On the use of fast methods for solving boundary value problems

D. J. Evans

Department of Computer Studies, University of Technology, Loughborough, Leicestershire LE11 3TU

Recently several fast computational methods have been developed for solving finite difference approximations to the standard partial differential equations of mathematical physics.
In this paper, the application of a new fast algorithm for solving tridiagonal systems with constant coefficients occurring in the solution of boundary value problems is presented.
It is shown that the development of such methods of solution for commonly occurring problems in mathematical and computational physics can yield fruitful gains in efficiency.

(Received February 1973)

1. Introduction

The use of digital computers to obtain approximate solutions of boundary value problems involving ordinary and partial differential equations is nowadays extensive. In using such machines there is always considerable motivation to find the most efficient methods of solution. Thus, for instance, in the solution of the linear diffusion equation in one space dimension, the well-known explicit method of solution (Richtmyer, 1957), although possessing great simplicity and ease of solution, is unacceptable because the stability condition $k\Delta t/\Delta x^2 \leqslant 0{\cdot}5$ imposes such a severe restriction on the forward integration time step that long computation times are inevitable.

Implicit methods are generally preferred because of their superior stability characteristics but suffer greatly from the disadvantage that they require the solution of tridiagonal systems of difference equations at each time step, which indirectly involves more work. However, these systems, usually solved by Gaussian elimination or one of its variants, are quite often given far greater generality than they actually possess or deserve, for in many cases the finite difference equations possess constant coefficients and are often symmetric in form. Thus, the need for a special purpose-built solution process in these instances can be clearly justified.

In the following sections, an exact factorisation of the coefficient matrix incorporating the implicit finite difference equations derived from a linear diffusion problem in both one and two space dimensions is obtained, resulting in an algorithmic method which is stable, semi-explicit and one for which the amount of computational effort is comparable to that of the explicit method. By extending these ideas to elliptic problems, a previous paper (Evans, 1972) demonstrated the applicability of the method to block tridiagonal systems where significant time reductions can be obtained for the direct solution of Poisson's equation over a two dimensional network (for a restricted class of problems). In this paper, the ideas are extended to elliptic problems involving the use of alternating direction methods and can be applied to a class of 2 point boundary value problems with which a more recent algorithm of Rose (1969) is unable to cope successfully.

2. An implicit method for solving the diffusion equation

We wish to obtain the solution of the diffusion equation

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(k\frac{\partial u}{\partial x}\right), \tag{2.1}$$

in the domain $R:[0 \leqslant x \leqslant 1; t \geqslant 0]$ subject to the initial condition,

$$u(x, 0) = f(x) \text{ for } 0 \leqslant x \leqslant 1 , \tag{2.1a}$$

and the Dirichlet boundary conditions,

$$u = u_0(t), x = 0$$

$$\text{and } u = u_1(t), x = 1 \text{ for } t \geqslant 0 , \tag{2.1b}$$

where $u_0(t)$ and $u_1(t)$ remain bounded as $t \to \infty$ and there are no discontinuities in the initial and boundary conditions.

The region R is covered by a rectangular mesh, the nodal points of which are given by,

$$x_i = ih;\ i = 0, 1, \ldots N;\ Nh = 1 ,$$

and

$$t_j = jl, j \geqslant 0 .$$

The constants h and l are the mesh and time increments respectively, and the ratio kl/h^2 is usually denoted by r.

The simplest implicit finite difference method for solving (2.1) is to replace it by the finite difference equations,

$$(u_{i,j+1} - u_{i,j})/l = k(u_{i-1,j+1} - 2u_{i,j+1} + u_{i+1,j+1})/h^2 + 0(h^2 + l) \quad (i = 1, 2, \ldots, N-1) , \tag{2.2}$$

where $u_{i,j}$ is the point $u(ih, jl)$ on the chosen rectangular grid. We now make the substitution (Evans, 1971),

$$r = kl/h^2 = \alpha/(1-\alpha)^2 \tag{2.3}$$

which gives the value of α as

$$\alpha = 0{\cdot}5\,[(2 + {}^1/r) - \{(2 + {}^1/r)^2 - 4\}^{\frac{1}{2}}] . \tag{2.4}$$

If we now substitute the parameter α into the equation (2.2), we see that the implicit finite difference equation at the point (i, j) has the form,

$$-\alpha u_{i-1,j+1} + (1 + \alpha^2)u_{i,j+1} - \alpha u_{i+1,j+1} = (1-\alpha)^2 u_{i,j} \tag{2.5}$$

which relates the three unknown u values at the $(j+1)l$th time to the known u value on the jlth line.

Applying this equation to the points $i = 1, 2, \ldots, N-1$ on each grid line, we obtain the system of $(N-1)$ linear equations in the $(N-1)$ unknowns

$u_{1,j+1}, u_{2,j+1}, \ldots, u_{N-1,j+1}$ in the form

$$A\mathbf{u}_{j+1} = B\mathbf{u}_j + \mathbf{f}_j , \tag{2.6}$$

where

$$A = \begin{bmatrix} 1+\alpha^2, & -\alpha, & & & & 0 \\ -\alpha, & 1+\alpha^2, & -\alpha & & & \\ & -\alpha & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & -\alpha \\ 0 & & & & -\alpha & 1+\alpha^2 \end{bmatrix}$$

$$B = \begin{bmatrix} (1-\alpha)^2, & & & 0 \\ & (1-\alpha)^2 & & \\ & & \ddots & \\ 0 & & & (1-\alpha)^2 \end{bmatrix}, \text{ and } \mathbf{f}_j = \begin{bmatrix} \alpha u_0(jl) \\ 0 \\ \vdots \\ 0 \\ \alpha u_1(jl) \end{bmatrix}.$$

In a similar way, we can treat the more accurate Crank–Nicholson finite difference method in which the term $\partial^2 u/_{\partial x}{}^2$ in equation (2.1) is replaced by the average of the central difference operators on the lines $(j + 1)l$ and jl.

It can be shown by a similar analysis to that given earlier that with the transformation,

$$r = 2\alpha/(1 - \alpha)^2 \tag{2.7}$$

or

$$\alpha = [(1 + 1/r) - \{(1 + 1/r)^2 - 1\}^{\frac{1}{2}}] \tag{2.8}$$

the finite difference equation for the Crank–Nicholson method becomes

$$-\alpha u_{i-1,j+1} + (1 + \alpha^2)u_{i,j+1} - \alpha u_{i+1,j+1} = \alpha u_{i-1,j} + (1 - 4\alpha + \alpha^2)u_{i,j} + \alpha u_{i+1,j} , \tag{2.9}$$

which when applied to the points $i = 1, 2, \ldots, N - 1$ on each grid line gives a linear system of equations similar to (2.6) but with

$$B = \begin{bmatrix} \delta & \alpha & & & \\ \alpha & \delta & \alpha & & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & & \ddots & \ddots & \alpha \\ & & & \alpha & \delta \end{bmatrix} \text{ and } \mathbf{f}_j \begin{bmatrix} \alpha u_0(jl) + \alpha u_0((j+1)l) \\ 0 \\ \vdots \\ 0 \\ \alpha u_1(jl) + \alpha u_1((j+1)l) \end{bmatrix}$$

where $\delta = 1 - 4\alpha + \alpha^2$.

From the stability considerations of the two implicit methods discussed in this section, we know them to be stable and free from rounding error growth for all values of $r > 0$. This condition can be verified to be satisfied for all α in the range $0 < \alpha < 1$ for both equations (2.6) and (2.9).

The solution at each time step is usually obtained by solving the implicit equations (2.6) and (2.9) by a variant of Gaussian elimination (Varga, 1963). For large time integrations, this can involve a considerable amount of computation. However, in very many cases, the implicit equation (2.6) yields a matrix A which is often symmetric with constant coefficients, and the need for a special purpose algorithm can be justified particularly if the equations contain time varying constants $k(t)$ or if frequent changes in the time step Δt (and hence r) are envisaged which would require recomputation of the tridiagonal matrix algorithm. A further application is when implicit methods are used in a Stefan problem and the range of integration varies at each time step.

3. Applications to boundary value problems

In the numerical solution of quasi-linear parabolic and elliptic partial differential equations with constant or time varying coefficients involving two space dimensions and specified boundary conditions by the alternating direction implicit methods, there occurs a similar computational problem in which the ideas developed in Section 2 can be exploited.

For diffusion problems involving two space dimension, we consider a square region R with sides of length unity and mesh coordinates specified by (ih, jh) for $0 \leqslant i, j \leqslant N$ where $h = \Delta x = \Delta y = 1/N$, the mesh increment. The initial conditions are specified in the plane region R with boundary S. The differential equations and the boundary conditions are defined on the sets R' and S' of all points (x, y, t) such that $t \geqslant 0$ and $(x, y) \in R$ and such that $t \geqslant 0$ and $(x, y) \in S$ respectively. Thus, we consider the equation

$$\frac{\partial u}{\partial t} = k\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right), (x, y, t) \in R' \tag{3.1}$$

with boundary and initial conditions

$$u(x, y, t) = g(x, y, t), (x, y, t) \in S , \tag{3.2}$$

and

$$u(x, y, 0) = f(x, y), (x, y) \in R . \tag{3.3}$$

A significant advance in the solution of such problems was made by Peaceman and Rachford (1955), who devised the method of alternating directions for solving the finite difference equations arising from (3.1). In this method, we use two forms of equation alternately in successive time steps; in the first the finite difference equations are implicit in the x-direction and explicit in the y-direction, and in the second the directions are interchanged.

Again, we use the relation

$$kl/h^2 = \alpha/(1 - \alpha)^2 , \tag{3.4}$$

to simplify the implicit finite difference equations derived from (3.1) in the x— and y— directions to obtain,

$$-\alpha u_{i-1,j,2n+1} + (1 + \alpha^2)u_{i,j,2n+1} - \alpha u_{i+1,j,2n+1} = \alpha u_{i,j-1,2n} + (1 - 4\alpha + \alpha^2)u_{i,j,2n} + \alpha u_{i,j+1,2n} , \tag{3.5}$$

and

$$-\alpha u_{i,j-1,2n+2} + (1 + \alpha^2)u_{i,j,2n+2} - \alpha u_{i,j+1,2n+2} = \alpha u_{i-1,j,2n+1} + (1 - 4\alpha + \alpha^2)u_{i,j,2n+1} + \alpha u_{i+1,j,2n+1} \tag{3.6}$$

for all points $1 \leqslant i, j \leqslant N - 1$ and subject to the boundary conditions (3.2).

The equations (3.5) and (3.6) when grouped together yield the following pair of compound matrix systems,

$$E\mathbf{u}_{2n+1} = F\mathbf{u}_{2n} + \mathbf{g}_1$$

and

$$E\tilde{\mathbf{u}}_{2n+2} = F\tilde{\mathbf{u}}_{2n+1} + \mathbf{g}_2 \tag{3.7}$$

where E and F are compound matrices of order $(N - 1 \times N - 1)$ and defined as

$$E = \begin{bmatrix} A & & & \\ & A & & 0 \\ & & \ddots & \\ 0 & & & A \end{bmatrix}_{N-1} \qquad F = \begin{bmatrix} \delta I & \alpha I & & \\ \alpha I & \ddots & \ddots & 0 \\ & \ddots & \ddots & \ddots \\ 0 & & \ddots & \alpha I \\ & & \alpha I & \delta I \end{bmatrix}_{N-1}$$

with A, δ and $\mathbf{f}$ as previously defined in Section 2 and $\mathbf{u}$, $\tilde{\mathbf{u}}$ are row-wise and column-wise compound vectors yielding the solution along rows and columns of the given domain R and $\mathbf{g}_1$ and $\mathbf{g}_2$ are vectors derived from the appropriate boundary conditions and ordered row-wise and column-wise accordingly.

Thus, we have shown that the alternating direction implicit methods can with a little analysis be expressed in a computational form whereby the tridiagonal systems which need to be solved along each row and column of the network can have a predetermined factorised form, thus eliminating the most tedious and time consuming part of the method.

Consider further then, for example, the second order elliptic partial differential equation

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} + 2\sigma u = f(x, y) \; 0 < x, y < 1 \; \sigma = \sigma(t) \tag{3.8}$$

in the unit square R with the Dirichlet boundary conditions,

$$u(x, y) = b(x, y), (x, y) \in \partial R , \tag{3.9}$$

where b is specified on ∂R, the boundary of R. Writing (3.8) as

$$\left[-\frac{\partial^2 u}{\partial x^2} + \sigma u\right] + \left[-\frac{\partial^2 u}{\partial y^2} + \sigma u\right] = f , \tag{3.10}$$

where each term in brackets represents a differential operator in the space variables x, y.

Then, with

$$Hu_{i,j} \equiv [(2 + \sigma h^2)u_{i,j} - u_{i+1,j} - u_{i-1,j}]/h^2 \quad (3.11)$$
$$Vu_{i,j} \equiv [(2 + \sigma h^2)u_{i,j} - u_{i,j+1} - u_{i,j-1}]/h^2$$

representing discrete approximations to these differential operators at the point (ih, jh) on a uniform mesh of side $h = 1/N$, the discrete matrix problem corresponding to (3.1), (3.2) can be written in the form,

$$(H + V)\mathbf{u} = \mathbf{f} \quad (3.12)$$

where H and V are real $(N-1)^2 \times (N-1)^2$ matrices and $\mathbf{f}$ is the appropriately modified right hand side vector. Of particular importance to this discussion is the form of these matrices. With the correct ordering of the points of the network (row-wise for H and column-wise for V), it can be shown that both systems have at most three elements per matrix row. Writing (3.12) as the pair of equations,

$$(H + \omega I)\mathbf{u} = (\omega I - V)\mathbf{u} + \mathbf{f}\ , \quad (3.13)$$
$$(V + \omega I)\mathbf{u} = (\omega I - H)\mathbf{u} + \mathbf{f}\ ;$$

we now insert iteration superscripts to define the Peaceman–Rachford alternating direction implicit method as

$$(H + \omega_m I)\mathbf{u}^{(m+\frac{1}{2})} = (\omega_m I - V)\mathbf{u}^{(m)} + \mathbf{f}\ , \quad (3.14)$$
$$(V + \omega_m I)\mathbf{u}^{(m+1)} = (\omega_m I - H)\mathbf{u}^{(m+\frac{1}{2})} + \mathbf{f}\ ;$$

where $\{\omega_m\}$, $(m = 1, 2, \ldots, N)$ is a sequence of positive numbers chosen to accelerate the convergence of the iteration method. Thus, we see from (3.14) that in order to carry out a single complete iteration, the direct solution of tridiagonal matrix equations on horizontal mesh lines and then vertical mesh lines is again necessary.

By making the substitution,

$$\alpha_m(\sigma + \omega_m)h^2 = (1 - \alpha_m)^2 \quad (3.15)$$

where $\{\alpha_m\}$ $(m = 1, 2, \ldots, N)$ is a parameter sequence analogous to the ω_m sequence but with a dual purpose role. It is designed not only to accelerate convergence, in the same way as ω_m but also to achieve finite difference equations which when grouped along the rows or columns of the network in the manner indicated above yields a coefficient matrix which is directly factorable, i.e. the matrices $(H + \omega_m I)$ and $(V + \omega_m I)$ each to possess a factorable form $P_m Q_m$ which will be discussed in Section 4.

The result of substituting (3.15) into (3.14) produces finite difference equations of the form,

$$-\alpha_m u_{i-1,j} + (1 + \alpha_m^2)u_{i,j} - \alpha_m u_{i+1,j} = g_1;\ (i = 1, 2, \ldots, N-1) \text{ for } j = 1, 2, \ldots, N-1 \quad (3.16)$$

and

$$-\alpha_m u_{i,j-1} + (1 + \alpha_m^2)u_{i,j} - \alpha_m u_{i,j+1} = g_2;\ (j = 1, 2, \ldots, N-1) \text{ for } i = 1, 2, \ldots, N-1$$

where the vectors g_1, g_2 are the suitably modified right hand sides of equation (3.14). These equations when grouped together give a sequence of directly factorable matrix equations. The parameter sequence α_m, $\{m = 1, 2, \ldots, N\}$ is then given by

$$\alpha_m = 2/\{2 + Z_m + (\{2 + Z_m\}^2 - 4)^{\frac{1}{2}}\}\ , \quad (3.17)$$

where ω_m, $(m = 1, 2, \ldots, N)$ are the Wachpress and Habetler parameters (Wachpress and Habetler, 1960) and $Z_m = (\sigma + \omega_m)h^2$.

An identical analysis can also be carried out on the Peaceman–Rachford ADI parameters.

Considerable economies can result by using the direct factorisation method outlined above in the solution of quasi-linear partial differential equations by the alternating direction implicit methods and its many variants, i.e. Douglas–Rachford and locally one dimensional methods. In a two dimensional region involving irregular boundaries, the order of the tridiagonal systems can vary from line to line. In addition, the acceleration parameter ω_m and hence α_m also varies from iteration to iteration. Thus, the number of tridiagonal systems which need to be solved can be very large indeed for practical problems especially if time varying constants $k(t)$ or $\sigma(t)$ are involved or if frequent changes in the time step are required.

4. Algorithmic solution of the implicit equations

We now consider the special linear systems occurring in Sections 2 and 3, i.e.

$$\begin{bmatrix} b & c & & & \\ a & b & c & & 0 \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot & \cdot \\ & & & & \cdot & \cdot & \cdot \\ & 0 & & & a & b & c \\ & & & & & a & b \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ d_n \end{bmatrix} \quad (4.1)$$

or in matrix notation,

$$A\mathbf{x} = \mathbf{d}\ .$$

We can rewrite the system (4.1) as

$$\begin{bmatrix} 1 + \beta\gamma, & -\gamma & & & \\ -\beta, & 1 + \beta\gamma, & -\gamma & & 0 \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot & \cdot \\ & & & & \cdot & \cdot & -\gamma \\ 0 & & & & & -\beta, & 1 + \beta\gamma \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ g_n \end{bmatrix} \quad (4.2)$$

where we have made the simple transformations,

$$\delta = \{b + (b^2 - 4ac)^{\frac{1}{2}}\},\ \gamma = -2c/\delta,\ \beta = -2a/\delta\ , \quad (4.3)$$

and

$$g_i = 2d_i/\delta, \text{ for } i = 1(1)n\ . \quad (4.4)$$

In Evans (1972), a new factorisation for the symmetric tridiagonal matrix with constant coefficient elements was introduced. We now extend this analysis further to include the unsymmetric version of this algorithm where the equivalent factorised form for (4.1) is given by

$$A \equiv PQ\ , \quad (4.5)$$

where P and Q are $(n \times n+1)$ and $(n+1 \times n)$ rectangular matrices of the form,

$$P = \begin{bmatrix} 1 & -\gamma & & & \\ & 1 & -\gamma & & \\ & & \cdot & \cdot & \\ & & & \cdot & \cdot \\ & & & & \cdot & \cdot \\ & & & & & 1 & -\gamma \end{bmatrix},\ Q = \begin{bmatrix} 1 & & & \\ -\beta & 1 & & \\ & -\beta & \cdot & \\ & & \cdot & \cdot \\ & & & \cdot & \cdot \\ & & & & -\beta & 1 \\ & & & & & -\beta \end{bmatrix} \quad (4.6)$$

Thus, the system (4.1) is more easily solved by rewriting in the form

$$PQ\,\mathbf{x} = \mathbf{g}$$

which with the introduction of the auxiliary vector $y_i (i = 1, 2, \ldots, n, n+1)$ yields the two alternative simpler systems,

$$Py = g$$

and (4.7)

$$Qx = y\ ,$$

which need to be solved to obtain the solution vector $\mathbf{x}$.

Now the triangular matrices P and Q although rectangular are both easily inverted forms and the linear systems (4.7) can be shown to give the following efficient algorithmic solution:

$$x_n = [g_n + \beta g_{n-1} + \ldots + \beta^{n-1} g_1 - \beta^{n-1}(\beta\gamma) g_1 - \ldots - \beta(\beta\gamma)^{n-1} g_{n-1} - (\beta\gamma)^n g_n]/[1 - (\beta\gamma)^{n+1}] \ , \quad (4.8)$$

and

$$y_{n+1} = -\beta x_n \ .$$

Then, a back substitution process yields the components of the auxiliary solution,

$$y_i = g_i + \gamma y_{i+1} \text{ for } i = n, n-1, \ldots 2, 1 \ , \quad (4.9)$$

whilst the components of the solution vector **x** are given by

$$x_1 = y_1$$

and a forward substitution of the form

$$x_i = y_i + \beta x_{i-1} \text{ for } i = 2, 3, \ldots, n-1 \ . \quad (4.10)$$

Since by definition $\beta, \gamma \leqslant |1|$, it follows immediately that the numerical processes defined by (4.8) to (4.10) are stable against the accumulation of rounding errors (Evans, 1972).

For the special case when $b = 2a = 2c$, and $\beta, \gamma = -1$ then equation (4.8) is indeterminate, and an alternative expression can be obtained. Thus, we have

$$x_n = [ng_n - (n-1)g_{n-1} + (n-2)g_{n-2} \ldots + (-1)^{n-1} g_1]/(n+1) \quad (4.11)$$

together with

$$y_{n+1} = x_n, y_i = g_i - y_{i+1}, \text{ for } i = n, n-1, \ldots, 2, 1 \ , \quad (4.12)$$

and finally

$$x_1 = y_1, x_i = y_i - x_{i-1}, \text{ for } i = 2, 3, \ldots, n-1 \ . \quad (4.13)$$

5. ALGOL program

```
procedure tridiagsolv(n, a, b, c, d);
value n;
integer n;
real a, b, c;
array d;
comment Procedure solves the set of linear equations Ax = d
   where the coefficient matrix is tridiagonal with constant
   sub-diagonal, diagonal and super-diagonal entries a, b, c
   respectively. During computation all the input vectors are
   over-written and the solution vector replaces d, the vector of
   constants. A work space of (n + 9) variables is used and the
   number of operations is of the order of
                  multiplications    4n
                  divisions          2n
                  additions          4n;
begin
integer i;
array y[1:n + 1];
real beta, betan, gamma, delta, sum1, sum2, res, eps;
delta := 0·5 × (b + sqrt(b ↑ 2 − 4 × a × c));
eps := 0·000005
beta := −a/delta; gamma := −c/delta;
for i := 1 step 1 until n do
  d[i] := d[i]/delta;
if abs(beta + 1·0) < eps and abs(gamma + 1·0) < eps
and abs(b ↑ 2 − 4 × a × c) < eps then goto special;
sum1 := 0;
for i := 1 step 1 until n do
  sum1 := d[i] + sum1 × beta;
betan := beta↑n;
sum2 := 0;
for i := n step −1 until 1 do
  sum2 := d[i] + sum2 × gamma;
sum1 := sum1 − sum2 × betan × gamma;
betan := (beta × gamma) ↑ (n + 1);
res := sum1/(1 − betan);
y[n + 1] := −beta × res;
for i := n step −1 until 1 do
  y[i] := d[i] + gamma × y[i + 1];
d[1] := y[1];
for i := 2 step 1 until n − 1 do
  d[i] := y[i] + beta × d[i − 1];
d[n] := res;
goto exit;
special: sum1 := d[1];
for i := 2 step 1 until n do
  sum1 := sum1 − (−1) ↑ i × i × d[i];
sum2 := (−1) ↑ n × sum1/(n + 1);
y[n + 1] := −sum2;
for i := n step −1 until 1 do
  y[i] := d[i] − y[i + 1];
d[1] := y[1];
for i := 2 step 1 until n − 1 do
  d[i] := y[i] − d[i − 1];
d[n] := y(n + 1);
exit: end of tridiagsolv;
```

6. Conclusions

Fast algorithmic solutions to the implicit difference equations of both initial and boundary value problems have been demonstrated to be feasible for equations with time varying constants or if frequent changes in the time step of integration is necessary. Further work is underway to increase the applicability of these ideas to problems involving more general boundary conditions. The manner in which they can be implemented has been briefly discussed previously by Evans (1966).

7. Acknowledgement

The author is grateful to the referee for providing constructive comments with regard to the ALGOL program.

References

Evans, D. J. (1966). Simplified Implicit Methods for the Finite Difference Solutions of Parabolic and Hyperbolic Partial Differential Equations, *JIMA*, Vol. 2, pp. 76-89.

Evans, D. J. (1971). The Numerical Solution of the Fourth Boundary Value Problem for Parabolic Partial Differential Equations, *JIMA*, Vol. 7, pp. 61-75.

Evans, D. J. (1972). An Algorithm for the Solution of Certain Tridiagonal Systems of Linear Equations, *The Computer Journal*, Vol. 15, pp. 356-359.

Hockney, R. W. (1965). A Fast Direct Solution of Poisson's Equation using Fourier Analysis, *JACM*, Vol. 12, pp. 95-113.

Peaceman, D. W. and Rachford, H. H. (1955). The Numerical Solution of Parabolic and Elliptic Differential Equations, *Journal SIAM*, Vol. 3, pp. 28-41.

Richtmyer, R. O. (1957). *Difference Methods for Initial Value Problems*, Interscience Pub. Inc. New York.

Rose, Donald J. (1969). An Algorithm for Solving a Special Class of Tridiagonal Systems of Linear Equations, *CACM*, Vol. 12, pp. 234-236.

Varga, R. S. (1963). *Matrix Iterative Analysis*, Prentice Hall, N.J.

Wachpress, E. L. and Habetler, G. J. (1960). An Alternating Direction Implicit Iteration Technique, *Journal SIAM*, Vol. 8, pp. 403-424.