A Fast Implicit Iterative Numerical Method for Solving Multidimensional Partial Differential Equations

William S. Helliwell*
Areté Associates, Inc., Santa Monica, Calif.

A new technique for solving the large system of linear algebraic equations associated with implicit differencing of multidimensional partial differential equations is presented. The coefficient matrix of the equations is factored, and then approximations to certain terms in the matrix are obtained from series expansions. The resulting system of equations is solved easily. The method is developed and demonstrated using a simple representative two-dimensional equation. Very good results are obtained when one direction is dominant.

I. Introduction

IMPLICIT finite-difference schemes for the solution of multidimensional partial differential equations are usually stable and therefore applicable to a large class of problems. However, they are difficult to implement and may require an excessive amount of computer storage and time. The long computing time arises from the need to solve the large system of linear algebraic equations that result from the differencing. The computing time can be reduced significantly by approximating the coefficient matrix of the linear equations with a matrix that produces a system of equations that are relatively easy to solve. Among such methods are the alternating direction method (ADI)[1] used by Beam and Warming[2] and Stone's strongly implicit method,[3] which has been tested by Lin et al.[4]

In this paper, a new technique for solving the large system of linear algebraic equations associated with implict differencing of multidimensional partial differential equations is presented. This method, called the pseudo-elimination method (PE), is shown to be faster than Stone's method for certain problems. The method is directly applicable to linear and linearized nonlinear systems of parabolic or elliptic partial differential equations. In order to discuss the method, a simple linear partial differential equation will be used; however, it should be kept in mind that the PE method is applicable to much more complicated problems.

The question of whether the method will work when applied to difficult problems is not addressed. The scope of this paper is limited to presenting the method and illustrating, via a simple problem, that the method has some merit and deserves further study.

II. Linear Equations

Consider the linear two-dimensional incompressible energy equation

$$\frac{\partial u}{\partial t}+AX\frac{\partial u}{\partial x}+AY\frac{\partial u}{\partial y}-KX\frac{\partial^2 u}{\partial x^2}-KY\frac{\partial^2 u}{\partial y^2}=q \qquad (1)$$

where t represents time; x and y space variables; KX, $KY \geq 0$; and AX, AY, KX, KY, q are functions of t, x, y. Suitable boundary conditions for a region in x,y space and initial conditions at $t=0$ throughout the region must be specified.

For simplicity, a rectangular region will be considered with a grid defined by

$$x_k=k\Delta x, \quad 0\leq k\leq K, \quad \Delta x=1/K$$

$$y_\ell=\ell\Delta y, \quad 0\leq \ell\leq L, \quad \Delta y=1/L$$

Time will be discretized by

$$t_j=j\Delta t, \quad j=0,1,2,\ldots$$

where Δt is a constant.

With the convention that $u_{jk\ell}$ is the discrete approximation to $u(t_j,x_k,y_\ell)$, and using a completely implicit differencing technique marching in time, Eq. (1) is approximated by

$$\frac{u_{j+1,k,\ell}-u_{jk\ell}}{\Delta t}+AX\frac{u_{j+1,k+1,\ell}-u_{j+1,k-1,\ell}}{2\Delta x}$$

$$+AY\frac{u_{j+1,k,\ell+1}-u_{j+1,k,\ell-1}}{2\Delta y}-KX\frac{u_{j+1,k+1,\ell}-2u_{j+1,k,\ell}+u_{j+1,k-1,\ell}}{\Delta x^2}$$

$$-KY\frac{u_{j+1,k,\ell+1}-2u_{j+1,k,\ell}+u_{j+1,k,\ell-1}}{\Delta y^2}=q_{j+1,k,\ell} \qquad (2)$$

for $1\leq k\leq K-1$, $1\leq \ell\leq L-1$. Additional equations are provided by the boundary conditions for $k=0$, $k=K$, $\ell=0$, $\ell=L$. If the equations are ordered $k=0$, $\ell=0$; $k=1$, $\ell=0$; ...; $k=K$, $\ell=0$; $k=0$, $\ell=1$; $k=1$, $\ell=1$; ...; $k=0$, $\ell=L$; $k=1$, $\ell=L$; ...; $k=K$, $\ell=L$, then the system of equations can be written in matrix notation

$$MU=F=Q+U_j/\Delta t \qquad (3)$$

where U is a column vector, as shown in Fig. 1, Q is a column vector composed of the $q_{j+1,k,\ell}$, and U_j is U at j instead of $j+1$.

M is a matrix with five nonzero diagonals (Fig. 1). For a specific equation, the main diagonal corresponds to $u_{j+1,k,\ell}$, the superdiagonal to $u_{j+1,k+1,\ell}$, and the subdiagonal to $u_{j+1,k-1,\ell}$, with a superdiagonal located $K+1$ elements away from the main diagonal corresponding to $u_{j+1,k,\ell+1}$, and a subdiagonal $K+1$ elements away from the main diagonal corresponding to $u_{j+1,k,\ell-1}$.

Most iterative schemes to solve Eq. (3) introduce another matrix N and define a sequence of iterates via

$$NU_{m+1}=(N-M)U_m+F; \quad m=0,1,2,\ldots; \qquad (4)$$

$$U_0=\text{initial guess}$$

Presented as Paper 77-644 at the AIAA 3rd Computational Fluid Dynamics Conference, Albuquerque, N. Mex., June 27-28, 1977; submitted July 15, 1977; revision received March 2, 1978.

Index category: Computational Methods.

*Senior Consultant.

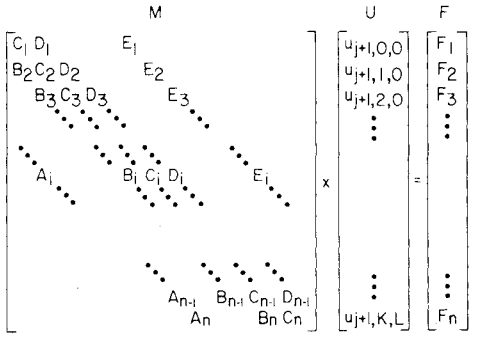

Fig. 1 Matrix structure.

or

$$N\delta U_{m+1} = -MU_m + F \tag{5}$$

where

$$\delta U_{m+1} = U_{m+1} - U_m$$

The problem is to choose an N that is easy to solve such that Eq. (4) produces iterates that converge. Many forms of N could be chosen which are easy to solve, the simplest being a diagonal matrix that would correspond to an explicit differencing scheme. Taking N equal to M minus the E superdiagonal produces the line Gauss-Seidel method, which was used by Lubard and Helliwell.[5,6] Stone[3] considered an N that was factorable into the product of upper and lower triangular matrices. Intuitively, a good matrix N would be one that is close to M in some sense. This is indeed the case, and a precise definition of N close to M will be given later.

In the method presented in this paper, a matrix N is not chosen directly but arises indirectly. The matrix M is a block tridiagonal matrix, that is,

$$M = \begin{bmatrix} S_0 & T_0 & & \\ R_1 & S_1 & T_1 & \\ & \ddots & \ddots & \ddots \\ & & R_L & S_L \end{bmatrix}$$

where S_ℓ is a tridiagonal matrix, and R_ℓ and T_ℓ are diagonal matrices. Each of the blocks is of order $K+1$. Since M is block tridiagonal, it can be factored to give

$$M = \begin{bmatrix} \beta_0 & & & \\ R_1 & \beta_1 & & \\ & \ddots & \ddots & \\ & & R_L & \beta_L \end{bmatrix} \begin{bmatrix} I & \gamma_0 & & \\ & I & \gamma_1 & \\ & & \ddots & \ddots \gamma_{L-1} \\ & & & I \end{bmatrix} \tag{6}$$

where I is the identity matrix, and

$$\beta_0 = S_0 \tag{7a}$$

$$\gamma_0 = \beta_0^{-1} T_0 \tag{7b}$$

$$\beta_\ell = S_\ell - R_\ell \beta_{\ell-1}^{-1} T_{\ell-1}; \quad \ell = 1,\ldots,L \tag{7c}$$

$$\gamma_\ell = \beta_\ell^{-1} T_\ell; \qquad \ell = 1,\ldots,L-1 \tag{7d}$$

Equation (3) can then be solved with a forward pass and a backward pass using the two factors of M. If U_ℓ denotes the column vector solution along the ℓth line, F_ℓ the corresponding right-hand side, and h_ℓ an intermediate $(K+1)$ vector, then the solution is obtained as follows:

Forward Pass

$$h_0 = \beta_0^{-1} F_0$$

$$h_\ell = \beta_\ell^{-1}[F_\ell - R_\ell h_{\ell-1}]; \quad \ell = 1,\ldots,L$$

Backward Pass

$$U_L = h_L$$

$$U_\ell = h_\ell - \gamma_\ell U_{\ell+1}$$

or

$$U_\ell = h_\ell - \beta_\ell^{-1} T_\ell U_{\ell+1}; \quad \ell = L-1, L-2,\ldots,0$$

The problem is that β_ℓ and γ_ℓ are, in general, full matrices that must be stored, and β_ℓ must be inverted. For large systems of equations, storage requirements and time consumption are prohibitive. However, both of these difficulties can be overcome by approximating β_ℓ^{-1} with a suitable matrix. From Eq. (7), β_ℓ^{-1} becomes

$$\beta_\ell^{-1} = [S_\ell - R_\ell \beta_{\ell-1}^{-1} T_{\ell-1}]^{-1}$$

which can be expanded in an infinite series as

$$\beta_\ell^{-1} = \left[\sum_{i=0}^{\infty} (S_\ell^{-1} R_\ell \beta_{\ell-1}^{-1} T_{\ell-1})^i\right] S_\ell^{-1} \tag{8}$$

The expansion is valid if the spectral radius of $S_\ell^{-1} R_\ell \beta_{\ell-1}^{-1} T_{\ell-1}$ is less than one. However, whether or not the expansion is valid, the formal manipulation leads to a useable approximation for β_ℓ^{-1}. Replacing $\beta_{\ell-1}^{-1}$ with a similar expression in Eq. (8) and dropping all terms involving products of four or more S_ℓ's produces

$$\beta_\ell^{-1} \approx (I + S_\ell^{-1} R_\ell S_{\ell-1}^{-1} T_{\ell-1}) S_\ell^{-1} \tag{9}$$

Substituting the inverse of the right-hand side of Eq. (9) for β_ℓ in Eq. (6) defines an N matrix to be used in Eq. (5). The algorithm to solve Eq. (5) for δU_{m+1} is the same as the algorithm to solve Eq. (3) for U. If $(U_\ell)_m$ denotes the mth iterate column vector solution along the ℓth line, then the forward and backward pass become as follows:

Forward Pass

$$h_0 = S_0^{-1}[-M(U_0)_m + F_0] \tag{10a}$$

$$h_\ell = (I + S_\ell^{-1} R_\ell S_{\ell-1}^{-1} T_{\ell-1}) S_\ell^{-1}[-M(U_\ell)_m + F_\ell - R_\ell h_{\ell-1}];$$

$$\ell = 1,\ldots,L$$

Backward Pass

$$(\delta U_L)_{m+1} = h_L \tag{10b}$$

$$(\delta U_\ell)_{m+1} = h_\ell - (I + S_\ell^{-1} R_\ell S_{\ell-1}^{-1} T_{\ell-1}) \times S_\ell^{-1} T_\ell (\delta U_{\ell+1})_{m+1}; \quad \ell = L-1,\ L-2,\ldots,0$$

In Eqs. (10a) and (10b), multiplications by S_ℓ^{-1} are obtained by solving a system of equations. For example, let v,w,z denote $(K+1)$ vectors; then the calculation of $(\delta U_\ell)_{m+1}$ in Eq. (10b) proceeds as follows: compute $z = T_\ell (\delta U_{\ell+1})_{m+1}$; solve $S_\ell\, v = z$; compute $z = T_{\ell-1}\, v$; solve $S_{\ell-1}\, w = z$; compute $z = R_\ell w$; solve $S_\ell\, w = z$; compute $(\delta U_\ell)_{m+1} = h_\ell - (v+w)$.

Since S_l is a tridiagonal matrix, then all matrix solutions indicated previously are easy to obtain, and very little extra storage is needed. (Refer to Isaacson and Keller[7] for a concise tridiagonal algorithm.) More rapid convergence could be achieved by including more terms in the approximation to β_l^{-1}; however, the amount of work would be increased. Before applying the method just described to a particular problem, a brief discussion of nonlinear problems and rate of convergence will be given.

III. Nonlinear Equations and Rate of Convergence

Differencing a nonlinear partial differential equation produces a system of nonlinear algebraic equations which can be written as

$$G(U)=0 \tag{11}$$

To solve Eq. (11), some form of iteration must be done. Since the Newton-Raphson method converges quadratically if the initial guess is close enough to the solution, then, when a good initial guess is known (this is very often the case for partial differential equations), this technique should be used. For large problems, character-handling languages such as FORMAC can be used to obtain the Jacobian.[8] If $M(U)$ is the Jacobian of G evaluated at U and U^0 is the initial guess, then the iteration procedure is defined as

$$M(U^n)\{U^{n+1}-U^n\}=-G(U^n);\quad n=0,1,2,\ldots \tag{12}$$

To solve this system of linear algebraic equations for each n, the technique in Sec. II is used. The linear iteration, Eq. (5), becomes

$$N\{U_{m+1}^{n+1}-U_m^{n+1}\}=-M(U^n)\{U_m^{n+1}-U^n\}-G(U^n);$$

$$m=0,1,2,\ldots;\quad U_0^{n+1}=U^n \tag{13}$$

N is a function of U^n.

The rate of convergence for the iteration procedure defined by Eqs. (12) and (13) will be studied now. Equation (13) with $m=0$ reduces to

$$U_l^{n+1}=U^n-N^{-1}G(U^n) \tag{14}$$

By an induction argument, it can be shown that, if m iterations are performed with Eq. (13), then

$$U_m^{n+1}=U^n-\{I-[I-N^{-1}M(U^n)]^m\}M^{-1}(U^n)G(U^n) \tag{15}$$

If U is the solution to Eq. (11), then, from Eq. (15) for a norm given by the symbol $\|\ \|$,

$$\|U_m^{n+1}-U\|=\|U^n-U-\{I-[I-N^{-1}M(U^n)]^m\}M^{-1}(U^n)$$

$$\times[G(U^n)-G(U)]\| \tag{16}$$

Since M is the Jacobian of G, then there is a U^* between U^n and U such that

$$G(U^n)-G(U)=M(U^*)\{U^n-U\}$$

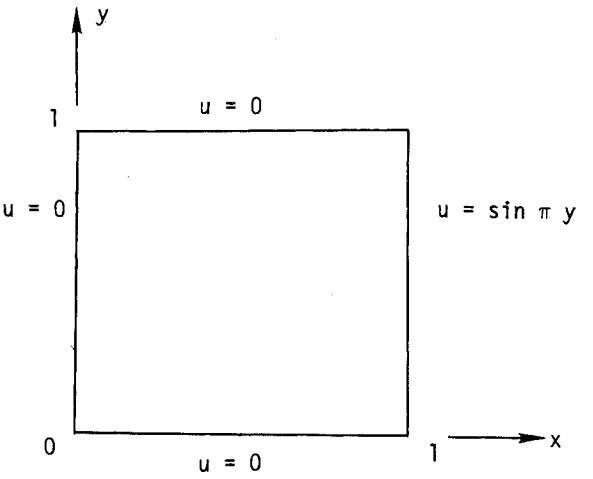

Fig. 2 Example problem.

Substitution into Eq. (16) yields

$$\|U_m^{n+1}-U\|\le\|I-\{I-[I-N^{-1}M(U^n)]^m\}$$

$$\times M^{-1}(U^n)M(U^*)\|\ \|U^n-U\|$$

As $n\to\infty$, $M^{-1}(U^n)\to M^{-1}(U)$, and $M(U^*)\to M(U)$. So,

$$\lim_{n\to\infty}\frac{\|U_m^{n+1}-U\|}{\|U^n-U\|}\le\|I-N^{-1}M(U)\|^m \tag{17}$$

Thus, if m iterations are done with Eq. (13), then the iterations defined by Eq. (12) will be improved by a factor of $\|I-N^{-1}M(U)\|^m$. Furthermore, from Eq. (14), it can be shown that doing only one linear iteration for each of m nonlinear iterations will also produce an improvement of $\|I-N^{-1}M(U)\|^m$. Therefore, for nonlinear problems, there is no point to iterate with Eqs. (12) and (13) but to use Eq. (14) instead. This greatly simplifies the computer program.

Equation (17) defines what is meant by N being close to M. The fastest rate of convergence is achieved when $\|I-N^{-1}M\|$ is the smallest for some norm. The question of whether $\|I-N^{-1}M\|$ is small for the pseudo elimination method presented in this paper is unanswered. For this method, the matrix N does not have a simple structure.

IV. Two-Dimensional Problem

The method has been applied to the steady form of Eq. (1) ($\delta u/\delta t\equiv 0$), with various values for the coefficients. Comparisons of accuracy as a function of work required have been made with Stone's strongly implicit method.[3]

For all cases, $K=L=30$, $u(0,y)=u(x,0)=u(x,1)=0$, $u(1,y)=\sin\pi y$, $Q=0$ everywhere, and the initial guess is identically zero (Fig. 2). The results for various values of the coefficients are presented in Table 1. The top number in each column is the factor by which the maximum residual error in Eq. (2) is reduced each iteration. The bottom number is the reduction after 10 iterations.

Table 1 Convergence results

	$KX=KY=1$ $AX=AY=0$	$KX=1, KY=0.1$ $AX=\sin 2\pi x$, $AY=\sin 2\pi y$	$AX=AY=0$	$KX=1, KY=0.01$ $AX=0.1\sin 2\pi x$, $AY=0.1\sin 2\pi y$	$AX=\sin 2\pi x$, $AY=\sin 2\pi y$	$AX=5\sin 2\pi x$, $AY=5\sin 2\pi y$	$AX=AY=-5$
PE	0.91 $10^{-0.42}$	0.52 $10^{-2.8}$	0.069 $10^{-11.6}$	0.069 $10^{-11.6}$	0.059 $10^{-12.3}$	Diverge	Diverge
Stone	0.67 $10^{-1.7}$	0.49 $10^{-3.1}$	0.24 $10^{-6.2}$	0.24 $10^{-6.2}$	0.48 $10^{-3.2}$	0.85 $10^{-0.71}$	Diverge

The effect of the neglected terms in the expansion, Eq. (9), is very apparent from the results presented in Table 1. For the centered differencing of Eq. (2), the R and T matrices vary directly with KY, and the S matrices vary with KX. So, for $KY \ll KX$, one would expect the spectral radius of $S_\ell^{-1} R_\ell \beta_\ell^{-1}{}_1 T_{\ell-1}$ to be small and the convergence of the sum in Eq. (8) to be rapid. Thus the approximation in Eq. (9) would be very good. Conversely, as AX increases, it would be expected that the S matrices would become less diagonally dominant, and as AY increases the R and T matrices become larger. The combination of these two results increases the spectral radius of $S_\ell^{-1} R_\ell \beta_{\ell-1}^{-1} T_{\ell-1}$, and the accuracy of the approximation in Eq. (9) deteriorates.

The pseudo-elimination method is very good for problems with a dominant direction, in fact twice as good as Stone's algorithm. For the uniform problem ($KX = KY = 1$), Stone's method converges faster; however, the pseudo-elimination method may be improved if the lines are interchanged every other iteration (like the alternating direction method).

Of the two methods, Stone's performs an iteration the fastest, and the pseudo-elimination method takes about 1.5 times as long. For large nonlinear problems where the coefficient matrix may have to be calculated every iteration, the time required to solve the equations is small compared to the total iteration time. The total running times for the different methods to perform one iteration would be much closer than just indicated. The rate of convergence is then the primary factor in the amount of time required to achieve a specified accuracy.

The storage requirements for the pseudo-elimination method are significantly less than Stone's method. The PE method needs only an amount of storage equal to the number of coefficients involved in two ℓ lines of the entire system of equations. This is comparable to the alternating direction methods.

V. Conclusions

The PE method is potentially applicable to any system of partial differential equations that, when differenced, produce a system of linear equations with the appropriate pentadiagonal coefficient matrix. The greatest chance of success for the method would be two-dimensional problems with one dominant direction.

One such problem is the high Reynolds number, laminar three-dimensional flow around a cone at angle of attack.[5,6] This flow problem reduces to a system of equations that are solved by marching in one space dimension, similar to Eq. (2). The PE method has been applied to the cone problem and resulted in a factor of 2 improvement in execution time over the line Gauss-Seidel method originally used on the problem.[9] The nonlinearities are handled as recommended in Sec. III. A stepsize restriction was necessary to guarantee convergence of the iterations. This result relates directly to the spectral radius of $S_\ell^{-1} R_\ell \beta_{\ell-1}^{-1} T_{\ell-1}$ and the validity of Eq. (9), since, as Δt decreases, the S matrices become large relative to the R and T matrices. Much more work is required to determine the benefits and limitations of the PE method.

References

[1] Douglas, J., Jr. and Gunn, J. E., "A General Formulation of Alternating Direction Methods," *Journal of Numerical Mathematics*, Vol. 6, 1964, pp. 428-453.

[2] Beam, R. M. and Warming, R. F., "An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation-Law Form," NASA Ames Research Center, 1976.

[3] Stone, H. L., "Iterative Solution of Implicit Approximations of Multidimensional Partial Differential Equations," *SIAM Journal of Numerical Analysis*, Vol. 5, Sept. 1968, pp. 530-558.

[4] Lin, C. L., Pepper, D. W., and Lee, S. C., "Numerical Methods for Separated Flow Solutions Around a Circular Cylinder," *Proceedings of the AIAA 2nd Computational Fluid Dynamics Conference*, 1975, pp. 91-100.

[5] Helliwell, W. S. and Lubard, S. C., "An Implicit Method for Three-Dimensional Viscous Flow with Applications to Cones at Angle of Attack," *Journal of Computers and Fluids*, Vol. 3, 1975, pp. 83-101.

[6] Lubard, S. C. and Helliwell, W. S., "Calculations of the Flow on a Cone at High Angle of Attack," *AIAA Journal*, Vol. 12, July 1974, pp. 965-974.

[7] Isaacson, E. and Keller, H. B., *Analysis of Numerical Methods*, Wiley, New York, 1966.

[8] Agopian, K., Collins, J., Lubard, S., and Helliwell, W., "NASA Viscous 3-D Flowfield Computations," R&D Associates, RDA-TR-6100-007, Oct. 1975.

[9] Gogineni, P. R. and Lewis, C. H., private communication, Dec. 1977.