SIAM J. SCI. STAT. COMPUT.
Vol. 12, No. 3, pp. 573-594, May 1991

BALANCING SPACE AND TIME ERRORS IN THE METHOD OF LINES FOR PARABOLIC EQUATIONS*

J. LAWSON†, M. BERZINS†, AND P. M. DEW†

Abstract. A new error control strategy for the time integration in the solution of parabolic equations using the method of lines is presented. The strategy aims to approximately balance the spatial discretisation and time integration errors so that they are of the same order of magnitude, but so that the time integration error is less than the spatial discretisation error. This is achieved by making use of the individual contributions of the spatial discretisation error and the time integration error to an existing estimate of the global error in the numerical solution. The new strategy is presented in the light of this global error indicator and a comparison between the new error control strategy and a similar existing strategy is made. Numerical results are used to illustrate the performance of this strategy.

Key words. parabolic equations, method of lines, spatial and temporal errors

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

1. Introduction. The method of lines is widely used in general-purpose software for the integration of time-dependent parabolic and elliptic-parabolic partial differential equations (pde's). Two of the factors influencing the performance of the method of lines are the choice of a spatial discretisation method and the positioning of the spatial discretisation points. The points should be chosen so that the computed solution accurately models the exact solution to the pde, or, in other words, so that the spatial discretisation error is controlled as far as possible. Once the spatial mesh has been chosen, it is desirable to integrate the ordinary differential equation (ode) system in time with just sufficient accuracy so that the temporal error does not significantly corrupt the spatial accuracy. However, in most existing software based on the method of lines, the standard procedure is to control the local time error per step [9] with respect to a supplied accuracy tolerance. It is difficult for the user to select a tolerance which is related to the spatial discretisation error. In addition, controlling the local time error per step does not always guarantee equivalent control of the global time error, although, for a given problem, decreasing the local time error will generally lead to a smaller global time error.

The purpose of this paper is to present a new error control strategy for the time integration which uses the global error estimating algorithm of Berzins [3], as a means of approximately balancing the contributions of the spatial and temporal errors to the global error in the computed solution. This leads to a strategy in which the local time error is controlled in such a way that it is a fraction of the change in the spatial error over each timestep. In this way the error tolerance used in the time integration is varied according to the size of the spatial discretisation error. The analysis of the new strategy shows that the time integration error remains below the spatial discretisation error and that the strategy is a form of local error per unit time step control, in contrast to the more usual local error per timestep control.

This paper is structured in the following way. Section 2 outlines the background theory and notation, given in Berzins [3], needed to describe the global error indicator. This allows the main contribution of the paper to be given in § 3 where the new error

* Received by the editors December 5, 1988; accepted for publication (in revised form) March 12, 1990.

† School of Computer Studies, The University, Leeds LS2 9JT, United Kingdom. The work of the first author was supported by Shell Research Limited and the Science and Engineering Research Council of the United Kingdom through a CASE studentship.

control strategy is fully described. In § 4, this new strategy is compared to that presented by Schönauer, Schnepf, and Raith [16], which is one of the few existing algorithms which estimate and combine the spatial and temporal error estimates. Finally, in §§ 5 and 6, a summary and discussion of the numerical experiments is presented. It is shown that the error control strategy appears to offer an effective method of balancing the spatial and temporal errors in the method of lines.

2. Problem class and method of solution.

2.1. Problem class. The problem class considered here is sufficiently general to illustrate the algorithm for error estimation. The algorithm extends naturally to systems of pde's and to equations in more than one space dimension, providing 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

$$\frac{\partial u}{\partial t}=\frac{\partial}{\partial x}r\left(x,t,u,\frac{\partial u}{\partial x}\right)+f\left(x,t,u,\frac{\partial u}{\partial x}\right), \tag{2.1}$$

defined on $(x,t)\in\Omega=[a,b]\times(0,t_e]$. The boundary conditions are of the form

$$\beta(a,t)r\left(a,t,u(a,t),\frac{\partial u}{\partial x}\right)=\alpha(a,t)u(a,t)-g_a(t) \tag{2.2}$$

and

$$\beta(b,t)r\left(b,t,u(b,t),\frac{\partial u}{\partial x}\right)=\alpha(b,t)u(b,t)-g_b(t) \tag{2.3}$$

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

$$u(x,0)=u_0(x),\qquad x\in[a,b]. \tag{2.4}$$

It is assumed that the pde defined above is well posed and has a unique continuous solution $u(x,t)$ for all $(x,t)\in\Omega$. The spatial mesh, the points at which the solution is to be computed, is defined by

$$\delta: a=x_1<x_2<\cdots<x_N=b. \tag{2.5}$$

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

$$h_j=x_{j+1}-x_j,\qquad h=\max h_j,\qquad j=1,2,\cdots,N-1. \tag{2.6}$$

The method of lines is used to solve the pde numerically. That is, the pde is discretised in space and the resulting system of time-dependent ode's is solved using existing techniques for solving ode's. Second-order finite-difference methods are commonly used to spatially discretise many of the parabolic equations that arise in practice. Although the error control strategy proposed in this paper can be used in conjunction with any of these schemes, the modified version of the box scheme [11] proposed by Skeel and Berzins [19] is used here. This scheme is particularly convenient as it spatially discretises the pde above into an ode system in normal form. The discretisation method can be written as

$$\frac{\partial U}{\partial t}(x_j,t)=\frac{1}{h_{j-1}+h_j}[2(R_{j+1/2}-R_{j-1/2})+(h_{j-1}f_{j-1/2}+h_jf_{j+1/2})],$$

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 $f_{j+1/2}$ 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

$$\beta(a, t)(c_{11/2})\frac{\partial U}{\partial t}(x_1, t)=2(\beta(a, t), R_{11/2}-(\alpha(a, t)u(a, t)-g_a(t)))/h_1+\beta(a, t)f_{11/2}$$

and the condition at $x=b$ is treated similarly. When $\beta=0$, substitution for the solution, u, will lead to a system of time-dependent ode's, although in practice it is easier to leave the boundary conditions as coupled algebraic equations. The initial condition is defined by evaluating the function $u_0(x)$ at the spatial meshpoints

$$U(x_i, 0)=u_0(x_i), \qquad i=2, \cdots, N-1.$$

For ease of exposition in the rest of the paper, we shall assume that this system of time-dependent ode's can be written as the normal form ode system

$$\dot{\underline{U}}=\underline{F}_N(t, \underline{U}(t)), \tag{2.7}$$

where the N-dimensional vector is defined by

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

In the case of at least one Dirichlet boundary condition, the system must be modified accordingly. In practice, (2.7) is a stiff or mildly stiff system of time-dependent ode's that can be solved using software based on the backward differentiation formulae, e.g., [5], [9].

2.2. Error control for stiff ode's. In most of the codes available for solving time-dependent ode's, the routines attempt to control the local time integration error in the computed solution with regard to an accuracy tolerance supplied by the user, TOL. The initial value problem to be solved is given by equation (2.7) with the true solution $\{\underline{U}(t_n)\}_{n=0}^{p}$ approximated by $\{\underline{V}(t_n)\}_{n=0}^{p}$ at a set of discrete times $0=t_0<t_1<\cdots<t_p=t_e$ by a time integration method with requested local error accuracy TOL. After taking a stepsize of size k_n, from $[t_n, t_{n+1}]$, the global time error in the ode solution at time t_{n+1} and for a tolerance TOL, is given by

$$\underline{ge}(t_{n+1}, \mathrm{TOL})=\underline{U}(t_{n+1})-\underline{V}(t_{n+1}). \tag{2.8}$$

The local solution on $[t_n, t_{n+1}]$, $\underline{y}_{n+1}(t, \mathrm{TOL})$, is defined as the solution of the ivp

$$\dot{\underline{y}}_{n+1}(t, \mathrm{TOL})=\underline{F}_N(t, \underline{y}_{n+1}(t, \mathrm{TOL})), \qquad \underline{y}_{n+1}(t_n, \mathrm{TOL})=\underline{V}(t_n), \tag{2.9}$$

and so the local error per step (LEPS) at t_{n+1} can be given by

$$\underline{le}_{n+1}(\mathrm{TOL})=\underline{V}(t_{n+1})-\underline{y}_{n+1}(t_{n+1}, \mathrm{TOL}) \tag{2.10}$$

and the local error per unit per step (LEPUS) is given by $\underline{le}_{n+1}(\mathrm{TOL})/k_n$.

Control over either the LEPS or the LEPUS, with respect to an accuracy tolerance TOL can be achieved by varying the time stepsize and also by using formulae of different orders. In order to balance the spatial and temporal errors, it is desirable that the error control strategy should yield a solution that satisfies tolerance proportionality; that is, there exists a linear relationship between the global time error and the requested accuracy. An error control strategy is said to satisfy tolerance proportionality (Stetter [20]) if the numerical solution is such that if $\underline{ge}(t, \text{TOL})$ is the global time error at time t for an accuracy requirement TOL, then

$$\underline{ge}(t, \text{TOL}) = \underline{\nu}(t)\text{TOL} + o(\text{TOL}), \tag{2.11}$$

where $\underline{\nu}(t)$ is independent of TOL and $\underline{\nu}(t)$ and $\underline{\nu}'(t)$ are bounded on $[0, t_e]$. Here, $o(\text{TOL})$ denotes a term that is numerically negligible compared with terms of order TOL in the same equation. The following theorem shows how tolerance proportionality can be achieved by employing suitable local error control strategies.

THEOREM 1 (Stetter [20]). *Equation* (2.11) *is satisfied for all t if and only if the local error $\underline{le}_{n+1}(\text{TOL})$ obtained for a fixed accuracy tolerance* TOL *satisfies*

$$\underline{le}_{n+1}(\text{TOL}) = \tilde{\underline{\gamma}}(t_n, t_{n+1})k_n\text{TOL} + o(\text{TOL}), \quad t \in [t_n, t_{n+1}], \quad n = 0, \cdots, p, \tag{2.12}$$

where $\tilde{\underline{\gamma}}(t, \tau)$ behaves like an integral mean over $[t, \tau]$ of a function that is independent of TOL *and bounded on $[0, t_e]$ and $k_n = t_{n+1} - t_n$.*

From this theorem, it follows that in order to obtain tolerance proportionality, the LEPUS must be controlled rather than the LEPS.

2.3. The global error indicator of Berzins [3]. An efficient strategy which controls the local time error in such a way that the spatial discretisation error dominates can be developed through considering the global error indicator of Berzins [3].

The vector of the values of the global error at the spatial meshpoints, at any time t, is defined by

$$\underline{E}(t) = \underline{u}(t) - \underline{V}(t), \tag{2.13}$$

where $\underline{u}(t)$ is the restriction of the exact pde 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 $\underline{ge}(t, \text{TOL})$, that is,

$$\underline{E}(t) = \underline{es}(t) + \underline{ge}(t, \text{TOL}). \tag{2.14}$$

The solution of the variational equation (see Shampine [17])

$$\dot{\underline{W}} = J\underline{W}, \qquad \underline{W}(t_n) = \underline{ge}(t_n, \text{TOL}) \tag{2.15}$$

(where $J = \partial \underline{F}_N / \partial \underline{U}$) is related to the global time error at the end of the step $\underline{ge}(t_{n+1}, \text{TOL})$ by

$$\underline{ge}(t_{n+1}, \text{TOL}) \approx \underline{W}(t_{n+1}) + \underline{le}_{n+1}(\text{TOL}). \tag{2.16}$$

Experiments, e.g., [7], [10], have shown that the reliability of the global time error estimate can be improved by using a modified version of the local time error, that is, by using

$$\underline{le}_{n+1}(\text{TOL}) = M^{-1}\widehat{\underline{le}}_{n+1}(\text{TOL}), \tag{2.17}$$

where $\widehat{\underline{le}}_{n+1}$ is the usual local error estimate when the ode system is in normal form and $M = I - k_n \gamma J$. This is so, even, as in this case, when the ode system is in normal form. The equation used by Berzins [3], for the evolution of $\underline{es}(t)$ is

$$\dot{\underline{es}}(t) = \frac{\partial \underline{F}_N}{\partial \underline{U}} \underline{es}(t) + \underline{TE}(t, \underline{u}(t)), \qquad \underline{es}(0) = \underline{0}, \tag{2.18}$$

where

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

is the spatial truncation error. Since, in general, the exact solution is unknown the spatial truncation error must be estimated. Berzins [3] shows that if the spatial discretisation error dominates the time integration error, then the estimate of the spatial truncation error found through Richardson extrapolation techniques is reliable. Equations (2.15) and (2.18) can be solved using the theta method and so the global error at t_{n+1} can be given by

$$\underline{E}(t_{n+1}) = \underline{es}(t_{n+1}) + \underline{W}(t_{n+1}) + \underline{le}_{n+1}(\mathrm{TOL}). \tag{2.19}$$

2.4. Computational considerations. When solving the system of ode's (2.7) using backward differentiation formulae, the use of $\theta = 1$ in calculating $\underline{W}(t)$ and $\underline{es}(t)$ leads to the efficient procedure (Berzins [3]) defined by

$$\underline{E}(t_{n+1}) = M^{-1}(\underline{E}(t_n) + k_n \underline{TE}(t_{n+1})) + \underline{le}_{n+1}(\mathrm{TOL}). \tag{2.20}$$

If the more usual form of the local error estimate $\widehat{\underline{le}}_{n+1}(\mathrm{TOL})$ is used by the time integrator, then (2.17) can be used to rewrite (2.20) as

$$\underline{E}(t_{n+1}) = M^{-1}[\underline{E}(t_n) + \widehat{\underline{le}}_{n+1}(\mathrm{TOL}) + k_n \underline{TE}(t_{n+1})]. \tag{2.21}$$

This means that the global error indicator still incorporates the modified form of the local error estimate given by equation (2.17). Although this procedure is of zero order, it has been found to work in practice. Alternative estimates for $\underline{E}(t_{n+1})$ are given by Berzins [3]; these could be used in place of that given by (2.21).

3. Balancing the space and time errors in the method of lines. The aim in this section is to develop a time integration error control strategy, in which the accuracy tolerance is automatically selected and subsequently adjusted. In order for the method of lines to be used efficiently, the time integration error should not dominate the error due to the spatial discretisation of the pde. However, it is difficult to select a priori an ode tolerance that will ensure that this is so. This is particularly difficult if the LEPS is controlled by the integrator, since the relationship between the ode global error and the chosen accuracy tolerance is not clear. In addition, the spatial accuracy may vary with time, so any fixed tolerance used in the ode integrator is unlikely to be related to the size of the changing spatial error. Thus, an ode tolerance which is related to the spatial discretisation error in some way and which can be modified accordingly as the spatial discretisation error varies is required. Furthermore, the estimation of the spatial truncation error relies on the time integration error being dominated by the spatial discretisation error and Theorem 1 of Stetter [20] suggests that the error control strategy should be based on a form of LEPUS control. The precise form of this error control will be established by considering the relative contributions of the local time error and the spatial discretisation error to the global error in the numerical solution.

3.1. A LEPUS control strategy. The following analysis is based on the assumption that the spatial discretisation error has the form

$$\underline{es}(t) = h^p \underline{\bar{\omega}}(t) + O(h^{p+1}) \tag{3.1a}$$

and that the truncation error of the spatial discretisation method has the form

$$\underline{TE}(t) = h^p \underline{\omega}(t) + O(h^{p+1}), \tag{3.1b}$$

where $\underline{\bar{\omega}}(t)$ and $\underline{\omega}(t)$ are bounded functions that depend on the derivatives of the true solution to the pde. The modified box scheme [19] used in this paper is second order, that is $p = 2$, but the extension to other values of p is straightforward. From (2.21), it can be seen that the estimate of the global error at t_{n+1} is governed by the global error at the previous time level t_n, the spatial truncation error and the ode local error over the current timestep. Thus, it follows that for the contribution of the spatial error to dominate the contribution of the temporal error to the global error, the time local error must be controlled, in some way, with respect to $k_n \underline{TE}(t_{n+1})$. This immediately suggests that the following local error control strategy be employed in the ode integrator:

$$\|\widehat{\underline{le}}_{n+1}(\mathrm{TOL})\| < \varepsilon k_n \|\underline{TE}(t_{n+1})\|, \tag{3.2a}$$

that is,

$$\|\widehat{\underline{le}}_{n+1}(TOL)\| < \varepsilon k_n h^2 \|\underline{\omega}(t_{n+1})\|, \tag{3.2b}$$

for h sufficiently small and where $\underline{\omega}(t)$ is a bounded function. In the strategies above, $\|\cdot\|$ is a weighted maximum or averaged L_2 vector norm. In this case the fixed global LEPUS tolerance TOL is effectively given by $\varepsilon \max_t \|\underline{TE}(t)\|$ as from (3.2b) it follows that

$$\|\widehat{\underline{le}}_{n+1}(\mathrm{TOL})\| < \varepsilon k_n h^2 \max_t \|\underline{\omega}(t)\|.$$

This expression shows that the LEPUS is controlled and that the local time error is restricted to be less than a fraction ε of the spatial truncation error. It must now be shown that if the local time error is controlled in this way then the spatial discretisation error will dominate the global time error and also that the estimated truncation error $\underline{TE}_{\text{est}}(t)$ will be a suitable approximation to the true truncation error $\underline{TE}_{\text{true}}(t)$.

It is assumed, first of all, that the LEPUS is actually controlled with respect to the true spatial truncation error $\underline{TE}_{\text{true}}(t_{n+1})$, that is,

$$\|\widehat{\underline{le}}_{n+1}(\mathrm{TOL})\| < \varepsilon k_n \|\underline{TE}_{\text{true}}(t_{n+1})\|, \tag{3.3}$$

where, from (2.18)

$$\dot{\underline{es}}(t) = J\underline{es}(t) + \underline{TE}_{\text{true}}(t).$$

Since (3.3) is a LEPUS control strategy, we know from Theorem 1 of Stetter [20] that the global time error is proportional to $\varepsilon \max_t \|\underline{TE}_{\text{true}}(t)\|$, at time t, or $\varepsilon h^2 \max_t \|\underline{\omega}(t)\|$, using (3.1b). Therefore, it follows that for a suitably small value of ε, the $O(h^2)$ spatial discretisation error will dominate the $O(\varepsilon h^2)$ global time error. In practice, the effect of using the estimated spatial truncation error, rather than the exact value, when estimating the global error must be considered. Thus the error control strategy at each step is given by

$$\|\widehat{\underline{le}}_{n+1}(\mathrm{TOL})\| < \varepsilon k_n \|\underline{TE}_{\text{est}}(t_{n+1})\| \tag{3.4a}$$

and in this case, the LEPUS tolerance is given by $\mathrm{TOL} = \varepsilon \max_t \|\underline{TE}_{\text{est}}(t)\|$.

The restriction of the spatial discretisation error to the coarse mesh

$$\delta^c : a = z_1 < z_2 < \cdots < z_M = b,$$

where $z_i = x_{2i-1}$, $i = 1, \cdots, M$, $M = (N+1)/2$, and N is odd, $\underline{es}^c(t)$, can be given by (Berzins [3])

$$\begin{aligned} \dot{\underline{es}}^c(t) &= J_M \underline{es}^c(t) + \underline{TE}^c_{\text{true}}(t) \\ &= J_M \underline{es}^c(t) + \underline{TE}^c_{\text{est}}(t) + \tfrac{4}{3}(\dot{\underline{ge}}^c(t, \text{TOL}) - J_M \underline{ge}^c(t, \text{TOL})), \end{aligned}$$

which may be written as

$$\text{(3.4b)} \qquad \dot{\underline{es}}^c(t) - \tfrac{4}{3}\dot{\underline{ge}}^c(t, \text{TOL}) = J_M(\underline{es}^c(t) - \tfrac{4}{3}\underline{ge}^c(t, \text{TOL})) + \underline{TE}^c_{\text{est}}(t).$$

Now, from the local error control strategy (3.4a) and from Theorem 1 of Stetter [20], the global time error is proportional to $\text{TOL} = \varepsilon \max_t \|\underline{TE}_{\text{est}}(t)\|$, that is,

$$\|\underline{ge}(t, \text{TOL})\| = O(\text{TOL}).$$

Therefore, the restriction of $\underline{ge}(t, \text{TOL})$ to the coarse mesh δ^c, $\underline{ge}^c(t, \text{TOL})$, also satisfies

$$\|\underline{ge}^c(t, \text{TOL})\| = O(\text{TOL}).$$

Defining $\text{TOL}^* = \max_t \|\underline{TE}^c_{\text{est}}(t)\|$, where $\underline{TE}^c_{\text{est}}(t)$ is the spatial truncation error defined on the coarse mesh, which has the form, for a sufficiently small value of h,

$$\underline{TE}^c_{\text{est}}(t) = 4h^2 \underline{\omega}^*(t).$$

Hence,

$$\|\underline{ge}^c(t, \text{TOL})\| = O\left(\text{TOL}^* \varepsilon \frac{\omega_r}{4}\right),$$

where

$$\omega_r = \frac{\max_t \|\underline{\omega}(t)\|}{\max_t \|\underline{\omega}^*(t)\|}$$

and so

$$\text{(3.4c)} \qquad \left\| \frac{4}{3} \underline{ge}^c(t, \text{TOL}) \right\| = O\left(\text{TOL}^* \varepsilon \frac{\omega_r}{3}\right).$$

The solution to equation (3.4b), with initial conditions $\underline{es}^c(t_0) = \underline{ge}^c(t_0) = 0$ is given by (Coppel [8])

$$\underline{es}^c(t) - \frac{4}{3} \underline{ge}^c(t, \text{TOL}) = Y(t) \int_{t_0}^{t} Y^{-1}(s) \underline{TE}^c_{\text{est}}(s)\, ds,$$

where $Y(t)$ is the fundamental matrix (Coppel [8]) for

$$\dot{\underline{es}}^c(t) - \frac{4}{3} \dot{\underline{ge}}^c(t, \text{TOL}) = J_M \left(\underline{es}^c(t) - \frac{4}{3} \underline{ge}^c(t, \text{TOL}) \right).$$

This shows that

$$\left\| \underline{es}^c - \frac{4}{3} \underline{ge}^c(t, \text{TOL}) \right\| = O(\text{TOL}^*).$$

A comparison of this and (3.4c) shows that, because of the $\varepsilon/3$ term in (3.4c), the spatial error dominates on the coarse mesh for a suitably small value of ε. Interpolating onto the fine mesh will lead to the same result—that is, the spatial discretisation error will dominate when using the error control strategy (3.4a).

3.2. A new LEPUS control strategy. Instead of comparing the local time error simply with the spatial truncation error, the local time error can be controlled with respect to the contribution of the spatial truncation error and the existing error from previous timesteps to the global error at the end of the next timestep. The following theorem shows that by controlling the local time error so that it is a fraction ε of the growth in the global error (without the local time error) over the interval $k_n = t_{n+1} - t_n$, the global time error will be dominated by the spatial discretisation error.

THEOREM 2. *The local time error control strategy given by*

$$\|\underline{le}_{n+1}(\mathrm{TOL})\| < \varepsilon \|\underline{E}(t_{n+1}) - \underline{E}(t_n) - \underline{le}_{n+1}(\mathrm{TOL})\| \tag{3.5}$$

will, for a suitable constant ε, yield a time integration error which is dominated by the spatial discretisation error. The assumption made is that the spatial discretisation error and the spatial trunction error have the same order of accuracy.

Proof. This theorem uses that given by Stetter [20] to show that the strategy given by (3.5) controls the LEPUS with respect to a tolerance that ensures that the spatial discretisation error dominates.

From (2.18), the growth of the spatial discretisation error over the timestep $[t_n, t_{n+1}]$ is governed, approximately, by

$$\underline{es}(t_{n+1}) - \underline{es}(t_n) = \int_{t_n}^{t_{n+1}} J\underline{v}(t)\, dt + \int_{t_n}^{t_{n+1}} \underline{TE}(t)\, dt, \qquad \underline{v}(t_n) = \underline{es}(t_n)$$

and the growth of the global time error over the same interval is governed by a similar equation (see Stetter [20], Shampine [18])

$$\underline{ge}(t_{n+1}, \mathrm{TOL}) - \underline{ge}(t_n, \mathrm{TOL}) = \underline{le}_{n+1}(\mathrm{TOL}) + \int_{t_n}^{t_{n+1}} J\underline{W}(t)\, dt,$$

where $\underline{W}(t_n) = \underline{ge}(t_n, \mathrm{TOL})$. Thus, the growth of the global error in the method of lines is governed, approximately, by

$$\underline{E}(t_{n+1}) - \underline{E}(t_n) = \underline{le}_{n+1}(\mathrm{TOL}) + \int_{t_n}^{t_{n+1}} J(\underline{v}(t) + \underline{W}(t))\, dt + \int_{t_n}^{t_{n+1}} \underline{TE}(t)\, dt,$$

where $\underline{E}(t_n) = \underline{v}(t_n) + \underline{W}(t_n)$ is the global error at time t_n. Therefore, the local error control strategy (3.5) may be written as

$$\|\underline{le}_{n+1}(\mathrm{TOL})\| \leqq \varepsilon \left\| \int_{t_n}^{t_{n+1}} (J(\underline{v}(t) + \underline{W}(t)) + \underline{TE}(t))\, dt \right\|. \tag{3.6}$$

It shall now be shown, by an inductive proof, that this is a LEPUS control strategy with respect to $\mathrm{TOL} = \varepsilon h^2$. Consider the strategy (3.6) over the first timestep $(t_0, t_1]$, that is,

$$\|\underline{le}_1(\mathrm{TOL})\| \leqq \varepsilon \left\| \int_{t_0}^{t_1} (J(\underline{v}(t) + \underline{W}(t)) + \underline{TE}(t))\, dt \right\|,$$

where $\underline{TE}(t)$ is proportional to h^2 since the spatial discretisation method is second order (assuming that the spatial truncation error and spatial discretisation error have the same rate of convergence). Now $\underline{v}(t)$ is the solution of $\dot{\underline{v}}(t) = J\underline{v}(t) + \underline{TE}(t)$ and so $\underline{v}(t)$ can be given by

$$\begin{aligned} \underline{v}(t) &= Y(t)Y^{-1}(t_0)\underline{v}(t_0) + Y(t)\int_{t_0}^{t} Y^{-1}(s)\underline{TE}(s)\, ds \\ &= Y(t)\int_{t_0}^{t} Y^{-1}(s)\underline{TE}(s)\, ds, \qquad t \in (t_0, t_1], \end{aligned}$$

where $Y(t)$ is the fundamental matrix (Coppel [8]) for the equation $\dot{\underline{v}}(t)=J\underline{v}(t)$. Since $\underline{TE}(t)$ is proportional to h^2 it follows that $\underline{v}(t)$, $t\in(t_0, t_1]$, is also proportional to h^2. On the other hand, $\underline{W}(t)$ can be defined in terms of the fundamental matrix $Y(t)$ for the homogeneous linear equation (2.15), $\dot{\underline{W}}(t)=J\underline{W}(t)$ (Coppel [8])

$$\begin{aligned}\underline{W}(t)&=Y(t)Y^{-1}(t_0)\underline{W}(t_0)\\&=Y(t)Y^{-1}(t_0)\underline{ge}(t_0,\mathrm{TOL})\\&=0,\qquad t\in(t_0, t_1].\end{aligned}$$

Thus, the strategy given by (3.6) over the first timestep $(t_0, t_1]$ is equivalent to

$$\begin{aligned}\|\underline{le}_1(\mathrm{TOL})\|&\leqq\varepsilon\left\|\int_{t_0}^{t_1}\left\{JY(t)\int_{t_0}^{t}Y^{-1}(s)\underline{TE}(s)\,ds+\underline{TE}(t)\right\}dt\right\|\\&=\mathrm{TOL}\left\|\int_{t_0}^{t_1}\left\{JY(t)\int_{t_0}^{t}Y^{-1}(s)\underline{\omega}(s)\,ds+\underline{\omega}(t)\right\}dt\right\|,\end{aligned}$$

where $\mathrm{TOL}=\varepsilon h^2$ and $\underline{\omega}(t)$ is given by (3.1b). This can also be expressed as

$$\underline{le}_1(\mathrm{TOL})=\underline{v}(t_1)k_1\,\mathrm{TOL}+O(\mathrm{TOL}),\tag{3.7}$$

where

$$\underline{v}(t_1)=\frac{1}{k_1}\int_{t_0}^{t_1}\left\{JY(t)\int_{t_0}^{t}Y^{-1}(s)\underline{\omega}(s)\,ds+\underline{\omega}(t)\right\}dt,$$

and $\underline{v}(t)$ is independent of TOL and is bounded on $(t_0, t_1]$. Thus the LEPUS is controlled with respect to $\mathrm{TOL}=\varepsilon h^2$ on the first timestep, according to Stetter's theorem [20]. This means that, from (3.7), condition (2.11) is satisfied; that is,

$$\underline{ge}(t_1,\mathrm{TOL})=\underline{\bar{v}}(t_1)\mathrm{TOL}+O(\mathrm{TOL}),$$

where, in this case, $\underline{\bar{v}}(t_1)=k_n\underline{v}(t_1)$.

Now, assume that strategy (3.5) controls the LEPUS with respect to $\mathrm{TOL}=\varepsilon h^2$ for time $t=t_2,\cdots,t_n$. Therefore, at $t=t_n$,

$$\underline{ge}(t_n,\mathrm{TOL})\propto\varepsilon h^2.$$

The control strategy over the timestep $[t_n, t_{n+1}]$ must now be considered, that is,

$$\|\underline{le}_{n+1}(\mathrm{TOL})\|\leqq\varepsilon\left\|\int_{t_m}^{t_{n+1}}J(\underline{v}(t)+\underline{W}(t))+\underline{TE}(t)\,dt\right\|.$$

Again, $\underline{v}(t)$ is the solution of $\dot{\underline{v}}(t)=J\underline{v}(t)+\underline{TE}(t)$, $\underline{v}(t_n)=\underline{es}(t_n)$ and so $\underline{v}(t)$ can be given as

$$\underline{v}(t)=Y(t)Y^{-1}(t_n)\underline{es}(t_n)+Y(t)\int_{t_n}^{t}Y^{-1}(s)\underline{TE}(s)\,ds,\qquad t\in[t_n, t_{n+1}],\tag{3.8}$$

where $Y(t)$ is the fundamental matrix for the equation $\dot{\underline{v}}(t)=J\underline{v}(t)$ (Coppel [8]). On the other hand, $\underline{W}(t)$ can be defined in terms of the fundamental matrix $Y(t)$ for the homogeneous linear equation (2.15) (Coppel [8]), that is,

$$\underline{W}(t)=Y(t)Y^{-1}(t_n)\underline{ge}(t_n,\mathrm{TOL}),\qquad t\in[t_n, t_{n+1}].$$

On setting $C(t) = Y(t)Y^{-1}(t_n)$ and $\underline{ge}(t_n, \mathrm{TOL}) = \underline{\gamma}_n(t_n)\mathrm{TOL}$ where $\mathrm{TOL} = \varepsilon h^2$, strategy (3.6) over the timestep $[t_n, t_{n+1}]$ is equivalent to

$$\|\underline{le}_{n+1}(t, \mathrm{TOL})\|$$

$$(3.9)\qquad \leqq \mathrm{TOL} \left\| \int_{t_n}^{t_{n+1}} \left\{ JC(t)(\underline{\bar{\omega}}(t_n) + \varepsilon\underline{\gamma}_n(t_n)) + JY(t)\int_{t_n}^{t} Y^{-1}(s)\underline{\omega}(s)\,ds + \underline{\omega}(t) \right\} dt \right\|,$$

where $\underline{\omega}(t)$ and $\underline{\bar{\omega}}(t)$ are defined by equations (3.1b) and (3.1a), respectively. From the inductive hypothesis $\underline{ge}(t_n, \mathrm{TOL}) = O(\varepsilon h^2)$, $\underline{\gamma}_n(t_n)$ must also be bounded above. Also $\underline{ge}(t_n, \mathrm{TOL})$ must be bounded above by $\underline{es}(t_n)$ for a sufficiently small value of ε. Therefore,

$$\underline{le}_{n+1}(\mathrm{TOL}) = \underline{v}(t_{n+1})k_n\mathrm{TOL} + O(\mathrm{TOL}),$$

where

$$\underline{v}(t_{n+1}) = \frac{1}{k_n}\left(\int_{t_n}^{t_{n+1}} \left\{ JC(t)(\underline{\bar{\omega}}(t_n) + \varepsilon\underline{\gamma}_n(t_n)) + JY(t)\int_{t_n}^{t} Y^{-1}(s)\underline{\omega}(s)\,ds + \underline{TE}^*(t) \right\} dt \right)$$

and is bounded on $[t_n, t_{n+1}]$. Although $\underline{v}(t_{n+1})$ is independent of TOL, it is not independent of ε. However, for a suitably small value of ε, $\varepsilon\underline{\gamma}_n(t_n)$ is dominated by $\underline{es}(t_n)$, from the inductive hypothesis. This means that, according to Stetter's theorem [20], the LEPUS is controlled over the step $[t_n, t_{n+1}]$ and so $\underline{ge}(t_{n+1}, \mathrm{TOL}) \propto \varepsilon h^2$. By induction, it follows that this is true for all timesteps. Since the spatial discretisation error is proportional to h^2, it follows that for a suitably small value of ε, the spatial discretisation error will dominate the global time error.

3.3. An alternative approach. When estimates of the spatial discretisation error can be computed directly (for example, see Babuška and Rheinboldt [2]) it is possible to employ the control strategy (3.5) in a slightly different form. In this case, the local time error is controlled so that it is a fraction of the change in the spatial discretisation error over the current timestep. The error control strategy has the form

$$(3.10)\qquad \|\underline{le}_{n+1}(\mathrm{TOL})\| < \varepsilon\|\underline{es}(t_{n+1}) - \underline{es}(t_n)\|.$$

The equivalent form of (3.6) is given by

$$\|\underline{le}_{n+1}(\mathrm{TOL})\| \leqq \varepsilon \left\| \int_{t_n}^{t_{n+1}} J\underline{v}(t) + \underline{TE}(t)\,dt \right\|.$$

If the analysis in the theorem given above is applied to this control strategy, then the local time error can also be given by (3.9). In this case, the only difference is that $\underline{v}(t_{n+1})$ will not depend on $\gamma_n(t_n)$ and so will be independent of both ε and TOL. Using the mean value theorem, the strategy (3.10) can be written as

$$\|\underline{le}_{n+1}(\mathrm{TOL})\| < \varepsilon k \left\| \frac{\partial \underline{es}(t)}{\partial t} \right\|$$

for $t \in (t_n, t_{n+1})$. From this, it follows that

$$\|\underline{le}_{n+1}(\mathrm{TOL})\| < \varepsilon k h^2 \max_t \left\| \frac{\partial \underline{\bar{\omega}}}{\partial t} \right\|$$

and that the LEPUS tolerance, to which the global time error is proportional, is given by

$$\mathrm{TOL} = \varepsilon h^2 \max_t \left\| \frac{\partial \underline{\bar{\omega}}}{\partial t} \right\|.$$

The advantage of this form of the LEPUS error control is that it enables the spatial and temporal errors within the method of lines to be balanced by making use of *any* estimate of the change in the spatial discretisation error over a timestep. The norm used in the time integration can be chosen so as to reflect the norm in which the spatial discretisation error is estimated.

3.4. Implementation of the new strategy. In practice, the control strategy (3.5) is difficult to apply directly since the term $\underline{E}(t_{n+1})$ is known only through the global error estimator using (2.20). Instead, we can approximate the strategy (3.5), by using equation (2.20), to substitute for $\underline{E}(t_{n+1})$, thus giving

$$\|\underline{le}_{n+1}(\mathrm{TOL})\| < \varepsilon \| M^{-1}(\underline{E}(t_n) + k_n \underline{TE}(t_{n+1})) - \underline{E}(t_n)\|,$$

which, using the definition of the matrix $M = I - k_n \gamma J$, gives

$$\begin{aligned} \|\underline{le}_{n+1}(\mathrm{TOL})\| &< \varepsilon \| M^{-1}(k_n \gamma J \underline{E}(t_n) + k_n \underline{TE}(t_{n+1}))\| \\ &= k_n \varepsilon \| M^{-1}(\gamma J \underline{E}(t_n) + \underline{TE}(t_{n+1}))\|. \end{aligned} \tag{3.11}$$

The expression (3.11) shows the form of the control strategy (3.5) that is used in practice and that the strategy is of the LEPUS form. The expression also shows how the time integration accuracy varies with the spatial truncation error and the overall error. In practice, we need to modify (3.11) to cater for a zero spatial error or for a very large spatial error. This may be done by ensuring that

$$\mathrm{TOL}_{\mathrm{min}} < \| M^{-1}(\gamma J \underline{E}(t_n) + \underline{TE}(t_{n+1}))\| < \mathrm{TOL}_{\mathrm{max}},$$

where $\mathrm{TOL}_{\mathrm{min}}$ is of the order of the machine unit roundoff error and $\mathrm{TOL}_{\mathrm{max}}$ is $O(1)$.

4. Comparison with a similar error control strategy.

4.1. The error control strategy of Schönauer, Schnepf, and Raith [16]. There have been very few attempts, up to the present, to either estimate or control the global error incurred within the method of lines. One attempt is that of Schönauer, Schnepf, and Raith [16] who present variable stepsize/variable order difference methods for the solution of parabolic pde's. The error control strategy used attempts to balance the errors (spatial and temporal for parabolic pde's) according to a prescribed tolerance. An estimate of the final error is also presented.

We have already seen that when using the method of lines to solve parabolic pde's the global error is a combination of the error in the spatial discretisation method and the error due to the integration of the ode's. The assumption used by Schönauer, Schnepf, and Raith [16] is that the space mesh is chosen initially and remains fixed throughout the calculation. This means that the error control strategy presented consists of designating the spatial error to be a *key error term* to which the time error must be adapted in order to efficiently use the method of lines. Thus, the next time stepsize (for a pth-order finite-difference method in time) is given by [16] as

$$k_n = k_{n-1}[\tfrac{1}{3}\max(\mathrm{TOL}, \|\underline{TE}(t_n)\|)/\|\underline{D}_t(t_n)\|]^{1/p}, \tag{4.1}$$

where $\underline{TE}(t_n)$ is the spatial truncation error estimate and $\underline{D}_t(t_n)$ is the temporal truncation error estimate at time t_n. This is equivalent to saying that $\underline{D}_t(t)$ is adapted to either the prescribed error tolerance TOL or to the spatial truncation error $\underline{TE}(t)$ with a safety factor of $\frac{1}{3}$.

To compare this strategy with that presented in § 3.1, it is assumed that

$$\mathrm{TOL} < \|\underline{TE}(t_n)\|.$$

Therefore the strategy given by (4.1) is equivalent to

$$\|\underline{D}_t(t_n)\|\left(\frac{k_n}{k_{n-1}}\right)^p \leqq \frac{1}{3}\|\underline{TE}(t_n)\|.$$

Suppose, also, that the time truncation error estimate is of order p, that is,

$$\|\underline{D}_t(t_n)\| \approx Ck_{n-1}^p,$$

where C is a constant which is independent of k_n, and so, the strategy (4.1) aims to ensure that

$$Ck_n^p \leqq \tfrac{1}{3}\|\underline{TE}(t_{n+1})\|, \tag{4.2}$$

or, equivalently, assuming that C does not vary rapidly over the step, it seeks to control the time truncation error $\underline{D}_t(t)$ so that

$$\|\underline{D}_t(t_{n+1})\| \leqq \tfrac{1}{3}\|\underline{TE}(t_{n+1})\|. \tag{4.3}$$

This error control strategy is obviously similar to that proposed in § 3.1, (3.2), with $\varepsilon \approx \frac{1}{3}$, providing that we can establish the relationship between $\|\underline{D}_t(t_{n+1})\|$ and $\|\widehat{\underline{le}}_{n+1}(\mathrm{TOL})\|$.

For the ode defined by (2.7) with true solution $\underline{U}$, computed solution $\underline{V}$ and local solution $\underline{y}_n$, the local truncation error of the numerical method used to solve (2.7) is given by

$$\underline{T}_{n+1}(k_n, \underline{U}) = \underline{\Psi}_p(t_n, \underline{U}(t_n))k_n^{p+1} + O(k_n^{p+2}), \tag{4.4}$$

where $\underline{\Psi}_p$ is the principal error function for the pth-order method. On the other hand, the local error at t_{n+1} can also be expressed in terms of the principal error function and the local solution $y_n(t)$ (see (2.10)) as

$$\begin{aligned}\widehat{\underline{le}}_{n+1}(\mathrm{TOL}) &= \underline{\Psi}_p(t_n, \underline{y}_n(t_n))k_n^{p+1} + O(k_n^{p+2}) \\ &= \underline{T}_{n+1}(k_n, \underline{y}_n) + O(k_n^{p+2}).\end{aligned} \tag{4.5}$$

Now, $\underline{D}_t(t_n)$ is actually an estimate of the local truncation error in the time derivative $\dot{\underline{U}}$ and so for an integration method based on the backward differentiation method,

$$\underline{D}_t(t_{n+1}) \approx \frac{1}{\gamma}\underline{\Psi}_p(t_n, \underline{U}(t_n))k_n^p + O(k_n^{p+1}), \tag{4.6}$$

where γ is the leading coefficient of the bdf formula (see Shampine [17]). Since the true solution to the ivp is not available, Schönauer, Schnepf, and Raith [16] estimate $\underline{D}_t(t)$ using the computed solution, and so

$$\|\underline{D}_t(t_{n+1})\| \approx \frac{\|\widehat{\underline{le}}_{n+1}(\mathrm{TOL})\|}{\gamma k_n}. \tag{4.7}$$

Therefore the error control strategy (4.2) is equivalent to

$$\frac{\|\widehat{\underline{le}}_{n+1}(\mathrm{TOL})\|}{\gamma k_n} < \frac{1}{3}\|\underline{TE}(t_{n+1})\|, \tag{4.8}$$

which is the same as that given by (3.2), with $\varepsilon = \gamma/3$, where the values of γ, as given by Shampine [17], vary from one at order one to 60/137 for order five.

In the numerical experiments reported in § 5, it was found that the performance of the error control strategy improved when using the more reliable form of the local error estimate given by (2.17). Thus the strategy becomes

$$\frac{\|le_{n+1}(\text{TOL})\|}{\gamma k_n} < \frac{1}{3}\|\underline{TE}(t_{n+1})\|,$$

where the value $\gamma = 1$ was used in the numerical experiments.

4.2. Neumann boundary conditions for the modified box scheme [19]. The error control strategy presented by Schönauer, Schnepf, and Raith [16], represented by (4.8), controls the LEPUS with respect to the spatial truncation error. Thus, the global time error will be $O(h^p)$ as long as the spatial truncation error is $O(h^p)$. Manteuffel and White [13] show that there are many discretisation schemes for which the spatial discretisation error is of a higher order than the spatial truncation error. This analysis applies to the modified version of the box scheme (Skeel and Berzins [19]). When there are Neumann boundary conditions, the spatial truncation error is only $O(h)$, but the spatial discretisation error remains $O(h^2)$. In this case, when the LEPUS is controlled with respect to the spatial truncation error, for example (3.2) and (4.8), the global time error will be $O(h)$ and will dominate the spatial discretisation error, which is $O(h^2)$.[1]

This difficulty does not occur when controlling the local time error according to strategy (3.5), even though Theorem 2 is based upon the assumption that the spatial discretisation and truncation errors are of the same order. This assumption is used when considering the solution of (3.8). However, the solution to this equation describes the continuous growth of the spatial discretisation error, which is approximated in the discrete form by (2.18). Since the analysis of Manteuffel and White [13] can be applied to the modified box scheme [19], the solution to (3.8) is $O(h^2)$, even if the spatial truncation error is $O(h)$ at the boundaries.

The analysis of Manteuffel and White [13] does not fully describe the situation close to the initial time $t = 0$. In this case, an error transient arises from the discontinuity between the zero initial condition and nonzero Neumann boundary conditions for the error defined by (2.18). This transient results in an $O(h)$ spatial discretisation error for the first few timesteps, after which the spatial discretisation error is $O(h^2)$. Experiments have shown that the time t^*, at which the increased order of accuracy is attained, appears to be related to the spatial element size h and the time stepsize k according to

$$t^* \propto \frac{h^2}{k^{1/2}}.$$

This behaviour of the error transient also means that, close to the initial time, the estimate of the spatial truncation error underestimates the true spatial truncation error. Hence, the global error is underestimated for the first few timesteps. This causes a slight inefficiency over the first few steps as the time integration error is unduly restricted by the strategy (3.5). As far as we are aware, this particular error behaviour does not seem to have been noted elsewhere and merits further study.

5. Numerical experiments. The following five problems were used to illustrate the performance of the global error indicator and to show the effectiveness of the LEPUS

[1] Schönauer, Schnepf, and Raith [16] use a one-sided formula with an additional point at the boundaries to ensure that the spatial discretisation and spatial truncation errors are of the same order and so ensure that (4.8) can be used to balance the space and time errors.

control strategy described in § 3. To show the latter, comparisons were made with two other local error control strategies. The five test problems are shown below.

PROBLEM 1. A convection diffusion problem, known as Burgers' equation, which is defined by

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

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

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

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

PROBLEM 2. This problem was used by Berzins and Dew [4] to provide an example of a problem with a nonlinear source term and with nonlinear boundary conditions:

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

with boundary conditions

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

and

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

The initial condition is 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 travelling wave solution:

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

with Dirichlet boundary conditions and initial condition 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:

$$\frac{\partial u}{\partial t}=\frac{\partial^2 u}{\partial x^2},\qquad (x,t)\in[0,1]\times(0,0.25],$$

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

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

PROBLEM 5. The work of Lindberg [12] suggests that controlling the LEPUS when solving stiff problems is inefficient. This stiff problem was included to test the performance of the new error control strategy when solving stiff ode's:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + (\pi^2 - 1)u - pu + (p\, e^{-t} + e^{-pt}) \sin \pi x, \qquad (x, t) \in [0, 1] \times (0, 1].$$

The Dirichlet boundary conditions and initial condition are consistent with the analytic solution

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

The solution to this problem contains both a slow transient e^{-t}, and a rapid transient e^{-pt} where $p = 5000$, characteristic of stiff problems.

5.1. Testing procedure. The test problems were solved using fixed, evenly spaced, space meshes containing 41, 81, 161, and 321 points. The SPGEAR (Adams/bdf methods) module of the SPRINT software [5], with the Linpack banded matrix routines, was used to solve the ode system in time obtained after discretising the pde in space using the method presented by Skeel and Berzins [19]. Three different error control strategies were used within the time integration routines.

(A) The mixed LEPS strategy used in SPRINT [5], that is, controlling the local error $\underline{le}_{n+1}(\text{TOL})$ so that

$$\left\| \frac{\underline{le}_{n+1}(\text{TOL})}{(\text{RTOL} \cdot |\underline{Y}(t)| + \text{ATOL})} \right\| < 1. \tag{5.1}$$

(B) The LEPUS strategy presented in § 3, that is, controlling the local error $\underline{le}_{n+1}(\text{TOL})$ so that

$$\|\underline{le}_{n+1}(\text{TOL})\| \leqq k_n \varepsilon \| M^{-1}(\gamma J \underline{E}(t_n) + \underline{TE}(t_{n+1}))\|. \tag{5.2}$$

(C) The LEPUS strategy presented by Schönauer, Schnepf, and Raith [16], that is, controlling the local error $\underline{le}_{n+1}(\text{TOL})$ so that

$$\|\underline{le}_{n+1}(\text{TOL})\| \leqq \tfrac{1}{3} k_n \|\underline{TE}(t_{n+1})\|. \tag{5.3}$$

The use of the error control strategies (5.2) and (5.3) required a slightly modified version of the ode integrator.

The performance of the global error estimate can be measured by defining the error index

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

The error index was calculated at the end of every timestep in the integration. More details about the performance of the error indicator can be found in Berzins [3].

Before comparing the performance of the three local error control strategies, the choice of the parameter ε in strategy (5.2) must be considered. Section 3 suggests that this parameter be chosen in order that the spatial discretisation error dominates the time integration error and that this will be so for some value of $\varepsilon < 1$. A second aim is to choose ε in order that the work needed, when employing this error control strategy, be kept to a minimum. Obviously, a large value for ε will require fewer ode timesteps

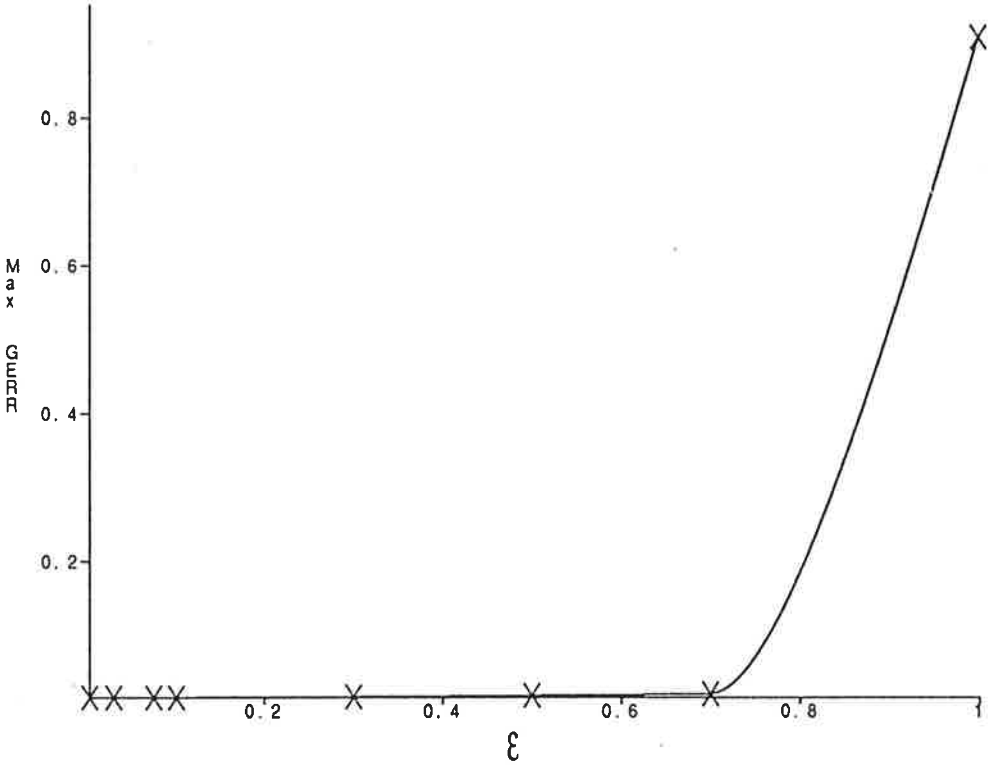

FIG. 1. *Problem* 1. *Graph showing relationship between* Max GERR *and ε.*

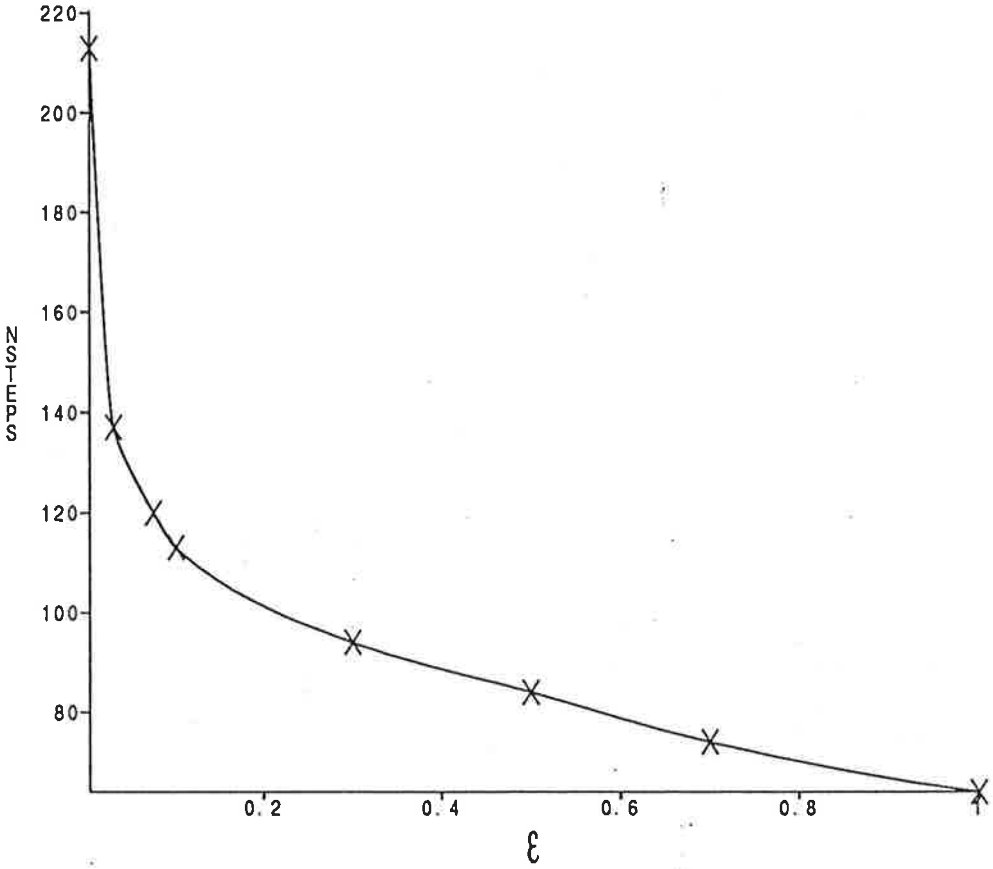

FIG. 2. *Problem* 1. *Graph showing relationship between* NSTEPS *and ε.*

TABLE 1
Results for Problem 1.

		TOL or	Max GERR at time					
METHOD	NPTS	ε	$T=0.01$	$T=0.56$	$T=1.0$	CPU	NSTEPS	AVG
A	41	0.5D−3	0.49D−2	0.71D−1	0.12D00	3	75	0.49
		0.1D−3	0.48D−2	0.71D−1	0.12D00	4	99	0.51
		0.5D−4	0.48D−2	0.71D−1	0.12D00	5	121	0.53
B	41	0.3	0.51D−2	0.73D−1	0.12D00	4	62	0.51
A	81	0.1D−3	0.13D−2	0.21D−1	0.25D−1	7	125	0.61
		0.5D−4	0.12D−2	0.21D−1	0.25D−1	9	144	0.65
		0.1D−4	0.12D−2	0.21D−1	0.25D−1	12	205	0.69
B	81	0.3	0.13D−2	0.21D−1	0.38D−1	8	94	0.61
A	161	0.5D−4	0.37D−3	0.54D−2	0.91D−2	16	155	0.68
		0.1D−4	0.35D−3	0.53D−2	0.91D−2	20	196	0.72
		0.5D−5	0.35D−3	0.53D−2	0.91D−2	23	217	0.74
B	161	0.3	0.35D−3	0.55D−2	0.92D−2	19	140	0.69
A	321	0.1D−4	0.88D−4	0.13D−2	0.22D−2	47	211	0.77
		0.5D−5	0.87D−4	0.13D−2	0.22D−2	53	241	0.81
		0.1D−5	0.87D−4	0.13D−2	0.22D−2	58	266	0.79
B	321	0.3	0.87D−4	0.13D−2	0.22D−2	47	185	0.73

but will lead to a larger time integration error, while a small value for ε will yield a smaller time integration error but more ode timesteps will be required. Figures 1 and 2 show, respectively, the maximum global errors incurred and the number of ode timesteps required when using values of ε in the range of 0.001 for 1.0 in solving Problem 1, Burgers' equation. These results indicate that we should set $\varepsilon \approx 0.3$, since for $\varepsilon > 0.3$ the maximum global errors increase sharply, while for $\varepsilon < 0.3$ the number of ode timesteps required is rather large. This value happens to be approximately the same as the "safety factor" used by Schönauer, Schnepf, and Raith [16], in the third error control strategy (5.3).

For the comparison of the different strategies, it is necessary to obtain solutions for which the spatial discretisation error dominates. In the case of the LEPS strategy (5.1), the global time error is not directly related to the LEPS tolerance and the spatial discretisation error domination can only be achieved by experimentation with a variety of different tolerances. To show that the results quoted are the most efficient with the spatial error dominating, Tables 1, 2(a), 3, 4(a), and 5 show the results obtained when using the LEPS control strategy (5.1) with the best accuracy tolerance in terms of efficiency, as well as with smaller and larger accuracy tolerances. The CPU *time* quoted, however, does not reflect the experimentation needed to actually find the best tolerance.

Key to Tables 1 to 4.
NPTS is the number of points used in the spatial mesh.
TOL is the value given to ATOL and RTOL used in the strategy (5.1).
Max GERR is the maximum grid error found at the specified output times.
CPU is the amount of CPU time used, measured in seconds.

TABLE 2(a)
Results for Problem 2.

METHOD	NPTS	TOL or ε	Max GERR at time			CPU	NSTEPS	AVG
			$T = 0.01$	$T = 0.89$	$T = 2.0$			
A	41	0.1D−4	0.23D−3	0.29D−3	0.76D−2	2.8	71	1.2
		0.5D−5	0.23D−3	0.28D−3	0.76D−2	3.2	83	1.2
		0.1D−5	0.23D−3	0.27D−3	0.76D−2	3.7	99	1.3
B	41	0.3	0.21D−3	0.27D−3	0.77D−2	3.3	56	1.3
A	81	0.5D−5	0.55D−4	0.76D−4	0.20D−2	4.4	75	1.0
		0.1D−5	0.58D−4	0.69D−4	0.19D−2	5.4	94	1.3
		0.5D−6	0.58D−4	0.68D−4	0.19D−2	6.4	110	1.3
B	81	0.3	0.52D−4	0.66D−4	0.20D−2	6.2	76	1.3
A	161	0.1D−5	0.14D−4	0.18D−4	0.49D−3	8.8	85	1.2
		0.5D−6	0.14D−4	0.17D−4	0.49D−3	10.4	99	1.2
		0.1D−6	0.14D−4	0.18D−4	0.50D−3	13.3	132	1.4
B	161	0.3	0.13D−4	0.17D−4	0.50D−3	10.3	80	1.4
A	321	0.5D−6	0.38D−5	0.33D−5	0.12D−3	20.8	94	1.1
		0.1D−6	0.35D−5	0.45D−5	0.12D−3	26.1	122	1.3
		0.5D−7	0.35D−5	0.44D−5	0.12D−3	27.2	127	1.5
B	321	0.3	0.34D−5	0.43D−5	0.13D−3	26.0	102	1.5

TABLE 2(b)
Results for Problem 2.

METHOD	NPTS	Max GERR at Time			CPU	NSTEPS	AVG
		$T = 0.01$	$T = 0.89$	$T = 2.0$			
C	41	0.20D−3	0.26D−2	0.76D−2	1.9	30	1.4
C	81	0.49D−4	0.10D−2	0.82D−2	3.4	40	1.8
C	161	0.13D−4	0.59D−3	0.79D−2	5.3	37	1.2
C	321	0.26D−5	0.13D−1	0.69D−2	12.8	48	1.6

NSTEPS is the number of timesteps used in the integration of the odes.
AVG is the average value of the error index sampled at the end of every timestep.
Method A is the strategy (5.1).
Method B is the strategy (5.2).
Method C is the strategy (5.3)
The results obtained when using method C are only shown where they differ significantly from the results obtained using method B.

5.2. Discussion of the numerical results. A comparison of the results shown in Tables 1 and 3 shows that when solving Problems 1 and 3, all three error control strategies yield solutions of comparable accuracy. The efficiency of each computation can be compared by observing the CPU times and the number of ode timesteps used. In these cases, it appears that controlling the LEPUS by using either (5.2) or (5.3) is

TABLE 3
Results for Problem 3.

METHOD	NPTS	TOL or ε	Max GERR at time			CPU	NSTEPS	AVG
			$T=0.01$	$T=0.56$	$T=1.0$			
A	41	0.1D−5	0.38D−5	0.11D−3	0.17D−3	0.8	19	1.0
		0.5D−6	0.37D−3	0.11D−3	0.17D−3	0.9	21	1.0
		0.1D−6	0.35D−5	0.11D−3	0.17D−3	1.1	27	1.0
B	41	0.3	0.35D−5	0.11D−3	0.17D−3	1.5	25	1.0
A	81	0.5D−6	0.13D−5	0.29D−4	0.43D−4	1.2	20	1.1
		0.1D−6	0.94D−6	0.29D−4	0.43D−4	1.6	26	1.1
		0.5D−7	0.91D−6	0.29D−4	0.43D−4	1.7	29	1.1
B	81	0.3	0.87D−6	0.29D−4	0.42D−4	2.0	25	1.1
A	161	0.1D−6	0.30D−6	0.72D−5	0.11D−4	2.6	24	1.1
		0.5D−7	0.26D−6	0.72D−5	0.11D−4	2.7	26	1.1
		0.1D−7	0.23D−6	0.72D−5	0.11D−4	3.8	33	1.1
B	161	0.3	0.22D−6	0.71D−5	0.11D−4	3.6	29	1.1
A	321	0.5D−7	0.12D−6	0.18D−5	0.27D−5	6.1	28	1.2
		0.1D−7	0.63D−7	0.18D−5	0.27D−5	7.1	33	1.1
		0.5D−8	0.58D−7	0.18D−5	0.27D−5	7.5	35	1.1
B	321	0.3	0.55D−7	0.18D−5	0.26D−5	7.9	33	1.1

as efficient as controlling the LEPS (5.1). However, in the case of the stiff problem, Problem 5, the LEPUS strategies have failed to ensure that the spatial discretisation error dominates the time integration error. The results in Table 5 show that, for this problem, the global error estimate is larger than the actual global error. This means that the tolerance used in the LEPUS strategies is too large to ensure that the spatial discretisation error dominates. One solution to this problem is to allow the parameter ε in (5.2) to vary adaptively. Such a strategy has been presented by Lawson [22], although both this and the global error estimating algorithm require more work.

The results in Tables 2(a) and 4(a) show that by either controlling the LEPS, (5.1), or the LEPUS according to (5.2) leads to solutions of comparable accuracy. However, the results in Tables 2(b) and 4(b) show that the LEPUS strategy given by (5.3) has failed to control the local time error sufficiently so that the spatial discretisation error dominates. The reason for this (see § 4) is that Problems 2 and 4 have Neumann boundary conditions, which means that the spatial truncation error for the modified box scheme [19] is $O(h)$, whereas the spatial discretisation error is $O(h^2)$. This means that when controlling the local time error using (5.3), the global time error will also be $O(h)$. Since, however, the spatial discretisation error is $O(h^2)$, the global time error dominates. This case apart, the results show that the LEPUS strategies, in particular (5.2), yield *automatically*, solutions which are generally of the same order of accuracy as those obtained when controlling the LEPS with tolerances chosen in order that the spatial discretisation error dominates (as it does for the LEPUS strategies). The user no longer has to experiment with different accuracy tolerances to find the solution for which the spatial error is dominant.

TABLE 4(a)
Results for Problem 4.

		TOL	Max GERR at time					
METHOD	NPTS	or ε	$T=0.01$	$T=0.11$	$T=0.25$	CPU	NSTEPS	AVG
A	41	0.1D−4	0.33D−3	0.47D−3	0.39D−3	1.6	39	1.3
		0.5D−5	0.33D−3	0.46D−3	0.36D−3	1.8	43	1.3
		0.1D−5	0.33D−3	0.46D−3	0.36D−3	2.2	54	1.3
B	41	0.3	0.27D−3	0.34D−3	0.25D−3	2.0	32	1.5
A	81	0.5D−5	0.83D−4	0.12D−3	0.10D−3	2.4	40	1.3
		0.1D−5	0.83D−4	0.12D−3	0.91D−4	3.1	51	1.3
		0.5D−6	0.83D−4	0.12D−3	0.91D−4	3.3	55	1.3
B	81	0.3	0.69D−4	0.88D−4	0.63D−4	3.6	41	1.5
A	161	0.1D−5	0.21D−4	0.29D−4	0.23D−4	4.7	44	1.3
		0.5D−6	0.21D−4	0.29D−4	0.22D−4	5.4	51	1.3
		0.1D−6	0.21D−4	0.29D−4	0.22D−4	6.7	65	1.4
B	161	0.3	0.18D−4	0.23D−4	0.16D−4	6.6	47	1.5
A	321	0.5D−6	0.47D−5	0.67D−5	0.41D−5	10.1	45	1.4
		0.1D−6	0.51D−5	0.69D−5	0.51D−5	12.2	56	1.4
		0.5D−7	0.51D−5	0.69D−5	0.51D−5	14.5	66	1.4
B	321	0.3	0.45D−5	0.59D−5	0.41D−5	14.4	55	1.5

TABLE 4(b)
Results for Problem 4.

		Max GERR at time					
METHOD	NPTS	$T=0.01$	$T=0.11$	$T=0.25$	CPU	NSTEPS	AVG
C	41	0.35D−3	0.79D−3	0.29D−3	1.3	18	1.3
C	81	0.83D−4	0.51D−3	0.14D−3	2.0	21	1.4
C	161	0.18D−4	0.17D−3	0.14D−4	3.8	25	2.0
C	321	0.58D−5	0.30D−4	0.38D−4	7.8	27	1.1

6. Conclusions and further developments. Our aim is to develop a fully automatic general-purpose algorithm for the numerical solution of parabolic equations using the method of lines. From practical experience, the local time error control strategy introduced in § 3, equation (3.5), appears to provide a promising starting point for the development of such an algorithm. This control strategy aims to maximise the efficiency of the method of lines by attempting to balance the spatial and temporal errors, although the spatial discretisation error is allowed to dominate in order that the estimation of the spatial truncation error remains valid. By computing the LEPUS accuracy tolerance at each timestep, not only have we enabled the error in the time integration to vary in relation to the spatial discretisation error, we have ensured that the method of lines is being used efficiently. The results for the stiff problem, Problem 5, indicate that although the LEPUS strategy (3.5) is automatic, it may still be more

TABLE 5
Results for Problem 5.

METHOD	NPTS	TOL or ε	Max GERR at time			CPU	NSTEPS	AVG
			$T=0.01$	$T=0.56$	$T=1.0$			
A A A	41	0.1D−4 0.5D−5 0.1D−5	0.76D−3 0.76D−3 0.76D−3	0.51D−3 0.40D−3 0.44D−3	0.12D−3 0.22D−3 0.29D−3	7 8 24	73 78 139	1.7 1.7 1.4
B	41	0.3	0.76D−3	0.15D−3	0.11D−3	9	86	1.7
A A A	81	0.5D−5 0.1D−5 0.5D−6	0.19D−3 0.19D−3 0.19D−3	0.67D−4 0.10D−3 0.11D−3	0.10D−4 0.79D−4 0.69D−4	12 39 15	78 139 102	1.7 1.4 1.8
B	81	0.3	0.19D−3	0.17D−3	0.11D−3	18	120	2.0
A A A	161	0.1D−5 0.5D−6 0.1D−6	0.48D−4 0.48D−4 0.48D−4	0.21D−4 0.27D−4 0.27D−4	0.25D−4 0.16D−4 0.17D−4	69 26 32	139 102 123	1.4 1.8 1.8
B	161	0.3	0.48D−4	0.32D−4	0.28D−4	39	145	1.9
A A A	321	0.5D−6 0.1D−6 0.5D−7	0.12D−4 0.12D−4 0.12D−4	0.59D−5 0.65D−5 0.67D−5	0.28D−5 0.42D−5 0.42D−5	51 63 73	102 123 140	1.8 1.8 1.8
B	321	0.3	0.12D−4	0.35D−4	0.84D−5	73	153	2.5

efficient to control the LEPS if a suitable value for the tolerance is known a priori. The situation in the method of lines is different from standard ode systems principally because there is an error already present from the spatial discretisation of the pde. The error control approach we have taken reflects this, in contrast to standard local error control where the tolerance is supplied by the user and experimentation is needed to balance the spatial and temporal errors.

Finally, an important area of research lies in the combination of the error control strategy (3.5) with the use of adaptive or moving mesh algorithms. The use of mesh modification techniques becomes especially important if the nature of the solution changes as time elapses. One or both of two approaches is usually adopted; the meshpoints move continuously with the computed solution (e.g., [14]), or the mesh is only adapted at certain discrete times during the computation (e.g., [5], [6]). Another approach is to combine these two ideas and so we have a moving mesh scheme with local refinement (e.g., [1], [15]). All of these methods attempt to place the meshpoints to follow the changing nature of the computed solution, thus attempting to reduce the errors incurred due to the spatial discretisation method. The aim to balance the spatial and temporal errors in the method of lines relates directly to the mesh modification techniques which seek to control the spatial discretisation error. The automatic algorithm developed by Lawson and Berzins ([21] and [22]) uses the pattern recognition technique presented by Bieterman and Babuška [6] to modify the spatial mesh at discrete times in order that the spatial discretisation error be controlled with respect to some tolerance. This means that the user has only to supply the definition of the problem to be solved, an initial coarse mesh, and an error tolerance for the energy norm of the spatial discretisation error. The algorithm controls the spatial discretisation error according to the tolerance, by suitably modifying the spatial mesh, and then ensures efficient use of the method of lines by using the strategy (3.5) in the time integrator.

Acknowledgment. The helpful comments made by the referees are appreciated.

REFERENCES

[1] S. Adjerid and J. E. Flaherty, *A moving finite element method with error estimation and refinement for one-dimensional time dependent partial equations*, SIAM J. Numer. Anal., 23 (1986), pp. 778-796.

[2] I. Babuška and W. C. Rheinboldt, *A posteriori error estimates for the finite element method*, Internat. J. Numer. Methods Engrg., 12 (1978), pp. 1597-1615.

[3] M. Berzins, *Global error estimation in the method of lines for parabolic equations*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 687-703.

[4] M. Berzins and P. M. Dew, *A note on C^0 Chebyshev methods for parabolic* P.D.E.'s, IMA J. Numer. Anal., 7 (1987), pp. 15-37.

[5] M. Berzins, P. M. Dew, and R. M. Furzeland, *Developing* P.D.E. *software using the method of lines and differential algebraic integrators*, Appl. Numer. Math., 5 (1989), pp. 375-397.

[6] M. Bieterman and I. Babuška, *An adaptive method of lines with error control for parabolic equations of the reaction-diffusion type*, J. Comput. Phys., 63 (1986), pp. 33-66.

[7] 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 Engrg., 20 (1984), pp. 1797-1813.

[8] W. A. Coppel, *Stability and Asymptotic Behaviour of Differential Equations*, Heath, Boston, 1965.

[9] A. C. Hindmarsh, ODEPACK—*a systematized collection of* O.D.E. *solvers*, in Advances in Computer Methods IV, R. Vichnevetsky and R. S. Stepleman, eds., IMACS, New Brunswick, CT, 1983.

[10] T. R. Hopkins, *Numerical solution of quasi-linear parabolic differential equations*, Ph.D. thesis, University of Liverpool, Liverpool, U.K., 1976; Numer. Math., 49 (1986), pp. 659-683.

[11] H. B. Keller, *A new difference scheme for parabolic problems*, in Numerical Solution of P.D.E.'s II, B. Hubbard, ed., SYNSPADE, Academic Press, London, 1970.

[12] B. Lindberg, *Characterisation of optimal stepsize sequences for methods for stiff differential equations*, SIAM J. Numer. Anal., 14 (1977), pp. 859-887.

[13] T. A. Manteuffel and A. B. White, Jr., *The numerical solution of* 2*nd-order* B.V.P.*s on non-uniform meshes*, Math. Comp., 47 (1986), pp. 511-535.

[14] K. Miller and R. N. Miller, *Moving finite elements*. I, SIAM J. Numer. Anal., 18 (1981), pp. 1019-1032.

[15] L. R. Petzold, *Observations on an adaptive moving grid method for* 1*-dimensional systems of partial differential equations*, Appl. Numer. Math., 3 (1987), pp. 347-360.

[16] W. Schönauer, E. Schnepf, and K. Raith, *Experiences in designing* P.D.E. *software with self adaptive variable step size/order difference methods*, Computing, 5 (1984), pp. 227-242.

[17] L. F. Shampine, *Global error estimation for stiff* O.D.E.'*s*, in Proc. Dundee Conference on Numerical Analysis, Lecture Notes in Mathematics, 1066, Springer-Verlag, New York, 1984.

[18] ———, *Tolerance proportionality in* O.D.E. *codes*, SMU Report 87-8, Department of Mathematics, Southern Methodist University, Dallas, TX, 1987.

[19] R. D. Skeel and M. Berzins, *A method for the spatial discretization of parabolic equations in one space variable*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 1-32.

[20] H. J. Stetter, *Considerations concerning a theory for* O.D.E. *solvers*, in Numerical Treatment of Differential Equations, R. Bulirsch, R. D. Grigorieff, and J. Schroder, eds., Lecture Notes in Mathematics 631, Springer-Verlag, New York, 1978, pp. 188-200.

[21] J. Lawson and M. Berzins, *Towards an automatic algorithm for the numerical solution of parabolic partial differential equations*, Proc. IMA Conference on Computational O.D.E.s, London, U.K., 1989, to appear.

[22] J. Lawson, *Towards error control for the numerical solution of parabolic equations*, Ph.D. thesis, School of Computer Studies, University of Leeds, Leeds, U.K., 1989.