SIAM J. SCI. STAT. COMPUT.
Vol. 9, No. 4, July 1988

005

GLOBAL ERROR ESTIMATION IN THE METHOD OF LINES FOR PARABOLIC EQUATIONS*

M. BERZINS†

Abstract. A method is described for obtaining an indication of the error in the numerical solution of parabolic partial differential equations using the method of lines. The error indicator is derived by using a combination of existing global error estimating algorithms for initial value problems in ordinary differential equations with estimates for the PDE truncation error. An implementation of the algorithm is described and numerical examples are used to illustrate the reliability of the error estimates that are obtained.

Key words. parabolic equations, method of lines, global error estimates

AMS (MOS) subject classifications. 65M20, 65M15

1. Introduction. The method of lines has proved to be an important general purpose technique for the integration of time-dependent parabolic and parabolic-elliptic partial differential equations. One of the desirable features of a general purpose method of lines algorithm for time-dependent PDE's is that both the method of spatial discretisation and the positioning of the spatial discretisation points should be chosen so as to model accurately the properties of the exact solution to the PDE. In other words, both the spatial mesh points and the discretisation method must be chosen to control the spatial errors as far as this is possible. In solving the ODE initial value problem defined by the discretisation method and the choice of mesh points, it is desirable that the ODE time integrator should control the local error, subject to the user's tolerance, and should also provide a means of estimating the global error incurred in the integration. Finally, in order to provide the user with information about the reliability of the numerical solution, estimates of the error in the space and time dimensions should be combined so as to provide an overall estimate of the error in the computed solution at any stage of the integration.

There has been much interest in the development of adaptive spatial mesh methods for parabolic PDE's. One of two approaches is usually adopted; the mesh is either refined continuously with the computed solution [12], [1] or mesh refinement is only performed at discrete time levels [2], [4]. An alternate approach is to combine the discrete and continuous approaches [13], [5]. All of these methods seek to place the mesh points to follow the changing nature of the solution. In addition, some methods seek to equidistribute the spatial discretisation error [1], [2].

As the time integration in the method of lines is most commonly performed by using the backward differentiation formulae it is possible to use the ODE global error estimators developed for this method (see Shampine [14]) to estimate the time integration error. The only restriction on the choice of global error estimator is that the estimator must be applicable to systems of differential algebraic equations, such as those arising from the spatial discretisation of the parabolic-elliptic PDE's.

At present there have been few attempts to combine the estimates of the spatial and temporal errors to produce an accurate error indicator. One example of a package which does attempt to estimate these errors is the software of Schönauer et al. [15]. The importance of such an indicator is that it allows the reliability of the numerical solution for both fixed and adaptive mesh methods to be evaluated. In addition, the

* Received by the editors January 12, 1987; accepted for publication (in revised form) August 5, 1987.
† Department of Computer Studies, University of Leeds, Leeds LS2 9JT, United Kingdom.

estimates of the error can be used in adaptive mesh schemes as a basis for mesh modification. It is the aim of this paper to demonstrate a simple algorithm that allows the overall error in the computed solution to be estimated. In order to make this algorithm as clear as possible the paper will consider a restricted class of parabolic equations and use a spatial discretisation method that is readily analysed. This paper is based on the earlier work of Berzins and Dew [3] but overcomes two limitations of the earlier work in that the error estimating algorithm is applicable to second-order finite difference discretisations and that a more accurate time integration method is used to monitor the evolution of the error.

The paper has nine sections. Section 2 is concerned with the class of parabolic equations that will be considered while § 3 describes the discretisation method that is used to spatially discretise the PDE's. Section 4 is concerned with the time integration and with a global error indicator for ODE initial value problems. This indicator is combined in § 5 with an estimate of the PDE truncation error to estimate the overall error in the computed solution. Sections 6 and 7 provide details of the estimate for the spatial truncation error and of the experiments to illustrate the capability of the error indicator, while §§ 8 and 9 consider the potential uses of and extensions to the error estimates.

2. Problem class. The problem class considered here is sufficiently general to illustrate the algorithm for error estimation. The algorithm extends naturally to systems of partial differential equations and to equations in more than one space dimension, provided that the same method of lines approach is employed.

For notational convenience, the class of parabolic PDE's to be considered will be written as

$$(2.1)\quad c(x,t)\frac{\partial u}{\partial t}=\frac{\partial}{\partial x}r\left(x,t,u,\frac{\partial u}{\partial x}\right)+f\left(x,t,u,\frac{\partial u}{\partial x}\right),\qquad (x,t)\in\Omega=[a,b]\times(0,t_e]$$

where we assume that there exist constants c_1 and c_2 such that

$$(2.2)\qquad 0<c_1<c(x,t)<c_2\quad \forall(x,t)\in\Omega.$$

The boundary conditions are taken to be of the form

$$(2.3)\qquad \beta(a,t)r\left(a,t,u(a,t),\frac{\partial u}{\partial x}\right)=g_a(t,u(a,t))$$

and

$$(2.4)\qquad \beta(b,t)r\left(b,t,u(b,t),\frac{\partial u}{\partial x}\right)=g_b(t,u(b,t))$$

for $t\in(0,t_e]$. The initial condition has the form

$$(2.5)\qquad u(x,0)=k(x),\qquad x\in[a,b].$$

We assume that the PDE defined by the above equation is well posed and has a unique continuous solution $u(x,t)$, for all $(x,t)\in\Omega$. The spatial mesh is defined by

$$(2.6)\qquad \delta:\quad a=x_1<x_2<\cdots<x_N=b.$$

This mesh partitions the interval $[a,b]$ into $N-1$ subintervals, of length h_j, where

$$(2.7)\qquad h_j=x_{j+1}-x_j,\qquad j=1,2,\cdots,N-1.$$

3. Spatial discretisation method. For many of the parabolic equations that arise in practice second-order finite difference methods are very popular. Recently Skeel [16] derived a modified form of the box scheme [11] that is particularly convenient as it spatially discretises the PDE of § 2 into an ODE system in normal form. The discretisation method can be written as

$$(3.1)\quad (h_{j-1}c_{j-1/2}+h_jc_{j+1/2})\frac{\partial U}{\partial t}(x_j,t)=2(R_{j+1/2}-R_{j-1/2})+(h_{j-1}f_{j-1/2}+h_jf_{j+1/2})$$

where $R_{j+1/2}$ and $R_{j-1/2}$ are defined by

$$R_{j+1/2}=r\left(\frac{x_j+x_{j+1}}{2},t,\frac{U(x_j,t)+U(x_{j+1},t)}{2},\frac{U(x_{j+1},t)-U(x_j,t)}{h_j}\right),$$

$$R_{j-1/2}=r\left(\frac{x_j+x_{j-1}}{2},t,\frac{U(x_j,t)+U(x_{j-1},t)}{2},\frac{U(x_j,t)-U(x_{j-1},t)}{h_{j-1}}\right),$$

and $j=2,\cdots,N-1$. The quantities $c_{j+1/2}$, $c_{j-1/2}$, $f_{j+1/2}$ and $f_{j-1/2}$ are defined similarly and $U(x_j,t)$ is the approximate solution defined by the spatial discretisation method at the point x_j. The boundary condition at $x=a$ is implemented as

$$(3.2)\quad \beta(a,t)(c_{3/2})\frac{\partial U}{\partial t}(x_1,t)=2(\beta(a,t)R_{3/2}-g_a(t,U(x_1,t)))/h_1+\beta(a,t)f_{3/2}$$

and the condition at $x=b$ is treated similarly. It should be noted that the functions $\beta(a,t)$ and $\beta(b,t)$ may be zero and so the boundary condition at, say, $x=b$ may reduce to the algebraic equation

$$(3.3)\qquad g_b(t,U(x_N,t))=0.$$

The initial condition is defined by evaluating the function $k(x)$ at the spatial mesh points

$$(3.4)\qquad U(x_i,0)=k(x_i),\qquad i=1,\cdots,N.$$

4. Integration in time. The system of differential algebraic equations in time defined by the spatial discretisation method (e.g., (3.1), (3.2), and (3.3)) can equivalently be written as

$$(4.1)\qquad A_N(t)\dot{\underline{U}}=\underline{F}_N(t,\underline{U}(t))$$

where the N-dimensional vector is defined by

$$\underline{U}(t)=\begin{bmatrix}U(x_1,t)\\U(x_2,t)\\\vdots\\U(x_N,t)\end{bmatrix}$$

where $A_N(t)$ is an N by N matrix with nonzeros only on its leading diagonal defined by (see (3.1))

$$(4.2)\quad [A_N(t)]_{i,j}=\delta_{i,j}(h_{j-1}c_{j-1/2}+h_jc_{j+1/2}),\qquad i=2,\cdots,N-1,\quad j=2,\cdots,N-1.$$

The first diagonal element is defined by the particular boundary condition (3.2) as

$$(4.3)\qquad [A_N(t)]_{1,1}=c_{3/2}\beta(a,t)$$

and the bottom diagonal element by the algebraic equation (3.3) as

$$[A_N(t)]_{N,N}=0.$$

The vector $\underline{F}_N(t, \underline{U}(t))$ is then defined by the right-hand side of (3.2) followed by the $N-2$ right-hand side of (3.1) followed by the left-hand side of (3.3). The initial condition for $\underline{U}(t)$ is given by (3.4).

4.1. Time integration: global error estimate. In practice (4.1) is a stiff or mildly stiff system of ordinary differential equations in time that is usually solved by software based on the backward differentiation formulae, e.g., [9], [4]. The ODE global error is defined by

$$\underline{ge}(t) = \underline{U}(t) - \underline{V}(t), \qquad t \in (0, t_e] \tag{4.4}$$

where $\underline{V}(t)$ is the approximation to $\underline{U}(t)$ that is computed by the time integration method at discrete times $t_1, t_2, \cdots, t_{\text{end}}$ using local error control and may be calculated for other values of t by using a suitable interpolating routine provided by the integrator.

Shampine [14] describes a number of methods for estimating the global error in ODE initial value problems. The algorithm best suited to estimating the overall error is an extension of one of Shampine's algorithms that is used by Chua and Dew [6] to estimate the global error in integrating differential-algebraic equations using the theta method. Suppose that an estimate of the ODE global error at time t_n has already been obtained $\underline{ge}(t_n)$ and that the integrator takes a step of size h to time t_{n+1} by using local error control and without using local extrapolation. The solution of the variational equation

$$A_N(t)\,\dot{\underline{W}} = J\underline{W}, \qquad \underline{W}(t_n) = \underline{ge}(t_n) \tag{4.5}$$

(where $J = (\partial \underline{F}_N / \partial \underline{U})$) is shown [14] to be related to the global error at the end of the step by

$$\underline{ge}(t_{n+1}) = \underline{W}(t_{n+1}) + \underline{le}_{n+1}(t_{n+1}) + O(\underline{ge}^2(t_{n+1})) \tag{4.6}$$

where $\underline{le}_{n+1}(t)$ is the local error incurred on a step from time t_n to time t, $t_n \leqq t \leqq t_{n+1}$ and consequently $\underline{le}_{n+1}(t_n) = 0$. Shampine integrates (4.5) by using s steps of size h/s of the well-known theta method (e.g., see Chua and Dew [6]) to compute an approximation to $\underline{W}(t_{n+1})$. This integration is accomplished by making use of the Jacobian matrix

$$M = A_N(t) - h\gamma J \tag{4.7}$$

that is calculated by the ODE integrator and by assuming that this matrix is constant over the interval $[t_n, t_{n+1}]$. Define $\theta = \gamma s$, where the integer s is chosen so that $\frac{1}{2} \leqq \gamma s \leqq 1$ and so that the theta method is stable. For the backward differentiation formulae of orders 1 to 5 the value of s is at most 2. The extension of the formula used by Shampine (and by [6] with $s = 1$) to integrate equation (4.5) is

$$\underline{W}_{n,i+1} = \left[\frac{\theta - 1}{\theta} + \frac{1}{\theta} M^{-1} A_N\left(t_n + \frac{ih}{s}\right)\right] \underline{W}_{n,i}, \qquad i = 0, 1, \cdots, s-1 \tag{4.8}$$

where $\underline{W}_{n,0} = \underline{ge}(t_n)$ and $\underline{W}_{n,s} = \underline{W}(t_{n+1})$. The global error $\underline{ge}(t_{n+1})$ is then given by

$$\underline{ge}(t_{n+1}) \approx \underline{le}_{n+1}(t_{n+1}) + \underline{W}(t_{n+1}) \tag{4.9}$$

where $\underline{le}_{n+1}(t_{n+1})$ is the local error estimate calculated by the integrator at the end of the step from time t_n to time t_{n+1}.

4.2. Implementation of the ODE global error estimator. A possible difficulty with the error indicator described above is that the integrator may, for reasons of efficiency, take a step of size h^* while still using the Jacobian matrix (4.7) that was calculated with a step of size h. The effect of this is to change the value of theta by a multiplicative factor of h/h^*; in codes such as the Hindmarsh [9] codes this factor may vary between 0.7 and 1.3. An acceptable way to keep the integration of the error equation stable is to change the integration strategy so that the Jacobian matrix is re-evaluated whenever the step size is changed. The effect of this change of the integrator's strategy will be seen not to be significant for the numerical experiments described in § 7. The reader should note that the Jacobian matrix is used in the *main* integration to ensure only that the Newton iteration converges whereas in the error estimating procedure the Jacobian is central to the method used to integrate the error equation (4.5). While it is theoretically difficult to ensure that the Jacobian will be updated by the integrator frequently enough for the error integration to produce good estimates of the error this does not seem to be a major problem in practice.

Experiments with the error indicator have shown that it is necessary to monitor the local error incurred in the integration of the error equation. This may be done by using divided differences of the term $\underline{W}(t)$ in estimating the $O(h^2)$ term in the local error of the theta method [10]. In the case when this error is significantly larger than the local error added to $\underline{W}(t_{n+1})$ then the integration for the global error will have become unreliable and must be terminated. This situation has been observed in practice when the main integration method uses a large step size, such as may be the case when low local error accuracy is requested by the user. Experiments have shown that the local error estimate used for differential-algebraic equations (e.g., [4], [6], and [10]) given by

$$\underline{le}_{n+1}(t_{n+1}) = M^{-1}A_N(t_{n+1})\underline{le}^*_{n+1}(t_{n+1}) \tag{4.10}$$

where $\underline{le}^*_{n+1}$ is the usual local error estimate when $A_N(t) = I$, improves the reliability of the global error estimate, even when $A_N(t)$ is the identity matrix. The option to use this form of the error estimate in the SPRINT software has therefore been used in the numerical experiments reported in § 7.

5. A combined ODE/PDE error indicator. The vector of the values of the overall error at the spatial mesh points at any time is defined by $\underline{E}(t)$, where

$$\underline{E}(t) = \underline{u}(t) - \underline{V}(t) \tag{5.1}$$

where $\underline{u}(t)$ is the restriction of the PDE exact solution to the mesh δ, i.e.,

$$[\underline{u}(t)]_i = u(x_i, t), \qquad i = 1, \cdots, N.$$

The vector $\underline{E}(t)$ may also be written as a combination of the restriction of the PDE spatial discretisation error $\underline{es}(t)$, as defined by

$$\underline{es}(t) = \underline{u}(t) - \underline{U}(t),$$

and the ODE global error $\underline{ge}(t)$ (see (4.4)),

$$\underline{E}(t) = \underline{es}(t) + \underline{ge}(t). \tag{5.2}$$

The function $\underline{es}(t)$ represents the accumulation of the spatial discretisation error at the mesh points, [7]. An equation for the evolution of this error may be derived by adding terms to both sides of (4.1) to obtain the identity

$$A_N(t)\dot{\underline{u}}(t) - A_N(t)\dot{\underline{U}} = \underline{F}_N(t, \underline{u}(t)) - \underline{F}_N(t, \underline{U}(t)) + A_N(t)\dot{\underline{u}}(t) - \underline{F}_N(t, \underline{u}(t)),$$

which on using the definition of $\underline{es}(t)$ may be written as

$$A_N(t)\dot{\underline{es}} = \underline{F}_N(t, \underline{u}(t)) - \underline{F}_N(t, \underline{U}(t)) + \underline{TE}(t, \underline{u}(t)), \tag{5.3}$$

where the vector of spatial truncation errors on the mesh δ, as denoted by $\underline{TE}(t, \underline{u}(t))$, is defined by

$$\underline{TE}(t, \underline{u}(t)) = A_N(t)\underline{\dot{u}} - \underline{F}_N(t, \underline{u}(t)) \tag{5.4}$$

and from the initial conditions (2.5) and (3.4)

$$\underline{es}(0) = 0.$$

By making use of the approximation

$$\frac{\partial \underline{F}_N}{\partial \underline{U}} \underline{es}(t) \approx \underline{F}_N(t, \underline{u}(t)) - \underline{F}_N(t, \underline{U}(t))$$

(5.3) can be written in vector notation as

$$A_N(t)\underline{\dot{es}}(t) = \frac{\partial \underline{F}_N}{\partial \underline{U}} \underline{es}(t) + \underline{TE}(t, \underline{u}(t)), \qquad \underline{es}(0) = \underline{0}. \tag{5.5}$$

The integration of (5.5) is performed by using s steps of size h/s of the theta method, as in the estimation of the ODE global error, (4.8),

$$\underline{es}_{n,i+1} = \left[\frac{\theta-1}{\theta} + \frac{1}{\theta} M^{-1} A_N\left(t_n + \frac{i+1}{s} h\right)\right] \underline{es}_{n,i} + M^{-1}\left[\frac{h\theta}{s} \underline{TE}_{n,i+1} + \frac{h(1-\theta)}{s} \underline{TE}_{n,i}\right] + \underline{te}^*(t) \tag{5.6}$$

where

$$\underline{TE}_{n,i} = \underline{TE}(t_n + ih/s, \underline{u}(t_n + ih/s)), \qquad i = 0, 1, \cdots, s-1;$$

and where the vector $\underline{te}^*(t)$ is the ODE truncation error that arises in numerically integrating the error equation using the theta method. This error is considered in § 6.2. Defining

$$\underline{Z}(t) = \underline{es}(t) + \underline{W}(t) \tag{5.7}$$

and adding (5.6) and (4.8) gives

$$\underline{Z}_{n,i+1} = \left[\frac{\theta-1}{\theta} + \frac{1}{\theta} M^{-1} A_N\left(t_n + \frac{i+1}{s}\right)\right] \underline{Z}_{n,i} + M^{-1}\left[\frac{h\theta}{s} \underline{TE}_{n,i+1} + \frac{h(1-\theta)}{s} \underline{TE}_{n,i}\right] + \underline{te}^*(t), \quad i = 0, 1, \cdots, s-1. \tag{5.8}$$

The overall error at the end of the step is then given by

$$\underline{E}(t_{n+1}) = \underline{Z}(t_{n+1}) + \underline{le}_{n+1}(t_{n+1}). \tag{5.9}$$

In order to compute the terms $\underline{TE}_{n,i+1}$ and $\underline{TE}_{n,i}$ in (5.6) the exact solution on the mesh δ at the intermediate time points $t_n + ih/s$ as approximated by

$$\underline{u}(t_n + ih/s) \approx \underline{U}(t_n + ih/s) + \underline{Z}_{n,i} + \underline{le}_{n,i} \tag{5.10}$$

must be known. The term $\underline{le}_{n,i}$ is the local error at the intermediate point $t_n + ih/s$. Equation (5.8) is thus a nonlinear equation for $\underline{Z}_{n,i+1}$ as this term is used (see (5.10)) in evaluating $\underline{TE}_{n,i+1}$. The issue of how to estimate the spatial truncation errors will be discussed in the next section.

6. Implementation of the error indicator. In this section estimates for the spatial truncation error are derived and the implementation of the error estimator derived above is described.

6.1. Derivation and estimation of PDE truncation error. The estimate of the spatial truncation error used relies on the result [16] that the components of the spatial truncation error vector for the discretisation method of § 2 using the mesh δ satisfy

$$(6.1)\qquad \begin{aligned} &[\underline{TE}(t, u(t))]_1 = O(h_1),\\ &[\underline{TE}(t, \underline{u}(t))]_i = O(h_i^2 + h_i h_{i-1} + h_{i-1}^2), \qquad i = 2, \cdots, N-1,\\ &[\underline{TE}(t, \underline{u}(t))]_N = 0 \end{aligned}$$

where the *spatial* mesh widths h_i and h_{i-1} are defined by (2.7). This result allows the spatial truncation error to be estimated by using Richardson extrapolation. The actual mesh δ used to compute the numerical solution to the PDE is used as the "fine" mesh in the Richardson extrapolation process. The "coarse" mesh Δ^c is defined by

$$(6.2)\qquad \Delta^c : a = z_1 < z_2 < \cdots < z_M = b$$

where

$$z_i = x_{2i-1}, \quad i = 1, \cdots, M, \qquad M = (N+1)/2, \quad N \text{ is odd},$$

and the mesh points x_i as defined by (2.6) are assumed to satisfy

$$x_{2i} = \tfrac{1}{2}(z_i + z_{i+1}), \qquad i = 1, \cdots, M$$

so that we can make use of Richardson extrapolation. Let $\underline{u}^c(t)$ be the restriction of the PDE solution $u(x, t)$ to the new mesh Δ^c. The vector of spatial truncation errors, $\underline{TE}^c(t, \underline{u}^c(t))$, on the coarse mesh Δ^c is defined in the same way as the truncation error on the fine mesh (5.4) by

$$(6.3)\qquad \underline{TE}^c(t, \underline{u}^c(t)) = A_M(t)\dot{\underline{u}}^c(t) - \underline{F}_M(t, \underline{u}^c(t))$$

and satisfies

$$(6.4)\qquad [\underline{TE}^c(t, \underline{u}^c(t))]_i = 4[\underline{TE}(t, \underline{u}(t))]_{2i-1} + O(h_i^3 + h_{i-1}^3)$$

except at the left-hand boundary, where

$$(6.5)\qquad [\underline{TE}^c(t, \underline{u}^c(t))]_1 = 2([\underline{TE}(t, u(t))]_1 + O(h_1^2).$$

The spatial truncation error will be estimated by making use of two assumptions. The first is that there exists an exact solution to the discretised PDE on the coarse mesh. In other words there is an M-dimensional vector $\underline{w}(t)$ which satisfies

$$(6.6)\qquad A_M(t)\dot{\underline{w}} - \underline{F}_M(t, \underline{w}(t)) = 0,$$

together with an appropriate initial condition. The second assumption is that the spatial error $\underline{es}(t)$ is second order with respect to the spatial mesh intervals h_i. Define the vector $\underline{U}^c(t)$ by

$$(6.7)\qquad [\underline{U}^c(t)]_i = [\underline{U}(t)]_{2i-1}, \qquad i = 1, \cdots, M.$$

From the second assumption it follows that

$$\underline{u}^c(t) - \underline{U}^c(t) = \tfrac{1}{4}[\underline{u}^c(t) - \underline{w}(t)]$$

and so

$$(6.8)\qquad \underline{u}^c(t) = \tfrac{4}{3}\underline{U}^c(t) - \tfrac{1}{3}\underline{w}(t).$$

From (4.4) it follows that

$$\underline{U}^c(t)=\underline{V}^c(t)+\underline{ge}^c(t) \tag{6.9}$$

where the M-dimensional vector $\underline{V}^c(t)$ is defined by

$$[\underline{V}^c(t)]_i=[\underline{V}(t)]_{2i-1}, \qquad i=1,\cdots,M$$

and $\dot{\underline{V}}^c(t)$ is similarly defined using $\dot{\underline{V}}(t)$. The M-dimensional vector $\underline{ge}^c(t)$ and its time derivative $\dot{\underline{ge}}^c(t)$ are defined in the same way from the vector $\underline{ge}(t)$. On substituting (6.8) and its time derivative into the right-hand side of (6.3) and manipulating the expression using (6.6) we get

$$A_M(t)\dot{\underline{U}}^c(t)-\underline{F}_M(t,\underline{U}^c(t))=\tfrac{3}{4}(A_M(t)\dot{\underline{u}}^c(t)-\underline{F}_M(t,\underline{u}^c(t)))+O\|\underline{ge}^c(t)\|^2. \tag{6.10}$$

Substituting (6.9) and the time derivative of this equation into the left side of (6.10) gives

$$\begin{aligned}A_M(t)\dot{\underline{V}}^c(t)-\underline{F}_M(t,\underline{V}^c(t))&=\frac{3}{4}(A_M(t)\dot{\underline{u}}^c(t)-\underline{F}_M(t,\underline{u}^c(t)))\\&\quad-A_M(t)\dot{\underline{ge}}^c(t)+\frac{\partial\underline{F}_M}{\partial\underline{u}^c(t)}\underline{ge}^c(t)+O\|\underline{ge}^c(t)\|^2\end{aligned} \tag{6.11}$$

and using the definition of the spatial truncation error, (5.4), it follows that

$$\underline{TE}^c(t,\underline{u}^c(t))\approx\tfrac{4}{3}[A_M(t)\dot{\underline{V}}^c-\underline{F}_M(t,\underline{V}^c(t))]. \tag{6.12}$$

The error in (6.12) is thus:

$$\frac{4}{3}(A_M(t)\dot{\underline{ge}}^c(t)-\frac{\partial\underline{F}_M}{\partial\underline{u}^c(t)}\underline{ge}^c(t))+O\|\underline{ge}^c(t)\|^2. \tag{6.13}$$

This expression is difficult to quantify for a general matrix $A_M(t)$, however when the ODE global error is dominated by the spatial truncation error then this term will be "small" and (6.12) will provide a suitable estimate of the spatial truncation error. As the ODE global error is only indirectly controlled by the local error control used in the time integration it is not clear at present how the validity of the estimate defined by (6.12) can be guaranteed in practice.

The implementation of the estimate based on (6.12) thus requires one call to the subroutine that implements the spatial discretisation method, though with vectors that are one-half the length of those used to compute the numerical solution. The spatial truncation errors of the "fine" mesh solution can then be recovered using (6.4) and (6.5). In the case when the spatial truncation error is required at the points t_n+ih/s, $0<i<s$ interpolation is used (see § 4.1) to generate the values of $\underline{V}(t)$ and $\dot{\underline{V}}(t)$ required for the spatial truncation error estimate. The spatial truncation error in the solution on the mesh δ may then be estimated by using (6.10) and by using linear interpolation to estimate the truncation error at the points x_{2i}, $i=1,\cdots,M-1$ which are not in the mesh Δ^c, i.e.,

$$[\underline{TE}(t,\underline{u}(t))]_{2i}=\frac{1}{4(h_{2i-1}+h_{2i})}([\underline{TE}^c(t,\underline{V}^c(t))]_i h_{2i}+[\underline{TE}^c(t,\underline{V}^c(t))]_{i+1}h_{2i-1}). \tag{6.14}$$

It should be noted that the lower order of accuracy of the boundary spatial truncation errors when derivative boundary conditions are present means that care has to be taken in this interpolation at the second and penultimate points of the fine mesh. The solution adopted here is to interpolate at these two points as though the spatial truncation error is zero at the boundaries. This provides estimates of the spatial truncation error at the second and penultimate points of the fine mesh with the correct power of the spatial mesh width.

6.2. Computational considerations. Equation (5.8) can be used to estimate the components of the error vector $\underline{Z}(t_{n+1})$ providing that we can estimate the vector $\underline{te}^*(t)$ that arises in numerically integrating the error equation (5.4). This term $\underline{te}^*(t)$ can be estimated from the usual error estimate for the theta method but it is not clear whether or not this term needs to be included to provide a reliable error estimate. It is possible to monitor the size of this term and to ensure that it does not dominate the terms $\underline{TE}_{n,i+1}$ and $\underline{TE}_{n,i}$ and so corrupt the error estimating procedure. In the case when the error grows too large in the parallel integration it would be possible to reject the step in the main integration and to force it to be retaken using a step size that is appropriate for both integrations. In the numerical experiments described below the term $\underline{te}^*(t)$ has not been estimated or monitored.

The terms on the right-hand side of (5.8) involving M^{-1} can be combined so that only one application of back substitution using the LU decomposition of the matrix M instead of two is required. Once the first time step of the integration has been taken only one new evaluation of the PDE truncation error per step of (5.8) is needed as the estimate of the truncation error used at the end of one step may also be used at the start of the next step.

The computational overhead of the error indicator has been found by experiment to be as large as a factor of between 2 and 3 when the local error requirement forces the use of the backward differentiation formulae of orders 4 and 5. This in turn requires the use of $s=2$ in (5.6) with a cost of two back substitutions and two calls to the spatial discretisation method subroutine. This contrasts with less than two (on average) back substitutions and applications of the discretisation method per step of the main integration (excluding the cost of forming and decomposing a banded Jacobian matrix).

In the case of the backward differentiation formulae the use of $\theta=1$ and $s=1$ in (5.8) (regardless of the value of γ in (4.7)) provides a less theoretically sound but much more efficient procedure defined by

$$\underline{Z}_{n,i+1} = M^{-1}A_N(t_{n+1})\underline{Z}_{n,i} + hM^{-1}\underline{TE}_{n,i+1}, \qquad i=0, \tag{6.15}$$

which, by using (5.9), can be written as

$$\underline{E}(t_{n+1}) = M^{-1}(A_N(t_{n+1})\underline{E}(t_n) + h\underline{TE}_{n+1}) + \underline{le}_{n+1}(t_{n+1}) \tag{6.16}$$

where the local error estimate is given by (4.10). (When $\underline{le}^*_{n+1}(t_{n+1})$, the more usual form of the local error estimate, is used by the time integrator, then (4.10) can be used to rewrite (6.16) as

$$\underline{E}(t_{n+1}) = M^{-1}[A_N(t_{n+1})(\underline{E}(t_n) + \underline{le}^*_{n+1}(t_{n+1})) + h\underline{TE}_{n+1}]$$

so that the global error indicator still incorporates the modified form of the local error estimate given by (4.10).)

Equation (6.16) requires only one back-substitution and one evaluation of the truncation error per step and is similar to the procedure employed by Berzins and Dew [3] with their Chebyshev polynomial method. In practice the reliability of the error estimates does not appear to be compromised by this scheme. In the case when the ODE time integration error dominates this procedure is consistent with the global error estimator of Dew and West [8]. It is however more difficult to estimate the error in the integration defined by (6.16).

In the case when this approach is used with the backward differentiation formulae or if the theta method is used in the main integration the overhead of the global error estimator is reduced to an acceptable level of between 10 and 30 percent for moderately sized problems.

7. Numerical experiments. The following four test problems illustrate the effectiveness of the error indicator derived above. (Details of grid errors and error indices for Problems 1-4 are illustrated in Tables 8-11 in the Appendix.)

PROBLEM 1. The first problem is Burgers' equation which is defined by

$$\frac{\partial u}{\partial t}=\varepsilon\frac{\partial^2 u}{\partial x^2}+u\frac{\partial u}{\partial x}, \qquad (x, t)\in(0, 1)\times(0, 1]$$

where the value of $\varepsilon = 0.015$ was used in the experiments. The solution satisfies Dirichlet boundary conditions and initial conditions consistent with the analytic solution defined by

$$u(x, t)=\frac{0.1A+0.5B+C}{A+B+C}$$

where $A=e^{(-0.05(x-0.5+4.95t)/\varepsilon)}$, $B=e^{(-0.25(x-0.5+0.75t)/\varepsilon)}$ and $C=e^{(-0.5(x-0.375)/\varepsilon)}$.

PROBLEM 2. This problem was used by Berzins and Dew [3] to provide an example of a problem with a nonlinear source term and with nonlinear boundary conditions:

$$\frac{\partial u}{\partial t}=\frac{\partial^2 u}{\partial x^2}-2\left(\frac{\partial u}{\partial x}\right)^2\frac{1}{u}+(2+4t^3x)u^2, \qquad (x, t)\in[0, 1]\times(0, 1],$$

with boundary conditions

$$\frac{\partial u}{\partial x}(0, t)=-u^2(-2+t^4)$$

and

$$\frac{\partial u}{\partial x}(1, t)=-u^2t^4.$$

The initial conditions are consistent with the analytic solution

$$u(x, t)=\frac{1}{2-x^2+xt^4}.$$

PROBLEM 3. This problem provides an example of a problem with a nonlinear source term and a traveling wave solution:

$$\frac{\partial u}{\partial t}=\frac{\partial^2 u}{\partial x^2}+u^2(1-u), \qquad (x, t)\in(0, 10)\times(0, 1]$$

with Dirichlet boundary conditions and initial conditions consistent with the analytic solution of

$$u(x, t)=\frac{1}{1+e^{p(x-pt)}}$$

where $p=0.5\sqrt{2}$.

PROBLEM 4. This problem is the heat equation with Neumann boundary conditions:

$$\frac{\partial u}{\partial t}=\frac{\partial^2 u}{\partial x^2}, \qquad (x, t)\in[0, 1]\times(0, 0.2],$$

with the boundary conditions

$$\frac{\partial u}{\partial x}(x, t)=\pi e^{-\pi^2 t}\cos(\pi x)$$

at $x=0$ and $x=1$. The initial condition is consistent with the analytic solution

$$u(x, t)=\sin(\pi x)\, e^{-\pi^2 t}.$$

7.1. Testing procedure. The testing procedure employed for each of the problems was as follows. Equally spaced meshes of 11, 21, 41, 81, and 161 points (NPTS) were used with a mixed local error test and local error tolerances (TOL) of 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, and 10^{-6}, respectively. The only exception to this was for Problem 2 where a tolerance of 10^{-7} was used in the case of 161 mesh points. The time integration module used was the SPGEAR module of the SPRINT software [4] with the Linpack banded matrix routines. The computed measure of the accuracy of the error estimates calculated was the error index $E_I(t)$ defined by

$$E_I(t)=\frac{\|\text{Estimated grid errors at time } t\|_\infty}{\|\text{Actual grid errors at time } t\|_\infty}.$$

This error index was calculated at the end of every time step of the integration. Tables 1–4 contain information on the values of the error index for the four test problems above.

Key to Tables 1–4.

Start is the value of the error index at the first output point ($t=0.01$).

Finish is the final value of the error index at the end of integration.

Average is the average value of the error index sampled at the end of every time step.

TABLE 1
Error indices for Problem 1.

NPTS	Start	Finish	Average	Max	Min	Cost 1	Cost 2
11	0.71	0.52	0.50	0.78	0.23	1.09	1.09
21	0.62	0.50	0.65	0.87	0.43	1.18	1.21
41	0.98	0.82	0.91	1.10	0.64	1.23	1.22
81	1.10	0.88	0.91	1.10	0.75	1.25	1.30
161	1.10	0.94	0.95	1.20	0.79	1.30	1.32

TABLE 2
Error indices for Problem 2.

NPTS	Start	Finish	Average	Max	Min	Cost 1	Cost 2
11	0.83	0.25	0.91	1.60	0.23	1.17	1.17
21	0.92	0.83	1.10	1.50	0.39	1.19	1.25
41	0.94	1.00	1.30	2.10	0.55	1.17	1.40
81	0.96	0.78	1.20	2.70	0.61	1.25	1.25
161	0.97	0.98	1.40	3.40	0.64	1.26	1.26

TABLE 3
Error indices for Problem 3.

NPTS	Start	Finish	Average	Max	Min	Cost 1	Cost 2
11	0.81	1.00	0.83	1.00	0.63	1.05	1.10
21	0.95	1.00	0.96	1.10	0.97	1.05	1.20
41	0.98	1.10	1.10	1.20	0.99	1.13	1.25
81	1.00	1.20	1.10	1.30	0.99	1.17	1.23
161	1.20	1.30	1.20	1.40	1.00	1.21	1.21

TABLE 4
Error indices for Problem 4.

NPTS	Start	Finish	Average	Max	Min	Cost 1	Cost 2
11	1.10	1.30	1.20	1.40	0.65	1.12	1.05
21	1.20	1.30	1.30	1.60	0.66	1.17	1.09
41	1.20	1.30	1.30	1.60	0.67	1.20	1.12
81	1.40	1.30	1.40	1.70	0.67	1.23	1.00
161	1.40	1.40	1.50	1.70	0.67	1.27	0.84

Max is the maximum value of the error index calculated at the end of every time step.

Min is the minimum value of the error index found throughout the range of integration.

Cost 1 is the computational overhead cost in supplying error estimates, neglecting the change in integrator strategy described in § 4.2, and is defined by

$$\text{Cost } 1 = \frac{\text{c.p.u. time including calculating error estimates}}{\text{c.p.u. time excluding calculating error estimates}}.$$

In this case the integrator uses the same stepsize sequence and number of Jacobian evaluations whether or not the global error estimates are calculated.

Cost 2 is the computational overhead cost in supplying error estimates and including the change in integrator strategy described in § 4.2, and is defined by

$$\text{Cost } 2 = \frac{\text{c.p.u. time including calculating error estimates}}{\text{c.p.u. time excluding calculating error estimates and using original integrator}}.$$

In this case the integrator uses slightly different stepsize sequences and numbers of Jacobian evaluations depending on whether or not the global error estimates are calculated.

7.2. Comments on numerical results. The error estimates for Problems 1 and 3 are in good agreement with the actual error. For Problems 2 and 4 the combination of derivative boundary conditions and the nonlinear source term makes it more difficult to provide good estimates of the error.

An interesting feature of the above results is that the error indices do not appear to converge to one as we increase the number of points. In particular the minimum value of the error index remains fixed at $\frac{2}{3}$ for the problems with derivative boundary conditions. There are two main sources of inaccuracy in the approach we have used:

(1) The global error from the ODE integrator may be corrupting the estimate of the space discretisation error.

(2) The time integration scheme used in integrating the error equation may be sufficiently inaccurate to degrade the quality of the error estimates.

In order to investigate this situation the *exact* spatial truncation error was used and the experiments repeated. This resulted in only minor changes in the quality of the error estimates. A similar situation occurred when a local error tolerance of 10^{-3} was used with the original estimate (6.12) of the spatial truncation error in that the overall quality of the error estimates was undiminished. Finally the experiments were repeated again using the original estimate of the spatial truncation error but using the backward Euler method for the main integration with a local error tolerance of 10^{-7}. This ensured that the spatial discretisation error was dominant and that the integration formula (6.16) could be justified on the basis of the approach of § 5. The error indices for Problems 1 and 4 are shown in Tables 5 and 6.

In Table 5 the error indicator exhibits good asymptotic behaviour as we increase the number of mesh points.

In Table 6 the asymptotic behaviour of the error estimates is also good apart from the consistently low value of the minimum error index. These low values of the error estimate occur at the very start of integration when the error is dominated by the initial error in approximating the derivative boundary conditions. Table 7 shows the values of this error at the end of the first time step for different values of NPTS using a local error tolerance of 10^{-8}. From Table 7 it is clear that the initial space error is first order and so the second assumption used to derive the spatial truncation error estimate does not apply. This causes the estimate (6.12) to underestimate the spatial truncation error

TABLE 5
Error indices for Problem 1.

NPTS	Start	Finish	Average	Max	Min
11	0.56	0.60	0.50	0.79	0.20
21	0.44	0.71	0.74	0.95	0.42
41	0.91	0.93	0.91	0.97	0.84
81	0.96	0.97	0.97	0.99	0.95
161	1.00	1.00	0.99	1.00	0.99

TABLE 6
Error indices for Problem 4.

NPTS	Start	Finish	Average	Max	Min
11	0.90	1.00	0.98	1.00	0.65
21	0.98	1.00	0.99	1.00	0.66
41	0.99	1.00	1.00	1.00	0.67
81	1.00	1.00	1.00	1.00	0.67
161	1.00	1.00	1.00	1.00	0.67

TABLE 7
Maximum grid errors for Problem 4 *at end of first step.*

NPTS	11	21	41	81
Error	$4.9E-6$	$2.4E-6$	$1.2E-6$	$6.0E-7$

for the first few time steps. Eventually the spatial discretisation error exhibits second-order behaviour and the error index tends to one. It is not clear if this behaviour is particular to this problem or to the discretisation method described in § 3. One solution to the difficulty would be to modify the discretisation scheme so that derivative boundary conditions are treated more accurately. This point requires further investigation.

Despite this difficulty with derivative boundary conditions for the discretisation method described in § 3, the error indicator provides a generally good indication of the overall error.

In general the computational cost of the approach has been reduced to an acceptable level for the class of problems that we have considered and the effect of changing the integration strategy does not seem significant. In the case of Problem 4 the new integration strategy is more efficient (once the overhead of the error indicator is taken into account) as the more up-to-date Jacobian matrix (4.7) results in faster convergence and allows the integrator to use fewer time steps.

Appendix 1 contains more detailed information on the size of the error indices and the size of the maximum grid error at a subset of the discrete times at which the error was estimated. The estimate of the maximum grid error computed by the error indicator is then given by

$$\text{Estimated grid error} = \text{actual grid error} \times \text{error index}.$$

The Appendix thus shows that the error estimates computed are in general in reasonably good agreement with the actual error.

8. Exploiting the global error estimates. The idea used to estimate the combined error can be extended to more general PDE's, provided that the PDE truncation error can be estimated in the same way and that local error control is used in the time integration. The implicit ODE form, (4.1), used in estimating the ODE global error will allow the error estimator to be extended to mixed ODE/PDE problems and to parabolic-elliptic systems of PDE's. Though in this case it may be necessary to modify the ODE global error estimating procedure to take account of differential-algebraic equations.

The error indicator can be applied to discrete time remeshing methods such as that used by Berzins, Dew and Furzeland [4]. The interpolation procedure used to interpolate from the old mesh to the new must also be applied to the error estimate and some attempt made to estimate the error introduced by interpolation. It might also be possible to try to balance the local contributions to the global error from both space and time.

9. Summary. From our practical experience, the error indicator derived above seems to be a promising means of estimating the total error in the numerical solution. Over a limited range of simple parabolic equations the indicator has been found to be reliable and a considerable improvement over the earlier method of Berzins and Dew [3] both in terms of the accuracy of time integration of the error equation and in terms of the accuracy of the PDE truncation error.

Further work needs to be done to consider how the time-integration scheme for the error equation might be made more reliable and how the time integration error can be controlled so that the spatial discretisation error estimate remains valid. Further work also needs to be done to extend the error estimate to differential-algebraic equations and to coupled ODE/PDE problems and to combine the error estimates with adaptive space remeshing in order to equally distribute the error between the space and time integrations.

TABLE 8
Output for Problem 1.

NPTS	Time		Max grid error at output points									
			0.10D−01	0.11D+00	0.22D+00	0.33D+00	0.44D+00	0.56D+00	0.67D+00	0.78D+00	0.89D+00	0.10D+01
11	GRID	ERROR	0.18D−02	0.24D−01	0.34D−01	0.44D−01	0.47D−01	0.90D−01	0.11D+00	0.10D+00	0.15D+00	0.14D+00
	ERROR	INDEX	0.71D+00	0.23D+00	0.42D+00	0.49D+00	0.49D+00	0.50D+00	0.58D+00	0.52D+00	0.48D+00	0.52D+00
21	GRID	ERROR	0.82D−03	0.62D−02	0.11D−01	0.16D−01	0.21D−01	0.30D−01	0.39D−01	0.39D−01	0.47D−01	0.56D−01
	ERROR	INDEX	0.62D+00	0.82D+00	0.82D+00	0.65D+00	0.70D+00	0.63D+00	0.58D+00	0.64D+00	0.69D+00	0.50D+00
41	GRID	ERROR	0.22D−03	0.18D−02	0.28D−02	0.42D−02	0.60D−02	0.81D−02	0.99D−02	0.12D−01	0.13D−01	0.14D−01
	ERROR	INDEX	0.98D+00	0.98D+00	0.90D+00	0.78D+00	0.74D+00	0.72D+00	0.75D+00	0.73D+00	0.68D+00	0.82D+00
81	GRID	ERROR	0.54D−04	0.44D−03	0.73D−03	0.11D−02	0.15D−02	0.20D−02	0.25D−02	0.29D−02	0.33D−02	0.36D−02
	ERROR	INDEX	0.11D+01	0.11D+01	0.10D+01	0.87D+00	0.83D+00	0.77D+00	0.80D+00	0.87D+00	0.81D+00	0.88D+00
161	GRID	ERROR	0.14D−04	0.11D−03	0.18D−03	0.26D−03	0.38D−03	0.50D−03	0.63D−03	0.74D−03	0.82D−03	0.90D−03
	ERROR	INDEX	0.11D+01	0.11D+01	0.10D+01	0.85D+00	0.82D+00	0.82D+00	0.88D+00	0.90D+00	0.90D+00	0.94D+00

TABLE 9
Output for Problem 2.

NPTS	Time		Max grid error at output points									
			0.10D−01	0.22D+00	0.44D+00	0.67D+00	0.89D+00	0.11D+01	0.13D+01	0.16D+01	0.18D+01	0.20D+01
11	GRID	ERROR	0.37D−02	0.36D−02	0.85D−02	0.84D−02	0.56D−02	0.51D−02	0.43D−02	0.76D−02	0.33D−01	0.99D−01
	ERROR	INDEX	0.83D+00	0.87D+00	0.96D+00	0.12D+01	0.12D+01	0.16D+01	0.13D+01	0.60D+00	0.79D+00	0.25D+00
21	GRID	ERROR	0.88D−03	0.82D−03	0.17D−02	0.14D−02	0.12D−02	0.12D−02	0.12D−02	0.15D−02	0.59D−02	0.29D−01
	ERROR	INDEX	0.92D+00	0.11D+01	0.11D+01	0.13D+01	0.13D+01	0.15D+01	0.12D+01	0.77D+00	0.64D+00	0.83D+00
41	GRID	ERROR	0.22D−03	0.20D−03	0.34D−03	0.33D−03	0.28D−03	0.31D−03	0.30D−03	0.39D−03	0.14D−02	0.77D−02
	ERROR	INDEX	0.94D+00	0.13D+01	0.11D+01	0.12D+01	0.12D+01	0.14D+01	0.21D+01	0.17D+01	0.84D+00	0.10D+01
81	GRID	ERROR	0.56D−04	0.44D−04	0.77D−04	0.84D−04	0.70D−04	0.77D−04	0.76D−04	0.89D−04	0.33D−03	0.19D−02
	ERROR	INDEX	0.96D+00	0.17D+01	0.11D+01	0.12D+01	0.12D+01	0.13D+01	0.17D+01	0.10D+01	0.81D+00	0.78D+00
161	GRID	ERROR	0.14D−04	0.99D−05	0.18D−04	0.20D−04	0.17D−04	0.19D−04	0.17D−04	0.22D−04	0.82D−04	0.49D−03
	ERROR	INDEX	0.97D+00	0.27D+01	0.11D+01	0.12D+01	0.12D+01	0.14D+01	0.12D+01	0.15D+01	0.11D+01	0.98D+00

TABLE 10
Output for Problem 3.

NPTS	Time		Max grid error at output points									
			0.10D−01	0.11D+00	0.22D+00	0.33D+00	0.44D+00	0.56D+00	0.67D+00	0.78D+00	0.89D+00	0.10D+01
11	GRID	ERROR	0.10D−03	0.10D−02	0.18D−02	0.22D−02	0.31D−02	0.38D−02	0.42D−02	0.43D−02	0.45D−02	0.48D−02
	ERROR	INDEX	0.81D+00	0.81D+00	0.81D+00	0.81D+00	0.10D+01	0.10D+01	0.10D+01	0.10D+01	0.10D+01	0.10D+01
21	GRID	ERROR	0.30D−04	0.23D−03	0.44D−03	0.52D−03	0.59D−03	0.65D−03	0.73D−03	0.80D−03	0.88D−03	0.94D−03
	ERROR	INDEX	0.95D+00	0.11D+01	0.11D+01	0.11D+01	0.11D+01	0.11D+01	0.10D+01	0.10D+01	0.10D+01	0.10D+01
41	GRID	ERROR	0.86D−05	0.50D−04	0.71D−04	0.92D−04	0.11D−03	0.13D−03	0.15D−03	0.17D−03	0.18D−03	0.20D−03
	ERROR	INDEX	0.98D+00	0.11D+01	0.11D+01	0.11D+01	0.11D+01	0.12D+01	0.12D+01	0.12D+01	0.11D+01	0.11D+01
81	GRID	ERROR	0.19D−05	0.96D−05	0.17D−04	0.24D−04	0.27D−04	0.30D−04	0.33D−04	0.37D−04	0.41D−04	0.45D−05
	ERROR	INDEX	0.10D+01	0.10D+01	0.12D+01	0.12D+01	0.13D+01	0.13D+01	0.13D+01	0.13D+01	0.12D+01	0.12D+01
161	GRID	ERROR	0.41D−06	0.25D−05	0.40D−05	0.52D−05	0.63D−05	0.72D−05	0.80D−05	0.88D−05	0.99D−05	0.11D−04
	ERROR	INDEX	0.12D+01	0.11D+01	0.12D+01	0.12D+01	0.12D+01	0.13D+01	0.14D+01	0.14D+01	0.13D+01	0.13D+01

TABLE 11
Output for Problem 4.

NPTS	Time		Max grid error at output points									
			0.10D−01	0.28D−01	0.56D−01	0.83D−01	0.11D+00	0.14D+00	0.17D+00	0.19D+00	0.22D+00	0.25D+00
11	GRID	ERROR	0.51D−02	0.75D−02	0.86D−02	0.89D−02	0.86D−02	0.78D−02	0.70D−02	0.64D−02	0.59D−02	0.56D−02
	ERROR	INDEX	0.11D+01	0.12D+01	0.12D+01	0.14D+01	0.14D+01	0.14D+01	0.14D+01	0.14D+01	0.13D+01	0.13D+01
21	GRID	ERROR	0.13D−02	0.19D−02	0.21D−02	0.20D−02	0.18D−02	0.17D−02	0.16D−02	0.16D−02	0.15D−02	0.15D−02
	ERROR	INDEX	0.12D+01	0.13D+01	0.14D+01	0.15D+01	0.15D+01	0.16D+01	0.15D+01	0.14D+01	0.14D+01	0.13D+01
41	GRID	ERROR	0.33D−03	0.47D−03	0.51D−03	0.49D−03	0.46D−03	0.44D−03	0.42D−03	0.41D−03	0.39D−03	0.38D−03
	ERROR	INDEX	0.12D+01	0.14D+01	0.14D+01	0.16D+01	0.16D+01	0.15D+01	0.15D+01	0.15D+01	0.14D+01	0.13D+01
81	GRID	ERROR	0.83D−04	0.12D−03	0.13D−03	0.12D−03	0.12D−03	0.11D−03	0.10D−03	0.97D−04	0.92D−04	0.89D−04
	ERROR	INDEX	0.14D+01	0.15D+01	0.16D+01	0.17D+01	0.17D+01	0.17D+01	0.16D+01	0.15D+01	0.14D+01	0.13D+01
161	GRID	ERROR	0.21D−04	0.29D−04	0.32D−04	0.30D−04	0.29D−04	0.27D−04	0.25D−04	0.24D−04	0.23D−04	0.22D−04
	ERROR	INDEX	0.14D+01	0.15D+01	0.17D+01	0.17D+01	0.17D+01	0.16D+01	0.16D+01	0.15D+01	0.14D+01	0.14D+01

Appendix 1. Details of the grid errors and error indices for Problems 1 to 4. See Tables 8-11.

Acknowledgment. Ian Gladwell's comments on an early version of this paper are appreciated.

REFERENCES

[1] S. ADJERID AND J. FLAHERTY, *Moving finite element method for time dependent equations*, SIAM J. Numer. Anal., 23 (1986), pp. 778-796.

[2] I. BABUŠKA AND M. BIETERMAN, *An adaptive method of lines with error control for parabolic equations of reaction-diffusion type*, J. Comput. Phys., 63 (1986), pp. 33-66.

[3] M. BERZINS AND P. M. DEW, *A note on C^0 Chebyshev methods for parabolic p.d.e.s in one space variable*, IMA J. Numer. Anal., 7 (1987), pp. 15-37.

[4] M. BERZINS, P. M. DEW, AND R. M. FURZELAND, *Developing P.D.E. software using the method of lines and differential algebraic integrators*, Paper presented at 1986 O.D.E. Conference, Albuquerque, New Mexico, Report 220, Department of Computer Studies, Leeds University, Leeds, England.

[5] J. G. BLOM, J. M. SANZ-SERNA, AND J. G. VERWER, *On simple moving grid methods for one dimensional evolutionary partial differential equations*, Report NM-R8620, September 1986, Centre for Mathematics and Computer Science, P.O. Box 4079, Amsterdam.

[6] T. S. CHUA AND P. M. DEW, *The design of a variable step integrator for the simulation of gas transmission networks*, Internat. J. Numer. Methods Engng., 20 (1984), pp. 1797-1813.

[7] M. J. P. CULLEN AND K. W. MORTON, *Analysis of evolutionary error in finite element and other methods*, J. Comput. Phys., 34 (1980), pp. 245-269.

[8] P. M. DEW AND M. WEST, *Estimating and controlling the local error in Gear's method*, BIT 19 (1978), pp. 135-137.

[9] A. C. HINDMARSH, ODERPACK—*A systematised collection of* O.D.E. *solvers*, in Advances in Computer Methods IV, R. Vichnevetsky and R. S. Stepleman, eds., IMACS, New Brunswick, 1986.

[10] T. R. HOPKINS, *Numerical solution of quasi-linear parabolic differential equations*, Ph.D. thesis, University of Liverpool, Liverpool, England, 1976.

[11] H. B. KELLER, *A new difference scheme for parabolic problems*, in Numerical Solution of P.D.E.s—II Synspade (1970), B. Hubbard, ed., Academic Press, New York, 1971.

[12] K. MILLER, *Alternate modes to control the nodes in the moving finite element method*, in Adaptive Computational Methods for PDEs, I. Babuška, J. Chandra, and J. E. Flaherty, eds., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1983.

[13] L. R. PETZOLD, *Observations on an adaptive moving grid method for one dimensional systems of partial differential equations*, URCL Preprint, 94629, Lawrence Livermore Laboratory, Livermore, California, 1986.

[14] L. F. SHAMPINE, *Global error estimation for stiff o.d.e.s.*, Proc. Dundee Conference on Numerical Analysis 1983, Springer-Verlag Lecture Notes in Mathematics 1066, Springer-Verlag, Berlin, New York, Heidelberg, 1984.

[15] W. SCHÖNAUER, E. SCHEMPF, AND K. RAITH, *The redesign and vectorization of the* SLDGL—*program package for the selfadaptive solution of nonlinear systems of elliptic and parabolic* P.D.E.s, in PDE Software: Modules, Interfaces and Systems, B. Engquist and T. Smedsaas, eds., North Holland, Amsterdam, 1984.

[16] R. D. SKEEL, *Improving routines for solving parabolic equations in one space variable*, Numer. Anal. Rept. 63, Department of Mathematics, University of Manchester, Manchester, England, 1981.