The explicit solution of the equation of heat conduction

By G. M. Birtwistle

In this paper we examine the n-dimensional extensions of the finite-difference schemes proposed by (*a*) Larkin, Lees, and Saul'yev and (*b*) du Fort and Frankel for the solution of the differential equation $u_t = u_{xx}$. The formulae obtained are explicit and stable for all values of the mesh ratios, but these ratios are found to be limited in practice by truncation error considerations. A comparison of the storage requirements and computing efficiency is made with other well known methods.

(First received November 1967 and in revised form May 1968)

1. Introduction

Consider the parabolic partial differential equation governing the conduction of heat in n space dimensions

$$u_t = \sum_{j=1}^{n} \frac{\partial^2 u}{\partial x_j^2}, \qquad (1)$$

subject to the initial condition that $u(x_1, x_2, \ldots, x_n; 0) = f(x_1, x_2, \ldots, x_n)$ where f is assumed to have a valid Fourier series expansion in the region of interest. The independent variables are taken to be bounded by $0 \leqslant x_1, x_2, \ldots, x_n \leqslant 1$ and $t \geqslant 0$, and we assume, for $t > 0$, that u takes the value zero on the bounding hyper-planes.

A solution to (1) is sought by embedding a grid into the region of integration and computing a finite-difference analogue of (1), at the nodal points, for as many time-steps as are deemed necessary. The following notation has been adopted:

k as the step-length along the t-axis,

h_j as the step-length along the x_j-axis,

$N_j = 1/h_j, \qquad (j = 1, 2, \ldots, n)$

$r_j = k/(h_j)^2.$

$$\Sigma = \sum_{j=1}^{n}, \quad N = \prod_{j=1}^{n} (N_j - 1), \quad R = \sum_{j=1}^{n} r_j.$$

$$U = u(p_1 h_1, p_2 h_2, \ldots, p_n h_n; qk) = u(p_j h_j; qk)$$

where

$p_j = 0, 1, 2, \ldots, N_j$ for $j = 1, 2, \ldots, n$.

Finally, we define the operators I, T, E_j, X_j and D_j by

$$IU = U,$$

$$T^M U = u(p_j h_j; (q + M)k),$$

$$E_j^M U = u(p_1 h_1, \ldots, p_{j-1} h_{j-1}, (p_j + M)h_j, p_{j+1} h_{j+1}, \ldots, p_n h_n; qk),$$

superfix omitted if $M = 1$

$$X_j = E_j + E_j^{-1} - 2I,$$

$$D_j = \frac{\partial}{\partial x_j}.$$

The $n + 1$ operators T and E_j clearly commute.

One class of methods for the solution of (1) may be written

$$(T - I)U = \Sigma r_j \{(sT + (1 - s)I)X_j\}U \qquad (0 \leqslant s \leqslant 1). \quad (2)$$

The classical explicit scheme, $(2, s = 0)$, has its choice of the n mesh ratios r_j restricted by stability considerations, which require $R \leqslant \frac{1}{2}$.

$(2, s = 1)$ is an implicit scheme, with no stability restrictions on R, which requires the inversion of a sparse square matrix of order N at each time-step. In practice one would use $(2, s = 1)$ only in one space dimension $(n = 1)$, and use the effective Alternating Direction Implicit methods of Peaceman, Rachford and Douglas [1, 8] for $n = 2$ or 3. The A.D.I. methods avoid the necessity of inverting a widely banded matrix at each time-step and require merely the repeated cyclic inversion of n tridiagonal symmetric matrices.

We shall examine the developments of two other explicit methods for the solution of (1), namely

(*a*) the Saul'yev–Larkin–Lees semi-explicit schemes [10, 6, 7] (abbreviated to SLL schemes) which overcome the more punitive storage requirements associated with the A.D.I. methods, although at the expense of an enlarged truncation error;

(*b*) the du Fort-Frankel schemes (DUFF) which were presented for the one-dimensional case in [2], and subsequently extended by Evans [3] for the solution of $u_{tt} + u_{xxxx} = 0$. The n-dimensional DUFF scheme has comparable storage and computing time requirements to the analogous SLL scheme, and links between the two schemes are explored in § 5.

In § 2, we note the storage locations needed and calculations required per mesh point for the standard schemes listed below.

(*a*) Explicit

$(2, s = 0)$ for $n = 1, 2, 3$.

(*b*) Implicit

$(2, s = 1)$ for $n = 1$ only.

* *Department of Applied Mathematics and Computing Science, The University, Sheffield.*

The A.D.I. methods of Peaceman and Rachford (PR), and Douglas and Rachford (DR) for $n = 2$ and $n = 3$ respectively.

In § 3 and § 4, SLL and DUFF schemes are developed, and the same computational details are noted for comparison.

2. Standard explicit and implicit schemes

In this, and the next two sections, we are to discuss, amongst other details, the calculations required per mesh point, and we introduce the notation $xM + yA + zD$ to denote x multiplications and y additions/subtractions and z divisions.

(a) Explicit

The classical explicit scheme is computed according to

$$TU = \{(1 - 2R)I + \Sigma\, r_j(E_j + E_j^{-1})\}U. \qquad (3)$$

If the r_j are all unequal, the calculation of each new component of the vector TU requires $(n + 1)M + 2nA + OD$. However, if the r_j are all equal, each new component of the vector TU needs but $2M + 2nA + OD$.

It is often the case that $R = \frac{1}{2}$, the maximum value allowed by stability restrictions, and then the totals listed above are each reduced by $1M + 1A + OD$ as the coefficient of IU is now zero. But more importantly, we are now able to step forward in time by computing only half the unknowns at any time-level as those unknown values of U on time-level $q + 1$ for which Σp_j is even(odd) are given entirely in terms of known U-values on time-level q for which Σp_j is odd(even). It follows that we need compute over our grid only those values of U for which $\Sigma p_j + q$ is either even (or odd), with a consequent halving of our expected work total.

The storage requirements can be effectively minimised to $O(N)$ locations when $R < \frac{1}{2}$ and $O(N/2)$ when $R = \frac{1}{2}$.

(b) Implicit

The fully explicit scheme $(2, s = 1)$ in the space dimension may be written as

$$(I - r_1X_1)TU = U. \qquad (4)$$

In higher space dimensions, $n = 2$ or 3, we are considering not $(2, s = 1)$ but the PR method [8] and the DR method [1], which in our notation are written

$$\left.\begin{aligned}(I - r_1X_1)T^{1/2}U &= (I + r_2X_2)U\\ (I - r_2X_2)TU &= (I + r_1X_1)T^{1/2}U\end{aligned}\right\}\text{(PR)} \qquad (5)$$

and

$$\left.\begin{aligned}(I - r_1X_1)T^{1/3}U &= (I + r_2X_2 + r_3X_3)U\\ (I - r_2X_2)T^{2/3}U &= T^{1/3}U - r_2X_2U\\ (I - r_3X_3)TU &= T^{2/3}U - r_3X_3U.\end{aligned}\right\}\text{(DR)} \qquad (6)$$

Thus for $n = 1$, 2 or 3, the implicit methods we are considering may be represented in the general matrix form

$$A_{jn}u_{q+j/n} = b \quad \text{(a known vector)} \qquad (7)$$

for each of the n sub-steps, where the matrices A_{jn} are, under a suitable re-ordering of the grid points, block diagonal with $N/(N_j - 1)$ leading submatrices C_j each of dimension $N_j - 1 (j = 1, 2, \ldots, n)$. Under this re-ordering, each of the C_j are of tridiagonal form, with components given by

$$(C_j)_{im} = \left.\begin{aligned}2r_j + 1 \quad & i = m\\ -r_j \quad & i = m \pm 1\\ 0 \quad & \text{otherwise.}\end{aligned}\right\} \qquad (8)$$

As the A_{jn} are block-diagonal, each sub-block can be solved for separately, and one way of solving a matrix equation of the form (7, 8) is to split the C_j into the product factors L_j and U_j with components

$$(L_j)_{im} = \begin{cases} w_{jm} & i = m\\ -r_j & i = m + 1\\ 0 & \text{otherwise}\end{cases}$$

and

$$(U_j)_{im} = \begin{cases} 1 & i = m\\ t_{ji} & i = m - 1\\ 0 & \text{otherwise}\end{cases}$$

respectively, where L_j and U_j have the same dimensions as C_j. The $\Sigma\,(2N_j - 3)$ unknowns, $w_{jm} (m = 1, 2, \ldots N_j - 1)$,and t_{ji} $(i = 1, 2, \ldots N_j - 2)$ are uniquely determined, and these decompositions are stable without interchanges due to the diagonal dominance of the C_j.

Having computed and stored the unknown elements of the matrices L_j and U_j the number of calculations required per mesh point for each of the n substeps totals

$n = 1$	$2N + 2A + 1D$	
	r_j unequal	r_j equal
$n = 2$	$4M + 4A + 1D$	$3M + 4A + 1D$
$n = 3$	$5M + 6A + 1D$	$3M + 6A + 1D$.

The number of storage locations required must include registers set aside to contain the components w_{jm} and t_{ji} of L_j and U_j. For $n = 2$, or 3, these will generally form a negligible proportion of the total. The number of locations required is then

$n = 1$	$O(3N)$
$n = 2$	$O(2N)$
$n = 3$	$O(3N)$

which are appreciable increases over the totals for the explicit schemes grouped as $(2, s = 0)$.

3. The SLL schemes

The basic SLL scheme is obtained from (2) by sharing the first term on the latter's right hand side between the forward and present time-levels. Collecting terms,

$$\{I + \Sigma\, r_j s(I - E_j)\}TU = \{I + \Sigma\, r_j(s(I - E_j) + X_j)\}U \qquad (9)$$

which can be written in matrix form as

$$C(s)u_{q+1} = D(s)u_q. \tag{10}$$

$C(s)$, the matrix representation of the space operator on the left hand side of (9), is a triangular matrix, and the solution of (10) is thus merely a process of forward substitution. (Equally well one could interchange the rôles of E_j and E_j^{-1} in (9) and produce another analogue of (1). There results a matrix problem which can be solved by backward substitution.)

Expanding (9) in the form of a Taylor series, one obtains

$$\{1 + \Sigma r_j s(-h_j D_j - h_j^2 D_j^2/2 - h_j^3 D_j^3/6 - \ldots)\}$$
$$ku_t + k^2 u_{tt}/2 + \ldots) = 2\Sigma r_j(h_j^2 D_j^2/2 + h_j^4 D_j^4/24 + \ldots)u,$$

i.e.

$$u_t = \Sigma(D_j^2 u + h_j^2 D_j^4/12 + sk/h_j D_j u_t + sk^2/h_j D_j u_{tt}) + \frac{k}{2}(sD_j^2 u_t - u_{tt}) + \ldots$$
$$= \Sigma D_j^2 u + O(h^2 + (1-s)k + k/h).$$

Thus as $h_j \to 0$ and $k \to 0$, we also require $k/h_j \to 0$ $(j = 1, 2, \ldots, n)$ for the scheme (9) to converge to the correct differential equation (1) (unless $s = 0$, in which case the scheme is the classical explicit formula $(2, s = 0)$). If the step lengths are equal and we take

$$h_j = h,\ r_j = r \qquad (j = 1, 2, \ldots, n),$$

then in practice this implies that $k/h = hr$ be 'small', which is more favourable as h decreases. The results of some numerical experiments are presented in § 6, and further comments on this scheme are left until there.

As the matrix $C(s)^{-1}D(s)$ does not have known eigenvalues, we analyse the stability of (16) by the Fourier series method. This assumes a set of errors $e(p_j h_j; q)$ exists at the grid point $(p_1 h_1, p_2 h_2, \ldots, p_n h_n; qk)$, and that $e(p_j h_j; q)$ has the finite trigonometric expansion

$$e(p_j h_j; q) = \sum_{t_1=1}^{N_1-1} \ldots \sum_{t_n=1}^{N_n-1} A(p_j; q) \prod_{m=1}^{n} \exp(ic_m p_m)$$

where $c_m = \pi t_m h_m$, and $A(p_j; q)$ represents the Fourier coefficients which are not required. Stability is assured if

$$|1 + \Sigma r_j s(1 - \exp(ic_m))| \geqslant |1 + \Sigma r_j s(1 - \exp(ic_m)) + 2\Sigma r_j(\cos(c_m) - 1)|$$

which implies

$$1 + s\Sigma r_j(1 - \cos(c_m)) \geqslant \Sigma r_j(1 - \cos(c_m))$$

i.e.

$$R \leqslant \frac{1}{2(1-s)}.$$

That R is not restricted when $s = 1$ agrees with results given by Larkin [6]. In the one dimensional case, this bound is certainly better than that derived by analysis of the spectral norm of the matrix $C(s)^{-1}D(s)$ (see [10], p. 37), and is considerably easier to obtain. In higher space dimensions, the work involved in attempting to obtain the stability criterion by spectral analysis would seem to be formidable.

Both Saul'yev [10] and Larkin [6] suggested ways of improving (9). As (9) is unconditionally stable only for $s = 1$, we follow Larkin in only using that value of s for the suggested improvements. In this case, (9) simplifies into

$$(T - I)U = \Sigma r_j\{(E_j - I)T + (E_j^{-1} - I)\}U, \tag{11}$$

and another scheme with similar properties is

$$(T - I)U = \Sigma r_j\{(E_j^{-1} - I)T + (E_j - I)\}U. \tag{12}$$

(i) Saul'yev and Larkin suggested using these alternately to produce an estimate of T^2U from U via TU by the formulae

$$\left.\begin{aligned}\{I + \Sigma r_j(I - E_j)\}TU &= \{I - \Sigma r_j(I - E_j^{-1})\}U \\ \{I + \Sigma r_j(I - E_j^{-1})\}T^2U &= \{I - \Sigma r_j(I - E_j)\}TU.\end{aligned}\right\} \tag{13}$$

Elimination of TU yields

$$\{I - \Sigma r_j X_j + \Sigma r_j(I - E_j^{-1})\,\Sigma r_j(I - E_j)\}T^2U = \{I + \Sigma r_j X_j + \Sigma r_j(I - E_j^{-1})\,\Sigma r_j(I - E_j)\}U.$$

Whilst this alternating scheme is obviously unconditionally stable, examination of the truncation error, assuming a valid Taylor expansion, shows (13) to approximate the equation

$$u_t = \Sigma(D_j^2 u + h_j^2 D_j^4 u/12) + (\Sigma(k/h_j)^2 D_j^2 u_t) - k(u_{tt} - \Sigma D_j^2 u_t) + \ldots$$

which represents (1) to within a truncation error of $O(h^2 + k^2 + (k/h)^2)$. This is a useful improvement for small (k/h), and the scheme (13) requires neither extra storage nor computing time over (11) or (12) alone.

(ii) Saul'yev and Larkin also proposed using (11) and (12) together to produce an averaged scheme for the solution of (1). It follows that such a scheme is equivalent to

$$\{I + \Sigma r_j(I - E_j)\}\{I + \Sigma r_j(I - E_j^{-1})\}TU = \tfrac{1}{2}\{2 - [\Sigma r_j(I - E_j)]^2 - [\Sigma r_j(I - E_j^{-1})]^2\}U. \tag{14}$$

Saul'yev suggested treating (14) as a new formula and solving it in that form, Larkin in the more tractable form of solving for TU by both (11) and (12) separately, and then averaging. The truncation error of this scheme, which has the same left hand side operator as (13), clearly contains a term of $O(k^2/h^2)$.

It requires double the computing time and storage space of (11, 12, 13) but it would be the only SLL scheme to give symmetric solutions to symmetric problems.

In n space dimensions, both the basic SLL schemes (11, 12) and the alternating form (13) require $(n+1)M + 2nA + OD$ calculations per point when the r_j are unequal, and $2M + 2nA + OD$ if the r_j are all equal. If $R = 1$, both these totals are reduced by $1M + 1A + OD$. However, there is no possibility with these schemes of computing over alternate points, as with the classical explicit scheme $(2, s = 0)$, and so the work total for

(11, 12, 13) with $R = 1$ is the same as for (2, $s = 0$) with $R = \frac{1}{2}$. The number of storage locations required is $O(N)$.

For the Larkin averaged scheme (29), these totals are doubled.

4. Truncation error and stability of the DUFF schemes

Du Fort and Frankel [2] suggested (for $n = 1$ only) taking the unstable three-level Richardson scheme and amending it by sharing the coefficient of U in it equally between TU and $T^{-1}U$.

In n dimensions, we would approximate (1) by

$$(T - T^{-1})U = 2\Sigma r_j(E_j + E_j^{-1} - T - T^{-1})U. \quad (15)$$

A Taylor series expansion yields

$$2ku_t + k^3u_{ttt}/3 + \ldots = 2\Sigma\{kD_j^2u + kh_j^2D_j^4u/12 + \ldots - (k^3/h_j^2)u_{tt} - (k^5/h_j^2)u_{tttt}/12 - \ldots$$

or

$$u_t = \Sigma(D_j^2u + h_j^2D_j^4u/12 - (k/h_j)^2u_{tt}) - k^2u_{ttt}/6 + \ldots$$

i.e.

$$u_t = \Sigma D_j^2u + O(h^2 + k^2 + n(k/h)^2).$$

For the stability analysis, we use the method detailed in Householder [5]. We write the finite difference analogue in matrix form as

$$(1 + 2R)u_{n+1} = 2\Sigma r_j Y_j u_n + (1 - 2R)u_{n-1}$$

where the n matrices Y_j are each representations of the operators $E_j + E_j^{-1}$. The Y_j thus have known eigenvalues and common eigenvectors. It follows from Householder's analysis that the stability of (15) hinges on the magnitudes of the roots of the quadratic equation

$$(1 + 2R)\mu^2 - 2\Sigma r_j y_j \mu - (1 - 2R) = 0. \quad (16)$$

We have stability if $|\mu_i| \leqslant 1$ whenever the y_j are replaced by eigenvalues of the corresponding Y_j. Since the y_j are of the form $y_j = 2\cos(\pi p_j h_j)$, it follows that $|\Sigma r_j y_j| < 2R$ and accordingly there is a v, dependent upon y_j, such that (16) may be replaced by

$$(1 + 2R)\mu^2 - 4R\cos v \,.\, \mu - (1 - 2R) = 0 \quad (17)$$

for any choice of y_j.

If the roots of (17) are complex, then they have equal modulus and

$$|\mu|^2 = |(1 - 2R)/(1 + 2R)| < 1.$$

If the roots of (17) are real, then

$$\mu = \frac{2R\cos v \pm \sqrt{[4R^2\cos^2 v + (1 - 4R^2)]}}{(1 + 2R)}$$

and since $R > 0$ and $0 < v < \pi$, this implies

$$0 < 4R^2\cos^2 v + 1 - 4R^2 < 1,$$

and it follows that $|\mu_i| < 1$.

We note that if $1 - 2R = 0$, then (15) becomes a two level scheme and then that one of the roots of (17) is zero. In this case, DUFF scheme (15) becomes identical to (7, $s = 0$) when $R = \frac{1}{2}$ in both formulae.

In [2], du Fort and Frankel pointed out that the computations using (15, $n = 1$) could be performed in two ways-under their so called 'leap frog' and 'pyramid' orderings. Both of these are of interest and both extend easily to $n > 1$.

In the leap frog ordering only one-half of the mesh values need be computed. For consider (15) for the calculation of the approximation to $u(p_jh_j; (p + 1)k)$. Let us denote all points for which $q + \Sigma p_j$ is even as E points, and odd as O points. Then all mesh points considered are either E or O points and from (15) it follows that if $(p_jh_j; (q + 1)k)$ is an E(O) then it is computed entirely in terms of E(O) points so that in the course of using (15), only E or only O points need be computed and stored. Thus as the computations proceed from time-level to time-level, we calculate successively the points for which Σp_j is even on one time-level and odd on the next, or vice versa.

Under the pyramid ordering we compute over one of the planes $t \pm \Sigma p_jh_j = \text{constant}$. Taking the positive sign, this reduces to a two level scheme centred on $(p_jh_j; t')$, where $t' = t + \Sigma x_j$, and yet omitting that time-level. In full

$$(T - T^{-1})U = 2\Sigma r_j\{(E_j - I)T + (E_j^{-1} - I)T^{-1}\}U \quad (18)$$

which can be used over planes t' even, or t' odd, without reference to intermediate levels.

Both these orderings reduce the expected work totals and locations required by one-half. Per point we perform $(n + 1)M + 2nA + OD$ calculations for r_j unequal and $2M + 2nA + OD$ for r_j all equal. If $R = \frac{1}{2}$, the leap frog scheme becomes two level, and the totals listed above are each reduced by $1M + 1A + OD$. $O(N)$ locations are needed for general R under both orderings.

5. Links between SLL and DUFF schemes

We first note that the DUFF leap frog and pyramid schemes are centred about a mid time level, whereas the SLL schemes are expanded about the backward level. Suppose now we do an SLL-type expansion on the classical unstable Richardson scheme instead of (2), sharing the right hand side of (1) between the forward and backward time-levels in the manner of (11, 12). This yields a formula identical to (18) in form, but one which is computed over lines $t = \text{constant}$. It is easy to show that with this ordering the formula is stable for all values of r_j and has truncation error of $O(h^2 + k^2 + k/h)$. It can, of course, be used in an alternating manner which further improves this to $O(h^2 + k^2 + k^2/h^2)$.† Since this new scheme can be computed over alternate time-levels then it requires no

† In § 6, these schemes are denoted by SLL*.

more storage space than $(9, s = 1)$. We have thus shown that this new SLL-type scheme and the DUFF pyramid scheme correspond exactly when computed over the points $(p_j h_j;\ t)$ and $(p_j h_j;\ t + \Sigma p_j h_j)$ respectively.

Now, consider the solution of the elliptic equation

$$\Sigma D_j^2 u = b(p_j h_j) \tag{19}$$

(in matrix form

$$(I - L - U)x = b)$$

by (*a*) successive over-relaxation; (*b*) fixed parameter first and second order Richardson methods.

(*a*) It is easy to show that the equation governing the error e^n of the nth iteration in the solution of (19) by S.O.R. is

$$e^{n+1} - e^n = \omega\Sigma(E_j e^{n+1} + E_j^{-1} e^n - 2Ie^n)/(2zh_j^2),$$

where $z = \Sigma\, 1/h_j^2$ and ω is the successive over-relaxation parameter. This may be rewritten as

$$e^{n+1} - e^n = (\omega/z(2 - \omega)) \,.\, \Sigma\,\{(E_j - I)e^{n+1} + (E_j^{-1} - I)e^n/h_j^2\} \tag{20}$$

i.e. the altering of $2Ie^n$ into $2Ie^{n+1}$ has no effect on convergence provided that ω is altered as shown. Now (20) is the same form as DUFF pyramid and the new SLL scheme, with

$$\omega/(z(2 - \omega)) = 2k \quad \text{or} \quad \omega = 4R/(1 + 2R).$$

Thus $0 < \omega < 2$, and this completes the correspondence.

(*b*) The solution of (19) by the first-order, single parameter (α) Richardson scheme yields an error equation

$$e^{n+1} - e^n = (\alpha/2z) \,.\, \Sigma\, X_j e^n/h_j^2$$

which is of the same form as $(7, s = 0)$ with

$$\alpha/2z = k \quad \text{i.e. } R = 2\alpha.$$

Using the second-order Richardson method with fixed parameters, α and β, the error in the solution of (19) obeys the equation

$$e^{n+1} - e^n = (\alpha/2z) \,.\, \Sigma X_j e^n/h_j^2 + \beta(e^n - e^{n-1})$$

which corresponds to DUFF leap frog with

$\alpha/2z = 2k/(1 + 2kz)$ and $\beta = (2kz - 1)/(2kz + 1)$. If α and β take their optimum values

$$\alpha = 4/(\sqrt{a} + \sqrt{b})^2 \quad \text{and} \quad \beta = \frac{(\sqrt{a} - \sqrt{b})^2}{(\sqrt{a} + \sqrt{b})^2},$$

where a and b are the maximum and minimum eigenvalues of $I - L - U$, then $a, b = -1 \pm \sqrt{(1 - 1/R^2)}$.

6. Numerical results and comments

The methods outlined were programmed to solve parabolic problems in one, two and three space dimensions. One such problem was

$$u_t = u_{xx} \quad : \quad u(0, t) = u(1, t) = 0, \quad t \geqslant 0$$

subject to the initial condition

$$f_1(x) = u(x, 0) = \sin(\pi x).$$

For various values of h and r, computed results for $10^5 \,.\, u(0{\cdot}5, t)$ are listed in **Table 1** in the format group

$$u(0{\cdot}5,\ 0{\cdot}2)$$
$$u(0{\cdot}5,\ 0{\cdot}4).$$

The rounded analytic solutions for $10^5 u$ are

13891
1930.

Further numerical problems included the initial values

$$f_2(x) = \begin{matrix} 1 & 0 < x < 1 \\ 0 & x = 0,\ 1 \end{matrix}$$

and

$$f_3(x) = 1 - |1 - 2x|,$$

and analogues of all three in higher space dimensions. The results quoted in Table 1 give an indication of the behaviour in the remaining problems.

Typically, we note that with the SLL and DUFF schemes the error does indeed diminish with decreasing k/h, and also that the Averaged SLL (14) and Alternating SLL (13) schemes give very similar results.

In this example, the basic SLL scheme (9) gives better results than either (13) or (14). This is not always so—a counter-example is provided by the methods of Evans [3] and Fairweather and Gourlay [4] for the solution of $u_{tt} + u_{xxxx} = 0$. Their solutions are achieved by splitting the fourth-order equation into an equivalent, simultaneous second-order pair $v_t = w_{xx}$, $w_t = -v_{xx}$, each of which can then be solved by our techniques. Given

$$v(x, 0) = x - x^2 \qquad 0 \leqslant x \leqslant 1$$
$$w(x, 0) = 0$$
$$v(0, t) = v(1, t) = w(0, t) = w(1, t) = 0,$$

Table 2 lists the error in computed values of u for $x = 0{\cdot}1, 0{\cdot}2, \ldots 0{\cdot}5$ at $t = 0{\cdot}02$ for $h = 0{\cdot}05$ and $r = \frac{1}{2}$ for SLL (9) and the Alternating SLL (13) schemes.

Table 2

Error in $10^7 \,.\, u$

METHOD	VALUE OF x 0·1	0·2	0·3	0·4	0·5
SLL	−301	−619	−669	−510	−134
AV.SLL	12	63	43	−7	−47

Error in $10^6 \,.\, u_{xx}$

SLL	331	2726	1694	2503	2555
AV.SLL	301	924	442	202	290

Table 1

Value of $10^5 . u$

$r = 1/2$

METHOD	VALUE OF h $h=1/10$	$h=1/20$	$h=1/40$	$h=1/80$
IMPLICIT	13435	13778	13863	13884
	1805	1898	1922	
SLL	14047	13930	13901	13894
	1960	1937	1931	
ALT.SLL	14658	14101	13946	13905
	2146	1988	1945	
AV.SLL	14599	14094	13945	13905
	2129	1986	1945	
DUFF	13435	13779	13863	13884
	1805	1898	1922	
SLL*	13829	13877	13888	13890
	1858	1912	1925	
ALT.SLL*	16292	14560	14067	13936
	2647	2119	1979	

$r = 2$

METHOD	VALUE OF h $h=1/10$	$h=1/20$	$h=1/40$	$h=1/80$
SLL	12871	13661	13835	13877
	1434	1811	1900	
ALT.SLL	22754	16401	14551	14060
	5191	2686	2117	
AV.SLL	20143	16068	14511	14055
	4000	2572	2104	
DUFF	634	11127	13223	13725
	−635	1216	1747	
SLL*	8361	12710	13620	13825
	−892	1389	1799	
ALT.SLL*		23538	16485	14556
		5548	2716	

$r = 1$

METHOD	VALUE OF h $h=1/10$	$h=1/20$	$h=1/40$	$h=1/80$
SLL	13829	13877	13888	13890
	1858	1912	1925	
ALT.SLL	16292	14560	14067	13936
	2647	2119	1979	
AV.SLL	15877	14511	14061	13935
	2508	2104	1977	
DUFF	11354	13268	13736	13852
	1265	1759	1887	
SLL*	12871	13661	13835	13877
	1434	1811	1900	
ALT.SLL*	22754	16401	14551	14060
	5191	2686	2117	

$r = 4$

METHOD	VALUE OF h $h=1/10$	$h=1/20$	$h=1/40$	$h=1/80$
SLL	8361	12710	13620	13825
	−892	1389	1799	
ALT.SLL	18704	23538	16485	14556
	3712	5548	2716	
AV.SLL	31716	21566	16230	14525
	10010	4600	2627	
DUFF	−29408	−35	11061	13211
	−32425	−654	1203	
SLL*		8276	12670	13609
		−953	1378	
ALT.SLL*		44392	23918	16533
		20254	5725	

In computing the SLL solutions in the one dimensional case, we are in fact solving a scheme representing

$$\alpha u_{xt} + u_t = u_{xx}$$

for 'small' α. Subjected to the transformation

$$x \to x$$

$$t \to t + \alpha x/2$$

this equation becomes

$$\frac{\alpha^2}{4} u_{tt} + u_t = u_{xx} \qquad (21)$$

(which is also the form of the DUFF scheme in one space dimension). It is easy to show that if the attenuation factor to $u_t = u_{xx}$ is λ, say, then the corresponding attenuation factors to (21) are

$$2\{-1 \pm \sqrt{(1 + \lambda\alpha^2)}\}/\alpha^2$$

or

$$-\frac{4}{\alpha^2} + \ldots$$

$$\sim$$

$$\lambda - \frac{1}{4}\alpha^2\lambda^2 + \ldots$$

This shows why α 'small' is a necessary condition for an accurate solution. (See also du Fort and Frankel [2], pp. 142–144.)

The reported results seem to indicate that out of the range of SLL and DUFF schemes (9) and (13) are to be preferred, and that a 'small value' of (h/k) would be less than 1/40, say. For very small values of h (1/80 or less) the schemes (9) and (13) can still therefore be competitive.

From the list of operations necessary per point, it can be seen that the Implicit and A.D.I. methods are not much more expensive than the Explicit schemes, and one can use a large value of R in them with more safety than in SLL schemes. Although they do require more working space, this is only likely to be serious in three dimensional problems on a small computer.

Acknowledgements

The author would like to acknowlege use of the facilities of the Computing Laboratory of Sheffield University, and of Norsk Regnesentralen, Oslo, where the computations were performed.

References

[1] Douglas, J., and Rachford, H. H. (1956). On the numerical solution of heat conduction problems in two or three space variables, *Tr. Am. Math. Soc.*, Vol. 82, p. 421.
[2] du Fort, E. C., and Frankel, S. P. (1953). Stability conditions in the numerical treatment of parabolic differential equations, *M.T.A.C.*, Vol. 7, p. 135.
[3] Evans, D. J. (1965). A stable explicit method for the finite difference solution of a fourth order parabolic partial differential equation, *Comp. J.*, Vol. 8, p. 280.
[4] Fairweather, G., and Gourlay, A. R. (1967). Some stable approximations to a fourth-order parabolic partial differential equation, *Math. Comp.*, Vol. 21, p. 1.
[5] Householder, A. S. (1958). The approximate solution of matrix problems, *JACM*, Vol. 5, p. 205.
[6] Larkin, B. K. (1964). Some stable explicit difference approximations to the diffusion equation, *Math Comp.*, Vol. 18, p. 196.
[7] Lees, M. (1961). Alternating direction and semi-explicit difference methods for parabolic partial differential equations, *Num. Math.*, Vol. 3, p. 398.
[8] Peaceman, D. W., and Rachford, H. H. (1955). The numerical solution of parabolic and elliptic differential equations, *JSIAM*, Vol. 3, p. 28.
[9] Richtmyer, R. D. (1957). Difference methods for initial value problems, *Interscience Tracts in Pure and Applied Mathematics, Tract* 4, New York.
[10] Saul'yev, V. K. (1964). *Integration of equations of the parabolic type by the method of nets*, Trans. G. J. Tee, Pergamon.

Book Review

Numerical Integration, by P. J. Davis and P. Rabinowitz, 1967; 230 pp. (Waltham, Massachusetts: *Blaisdell Publishing Company*, $7.50.)

This monograph is intended to give a concise but comprehensive account of the available formulas and methods for the numerical evaluation of integrals. The subject is presented from the application point of view and the treatment is modern and computer oriented.

Chapter 1 contains an introduction to the concept of numerical integration and also gives some useful analytical background material. Chapter 2, by far the longest, deals with approximate integration over a finite interval and includes special sections on the treatment of periodic functions, rapidly oscillatory functions, contour integrals, improper integrals and indefinite integration. In Chapter 3 is a brief description of special methods and formulas for integration over an infinite range, while Chapter 4 covers the subject of error analysis and includes a treatment of the effect of round-off error following the general analysis given by J. H. Wilkinson. The effect of truncation error is analysed through Peano's theorem, differences, the theory of analytic functions, and functional analysis. The topic of multi-dimensional integrals follows and includes a large section devoted to Monte Carlo methods. The last chapter deals with automatic integration schemes, which are classified as adaptive or non-adaptive and iterative or non-iterative, and has special sections devoted to Romberg's scheme and methods using Chebyshev polynomials. The book is concluded by five appendices containing a reprint of the article 'On the Practical Evaluation of Integrals' by M. Abramowitz, some FORTRAN programs, and bibliographies of ALGOL procedures, tables, books and articles.

The text is exceptionally well written, and very well organised, and the link between theory and application is, throughout, strongly forged with the help of numerous illustrative examples. The authors have carefully dug into the literature and concluded each section with a useful set of references. The book is a 'must' for any one interested in the practical evaluation of integrals, and will undoubtedly be of great value to all who wish to make a serious study of numerical integration.

P. C. Chakravarti (London)