AN EXTRAPOLATION PROCEDURE FOR SOLVING LINEAR SYSTEMS*

By
THOMAS A. OLIPHANT
Los Alamos Scientific Laboratory
University of California
Los Alamos, New Mexico

1. Introduction. There are many physical problems in which one wishes to solve linear partial differential equations. For example, in steady-state diffusion theory we may encounter the equation,

$$\nabla \cdot \{f(\mathbf{r})\nabla\theta\} + \lambda(\mathbf{r})\theta + S(\mathbf{r}) = 0. \tag{1}$$

A particularly useful approach to solving equations of this type is through the use of finite difference schemes. The simplest finite difference schemes for solving this problem in two space dimensions involve approximating the second partial derivative term by a five-point difference. We will confine our attention to solving (1) in two dimensions using five-point difference schemes although the method is applicable to more dimensions and higher order partial derivative terms.

Our procedure then involves replacing (1) by a set of five-point difference equations defined over a space mesh. Our general mesh is set up as shown in Figure 1. We define our problem in terms of an $N \times N$ mesh. Our actual physical problem may have boundaries such as those indicated by the dotted line. In such a case we simply consider only the points which fall inside the dotted line as internal points. However, for notational convenience we still maintain the $N \times N$ mesh notation. For the sake of simplicity we will begin in Section 2 by considering a problem with $N \times N$ internal points. Then in Section 3 we will indicate how the boundary conditions are treated for more general boundary shapes.

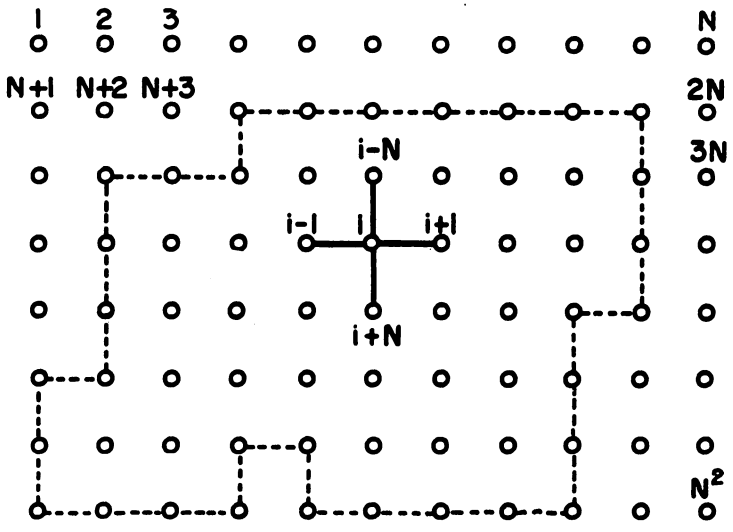

FIG. 1. A typical mesh.

*Received November 6, 1961. Work performed under the auspices of the U. S. Atomic Energy Commission.

The five-point difference equation at a point i in the mesh is

$$B_i^{i-N}\theta_{i-N} + B_i^{i-1}\theta_{i-1} + B_i^{i+1}\theta_{i+1} + B_i^{i+N}\theta_{i+N} + B_i^i\theta_i + s_i = 0. \tag{2}$$

Thus, our problem reduces to a set of simultaneous, linear equations of a particular type. In matrix notation (2) is written,

$$B\boldsymbol{\theta} + \mathbf{s} = 0. \tag{3}$$

We will adopt the convention that capital letters refer to matrices and lower case, bold faced letters refer to vectors. The matrix B has the following appearance.

$$E = \begin{bmatrix}
B_1^1 & B_1^2 & 0 & \cdot & \cdot & \cdot & 0 & B_1^{N+1} & 0 & \cdot & \cdot & \cdot \\
B_2^1 & B_2^2 & B_2^3 & 0 & \cdot & \cdot & \cdot & 0 & B_2^{N+2} & 0 & \cdot & \cdot \\
0 & B_3^2 & B_3^3 & B_3^4 & 0 & \cdot & \cdot & \cdot & 0 & B_3^{N+3} & 0 & \cdot \\
\cdot & 0 & B_4^3 & & & & & & & & & \\
\cdot & \cdot & 0 & & & & & & & & & \\
\cdot & \cdot & \cdot & & & & & & & & & \\
0 & \cdot & \cdot & & & & & & & & & \\
B_{N+1}^1 & 0 & \cdot & & & & & & & & & \\
0 & B_{N+2}^2 & 0 & & & & & & & & & \\
\cdot & 0 & B_{N+3}^3 & & & & & & & & & \\
\cdot & \cdot & 0 & & & & & & & & & \\
\cdot & \cdot & \cdot & & & & & & & & &
\end{bmatrix} \tag{4}$$

The arrows indicate the non-vanishing diagonals. All the remaining elements are equal to zero.

Most of the methods which have proved useful for solving such a linear system involve two basic techniques—matrix factorization and relaxation methods. In the corresponding one-dimensional problem the matrix B is tri-diagonal and can be factored exactly in a very trivial way [1] thus allowing a very practical, non-iterative method for solving the problem. This factorization is not so easy in the case of the matrix (4) and there seems to be no very practical way of obtaining a non-iterative solution of (3). However, use has been made of the basic idea of matrix factorization in setting up iterative procedures for solving (3). For example, one can replace the time by an iterative parameter in the alternating direction method [2] and also in a method proposed by Baker and Oliphant [3]. Another method has been recently proposed by the present author [4]. This latter method will be referred to below as Method I. Although Method I was originally applied to time-dependent problems involving nine-point space differences, it can equally well be applied directly to steady state problems involving five-point space differences as will be shown below. When we refer to Method I below we will understand it to be applied in the latter sense. Two of the best known types of relaxation methods are the methods of extrapolated simultaneous iteration [5] and the methods of extrapolated successive iteration [5], [6]. The latter methods are sometimes referred to as successive over-relaxation methods and we will refer to them collectively (including the case of under-relaxation) as the S. R. method.

The basis of the present work is the recognition of the similarity between Method I and the S. R. method. In the latter method the whole upper triangular part of the matrix B is transposed to the right-hand side of the equation to operate on the guessed solution, whereas in Method I only part of it is. Because of the similarity of the above two methods, it was found that the extrapolative techniques used in the S. R. method can also be applied to Method I. Another aspect of the present work is the use of the more standard matrix notation in contrast to the notation used previously [4] in which the vectors had a double subscript.*

2. Derivation of the Method. In order to illustrate fully the connection between Method I and the S. R. method we formulate the problem generally with a parameter k. For $k = 0$ the method reduces to pure S. R., for $k = 1$ the method reduces purely to Method I and there are various combinations of the two methods for other values of k.

Now, let us consider the solution of (2). The matrix B can be broken up as follows.

$$B = L + U + D \tag{5}$$

where L contains just the lower two diagonal parts of B, U contains just the upper two diagonal parts of B and D contains just the main diagonal of B. Then (3) can be written,

$$(L + U + D)\boldsymbol{\theta} = -\mathbf{s}. \tag{6}$$

Instead of transposing to the right hand side the whole matrix U, we retain a fraction k of U on the left hand side. Thus,

$$(L + kU + D)\boldsymbol{\theta} = -\mathbf{s} - (1 - k)U\boldsymbol{\theta}. \tag{7}$$

We rewrite this as

$$C\boldsymbol{\theta} = \mathbf{d} \tag{8}$$

where

$$C = L + kU + D \tag{9}$$

and

$$\mathbf{d} = -\mathbf{s} - (1 - k)U\boldsymbol{\theta}. \tag{10}$$

Now, let us define a lower triangular matrix W of the form

$$W = \begin{bmatrix} W_1^1 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ W_2^1 & W_2^2 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & W_3^2 & W_3^3 & 0 & \cdot & \cdot & \cdot & \cdot \\ \cdot & 0 & W_4^3 & & & & & \\ \cdot & \cdot & 0 & & & & & \\ \cdot & \cdot & \cdot & & & & & \\ 0 & \cdot & \cdot & & & & & \\ W_{N+1}^1 & 0 & \cdot & & & & & \\ 0 & W_{N+2}^2 & 0 & & & & & \\ & 0 & W_{N+3}^3 & & & & & \\ \cdot & \cdot & 0 & & & & & \\ \cdot & \cdot & \cdot & & & & & \end{bmatrix} \tag{11}$$

*The author is greatly indebted to P. M. Stone of Los Alamos for pointing out the obscurities of the previous formulation and assisting in the reformulation of the present work.

and an upper triangular matrix V of the form

$$V = \begin{bmatrix} 1 & V_1^2 & 0 & \cdot & \cdot & \cdot & 0 & V_1^{N+1} & 0 & \cdot & \cdot & \cdot \\ 0 & 1 & V_2^3 & 0 & \cdot & \cdot & \cdot & 0 & V_2^{N+2} & 0 & \cdot & \cdot \\ \cdot & 0 & 1 & V_3^4 & 0 & \cdot & \cdot & \cdot & 0 & V_3^{N+3} & 0 & \cdot \\ \cdot & \cdot & 0 & \searrow & \searrow & & & & \searrow & & & \\ \cdot & \cdot & & & & & & & & & & \\ \cdot & \cdot & & & & & & & & & & \\ \cdot & \cdot & & & & & & & & & & \\ \cdot & \cdot & & & & & & & & & & \\ \cdot & \cdot & & & & & & & & & & \end{bmatrix} \tag{12}$$

The arrows indicate the non-vanishing diagonals. All the remaining elements are equal to zero. The matrix elements of W and V can be determined so that the matrix relation

$$WV = C + H \tag{13}$$

holds where H has non-vanishing elements only where C has vanishing elements and vice versa. On forming the matrix product WV, we see that the matrix H must have the form

$$H = \begin{bmatrix} 0 & 0 & & & \cdot & \cdot & 0 & 0 & 0 & \cdot & \cdot & \cdot \\ 0 & 0 & & & & & & 0 & H_2^{N+1} & 0 & \cdot & \cdot \\ \cdot & & & & & & & & 0 & H_3^{N+2} & 0 & \cdot \\ \cdot & & & & & & & & & \searrow & & \\ \cdot & & & & & & & & & & & \\ 0 & & & & & & & & & & & \\ 0 & 0 & & & & & & & & & & \\ 0 & H_{N+1}^2 & 0 & & & & & & & & & \\ \cdot & 0 & H_{N+2}^3 & & & & & & & & & \\ \cdot & \cdot & 0 & & & & & & & & & \\ \cdot & \cdot & \cdot & \searrow & & & & & & & & \end{bmatrix} \tag{14}$$

The representation of the non-vanishing diagonals by arrows is the same as before. Comparing the elements of WV with those of C, we see that the elements of W and V are completely determined by the following chain of algebraic equations

$$W_i^{i-N} = B_i^{i-N} \tag{15}$$

$$W_i^{i-1} = B_i^{i-1} \tag{16}$$

$$W_i^i = B_i^i - W_i^{i-N} V_{i-N}^i - W_i^{i-1} V_{i-1}^i \tag{17}$$

$$V_i^{i+N} = \frac{kB_i^{i+N}}{W_i^i} \tag{18}$$

$$V_i^{i+1} = \frac{kB_i^{i+1}}{W_1^i} \tag{19}$$

The elements of WV corresponding to the non-vanishing elements of H are now already determined. We define H by setting the elements of H equal to these remaining elements of WV which were not used in setting up the chain (15)-(19). The non-vanishing elements of H are therefore given by

$$H_i^{i+N-1} = W_i^{i-1} V_{i-1}^{i+N-1} \tag{20}$$

for the upper diagonal and

$$H_i^{i-N+1} = W_i^{i-N} V_{i-N}^{i-N+1} \tag{21}$$

for the lower diagonal.

Operating on $\boldsymbol{\theta}$ with (13), we have

$$WV\boldsymbol{\theta} = C\boldsymbol{\theta} + H\boldsymbol{\theta}. \tag{22}$$

Then, using (8), we replace $C\boldsymbol{\theta}$ in (22) by $\mathbf{d}$ obtaining

$$WV\boldsymbol{\theta} = \mathbf{d} + H\boldsymbol{\theta}. \tag{23}$$

Replacing $\mathbf{d}$ by its definition (10), we write

$$WV\boldsymbol{\theta} = -\mathbf{s} + [H - (1 - k)U]\boldsymbol{\theta} \tag{24}$$

where now the appearance of $\boldsymbol{\theta}$ is explicit. We now set up our iterative procedure in the following way.

$$WV\boldsymbol{\theta}^{(n+1)} = -\mathbf{s} + [H - (1 - k)U]\boldsymbol{\theta}^{(n)} \tag{25}$$

where the superscript (n) refers to the n-th iterate. We rewrite this as

$$WV\boldsymbol{\theta}^{(n+1)} = \mathbf{h} \tag{26}$$

where

$$\mathbf{h} = -\mathbf{s} + [H - (1 - k)U]\boldsymbol{\theta}^{(n)} \tag{27}$$

The elements of $\mathbf{h}$ are then determined by the algebraic relation

$$h_i = -s_i + [W_i^{i-N} V_{i-N}^{i-N+1} \theta_{i-N+1}^{(n)} + W_i^{i-1} V_{i-1}^{i+N-1} \theta_{i+N-1}^{(n)}] - (1 - k)[B_i^{i+N} \theta_{i+N}^{(n)} + B_i^{i+1} \theta_{i+1}^{(n)}]. \tag{28}$$

Since W and V are easy to invert, we can calculate $\boldsymbol{\theta}^{(n+1)}$ by using

$$\boldsymbol{\theta}^{(n+1)} = V^{-1} W^{-1} \mathbf{h}. \tag{29}$$

This can be written as

$$\boldsymbol{\theta}^{(n+1)} = V^{-1}\mathbf{g} \tag{30}$$

where

$$\mathbf{g} = W^{-1}\mathbf{h}. \tag{31}$$

Writing (31) and (30) out algebraically, we obtain for the elements of $\mathbf{g}$ and $\boldsymbol{\theta}^{(n+1)}$, respectively,

$$g_i = \frac{h_i - W_i^{i-N} g_{i-N} - W_i^{i-1} g_{i-1}}{W_i^i} \tag{32}$$

and

$$\theta_i^{(n+1)} = g_i - V_i^{i+N}\theta_{i+N}^{(n+1)} - V_i^{i+1}\theta_{i+1}^{(n+1)}. \tag{33}$$

We now summarize our basic computational formulas in algebraic form for the unextrapolated case.

$$W_i^{i-N} = B_i^{i-N} \tag{34}$$

$$W_i^{i-1} = B_i^{i-1} \tag{35}$$

$$W_i^i = B_i^i - W_i^{i-N}V_{i-N}^i - W_i^{i-1}V_{i-1}^i \tag{36}$$

$$V_i^{i+N} = \frac{kB_i^{i+N}}{W_i^i} \tag{37}$$

$$V_i^{i+1} = \frac{kB_i^{i+1}}{W_i^i} \tag{38}$$

$$h_i = -s_i + [W_i^{i-N}V_{i-N}^{i-N+1}\theta_{i-N+1}^{(n)} + W_i^{i-1}V_{i-1}^{i+N-1}\theta_{i+N-1}^{(n)}] - (1-k)[B_i^{i+N}\theta_{i+N}^{(n)} + B_i^{i+1}\theta_{i+1}^{(n)}] \tag{39}$$

$$g_i = \frac{h_i - W_i^{i-N}g_{i-N} - W_i^{i-1}g_{i-1}}{W_i^i} \tag{40}$$

$$\theta_i^{(n+1)} = g_i - V_i^{i+N}\theta_{i+N}^{(n+1)} - V_i^{i+1}\theta_{i+1}^{(n+1)}. \tag{41}$$

Our computation proceeds in the following way. First, we sweep through the mesh from low to high values of i, computing the W and V elements using (34)-(38) and storing them for later use in the iterations. Our iteration procedure then goes as follows. We sweep from low to high values of i computing and storing g_i using (39), (40) and the n-th iterate $\theta_i^{(n)}$. Then, using (41), we obtain $\theta_i^{(n+1)}$ by sweeping back from high to low values of i. We continue to iterate until successive iterates agree with each other to the desired accuracy.

The extrapolated method is obtained by simply replacing (41) by

$$\theta_i^{(n+1)} = \omega[g_i - V_i^{i+N}\theta_{i+N}^{(n+1)} - V_i^{i+1}\theta_{i+1}^{(n+1)}] + (1-\omega)\theta_i^{(n)} \tag{42}$$

where ω is the extrapolation parameter.

3. The boundary conditions. We now set up our computational formulas to take care of such regions as the one enclosed by the dotted line in Figure 1. First, we define the quantities Γ_i and Δ_i .

$$\Gamma_i = \begin{cases} 1 & \text{if } i \text{ is an internal point.} \\ 0 & \text{if } i \text{ is not an internal point.} \end{cases} \tag{43}$$

$$\Delta_i = 1 - \Gamma_i \tag{44}$$

Our computational formulas are written,

$$W_i^{i-N} = B_i^{i-N}\Gamma_{i-N} \tag{45}$$

$$W_i^{i-1} = B_i^{i-1}\Gamma_{i-1} \tag{46}$$

$$W_i^i = B_i^i - W_i^{i-N}V_{i-N}^i - W_i^{i-1}V_{i-1}^i \tag{47}$$

$$V_i^{i+N} = \frac{kB_i^{i+N}}{W_i^i}\,\Gamma_{i+N} \tag{48}$$

$$V_i^{i+1} = \frac{kB_i^{i+1}}{W_i^i}\,\Gamma_{i+1} \tag{49}$$

$$\begin{aligned} h_i = -s_i + [W_i^{i-N}V_{i-N}^{i-N+1}\theta_{i-N+1}^{(n)}\Gamma_{i-N+1} + W_i^{i-1}V_{i-1}^{i+N-1}\theta_{i+N-1}^{(n)}\Gamma_{i+N-1}] \\ - (1-k)[B_i^{i+N}\theta_{i+N}^{(n)}\Gamma_{i+N} + B_i^{i+1}\theta_{i+1}^{(n)}\Gamma_{i+1}] \\ - [B_i^{i-N}\theta_{i-N}^{(b)}\,\Delta_{i-N} + B_i^{i-1}\theta_{i-1}^{(b)}\,\Delta_{i-1} \\ + B_i^{i+1}\theta_{i+1}^{(b)}\,\Delta_{i+1} + B_i^{i+N}\theta_{i+N}^{(b)}\,\Delta_{i+N}]. \end{aligned} \tag{50}$$

The $\theta_i^{(b)}$ are boundary values.

$$g_i = \frac{h_i - W_i^{i-N}g_{i-N}\Gamma_{i-N} - W_i^{i-1}g_{i-1}\Gamma_{i-1}}{W_i^i} \tag{51}$$

$$\theta_i^{(n+1)} = \omega[g_i - V_i^{i+N}\theta_{i+N}^{(n+1)}\Gamma_{i+N} - V_i^{i+1}\theta_{i+1}^{(n+1)}\Gamma_{i+1}] + (1-\omega)\theta_i^{(n)} \tag{52}$$

The computational procedure is the same as that described in Section 2 except that now it is understood that we actually calculate only when i is an interior point of the region.

4. The convergence condition. Let us define the errors $\boldsymbol{\varepsilon}^{(n+1)}$ and $\boldsymbol{\varepsilon}^{(n)}$ by

$$\boldsymbol{\theta}^{(n+1)} = \boldsymbol{\theta}_{\text{true}} + \boldsymbol{\varepsilon}^{(n+1)} \tag{53}$$

$$\boldsymbol{\theta}^{(n)} = \boldsymbol{\theta}_{\text{true}} + \boldsymbol{\varepsilon}^{(n)}. \tag{54}$$

Substituting (53) and (54) into (25) we obtain

$$\boldsymbol{\varepsilon}^{(n+1)} = V^{-1}W^{-1}[H - (1-k)U]\boldsymbol{\varepsilon}^{(n)}. \tag{55}$$

This is of the form

$$\boldsymbol{\varepsilon}^{(n+1)} = K\boldsymbol{\varepsilon}^{(n)} \tag{56}$$

where

$$K = V^{-1}W^{-1}[H - (1-k)U]. \tag{57}$$

Therefore, we see that the convergence condition can be written in terms of the norm $N(K)$ of K. It is

$$N(K) < 1. \tag{58}$$

Thus, if we can compute $N(K)$, we can tell whether a given problem will converge. But the norm $N(K)$ can be written

$$N(K) = \underset{|\mathbf{x}|\neq 0}{\text{l.u.b.}}\,\frac{|K\mathbf{x}|}{|\mathbf{x}|}. \tag{59}$$

Since the matrix K is easily obtained, we can compute the norm $N(K)$. To get a complete computation of the norm, we must take a complete set of independent vectors $\mathbf{x}$ and select the least upper bound as shown in (59).

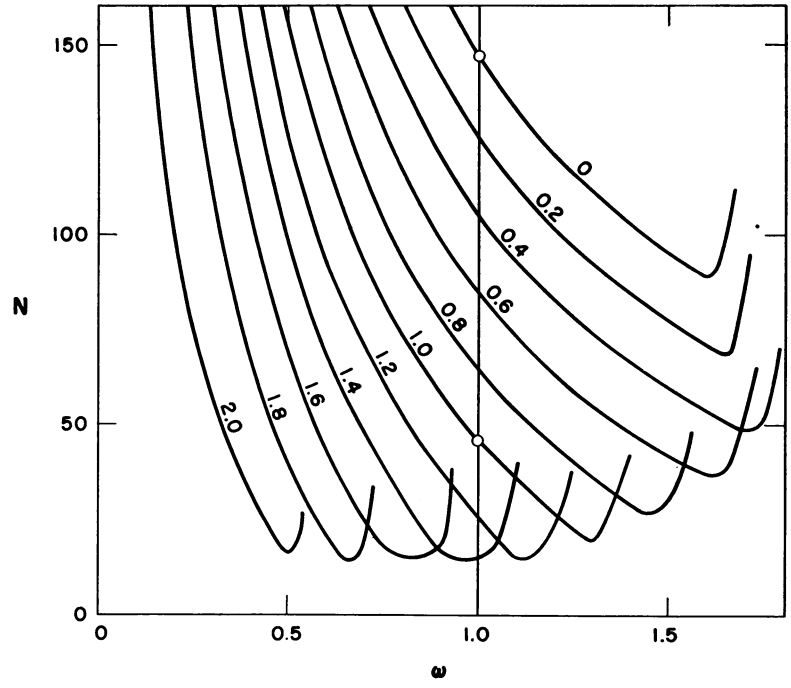

FIG. 2. The number of iterations N is plotted against the extrapolation parameter ω for various values of the parameter k. The value of k for each curve is indicated in the graph.

5. A parameter study on Laplace's equation. In the particular case of Laplace's equation we have

$$B_i^{i-N} = B_i^{i-1} = B_i^{i+1} = B_i^{i+N} = 1 \tag{60}$$

and

$$B_i^i = -4. \tag{61}$$

The two free parameters are k and ω. The particular boundary value problem which we considered consisted of a rectangular 10×10 mesh. The function θ was taken to be unity on all of the boundary. The first guess $\theta^{(1)}$ was taken to be 10^{-3} at each internal point. The solution was iterated until it agreed with the correct solution to within an absolute error of less than one part in 10^5. The number N of iterations required for this accuracy was recorded for each pair of parameters (k, ω). A family of curves was thereby obtained and plotted in Figure 2. For each value of k selected, N was plotted as a function of ω. Each curve of the family corresponds to a given k.

For $k = 0$ we have pure S. R. For $\omega > 1$ we have successive over relaxation, for $\omega < 1$ we have successive under relaxation, and for $\omega = 1$ we have pure successive relaxation. The point on the $k = 0$ curve of Figure 2 corresponding to the last case is encircled. For $k = 1$ we have extrapolated Method I. The point on the $k = 1$ curve corresponding to $\omega = 1$ corresponds to pure Method I and is also encircled. Curves are plotted for various other values of k. As is apparent from Figure 2, we obtain better convergence for higher values of k than in the conventional S. R. method which corre-

sponds to $k = 0$. Judging from the shapes of the curves, it seems that the best value for k lies anywhere from 1.2 to 1.4. With values of k in this region, the best value of ω occurs at the minimum of a given curve for the particular value of k used.

References

1. G. H. Bruce, D. W. Peaceman, H. H. Rachford, Jr., and J. D. Rice, *Calculation of unsteady-state gas flow through porous media*, Trans. Am. Inst. Mining Met. Engrs. **198**, 79–92 (1953)
2. J. Douglas, Jr. and D. W. Peaceman, *Numerical solutions of two-dimensional heat-flow problems*, A. I. Ch. E. Journal 1, 505–512 (1955)
3. G. A. Baker, Jr. and T. A. Oliphant, *An implicit numerical method for solving the two-dimensional heat equation*, Quart. Appl. Math. **17**, 361–373 (1960)
4. T. A. Oliphant, *An implicit numerical method for solving two-dimensional, time-dependent diffusion problems*, Quart. Appl. Math. **19**, 221–229 (1961)
5. S. P. Frankel, *Convergence rates of iterative treatment of partial differential equations*, Math. Tables Aids Computation **4**, 65–75 (1950)
6. D. Young, *Iterative methods for solving partial difference equations of elliptic type*, Trans. Am. Math. Soc. **76**, 92–111 (1954)