Case Study: A Visual Tool for Moving Mesh Numerical Methods

Daryl H. Hepting * daryl h@cs.sfu.ca
Weizhang Huang † whuang@math.ukans.edu
Robert D. Russell ‡ rdr@cs.sfu.ca

Abstract

Time-dependent partial differential equations are indispensable in scientific and engineering research, where they are used to model physical phenomena such as fluid flow and combustion. Moving mesh numerical solution methods have proven to be particularly well-suited to compute solutions to such problems which change non-uniformly over time. Computer visualization of moving mesh methods is an extension of an approach using computers to study mathematics, first advocated by von Neumann. Uses of this visualization include verifying the correct operation of the numerical algorithm; providing insight into improvements to the numerical algorithm; and interpreting the operation of the numerical algorithm to a larger community of mathematics researchers.

1 Introduction

Traditionally, numerical methods for solving time dependent partial differential equations have used a fixed spatial mesh and often a fixed time step. Such fixed mesh methods have a natural geometric interpretation which is rather static and comprehensible. In order to more accurately and reliably arrive at a solution for the underlying physical problem, variable time steps were introduced to minimize numerical error. Currently so-called adaptive methods, which also vary the spatial mesh, are of great interest. For these methods, the mesh points are allowed to move in order to follow time-dependent features. These features are particularly troublesome for nonlinear problems in areas like fluid mechanics and combustion.

Although moving mesh methods also have a natural geometric interpretation, the added dynamic of the moving mesh makes visualization an invaluable tool in the study of these methods. This visualization can facilitate concrete understanding of the operation of the moving mesh methods, apart from the equations which it is used to solve. Such an understanding is important to validate the algorithm and lend confidence to its use for new problems; provide insight leading in turn to improvements to the algorithm; and effectively illustrate the algorithm to the larger research community.

In this paper we review related work and briefly describe the problem under study, describe the graphical system used to visualize the moving mesh method, describe two numerical examples, and indicate directions for future work.

2 Background

The notion that pictures can aid in mathematical discovery and understanding occurs throughout history. It is best exemplified by considering the oldest areas of mathematics, geometry and nonlinear dynamics. Kepler's discovery of the elliptic planetary orbits is a prime example of the intertwining of visualization with early nonlinear dynamics which also provided the motivation for the development of Newton's calculus.

For mathematicians, the advent of powerful computers meant that problems previously intractable by analytic techniques could now be tackled numerically. More than that, the experimental approach to mathematics which von Neumann [8] espoused – thorough computational studies of well-chosen problems intended to "break the deadlock" in work on many complicated nonlinear problems – became possible.

The enormous increase in power of computers since von Neumann wrote on the subject has in many respects not seen a commensurate improvement in understanding of nonlinear problems. With the advent of computer graphics as a sophisticated tool for visualization, the possibility of von Neumann's approach is extended to more fully include the visual sense (also exemplified by the "synergetic" approach of Zabusky *et al.* [10]). Computational fluid dynamics (CFD) is one area which has benefitted greatly from computer-graphical tools (see [9] for example), which allow a visual interpretation which would otherwise not be possible.

*School of Computing Science, Simon Fraser University, Burnaby, British Columbia, CANADA V5A 1S6

†Department of Mathematics, University of Kansas, Lawrence, Kansas, USA

‡Department of Mathematics and Statistics, Simon Fraser University, Burnaby, British Columbia, CANADA V5A 1S6

At the core of almost any sophisticated mathematical model in science and engineering is a nonlinear partial differential equation (PDE). Many of the fundamental questions for PDEs involve simple problems posed in only one space dimension. Two classic examples of time-dependent problems, which serve as examples in this paper, are the well-known Burgers' equation

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(\epsilon \frac{\partial u}{\partial x} - \frac{u^2}{2}\right). \tag{1}$$

and

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + u^2 \tag{2}$$

which is representative of so-called blow-up problems which arise in modelling combustion.

The nonlinearities in these problems are manifested by regions of very sharp solution variation which move over time. The moving mesh methods described by Huang *et al.* [5] are particularly well-suited to equations like (1) and (2) above, because they adapt well to large variations in the solutions over time. In fact, methods which are not adaptive will often fail for these equations.

Traditionally, mathematical software for these time-dependent PDEs has been based on numerical methods which perform spatial adaptation of the mesh at a fixed time level, with a high resulting overhead. Although the fully adaptive moving mesh methods are in principle more appealing and potentially much more efficient, early versions were met with only mixed success. A lack of theoretical underpinning has plagued the development of these methods, and consequently there has been considerable controversy about their true potential.

The approach of using moving mesh PDEs (MMPDEs) provides the desired theoretical framework and justification for these adaptive methods. A coordinate transformation is used to determine the computational coordinate system for which the solution to the PDE is computed on a uniform mesh (which is a moving mesh in the physical coordinate). Figure 1 illustrates this mapping. One consequence of the MMPDE approach is that the problem of solving a PDE becomes that of computing both the solution to the PDE *and* this coordinate transformation. The solution of the MMPDE is used to position the mesh points for the evaluation of the physical PDE. In general, the MMPDE solution yields a mesh in which some measure of the physical error in the solution has been equidistributed. The error in this case is measured by a so-called monitor function. There may be

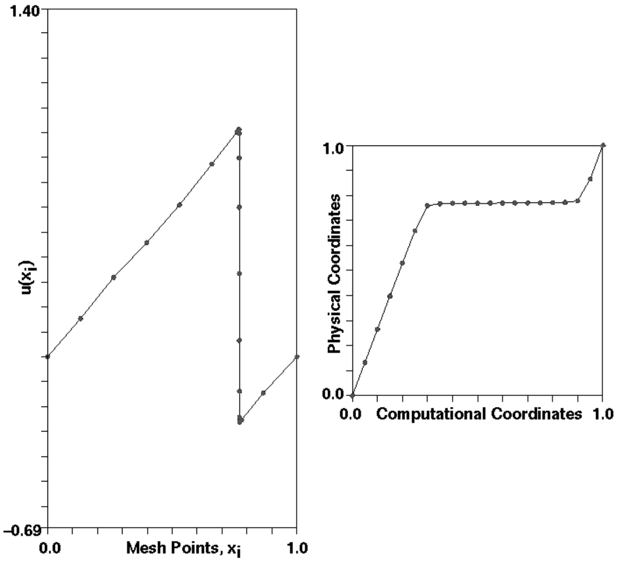

Figure 1: The left image depicts a solution curve for Burgers' equation at particular time t_j with the location of mesh points indicated. The right image shows the mapping of computational coordinates to physical coordinates.

a choice of appropriate MMPDE's and monitor functions, depending on the characteristics of the problem to be solved.

3 Visualization

This work represents a first attempt to develop a visual tool for the analysis and development of moving mesh numerical methods, which also facilitates communication about and exploration of the key issues related to these methods. While significant knowledge of these key issues came with the development of the numerical methods through the computational experimentation conducted by Huang *et al.* [5], that knowledge has been greatly augmented with the concurrent development of comprehensible visual forms in the context of this tool.

The tool design corresponds closely to the structure outlined by Haber and McNabb [4]. Under their classification scheme, this tool adheres to the post-simulation analysis approach. After computation has completed, a history of events comprising the solution values and mesh positions at each time step is used as the basis of the visualization. The contents of the computational history may change based on an iterative process of computation and visualization influenced by discovered trends and the needs of the researcher.

A tool for the communication and exploration of moving mesh methods requires a wide variety of representations and mappings for many display quantities to encourage understanding and the development of insights, through a process which Finke [3] calls "combinational play".

Multiple representations are used to provide the researcher with different physical and conceptual views of the computational history. The representations include:

- physical and computational coordinate solution trajectories: plot of the function $u(x)$ for all mesh points x_i in physical and computational coordinates.
- mesh trajectory: plot of the mesh points x_i for all times t_j of the computation.
- coordinate transformation: plot of the computational coordinates versus the physical coordinates.
- physical and computational coordinate solution surface: plot of the solution trajectories (in either physical or computational coordinates) through time to form a surface.

A key feature of this tool is its use of several simple views in a highly coordinated fashion [1, 7, 6]. Information about the currently selected time step and mesh point is passed to all views to create a more meaningful display in which cutting planes and other devices identify common points of reference within disparate views.

Various display quantities can be derived from the computational history and by mapping them onto different representations, they can be used to add insight about the algorithm performance. Many such quantities are possible, but the following are of particular interest:

- quasi-uniformity: a measure of the mesh uniformity taken by finding the maximum of ratios of adjacent mesh gap sizes.
- node speed: the rate of mesh movement.
- arc length: the distance between mesh points along a particular solution curve.

4 Examples

4.1 Burgers' equation

Burgers' equation is well-known since its variety of solution behaviors often makes it an ideal test problem for numerical schemes. It is the simplest mathematical formulation of the competition between convection and diffusion and its study advocated by von Neumann [8] because it captures many properties of turbulence in a simple form.

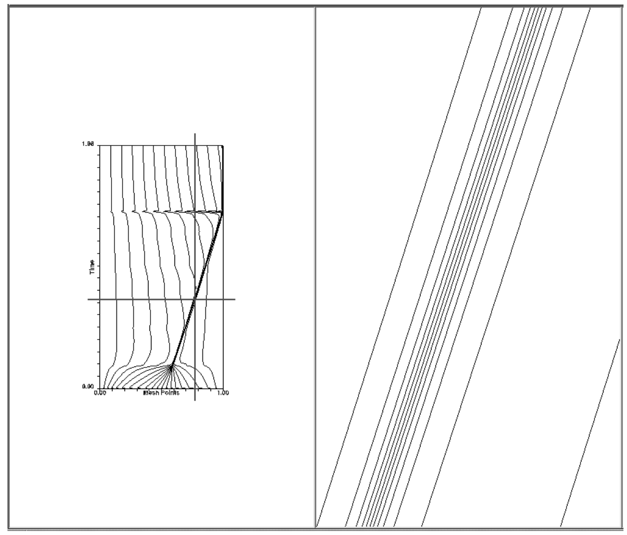

Figure 2: The left image shows a global perspective on the mesh trajectories, in which the cross-hair indicates the desired point of focus. The right image shows a magnification, done interactively, about that point to verify that the mesh points do not cross.

The effectiveness of the moving mesh method is judged here by its ability to follow the shock wave which develops in the solution. Because many mesh points gather at nearly the same positions along the shock wave to faithfully capture its behaviour, as illustrated in Figure 1, the possibility of mesh crossings may seem to exist. Indeed, such crossings were a difficulty of early attempts to define a moving mesh method. Figure 2 shows an interactive magnification facility in the tool which allows immediate verification of the property that mesh points do not cross.

The mesh trajectories for Burgers' equation (see Figure 3a) indicate the rightward movement of the mesh as it captures the shock wave. However, the behaviour of individual mesh points throughout all time steps is not clear because of the concentration of mesh points on the shock wave. Greatly improved clarity is achieved by adding colour to uniquely identify each gap between adjacent mesh points (see Figure 3b). The new figure clearly shows how mesh points are captured by the shock wave and a question arises naturally about whether the number of mesh points used in computation can be decreased in these situations.

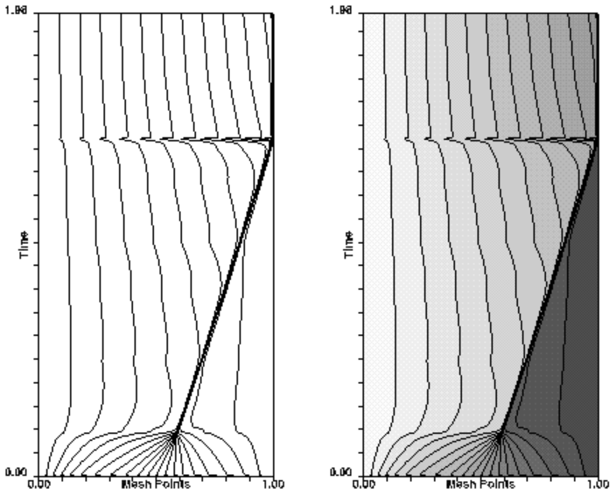

Figure 3: Both images in this figure show the mesh trajectories, but the figure on the left does not give a sense of the individual mesh node behaviour over time. With the use of colour in the right image, it is possible to gain a clear sense of the position of each mesh node over time.

4.2 Blow-up problem

Blow-up problems have a scaling invariance which provides an elegant means to describe the underlying solution structure. In this challenging numerical problem, moving mesh methods are essential to capture the solution behaviour in the blow-up region. MMPDEs and monitor functions are chosen to preserve the scaling invariance of the underlying problem in such a way that close to the blow-up point mesh points are placed automatically so that the ignition kernel, which is well known to be a natural coordinate in describing the behavior of blow-up, approaches a constant. This scaling invariance property is easily verified through visualization, illustrated in Figure 4.

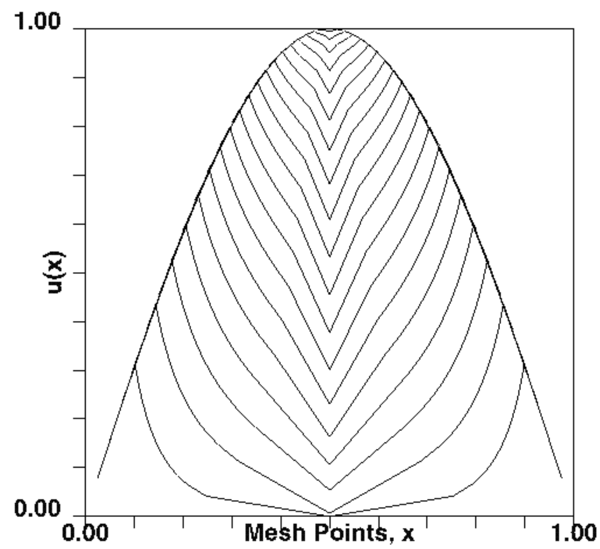

Figure 4: Scaling invariance of the blow-up problem is shown by plotting the mesh node trajectories against function values, computed as a ratio with the maximum value of u at each time step j ($\frac{u(x_i)}{\max(u_j)}$). Notice how the shape of the graph is maintained over a variety of scales as $max(u_j)$ blows up. It is also possible to see the movement of the mesh nodes to the point of blow up.

5 Conclusions

The visualization tool described herein has provided a clear means of communicating the performance of the moving mesh methods of Huang *et al.* [5]. This alone will prove valuable in furthering understanding of these numerical methods for time-dependent PDEs.

Insights about these well-known problems have also come from this visualization tool, providing interesting avenues for future research.

Visualization allows interesting qualitative features of the solution to be identified and investigated. As Zabusky *et al.* [10] and Budd *et al.* [2] state, algorithms which work well for known problems allow analytically uncharted territory to be explored with greater confidence. This new territory includes other problems in one dimension and problems in higher space dimensions. With the extension to higher dimensions, new challenges for the visualization will also need to be met.

References

[1] M. H. Brown and J. Hershberger. Color and sound in algorithm animation. *Computer*, 25(12):52–63, 1992.

[2] C. J. Budd, W. Huang, and R. D. Russell. Moving mesh methods for problems with blow-up. *SIAM Journal on Scientific Computing*, to appear.

[3] R. Finke. *Creative Imagery: Discoveries and Inventions in Visualization.* Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1990.

[4] R. B. Haber and D. A. McNabb. Visualization idioms: A conceptual model for scientific visualization systems. In *Visualization in Scientific Computing*, pages 74–93. IEEE Computer Society Press, 1990.

[5] W. Huang, Y. Ren, and R. D. Russell. Moving mesh partial differential equations based on the equidistribution principle. *SIAM Journal on Numerical Analysis*, 31:709–730, 1994.

[6] K. L. Norman, L. J. Weldon, and B. Shneiderman. Cognitive layouts of windows and multiple screens for user interfaces. *Int. J. Man-Machine Studies*, 25:229–248, 1986.

[7] M. Visvalingam and J. D. Whyatt. The Douglas-Peucker algorithm for line simplification: Re-evaluation through visualization. *Computer Graphics Forum*, 9:213–228, 1990.

[8] John von Neumann. Recent theories of turbulence. In A. Taub, editor, *Collected Works of John von Neumann*, volume 6. MacMillan, New York, 1963.

[9] N. J. Zabusky and D. Silver. Case study: Visualizing classical problems in cfd. In A. E. Kaufman and G. M. Nielson, editors, *Proceedings of Visualization '92*, pages 436–440, 1992.

[10] N. J. Zabusky, D. Silver, R. Pelz, et al. Visiometrics, juxtaposition and modeling. *Physics Today*, 46(3):24–31, 1993.