On the mesh relaxation time in the moving mesh method

John Stockie

http://www.math.sfu.ca/~stockie/

Department of Mathematics

Simon Fraser University

August 3, 2005

Acknowledgments

Joint work with:

Ali Reza Soheili
University of Sistan & Baluchestan
Zahedan, Iran

Funding was provided by:

NSERC CRSNG

Outline

- Background: MMPDE's
- Motivation: Self-similar blow-up
- Numerical simulations
- Future work

Background: Moving mesh PDE's

Moving mesh method

- $x =$ physical coordinate, $\quad \xi =$ computational coordinate
- One-to-one mapping: $x(\xi, t)$, $\xi \in [0, 1]$
- Monitor function: $M(x, t) > 0$
- Equidistribution Principle (EP, integral form):

$$\int_0^{x(\xi,t)} M(x', t)\, dx' = \xi\, \theta(t) \quad \text{where} \quad \theta(t) = \int_0^1 M(x', t)\, dx'$$

- Equidistribution Principle (EP, differential form):

$$\frac{\partial}{\partial \xi}\left(M \frac{\partial x}{\partial \xi} \right) = 0$$

Moving mesh PDE's

Huang, Ren & Russell (1994):

- Equidistribution Principle

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

Moving mesh PDE's

Huang, Ren & Russell (1994):

- Equidistribution Principle

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

- Introduce a **relaxation time $\boldsymbol{\tau}$**

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

Moving mesh PDE's

Huang, Ren & Russell (1994):

- Equidistribution Principle

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

- Introduce a **relaxation time $\boldsymbol{\tau}$**

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

- Expand in Taylor series:

(MMPDE4) $$\frac{\partial}{\partial \xi}\left(M\frac{\partial \dot{x}}{\partial \xi}\right)=-\frac{\mathbf{1}}{\boldsymbol{\tau}}\frac{\partial}{\partial \xi}\left(M\frac{\partial x}{\partial \xi}\right)$$

(MMPDE6) $$\frac{\partial^2 \dot{x}}{\partial \xi^2}=-\frac{\mathbf{1}}{\boldsymbol{\tau}}\frac{\partial}{\partial \xi}\left(M\frac{\partial x}{\partial \xi}\right)$$

Mesh relaxation time

- While a large effort has been expended on developing better monitor functions, almost no attention has been paid to the choice of τ
- τ is identified as an important parameter **BUT** it's
  - always taken to be constant
  - tuned by trial-and-error for a given problem
- τ can be interpreted in several ways:
  - a relaxation time for the mesh to satistfy the EP
  - temporal smoothing
  - a damping factor – **Adjerid & Flaherty (1986)**
  - a delay factor – **Furzeland et al. (1990)**

Mesh relaxation time (2)

In practice, the choice of τ is a trade-off:

- τ must be large enough to avoid oscillations in the mesh (stability)
- τ must be small enough that mesh can respond to changes in the solution (accuracy)

	accuracy	*stiffness / cost*
small $\boldsymbol{\tau}$	increased	more stiff / higher cost
large $\boldsymbol{\tau}$	decreased	less stiff / lower cost

Basic idea

- For problems with complex behaviour (esp. time variations on fast and slow scales) choosing a single constant value of τ seems inappropriate
- Instead, we want

mesh time scale $\approx$ solution time scale

- Examples:

a. *blow-up:* initial rapid motion of mesh points – **Budd, Huang & Russell (1996)**

b. *moving fronts:* with variable front speed – **JS, Mackenzie & Russell (2001)**

c. *Gierer-Meinhardt:* very slow spike motion, with rapid, spontaneous changes – **Iron & Ward (2002)**

Aim: Increase accuracy and efficiency by varying $\tau(t)$ throughout a computation

Blow-up problems

A simple blow-up model

- One of the simplest models for blow-up is (for $p > 1$)

$$u_t = u_{xx} + u^p \qquad \text{subject to} \qquad \begin{aligned} u(0,t) &= u(1,t) = 0 \\ u(x,0) &= u_0(x) \end{aligned}$$

- The solution “blows up” at $x = x^*$ and $t = t^*$ if:

$$u(x^*, t) \to \infty \quad \text{as } t \to t^*$$
$$\text{and } \; u(x,t) \to u(x,t^*) < \infty \quad \text{if } x \neq x^*$$

- There is a similarity solution with asymptotic behaviour

$$u(x,t) \longrightarrow \beta^\beta (t^* - t)^{-\beta} \left(1 + \frac{\mu^2}{4p\beta}\right)^{-\beta} \quad \text{as } t \to t^*$$

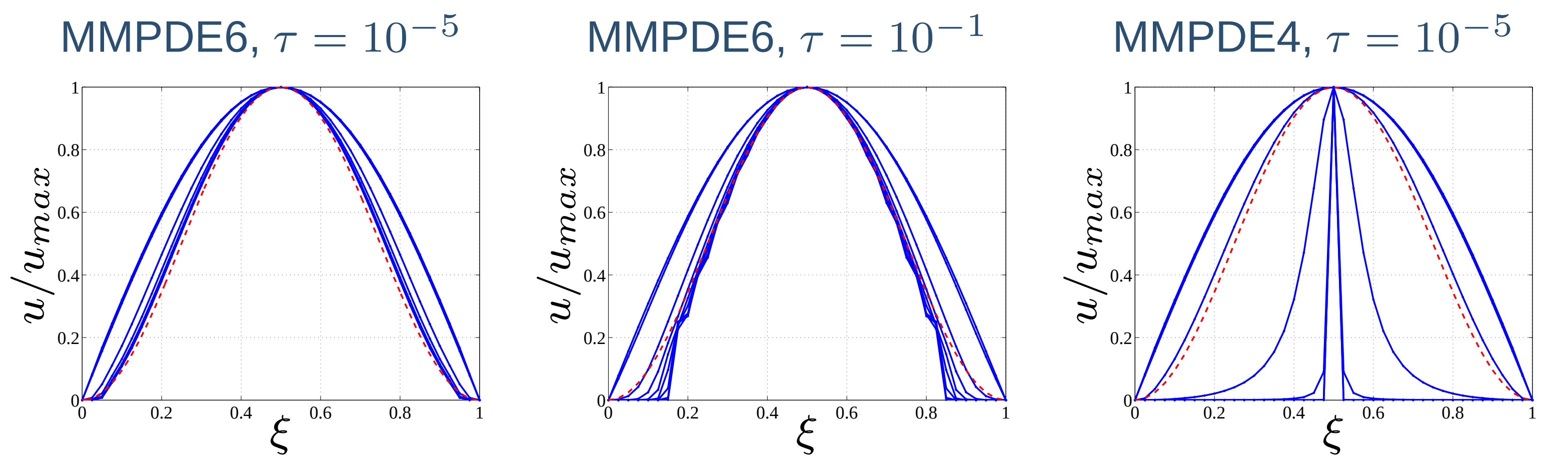
where $\beta = \frac{1}{p-1}$ and μ is the *ignition kernel*

– Bebernes & Bricher (1992)

Moving mesh calculations of blow-up

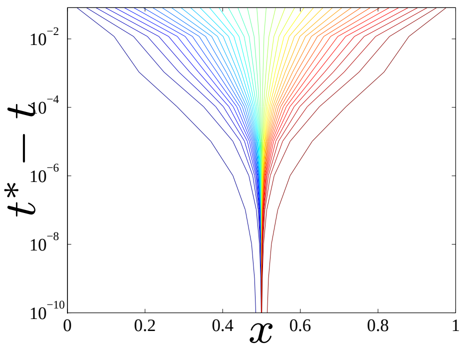
Budd, Huang & Russell (1996):

- Used the MMPDE approach to compute self-similar solutions (fixed mesh calculations are pointless!)
- Derived the monitor $M = u^{p-1}$ needed to capture self-similarity
- With $p = 2$ and $N = 40$ points:

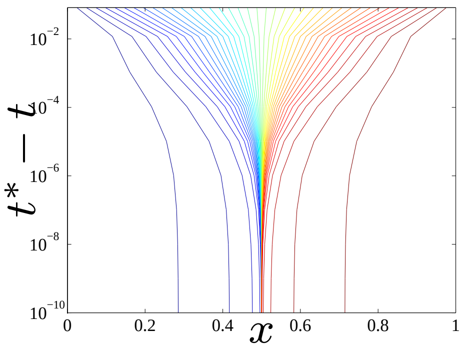


Moving mesh calculations of blow-up (2)

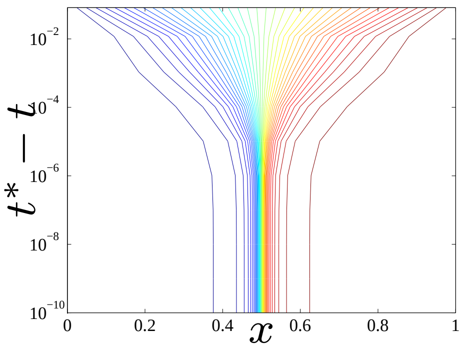
MMPDE6, $\tau = 10^{-5}$



MMPDE6, $\tau = 10^{-1}$



MMPDE4, $\tau = 10^{-5}$



Provided τ is taken small enough (e.g., $\tau = 10^{-5}$):

- both the computed solution and mesh capture self-similarity
- blow-up behaviour can be computed accurately

Choice of τ

Some insight is afforded by a scaling argument in BHR'96:

- The solution and monitor satisfy:

$$u \sim (t^* - t)^\beta \quad \text{and} \quad M = u^{p-1} \sim (t^* - t)$$

- The mesh has a natural timescale:

$$T_{mesh} = O(\tau) \qquad \text{(for MMPDE4)}$$
$$T_{mesh} = O(\tfrac{\tau}{M}) \sim \tau(t^* - t) \qquad \text{(for MMPDE6)}$$

- **Conclude:** MMPDE6 with $\tau = 10^{-5}$ (constant) allows the mesh to evolve even for t close to t^*, but MMPDE4 does not

Choice of τ

Some insight is afforded by a scaling argument in BHR'96:

- The solution and monitor satisfy:

$$u \sim (t^* - t)^\beta \quad \text{and} \quad M = u^{p-1} \sim (t^* - t)$$

- The mesh has a natural timescale:

$$T_{mesh} = O(\tau) \qquad \text{(for MMPDE4)}$$

$$T_{mesh} = O(\tfrac{\tau}{M}) \sim \tau(t^* - t) \qquad \text{(for MMPDE6)}$$

- **Conclude:** MMPDE6 with $\tau = 10^{-5}$ (constant) allows the mesh to evolve even for t close to t^*, but MMPDE4 does not

Our claim: τ need not always be so small (i.e., the mesh equation is unnecessarily stiff). A more sensible choice is:

$$\tau = \tau_o \max_x M \qquad \text{with } \tau_o \ll 1$$

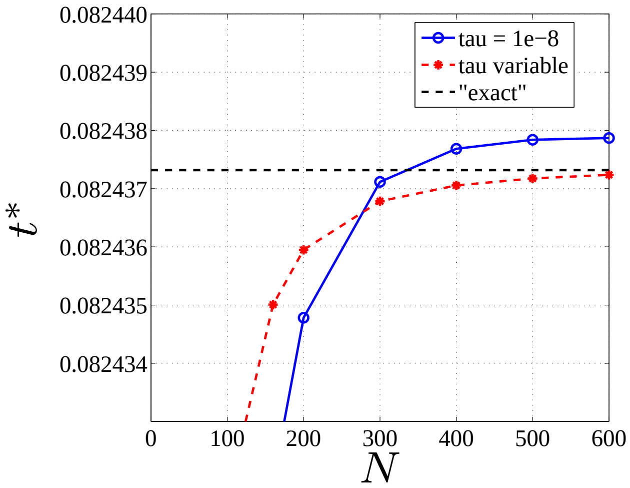
Numerical simulations

Variable τ results

- Take $p = 2$, MMPDE6, MOL + DASSL
- $\tau(t) = \tau_o \max_x M$, force $\tau \in [10^{-8}, 10^{-1}]$
- Compare to simulation with constant $\tau = 10^{-8}$

Variable τ results demonstrate:

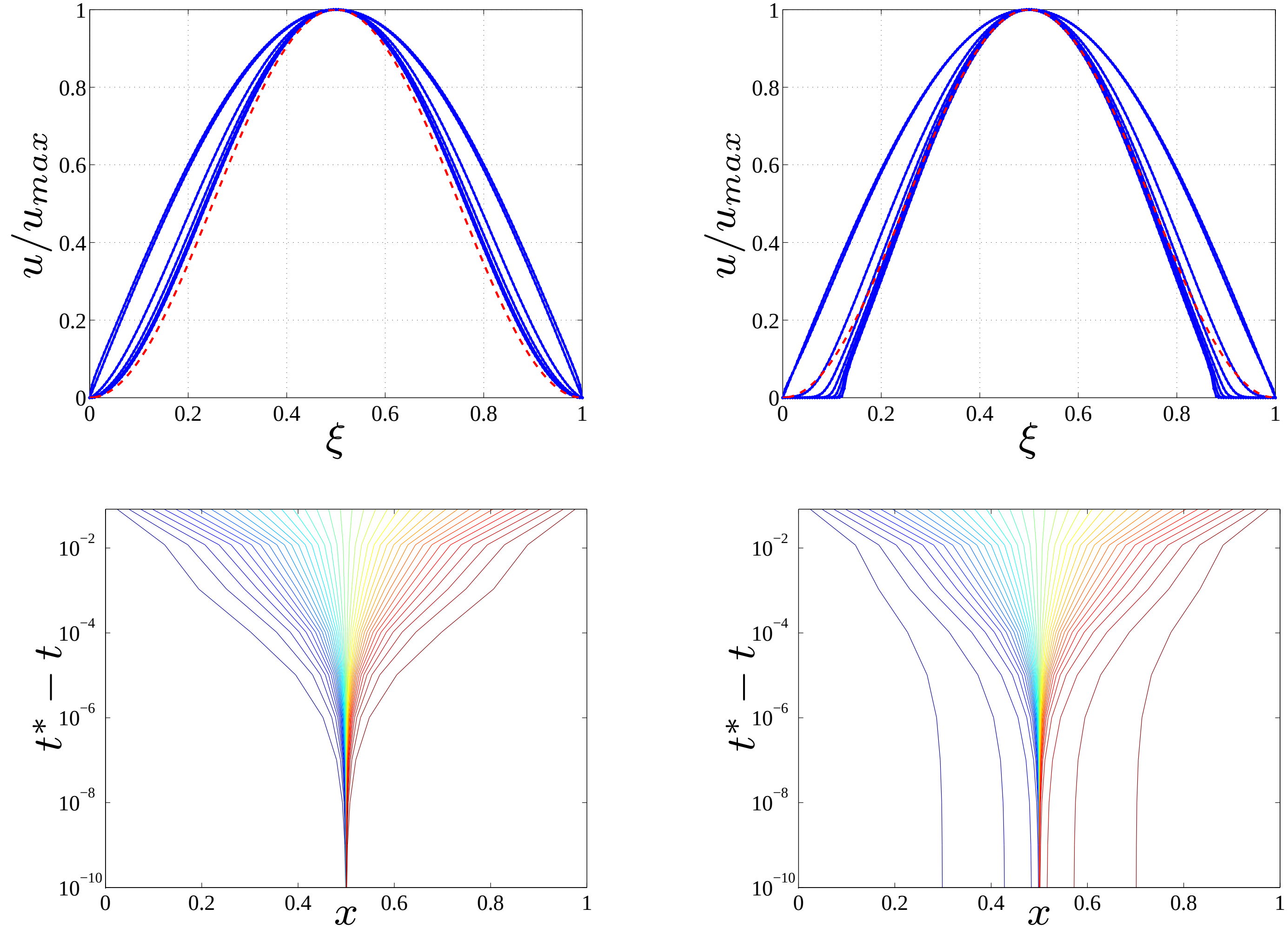
- CPU time is reduced by at least a factor of three
- $\max_x u$ is at least 3 orders of magnitude larger (computes further into blow-up)
- t^* is computed more accurately



Variable τ results (2)

With $N = 200$ points:

$\tau = 10^{-8}$ | τ variable

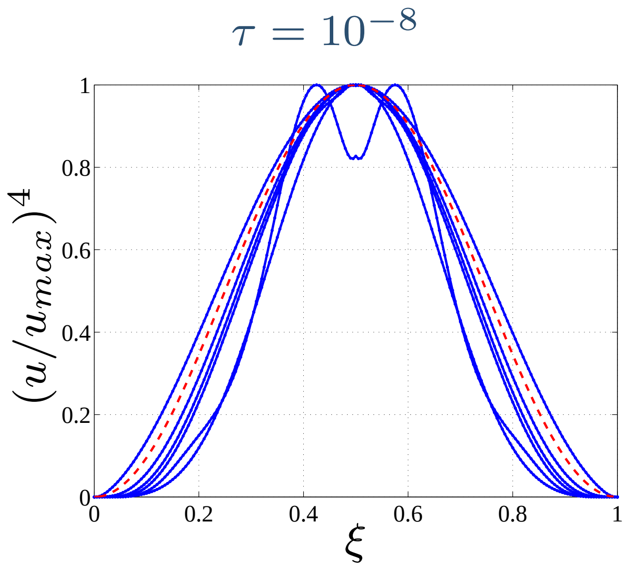


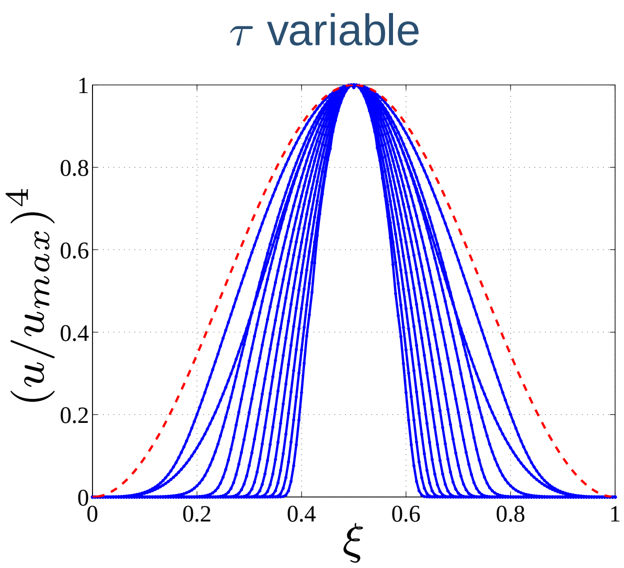
Variable τ results (3)

Summary of results:

- τ needs to be small only during the time leading up to blow-up, as mesh points race into the peak
- For t closer to t^*, it's sufficient to take $\tau = \tau_{max} = 0.1$
- Variable τ improves both accuracy and efficiency, but leads to a slight loss of self-similarity

More severe blow-up ($p = 5$)





- Non-physical oscillations appear in constant τ results, but not with variable τ
- Similar improvement in efficiency
- Variable τ results show a more significant deviation from self-similarity

Conclusions

- Choosing mesh relaxation parameter τ constant is not optimal
- Significant improvements in accuracy and cost can be obtained by varying τ sensibly for blow-up problems
- We need another approach for calculating τ adaptively in general situations

Future work

- Determine a more general form of $\tau(t)$ which is robust and applicable to other problems
  – see **Hyman & Larrouturou (1989)**, **W. Huang (2001)**
- More extensive studies involving other PDE's
- Analytical investigation, going back to the EP

References

[1] S. Adjerid and J. E. Flaherty. A moving finite element method with error estimation and refinement for one-dimensional time dependent partial differential equations. *SIAM J. Numer. Anal.*, 23(4):778–796, 1986.

[2] J. Bebernes and S. Bricher. Final time blowup profiles for semilinear parabolic equations via center manifold theory. *SIAM J. Math. Anal.*, 23(4):852–869, 1992.

[3] C. J. Budd, W. Huang, and R. D. Russell. Moving mesh methods for problems with blow-up. *SIAM J. Sci. Comput.*, 17(2):305–327, 1996.

[4] R. M. Furzeland, J. G. Verwer, and P. A. Zegeling. A numerical study of three moving grid methods for one-dimensional partial differential equations which are based on the method of lines. *J. Comput. Phys.*, 89(2):349–388, 1990.

[5] W. Huang. Practical aspects of formulation and solution of moving mesh partial differential equations. *J. Comput. Phys.*, 171(2):753–775, 2001.

[6] W. Huang, Y. Ren, and R. D. Russell. Moving mesh partial differential equations (MMPDES) based on the equidistribution principle. *SIAM J. Numer. Anal.*, 31(3):709–730, 1994.

[7] J. M. Hyman and B. Larrouturou. Dynamic rezone methods for partial differential equations in one space dimension. *Appl. Numer. Math.*, 5:435–450, 1989.

[8] D. Iron and M. J. Ward. The dynamics of multispike solutions to the one-dimensional Gierer-Meinhardt model. *SIAM J. Appl. Math.*, 62(6):1924–1951, 2002.

[9] J. M. Stockie, J. A. Mackenzie, and R. D. Russell. A moving mesh method for one-dimensional hyperbolic conservation laws. *SIAM J. Sci. Comput.*, 22(5):1791–1813, 2001.