Journal of Computational Physics **167,** 372–392 (2001)
doi:10.1006/jcph.2000.6679, available online at http://www.idealibrary.com on IDEAL®

On The Numerical Solution of One-Dimensional PDEs Using Adaptive Methods Based on Equidistribution

G. Beckett,[1] J. A. Mackenzie, A. Ramage, and D. M. Sloan

Department of Mathematics, University of Strathclyde, Livingstone Tower, 26 Richmond Street, Glasgow G1 1XH, Scotland
E-mail: d.sloan@strath.ac.uk

Received March 21, 2000; revised November 13, 2000

Numerical experiments are described that illustrate some important features of the performance of moving mesh methods for solving one-dimensional partial differential equations (PDEs). The particular method considered here is an adaptive finite difference method based on the equidistribution of a monitor function and it is one of the moving mesh methods proposed by W. Huang, Y. Ren, and R. D. Russell (1994, *SIAM J. Numer. Anal.* **31** 709). We show how the accuracy of the computations is strongly dependent on the choice of monitor function, and we present a monitor function that yields an optimal rate of convergence. Motivated by efficiency considerations for problems in two or more space dimensions, we demonstrate a robust and efficient algorithm in which the mesh equations are uncoupled from the physical PDE. The accuracy and efficiency of the various formulations of the algorithm are considered and a novel automatic time-step control mechanism is integrated into the scheme.

Key Words: adaptivity; equidistribution; moving meshes.

1. INTRODUCTION

Many evolutionary problems involving linear or nonlinear partial differential equations (PDEs) have solutions with sharp transitions such as boundary layers or steep wave fronts. Over the past decade it has generally been accepted, at least for problems in one space dimension, that adaptive or moving mesh methods are capable of resolving the sharp transitions to acceptable degrees of accuracy without using an excessive number of mesh points.

[1] Supported by EPSRC Grant GR/M26459.

0021-9991/01 $35.00

The methods, by continuously relocating the mesh points to follow features of the computed solution, provide an ideal adaptive strategy for solving problems of this type. Moving mesh methods use nonuniform spatial meshes and, as time evolves, they concentrate the grid in spatial regions of high activity. A useful approach in adaptive schemes is the concept of equidistribution, which seeks to distribute some function (referred to as the monitor function) uniformly over the domain of the problem. The adaptive methods considered in this paper are based on equidistribution, and the monitor function is taken to be a measure of the local solution variation.

There has been an extensive study of moving mesh methods and applications thereof in the one-dimensional case (see, for example, [7, 9–11, 18] and references therein). In two space dimensions, several methods have been developed to determine mesh movement. For example, the moving finite element method of Miller and Miller [17] determines the mesh movement by minimizing the residual for the governing PDEs. Recently, Huang and Russell [12, 13] have developed a moving mesh strategy based on the solution of a system of moving mesh PDEs that is derived from the gradient flow equation of a carefully designed functional. This functional takes account of the key objectives, and it contains terms that deal with mesh adaptation, quality control, and smoothness.

No convergence analysis has been produced for moving mesh methods, and insight into the behavior of the methods has to be obtained by means of numerical experiments. In the course of conducting experiments on problems in one and two space dimensions it became apparent to us that, even in the one-dimensional case, many aspects of the behavior of moving mesh methods had still not been elucidated. The motivation for the work presented in this paper stems from the observation that further important features of the performance of one-dimensional moving mesh methods can be illustrated using simple numerical experiments.

The paper describes some results on the numerical solution of the one-dimensional viscous Burgers' equation using one of the moving mesh methods proposed by Huang *et al*. [9]. Section 2 presents the differential problem on which the numerical experiments are performed, together with the moving mesh partial differential equation (MMPDE) that is used to generate the grid. This section also deals with the discretization of the PDEs and aspects of the adaptive process such as the choices of monitor function, smoothing processes, and algorithms for integration over time. Section 3 presents an analysis of the errors incurred when the initial function is approximated by its piecewise linear interpolants on nonuniform initial grids that are generated by equidistribution of each of the monitor functions used in this investigation. This analysis provides reliable predictions for how the accuracy of the computations on adaptive meshes depends on the choice of monitor function. It deals also with variations in accuracy and efficiency of the moving mesh approach brought about by variations in the method of implementation. In Section 3, the moving mesh calculations are performed using a fixed time step.

The important property of time-step control is considered in Section 4. High computational efficiency can be achieved only if time-step control methods are used that suit the special features of these methods. To this end, we present a robust and efficient time-step control mechanism that makes use of error indicators for the accuracy of the grid and the accuracy of the solution of the physical PDE. The problem of time-step control has been examined by Verwer and Blom [20], and more recently by Cao, Huang, and Russell [6], and Huang [8].

Section 5 contains conclusions and comments on our numerical investigations.

2. TEST PROBLEM AND ADAPTIVE SCHEME

2.1. *Test Problem*

All numerical computations described in the paper were performed on the one-dimensional Burgers' equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \varepsilon\frac{\partial^2 u}{\partial x^2} = 0, \quad x \in \Omega_p := (0, 1), \quad t \in (0, T], \tag{2.1}$$

subject to initial and boundary conditions taken from the exact solution

$$u(x, t) = c - \frac{1}{2}\tanh\left[\frac{1}{4\varepsilon}(x - ct - x_0)\right], \tag{2.2}$$

where $c = 1/2$ and ε is a constant that satisfies $0 < \varepsilon \ll 1$. The solution describes a traveling front joining an upstream state $u = 1$ and a downstream state $u = 0$. The front moves with velocity c and is initially at location $x = x_0$. In the computations we use the initial location $x_0 = 1/4$. This test problem has been used in computational experiments by Blom *et al*. [5] and by Mulholland *et al*. [18].

2.2. *Adaptive Moving Mesh Method*

Here we give a brief outline of the moving mesh method that is used to generate the time-dependent grid and the approximate solution of (2.1). Further details may be found in the papers by Huang *et al*. [9] and Mulholland *et al*. [18]. Equation (2.1) is first recast in terms of the independent variables ξ and t, where ξ is defined by a one-to-one coordinate transformation of the form

$$x = x(\xi, t), \quad \xi \in \Omega_c = (0, 1), \quad t \in (0, T], \tag{2.3}$$

from computational space $\Omega_c \times (0, T]$ to physical space $\Omega_p \times (0, T]$. At time t, the map (2.3) defines a set of nodes in Ω_p that corresponds to a uniform mesh on Ω_c. This uniform mesh is given by

$$\xi_i = i/N, \quad i = 0, 1, \ldots, N, \tag{2.4}$$

and the related mesh on Ω_p is the grid

$$\Delta_N := \{0 = x_0(t) < x_1(t) < \cdots < x_N(t) = 1\}, \tag{2.5}$$

where

$$x_i(t) = x(\xi_i, t), \quad i = 0, 1, \ldots, N. \tag{2.6}$$

It is convenient to express the time derivative in (2.1) in Lagrangian form [9] and we therefore write the equation as

$$\dot{u} - \dot{x}\frac{\partial u}{\partial x} + u\frac{\partial u}{\partial x} - \varepsilon\frac{\partial^2 u}{\partial x^2} = 0, \tag{2.7}$$

where $\dot{u}$ and $\dot{x}$ denote derivatives with respect to t in which ξ is held constant. In terms of the independent variables ξ and t, Eq. (2.7) becomes

$$x_\xi \dot{v} - \frac{\partial v}{\partial \xi}\dot{x} + \frac{1}{2}\frac{\partial}{\partial \xi}(v^2) - \varepsilon \frac{\partial}{\partial \xi}\left(\frac{1}{x_\xi}\frac{\partial v}{\partial \xi}\right) = 0, \tag{2.8}$$

where

$$x_\xi \equiv \frac{\partial x}{\partial \xi} \quad \text{and} \quad v(\xi, t) \equiv u(x(\xi, t), t).$$

In the moving mesh method a mesh generating equation, based on equidistribution of a monitor function, is combined with (2.8) to give a system of equations that determines the time evolution of $x(\xi, t)$ and $v(\xi, t)$. Here we use a moving mesh PDE that is the one-dimensional analogue of one of the two-dimensional methods proposed by Huang and Russell [13]. The map (2.3) is generated as the solution of the PDE

$$\begin{gathered} \frac{\partial x}{\partial t} = \frac{1}{\tau}\left(M\frac{\partial x}{\partial \xi}\right)^{-2}\frac{\partial}{\partial \xi}\left(M\frac{\partial x}{\partial \xi}\right), \quad \xi \in \Omega_c, \\ x(0, t) = 0, \quad x(1, t) = 1, \end{gathered} \tag{2.9}$$

where τ is a positive constant known as the temporal smoothing parameter and $M(u(x, t))$ is the monitor function. Factors that influence the choice of τ are discussed by Huang [8]. The initial condition, $x(\xi, 0)$, is obtained by equidistribution of a monitor function based on the exact solution (2.2) at $t = 0$, and details of this are given where the algorithm is presented in Section 2.3.

In this paper, computations are performed using two distinct monitor functions. The first of these is the popular arc-length monitor function (henceforth referred to as the AL monitor function)

$$M(u(x, t)) = \sqrt{\gamma + \left(\frac{\partial u}{\partial x}(x, t)\right)^2}, \tag{2.10}$$

where γ is a user-prescribed parameter. The second monitor function is a modification of one that has been used to great effect by Beckett and Mackenzie [2, 3] for steady reaction–diffusion and convection–diffusion problems. This function—henceforth referred to as the BM monitor function—has the form

$$M(u(x, t)) = \alpha + \left|\frac{\partial u}{\partial x}(x, t)\right|^{\frac{1}{m}}, \tag{2.11}$$

where α and m are positive constants. In the calculations involving the BM monitor function we use the value $m = 2$, which is suggested by the analysis presented by Beckett and Mackenzie [2, 3]. They use the second spatial derivative of u in (2.11), and they show how α may be chosen in terms of m to control the proportion of grid points located in steep layers. For the evolutionary problem considered here we follow Beckett and Mackenzie [2, 3] and define $\alpha = \alpha(t)$ in (2.11) by

$$\alpha = \int_0^1 \left|\frac{\partial u}{\partial x}(x, t)\right|^{\frac{1}{m}} dx. \tag{2.12}$$

Since equidistribution of M on [0, 1] is governed at time t by the condition [6]

$$\int_0^{x(\xi,t)} M(u(s,t))\,ds = \xi \int_0^1 M(u(x,t))\,dx, \tag{2.13}$$

it follows that $\alpha = \alpha(t)$ is given by

$$\int_0^{x(\xi,t)} M(u(s,t))\,ds = 2\alpha\xi. \tag{2.14}$$

If we assume that $|\frac{\partial u}{\partial x}|$ is negligible outside a steep layer then it follows from (2.11) and (2.14) that an element of length $dx \subset \Omega_p$ in a smooth region of flow is related to an element of length $d\xi \subset \Omega_c$ by

$$dx \sim 2d\xi.$$

This shows that for the BM monitor function, with α given by (2.12), approximately half of the grid points will be located outside the steep layers. The grid has some similarities to Shishkin grids that have been used extensively to solve steady singular perturbation problems (see, for example, [19]). A nodal distribution of this type is generated by the equidistribution of the AL monitor function (2.10) when γ is set equal to unity. This value is used throughout the paper and the AL monitor function is henceforth given by (2.10) with $\gamma = 1$.

2.3. *Discretization of (2.8) and (2.9)*

We seek approximations to the time-dependent vectors $\{x_i\}_{i=0}^N$ and $\{v_i\}_{i=0}^N$, where

$$v_i = v_i(t) = v(\xi_i, t) = u(x(\xi_i, t), t) \tag{2.15}$$

and x_i is given by (2.6). Equation (2.8) is discretized on the equispaced grid (2.4) to give

$$\begin{aligned}(x_{i+1} - x_{i-1})\dot{v}_i - (v_{i+1} - v_{i-1})\dot{x}_i + \frac{1}{2}((v_{i+1})^2 - (v_{i-1})^2)\\ - 2\varepsilon\left(\frac{v_{i+1} - v_i}{x_{i+1} - x_i} - \frac{v_i - v_{i-1}}{x_i - x_{i-1}}\right) = 0\end{aligned} \tag{2.16}$$

for $i = 1, 2, \ldots, N-1$, with v_0 and v_N given by the exact solution (2.2) at $x = 0$ and $x = 1$, respectively. This discretization is second-order accurate on an even grid. The discretization of (2.9) requires the evaluation of the monitor function M at $(\xi_{i+\frac{1}{2}}, t)$, and to this end we approximate $\frac{\partial u}{\partial x}$ at $(x(\xi_{i+\frac{1}{2}}, t), t)$ by

$$g_{i+\frac{1}{2}} = \left(\frac{v_{i+1} - v_i}{x_{i+1} - x_i}\right) \tag{2.17}$$

for $i = 0, 1, \ldots, N-1$. For the BM monitor function α is given by a quadrature approximation in (2.12) as

$$\alpha = \sum_{i=0}^{N-1} \left|g_{i+\frac{1}{2}}\right|^{\frac{1}{m}} (x_{i+1} - x_i). \tag{2.18}$$

The approximations to the monitor functions (2.10) and (2.11) at $(x(\xi_{i+\frac{1}{2}}, t), t)$ are given, respectively, by

$$M_{i+\frac{1}{2}} = \sqrt{1 + g^2_{i+\frac{1}{2}}} \tag{2.19}$$

and

$$M_{i+\frac{1}{2}} = \alpha + \left|g_{i+\frac{1}{2}}\right|^{\frac{1}{m}}, \quad i = 0, 1, \ldots, N-1. \tag{2.20}$$

If the spatial derivatives in (2.9) are discretized by second-order central differences on the grid (2.4) we obtain the semidiscrete system of moving mesh equations defined by

$$\dot{x}_i = \frac{4}{\tau}(\tilde{M}_i(x_{i+1} - x_{i-1}))^{-2}\left(\tilde{M}_{i+\frac{1}{2}}(x_{i+1} - x_i) - \tilde{M}_{i-\frac{1}{2}}(x_i - x_{i-1})\right), \tag{2.21}$$

for $i = 1, 2, \ldots, N-1$, with $x_0 = 0$ and $x_N = 1$.

In (2.21), $\tilde{M}_{i+\frac{1}{2}}$ is a smoothed monitor function defined as in [18] by

$$\tilde{M}_{i+\frac{1}{2}} = \frac{\sum_{k=i-p}^{i+p} M_{k+1/2}(q/(q+1))^{|k-i|}}{\sum_{k=i-p}^{i+p}(q/(q+1))^{|k-i|}}, \tag{2.22}$$

where q is a positive real number and p is a nonnegative integer. Furthermore, in (2.22) the summations contain only those terms that are well defined ($0 \le k \le N-1$). The term $\tilde{M}_i$ in (2.21) is given by

$$\tilde{M}_i = \frac{\tilde{M}_{i-\frac{1}{2}}\left(x_{i+\frac{1}{2}} - x_i\right) + \tilde{M}_{i+\frac{1}{2}}\left(x_i - x_{i-\frac{1}{2}}\right)}{\left(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}\right)}. \tag{2.23}$$

The solution $\{x_i, v_i\}_{i=0}^N$ is obtained using a numerical integration of the systems (2.16) and (2.21) from suitably chosen initial states. To effect the numerical integration from $t = t_n$ to $t = t_{n+1}$ the systems of differential equations are uncoupled. Equation (2.16) is regarded as an equation in $\{v_i\}_{i=0}^N \equiv \mathbf{v}$ in which approximations to the node locations are available at $t = t_n$ and $t = t_{n+1}$, and (2.21) is the governing equation for $\{x_i\}_{i=0}^N \equiv \mathbf{x}$ in which $\tilde{M}_{i\pm\frac{1}{2}}$ and the term $(\tilde{M}_i(x_{i+1} - x_{i-1}))^{-2}$ are known for $i = 1, 2, \ldots, N-1$. Given $\mathbf{x}$ and $\mathbf{v}$ at $t = t_n$, the smoothed monitor function is evaluated at this value of t, and the term $(x_{i+1} - x_{i-1})$ is evaluated using approximations to $\mathbf{x}$ at $t = t_{n+1}$. The approximation used is the value of $\mathbf{x}$ at $t = t_n$ or the value in the preceding cycle when an iterative approach is employed to solve the decoupled system. The mesh is computed at $t = t_{n+1}$ using an implicit Euler—or first-order BDF (BDF1)—approximation to the linearized system (2.21). The vector $\mathbf{v}$ is then computed at $t = t_{n+1}$ by means of a second-order singly diagonally implicit Runge–Kutta (SDIRK) method [14]. Suppose this SDIRK method (SDIRK2) is employed to integrate the system

$$\dot{\mathbf{v}} = \mathbf{f}(t, \mathbf{v}), \tag{2.24}$$

where $\mathbf{f}$: $\mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$. The method is represented by the Butcher array

$$\begin{array}{c|c} \mathbf{c} & A \\ \hline & \mathbf{b}^T \end{array} = \begin{array}{c|cc} \gamma & \gamma & 0 \\ 1 & 1-\gamma & \gamma \\ \hline & 1-\gamma & \gamma \end{array}, \tag{2.25}$$

where $\gamma = (2 - \sqrt{2})/2$. Integration of (2.24) from $t = t_n$ to $t = t_{n+1} = t_n + \Delta t_n$ is given by

$$\begin{aligned} \mathbf{k}_1 &= \mathbf{f}(t_n + \gamma \Delta t_n, \mathbf{v}_n + \gamma \Delta t_n \mathbf{k}_1), \\ \mathbf{k}_2 &= \mathbf{f}(t_n + \Delta t_n, \mathbf{v}_n + (1-\gamma)\Delta t_n \mathbf{k}_1 + \gamma \Delta t_n \mathbf{k}_2), \\ \mathbf{v}_{n+1} &= \mathbf{v}_n + \Delta t_n((1-\gamma)\mathbf{k}_1 + \gamma \mathbf{k}_2). \end{aligned} \tag{2.26}$$

Here $\mathbf{v}_n$ denotes the value of $\mathbf{v}$ at $t = t_n$. Solution of (2.26) for $\mathbf{k}_1$ and $\mathbf{k}_2$ is obtained using a Newton iteration, terminating when convergence is achieved in the l_∞ norm to within a tolerance KTOL, which is set at $\varepsilon \times 10^{-6}$.

In the computations described in Section 3 the time step Δt_n is fixed at a value Δt throughout the integration, and in the computations described in Section 4 the second-order method is combined with a first-order method to estimate the local error in the latter. This error indicator is used to control the time step so that the first-order integration is carried out to an acceptable degree of accuracy. If $\hat{\mathbf{v}}_{n+1}$ is the approximation to $\mathbf{v}$ at $t = t_{n+1}$ given by the first-order method, the error indicator for $\hat{\mathbf{v}}_{n+1}$ is

$$\text{ERR} = \|\mathbf{v}_{n+1} - \hat{\mathbf{v}}_{n+1}\|_{l_\infty}. \tag{2.27}$$

If ERR >ETOL, where ETOL is a preset error tolerance, the computed solution at $t = t_{n+1}$ is rejected and the solution is recomputed with a smaller time-step. If ERR $\leq$ ETOL, the next time step is given by

$$\Delta t_{n+1} = \Delta t_n \left(\frac{\text{ETOL}}{\text{ERR}} \right)^{\frac{1}{2}}. \tag{2.28}$$

In practice, a more sophisticated time-step revision is used as described in Hairer and Wanner [14]. This is given by

$$\Delta t_{n+1} = \Delta t_n \times \min\left(\text{maxfac}, \max\left[\text{minfac}, \eta \left(\frac{\text{ETOL}}{\text{ERR}} \right)^{\frac{1}{2}} \right] \right), \tag{2.29}$$

where maxfac is a number in the range $1.5 \leq \text{maxfac} \leq 3$, $\eta \sim 0.6$ and minfac is set to 0.1. The update given by (2.29) prevents rapid changes in the time step. In practical computations, the local error indicator ERR may not give an accurate indication of the global error and it may be claimed that the accepted approximation should be that given by the second-order

method. The time-step control process is then simply taken to be a guide to the stepsize selection. This relaxation is adopted in the calculations described in Section 4.

To maximize computational efficiency, the first-order scheme that is combined with (2.26) should use the same vector $\mathbf{c}$ and matrix A as those given in (2.25). An embedded first-order SDIRK scheme satisfying these conditions gives the approximation

$$\hat{\mathbf{v}}_{n+1} = \mathbf{v}_n + \Delta t_n \mathbf{k}_1, \tag{2.30}$$

where $\mathbf{k}_1$ is given in (2.26). In the process of integrating (2.16) from t_n to t_{n+1} by means of SDIRK2, the vector $\dot{\mathbf{x}}$ is replaced by $(\mathbf{x}_{n+1} - \mathbf{x}_n)/\Delta t_n$ and $\mathbf{x}$ is evaluated in $[t_n, t_{n+1}]$ using the linear interpolant

$$\mathbf{x} := \mathbf{x}_n + \dot{\mathbf{x}}(t - t_n). \tag{2.31}$$

Here $\mathbf{x}_n$ denotes the approximation to $\mathbf{x}$ at $t = t_n$.

The initial state for $\mathbf{x}$ at $t = t_0$ is given by integrating the moving mesh Eq. (2.21) to a steady state, starting with an evenly spaced mesh. In this preliminary integration the exact solution (2.2) at $t = 0$ is used to evaluate the monitor function, and a steady state is assumed to be achieved when the maximum nodal deviation between meshes on two consecutive time steps is less than a prescribed tolerance, MTOL. In the computations to be described later we choose MTOL $= \varepsilon \times 10^{-3}$. If the mesh generated by this preliminary integration of (2.21) is denoted by $\{x_i(0)\}_{i=0}^N = \mathbf{x}_0$ then the exact solution (2.2) gives the initial values of $\{v_i\}_{i=0}^N = \mathbf{v}_0$ as $\{u(x_i(0), 0)\}_{i=0}^N$.

The key steps in evolving the solution of (2.16) and (2.21) from t_n to t_{n+1} are summarised in the *one-pass* algorithm below.

ONE-PASS ALGORITHM

(i) Form initial conditions $\mathbf{x}_0$ and $\mathbf{v}_0$ as described previously. Select time step and the various parameters. $n := 0$

(ii) Evaluate monitor function at t_n and obtain $\mathbf{x}_{n+1}$ from (2.21) using BDF1.

(iii) Obtain $\mathbf{v}_{n+1}$ from (2.16) using SDIRK2, with $\mathbf{x}$ and $\dot{\mathbf{x}}$ given by (2.31).

(iv) $n := n + 1$. Go to (ii).

Some modifications of this basic algorithm are adopted in the various numerical experiments to be described. One useful modification is referred to as a *two-pass* solution. In this case, a return is made to step (ii) after $\mathbf{v}_{n+1}$ has been obtained in step (iii): the monitor function is evaluated at t_{n+1} and $\mathbf{x}_{n+1}$ is recomputed by means of (2.21). Steps (iii) and (iv) are then applied. This recycle on steps (ii) and (iii) is repeated in the *multipass* mode until the approximations to $\mathbf{x}$ at t_{n+1} have converged to within a tolerance MTOL in the l_∞ norm. Methods of solution based on decoupling the mesh equations from the physical PDE have been used by other authors. For example, Verwer and Blom [20] decoupled the systems. They used one pass for the mesh equations and two for the physical PDE at each time step.

We shall see in Section 4 that time-step control based entirely on the error indicator $\|\mathbf{v}_{n+1} - \hat{\mathbf{v}}_{n+1}\|_{l_\infty}$ does not perform well in all situations. This may be because the discrepancy between $\mathbf{v}_{n+1}$ and $\hat{\mathbf{v}}_{n+1}$ in (2.27) is not significant if the mesh placement is poor. In this case, (2.28) will overestimate the permissible time step.

3. ACCURACY CONSIDERATIONS

3.1. *Interpolation Error on Initial Grid*

The Appendix considers the error incurred when the function

$$u(x) = \tanh\left(\frac{x - x_0}{\varepsilon}\right), \quad 0 < x_0 < 1, \tag{3.1}$$

is approximated in $[0, 1]$ by its piecewise linear interpolant on a nonuniform grid determined by equidistribution of the AL monitor function (2.10) and the BM monitor function (2.11) with $m = 2$. The error behavior as a grid is refined will depend on the choice of norm. When using difference methods that are designed to resolve layers in the solution the maximum norm is appropriate due to the need to measure the error within the layers. Other norms involve averages of the error and this smooths out rapid changes and prevents the norm from capturing the local behavior of the error within the layer. Analyses of difference schemes that exhibit uniform convergence with respect to the perturbation parameter are presented in [16, 19], where it is shown that the appropriate norm is the maximum discrete norm. Accordingly, the linear interpolation error is measured in the L_∞ norm and the errors in discrete solutions of (2.1), (2.2) are measured in the l_∞ norm.

For the AL monitor function the tightest bound obtained in the L_∞ norm for the error is of the form CN^{-1}, where C is a constant that is independent of N. In the case of the BM monitor function, the corresponding bound on the error is CN^{-2}.

The theoretical bounds on the linear interpolation error were checked numerically for the tanh profile given by the initial value of the exact solution (2.2). The equidistributed grid may be determined exactly for the BM monitor function from the discrete form of the equidistribution condition (2.13), which is

$$\int_0^{x_i} M(u(x,t))\,dx = \xi_i \int_0^1 M(u(x,t))\,dx. \tag{3.2}$$

The grid is obtained by numerical solution of (3.2) for the AL monitor function. An approximation to the L_∞ error is obtained by computing the error at 10 evenly spaced points in each interval (x_i, x_{i+1}), $i = 0, 1, \ldots, N-1$.

Tables I and II show the L_∞ errors and the estimated convergence rates for the BM and AL monitor functions at several values of ε.

TABLE I

L_∞ Errors and Convergence Rates for the Linear Interpolant Approximating $u(x, 0)$ in (2.2) on Equidistributed Grid with BM Monitor Function

	$\varepsilon = 1 \times 10^{-4}$		$\varepsilon = 1 \times 10^{-6}$		$\varepsilon = 1 \times 10^{-8}$	
N	$\|e\|_{L_\infty}$	Conv. rate	$\|e\|_{L_\infty}$	Conv. rate	$\|e\|_{L_\infty}$	Conv. rate
8	9.30×10^{-2}		1.12×10^{-1}		1.19×10^{-1}	
16	2.34×10^{-2}	1.99	2.89×10^{-2}	1.96	3.09×10^{-2}	1.95
32	5.58×10^{-3}	2.07	7.18×10^{-3}	2.01	7.70×10^{-3}	2.01
64	1.30×10^{-3}	2.11	1.76×10^{-3}	2.03	1.90×10^{-3}	2.02

TABLE II
Results as in TABLE I Using the AL Monitor Function

	$\varepsilon = 1 \times 10^{-4}$		$\varepsilon = 1 \times 10^{-6}$		$\varepsilon = 1 \times 10^{-8}$	
N	$\|e\|_{L_\infty}$	Conv. rate	$\|e\|_{L_\infty}$	Conv. rate	$\|e\|_{L_\infty}$	Conv. rate
8	1.37×10^{-1}		1.72×10^{-1}		1.80×10^{-1}	
16	6.67×10^{-2}	1.04	8.58×10^{-2}	1.01	9.61×10^{-2}	0.90
32	3.16×10^{-2}	1.08	4.22×10^{-2}	1.02	4.74×10^{-2}	1.02
64	1.51×10^{-2}	1.07	2.06×10^{-2}	1.03	2.35×10^{-2}	1.01

The results displayed in Tables I and II enable us to anticipate the limitations on the accuracy that might be expected when the test problem (2.1), (2.2) is solved using the adaptive moving mesh method presented in Section 2. Table I shows that the interpolation error on the initial grid behaves like C/N^2 if the grid is generated by equidistribution of the BM monitor function. For the AL monitor function, Table II shows that the interpolation error on the initial grid behaves like C/N. In both cases, the results suggest that the constant C is independent of N and ε, and the errors behave like the theoretical bounds that are presented in the Appendix.

The significance of the results presented in Tables I and II in relation to the solution of (2.1), (2.2) by a moving mesh method is that the anticipated error behavior is at best $O(N^{-1})$ and $O(N^{-2})$ for the AL and BM monitor functions, respectively. The numerical results presented in the following sections will vindicate these expectations.

3.2. *Accuracy in Relation to Method of Solution With Fixed Time Step*

Here we consider how the accuracy of the computed solution is influenced by the method adopted to solve the semidiscrete moving mesh Eq. (2.21). Specifically, we compare the performance of the *one-pass* algorithm with two *multipass* algorithms. Results are presented for both the BM and AL monitor functions with $\varepsilon = 2.0 \times 10^{-3}$. A fixed time step is employed, the size of which is chosen experimentally to provide comparable spatial and temporal components of the total computational error. To provide control data, we solve the discretized PDE (2.16) on a mesh obtained by exact equidistribution of the BM monitor function at each time step. This is achieved by exact solution of (3.2) with $u(x, t)$ given by the analytic solution (2.2) at $t = t_{n+1}$. The replacement of step (ii) in the *one-pass* algorithm by this solution process for $\mathbf{x}_{n+1}$ gives an exactly equidistributed grid that we shall refer to as EEG. The test problem (2.1), (2.2) was solved over the interval $0 < t \leq 1.0$ using the *one-pass* algorithm on the EEG: maximum nodal errors and estimated convergence rates are presented in Table III. It may be seen that high accuracy is achieved in the l_∞ norm for a time step that is large relative to ε. Note also that the maximum nodal error decays at second order in both Δt and N^{-1}, which is the optimal rate of convergence for this discretization.

Results presented in Table III may be used as a benchmark against which we can measure the accuracy of the fixed time-step adaptive scheme described in Section 2. The decoupled system in $\mathbf{x}$ and $\mathbf{v}$ is solved using the *one-pass* and *multipass* algorithms described in Section 2. Tables IV, V, and VI present results for *four-pass*, *two-pass*, and *one-pass* algorithms, respectively, with equidistribution based on the BM monitor function. The temporal and spatial smoothing parameter values used in Eqs. (2.9) and (2.22) are $\tau = 0.1$, $q = 2.0$, and $p = 3$. One additional feature introduced here involves the use of a relaxation parameter, ω, in the determination of $\mathbf{x}$ at $t = t_{n+1}$. If $\mathbf{x}^*_{n+1}$ denotes the value computed at step (ii) in

TABLE III
EEG Results for the BM Monitor Function

N	$(\Delta t)^{-1}$	$\|\mathbf{e}\|_{l_\infty}$	Conv. rate
32	50	5.82×10^{-3}	
64	100	1.03×10^{-3}	2.50
128	200	2.27×10^{-4}	2.18
256	400	5.55×10^{-5}	2.04
512	800	1.38×10^{-5}	2.00

TABLE IV
Four-pass Algorithm with BM Monitor Function

N	$(\Delta t)^{-1}$	$\|\mathbf{e}\|_{l_\infty}$	Conv. rate	Cost
32	50	6.58×10^{-3}		1.36
64	100	7.00×10^{-4}	3.23	3.52
128	200	1.84×10^{-4}	1.92	12.48
256	400	5.03×10^{-5}	1.87	48.6
512	800	1.32×10^{-5}	1.93	207

TABLE V
Two-pass Algorithm with BM Monitor Function

N	$(\Delta t)^{-1}$	$\|\mathbf{e}\|_{l_\infty}$	Conv. rate	Cost
32	50	5.34×10^{-1}		1.00
64	100	7.79×10^{-4}	9.42	1.86
128	200	5.13×10^{-4}	0.60	6.65
256	400	7.17×10^{-5}	2.84	25.4
512	800	1.10×10^{-5}	2.70	95.2

TABLE VI
One-pass Algorithm with BM Monitor Function

N	$(\Delta t)^{-1}$	$\|\mathbf{e}\|_{l_\infty}$	Conv. rate	Cost
32	3200	3.52×10^{-3}		10.3
64	6400	7.03×10^{-4}	2.32	39.8
128	12800	1.89×10^{-4}	1.90	159

the *one-pass* algorithm then the relaxed value, $\mathbf{x}_{n+1}$, that is used in step (iii) is given by

$$\mathbf{x}_{n+1} := \omega \mathbf{x}^*_{n+1} + (1-\omega)\mathbf{x}_n, \tag{3.3}$$

where $\omega = 0.2$ throughout.

The computational costs displayed in Table IV and in all subsequent tables are the costs in flops of the integrations over the interval $0 < t \leq 1.0$, normalized relative to the cheapest computation in the complete set. The cheapest computation is the *two-pass* computation whose results are displayed in Table V with $N = 32$.

The maximum nodal error in the solution given by the *four-pass* algorithm is marginally smaller than that in the solution computed on the EEG, but the convergence rate is approaching second order from below as N increases and Δt decreases, with $N \times \Delta t$ held constant. In the asymptotic limit as $N \to \infty$ and $\Delta t \to 0$, the difference between the EEG results and the *four-pass* results becomes negligible. The small discrepancy between the two modes of solution can be attributed to the moderating effects of the smoothing that is incorporated in the decoupled algorithms to improve the robustness of the decoupled solver.

In comparison, Table V shows that the convergence behavior of the *two-pass* algorithm is more erratic than that of the *four-pass* algorithm as N increases and Δt decreases. However, the error in the *two-pass* computation approaches that in the *four-pass* computation as N increases, with $N \times \Delta t$ held constant. Note also that the computational cost of a *two-pass* computation is approximately half that of a *four-pass* computation for a specified, and sufficiently large, value of N. The source of the erratic error behavior for low N and large Δt is the error peak near $t = 0$ that occurs as the grid adapts to the moving front. This error peak is less pronounced in the case of the *four-pass* algorithm.

Finally, we consider the *one-pass* results presented in Table VI. The low computational cost makes this a popular choice, especially in multidimensional computations. However, because of the linearization of (2.21) that involves the evaluation of the monitor function at $t = t_n$, it is clear that the grid predicted by the *one-pass* algorithm will be based on the numerical solution at the previous time step. In practice, this limits the maximum time step that may be used, and the time-step restriction depends on the value of ε and on the rate of propagation of the front. The results in Table VI confirm this defect: to obtain accuracy that is comparable with the *multipass* algorithms the time step has to be reduced drastically. Indeed, with a reduction in time step by a factor of 64 the accuracy of the *one-pass* algorithm becomes comparable with the accuracies achieved by the *four-pass* and EEG computations. Tables III, IV, and VI demonstrate, however, that the additional time steps required by the *one-pass* algorithm to achieve this parity with the *multipass* algorithms make the former a much more expensive mode of computation.

Tables VII–IX present results of analogous computations using the AL monitor function. The relative performances of the *four-pass*, *two-pass*, and *one-pass* algorithms are similar to those produced in the computations with the BM monitor function. However, the accuracies achieved with the AL monitor function differ considerably from those achieved with the BM monitor function. The nodal error in the AL computations does not decay at second order in Δt and N^{-1}: indeed, the convergence rate is slightly less than first order. Since the work load involved in using the AL monitor function is comparable to that for the BM monitor function, the efficiency of the AL computation is considerably lower than that of the BM computation.

The results presented in this section are summarized in Fig. 1, where we show a log–log plot of the l_∞ error against computational cost for each of the two monitor functions in the

TABLE VII
Four-pass Algorithm with AL Monitor Function

N	$(\Delta t)^{-1}$	$\|\mathbf{e}\|_{l_\infty}$	Conv. rate	Cost
32	200	2.80×10^{-3}		2.76
64	400	1.43×10^{-3}	0.97	11.3
128	800	7.94×10^{-4}	0.85	41.6
256	1600	4.52×10^{-4}	0.81	87.8
512	3200	1.97×10^{-4}	1.20	318

TABLE VIII
Two-pass Algorithm with AL Monitor Function

N	$(\Delta t)^{-1}$	$\|\mathbf{e}\|_{l_\infty}$	Conv. rate	Cost
32	200	2.84×10^{-3}		1.63
64	400	1.41×10^{-3}	1.01	6.05
128	800	7.92×10^{-4}	0.84	20.5
256	1600	4.51×10^{-4}	0.81	79.9
512	3200	1.97×10^{-4}	1.20	318

TABLE IX
One-pass Algorithm with AL Monitor Function

N	$(\Delta t)^{-1}$	$\|\mathbf{e}\|_{l_\infty}$	Conv. rate	Cost
32	1600	2.78×10^{-3}		5.66
64	3200	1.43×10^{-3}	0.96	22.2
128	6400	7.97×10^{-4}	0.84	88.0

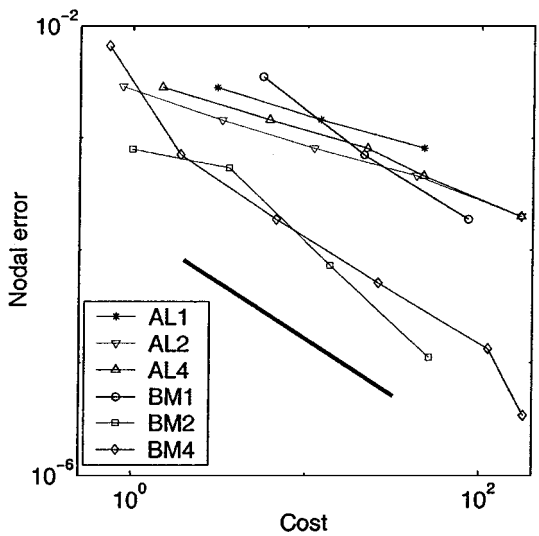

FIG. 1. Cost comparison of AL and BM monitor functions for various pass strategies.

one-, *two-*, and *four-pass* modes of operation for a range of values of N and Δt. In each of the three modes it is seen that the slope of the graph for the BM monitor function is steeper than that for the AL monitor—approximately -1 in the BM case and -0.5 in the AL case. (The solid line indicates a slope of -1.) With the exception of BM1 (BM in the *one-pass* algorithm), the BM monitor function is always more efficient than the AL monitor function. Over the range of values of N and Δt considered here, BM2 and BM4 are comparable, particularly for large values of N and small values of Δt.

4. TIME-STEP CONTROL

4.1. *Control Mechanisms*

In this section we describe numerical experiments that illustrate features which are useful in the construction of a reliable time-step control mechanism. As outlined in Section 2, time-step control requires the availability of a reliable error indicator in the form of Eq. (2.27). The SDIRK2 scheme (2.26) and the embedded first-order SDIRK scheme (2.30) give rise to the error indicator

$$\text{ERR} = \|\mathbf{v}_{n+1} - \hat{\mathbf{v}}_{n+1}\|_{l_\infty} = \|\gamma \Delta t_n (\mathbf{k}_2 - \mathbf{k}_1)\|_{l_\infty}. \tag{4.1}$$

This error indicator is utilized in (2.29) to provide time-step revision and, as intimated immediately after (2.29), the more accurate SDIRK2 approximation, $\mathbf{v}_{n+1}$, is carried forward. Throughout this section, the temporal and spatial smoothing parameter values used in Eqs. (2.9) and (2.22) are $\tau = 0.1$, $q = 2.0$, and $p = 3$; the error control parameter values in (2.29) are minfac $= 0.1$, maxfac $= 2.0$, and $\eta = 0.64$, and the singular perturbation parameter ε is set at 2.0×10^{-3}.

In Table X and Fig. 2, we present numerical results for the BM monitor function using the *four-pass* algorithm. It can be seen that the nodal error is comparable to that in Table IV, where an appropriate fixed time-step was selected a priori. The results in Table X demonstrate second-order convergence with respect to Δt and N^{-1}. However, there are several unsatisfactory aspects to these results. First, the computational cost is higher in Table X than in Table IV. Second, as seen clearly in the inset in Fig. 2, the mesh points do not follow smooth trajectories as time evolves. Third, the time-step history displayed in Fig. 2 is highly erratic. While these factors do not have a significant effect on the nodal error in this computation, they are undesirable—diminishing the robustness of the time-step control mechanism and reducing the overall efficiency of the algorithm.

TABLE X
Four-pass Algorithm with BM Monitor Function and Time-step Control Applied Only to the Physical PDE: Δt_0 = 1/400

N	ETOL	$\|\mathbf{e}\|_{l_\infty}$	Conv. rate	Cost
32	5.00×10^{-4}	3.76×10^{-3}		5.88
64	1.25×10^{-4}	7.37×10^{-4}	2.35	10.7
128	3.00×10^{-4}	1.96×10^{-4}	1.91	35.3
256	7.50×10^{-6}	5.15×10^{-5}	1.93	76.6
512	2.00×10^{-6}	1.36×10^{-5}	1.92	263

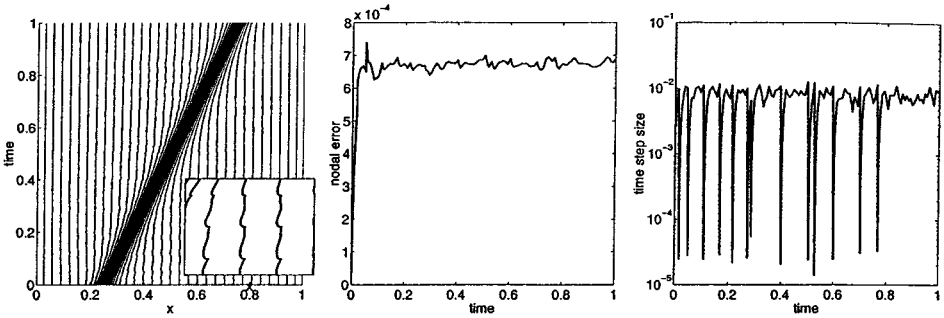

FIG. 2. Mesh trajectories, error, and time-step history for BM monitor, four-pass algorithm, with time-step control for physical PDE only.

Numerical experiments suggest that the reason for the poor performance of the time-step control mechanism used for Table X and Fig. 2 is its lack of consideration for the accuracy of the grid point location. It has been observed by several authors (see, for example, [1]) that it is not necessary to determine the grid with the same level of accuracy as the physical solution. Consequently, small errors in the solution of the physical PDE may coexist with larger errors in the grid location. Time-step control based solely on the accuracy of the solution of the physical PDE is likely, therefore, to be insensitive to any deterioration in the grid. This problem has been identified by Blom *et al.* [5] where, to compensate for a lack of control of the grid accuracy, the following modifications are made: if at time t_{n+1} a time-step failure occurs then not only is the current solution $(\mathbf{x}_{n+1}, \mathbf{v}_{n+1})$ rejected but so also is the solution at the previous step, $t = t_n$. Furthermore, in the steps immediately after the time-step failure additional constraints are imposed to prevent the time-step size from increasing too rapidly and giving rise to additional time-step failures. Figure 3 presents numerical results for the computation displayed in Fig. 2, with these additional time-step constraints imposed. It may be seen that some of the unsatisfactory aspects of the results presented in Fig. 2 have been alleviated. The mesh trajectories are significantly smoother than in Fig. 2 (see the inset in Fig. 3) and the erratic behavior of the time-step history has been reduced. However, the computational cost is comparable to that incurred for Fig. 2, this being a penalty brought about by the large number of time-step failures.

An alternative approach to that adopted by Blom *et al.* [5], is to incorporate additional measures of the mesh accuracy into the time-step control mechanism. For a *multipass*

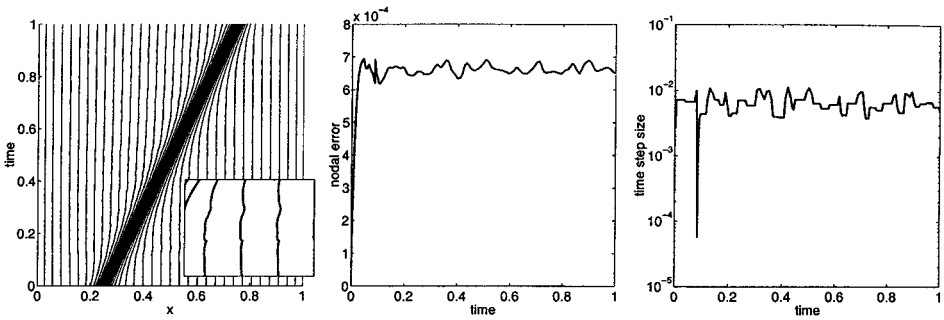

FIG. 3. Mesh trajectories, error, and time-step history for BM monitor, four-pass algorithm, with time-step control for physical PDE and mesh as in Blom *et al.* [5].

TABLE XI
Four-pass Algorithm with BM Monitor Function and Modified Time-step Control: $\Delta t_0 = 1/400$

N	ETOL	$\|\mathbf{e}\|_{l_\infty}$	Conv. rate	Cost
32	5.00×10^{-4}	3.06×10^{-3}		3.95
64	1.25×10^{-4}	6.92×10^{-4}	2.15	6.46
128	3.00×10^{-5}	1.88×10^{-4}	1.90	20.1
256	7.50×10^{-6}	5.09×10^{-5}	1.88	56.1
512	2.00×10^{-6}	1.35×10^{-5}	1.91	210

algorithm, a simple indication of the accuracy of the grid may be provided cheaply by

$$\text{mesh\_err} = \|\mathbf{x}_{n+1} - \hat{\mathbf{x}}_{n+1}\|_{l_\infty}, \tag{4.2}$$

where $\mathbf{x}_{n+1}$ denotes the final mesh at $t = t_{n+1}$ and $\hat{\mathbf{x}}_{n+1}$ denotes the mesh at the previous pass in the algorithm. If mesh_err $>$ MESH_TOL, where MESH_TOL is a user-prescribed error tolerance, then the solution at $t = t_{n+1}$ is rejected and the computation is repeated using a smaller time step. If mesh_err $\leq$ MESH_TOL then the next time step is computed as

$$\Delta t_{n+1} = \Delta t_n \times \min\left(\text{maxfac}, \max\left[\text{minfac}, \frac{\log(\text{mesh\_err})}{\log(\text{MESH\_BAL})}\right]\right), \tag{4.3}$$

where MESH_BAL is a user-chosen parameter satisfying MESH_BAL $<$ MESH_TOL. A value for Δt_{n+1} is also computed using (2.29) and the smaller of the two values is adopted.

Table XI and Fig. 4 show revised results using the modified time-step control mechanism with MESH_TOL $= 0.6\varepsilon$ and MESH_BAL $= 0.3\varepsilon$. As can be seen, the nodal errors incurred are comparable with those displayed in Table IV. A comparison of Tables X and XI shows that the introduction of the additional time-step control has brought about a significant reduction in computational cost and the number of time-step failures has been reduced to zero. The computational cost in Table XI is comparable with that in Table IV. Furthermore, the mesh trajectories are now much smoother (see the inset in Fig. 4) and the time-step size settles down to an appropriate value of approximately 6.0×10^{-3}.

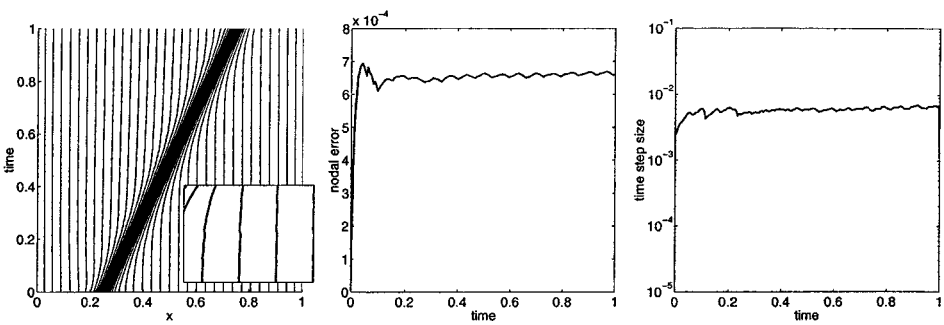

FIG. 4. Mesh trajectories, error, and time-step history for BM monitor, four-pass algorithm, with time-step control for the physical PDE and mesh.

TABLE XII
Four-pass Algorithm with AL Monitor Function and Modified Time-step Control: Δt_0 = 1/400

N	ETOL	$\|\mathbf{e}\|_{l_\infty}$	Conv. rate	Cost
32	5.00×10^{-4}	2.84×10^{-3}		2.28
64	2.50×10^{-4}	1.47×10^{-3}	0.95	5.69
128	1.00×10^{-4}	7.93×10^{-4}	0.89	19.6
256	5.00×10^{-5}	4.44×10^{-4}	0.84	50.8
512	2.50×10^{-5}	1.88×10^{-4}	1.24	164

Results are also presented for the AL monitor function in Table XII, with the *four-pass* algorithm and the modified time-step control mechanism (all parameters are as in the BM case). Once again, we see good agreement, with respect to nodal error and convergence rate, between these results and the corresponding fixed time-step results presented in Table VII. Note that the computational cost of the variable time-step algorithm with the AL monitor function is significantly less than that of the corresponding fixed time-step algorithm.

4.2. *Tracking the Front as It Leaves the Domain*

It is well known that additional numerical difficulties are encountered if we attempt to track the front as it leaves the domain [18]. For the test problem (2.1), (2.2), this occurs at $t = 1.5$ and it provides a good test of the robustness of the *multipass* algorithm.

Figures 5 and 6 show numerical results for the BM and AL monitor functions, with time-step control based on error indicators for the solution of the physical PDE and the mesh as for Table XI and Fig. 4. Parameter values are chosen as in Section 4.1. In both cases, we see that as the front approaches the righthand boundary of the domain, the nodal error increase induces multiple time-step failures, and this causes the time-step size to fall by two orders of magnitude. The increase in error is due, in part, to a spatial component that cannot be eliminated by a reduction in the time-step size and, as a consequence, some rise in error is inevitable. As the front leaves the domain, the contribution of $|u'|$ to the AL monitor function diminishes. This means that the constant floor in (2.10) becomes dominant and this results in a rapid propagation of mesh points to the left to produce a

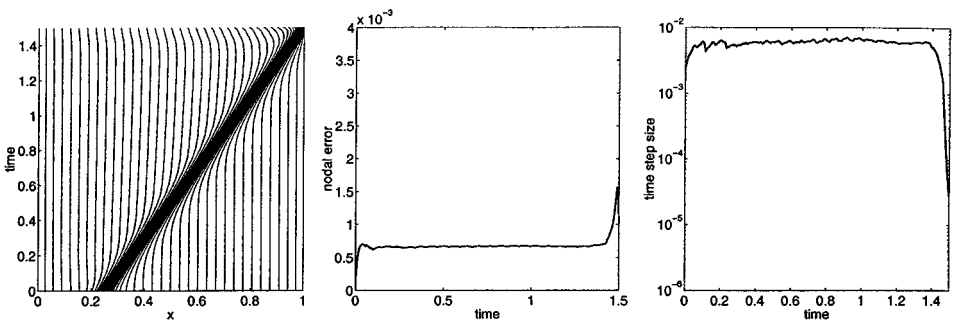

FIG. 5. Mesh trajectories, error, and time-step history for BM monitor, four-pass algorithm, with time-step control for physical PDE and mesh. Front leaving the domain.

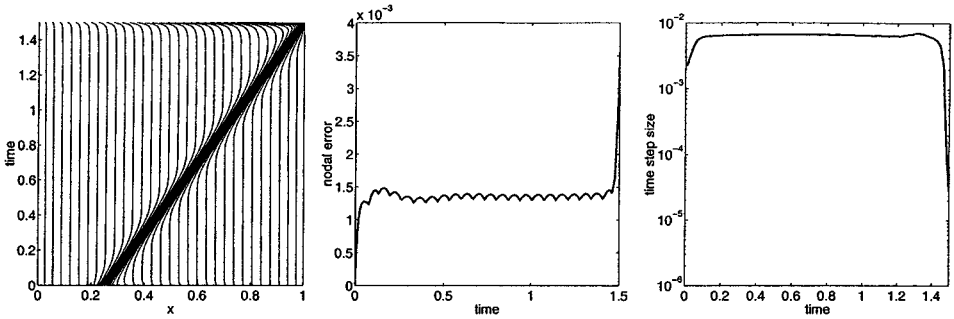

FIG. 6. Mesh trajectories, error, and time-step history for AL monitor, four-pass algorithm, with time-step control for physical PDE and mesh. Front leaving the domain.

uniform grid. This transformation of the grid occurs in a very small time frame of order ε and it causes additional numerical difficulties since terms in (2.8) that depend on the mesh velocity $\dot{\mathbf{x}}$ become very large. For the AL monitor function, the error increases by a factor of six to 8.31×10^{-3} as the front leaves the domain. In contrast, the floor α in the BM monitor function is automatically rescaled as the $|u'|$ contribution to the monitor function diminishes. This results in a much smoother transformation to a uniform grid. The effect of this smoother transition is that the nodal error for the BM case is much smaller, peaking at 1.56×10^{-3}, approximately double the error before the influence of the boundary is felt. In both examples, the dramatic reduction in the time-step size causes large increases in the computational costs.

5. CONCLUSIONS AND COMMENTS

The one-dimensional Burgers' equation has been used to elucidate several aspects of the solution of moving mesh partial differential equations. The moving mesh method used in the numerical experiments is one of those proposed by Huang *et al.* [9] and it is based on the equidistribution of a monitor function. By considering the familiar arc-length monitor function and one proposed by Beckett and Mackenzie [2, 3] it has been shown that the choice of monitor function has a significant influence on the accuracy of the computed results. Of particular interest is the demonstration that optimal convergence rates can be achieved by methods based on the Beckett and Mackenzie monitor function. These results are in line with predictions based on an analysis of the error incurred when the initial function is approximated by its piecewise linear interpolant on a nonuniform grid that is generated by equidistribution.

The moving mesh partial differential equations were solved by decoupling the mesh equations from the physical PDE. An algorithm has been presented that uses a first-order BDF method for the time integration of the linearized mesh equation and a second-order singly diagonally implicit Runge–Kutta (SDIRK) method for the physical PDE. It has been shown that the efficiency, accuracy, and robustness of the moving mesh equations all depend on the method that is adopted for the solution of the decoupled system. Numerical experiments using a fixed time step have shown how an efficient and accurate solution algorithm can be constructed. Finally, time-step control mechanisms were considered and a robust and efficient method presented that makes use of error indicators for the accuracy of the grid and the accuracy of the solution of the physical PDE. Work is being completed on

extending the algorithm to deal with problems in two space dimensions [4], and this work will be presented elsewhere.

APPENDIX

Interpolation Error on a Nonuniform Grid

In this section we consider the error incurred when the function

$$u(x) = \tanh\left(\frac{x - x_0}{\varepsilon}\right), \quad 0 < x_0 < 1,$$

is approximated by its piecewise linear interpolant, $\bar{u}(x)$. To begin with, we shall assume that the nonuniform mesh is generated by equidistributing the solution arc-length. From the definition of the interpolant it is clear that there exists a positive constant c_1 that is independent of $u(x)$ and ε such that

$$\|e\|_{L_\infty(x_{j-1},x_j)} \equiv \max_{x\in(x_{j-1},x_j)} |u(x) - \bar{u}(x)| \le c_1 \int_{x_{j-1}}^{x_j} |u'(x)|\,\mathrm{d}x.$$

Using the equidistribution principle we have

$$\int_{x_{j-1}}^{x_j} |u'(x)|\,\mathrm{d}x \le \int_{x_{j-1}}^{x_j} \sqrt{1 + (u'(x))^2}\,\mathrm{d}x = \frac{1}{N}\int_0^1 \sqrt{1 + (u'(x))^2}\,\mathrm{d}x \le \frac{c_2}{N},$$

where c_2 is independent of ε. Hence, there exists a C, independent of N and ε, such that

$$\|e\|_{L_\infty(0,1)} \le CN^{-1}.$$

Equidistribution of the solution arc-length therefore leads to a mesh that results is a suboptimal rate of convergence. Numerical experiments confirm that this estimate is sharp.

The second monitor function we consider is similar to that used by Beckett and Mackenzie [2, 3] for singularly perturbed boundary value problems. The monitor takes the form

$$M(x) = \alpha + |u'(x)|^{1/2}, \quad \alpha = \int_0^1 |u'(x)|^{1/2}\,\mathrm{d}x.$$

From standard interpolation theory there exists a positive constant c_3 that is independent of $u(x)$ and ε such that

$$\|e\|_{L_\infty(x_{j-1},x_j)} \le c_3 h_j^2 |u''(\eta_j)|,$$

where $\eta_j \in (x_{j-1}, x_j)$. Furthermore, we have

$$|u''(\eta_j)| \le \varepsilon^{-1}|u'(\eta_j)| \le c_4\varepsilon^{-1} \inf_{x\in(x_{j-1},x_j)} |u'(x)|,$$

where

$$c_4 = \max_{1\le j\le N}\left(\sup_{x\in(x_{j-1},x_j)} |u''(x)|h_j\right).$$

Using the global equidistribution principle we therefore have

$$\|e\|_{L_\infty(x_{j-1},x_j)} \le c_5\varepsilon^{-1}\left(\inf_{x\in(x_{j-1},x_j)} |u'(x_j)|^{1/2}h_j\right)^2$$
$$\le c_5\varepsilon^{-1}\left(\int_{x_{j-1}}^{x_j} \alpha + |u'(s)|^{1/2}\,\mathrm{d}s\right)^2 = c_5\varepsilon^{-1}\left(\frac{2\alpha}{N}\right)^2,$$

where $c_5 = c_3c_4$. Finally, we have

$$\alpha = \int_0^1 |u'(s)|^{1/2}\,\mathrm{d}s < c_6\varepsilon^{1/2}.$$

Hence, there exists a C, independent of N, such that

$$\|e\|_{L_\infty(0,1)} \le CN^{-2}.$$

Note that the constants c_3 and c_6 are independent of ε. Numerical experiments suggest that c_4 and hence C are also ε independent.

REFERENCES

1. I. Babuška and W. C. Rheinboldt, Analysis of optimal finite element meshes in $\mathbb{R}^1$, *Math. Comput.* **33**, 435 (1979).
2. G. Beckett and J. A. Mackenzie, On a uniformly accurate finite difference approximation of a singularly perturbed reaction–diffusion problem using grid equidistribution, *J. Comput. Appl. Math.*, in press.
3. G. Beckett and J. A. Mackenzie, Convergence analysis of finite difference approximations on equidistributed grids to a singularly perturbed boundary value problem, *Appl. Numer. Math.* **35**, 109 (2000).
4. G. Beckett, J. A. Mackenzie, A. Ramage, and D. M. Sloan, Computational solution of two-dimensional unsteady PDEs using moving mesh methods, in preparation.
5. J. G. Blom, J. M. Sanz-Serna, and J. G. Verwer, On simple moving grid methods for one-dimensional evolutionary partial differential equations, *J. Comput. Phys.* **74**, 191 (1988).
6. W. Cao, W. Huang, and R. D. Russell, A moving mesh method in multiblock domains with application to a combustion problem, *Numer. Meth. Partial Differential Equations* **15**, 449 (1999).
7. D. F. Hawken, J. J. Gottlieb, and J. S. Hansen, Review of some adaptive node-movement techniques in finite element and finite difference solutions of PDEs, *J. Comput. Phys*. **95**, 254 (1991).
8. W. Huang, *Practical Aspects of Formulation and Solution of Moving Mesh PDEs*, Technical Report (No. 99-11-02) (Department of Mathematics, University of Kansas).
9. W. Huang, Y. Ren, and R. D. Russell, Moving mesh partial differential equations (MMPDEs) based on the equidistribution principle, *SIAM J. Numer. Anal*. **31**, 709 (1994).
10. W. Huang, Y. Ren, and R. D. Russell, Moving mesh methods based on moving mesh partial differential equations, *J. Comput. Phys*. **113**, 279 (1994).
11. W. Huang and R. D. Russell, A moving collocation method for the numerical solution of time dependent differential equations, *Appl. Numer. Math*. **20**, 101 (1996).
12. W. Huang and R. D. Russell, A high dimensional moving mesh strategy, *Appl. Numer. Math*. **26**, 63 (1997).
13. W. Huang and R. D. Russell, Moving mesh strategy based on a gradient flow equation for two-dimensional problems, *SIAM J. Sci. Comput*. **20**, 998 (1999).
14. E. Hairer and G. Wanner, *Solving Ordinary Differential Equations: II* (Springer-Verlag, New York, 1991).
15. J. D. Lambert, *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*, 2nd ed. (Wiley, Chichester, 1991).

16. J. J. H. Miller, E. O'Riordan, and G. I. Shishkin, *Fitted Numerical Methods for Singular Perturbation Problems* (World Scientific, London, 1996).
17. K. Miller and R. N. Miller, Moving finite elements I, *SIAM J. Numer. Anal*. **18**, 1019 (1981).
18. L. S. Mulholland, Y. Qiu, and D. M. Sloan, Solution of evolutionary partial differential equations using adaptive finite differences with pseudospectral post-processing, *J. Comput. Phys*. **131**, 280 (1997).
19. H.-G. Roos, M. Stynes, and L. Tobiska, *Numerical Methods for Singularly Perturbed Differential Equations* (Springer-Verlag, Berlin, 1996).
20. J. G. Verwer and J. G. Blom, An adaptive moving grid method for one-dimensional systems of PDEs, *J. Comput. Phys*. **82**, 454 (1989).