JOURNAL OF COMPUTATIONAL PHYSICS 27, 256–271 (1978)

The Numerical Solution of Linear Time-Dependent Partial Differential Equations by the Laplace and Fast Fourier Transforms

E. HUNTLEY, W. M. PICKERING, AND A. S. I. ZINOBER

Department of Applied Mathematics and Computing Science, University of Sheffield, England

Received January 3, 1977; revised August 1, 1977

The method proposed may be used to solve a large class of linear parabolic and hyperbolic equations involving as independent variables time and two further spatial variables on a rectangular domain. The Laplace transform is used to reduce by one the number of independent variables, the resulting subsidiary equation having then to be one of the class which may be solved numerically by the FFT method. An effective numerical Laplace transform inversion algorithm is used to recover the solution of the original equation. The method has been successfully applied to a variety of test problems and found to be as accurate as established ADI methods. It is easily programmed and handles a wide range of time-varying boundary conditions and source functions without complication. For such problems it can be very competitive with regard to computation times. The method possesses the single-step property in that the solution at a particular time point may be evaluated without requiring the solution at intermediate time points, in contrast with the requirement for most finite difference methods. Both parabolic and hyperbolic equations are solved by precisely the same algorithm.

1. INTRODUCTION

The use of numerical Laplace transform algorithms in the solution of linear parabolic or hyperbolic partial differential equations involving time (or indeed a spatial variable) defined on the semi-infinite interval $[0, \infty)$ has been proposed in [1, 2]. A practical implementation using a particular numerical inversion scheme for the Laplace transform has been discussed by Zinober and Huntley [3].

The essential idea is that after applying the Laplace transform to the PDE (which may have source functions and/or boundary conditions which are time varying), the number of independent variables is reduced by one but the equation involves instead a complex parameter s. This (elliptic) equation may be solved by any suitable numerical method giving values of the Laplace transform of the solution over a spatial grid. The solution itself is recovered by a numerical Laplace inversion algorithm which means in practice that for a given time point the elliptic equation has to be solved for several specified values of the parameter s. The method is a single-step method in that the computations need only be performed for the specific time points for which the solution is required, in contrast to the several time steps required by conventional finite difference methods.

0021-9991/78/0272-0256$02.00/0

The purpose of the present paper is to put forward a novel combination of numerical transform techniques by employing the fast Fourier transform (FFT) to solve the subsidiary elliptic equation, defined over a rectangular spatial domain. Since the dependent variable is now complex it means that small changes have to be made to the FFT algorithms for real equations discussed in [4–7].

The actual FFT algorithms used incorporated the five-point or nine-point approximations to the subsidiary elliptic equation. The numerical Laplace transform inversion algorithm used was the one developed in [8–13] based upon the Padé approximation to e^z, but there are others in the literature which could equally well have been used, for example, [14].

The essential features of these algorithms are described and results are presented for four test problems. These demonstrate that the method gives results of comparable accuracy with standard finite difference algorithms. The merits of the method are discussed in the concluding section.

2. Formulation of the Problem and the Use of the Laplace Transform

The use of the Laplace transform to reduce the number of independent variables in the partial differential equation and boundary conditions is a device which can in principle be applied to a very large class of problems [1–3]. Here we concentrate on a restricted class for which the resulting transformed equation can be easily solved by FFT methods. The spatial domain is taken to be rectangular. Using the classification of Le Bail [6] the transformed equation must take the form

$$\left[a_1 \frac{\partial^2}{\partial x^2} + a_2(y) \frac{\partial^2}{\partial y^2} + a_3 \frac{\partial}{\partial x} + a_4(y) \frac{\partial}{\partial y} + a_5(y)\right] U(x, y) = \rho(x, y) \qquad (2.1)$$

which includes as special cases, Poisson's equation with Dirichlet, Neumann, or periodic boundary conditions, and parabolic or hyperbolic equations with appropriate initial and boundary conditions. Since the coefficients a_i may be functions of an additional complex parameter s, the most general type of equation which would after Laplace transformation lead to an equation of form (2.1) is

$$\left[F_1(D) \frac{\partial^2}{\partial x^2} + F_2(D) \frac{\partial^2}{\partial y^2} + F_3(D) \frac{\partial}{\partial x} + F_4(D) \frac{\partial}{\partial y} + F_5(D)\right] u(x, y, t) = g(x, y, t) \quad (2.2)$$

in which $D \equiv \partial/\partial t$ and each of the $F_i(D)$ expressions is a linear operator of the form

$$F_i(D) = b_{i,r}D^r + b_{i,r-1}D^{r-1} + \cdots + b_{i,0}\,.$$

The coefficients b_{ij} are allowed to be functions of y for $i = 2, 4$, or 5.

If we denote by $U(x, y, s)$ the Laplace transform of $u(x, y, t)$ with respect to t, then the transform of a typical term in equation (2.2) is

$$\mathscr{L}\left[D^j \frac{\partial^k u}{\partial y^k}\right] = s^j \frac{\partial^k U}{\partial y^k} - s^{j-1} \frac{\partial^k u}{\partial y^k} - s^{j-2} D \frac{\partial^k u}{\partial y^k} - \cdots - D^{j-1} \frac{\partial^k u}{\partial y^k}$$
$$k = 0, 1, 2; j = 1,..., n, \qquad (2.3)$$

where the terms $\partial^k u/\partial y^k,..., D^{j-1}(\partial^k u/\partial y^k)$ have to be specified at time $t = 0$. Thus, for example,

$$\mathscr{L}\left[F_2(D) \frac{\partial^2 u}{\partial y^2}\right] = F_2(s) \frac{\partial^2 U}{\partial y^2}(x, y, s) + H_2(x, y, s)$$

where $H_2(x, y, s)$ represents all the initial condition information. Hence the transform of Eq. (2.2) can be written as

$$\frac{\partial^2 U}{\partial x^2} + \frac{F_2(s)}{F_1(s)} \frac{\partial^2 U}{\partial y^2} + \frac{F_3(s)}{F_1(s)} \frac{\partial U}{\partial x} + \frac{F_4(s)}{F_1(s)} \frac{\partial U}{\partial y} + \frac{F_5(s)}{F_1(s)} U = \frac{G(x, y, s) + H(x, y, s)}{F_1(s)} \qquad (2.4)$$

where $H(x, y, s) = \sum_{i=1}^{4} H_i(x, y, s)$ and $G = \mathscr{L}[g]$. It is assumed that s is not a zero of $F_1(s)$, which is a reasonable assumption, as may be seen later from (3.3). Equation (2.4) has to be solved for $U(x, y, s)$ for particular values of the complex variable s and thus U is itself complex.

The boundary conditions for (2.4) are obtained by taking the Laplace transform with respect to t of the conditions specified for (2.2). Thus, for example, suppose that on the line $y = L$ it is given that

$$u(x, L, t) = w(x, t),$$

then the corresponding transformed boundary condition for Eq. (2.4) is

$$U(x, L, s) = W(x, s).$$

If, in particular, u is independent of t on the boundary then

$$u(x, L, t) = w(x) \qquad \text{and} \qquad U(x, L, s) = w(x)/s.$$

3. Description of the Algorithms Used

3.1. *Fast Fourier Transform Algorithms*

In order to construct a complete solution to Eq. (2.1) for a particular value of t, it is necessary to solve (2.4) several times for different complex values of s. Hence an efficient method of solution of (2.4) is desirable. For this reason the method of fast Fourier transforms, described for Poisson's equation by Hockney [4, 5] and for a more general class of partial differential equations by Le Bail [6], was employed.

In the remainder of this section a brief description of both five-point and nine-point methods (assuming Dirichlet boundary conditions) is presented for the solution of equations of the form

$$\nabla^2\phi + S\phi = q, \tag{3.1}$$

where ϕ, q, and S are complex. This particular equation is studied in detail since for all the test problems described in Section 4, the subsidiary elliptic equation takes the form (3.1), where S is a function of s only.

For Poisson's equation with simple boundary conditions, Pickering [7] demonstrated that the extra accuracy, obtained using the nine-point [15] rather than the five-point method, is achieved with relatively little extra computational effort. (For grids of 15×15, 31×31, and 63×63 points the nine-point method was found to take approximately 28% more computer time than the five-point method). Moreover since Eq. (3.1) is only slightly more complicated than Poisson's equation it was to be expected that the nine-point method would show similar advantages over the five-point method for the present problem.

The Fourier transform was employed in the x direction and the region was overlayed with a square grid of side h. [The five-point method may also be employed on a rectangular grid.] Furthermore, $\boldsymbol{\phi}_j$ denotes the vector of internal nodal ϕ values along a grid row parallel to the x axis (so that the ith component of $\boldsymbol{\phi}_j$ is ϕ_{ij} ; $i = 1, 2,..., n$).

For both methods the complete set of finite difference equations derived from Eq. (3.1) may be written in the form

$$A\boldsymbol{\phi}_j + B(\boldsymbol{\phi}_{j+1} + \boldsymbol{\phi}_{j-1}) = \mathbf{Q}_j \qquad (j = 1, 2,..., n)$$

where A and B are nth-order symmetric matrices.

For the five-point method

$$A = A_0 + Sh^2 I_n ,$$

where

$$A_0 = \begin{bmatrix} -4 & 1 & 0 & 0 & \cdots & 0 \\ 1 & -4 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -4 & 1 & 0 & 0 \\ & & & \cdots & & \\ 0 & \cdots & 0 & 1 & -4 & 1 \\ 0 & & \cdots & 0 & 1 & -4 \end{bmatrix},$$

$$B = I_n ,$$

and

$$\mathbf{Q}_j = \begin{bmatrix} h^2 q_{1,j} - \phi_{0,j} \\ h^2 q_{2,j} \\ \vdots \\ h^2 q_{n-1,j} \\ h^2 q_{n,j} - \phi_{n+1,j} \end{bmatrix}.$$

The eigenvalues, λ_k, and corresponding eigenvectors, $\mathbf{x}_k$, of A_0 are given by

$$\left.\begin{aligned} \lambda_k &= -4 + 2\cos\theta_k \\ \text{and} \quad \mathbf{x}_k &= [\sin\theta_k, \sin 2\theta_k, \ldots, \sin n\theta_k]^T, \end{aligned}\right\} \quad (k = 1, 2, \ldots, n), \tag{3.2}$$

respectively, where

$$\theta_k = k\pi/(n+1).$$

For the nine-point method,

$$A = A_1 + \tfrac{1}{2}Sh^2(12 - Sh^2)\, I_n,$$

where

$$A_1 = 4\begin{bmatrix} -5 & 1 & \cdot & \cdot & \cdot & \cdot & 0 \\ 1 & -5 & 1 & 0 & \cdot & \cdot & 0 \\ 0 & 1 & -5 & 1 & 0 & \cdot\ \cdot & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdot & \cdot & 0 & 1 & -5 & 1 \\ 0 & \cdot & \cdot & \cdot & 0 & 1 & -5 \end{bmatrix},$$

$$B = \tfrac{1}{4}A_1 + 9I_n,$$

and

$$\mathbf{Q}_j = \begin{bmatrix} \tfrac{1}{2}h^2(12 - Sh^2)\, q_{1,j} + \tfrac{1}{2}h^4\nabla^2 q_{1,j} - 4\phi_{0,j} - \phi_{0,j+1} - \phi_{0,j-1} \\ \tfrac{1}{2}h^2(12 - Sh^2)\, q_{2,j} + \tfrac{1}{2}h^4\nabla^2 q_{2,j} \\ \vdots \\ \tfrac{1}{2}h^2(12 - Sh^2)\, q_{n-1,j} + \tfrac{1}{2}h^4\nabla^2 q_{n-1,j} \\ \tfrac{1}{2}h^2(12 - Sh^2)\, q_{n,j} + \tfrac{1}{2}h^4\nabla^2 q_{n,j} - 4\phi_{n+1,j} - \phi_{n+1,j+1} - \phi_{n+1,j-1} \end{bmatrix}.$$

The eigenvalues, μ_k, of A_1 are given by

$$\mu_k = -20 + 8\cos\theta_k \qquad (k = 1, 2, \ldots, n)$$

and the corresponding eigenvectors are given by Eq. (3.2).

Thus, for both methods, a suitable orthogonal basis for the expansion of each $\boldsymbol{\phi}_j$ is provided by Eq. (3.2) and fast Fourier transforms may be used to obtain a solution of Eq. (3.1). The odd/even reduction process described in [4, 5] was used in both methods.

3.2. *Numerical Inversion of the Laplace Transform*

Several approaches to the numerical inversion of the Laplace transform have been proposed (see, in particular, [8–14, 16, 17]. The algorithm proposed by Vlach *et al.* [8–11], based upon the Padé approximation to e^z, is the one we shall adopt.

The inversion formula for the Laplace transform is

$$v(t) = (2\pi i)^{-1} \int_{c-i\infty}^{c+i\infty} V(s)\, e^{st}\, ds$$

where c is a real constant greater than the real part of any pole of $V(s)$. Making the substitution $z = st$, this becomes

$$v(t) = (2\pi it)^{-1} \int_{c'-i\infty}^{c'+i\infty} V(z/t)\, e^z\, dz.$$

In the numerical inversion method described by Vlach [8], e^z is replaced by the Padé approximation

$$e^z = \xi_{N,M}(z) = P_N(z)/Q_M(z)$$

where $P_N(z)$ and $Q_M(z)$ are polynomials of degree N and M, respectively. The poles of $\xi_{N,M}(z)$ may be shown [8] to be all simple poles and so a partial fraction expansion gives

$$\xi_{N,M}(z) = \sum_{j=1}^{M} K_j/(z - z_j),$$

where the poles and residues of $\xi_{N,M}(z)$ are, respectively, z_j and K_j. Hence, replacing e^z by $\xi_{N,M}(z)$ and completing the contour in the right-hand plane by a semicircular arc at infinity, the numerical approximation to the integral becomes

$$\hat{v}(t) = -\frac{1}{t} \sum_{j=1}^{M} K_j V\left(\frac{z_j}{t}\right), \tag{3.3}$$

provided that the function $V(z_j/t)\, \xi_{N,M}(z_j)$ has at least two more finite poles than zeros [10], ensuring that the contribution from the semicircular arc is zero. Note that for a fixed pair M, N, the constants K_j and z_j may be regarded as known parameters. A suitable choice for most problems is $N = M - 2$.

If M is even, all the roots of $Q_M(z)$ occur as complex conjugate pairs and Eq. (3.3) may then be modified to give

$$\begin{aligned} \hat{v}(t) &= -\frac{1}{t} \sum_{j=1}^{M'} K_j V\left(\frac{z_j}{t}\right) - \frac{1}{t} \sum_{j=1}^{M'} \bar{K}_j V\left(\frac{\bar{z}_j}{t}\right) \\ &= -\frac{2}{t} \sum_{j=1}^{M'} \mathscr{R}\left[K_j V\left(\frac{z_j}{t}\right)\right], \qquad t > 0 \end{aligned} \tag{3.4}$$

where the bar denotes the complex conjugate and $M' = M/2$. The values of K_j and z_j are tabulated in [11] for various values of M and N but they can be easily calculated by a Newton–Raphson procedure.

Singhal and Vlach have proved [10] that Eq. (3.4) inverts exactly the first $M + N + 1$ terms of the Taylor series of any function $v(t)$. For small values of t, smaller values of M tend to give the more accurate results, since with increasing M the residues K_j grow rapidly in magnitude and computer round-off increases the error. The accuracy of the method for a number of test functions is discussed in [10].

Formulas for evaluating time derivatives of the solution can be obtained from the standard expression

$$\mathscr{L}[\partial v/\partial t] = sV(s) - v(0).$$

The terms involving initial conditions do not contribute to the inversion and the expression for the first derivative is

$$\frac{\partial v}{\partial t}(t) = -\frac{1}{t}\sum_{i=1}^{M}\frac{z_j}{t}K_jV\left(\frac{z_j}{t}\right), \qquad t > 0. \tag{3.5}$$

Similar expressions may be written down for higher order derivatives.

Throughout the calculations M was taken to be even, so that Eq. (3.4) was used. The choice $N = M - 2$ was satisfactory in the evaluation of $\partial u/\partial t$ in problem 2, but could not be used to evaluate $\partial^2 u/\partial t^2$. For the latter we would have required $N = M - 4$ if we had wished to use Eq. (3.4), or $N = M - 3$ using Eq. (3.3).

4. Numerical Results

4.1. *Test Problems*

As was stated in Section 2, a broad class of equations can be treated by this method. We shall demonstrate the merits of the method by reference for the most part to relatively straightforward model problems with just one example from the more general class.

Problem 1. The wave equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{\partial^2 u}{\partial t^2}$$

was solved on the square region $0 \leqslant x \leqslant 1, 0 \leqslant y \leqslant 1$, subject to the condition $u = 0$ on the boundaries and with initial conditions

$$u(x, y, 0) = \sin \pi x \sin \pi y, \qquad \frac{\partial u}{\partial t}(x, y, 0) = 0.$$

This has analytical solution $u(x, y, t) = \cos(2^{1/2}\pi t)\sin(\pi x)\sin(\pi y)$. The equation to be solved by FFT is

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} - s^2 U = -s\sin(\pi x)\sin(\pi y),$$

which has the form of (3.1).

Problem 2. The heat conduction equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{\partial u}{\partial t}$$

was solved for $u(x, y, t)$ and $\partial u(x, y, t)/\partial t$ on the same region, with $u = 0$ on the boundaries and initial condition

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

The analytic solution is $u(x, y, t) = \exp(-2\pi^2 t) \sin(\pi x) \sin(\pi y)$. The equation to be solved by FFT is

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} - sU = -\sin(\pi x) \sin(\pi y).$$

A comparison of the formulations for these two problems illustrates an important feature of the method. The effect of changing from the hyperbolic wave equation problem to the parabolic heat conduction equation is merely to alter coefficients in the subsidiary equation. The extra programming is consequently negligible.

For both problems the numerical solutions were obtained over a spatial grid containing 15×15 internal mesh points. Both the five-point and the nine-point FFT algorithms were tested. The parameter M', controlling the accuracy of the numerical Laplace inversion algorithm, was varied and given values $M' = 2, 5$, or 8. For Problem 2, as another comparison, results for both u and $\partial u/\partial t$ were also determined by the Peaceman–Rachford ADI method.

A suitable test problem with a time-varying source function and boundary conditions was taken from Gourlay and McGuire [18] and is reproduced below.

Problem 3. The equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + g(x, y, t)$$

was solved over the square $0 \leqslant x \leqslant 2$, $0 \leqslant y \leqslant 2$ with

$$g(x, y, t) = \sin x \exp(-t)/(1 + y)^2 - 2x - 6xy$$

and initial and boundary conditions appropriate to the solution

$$u(x, y, t) = \sin x \ln(1 + y) \exp(-t) + x^3 y + xy^2.$$

This problem was solved using the FFT–LT algorithm for various grids and values of M'. For comparison, results were also obtained using the Peaceman–Rachford ADI method as formulated in [18]. (The computations in [18] were for a 19×19 grid which is unsuitable for the usual FFT method.)

In Section 2 it was stated that the type of equation which can be treated is one which may involve quite elaborate operator expressions in $\partial/\partial t$. The three problems formulated so far do not really demonstrate the power of the algorithm in its ability to deal easily with such terms. We therefore consider the following equation which is more indicative of the class of problems which may be handled.

Problem 4. On the domain $0 \leqslant x \leqslant 1$, $0 \leqslant y \leqslant 1$, the equation

$$\frac{\partial^3 u}{\partial t\,\partial x^2} + \frac{\partial^3 u}{\partial t\,\partial y^2} + \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - \frac{\partial^2 u}{\partial t^2} - 3\frac{\partial u}{\partial t} - 2u = -e^{-t}$$

or

$$(1 + D)\frac{\partial^2 u}{\partial x^2} + (1 + D)\frac{\partial^2 u}{\partial y^2} - (D^2 + 3D + 2)\,u = -e^{-t}$$

has the solution

$$u = (x^2 + y^2 + t)\,e^{-t}$$

if the initial conditions are

$$u = x^2 + y^2, \qquad \frac{\partial u}{\partial t} = 1 - x^2 - y^2, \qquad \frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 u}{\partial y^2} = 2$$

and the boundary conditions are

$$u(0, y, t) = (y^2 + t)\,e^{-t}, \qquad u(x, 0, t) = (x^2 + t)\,e^{-t},$$
$$u(1, y, t) = (1 + y^2 + t)\,e^{-t}, \qquad u(x, 1, t) = (1 + x^2 + t)\,e^{-t}.$$

The subsidiary equation to be solved by FFT is

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} - \left(\frac{s^2 + 3s + 2}{s + 1}\right) U = -\frac{1}{(s + 1)^2} - \frac{1}{(s + 1)}\,[(x^2 + y^2)(s + 2) - 3].$$

4.2. *Results*

Results for the wave equation (Problem 1) are given in Table I. The five-point formula for the FFT gives results which may be seen to be very accurate for $0 < t \leqslant 0.1$ and reasonably accurate for $0.1 < t \leqslant 0.5$. The use of the larger value of M' ($M' = 5$) gives no improvement in accuracy for small values of t but it does allow computation over a greater time range than does $M' = 2$. Results obtained using $M' = 8$ were in general slightly inferior to those quoted for the smaller values of M' since computer round-off increases the error.

Since, for $0 < t \leqslant 0.5$, changing M' produces negligible changes in the accuracy, it may be inferred that for this time range, the errors are predominantly associated with the spatial grid and the FFT algorithm chosen. That this is so may be seen by comparing the five-point with the nine-point results over the same 15×15 grid and

TABLE I

Solution $u(x, y, t)$ for the Wave Equation (Problem 1) at the Midpoint for Various t Values[a]

t	Analytic solution	Five-point FFT $M' = 2$	Five-point FFT $M' = 5$	Nine-point FFT $M' = 5$
0.01	0.999013	3.2×10^{-6}	3.3×10^{-6}	1.6×10^{-7}
0.02	0.996055	1.3×10^{-5}	1.3×10^{-5}	1.0×10^{-8}
0.05	0.975427	7.9×10^{-5}	7.8×10^{-5}	2.6×10^{-7}
0.1	0.902917	3.1×10^{-4}	3.1×10^{-4}	2.4×10^{-6}
0.2	0.603517	1.1×10^{-3}	1.1×10^{-3}	8.6×10^{-6}
0.5	−0.605700	2.1×10^{-3}	2.8×10^{-3}	2.2×10^{-5}
1.0	−0.266255	6.1×10^{-2}	6.9×10^{-3}	5.3×10^{-5}
2.0	−0.858216	—	7.4×10^{-3}	2.8×10^{-5}

[a] 15×15 grid; absolute errors quoted.

using the same value $M' = 5$. The results from the nine-point formula are seen to be excellent over the large time span $0 < t < 2$.

Results for the heat conduction equation (Problem 2) are given in Tables II and III. The results for the five-point FFT are seen to be accurate for the time range $0 < t \leqslant 0.1$ using $M' = 2$ and for $0 < t \leqslant 0.2$ using $M' = 5$. The results using the nine-point formula are again seen to show the same order of improvement observed for the wave equation.

Table II also contains results from the Peaceman–Rachford ADI method applied directly to the PDE (without Laplace transformation) using a time step of 0.001 and

TABLE II

Solution $u(x, y, t)$ for the Heat Conduction Equation (Problem 2) at the Midpoint for Various t Values[a]

t	Analytic solution	Five-point FFT $M' = 2$	Five-point FFT $M' = 5$	Nine-point FFT $M' = 5$	P–R ADI $\delta t = 0.001$
0.005	0.906018	2.9×10^{-4}	2.9×10^{-4}	—	2.8×10^{-4}
0.01	0.820869	5.2×10^{-4}	5.2×10^{-4}	4.0×10^{-6}	5.2×10^{-4}
0.02	0.673825	8.5×10^{-4}	8.5×10^{-4}	6.6×10^{-6}	8.5×10^{-4}
0.05	0.372708	1.2×10^{-3}	1.2×10^{-3}	9.2×10^{-6}	1.2×10^{-3}
0.1	0.138911	1.0×10^{-3}	8.8×10^{-4}	6.9×10^{-6}	8.7×10^{-4}
0.2	1.929×10^{-2}	2.4×10^{-3}	2.5×10^{-4}	1.8×10^{-6}	2.4×10^{-4}
0.5	5.172×10^{-5}	1.1×10^{-2}	1.8×10^{-6}	5.3×10^{-8}	1.7×10^{-6}
1.0	2.67×10^{-10}	9.6×10^{-3}	3.2×10^{-5}	3.3×10^{-5}	1.7×10^{-10}

[a] 15×15 grid; absolute errors quoted.

TABLE III

Values of $\partial u(x, y, t)/\partial t$ for the Heat Conduction Equation (Problem 2) at the Midpoint for Various t Values[a]

t	Analytic solution	Five-point FFT $M' = 5$	P–R ADI $\delta t = 0.001$
0.005	-17.8841	5.2×10^{-2}	5.0×10^{-2}
0.01	-16.2023	4.2×10^{-2}	4.0×10^{-2}
0.02	-13.3008	2.6×10^{-2}	2.5×10^{-2}
0.05	-7.35696	3.2×10^{-4}	1.3×10^{-4}
0.1	-2.74200	8.6×10^{-3}	8.7×10^{-3}
0.2	-0.380894	3.6×10^{-3}	3.6×10^{-3}
0.5	-1.1021×10^{-3}	3.4×10^{-5}	2.9×10^{-5}
1.0	-5.28×10^{-9}	6.3×10^{-4}	3.2×10^{-9}

[a] 15×15 grid; absolute errors quoted.

the same 15×15 spatial grid. The results are seen to be virtually identical with the five-point FFT results with $M' = 5$. This confirms that the errors are almost entirely due to the spatial discretization, the numerical Laplace transform giving a very small contribution to the error.

One of the merits of the Laplace transform approach is that it is a simple matter to produce values for time derivatives of the solution by means of Eq. (3.5). The solution for $\partial u/\partial t$ for this problem is shown in Table III and is compared with results from the ADI method. Again agreement between the two methods is excellent.

The two problems considered so far both had simple boundary conditions. The algorithm we are proposing very readily handles boundary conditions which may or may not be functions of the appropriate spatial variables but which *are* functions of time as in Problem 3. Upon taking the Laplace transform they become functions of the algebraic Laplace transform variable s and so the boundary conditions are treated in precisely the same way as time-invariant boundary conditions with no extra programming or computational effort.

The results for Problem 3 are presented in detail since this is a good example to illustrate the main sources of error. Figure 1 shows graphs of absolute error at the midpoint of the mesh as a function of time and for three mesh sizes. (The parameters are $M' = 5$ in the five-point FFT–LT algorithm and $\delta t = 0.01$ in the ADI algorithm.) The errors from the two algorithms are seen to be very similar over the time interval $0 < t \leqslant 2$. The predominant source of error is again due to spatial discretization, errors for small t being reduced, as expected, by a factor of 4 when the spatial mesh size is halved.

Further results are given in Fig. 2 which shows the errors for the transform algorithm only but over an extended time range and for two values of M'. The two values of M' are seen to give consistent results for small t. With $M' = 2$ the errors due to the Laplace transform part of the algorithm become predominant for times greater

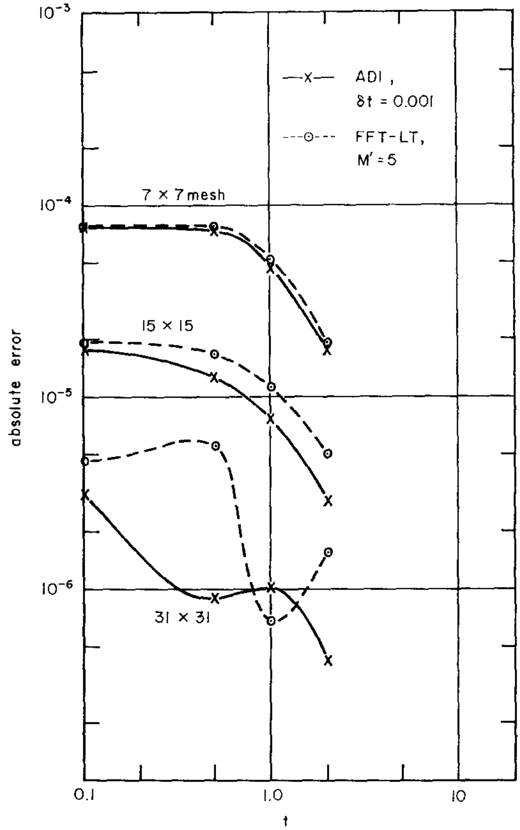

FIG. 1. Comparison between midpoint errors given by FFT-LT and ADI algorithms. (Problem 3, three mesh sizes).

TABLE IV

Solution $u(x, y, t)$ for Problem 4 at the Midpoint for Various t Values[a]

t	Analytic solution	Five-point FFT $M' = 2$	Five-point FFT $M' = 5$
0.1	0.542902	1.9×10^{-10}	7.5×10^{-8}
0.2	0.573111	8.2×10^{-10}	1.8×10^{-7}
0.5	0.606530	3.2×10^{-7}	1.2×10^{-7}
1.0	0.551820	1.0×10^{-5}	1.3×10^{-7}
2.0	0.338338	6.0×10^{-4}	1.8×10^{-9}
5.0	0.037059	8.0×10^{-3}	8.4×10^{-8}
10.0	0.000476	3.1×10^{-4}	1.1×10^{-6}

[a] 15 × 15 grid; absolute errors quoted.

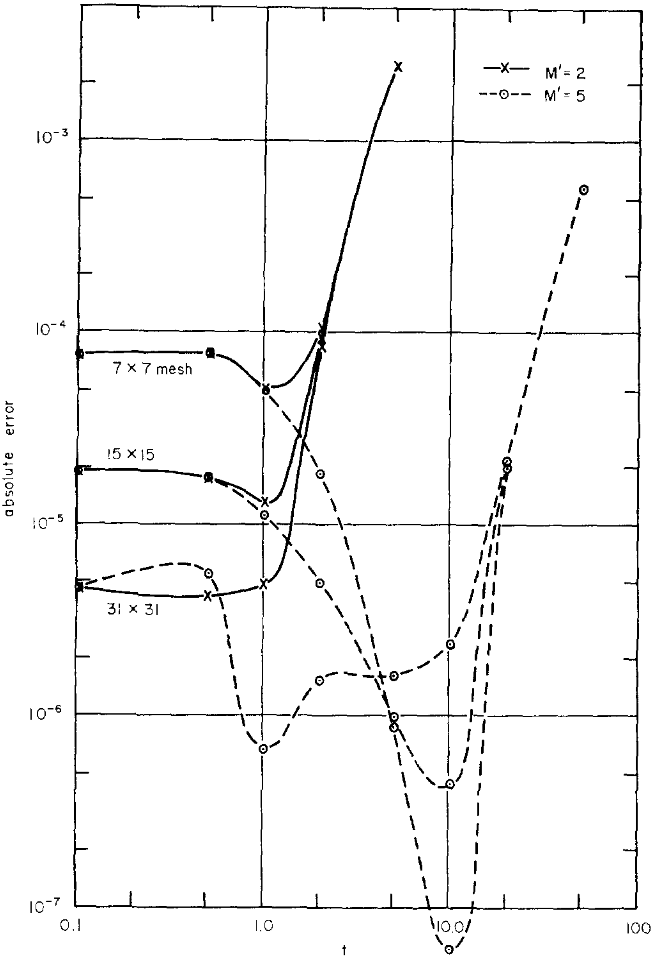

FIG. 2. Effect of M' and mesh size on midpoint errors given by FFT–LT algorithm. (Problem 3).

than $t = 2$. With $M' = 5$, because effectively the number of terms in the Taylor series approximation is increased, a high degree of accuracy is maintained until $t = 20$. Results obtained with $M' = 8$ are generally inferior because of excessive round-off error unless double-precision computer arithmetic is used.

Table IV contains typical numerical results for Problem 4. This example was chosen to illustrate the complexity of equation and boundary conditions which may be handled by the method. Very satisfactory results are abtained, particularly with the larger value of M'.

4.3. *Computation Time*

It is difficult to present a definitive statement on computation time because of the wide variety of problems which can be treated. The following considerations would apply to any problem. Assuming a value for M' (the parameter in the Laplace transform inversion algorithm effectively governing the number of terms in the Taylor expansion) for a given time point the subsidiary elliptic equation has to be solved M' times in complex arithmetic for the complex function $U(x, y, s)$. The method is basically a single-step method so that the computations need only be performed for the specific time points for which the solution is required.

By contrast, when using finite difference methods, the problem and the method of solution will dictate how many small time increments are needed to achieve satisfactory accuracy. Comparisons are therefore difficult to make and probably the only general statement that can be made is that problems requiring a small δt in the finite difference representation will also require a large M'.

Our experience with Problem 3 with its rather involved boundary conditions is of interest as a guide to the potential that the method possesses. Table V gives the computation times on an ICL 1906S computer using Fortran. For each of the mesh sizes considered, the computation time per time increment is given for the Peaceman–Rachford ADI method and the computation time per unit M' is given for the transform method. The total computation time may then be assessed as follows using the results for the 15×15 grid as an illustration. In order to evaluate the solution at $t = 1$, the ADI method (with $\delta t = 0.01$) would require 14.5 sec, whereas the transform method (with $M' = 2$) would require 1.04 sec. The respective absolute errors at the midpoint are 7.47×10^{-6} and 1.31×10^{-5}.

TABLE V

Typical Computation Times for Problem 3 on an ICL 1906S Computer

Grid size	ADI (time per δt) (sec)	FFT–LT (time per unit M') (sec)	Time to $t = 1$ (sec)	
			ADI ($\delta t = 0.01$)	FFT–LT ($M' = 2$)
7×7	0.035	0.13	3.5	0.26
15×15	0.145	0.52	14.5	1.04
31×31	0.585	2.2	58.5	4.4

This demonstrates that the method can be very competitive with regard to computation time. However, it must be mentioned that for a problem, such as Problem 2, in which the boundary conditions do not vary with time and so do not have to be evaluated at each time step the ADI methods become relatively more effective.

To summarize, the transform method would generally be best if a given problem contained any of the following features: if it was other than the basic heat conduction equation, if the boundary conditions and source functions were time varying, or if the solution had to be calculated over an extended time range on a coarse time grid.

5. Conclusions

In this paper we have put forward a method for the numerical solution of a large class of time-dependent linear partial differential equations defined on a square or rectangular spatial domain. The method combines a numerical Laplace transform inversion algorithm with either a five- or nine-point fast Fourier transform method. Numerical results presented for four test problems justify the following conclusions.

With regard to accuracy, the main errors arise from the spatial discretization. Thus the five-point formula gives results which are very similar to those obtained using established ADI methods and the same mesh. The nine-point formula gives results which are significantly better, as would be expected.

The main advantages of the method are:

(1) It is as accurate as the Peaceman–Rachford ADI method on the same spatial grid.

(2) It is very easy to program once the fast Fourier transform algorithm has been programmed.

(3) The program is very flexible and only trivial changes are required to adapt it from solving, say, the hyperbolic equation of Problem 1 to the parabolic equation of Problem 2.

(4) The class of problems with time-varying boundary conditions and source functions is no more difficult to handle than time-invariant functions. For this class the method can give the solution at specified time points appreciably faster than the Peaceman–Rachford ADI method.

(5) Derivatives of the solution with respect to time are easily and accurately evaluated (though this may involve some adjustment of the parameters M', N in the Laplace transform inversion formula if higher order derivatives are required).

(6) The method has the single-step property. This allows the choice of using a larger value of M' and evaluating the solution directly at a specified time point. Alternatively with a smaller value of M', the solution may be evaluated at intermediate time points and the algorithm restarted at each of these points using the current solution (and its derivatives if necessary) as the initial condition(s). Since one is usually seeking a time history of the solution this latter course is probably the best.

The single-step property is also possessed by methods based on approximations to the matrix exponential function. For example, the Chebyshev rational method described by Cavendish *et al.* [19] for certain linear parabolic equations, can deal with problems with boundary conditions and source terms which are piecewise-constant in time. Our method has the advantage that it can deal with a wide class of problems with time-dependent boundary conditions and source terms, but the spatial dependence of the coefficients must be restricted to the forms indication in Eq. (2.1) if FFT methods are to be used to solve the subsidiary elliptic equation.

REFERENCES

1. R. BELLMAN, R. E. KALABA, AND J. A. LOCKETT, "Numerical Inversion of the Laplace Transform," Elsevier, New York, 1966.
2. E. ANGEL AND R. BELLMAN, "Dynamic Programming and Partial Differential Equations," Academic Press, New York/London, 1972.
3. A. S. I. ZINOBER AND E. HUNTLEY, The numerical solution of linear time-dependent partial differential equations by the Laplace transform and finite difference approximations, in preparation.
4. R. W. HOCKNEY, *J. Assoc. Comput. Mach.* **12** (1965), 95.
5. R. W. HOCKNEY, "Methods in Computational Physics," Vol. 9, pp. 136–210, Academic Press, New York/London, 1970.
6. R. C. LE BAIL, *J. Comp. Phys.* **9** (1972), 440.
7. W. M. PICKERING, *J. Inst. Maths. Appl.* **19** (1977), 337.
8. J. VLACH, *J. Franklin Inst.* **288** (1969), 99.
9. K. SINGHAL AND J. VLACH, *Elec. Lett.* **7** (1971), 413.
10. K. SINGHAL AND J. VLACH, *J. Franklin Inst.* **299** (1975), 109.
11. K. SINGHAL, J. VLACH, AND M. NAKHLA, *Arch. Elek. Übertr.* **30** (1976), 157.
12. V. ZAKIAN, *Elec. Lett.* **5** (1969), 120.
13. V. ZAKIAN, *J. Math. Anal. Appl.* **50** (1975), 191.
14. A. TALBOT, "The Accurate Numerical Inversion of Laplace Transforms," Brunel Univ., Dept. of Mathematics TR/61, 1976.
15. W. G. BICKLEY, *Quart. J. Mech.* **1** (1948), 35.
16. C. J. WELLEKENS, *Elec. Lett.* **6** (1970), 742.
17. R. PIESSENS, *J. Inst. Maths. Appl.* **10** (1972), 185.
18. A. R. GOURLAY AND G. R. MCGUIRE, *J. Inst. Maths. Appl.* **7** (1971), 216.
19. J. C. CAVENDISH, W. E. CULHAM, AND R. S. VARGA, *J. Comp. Phys.* **10** (1972), 354.

581/27/2-8