JOURNAL OF COMPUTATIONAL PHYSICS 12, 513–525 (1973)

Mesh Refinements for Parabolic Equations*

MELVYN CIMENT†

Mathematical Analysis Division, Naval Ordnance Laboratory, White Oak, Silver Spring, Maryland 20910

AND

ROLAND A. SWEET

Department of Mathematics, University of Colorado, Denver Center, Colorado Denver, 80203

Received December 28, 1972

Consider a finite difference approximation of the Crank–Nicolson type, using a local mesh refinement to an initial-boundary value problem for the parabolic equation

$$m(x)u_t = (s(x)u_x)_x + f(x, t).$$

We consider the above difference approximation to $u(x, t)$ on a grid pattern, where the middle third of the unit interval is refined with respect to the rest of the interval. Using the standard matrix arguments, we prove the matrix stability (spectral radius less than one) for the above class of problems. The local mesh refinement is achieved without loss of the second-order spatial accuracy, or of computational efficiency in solving the resulting equations. In two dimensions, we use our techniques to adapt the alternating direction schemes to a general mesh refinement pattern. Numerical experiments are presented which illustrate the computational effectiveness of the mesh refinement technique.

1. INTRODUCTION

Mesh refinement schemes for the numerical solution of initial-boundary value problems of parabolic type were proposed by Ciment in [1]. After obtaining very general results on difference shemes for initial-boundary value problems, Osher [3] and Varah [6] used the mesh refinement representation developed by Ciment in [2] to show that these parabolic methods are convergent. However,

* This research was partially supported by the Marathon Oil Company.
† This research supported in part by the Office of Naval Research, contract number NR 044-377.

581/12/4-6

their proofs were for implicit schemes on an infinite interval, a noncomputable case, since it gives rise to a doubly infinite matrix system.

In this paper we consider the actual computational problem that one faces in performing the implicit mesh refinement calculations on a finite interval. Failing to prove convergence, we present a proof of the matrix stability (eigenvalues all less than one in absolute value [7]) of the associated iteration matrix.

In two space dimensions, we show how to adapt the alternating direction schemes of Peaceman and Rachford [4] to a general mesh refinement pattern. For this case, we present the results of numerical experiments which indicate under what circumstances the various methods may be useful.

Some of the computations in this paper were carried out by J. P. Brzozowski of the Denver Research Center of Marathon Oil Company, and we welcome this opportunity to express our thanks to him.

2. Mesh Refinement Schemes

We are interested in approximating the solution of the following initial-boundary value problem. Consider

$$m(x)\, u_t = (s(x)\, u_x)_x + f(x, t) \tag{1a}$$

on the semi-infinite strip $\{(x, t) : 0 < x < 1,\ t > 0\}$, where m, s, and f are given functions satisfying

(i) $m(x) \in C[0, 1]$ and $\min_{0 \leqslant x \leqslant 1} m(x) = \underline{m} > 0$,

(ii) $s(x) \in C^1[0, 1]$ and $\min_{0 \leqslant x \leqslant 1} s(x) = \underline{s} > 0$.

Without loss of generality we take for boundary conditions

$$u(0, t) = u(1, t) = 0 \qquad (t > 0), \tag{1b}$$

and assume the initial condition is given by

$$u(x, 0) = \varphi(x) \qquad (0 \leqslant x \leqslant 1), \tag{1c}$$

φ being at least continuous on $[0, 1]$.

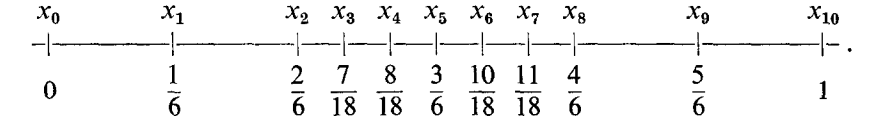
We approximate $u(x, t)$ with a grid using a uniform time step, but with two different spatial mesh patterns. More precisely, let $h > 0$ and $k > 0$ be given, and let $M > 1$ be a positive integer.

We define the grid R by

$$R = \begin{cases} t_n = nk, & n = 0, 1, 2,... \\ (x_j, t_n): x_j = \begin{cases} jh, & 0 \leqslant j \leqslant p \\ x_p + (j-p)\dfrac{h}{M}, & p+1 \leqslant j \leqslant q. \\ x_q + (j-q)h, & q+1 \leqslant j \leqslant N \end{cases} \\ x_{N+1} = x_q + (N+1-q)h = 1 \end{cases}$$

Note that p, q, and N depend on the refinement desired.

As an example, for $h = \frac{1}{6}$, and refining the grid in $[\frac{1}{3}, \frac{2}{3}]$ by a factor of $M = 3$ (hence, $p = 2$ and $q = 8$), we would have the grid depicted as



We define $v_{j,n}$ as the difference approximation to $u(x_j, t_n)$ which solves the following standard system of equations

$$\begin{aligned} v_{j,0} &= \varphi(x_j) \\ m_j(v_{j,n+1} - v_{j,n}) &= \lambda\{\theta[s_{j-1}v_{j-1,n+1} - (s_{j-1} + s_j)\, v_{j,n+1} + s_j v_{j+1,n+1}] \\ &\quad + (1-\theta)[s_{j-1}v_{j-1,n} - (s_{j-1} + s_j)\, v_{j,n} + s_j v_{j+1,n}]\} \\ &\quad + kf_{j,n}, \end{aligned} \tag{2a}$$

where $\lambda = (k/h^2)$, $m_j = m(x_j)$, $s_j = s(x_j + (h/2))$, and $f_{j,n} = f(x_j, t_n)$, for $1 \leqslant j \leqslant p-1$ and $q+1 \leqslant j \leqslant N$. The number $\theta \in [0, 1]$ may be chosen to obtain several of the well-known difference schemes (Richtmyer and Morton [5]). (For example, $\theta = \frac{1}{2}$ corresponds to the Crank–Nicolson scheme, and $\theta = 1$ is the standard implicit method.) For $p+1 \leqslant j \leqslant q-1$, we use (2a), with h replaced by (h/M). Note that this changes λ to λM^2, and changes $s(x_j + (h/2))$ to $s(x_j + (h/2M))$. There remains to define equations for the two interface points, x_p and x_q. For these we may use several approximations:

(i) The *3-point* central difference formula for nonuniform spacing, which is first-order accurate (see [7] Page 178).

(ii) A *4-point* difference formula for nonuniform spacing, which is second-order accurate.

(iii) The standard central difference formula, second-order accurate, but according to the consecutive ordering we have adopted, this involves two adjacent points and one point a distance of M points away. We call this the *integer-fit* method.

In our numerical experiments, we found that the third approximation is generally the most accurate method. In the following, we shall provide a proof which indicates the matrix stability [7] of our mesh refinement problem, using the approximation of type (iii). Our proof will clearly hold for the approximation of type (i). For type (ii), however, no such proof was found.

For type (iii) we are able to maintain an overall accuracy of $0(h^2)$, and we need very little extra computational effort, as will be demonstrated later. Hence, we take the remaining two difference equations as

$$\begin{aligned} m_p(v_{p,n+1} - v_{p,n}) = \lambda\{&\theta[s_{p-1}v_{p-1,n+1} - (s_{p-1} + s_p)\, v_{p,n+1} + s_p v_{p+M,n+1}] \\ &+ (1 - \theta)[s_{p-1}v_{p-1,n} - (s_{p-1} + s_p)\, v_{p,n} + s_p v_{p+M,n}]\} \\ &+ kf_{p,n} \end{aligned} \tag{2b}$$

and

$$\begin{aligned} m_q(v_{q,n+1} - v_{q,n}) = \lambda\{&\theta[s_{q-1}v_{q-M,n+1} - (s_{q-1} + s_q)\, v_{q,n+1} + s_q v_{q+1,n+1}] \\ &+ (1 - \theta)[s_{q-1}v_{q-M,n} - (s_{q-1} + s_q)\, v_{q,n} + s_q v_{q+1,n}]\} \\ &+ kf_{q,n}\,. \end{aligned} \tag{2c}$$

We now study the stability of this family of difference schemes. To accomplish this, let us rewrite the difference scheme in matrix notation as

$$\begin{aligned} V_0 &= \Phi \\ (D + \lambda\theta J)\, V_{n+1} &= [D - \lambda(1 - \theta)J]\, V_n + kF_n\,, \end{aligned} \tag{3}$$

where

$$V_n = \begin{pmatrix} v_{1,n} \\ v_{2,n} \\ \vdots \\ v_{N,n} \end{pmatrix}, \qquad F_n = \begin{pmatrix} f_{1,n} \\ f_{2,n} \\ \vdots \\ f_{N,n} \end{pmatrix}, \qquad \Phi = \begin{pmatrix} \varphi(x_1) \\ \varphi(x_2) \\ \vdots \\ \varphi(x_N) \end{pmatrix}$$

$D = \text{diag}\{m_1, m_2, ..., m_N\}$, and J is an almost tridiagonal matrix. In fact, its jth row is $\{0, ..., 0, -s_{j-1}, s_{j-1} + s_j, -s_j, 0, ..., 0\}$, for $1 \leqslant j \leqslant p - 1$ and $q + 1 \leqslant j \leqslant N$; and

$$\{0, ..., 0, -M^2 s_{j-1}, M^2(s_{j-1} + s_j), -M^2 s_j, 0, ..., 0\}, \qquad \text{for} \quad p + 1 \leqslant j \leqslant q - 1.$$

Each of the two remaining rows corresponding to the interfaces has three nonzero elements in them, one on the main diagonal, one beside it, and one a distance M away. Row p is

$$\{0, ..., 0, -s_{p-1}, s_{p-1} + s_p, 0, ..., 0, -s_p, 0, ..., 0\},$$

and row q is

$$\{0,\ldots, 0, -s_{q-1}, 0,\ldots, 0, s_{q-1} + s_q, -s_q, 0,\ldots, 0\}.$$

It is easily seen that $D^{-1}J$ is irreducible, and diagonally dominant (Varga [7]). Hence, by an extension of Gerschgorin's theorem, its eigenvalues τ_i satisfy

$$\mathrm{Re}(\tau_i) > 0, \qquad 1 \leqslant i \leqslant N.$$

Thus, the matrix $D + \lambda\theta J = D(I + \lambda\theta D^{-1}J)$ is nonsingular, insuring that V_{n+1} is always uniquely defined. Solving for V_{n+1} in (3), we obtain the iteration matrix $C = (I + \lambda\theta D^{-1}J)^{-1}[I - \lambda(1 - \theta)\, D^{-1}J]$.

To prove convergence, one needs to show that $\| C^k \|$ is uniformly bounded [5] as the grid sizes tend to zero. For the cases $\theta = 0$, 1, the standard maximum principle arguments yield such proofs. However, for general θ, we were unable to prove convergence. An obviously necessary condition for convergence, referred to as *matrix stability* (Varga [7]), requires that the spectral radius of C be less than or equal to one. This latter (weaker) type of stability holds for (2).

THEOREM. *The mesh refinement scheme* (2) *gives rise to a stable iteration matrix when*:

(a) $0 \leqslant \theta < \frac{1}{2}$, *if*

$$\lambda < \frac{\underline{m}}{2(1 - 2\theta)\, M^2 \bar{s}},$$

where $\bar{s} = \max_{0 \leqslant x \leqslant 1} s(x)$,

(b) $\frac{1}{2} \leqslant \theta \leqslant 1$, *for all* $\lambda > 0$.

Proof. The eigenvalues μ_i of C are given by

$$\mu_i = \frac{1 - \lambda(1 - \theta)\, \tau_i}{1 + \lambda\theta\tau_i}, \qquad 1 \leqslant i \leqslant N. \tag{4}$$

Let $\tau = \alpha + i\beta$, α, β real, be an eigenvalue of $D^{-1}J$. We know that $\alpha > 0$. By simple algebra we get, using (4),

$$1 > |\, \mu_i \,|^2 = \frac{[1 - \lambda(1 - \theta)\alpha]^2 + [\lambda(1 - \theta)\beta]^2}{[1 + \lambda\theta\alpha]^2 + [\lambda\theta\beta]^2}$$

is true if and only if

$$-2\alpha\lambda + \lambda^2(1 - 2\theta)(\alpha^2 + \beta^2) < 0$$

is true.

Since $\alpha > 0$, for $\frac{1}{2} \leqslant \theta \leqslant 1$, the above inequality is satisfied for all $\lambda > 0$. When $0 \leqslant \theta < \frac{1}{2}$, the above inequality is true if

$$\lambda < \frac{2}{1 - 2\theta}\left(\frac{\alpha}{\alpha^2 + \beta^2}\right).$$

An upper bound on λ, which insures our inequality, may be determined by finding the minimum of

$$g(\alpha, \beta) = \frac{\alpha}{\alpha^2 + \beta^2},$$

where $\alpha + i\beta$ lies in the union of the Gershgorin discs of $D^{-1}J$. A typical disc $\mathscr{D}_j$ has the form

$$\mathscr{D}_j = \{z : | z - r_j | \leqslant r_j, \text{ where } r_j = (D^{-1}J)_{jj}\}.$$

If $\alpha + i\beta \in \mathscr{D}_j$, then

$$(\alpha - r_j)^2 + \beta^2 \leqslant r_j^2.$$

Hence,

$$g(\alpha, \beta) \leqslant \frac{\alpha}{\alpha^2 + r_j^2 - (\alpha - r_j)^2} = \frac{1}{2r_j},$$

so

$$\min_{\alpha+i\beta \in \bigcup_{j=1}^{M} \mathscr{D}_j} g(\alpha, \beta) \leqslant \frac{1}{2 \max\limits_{1 \leqslant j \leqslant N} r_j} \leqslant \frac{\underline{m}}{4\bar{s}M^2}.$$

Finally, if we take

$$\lambda < \frac{\underline{m}}{2(1 - 2\theta)\,\bar{s}M^2},$$

then $| \mu_i | < 1$.

3. Solution of Equations

For $\theta > 0$, the solution of (3) at each time step involves solving a linear system of equations of the form

$$AV_{n+1} = B(V_n, F_n), \tag{5}$$

where A is almost a tridiagonal matrix and B is a vector formed from known quantities. Solving this system is best accomplished by initially factoring A as the

product of a lower unit triangular matrix L, and an upper triangular matrix R (Wilkinson [8]), and then, at each time step, solving two triangular systems of equations.

Since A is nearly tridiagonal, one would expect L and R to have nearly the same form—namely, bidiagonal—as when A is tridiagonal. We show now that very few nonzero elements are added in L and R, due to the form of A.

Let $A = (a_{ij})$ be a $N \times N$ matrix where $a_{ij} = 0$, if $|i - j| > 1$, except $a_{p,p+m} \neq 0$ and $a_{q,q-M} \neq 0$. Letting $L = (\ell_{ij})$ and $R = (r_{ij})$, one easily verifies that the nonzero elements of L are

$$\ell_{ii} = 1 \qquad (i = 1, 2,..., N)$$

$$\ell_{i,i-1} = \frac{1}{r_{i-1,i-1}} a_{i,i-1} \qquad (i = 2,..., N-1;\ i \neq q)$$

$$\ell_{q,q-M} = \frac{1}{r_{q-M,q-M}} a_{q,q-M}$$

$$\ell_{q,q-i} = \frac{1}{r_{q-i,q-i}} (a_{q,q-i} - \ell_{q,q-i-1} r_{q-i-1,q-i}) \qquad (i = M-1, M-2,..., 1),$$

and the nonzero elements of R are

$$r_{1,1} = a_{1,1}$$

$$r_{ii} = a_{ii} - \ell_{i,i-1} r_{i-1,i} \qquad (i = 2,..., N)$$

$$r_{i-1,i} = a_{i-1,i} \qquad (i = 2, 3,..., N;\ i \neq p + m)$$

$$r_{p,p+M} = a_{p,p+M}$$

$$r_{p+i,p+M} = a_{p+i,p+M} - \ell_{p+i,p+i-1} r_{p+i-1,p+M} \qquad (i = 1, 2,..., M-1).$$

An operation count (one operation is either a multiplication or division) shows that a total of $2(N-1) + 3(M-1)$ operations are required to compute L and R, initially. The term $3(M-1)$ is due to the "extra" elements of A. To solve the two tirangular systems at each time step, $3N - 2 + 2(M-1)$ operations are required, where the factor $2(M-1)$ is, again, from the „extra" elements of A. Clearly, M will be small compared to N, and so, very little additional work is required to solve the implicit system of equations.

4. Higher Dimensions

Consider the model initial-boundary value problem

$$\begin{aligned} u_t &= u_{xx} + u_{yy} + f(x, y, t); \qquad 0 \leqslant x, y \leqslant 1 \qquad t \geqslant 0 \\ u(x, y, 0) &= \Phi(x, y), \end{aligned} \tag{6}$$

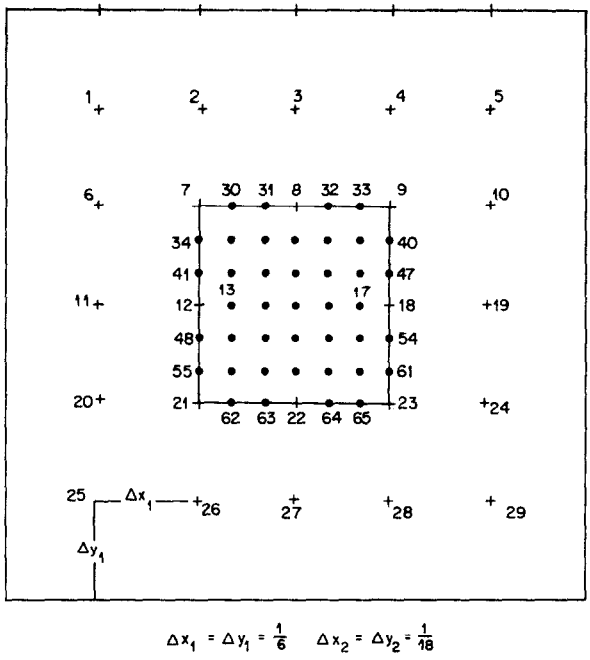


FIGURE 1

with the solution prescribed on the sides of the square. We treat the situation of refining the mesh in a region of interest (see Fig. 1) while maintaining a uniformly coarser grid elsewhere. The difficulty usually encountered is a sharp increase in the complexity of the equations to be solved. Our method avoids this by adapting the alternating direction (ADI) scheme of Peaceman and Rachford [4] to our general mesh refinement pattern. The ADI equations are

$$\begin{aligned} v_{j,k}^{n+1/2} - v_{j,k}^{n} &= \frac{\lambda}{2}\left[(\delta_x{}^2 v_{j,k})^{n+1/2} + (\delta_y{}^2 v_{j,k})^{n}\right] + \frac{\Delta t}{2} f_{j,k}^{n} \\ v_{j,k}^{n+1} - v_{j,k}^{n+1/2} &= \frac{\lambda}{2}\left[(\delta_x{}^2 v_{j,k})^{n+1/2} + (\delta_y{}^2 v_{j,k})^{n+1}\right] + \frac{\Delta t}{2} f_{j,k}^{n+1/2}, \end{aligned} \tag{7}$$

where $\lambda = \Delta t/\Delta x^2$, and $(\delta_x{}^2 v_{j,k})^n \equiv v_{j+1,k}^n - 2v_{j,k}^n + v_{j-1,k}^n$.

With ADI, it will be seen that our one-dimensional method becomes the basic component for higher dimensional mesh refinement problems. We were unable to prove the matrix stability of this procedure, but numerical experiments presented in Section 5 indicate that the method is accurate.

For simplicity, we consider a mesh pattern where the region

$$D_r = \{(x, y) : \tfrac{1}{3} \leqslant x \leqslant \tfrac{2}{3}, \tfrac{1}{3} \leqslant y \leqslant \tfrac{2}{3}\} \tag{8}$$

has different (finer) mesh spacings than the remaining region. Namely,

$$\Delta x_2 = \Delta y_2 = \frac{1}{18} \text{ in } D_r\,; \qquad \Delta x_1 = \Delta y_1 = \frac{1}{6} \text{ outside } D_r\,. \tag{9}$$

The way one numbers (orders) the mesh points (unknowns) can significantly alter the ease with which one can solve the resulting equations. Figure 1 exhibits the resulting pattern, containing 65 points and their ordering.

We now describe the approximation for the mesh pattern of Fig. 1. The generalizations to more complicated patterns will be straightforward and, thus, will be omitted from our discussion. Initially, say $t = n\,\Delta t$, we solve Eq. (7) for the grid values at the points 1–29. Note that to obtain these values on any one line, we solve the associated system of equations along that line as in a one-dimensional implicit method of Section 2. For this example, on the lines {1–5}, {6–10}, {20–24}, {25–29}, we obtain the standard tridiagonal systems. In this simple pattern only one "coarse" line cuts through D_r, namely, points {11–19}. The system for this line can be solved by the one-dimensional methods of Section 2.

Among values at {1–29}, we have also obtained the values, at time $t = (n + \frac{1}{2})\,\Delta t$, along all the "coarse" points of D_r. The data at the points {7, 12, 21, 9, 18, 23} allows us to use any desired degree of interpolation to obtain the values at the points {34, 41, 48, 55, 40, 47, 54, 61}. Now the x sweep inside D_r is carried out, using uniform alternating direction on {34–61}. Completing this, we now interpolate in the data on the top and bottom of D_r, at points {30–33} and {62–65}. Note that we describe this interpolation at the end to emphasize that in doing the x sweep in D_r, one uses the data from $t = n\,\Delta t$ on the top and bottom of D_r in the *explicit* part of (7). This completes the x sweep, and all our grid values are at $t = (n + \frac{1}{2})\,\Delta t$. The y sweep is carried out analogously, and completes the full time step.

In the previous example, the method on each line is the uniform grid method, except on the lines cutting through D_r. To investigate which condition is best to use at the interface points, we used all interface conditions listed in Section 2 in several computational experiments.

V. Computational Results

We solved Eq. (1a) on the interval [0, 4], using three different grid regions. The true solution $u(x, t) = \exp[x(-(37/6) + (13/6)\,x) + t]$ grows very rapidly near the right side. Starting with a $\Delta x_1 = .20$, we refined, at $x_p = 3.0$, by a factor of 10,

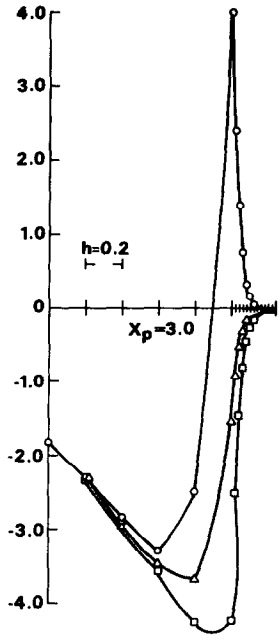


FIG. 2A. Crank–Nicolson Percent Error, 10:1 Mesh Ratio at $X = 3.0$, 20 time steps. △ *4PT*; ○ *3PT*; □ *INT*.

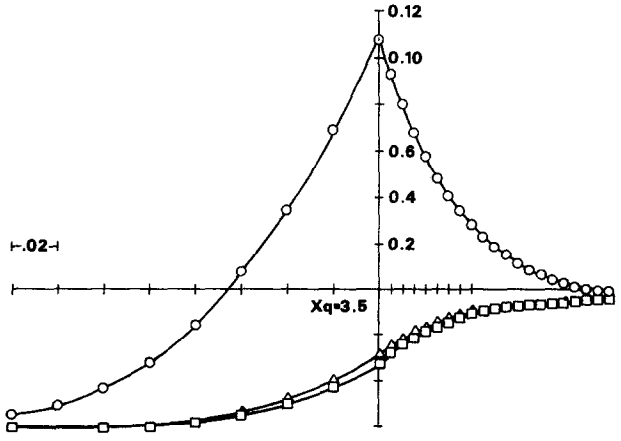


FIG. 2B. Crank–Nicolson Percent Error, 4:1 Mesh Ratio at $X = 3.5$. △ *4PT*; ○ *3PT*; □ *INT*.

so that $\Delta x_2 = .02$, and again refined at about $x_q = 3.5$, down to $\Delta x_3 = .005$, and $\Delta t = .01$. The computing times for all methods were within a 2% difference of each other.

The results shown in Figs 2A and 2B are characteristic for all our calculations. Namely, the *three-point* method causes an abrupt oscillation of the approximate solution, while the other two methods are smoothly varying across the refinement interface. For smooth problems, even though this overshoot creates a larger error for the *three-point* method, it will eventually be damped out, and can actually cause an improvement of the error profile after many time steps. However, in most situations the refined area is usually changed after several time steps, and in such cases one would expect that the *three-point* method might introduce many oscillations.

TABLE I

$\Delta x_1 = \Delta y_1 = 1/15$; $\Delta x_2 = \Delta y_2 = 1/90$; $P_1 = P_2 = \frac{1}{6}$; $\Delta t = 2 \times 10^{-4}$

Alternating direction	t, time	E_c	E_{int}
Mesh refinement			
(*a*) Integer-fit[a]	$10\Delta t$	3.7×10^{-2}	7.6×10^{-3}
(*b*) 4-point	$10\Delta t$	3.1×10^{-2}	7.8×10^{-3}
(*c*) 3-point	$10\Delta t$	5.5×10^{-2}	1.2×10^{-2}

[a] For the entries in (*a*), E_c is a 5% relative error while E_{int} is 0.12%.

In Table I, we list the results of solving (6) by using our method, with the three interface conditions of Section 2.

We took the analytical solution as

$$u(x, y, t) = \frac{1}{(t + \alpha)\beta} \cdot \exp\left\{\frac{(x - 1/2)^2 + (y - 1/2)^2}{-4(t + \alpha)}\right\},$$

with $\beta = 40$ and $\alpha = 1/432$.

To describe the accuracy of the refined scheme away from the interface, we compare the error in the interior of D_r at points at a distance of at least, a coarse mesh-width away from the boundary of D_r. The maximum deviation of our approximation from the solution over all such points we denote by E_{int}. The maximum error over the remaining points in D_r and over all the coarse points, we denote by E_c. Note that, for all our calculations, we always used third-order Lagrange interpolation along the interface.

For the time interval indicated in Table I, this function exhibits large gradients near D_r. In fact, at $t = 0$, $u(x, y, 0)$ has a maximum value of 10.8, and u_x, u_y

take on values greater than 80 in absolute value in D_T. However, the region of large gradients begins to spread out, and our method remains an effective computational procedure only if we have a dynamically changing grid. In this case, the relative errors reflect the effectiveness of the refinement technique.

TABLE II

Integer-fit: 10 time steps; $\Delta t = 0.006$

	E_i	E_{int}	Work estimate
1. No refinements $\Delta x = \Delta y = 1/15$	0.004608	0.010235	$\sim(15)^2$
2. No refinement $\Delta x = \Delta y = 1/60$	0.000277	0.000671	$\sim 16(15)^2$
3. Refinement $\Delta x_1 = \Delta y_1 = 1/15$ $P_1 = P_2 = 4$	0.00506	0.00201	$\sim 2.77(15)^2$

In Table II, we show the results for a case where the region of sharp gradients is more static. We solved for $u(x, y, t) = 1 + \exp[-20t[(x - \frac{1}{2})^2 + (y - \frac{1}{2})^2]]$ a solution of (6) on D_T as in (8), using the *integer-fit* method. With the mesh refinement, we achieve an increase of accuracy by a factor of, at least, five over the interior square (much more away from the interface), by using only 2.77 as many operations. As shown in Table II, the error, with no refinement, is inversely proportional to the number of grid points. Thus, the only other way to have obtained this increased accuracy in the *interior region* would have been to use a uniform grid

$$\Delta x = \Delta y = \frac{1}{15\sqrt{5}}.$$

This would have required nearly twice as much storage and calculation as compared to the mesh refinement scheme.

REFERENCES

1. M. CIMENT, Mesh Refinement Techniques—Fundamental Considerations, Internal Report No. 69-5, Marathon Oil Company, Denver Research Center, January, 1969.
2. M. CIMENT, Stable difference schemes with uneven mesh spacings, *Math. Comp.* **25** (1971), 219–227.
3. S. OSHER, Mesh refinements for the heat equation, *SIAM J. Numer. Anal.* **7** (1970), 199–205.
4. D. W. PEACEMAN AND H. H. RACHFORD, JR., The numerical solution of parabolic and elliptic differential equations, *SIAM J.* **3** (1955), 28–41.

5. R. D. RICHTMYER AND K. W. MORTON, "Difference Methods for Initial-Value Problems," 2nd Edition, Interscience, New York, 1967.
6. J. VARAH, Stability of difference approximations to the mixed initial boundary value problem for parabolic systems, *SIAM J. Numer. Anal.* **8** (1971), 598–615.
7. R. S. VARGA, "Matrix Iterative Analysis," Prentice-Hall, New York, 1962.
8. J. H. WILKINSON, "The Algebraic Eigenvalue Problem," Oxford University Press, London/ New York, 1965.