Available online at www.sciencedirect.com
ScienceDirect
Journal of the Franklin Institute 344 (2007) 757–764
Journal of The Franklin Institute
www.elsevier.com/locate/jfranklin

An adaptive mesh method with variable relaxation time

Ali R. Soheili[a,*], John M. Stockie[b]

[a]Department of Mathematics, Sistan and Baluchestan University, Zahedan, Iran
[b]Department of Mathematics, Simon Fraser University, Burnaby, BC, Canada

Received 8 February 2006; accepted 8 February 2006

Abstract

Moving mesh partial differential equations have been widely used in the last decade for solving differential equations exhibiting large solution variations such as shock waves and boundary layers.

In this paper, we have applied a dynamic adaptive method for solving time-dependent differential equations. The mesh velocities are governed by an equation in which a *relaxation time* is employed to move nodes in such a way that they remain concentrated in regions of rapid variation of the solution. A numerical example involving a blow-up problem shows the advantage of using a variable relaxation time over a fixed one.

Keywords: Moving mesh method; Blow-up problem; Relaxation time

1. Introduction

Adaptive mesh methods have been widely used for approximating partial differential equations that involve large solution variations. Several moving mesh approaches have been derived and many people have discussed the significant improvements in accuracy and efficiency that can be achieved with respect to fixed mesh methods [1–4]. For the type of dynamical moving mesh method considered here, another partial differential equation governing the mesh evolution is solved alongside the original [5,6].

*Corresponding author. Tel.: +98 915 141 9041; fax: +98 541 244 6409.
E-mail addresses: soheili@hamoon.usb.ac.ir (A.R. Soheili), stockie@math.sfu.ca (J.M. Stockie).

0016-0032/$30.00
doi:10.1016/j.jfranklin.2006.02.014

An ideal class of problems for examining the behaviour of moving mesh methods is that of blow-up of temperature in a reacting medium. One of the simplest equations in this class will form the basis of our numerical simulations [7–9]:

$$
\begin{aligned}
&u_t = u_{xx} + f(u),\\
&u(0,t) = u(1,t) = 0,\\
&u(0,x) = u_0(x).
\end{aligned}
\tag{1}
$$

If $u_0(x)$ is sufficiently large, positive and has a single non-degenerate maximum, then there is a blow-up time $T<\infty$ and a unique blow-up point x^* such that

$$u(x^*,t)\longrightarrow\infty \quad \text{as} \quad t\longrightarrow T$$

and

$$u(x,t)\longrightarrow u(x,T)<\infty \quad \text{if } x\neq x^*.$$

The paper is organized as follows: In Section 2, we review briefly moving mesh methods in which the mesh equations incorporate a relaxation time τ. Blow-up problems are introduced in Section 3 and in Section 4, we improve the moving mesh by describing an extension to the method which uses a variable relaxation time. The numerical experiments in Section 5 illustrate the advantages of this new method.

2. Moving mesh methods

Let x and ξ denote the physical and computational coordinates, respectively, both of which are assumed to be in [0, 1]. Define a one-to-one coordinate transformation by

$$
\begin{aligned}
&x = x(\xi,t), \quad \xi\in[0,1],\\
&x(0,t) = 0, \quad x(1,t) = 1.
\end{aligned}
$$

The computational coordinate is discretized on a uniform mesh given by

$$\xi_i = \frac{i}{N}, \quad i = 0,1,\ldots,N, \tag{2}$$

where N is a certain positive integer and the corresponding non-uniform mesh in x is denoted by

$$0 = x_0 < x_1(t) < x_2(t) < \cdots < x_{N-1}(t) < x_N = 1.$$

For a chosen monitor function $M(x,t)>0$, the moving mesh $x_i(t)$ satisfies the following equidistribution principle (EP) for all values of time t. The equidistribution principle is one of the most important concepts in the development of moving mesh methods [1]:

$$\int_{x_{i-1}(t)}^{x_i(t)} M(x,t)\,\mathrm{d}x = \frac{1}{N}\int_0^1 M(x,t)\,\mathrm{d}x = \frac{\theta(t)}{N}$$

or

$$\int_0^{x_i(t)} M(x,t)\,\mathrm{d}x := \frac{i}{N}\theta(t) = \xi_i\theta(t). \tag{3}$$

Differentiating the above equation yields an equivalent differential form

$$\frac{\partial}{\partial \xi}\left(M\frac{\partial x}{\partial \xi}\right)(\xi, t) = 0, \tag{4}$$

where $x(0,t) = 0$ and $x(1,t) = 1$.

A moving mesh equation can be derived by taking the mesh to satisfy the above EP equation at a later time $t+\tau$ instead of at t, since $\theta(t)$ has been omitted in the above differential form of the EP. In this case, the mesh should satisfy

$$\frac{\partial}{\partial \xi}\left\{M(x(\xi, t+\tau), t+\tau)\frac{\partial}{\partial \xi}x(\xi, t+\tau)\right\} = 0,$$

where the parameter τ is called a *relaxation time* and also has the effect of introducing temporal smoothing. By expanding the terms $(\partial/\partial\xi)x(\xi, t+\tau)$ and $M(x(\xi, t+\tau), t+\tau)$ in Taylor series and dropping certain higher order terms, various MMPDEs can be obtained [1]. In this paper, we will employ two moving mesh methods labelled MMPDE4 and MMPDE6 below

$$\tau\frac{\partial}{\partial \xi}\left(M\frac{\partial \dot{x}}{\partial \xi}\right) = -\frac{\partial}{\partial \xi}\left(M\frac{\partial x}{\partial \xi}\right), \quad \text{(MMPDE4)}$$

$$\tau\frac{\partial^2 \dot{x}}{\partial \xi^2} = -\frac{\partial}{\partial \xi}\left(M\frac{\partial x}{\partial \xi}\right), \quad \text{(MMPDE6).}$$

In practice, it is important to smooth the monitor function in space as well [10]. To this end, the monitor function values M_i appearing in the discretized form of the MMPDEs above are replaced with

$$\tilde{M}_i = \left(\frac{\sum_{j=i-ip}^{i+ip}(M_j)^2(\gamma/1+\gamma)^{|j-i|}}{\sum_{j=i-ip}^{i+ip}(\gamma/1+\gamma)^{|j-i|}}\right)^{1/2}, \tag{5}$$

where we take $\gamma = 2$ and $ip = 4$.

3. Blow-up problems

In this paper, we consider the semi-linear parabolic differential equation

$$u_t = u_{xx} + u^p, \tag{6}$$

where $p>1$ and $0\leqslant x\leqslant 1$, having boundary conditions $u(0,t) = u(1,t) = 0$ and initial condition $u(0,x) = u_0(x)$. If the initial value $u_0(x)$ is sufficiently large and has a single non-degenerate maximum then there exists a blow-up point x^* [11]. We restrict ourselves to the case $p = 5$ and, following [7], we take $u_0(x) = 20\sin(\pi x)$.

4. Variable relaxation time

In most other work on moving mesh methods, the relaxation time τ is taken to be constant, or at best the numerical results have been presented with a few different values for τ [9,2]. Here, we take τ as a function of time, or of the solution. In practice, we have had to fix τ within a very small neighborhood of the blow-up point owing to the breakdown of the solution in the blow-up problem we are investigating.

One form of the relaxation time, motivated by the analysis of chemotactic blow-up problems presented in [8], is

$$\tau_1(t) = \begin{cases} |T-t|^{\alpha}, & t<t^*, \\ |T-t^*|^{\alpha} & \text{otherwise.} \end{cases} \tag{7}$$

where $\alpha = 1$, $t^* = T - 10^{-6}$, and T is an estimate of the blow-up time. This is a natural choice since more temporal smoothing (i.e., a smaller value of τ) is required in regions where the solution blows up. However, the relaxation time in Eq. (7) is applicable only to problems of the form (6) and hence is not useful for a general moving mesh solver. On the other hand, the dependence of the monitor function for blow-up problems is shown in [8] to be of the form $|T-t|^{-\alpha}$, which suggests the alternate choice

$$\tau_2(t) = \frac{C}{\max_x (M(x,t))}, \tag{8}$$

where C is some constant. This second form of the relaxation time is more general because it is based on the solution through value of the monitor function. Since the monitor function takes on large values in regions where the mesh must be refined, τ is necessarily smaller in these regions as expected. Both forms of the relaxation time, $\tau_1(t)$ and $\tau_2(t)$ will be employed in our numerical experiments.

5. Numerical experiments

As a first example, consider solving the mesh equation only for the given function

$$u(x,t) = \mathrm{e}^{-\pi^2 t}\sin(\pi x), \quad 0 \leqslant x \leqslant 1.$$

This function was used in [12] to study the stability of various mesh equations, and as a numerical example in [1]. Since $u_x(x,t) \to 0$ in the limit as $t \to \infty$, then for the typical arc-length monitor function $M_1 = \sqrt{1+u_x^2}$, we have $M_1(x,t) \to 1$ as $t \to +\infty$ and therefore the equidistributed mesh should tend to a uniform mesh in space. Fig. 1 shows the mesh trajectories for the above example using MMPDE6 and a uniform initial mesh. This figure demonstrates that the mesh trajectories depend quite strongly on the value of the relaxation time τ, and that there is a limiting value τ_0 such that for $\tau < \tau_0$ the mesh trajectories are relatively unchanged.

For the blow-up problem (6), if we apply moving mesh method to solve with an underlying scaling invariance, the MMPDE4 or MMPDE6 should be invariant under a scaling formula [9]. That means, in the moving mesh equation, the relaxation parameter τ and the monitor function M can indeed be suitably chosen to meet this requirement. Budd et al. in [9] described a scaling formulae, which if τ is taken constant and $M_2(u) = u^{p-1}$, then MMPDE4 and MMPDE6 can be made invariant under that scaling relations. Here, we also consider a new monitor function $M_3(u) = u^{(p-1)/2}$ and our numerical experiment shows the problem be integrated further and the solution has a good accuracy.

Now, consider the blow-up problem (6) with $p = 5$ and the boundary conditions $u(0,t) = u(1,t) = 0$ and the initial condition $u(x,0) = 20\sin(\pi x)$. This differential equation

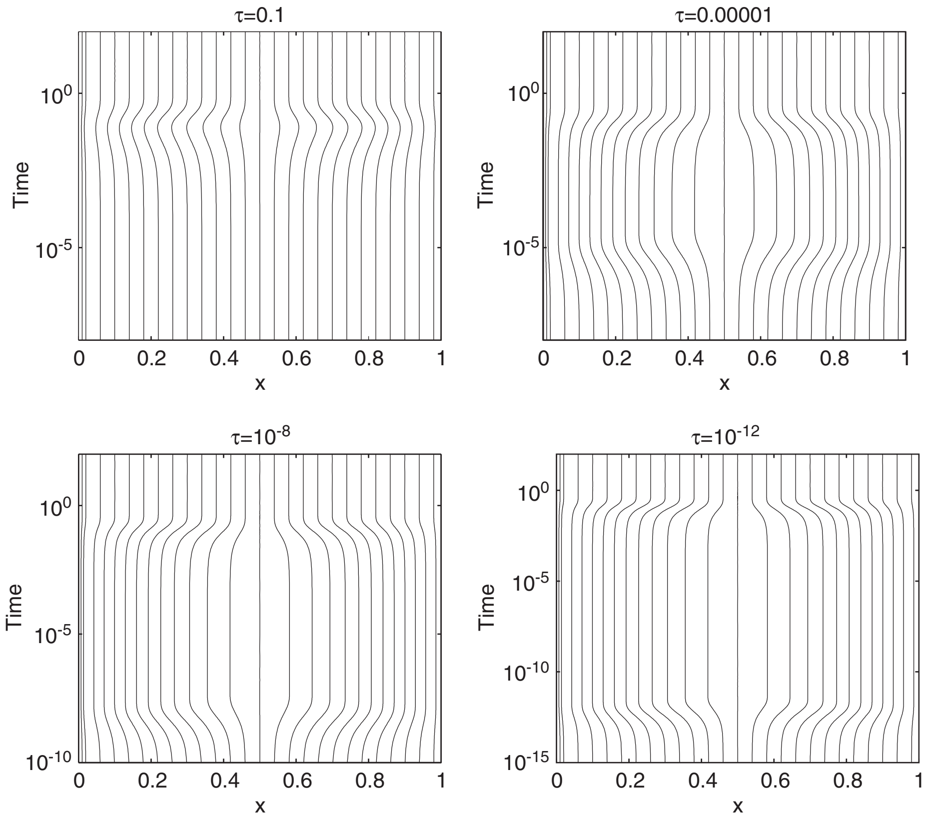

Fig. 1. Mesh trajectories with different values of τ obtained with MMPDE6. The monitor function is the arclength function.

is transformed into the computational domain to obtain

$$\dot{u} - \frac{u_\xi}{x_\xi}\dot{x} = \frac{1}{x_\xi}\left(\frac{u_\xi}{x_\xi}\right)_\xi + u^5. \tag{9}$$

After discretizing in space only, the resulting ODE system is solved using the Matlab routine "ode15s". We use tolerances $rtol = atol = 10^{-8}$, and implement a dynamically changing relaxation time through the use of option "OutputFcn" set with the "odeset" command in Matlab. The discrete form of Eq. (9) is coupled with discrete moving mesh equations (MMPDE4 or MMPDE6), and the smoothed monitor function $\tilde{M}$ is used.

Three different monitor functions are used, including the arclength monitor

$$M_1(x,t) = \sqrt{1 + u_x^2},$$

the monitor function derived analytically by [2]

$$M_2(x,t) = u^{p-1},$$

and the new monitor function

$$M_3(x,t) = u^{(p-1)/2}.$$

Numerical results are presented with the two different relaxation times $\tau_1(t)$ and $\tau_2(t)$, as well as a fixed value of $\tau = 10^{-5}$.

The computed CPU time and blow-up time are displayed in Fig. 2 for different values of N. To obtain an estimate of the actual blow-up time T appearing in Eq. (7) for $p = 5$, we performed a highly resolved calculation to obtain $T \approx 1.56259 \times 10^{-6}$. We can conclude that $M_3(x,t) = u^{(p-1)/2}$ yields a better approximation of the solution blow-up. For the arc-length monitor function with $\tau_1(t)$, the blow-up time is significantly less accurate than the other cases. Except for the arc-length monitor function with $\tau_2(t)$, we can say that there is an approximately linear increase in CPU time with number of mesh points N.

The legend for the second plot of Figs. 2 and 3 are the same as the first plot of Fig. 2. Fig. 3 indicates that the value of $u_{\max}$ computed with the new monitor function $M_3(x,t)$ and relaxation time $\tau_1(t)$ is significantly bigger than for the other monitor functions, which can be interpreted as having higher accuracy, since the problem can be integrated further into the blow-up. Notice also that $u_{\max}$ is approximately constant as N is increased.

The computations also show that both the blow-up time and blow-up point be effected by the initial conditions. By the scaling transformation [7,13], we expect the solution to have a symmetric peck centered on x^*. Finally, we present a plot of $u/u_{\max}$ as a function of ξ in Fig. 4. We have plotted the results at time slices when $u_{\max} = 10^k$, for $k = 8, 9, \ldots, 15$. In this figure, MMPDE6 is used for the mesh equation and the monitor function is $M_3(x,t)$. This figure shows that the self-similar behaviour of the blow-up solution about the point $x^* = \frac{1}{2}$ is captured quite accurately; otherwise, the resolution of the adaptive mesh would degrade and the solution would degenerate to only a few points within the blow-up region.

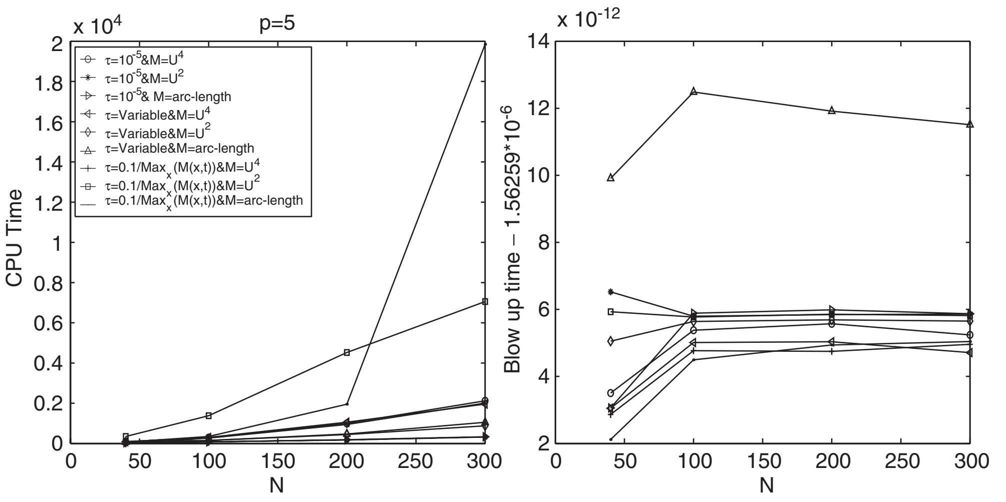

Fig. 2. Variation of the CPU time and blow-up times with the number of mesh points N, choosing different monitor functions $M(x,t)$ and relaxation times τ.

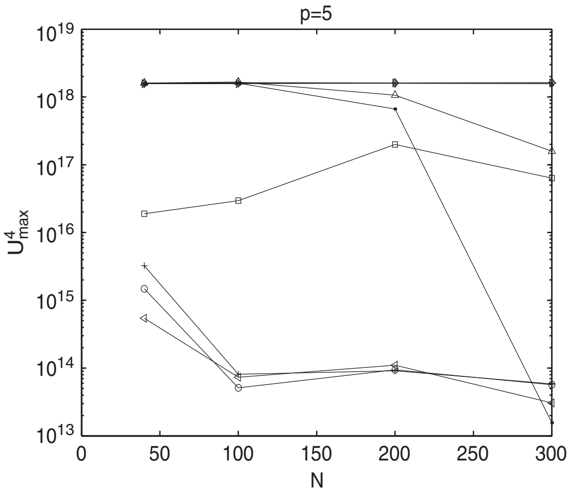

Fig. 3. The value $\max_x u(x,t)^4$ when t tends to the blow-up time. Results are reported for number of mesh points N, and choosing the same monitor functions $M(x,t)$ and relaxation times τ as labelled in Fig. 2.

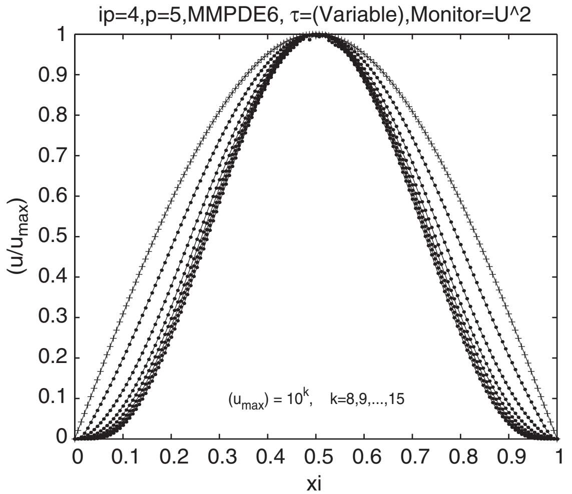

Fig. 4. MMPDE6 with smoothed monitor function ($M_3(x,t)$ with $ip = 4$) and variable relaxation time τ_1. The value of $u/\max(u)$ is plotted as a function of ξ for the times corresponding to when $u_{\max} = 10^k$ for $k = 8, 9, \ldots, 15$. The first curve (denoted with a '+') is the initial condition $u(x,0)/\max(u(x,0))$.

6. Conclusion

In this paper, we have considered one class of blow-up problem and applied the moving mesh method with variable relaxation time. We have applied two forms of the relaxation times Eqs. (7) and (8), the latter of which is suitable for general purpose simulations. Also, a new monitor function $M_3(x,t) = u^{(p-1)/2}$ was defined and the results compared with the arc-length and $M_2(x,t) = u^{p-1}$ monitor functions. In our numerical experiments with the new monitor function, preserve scaling invariance property and a blow-up problem may be integrated further than otherwise possible.

References

[1] W. Huang, Y. Ren, R.D. Russell, Moving mesh partial differential equations (MMPDEs) based on the equidistribution principle, SIAM J. Numer. Anal. 31 (1994) 709–730.
[2] W. Huang, Y. Ren, R.D. Russell, Moving mesh methods based on moving mesh partial differential equations, J. Comput. Phys. 113 (1994) 279–290.
[3] J.M. Sanz-Serna, I. Christie, A simple adaptive technique for nonlinear wave problems, J. Comput. Phys. 67 (1986) 348–360.
[4] J.M. Stockie, J.A. Mackenzie, R.D. Russell, A moving mesh method for one-dimensional hyperbolic conservation laws, SIAM J. Sci. Comput. 22 (2001) 1791–1813.
[5] S. Li, L. Petzold, Moving mesh methods with upwinding schemes for time-dependent PDEs, J. Comput. Phys. 131 (1997) 368–377.
[6] R.M. Furzeland, J.G. Verwer, P.A. Zegeling, A numerical study of three moving grid methods for 1D PDEs which are based on the method of lines, J. Comput. Phys. 89 (1990) 349–399.
[7] C.J. Budd, W. Huang, R.D. Russell, Moving mesh methods for problems with blow-up, SIAM J. Sci. Comput. 17 (1996) 305–327.
[8] C.J. Budd, R. Carretero-González, R.D. Russell, Precise computations of chemotactic collapse using moving mesh methods, J. Comput. Phys. 202 (2005) 463–487.
[9] C.J. Budd, J. Chen, W. Huang, R.D. Russell, Moving mesh methods with applications to blow-up problems for PDEs, in: D.F. Griffiths, G.A. Watson (Eds.), Numerical Analysis 1995: Proceedings of 1995 Biennial Conference on Numerical Analysis, Pitman Research Notes in Mathematics, Longman Scientific and Technical, 1996, pp. 1–17.
[10] W. Huang, R.D. Russell, Analysis of moving mesh partial differential equations with spatial smoothing, SIAM J. Numer. Anal. 34 (1997) 1106–1126.
[11] J. Bebernes, D. Eberly, Mathematical Problems for Combustion Theory, Applied Mathematics Science, vol. 83, Springer, Berlin, 1986.
[12] J. Coyle, J. Flaherty, R. Ludwig, On the stability of mesh equidistribution strategies for time-dependent partial differential equations, J. Comput. Phys. 62 (1986) 26–39.
[13] C.J. Budd, V.A. Galaktionov, Stability and spectra of blow-up in problems with quasi-linear gradient diffusivity, Proc. R. Soc. London A 454 (1998) 2371–2407.