Hybrid Computer Implementation of the Alternating Direction Implicit Procedure for the Solution of Two-Dimensional, Parabolic, Partial-Differential Equations

KENNETH A. BISHOP and DON W. GREEN

University of Kansas, Lawrence, Kansas

The solution of partial-differential equations is a problem which is encountered throughout the spectrum of engineering practice. Regardless of the discipline involved, mathematical description of a physical phenomenon which involves the transfer of mass, energy, or momentum through a geometric body implies solution of this class of equations. Within the realm of the chemical engineer, applications such as flow and/or chemical reaction in packed beds, heat and/or mass transfer in geometrically complex shapes, and the design of distributed parameter equipment, such as heat exchangers or tubular reactors, are typical problems which require the solution of partial-differential equations.

Several techniques, which involve finite-difference approximations, are available for solving this class of equations. In virtually all instances, the use of high speed computing equipment is essential to their application. For the subclass of parabolic, partial-differential equations, which is considered herein, both analogue and digital computers have been employed.

The use of analogue computers for implementing the finite-difference approximation algorithms has been reported by a number of authors (*1, 2, 6, 12*). In each case, the limiting factor on the size of problem which can be attacked lies in the parallel nature of analogue computation. Since at least one integrator is required for each grid node, only the very largest computation facilities are capable of handling significant problems.

The use of digital computers for implementing these algorithms has been very successful. As computer speeds have increased, so the size of problems solved has increased. It is not unusual, for example, to find petroleum reservoir engineering applications which employ models containing in excess of a thousand grid nodes (*4*). Unfortunately, the serial nature of digital computation combined with the requirements on step size or on the number of iterations for algorithm convergence often imply lengthy and therefore costly solutions.

The concept of the hybrid computer appears to offer a means for reducing the cost of solving many partial-differential equations. Extensive effort has been expended in solving problems in one space dimension by hybrid techniques (*7, 10, 11*). The work of Coughanowr et al. (*3*) on tubular reactor design provides an excellent example of the efficient blend of the two types of machines.

The application reported here proceeds by way of an example problem to adapt a well-known numerical technique for the solution of multidimensional partial-differential equations to implementation on a hybrid computer. The example problem is then solved for the purpose of comparing the precision of the hybrid solution with that of a standard digital algorithm. Finally, an estimate of the hybrid algorithm's speed is compared with that of the digital algorithm.

STATEMENT OF THE PROBLEM

An illustrative problem, the solution of which involves a partial-differential equation, is the prediction of transient temperature behavior in a flat plate in which there is a heat sink. For purposes of clarity in the illustration of the hybrid technique, a very uncomplicated problem has been chosen. It may be stated as follows. Predict the temperature, as a function of time and two space variables, extant in a square plate of constant thickness. The dimensions are L by L by h. Initially the plate is at a uniform temperature. At a specified time, heat removal at a constant rate is begun at a point sink which exists in the center of the plate, and this constant rate heat removal is continued for all time. The plate is insulated at top and bottom ($\pm h/2$) so that no heat transfer occurs there. Two adjacent edge boundaries are such that there is no heat flow across them. The two remaining edge boundaries are maintained at the constant initial temperature throughout all time. The flat plate material is assumed to be uniform and isotropic in the region of interest.

MATHEMATICAL MODEL

A mathematical model of the temperature behavior in the plate may be derived from an energy balance on a rectangular, parallel piped element in the plate. The resulting partial differential equation is

$$\frac{\partial T}{\partial t} = \alpha \left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right] + \lambda R \qquad (1)$$

in which $\lambda = 1.0$ at $x = y = L/2$ and 0.0 elsewhere. The parameters R and α are assumed constant.

The constant temperature and no-flow behavior at the plate edges is expressed by the following boundary conditions:

$$T(0, y, t) = T(x, 0, t) = T(x, y, 0) = T_{\text{initial}}$$
$$\left.\frac{\partial T}{\partial x}\right|_{L,y,t} = \left.\frac{\partial T}{\partial y}\right|_{x,L,t} = 0.0 \tag{2}$$

Definition of dimensionless equivalents of the time and space coordinates as well as for temperature, that is, $t^* = \alpha t/L^2$, $x^* = x/L$, $y^* = y/L$, and $T^* = T/T_{\text{initial}}$, permits writing of the equation in a dimensionless, semibounded form

$$\frac{\partial T^*}{\partial t^*} = \frac{\partial^2 T^*}{\partial x^{*2}} + \frac{\partial^2 T^*}{\partial y^{*2}} + \frac{\lambda R L^2}{\alpha T_{\text{initial}}} \tag{3}$$

in the region $0 \leq x^*, y^* \leq 1.0$, $0 \leq t^* \leq \infty$ with the boundary conditions

$$T^*(0, y^*, t^*) = T^*(x^*, 0, t^*) = T^*(x^*, y^*, 0) = 1.0$$
$$\left.\frac{\partial T^*}{\partial x^*}\right|_{1,y^*,t^*} = \left.\frac{\partial T^*}{\partial y^*}\right|_{x^*,1,t^*} = 0.0 \tag{4}$$

This type of equation appears in the description of many kinds of physical processes other than heat transfer. Notable are the cases of single-phase flow in a petroleum reservoir and diffusion with chemical reaction in a multidimensional system.

THE COMPUTATIONAL ALGORITHM

Difference-Differential Representation

The extents of the x^* and y^* coordinates are divided into n (an even number) segments of length $\Delta x^* = \Delta y^* = \Delta^* = 1/n$, generating a square grid containing $(n + 1)^2$ nodes. Subscripting the temperature extant at a node with an i, j ($i = 0, 1, \cdots n$; $j = 0, 1, \cdots n$), where the i, j corresponds to the column, row designation of the node, provides a convenient notation with which to identify its location. For example, the temperature at the heat sink located in the center of the region is expressed as

$$T^*_{\frac{n}{2},\frac{n}{2}}(t^*) = T^*_{\text{sink}}(t^*)$$

As long as attention is restricted to interior nodes ($i, j = 1, 2 \cdots n - 1$), the spatial derivatives appearing in the partial differential equation may be expressed as

$$\frac{\partial^2 T^*}{\partial x^{*2}} = (T^*_{i+1,j} - 2T^*_{i,j} + T^*_{i-1,j})/\Delta x^{*2} + O(\Delta^{*2})$$
$$\frac{\partial^2 T^*}{\partial y^{*2}} = (T^*_{i,j+1} - 2T^*_{i,j} + T^*_{i,j-1})/\Delta y^{*2} + O(\Delta^{*2}) \tag{5}$$

These expressions permit recasting the partial-differential equation as a set of simultaneous, first-order, ordinary, differential equations of the form

$$\frac{dT^*_{ij}}{dt^*} = \frac{T^*_{i+1,j} - 2T^*_{i,j} + T^*_{i-1,j}}{\Delta^{*2}} + \frac{T^*_{i,j+1} - 2T^*_{i,j} + T^*_{i,j-1}}{\Delta^{*2}} + \frac{\lambda_{ij} R L^2}{\alpha T_{\text{initial}}} + O(\Delta^{*2}); \quad T^*_{i,j}(0) = 1.0 \tag{6}$$

where $\lambda_{ij} = 1$ only for $i = j = n/2$ and 0 elsewhere.

Inclusion of Δ^{*2} in a new dimensionless time variable ($t = t^*/\Delta^{*2}$) and expression of the dimensionless temperature in terms of the difference $F_{ij}(t) = T^*_{ij}(t) - T^*_{ij}(0)$ leads to an expression for interior nodes ($i, j = 1, 2 \cdots n - 1$) of the following form:

$$\frac{dF_{ij}(t)}{dt} = F_{i+1,j}(t) - 2F_{ij}(t) + F_{i-1,j}(t) + F_{i,j+1}(t) - 2F_{ij}(t) + F_{i,j-1}(t) + \pi_{i,j}(0) + \frac{\lambda_{ij} R L^2 \Delta^{*2}}{\alpha T_{\text{initial}}} + O(\Delta^{*4}); \quad F_{ij}(0) = 0.0 \tag{7}$$

in which $\pi_{ij}(0) = T^*_{i+1,j}(0) + T^*_{i-1,j}(0) + T^*_{i,j+1}(0) + T^*_{i,j-1}(0) - 4T^*_{i,j}(0)$. The definition of $F(t)$ will be seen to have simplified the analogue portion of the algorithm and have permitted scaling for very precise calculation. The spatial boundary conditions become

$$F_{0,j}(t) = F_{i,0}(t) = 0 \qquad i, j = 0, 1, 2 \cdots n \tag{8}$$

$$\frac{dF_{n,j}(t)}{dt} = 2[F_{n-1,j}(t) - F_{n,j}(t)] + F_{n,j+1}(t) - 2F_{n,j}(t) + F_{n,j-1}(t) + \pi_{n,j}(0) + O(\Delta^{*4}); \quad F_{n,j}(0) = 0.0 \quad j = 1, 2 \cdots n - 1$$

$$\frac{dF_{i,n}(t)}{dt} = F_{i+1,n}(t) - 2F_{i,n}(t) + F_{i-1,n}(t) + 2[F_{i,n-1}(t) - F_{i,n}(t)] + \pi_{i,n}(0) + O(\Delta^{*4}); \quad F_{i,n}(0) = 0.0 \quad i = 1, 2 \cdots n - 1 \tag{9}$$

$$\frac{dF_{n,n}(t)}{dt} = 2[F_{n-1,n}(t) + F_{n,n-1}(t) - 2F_{n,n}(t)] + \pi_{n,n}(0) + O(\Delta^{*4}); \quad F_{n,n}(0) = 0.0$$

where

$$\pi_{n,j}(0) = 2T^*_{n-1,j}(0) + T^*_{n,j+1}(0) + T^*_{n,j-1}(0) - 4T^*_{n,j}(0)$$
$$\pi_{i,n}(0) = T^*_{i+1,n}(0) + T^*_{i-1,n}(0) + 2T^*_{i,n-1}(0) - 4T^*_{i,n}(0)$$
$$\pi_{n,n}(0) = 2T^*_{n-1,n}(0) + 2T^*_{n,n-1}(0) - 4T^*_{n,n}(0)$$

Equations (9) were obtained by using reflection conditions to describe the no-flow boundaries; that is

$$T_{i,n+1} = T_{i,n-1} \qquad 0 \leq i \leq n$$
$$T_{n+1,j} = T_{n-1,j} \qquad 0 \leq j \leq n$$

Alternating Direction Implicit Procedure (ADI) (5)

This well-known numerical technique provides a mechanism for replacing the solution of the n^2 simultaneous equations given by Equations (7), (8), and (9) with the repeated solution of a set of n simultaneous equations. This is accomplished by writing the equations twice, first explicit in one direction for a set time increment and then explicit in the other direction for an equal subsequent time increment. Looking at Equations (7), (8), and (9), we see that if the y direction (j index) derivative is evaluated using the constant temperature values that exist at the beginning of the time step, then the n simultaneous equations corresponding to a row can be solved. This can be repeated row by row for the set time increment $1/2\Delta t$. Next, the equations are written explicit in the x direction; that is, the x direction derivatives are evaluated by using the constant temperature values which existed at the end of the first calculation. This allows a column-by-column solution of n simultaneous equations. The entire process is then repeated, thus advancing the solution time step by

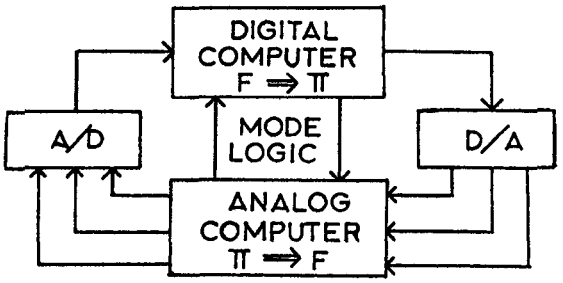

Fig. 1. Schematic representation of the computational responsibilities for the ADIP hybrid algorithm.

time step. For this calculation it is necessary to interpret $F_{ij}(t)$ as $[T^*_{ij}(t) - T^*_{ij}(k\Delta t/2)]$, where $k\Delta t/2$ is the time at the end of the previous time step ($k = 0, 1, 2, 3, \cdots \infty$).

Application of the ADI technique to Equations (7), (8), and (9) results in the following set of equations for a calculation which is implicit in the x^* direction and explicit in the y^* direction (calculation of the j^{th} row):

$$F_0(t) = 0.0$$

$$\frac{dF_1(t)}{dt} = F_2(t) - 2F_1(t) + F_0(t) + \pi_{1,j}(k\Delta t/2); \quad F_1(k\Delta t/2) = 0.0$$

$$\vdots$$

$$\frac{dF_i(t)}{dt} = F_{i+1}(t) - 2F_i(t) + F_{i-1}(t) + \pi_{ij}(k\Delta t/2) + \frac{\lambda_{ij}RL^2\Delta^{*2}}{\alpha T_{\text{initial}}}; \quad F_i(k\Delta t/2) = 0.0 \tag{10}$$

$$\vdots$$

$$\frac{dF_n(t)}{dt} = 2F_{n-1}(t) + 2F_n(t) + \pi_{n,j}(k\Delta t/2); \quad F_n(k\Delta t/2) = 0.0$$

where $k = 0, 2, 4, 6 \cdots \infty$. Note that if $j = n$, π_{1j}, π_{ij}, $\pi_{n,j}$ are replaced by π_{1n}, π_{in}, π_{nn} as previously defined.

It also should be noted that the terms $F_{i,j-1}(k\Delta t/2)$, $F_{ij}(k\Delta t/2)$, and $F_{i,j+1}(k\Delta t/2)$ which appear in the ADI representation of Equations (7), (8), and (9) are each zero by definition; therefore, the explicit nature of the y direction behavior is completely contained in the $\pi_{ij}(k\Delta t/2)$ term. Similarly, application of the technique on the subsequent time increment, where the x^* direction is handled explicitly and the y^* direction implicitly (calculation for column i), gives

$$F_0(t) = 0.0$$

$$\frac{dF_1(t)}{dt} = F_2(t) - 2F_1(t) + F_0(t) + \pi_{i,1}(k\Delta t/2); \quad F_1(k\Delta t/2) = 0.0$$

$$\vdots$$

$$\frac{dF_j(t)}{dt} = F_{j+1}(t) - 2F_j(t) + F_{j-1}(t) + \pi_{i,j}(k\Delta t/2) + \frac{\lambda_{ij}RL^2\Delta^{*2}}{\alpha T_{\text{initial}}}; \quad F_j(k\Delta t/2) = 0.0 \tag{11}$$

$$\vdots$$

$$\frac{dF_n(t)}{dt} = 2F_{n-1}(t) - 2F_n(t) + \pi_{i,n}(k\Delta t/2); \quad F_n(k\Delta t/2) = 0.0$$

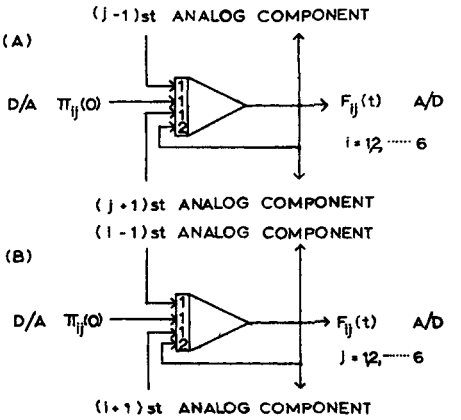

Fig. 2. General interior node implementations for (A) y direction and (B) x direction sweeps.

where $k = 1, 3, 5, 7 \cdots \infty$. Note that if $i = n$, $\pi_{i,1}$, π_{ij}, π_{in} become $\pi_{n,1}$, $\pi_{n,j}$ and π_{nn}. It should also be noted that with the exception of the $\pi_{ij}(k\Delta t/2)$ and λ_{ij}, these two sets of equations are identical.

Hybrid Algorithm

The implementation of the ADI technique by using a hybrid computer is quite straightforward. The computational responsibilities are assigned as follows:

1. Digital computer responsibility. Computation, storage, and appropriate disposition of π_{ij}, T^*_{ij} together with analogue mode control.
2. Analogue computer responsibility. Computation of F_i, F_j, together with generation of overload interrupts.
3. Digital logic (analogue computer). Execution of point sink introduction.
4. Interface. Analogue to digital conversion of F_i, F_j and digital to analogue conversion of π_{ij}, as well as transmission of analogue mode commands and interrupts.

These responsibilities are indicated schematically in Figure 1.

SAMPLE PROBLEM

The computational algorithm has been implemented for a very small (forty-nine node) grid in order to establish its validity without requiring extensive hybrid facilities. The analogue requirement was satisfied through the use of an Electronic Associates, Inc., TR-48/DES-30 iterative analogue computer. The digital requirement was met through the use of a desk calculator. The analogue/digital and digital/analogue conversion requirements were met by reading out the $F(t)$ information on a five-digit voltmeter and by manually setting the coefficient potentiometers corresponding to the $\pi(k\Delta t/2)$ information. The computation period and the heat-sink logic were controlled by the DES-30. Figures 2*a* and *b* are representations of general interior node implementations for x direction and y direction sweeps. The calculations of π_{ij} associated with Equations (7), (8), and (9) were executed by using the desk calculator.

The number 0.305 was chosen as a typical value for the dimensionless heat-sink strength ($\lambda_{ij}RL^2\Delta^{*2}/\alpha T_{\text{initial}}$) in Equation (7). This corresponds, for example, to a situation in which a 551 B.t.u./(hr.)(cu.ft.) heat sink is applied at the center of a 6 ft. by 6 ft. by 1 ft. 304 stainless steel plate (8), the initial temperature of which was 200°F.

A digital algorithm (FORTRAN IV) employing the Thomas technique for solution of the tridiagonal matrix produced by standard application of the ADI technique

TABLE 1. COMPARISON OF DIGITAL AND HYBRID RESULTS FOR DIMENSIONLESS PRESSURE DISTRIBUTION AFTER FIVE TIME INCREMENTS

Digital Results

Δy* \ Δx*	0	1	2	3	4	5	6
0	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
1	1.00000	0.99185	0.98329	0.97683	0.98076	0.98556	0.98743
2	1.00000	0.98329	0.96300	0.94131	0.95837	0.97167	0.97617
3	1.00000	0.97683	0.94131	0.86334	0.93545	0.96210	0.96931
4	1.00000	0.98076	0.95837	0.93545	0.95240	0.96611	0.97084
5	1.00000	0.98556	0.97167	0.96210	0.96611	0.97227	0.97485
6	1.00000	0.98743	0.97617	0.96931	0.97084	0.97485	0.97672

Hybrid Results

Δy* \ Δx*	0	1	2	3	4	5	6
0	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
1	1.00000	0.99188	0.98312	0.97692	0.98067	0.98554	0.98742
2	1.00000	0.98323	0.96312	0.94129	0.95847	0.97169	0.97613
3	1.00000	0.97691	0.94122	0.86396	0.93551	0.96214	0.96943
4	1.00000	0.98075	0.95802	0.93555	0.95247	0.96619	0.97078
5	1.00000	0.98557	0.97138	0.96210	0.96601	0.97225	0.97482
6	1.00000	0.98736	0.97617	0.96916	0.97071	0.97475	0.97669

to Equations (3) and (4) was written in order to check the hybrid algorithm's performance. This program was used to generate a solution which was identical in terms of grid and time step size to that produced by the hybrid algorithm.

The size of the dimensionless time step was chosen by running the digital algorithm repeatedly, halving the time increment at each repetition until the temperatures computed on successive runs differed by less than five parts in one hundred thousand. These temperatures were then taken as the true solution. The size of the dimensionless time increment Δt for the computation of the true solution and for the hybrid calculation was unity, which corresponds to a Δt of 5.32 hr. for the 304 stainless steel case.

RESULTS

The results of the hybrid calculation at dimensionless time $t = 5\Delta t$ are presented in Figures 3 and 4 and appear for purposes of a precision comparison with the true solution in Table 1. In Figure 3 the dimensionless temperature T^* is plotted vs. the dimensionless distance x^* for each of the seven values of y^* which correspond to node locations.

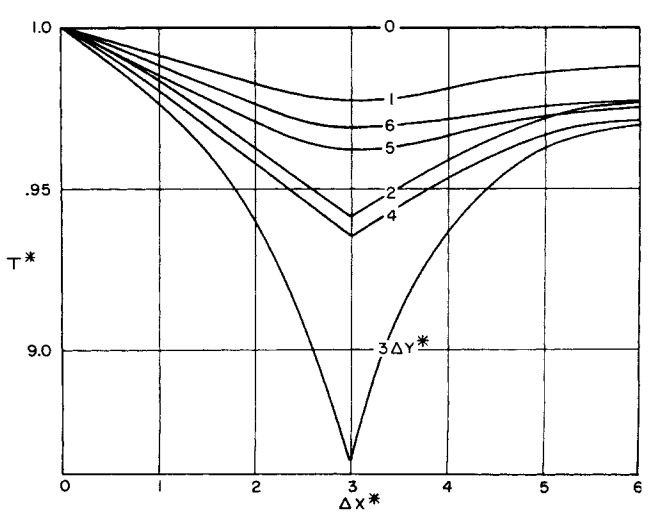

Fig. 3. Dimensionless temperature profiles after five time increments.

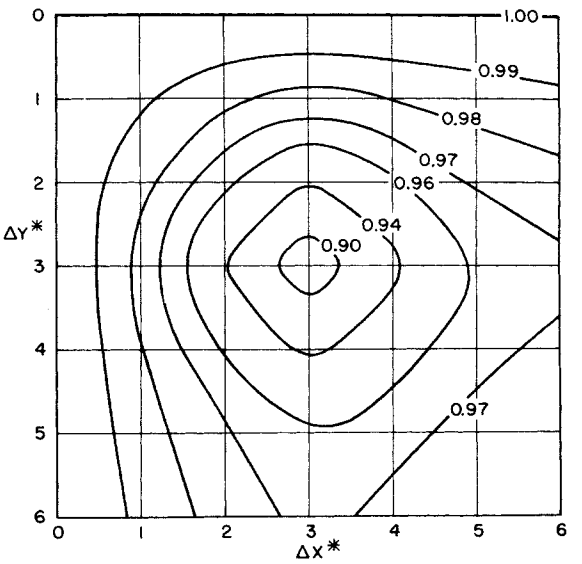

Fig. 4. Dimensionless temperature distribution after five time increments.

Figure 4 is an isopotential plot of the dimensionless temperature distribution after five time increments. This figure clearly shows the effect of the heat sink and of the boundary conditions on the temperature distribution.

A precision comparison between the true and hybrid solutions at each time increment revealed that the largest error occurred after one time increment and at the heat-sink node. This error was less than nine parts in one thousand. The magnitude of the errors steadily decreases with increasing dimensionless time; at five time increments, the largest error still at the heat-sink node is less than eight parts in ten thousand. The largest deviation from the diagonal symmetry apparent in the digital or true solution at five time increments is less than five parts in ten thousand. Typically, the magnitude of the errors is one part in ten thousand.

An estimate of relative computation speeds may be made by assuming that the analogue portion of a hybrid machine can execute its responsibilities during the time required for the digital calculations. This assumption is reasonable because of the ability to radically time scale analogue circuits, and the certainty of its validity improves with increasing grid size. If the assumption is valid, then one may simply count and compare the basic digital operations required for each algorithm.

As a satisfactory approximation for timing considerations, only the interior nodes are considered. For the hybrid calculation of each point at each $\Delta t/2$, the digital machine must compute

$$\pi_{i,j} = T^*_{i+1,j} + T^*_{i-1,j} + T^*_{i,j+1} + T^*_{i,j-1} - 4T^*_{i,j}$$

$$T^*_{i,j}[(k+1)\Delta t/2] = T^*_{i,j}(k\Delta t/2) + F_{i,j}[(k+1)\Delta t/2]$$

TABLE 2. OPERATION TIME DATA FOR GE 635 DIGITAL COMPUTER

Operation	GE 635 add-time units	Thomas tech. add-time units	Hybrid add-time units
Floating point add	1.0	4.0	5.0
Floating point multiply	2.2	4.4	2.2
Floating point divide	5.3	5.3	0.0
		13.7	7.2

that is, it makes five additions and one multiplication.

Using the nomenclature of Rachford and Peaceman (9), one may write the digital algorithm as

$$A_r T^*_{r-1} + B_r T^*_r + C_r T^*_{r+1} = D_r$$

where $A_r = C_r = -1$, $B_r = 2 + 2/\Delta t$, and $D_r = T^*_{i,j+1}{}^k - (2 - 2/\Delta t)\ T^*_{i,j}{}^k + T^*_{i,j-1}{}^k$. Then, the Thomas technique requires the following calculations:

$$W_r = B_r - A_r b_{r-1} = (2 + 2/\Delta t) + b_{r-1}; \quad W_1 = B_1$$
$$b_r = C_r/W_r = -1/W_r$$
$$g_r = (D_r - A_r g_{r-1})/W_r$$
$$= [T^*_{i,j+1}{}^k - (2 - 2/\Delta t)\ T^*_{i,j}{}^k + T^*_{i,j}{}^k + g_{r-1}]/W_r$$
$$T^*_r = g_r - b_r T^*_{r+1}$$

A close inspection of this algorithm for this application reveals that the W_r's and b_r's need be calculated only once; therefore, counting the computational operations gives a total of four additions, two multiplications, and one division as the requirement for each node calculation for each $(\Delta t/2)$ time step (13).

By assuming identical speed characteristics for both pure digital and hybrid digital machines and by using the speed-per-operation ratio of the GE 635 digital computer as an example, the estimate of relative speeds may be made (Table 2).

The hybrid algorithm requires 7.2 add-time units as compared with 13.7 add-time units for the digital algorithm. If all of the assumptions made in arriving at this estimate are valid, then one could expect a calculational speed increase of a factor of 1.9.

CONCLUSIONS

A viable algorithm for the solution of multidimensional partial-differential equations by using a hybrid computer has been elucidated. This algorithm consists basically of implementing a standard digital technique, the ADI procedure for a hybrid machine. Care in expressing the equations in terms of potential differences $F(t)$ simplifies the analogue program and permits scaling appropriate for very precise calculation.

The ADI hybrid algorithm has been applied to the solution of a small grid size problem, giving results which differ, typically, from those of a standard digital routine by less than three parts in ten thousand. Application of the technique to larger grid sizes, multiple source/sinks, and/or more complex boundaries presents no fundamental difficulties.

A conservative estimate of the hybrid algorithm's speed relative to that of the digital algorithm has been made which indicates that the hybrid calculation could be expected to be on the order of twice as fast as the digital implementation.

FUTURE EXTENSIONS

The simplicity of the application presented herein is desirable for purposes of illustration. Efforts are currently underway to examine the effect of size of spatial and time increments on the convergence and speed of the hybrid algorithm. The algorithm is implemented on a hybrid machine which combines an EAI 8800 analogue computer and a DDP 24 digital computer.

A modification of the basic algorithm which eliminates 3.2 add-time units from the base digital time requirement by shifting more computational responsibility to the analogue is also being evaluated.

Efforts are also underway to apply the algorithm to problems involving complications such as nonlinear variation of physical parameters, simultaneous material and energy balances, and three space dimensions.

ACKNOWLEDGMENT

The authors wish to acknowledge the cooperation of the University of Kansas Computation Center in providing the required GE 635 computer time.

NOTATION

Cp = heat capacity, B.t.u./(lb.$_m$/°F.)
F = $T^*(t) - T^*(k\Delta t/2)$ dimensionless temperature difference
h = thickness of plate, ft.
i, j = spatial increment indexes
k = time step index
L = maximum extent of x and y, ft.
n = maximum value of i, j
Q = volumetric heat flow rate, B.t.u./(hr.)(cu.ft.)
R = heat-sink term, °F./hr.
r = subscript for W, B, A, see reference 9
T = temperature, °F.
T^* = dimensionless temperature
T_{initial} = initial value of T
t = dimensionless time
t^* = dimensionless time
t = time, hr.
x, y = distances, ft.
x^* = dimensionless distance (x)
y^* = dimensionless distance (y)
W, B, A, C, b, g = functional groups appearing in Thomas technique

Greek Letters

α = thermal diffusivity
Δ^* = dimensionless distance increment
λ = switching parameter
π = dimensionless temperature residual
ρ = density, lb.$_m$/cu.ft.

LITERATURE CITED

1. Ames, W. F., "Non-Linear Partial Differential Equations in Engineering," Academic Press, New York (1965).
2. Bishop, K. A., "Solution of Partial Differential Equations Using A General Purpose Analog Computer," Computer Applications in the Earth Sciences: Colloquium on Simulation, Computer Contribution No. 22, State Geological Survey (Kansas) (1968).
3. Coughanowr, D. R., R. Vichnevetsky, J. P. Landauer, and A. Carlson, "Hybrid Simulation of a Control System for a Tubular Reactor," EAI Hybrid Computation Applications Study, 6.4. 2h (1965).
4. Dempsey, John R., Ph.D. thesis, Univ. Kansas, Lawrence (1966).
5. Douglas, J., Jr., and D. W. Peaceman, *AIChE J.*, **1**, No. 4, (1955).
6. Howe, R. M., and V. S. Haneman, *Proceedings IRE*, **4**, 1497-1508 (1953).
7. Mitchell, E. E. L., *SAG Rept. No. 19* (1963).
8. "Perry's Chemical Engineers' Handbook," 4 ed., pp. 23-36, McGraw-Hill, New York (1963).
9. Rachford, H. H., and D. W. Peaceman, *J. Ind. Appl. Math.*, 3, (1955).
10. Vichnevetsky, Robert, "A New Stable Computing Method for the Serial Hybrid Computer Integration of Partial Differential Equations," Thompson Book Co. (1968).
11. ———, "Application of Hybrid Computers to the Integration of Partial Differential Equations of the First and Second Orders," IFIP Congress 68, Edinburgh, Scotland (Aug. 5-10, 1968).
12. Witsenhausen, Hans, EAI *Princeton Computation Center Rept. No. 188* (June, 1961).
13. Peaceman, Donald W., Private communication (reviewer).