\pi y/b)\cos(\pi z/c)$)

$a:b:c$	$h_x = h_y = h_z$	Number of unknowns $(l \cdot m) \times n$	Max(ε)	Processing times (seconds) Version I	Version II
1:1:1.0	1/4	27	0.16(−2)	0.43	0.63
1:1:1.0	1/8	343	0.99(−4)	6.64	7.14
1:1:1.0	1/16	3375	0.50(−5)	109.85	78.54
1:1:2.0	1/4	63	0.16(−2)	0.87	1.26
1:1:2.0	1/8	735	0.89(−4)	13.78	14.99
1:1:2.0	1/16	6975	0.46(−5)	226.18	161.42
1:1:3.0	1/4	99	0.15(−2)	1.26	1.86
1:1:3.0	1/8	1127	0.95(−4)	20.92	22.90
1:1:3.0	1/16	10575	0.52(−5)	342.46	244.26
4:4:0.5	1/8	2883	0.16(−2)	180.26	79.80
4:4:1.0	1/4	675	0.15(−2)	22.79	16.31
4:4:1.0	1/8	6727	0.90(−4)	410.18	180.54
4:4:1.5	1/4	1125	0.25(−3)	37.29	26.69
4:4:1.5	1/8	10571	0.18(−4)	642.46	282.26

maximum absolute error increases when one passes from smooth solutions φ to solutions with components which oscillate around a mean value—e.g., like $\cos(2\pi x/a)$—or with components which cause the sixth derivative of φ to take large values—e.g., like $\cosh(2x/a)$.

The use of FFT algorithms (Program Version II) seems justified when $l+1$ or $m+1$ is greater than or equal to 16. When both $l+1$ and $m+1$ are equal to 16 the ratio of the processing times of Program Version I, (no FFT algorithms used) to those referring to Program Version II, is about 1.4 while for $l+1$ and $m+1$ equal to 32 the value of this ratio increases to 2.3.

A preliminary estimation of the dependence of processing times T_1 and T_2 for Program Versions I and II, respectively, on the number of the difference equations $(l \cdot m) \times n$, can be worked out using the data in Table IV. From a least squares analysis one has

$$T_1 = (113.13) \times [(l/15) \cdot (m/15)]^{1.32} \cdot (n/15)^{0.92} \qquad (R^2 = 0.95)$$

$$T_2 = (\ 78.76) \times [(l/15) \cdot (m/15)]^{1.06} \cdot (n/15)^{0.95} \qquad (R^2 = 0.92).$$

The numbers 113.13 and 78.76 denote the approximate processing times when $l = m = n = 15$ for Program Versions I and II, respectively, using an AMSTRAD 1512 personal computer with a FORTRAN 77 compiler associated with a 10^{-6} single precision accuracy. Obviously these parameters will vary according to the hardware characteristics of the computer used as well as according to the possibilities of the compiler for the programming language employed.

Some Aspects of the Numerical Performance of the Algorithm

Although the solution algorithm outlined in the previous sections has been based on a system of matrix equations, it is evident that the compact matrix formulation (CMF) of the problem has as its counterpart a conventional formulation (CF) of the system of difference equations. For example, in the case of Dirichlet conditions this CF is related to a "big" symmetric matrix $\tilde{D}$ operating on a vector of $l \times m \times n$ unknowns. One can prove in a rather direct but tedious fashion that $\tilde{D}$ is a block tridiagonal matrix of order $l \times m \times n$ whose "elements" are themselves block tridiagonal matrices of order $l \times m$, while the "elements" of these latest matrices are conventional tridiagonal matrices of order l.

In trying to determine error bounds of the solution in the case of CMF algorithms, as well as their error-propagating features, one could perform an error analysis which combined with the built-in accuracy characteristics of the machine would classify them with respect to numerical reliability. For such an analysis, tools and techniques associated with conventional forms of matrix computations could be employed if each step of the CMF algorithm is "translated" into the equivalent form of the CF algorithm.

However, such a detailed and probably difficult error analysis will not be carried out in this paper.

Instead, the numerical reliability of the algorithm will be judged according to the value of the spectral condition number (SCN) of the "big" matrix which for a direct method for solving systems of linear equations represents a more or less acceptable approach.

The spectral condition number (SCN) of a matrix $\tilde{B}$ is defined as

$$\mathrm{SCN}(\tilde{B})^2 = \max |\lambda(\tilde{B}'B)| / \min |\lambda(\tilde{B}'B)|, \tag{10a}$$

where $'$ denotes the transpose matrix and $\lambda(\tilde{B}'\tilde{B})$ an eigenvalue of $\tilde{B}'\tilde{B}$. When $\tilde{B}$ is symmetric, (10a) is equivalent to

In either case of CMF or CF algorithms, the eigensolutions constitute a basis for expressing the ensemble of unknowns and this intrinsic property of the model's eigensystem does not depend on any particular formulation. This suggests that the correspondence between the two representations of the eigensystem in compact and conventional form, might be used to determine the eigenvalues and eigenvectors of the "big" matrix by taking advantage of the solution method associated with the CMF algorithm.

In studying such a correspondence, an eigenvector of the above mentioned "big" matrix $\tilde{D}$ associated with a certain eigenvalue λ is regarded not as a column of $l \times m \times n$ elements but as a sequence of n matrices E_k $(k = 1, 2, ..., n)$, each one having l rows and m columns.

In order to determine the elements of this eigenvector, one should use the matrix equation (3) itself and put E_k and $\lambda \cdot E_k$ in place of the matrices ϕ_k and RHS_k, respectively. One next sets E_k equal to $w_{kij}(U_i V_j')$, where w_{kij} is a scalar depending only on k for a definite pair of indices ij while the symbols U_i and V_j' have the following meanings:

U_i = the common ith eigencolumn of all the A matrices which in the matrix system (3) premultiply the unknown ϕ's.

V_j' = the common jth eigenrow (left-hand eigenvector) of all the A matrices which in the matrix system (3) postmultiply the unknown ϕ's.

With the above choice for the matrices E_k, one can prove by direct substitution in (3) that the conventional system relating all the ij elements of the emerging n matrix equations has exactly the form (9) with $(\Psi_{k-1})_{ij}$, $(\Psi_k)_{ij}$, $(\Psi_{k+1})_{ij}$ replaced by w_{k-1ij}, w_{kij}, w_{k+1ij}, respectively, while $\lambda \cdot w_{kij}$ takes the place of $(U^{-1}\,\mathrm{RHS}_k\,V)_{ij}$. This, in fact, means that the n eigencolumns and the corresponding n eigenvalues of the conventional system (9), when used in conjunction with U_i and V_j', lead to the n eigencolumns and the corresponding eigenvalues of the "big" matrix $\tilde{D}$ which

are associated with a definite pair of indices ij. Proceeding in this way, all the $l \times m \times n$ eigencolumns and eigenvalues of the "big" matrix can be found.

The above approach for determining the eigensystem of the "big" matrix extends the scope of a similar approach presented in [1] which concerns the non-zero solution of the matrix equation $A\phi + \phi B = \lambda \cdot \phi$ and its correspondence to the eigensystem of the "big" conventional matrix.

Similar considerations applied to problems with periodic or Neumann conditions help to determine the eigensystem of these problems as well.

For k_x, k_y, and σ constants and Dirichlet conditions on the faces $z = z_0$ and $z = z_0 + c$, the matrix of the conventional system (9) is symmetric and conscquently its eigenvalues are real.

More precisely, one can prove, after applying the already sketched procedure, that

$$\begin{aligned}\lambda_{kij}(\tilde{D}) = c_1 &+ 2\{c_3 \cos[i\pi/(l+1)] + c_2 \cos[j\pi/(m+1)] \\ &+ c_4 \cos[k\pi/(n+1)]\} \\ &+ 4\{c_5 \cos[i\pi/(l+1)] \cos[j\pi/(m+1)] \\ &+ c_7 \cos[j\pi/(m+1)] \cos[k\pi/(n+1)] \\ &+ c_6 \cos[k\pi/(n+1)] \cos[i\pi/(\lambda+1)\} \\ &+ 8c_8 \cos[i\pi/(l+1)] \cos[j\pi/(m+1)] \cos[k\pi/(n+1)]. \qquad (11)\end{aligned}$$

In the above formula, $c_1, \ldots, c_8$ (see Fig. 1) are constants since k_x, k_y, and σ are now independent of z. When periodic or Neumann conditions on the faces $z = z_0$ and/or $z = z_0 + c$ prevail then for k_x, k_y, and σ constants, the eigenvalues of the conventional system (9) are still real, since the coefficient matrix of the system in question is now of the type B or C and it is known (see Appendix) that the eigenvalues of such matrices are real.

For the numerical experiments considered in this paper, the boundary conditions are of the Dirichlet type and the problems are of such a character that $\xi(z) = \eta(z) = \zeta(z) = \beta$ (=constant) while $\sigma(z)$ stands for another constant function of z taking only one of the values $-1, 0, 1$ at a time. Under the above assumptions the quantities $c_1, \ldots, c_8$ (see Fig. 1) are also constants and are given as follows:

$$\begin{aligned} c_1 &= -7200\beta + 1000\sigma \\ c_2 = c_3 = c_4 &= 720\beta + 100\sigma \\ c_5 = c_6 = c_7 &= 216\beta + 10\sigma \\ c_8 &= 36\beta + \sigma. \end{aligned} \qquad (12)$$

Using the latest expressions for $c_1, \ldots, c_8$ in (11), one can compute the value of the λ_{kij} eigenvalue of $\tilde{D}$.

The results is the following:

$$\begin{aligned}\lambda_{kij}(\tilde{D}) = & -7200\beta + 1000\sigma \\ & + 2(720\beta + 100\sigma)\{\cos[i\pi/(l+1)] + \cos[j\pi/(m+1)] \\ & + \cos[k\pi/(n+1)]\} \\ & + 4(216\beta + 10\sigma)\{\cos[i\pi/(l+1)]\cos[j\pi/(m+1)] \\ & + \cos[j\pi/(m+1)]\cos[k\pi/(n+1)] \\ & + \cos[k\pi/(n+1)]\cos[i\pi/(l+1)]\} \\ & + (36\beta + \sigma)\cos[i\pi/(l+1)]\cos[j\pi/(m+1)]\cos[k\pi/(n+1)]. \end{aligned} \tag{13}$$

When $l=m=n$, $\beta=(n+1)^2$, and $\sigma=-1, 0, 1$, i.e., for pdes of the type $\nabla^2\varphi + \sigma\cdot\varphi = f$ with Dirichlet boundary conditions on the faces of the unit cube, a simple program in FORTRAN was written in order to find SCN($\tilde{D}$) for $n+1=4$, 8, and 16. In this way the values 3.30 ± 0.10, 11.98 ± 0.39, and 46.59 ± 1.56 for $n+1=4$, 8, and 16, respectively, were determined, in which the influence of σ-values is seen to be insignificant. The above three groups of SCNs are taken as the most unfavourable error-magnifying factors of the algorithm. When these numbers are combined with the fourh-order accuracy in the right-hand side of the discrete model employed, they lead to the numbers 0.013 ($=3.30\times 0.25^4$), 0.003 ($=11.98\times 0.125^4$), and 0.0007 ($=46.59\times 0.0625^4$) for $n+1=4$, 8, and 16, respectively. These numbers are taken as representing the accuracy of the results.

For 7-point finite difference approximations and k_x, k_y, and σ constants, i.e., for the usual case of the difference equations in 3D with $O(h^4)$ truncation error, the matrix equation for Dirichlet boundary conditions which corresponds to (3) is of the following type:

$$c_4\phi_{k-1} + A_l[c_1/2, c_3]\,\phi_k + \phi_k A_m[c_1/2, c_2] + c_4\phi_{k+1} = \mathrm{RHS}_k. \tag{14}$$

The quantities c_1, c_2, c_3, c_4 (see Fig. 1) are the only ones which differ from zero and one can prove that

$$\begin{aligned} c_1 &= -2(k_z/h_z^2)\cdot[(k_x/k_z)\cdot(h_z^2/h_x^2) + (k_y/k_z)\cdot(h_z^2/h_y^2) + 1] + \sigma \\ c_2 &= (k_z/h_z^2)\cdot(k_y/k_z)\cdot(h_z^2/h_y^2) \\ c_3 &= (k_z/h_z^2)\cdot(k_x/k_z)\cdot(h_z^2/h_x^2) \\ c_4 &= k_z/h_z^2. \end{aligned} \tag{15}$$

The spectral condition number of the "big" system which corresponds to the n matrix equations (14) is again given as $\mathrm{SCN}(\tilde{D}) = \max|\lambda_{kij}(\tilde{D})|/\min|\lambda_{kij}(\tilde{D})|$, where in the present case

$$\lambda_{kij}(\tilde{D}) = c_1 + 2\{c_3\cos[i\pi/(l+1)] + c_2\cos[j\pi/(m+1)] + c_4\cos[k\pi/(n+1)]\}$$

or, when $k_x/h_x^2 = k_y/h_y^2 = k_z/h_z^2 = \beta$,

$$\lambda_{kij}(\tilde{D}) = -6\beta + \sigma + 2\beta\{\cos[i\pi/(l+1)] + \cos[j\pi/(m+1)] + \cos[k\pi/(n+1)]\}.$$

The three groups of SCNs for the case of 7-point finite difference approximation schemes when $n+1 = 4$, 8, and 16 are now represented by the numbers 5.85 ± 0.17, 25.29 ± 0.83, and 103.17 ± 3.46, respectively. The latest three groups of SCNs are again taken as the most unfavourable error-magnifying factors of the algorithm and, when combined with models of second-order accuracy in the right-hand side, lead to the numbers 0.36 $(=5.85 \times 0.25^2)$, 0.39 $(=25.29 \times 0.125^2)$, and 0.40 $(=103.17 \times 0.0625^2)$ for $n+1 = 4$, 8, and 16, respectively. These numbers represent (as in the case of models with fourth-order accuracy in the right-hand side) the accuracy of the results and it is seen that their value is roughly constant. This means that when the influence of σ on SCN is insignificant, the SCNs of 3D discrete models based on 7-point finite difference approximations are still inversely proportional to h^2, as are the SCNs of 2D models constructed by means of 5-point finite difference approximations [2].

Before closing this section it seems appropriate that one should give, at least for the examples considered in this paper, the number of floating point operations (FPOs) as a function of the indices l, m, and n whose product is equal to the number N of unknowns, so that comparisons to other algorithms can be made.

Towards this end, note that the main computational stages in solving the system of discrete equations for pde (1) when Dirichlet conditions prevail and after the n $l \times m$ matrices RHS_k $(k = 1, 2, ..., n)$ have been formed, are the following:

Stage I. Perform n matrix transforms of the type $U\,\mathrm{RHS}_k\,V$, where U and V denote symmetric orthonormal matrices of order l and m respectively, while RHS_k denotes an $l \times m$ matrix with elements dependent on the values of $f(x, y, z)$—see Fig. 1—at "level" k.

Stage II. Solve $l \times m$ tridiagonal systems of type (9) each one having n unknowns.

Stage III. Perform n matrix transforms of the type $U\Psi_k V$ with Ψ_k a $l \times m$ matrix of known elements.

When no FFT algorithms are used, the first half of each of the n matrix transforms of Stage I which leads to the $l \times m$ matrix $U\,\mathrm{RHS}_k$ requires $l \times m(l \cdot \mathrm{Muls} + (l-1) \cdot \mathrm{Adds})$ with Muls and Adds standing for multiplications and additions, respectively, when of course the elements of U have been computed.

However, when $l+1$ is a power of 2 the sine transform which gives the elements of the jth column of $U\,\mathrm{RHS}_k$ is calculated by means of a special procedure described in [3].

This is done after forming an auxiliary sequence having $l+1$ terms whose real part is composed of the differences between two consecutive odd-numbered

Each of the l terms of the sine transform which corresponds to an element of the jth column of $U\,\mathrm{RHS}_k$ is finally calculated by a linear combination of real and imaginary parts of the Fourier transform of a suitable complex sequence of length $(l+1)/2$, which is gradually formed from the previously mentioned auxiliary one.

When the butterfly scheme for computing FFTs is used, it is known that the calculation of the Fourier coefficients of an $(l+1)/2$-length sequence requires

$$(l+1)[\log_2(l+1)-1]\cdot\mathrm{Muls}+1.5(l+1)[\log_2(l+1)-1]\cdot\mathrm{Adds}$$

i.e.,

$$[(l+1)/4]\cdot\log_2[(l+1)/2]\text{ complex multiplications plus}$$

$$[(l+1)/2]\cdot\log_2[(l+1)/2]\text{ complex additions} \qquad [16]$$

Based on the procedure presented in [3] and taking into account the fact that the final expression giving the sine transform terms is composed of an antisymmetric part and a symmetric part, the total number of FPOs required to form the elements of the jth column of $U\,\mathrm{RHS}$ equals

$$(l+1)[\log_2(l+1)+1/2]\cdot\mathrm{Muls}+(l+1)[1.5\log_2(l+1)+7/2]\cdot\mathrm{Adds}. \qquad (16)$$

At this point it must be noticed that the number of FPOs per subdivision point is given in [12] as $[\log_2(l+1)-1/2]\cdot\mathrm{Muls}+[1.5\log_2(l+1)+5/2]\cdot\mathrm{Adds}$, which in effect means that the author's routine for calculating sine transforms is less efficient than the corresponding routine associated with the numerical experiments presented in that paper.

Consequently the data quoted in Tables I, II, III, and IV refer to a computer program which requires

$$(l+1)[\log_2(l+1)+0.5]\cdot\mathrm{Muls}+(l+1)[1.5\log_2(l+1)+3.5]\cdot\mathrm{Adds}$$

in order to calculate the elements of the jth column of $U\,\mathrm{RHS}_k$.

From the previous analysis one sees that the author's programme for the first half of each of the n transforms of Stage I requires $m\cdot(l+1)[\log_2(l+1)+0.5]\cdot\mathrm{Muls}+m(l+1)[1.5\log_2(l+1)+3.5]\cdot\mathrm{Adds}$ when FFT algorithms are employed.

Using exactly the same kind of arguments it is seen that the second half of each of the n transforms of Stage I which leads to the $l\times m$ matrix $(U\,\mathrm{RHS}_k)\,V$ requires $l\times m(m\cdot\mathrm{Muls}+(m-1)\cdot\mathrm{Adds})$ when no FFT algorithms are used, but $l\cdot(m+1)[\log_2(m+1)+0.5]\cdot\mathrm{Muls}+l\cdot(m+1)[1.5\log_2(m+1)+3.5]\cdot\mathrm{Adds}$ when FFT algorithms are employed.

In finding the number of floating point operations needed for Stage II, it is noticed that for Dirichlet conditions and when k_x, k_y, σ—see (1)—are constants, only two coefficients are computed for each of the n symmetric tridiagonal systems (9), since all the diagonal elements of the coefficient matrix are equal and the same applies to the non-zero off-diagonal ones.

Provided that the eigenvalues which appear in the formulas for calculating these diagonal and off-diagonal elements have been computed already (see Appendix), one needs only $2 \cdot \text{Muls} + 4 \cdot \text{Adds}$ to form the coefficient matrix for the system in question.

The solution of each of the $l \times m$ tridiagonal systems (9) which will give the n unknown quantities $(\Psi_k)_{ij}$ $(k = 1, 2, ..., n)$ requires $3(n-1) \cdot \text{Muls} + 3(n-1) \cdot \text{Adds} + (2n-1) \cdot \text{Divs}$—see [15]—so that, for Stage II, one needs $3l \times m(n-1) \cdot \text{Muls} + 3l \times m(n-1) \cdot \text{Adds} + l \times m(2n-1) \cdot \text{Divs} + 2l \times m \cdot \text{Muls} + 4l \times m \cdot \text{Adds}$ (Divs stands for divisions).

Stage III is identical to Stage I as far the number of FPOs is concerned and consequently the numbers of floating-point operations for this stage are given by the corresponding expressions for Stage I.

When either $l+1$ or $m+1$ or both are powers of 2 and FFT algorithms are applied, it is necessary to multiply the elements of $U \Psi_k V$ by $1/8(l+1)$ or $1/8(m+1)$ or $1/64(l+1)(m+1)$, respectively, to get the correct results, which obviously means that the total number of multiplications is increased by $l \times m \times n$ in these cases.

Summarizing, one can give the following expressions for the required FPOs:

$$\begin{aligned} \text{FPOs}\{\text{no FFTs}\} = (l \times m \times n) \cdot [&2l + 2m + 3 - 1/n) \cdot \text{Muls} \\ &+ (2l + 2m - 1 + 1/n) \cdot \text{Adds} + (2 - 1/n) \cdot \text{Divs}] \end{aligned} \tag{17}$$

$$\begin{aligned} \text{FPOs}\{\text{with FFTs}\} = (l \times m \times n) \cdot [&6 + (1/l + 1/m) + 2(1 + 1/l) \log_2(l+1) \\ &+ 2(1 + 1/m) \log_2(m+1) + 1/n] \cdot \text{Muls} \\ &+ (l \times m \times n) \cdot [17 + 7(1/l + 1/m) + 3(1 + 1/l) \log_2(l+1) \\ &+ 3(1 + 1/m) \log_2(m+1) + 1/n] \cdot \text{Adds} \\ &+ (l \times m \times n) \cdot [2 - 1/n] \cdot \text{Divs}. \end{aligned} \tag{18}$$

When $l \times m \times n = N^{1/3}$, (17) and (18) be written as follows:

$$\begin{aligned} \text{FPOs}\{\text{no FFTs}\} = N \cdot (4N^{1/3} + 3 - N^{-1/3}) \cdot \text{Muls} + N \cdot (4N^{1/3} - 1 + N^{-1/3}) \cdot \text{Adds} \\ + N \cdot (2 - N^{-1/3}) \cdot \text{Divs} \end{aligned} \tag{17a}$$

$$\begin{aligned} \text{FPOs}\{\text{with FFTs}\} = N \cdot [&6 + N^{-1/3} + 4(1 + N^{-1/3}) \log_2(1 + N^{1/3})] \cdot \text{Muls} \\ &+ N \cdot [17 + 15N^{-1/3} + 6(1 + N^{-1/3}) \log_2(1 + N^{1/3})] \cdot \text{Adds} \\ &+ N \cdot [2 - N^{1/3}] \cdot \text{Divs}. \end{aligned} \tag{18a}$$

Expressions (17) and (18) may be used to compare the performance of the algorithm presented in this paper to other algorithms on the basis of operation counts, although one should have in mind that in stating (17) the equality of absolute values of $\sin[ij\pi/(l+1)]$ and $\sin[i(l+1-j)\,\pi/(l+1)]$ or of $\sin[ij\pi/(m+1)]$ and $\sin[i(m+1-j)\,\pi/(m+1)]$ was not taken into account.

Other factors which influence the solution time as, e.g., the total number of data transfers required, the structure of the compiler used, as well as the computer architecture, have not been considered.

Possibility of an Extension

The solution method introduced in the preceding sections for pdes of the type (1) in 3D rectangular parallelepipeds, can also be used to solve pdes of the type

$$k(r)\cdot[(1/r)\cdot\partial(r\cdot\partial\varphi/\partial r)/\partial r]+k_\tau(r)\cdot\partial^2\varphi/\partial\tau^2 + k_z(r)\cdot\partial^2\varphi/\partial z^2+\beta(r)\cdot\varphi=f(z,\tau,r) \tag{19}$$

in cylindrical domains with an annular cross section, characterised by the relations:

$$z_0\leqq z\leqq z_0+c,\qquad 0\leqq\tau\leqq 2\pi,\qquad r_1\leqq r\leqq r_2.$$

Indeed, if the above equation will be first multiplied by r^2 and then the variable $\rho=\ln(r/r_0)$ is introduced, one can see that (19) is transformed to

$$\tilde{k}(\rho)\cdot\partial^2\varphi/\partial z^2+\tilde{k}_\tau(\rho)\cdot\partial^2\varphi/\partial\tau^2+\partial^2\varphi/\partial\rho^2+\sigma(\rho)\cdot\varphi=\tilde{f}(z,\tau,\rho),$$

where

$$\tilde{k}_z(\rho)=\exp(2\rho)\cdot[k_z(r_0\cdot\exp(\rho))/k(r_0\cdot\exp(\rho))]$$
$$\tilde{k}_\tau(\rho)=\exp(2\rho)\cdot[k_\tau(r_0\cdot\exp(\rho))/k(r_0\cdot\exp(\rho))]$$
$$\sigma(\rho)=\exp(2\rho)\cdot[\beta(r_0\cdot\exp(\rho))/k(r_0\cdot\exp(\rho))]$$
$$\tilde{f}(z,\tau,\rho)=[\exp(2\rho)/k(r_0\exp(\rho))\cdot f(z,\tau,r_0\cdot\exp(\rho)).$$

In addition, the cylindrical domain with the annular cross section is transformed to a 3D rectangular parallelepiped such that

$$z_0\leqq z\leqq z_0+c,\qquad 0\leqq\tau\leqq 2\pi,\qquad \ln(r_1/r_0)\leqq\rho\leqq\ln(r_2/r_0),$$

meaning in effect that the equivalence of problems (1) and (19) is complete.

Obviously the solution φ related to boundary value problems in cylindrical domains is periodic with respect to the angle τ and has a period equal to 2π.

Appendix

Expressions concerning the eigenvalues and right eigenvectors for matrices of the A, B, or C type, are presented in Table V.

In these expressions, ν stands for the order of the matrix, λ_j denotes the jth eigenvalue, whereas $(X)_{ij}$ and $(X^{-1})_{ij}$ denote the ij elements of the eigenvector matrix

TABLE V

Matrix	$(\lambda_j - p)/2q$	$(X)_{ij}$	$(X^{-1})_{ij}$
$A_v[p, q]$	$\cos[j\pi/(v+1)]$	$\sin[ij\pi/(v+1)]$	$\sin[ij\pi/(v+1)]$
$B_v[p, q]$	$\cos[2(j-1)\,\pi/v]$	$\sin[2i(j-1)\,\pi/v + \pi/4]$	$\sin[2(i-1)\,j\pi/v + \pi/4]$
$C_v^{\sqcup}[p, q]$	$\cos[(j-1/2)\,\pi/v]$	$\cos[(i-1)(2j-1)\,\pi/2v]$	$1/2\ (j=1)$ $\cos[(2i-1)(j-1)\,\pi/2v]$
$C_v^{\sqcup}[p, q]'$	$\cos[(j-1/2)\,\pi/v]$	$1/2\ (i=1)$ $\cos[(i-1)(2j-1)\,\pi/2v]$	$\cos[(2i-1)(j-1)\,\pi/2v]$
$C_v^{L}[p, q]$	$\cos[(j-1/2)\,\pi/v]$	$\sin[i(2j-1)\,\pi/2v]$	$\sin[(2i-1)\,j\pi/2v]$ $(-1)^{i-1}\cdot 1/2\ (j=v)$
$C_v^{L}[p, q]'$	$\cos[(j-1/2)\,\pi/v]$	$\sin[i(2j-1)\,\pi/2v]$ $(-1)^{j-1}\cdot 1/2\ (i=v)$	$\sin[(2i-1)\,j\pi/2v]$
$C_v^{L\sqcup}[p, q]$	$\cos[(j-1)\,\pi/(v-1)]$	$1/\sqrt{2}\ (j=1)$ $\cos[(i-1)(j-1)\,\pi/(v-1)]$ $(-1)^{i-1}/\sqrt{2}\ (j=v)$	

and its inverse, respectively. Finally, the symbol $'$ stands for the transpose of a matrix while, in all cases, the normalizing constant which multiplies the elements of X or X^{-1} equals $(2/v)^{1/2}$ except for the matrices $A_v[p, q]$ and $C_v^{L\sqcup}[p, q]$, where the constant in question equals $[2/(v+1)]^{1/2}$ and $[2/(v-1)]^{1/2}$, respectively.

The elements of the inverse eigenvector matrix for $C_v^{L\sqcup}[p, q]$ and also the elements of both the eigenvector matrix and its inverse for $C_v^{L\sqcup}[p, q]'$, have not been included in Table V.

The inverse eigenvector matrix for $C_v^{L\sqcup}[p, q]$ is equal to the transpose of its eigenvector matrix after its first and last columns have been divided by 2.

The eigenvector matrix for $C_v^{L\sqcup}[p, q]'$ equals the eigenvector matrix for $C_v^{L\sqcup}[p, q]$, provided that its first and last rows have been divided by 2, while its inverse is identical to the transpose eigenvector matrix of $C_v^{L\sqcup}[p, q]$.

REFERENCES

1. W. G. BICKLEY AND J. MCNAMEE, *Philos. Trans. Roy. Soc.* London **252**, 69 (1960).
2. G. BIRKHOFF AND R. E. LYNCH, *Numerical Solution of Elliptic Problems* (SIAM, Philadelphia, 1984), pp. 81, 91.
3. J. W. COOLEY, P. A. W. LEWIS, AND P. D. WELCH, *J. Sound Vib.* **12**, 315 (1970).
4. D. J. EVANS AND M. HATZOPOULOS, *Comput. J.* **19**, 184 (1976).
5. R. W. HOCKNEY, "The Potential Calculation and Some Applications," in *Methods of Computational Physics*, edited by B. Adler, S. Fernback, and M. Rotenberg (Academic Press, New York, 1969), Vol. 9, p. 139.
6. G. KNORR, G. JOYCE, AND A. J. MARCUS, *J. Comput. Phys.* **38**, 227 (1980).
7. R. E. LYNCH, J. R. RICE, AND D. H. THOMAS, *Numer. Math.* **6**, 185 (1964).

8. R. E. LYNCH AND J. R. RICE, "The HODIE Method and Its Performance for Solving Elliptic Partial Differential Equations," in *Recent Advances in Numerical Analysis*, edited by C. de Boor and H. Golub (Academic Press, New York, 1978), p. 144.
9. P. MERCIER AND M. DEVILLE, *J. Comput. Phys.* **39**, 443 (1981).
10. W. PROSKUROWSKI AND O. WIDLUND, *Math. Comput.* **30**, 433 (1976).
11. S. SHAANAN AND J. FERZIGER, Report TF-3, Department of Mechanical Engineering, Stanford University, 1975 (unpublished).
12. C. TEMPERTON, *J. Comput. Phys.* **31**, 1 (1979).
13. R. B. WILHELMSON, *J. Atmos. Sci.* **31**, 1629 (1974).
14. R. B. WILHELMSON AND J. H. ERICKSON, *J. Comput. Phys.* **25**, 319 (1977).
15. G. DAHLQUIST AND A. BJORCK, *Numerical Methods* (Prentice–Hall, Englewood Cliffs, NJ, 1974), p. 167.
16. A. V. OPPENHEIM AND R. W. SCHAFER, *Digital Signal Processing* (Prentice–Hall, Englewood Cliffs, NJ, 1975), p. 296.