Numer. Math. Theor. Meth. Appl.
doi: 10.4208/nmtma.2016.m1508

Vol. **9**, No. 3, pp. 432-450
August 2016

Mesh Spacing Estimates and Efficiency Considerations for Moving Mesh Systems

Joan Remski*

Department of Mathematics and Statistics, The University of Michigan-Dearborn, Dearborn, MI 48128, U.S.A.

Received 12 April 2015; Accepted 31 July 2015

Abstract. Adaptive numerical methods for solving partial differential equations (PDEs) that control the movement of grid points are called moving mesh methods. In this paper, these methods are examined in the case where a separate PDE, that depends on a monitor function, controls the behavior of the mesh. This results in a system of PDEs: one controlling the mesh and another solving the physical problem that is of interest. For a class of monitor functions resembling the arc length monitor, a trade off between computational efficiency in solving the moving mesh system and the accuracy level of the solution to the physical PDE is demonstrated. This accuracy is measured in the density of mesh points in the desired portion of the domain where the function has steep gradient. The balance of computational efficiency versus accuracy is illustrated numerically with both the arc length monitor and a monitor that minimizes certain interpolation errors. Physical solutions with steep gradients in small portions of their domain are considered for both the analysis and the computations.

AMS subject classifications: 65M50, 65M06

Key words: Numerical solution, partial differential equation, moving mesh method, monitor function.

1. Introduction

Moving mesh methods are a class of numerical techniques for solving partial differential equations (PDEs) where the mesh is allowed to evolve in order to better approximate the solution. The idea is to adaptively move grid points to places in the domain where the solution changes in a significant way, leaving fewer mesh points in regions where the solution does not change very much. The location of the grid points is determined by a mapping from a computational domain, Ω_C, into the physical domain where the PDE is originally defined, Ω_P ([1, 2]). There are several methods for determining the mesh mapping, but in this paper we focus on using a monitor function that

*Corresponding author. *Email address:* remski@umich.edu (J. Remski)

 ©2016 Global-Science Press

https://doi.org/10.4208/nmtma.2016.m1508 Published online by Cambridge University Press

defines some measure of error, with the mesh mapping function distributing this error in some optimal way.

Using a monitor function is convenient, since the mesh transform can be defined as the solution of a PDE and solved simultaneously, along with the original PDE. This results in a moving mesh system of equations, with one PDE controlling the movement of the mesh points and another PDE defining the physical process of interest. While this coupled system incurs more computational cost, since it requires that the mesh points be tracked accurately, it avoids the use of interpolation on dynamically changing grids in the physical domain. Updating the mesh via such interpolation schemes may cause numerical instabilities and inaccuracies, especially when the mesh has large deformations.

Even though much work has been done in studying the individual components of the moving mesh system, few papers consider the computational properties of the entire system. In [3], the distribution of mesh points is viewed as a constraint on the physical PDE and the stability of the resulting system of DAEs is discussed. For physical problems with steep gradients or large function values, [4] considers the efficiency of the overall system and shows that for a moving mesh system using the arc length (AL) monitor, there is a gain in efficiency when compared with solving the original physical PDE on a fixed mesh. Since more mesh points are moved toward the region where the solution changes most, this moving mesh system gives a more accurate solution to the physical PDE, especially when compared to a fixed mesh solution. This shows that by the proper design of a moving mesh system, one can gain both accuracy and efficiency. The current paper extends the work done in [4] by establishing a similar efficiency result for a general class of monitor functions. Also, by considering a discretized version of the moving mesh system, we establish estimates that describe the accuracy of the computed solution in terms of the mesh spacing in the physical domain. In certain circumstances, these results allow the user to control the level of accuracy in the physical solution or the level of efficiency in the system by selecting an appropriate monitor function.

The choice of the monitor function plays a critical role in the grid distribution and is not always a simple decision [5]. The characteristics of the solution to the physical PDE, like large gradients or even singularities, usually determine which monitor functions are suitable. However, numerical complexities in the physical solution can result in similar complexities in the mesh equations, because the mesh information appears in the coefficients of the transformed physical PDE. For example, a physical solution with a large gradient may force the mesh to move more points to this region. But this uneven distribution can result in large gradients in the equation for the mesh transform. So while the moving mesh solution gains accuracy in regions where the physical solution exhibits great change, the coupled system of physical PDE plus mesh PDE may have added computational cost.

In this paper, we examine a set of monitor functions that resemble the AL monitor ([1, 2]) and a monitor that optimizes interpolation errors (OI) [1]. While these monitor functions share a similar form and seem to affect the qualitative behavior of the grid

https://doi.org/10.4208/nmtma.2016.m1508 Published online by Cambridge University Press

points in a similar manner, the moving mesh system inherits different computational properties. Motivated by recent work using moving mesh methods to resolve steep phase transitions in materials science and biology ([6-9]), we work under the assumption that the physical solution has very large gradients in small regions of the domain. By assuming a specific form for the physical solution and using a small parameter, ϵ, to characterize these steep gradients, we show that the mesh transformation distributes this computational difficulty between the computed solution and the computed mesh transform. This trade off is illustrated by looking at the AL monitor and the OI monitor, with each choice resulting in a system with different computational burden.

Monitor functions used in moving mesh applications, like the AL monitor, typically include a user specified parameter. This parameter is usually referred to as an intensity parameter, and it controls the spacing of the mesh points as they are redistributed. We show that the norms of the physical solution, under the mesh transformation, depend directly on this parameter. By allowing this parameter to depend on ϵ, we can distribute the computational cost of the moving mesh system between the mesh equation and the physical PDE. For certain parameter choices, the overall system is shown to be more efficient than the original physical PDE. Additionally, we show that the mesh spacing in the physical domain depends directly on this parameter. By using this mesh spacing estimate, one may choose the value of the intensity parameter to directly affect the level of accuracy, instead of choosing the parameter in an ad hoc fashion, which is rather common in the literature.

In many applications, it is beneficial to understand how the choice of the monitor function affects the spacing between grid points once the grid has been redistributed. For example, when coupling a finite difference method with a moving mesh, the local truncation error is expressed directly in terms of the grid spacing [3]. Under the assumptions described above, we use the discrete form of the equidistribution principle to derive estimates on the mesh spacing in regions where the solution has large variation. These estimates follow from looking at the discrete analog of the equations that describe how the computational difficulties are shared between the transformed physical equation and the mesh transform equation. They illustrate the trade off between more dense grid spacing in regions where the solutions change most with added numerical complexity in the overall system. We consider the AL and OI monitor functions as examples and verify the properties numerically.

2. Moving mesh systems

To begin, let $u(x,t)$ represent the solution to a PDE for $x \in \Omega_P$. In the work that follows we assume the physical domain is one dimensional and that u is smooth and bounded, though it may have large gradient values as we will describe later. We denote the computational domain by $\Omega_C = [0,1]$ and use ξ to describe the spatial variable in Ω_C. For simplicity, assume $\Omega_C = \Omega_P$. One goal in mesh redistribution is to balance properties like large gradients between the mesh mapping and the transformed physical solution defined by $v(\xi,t) = u(x(\xi),t)$.

https://doi.org/10.4208/nmtma.2016.m1508 Published online by Cambridge University Press

The mesh transform maps $\Omega_C = [0,1]$ into $\Omega_P = [0,1]$ and is denoted by $x(\xi)$, with inverse map given by $\xi(x)$. We assume the mesh mapping satisfies:

$$\int_0^1 x_\xi d\xi = \int_0^1 \xi_x dx = 1. \tag{2.1}$$

Let ξ_j represent a uniform partition of $[0,1]$ with $\xi_j = j/N$ for $j = 0, 1, ...N$. From [1], the mesh transform $x(\xi)$ is called equidistributing if:

$$\int_0^{x(\xi_j)} \omega(x)\, dx = \sigma \xi_j \tag{2.2}$$

where $\sigma = \int_0^1 \omega(x)\, dx$ and ω is the monitor function. Note that (2.2) implies that the integral of the monitor function on $[x(\xi_j), x(\xi_{j+1})]$ is the same for all $j = 0, 1, ...N-1$. Replacing ξ_j with the continuous variable ξ in (2.2), it is not difficult to show that when a mesh transform is equidistributing, it also satisfies the PDE:

$$\frac{\partial}{\partial \xi}(\omega x_\xi) = 0\,. \tag{2.3}$$

We note that (2.3) represents one way to use a PDE to equidistribute the mesh, with many other examples given in the literature ([10-12]).

It is usually convenient to update the mesh mapping by solving a time dependent PDE whose steady-state solution is given by (2.3). Using this method, the mesh is updated by solving:

$$\frac{\partial x}{\partial t} = \mu \frac{\partial}{\partial \xi}(\omega x_\xi)\,, \tag{2.4}$$

where μ is a parameter that controls the rate at which the mesh points evolve.

In this paper, we restrict our focus to a class of monitor functions of the form:

$$\omega = \left(1 + \alpha v_\xi^2\right)^{1/q}\,, \tag{2.5}$$

for $q \geq 1$ and intensity parameter, α, controlling the distribution of mesh points. This choice is based on two popular monitor functions that share this form. The first is the arc length monitor function (AL) used in [2, 13, 14]:

$$\omega = \sqrt{1 + \alpha v_\xi^2}\,.$$

The second monitor function comes from minimizing polynomial interpolation errors and is discussed in detail in [1]:

$$\omega = \left(1 + \alpha v_\xi^2\right)^{1/3}\,, \tag{2.6}$$

where the intensity parameter has the form:

$$\frac{1}{\alpha} = \left(\int_0^1 v_\xi^{2/3}\, dx\right)^3\,. \tag{2.7}$$

We will refer to this monitor as the optimal interpolation mesh monitor (OI).

https://doi.org/10.4208/nmtma.2016.m1508 Published online by Cambridge University Press

2.1. Balancing computational efficiency

In order to understand how the moving mesh system distributes the computational burden of resolving steep gradients, some estimates on the bounds of the transformed solution and on the mesh transform are helpful. Throughout the rest of the paper, we assume the physical solution depends on some small parameter ϵ so that:

$$\|u_x\| = O(\epsilon^{-1})\,. \tag{2.8}$$

In what follows, we define $\|\cdot\|$ as the sup norm on $[0,1]$ with $\|f\| = \sup_{x\in[0,1]} |f(x)|$. We use the standard big O notation: $u(x) = O(\epsilon^p)$ if and only if $M_1\epsilon^p \le u(x) \le M_2\epsilon^p$ holds for some constants M_1 and M_2. In other words, as ϵ tends to zero, the physical solution is characterized by steep gradients. Note that we use M, and M_i for $i = 1, 2$ to represent finite numbers which might be different in value.

In [4], it was shown that using (2.3) with the AL monitor function, the mesh transform function, ξ_x, can be written as the root of a polynomial whose coefficients depend on the physical solution. Under the assumption (2.8) and for solutions with bounded arc length independent of the parameter ϵ, the solution in computational domain, $v(\xi) = u(x(\xi))$, and the mapping from the physical domain to the computational domain, $\xi(x)$, satisfy the following bounds:

$$\|v_\xi\| = O(\epsilon^{-1/2}) \quad \text{and} \quad \|\xi_x\| = O(\epsilon^{-1/2})\,. \tag{2.9}$$

Note that the estimates in (2.9) are balanced, with the solution in the computational domain sharing the same order of magnitude with respect to ϵ as the mesh transform. This shows that by choosing the AL monitor function we can keep both v_ξ and ξ_x balanced, and more importantly of a lower order with respect to ϵ than the original solution. Here both equations may be solved with reasonable accuracy using fewer mesh points than necessary for the original PDE without a moving mesh.

We next present an estimate that is similar to (2.9) for the monitor (2.5) with $q \ge 1$, that includes the dependence on the intensity parameter, α.

Theorem 2.1. *Let $u(x)$ be the physical solution such that*

(i) $\|u_x\| = O(\epsilon^{-1})$ *in* Ω *with meas*$(\Omega) = O(\epsilon)$*, and*

(ii) $\|u_x\| = O(\epsilon^p)$ *in* $[0,1] \setminus \Omega$ *for some* $p > 0$*.*

Then using (2.3) *with the monitor function:*

$$\omega = \left(1 + \alpha v_\xi^2\right)^{1/q}\,, \tag{2.10}$$

for constant $\alpha > 0$ and $q \ge 1$, the solution in computational domain, $v(\xi) = u(x(\xi))$, and the mapping from the physical domain to the computational domain, $\xi(x)$, satisfy the following bounds for ϵ sufficiently small:

$$\|v_\xi\|^{q+2} = O\left(\alpha^{-1}\epsilon^{-q}\right) \quad \textit{and} \quad \|\xi_x\|^{q+2} = O\left(\alpha\epsilon^{-2}\right)\,. \tag{2.11}$$

https://doi.org/10.4208/nmtma.2016.m1508 Published online by Cambridge University Press

Before we present the proof, note that the choice of the exponent in the monitor function distributes the computational difficulty of resolving large gradients from the mesh equation to the transformed physical equation. For constant α, increasing q leads to larger gradients in the computed solution, v, and smaller gradients in the mesh transform function, ξ.

Proof. Following [4], we integrate (2.3) to get:

$$\omega x_\xi = \sigma\,,$$

for some constant σ. Replacing ω with the monitor (2.10) and substituting $v_\xi = u_x x_\xi$ gives, for any $\alpha > 0$,

$$\left(1 + \alpha u_x^2 x_\xi^2\right)^{1/q} x_\xi = \sigma\,, \tag{2.12}$$

which implies $\sigma \geq x_\xi$. By integrating this last inequality and using our assumption on the mesh transform,

$$\sigma \geq \int_0^1 x_\xi d\xi = 1\,.$$

Next, we multiply (2.12) by $\xi_x^{(q+2)/q}$ to get:

$$\sigma \xi_x^{(q+2)/q} = (\xi_x^2 + \alpha u_x^2)^{1/q}\,,$$

and by taking both sides to the power q:

$$\sigma^q \xi_x^{q+2} - \xi_x^2 - \alpha u_x^2 = 0\,. \tag{2.13}$$

Since ξ is monotonic, ξ_x is a positive, real root of the function:

$$f_q(z) = \sigma^q z^{q+2} - z^2 - \alpha u_x^2\,. \tag{2.14}$$

When $\alpha > 0$, it is not hard to see that f_q has exactly one root on $[0, \infty)$.

In Ω, for ϵ sufficiently small, the bounds on u_x and the intermediate value theorem imply f_q has a root in

$$\left[\left(M_1^2 \sigma^{-q} \alpha \epsilon^{-2}\right)^{1/(q+2)}, \left(2M_2^2 \sigma^{-q} \alpha \epsilon^{-2}\right)^{1/(q+2)}\right]\,,$$

where M_1 and M_2 are the constants defined by (2.8):

$$M_1 \epsilon^{-1} \leq \|u_x\| \leq M_2 \epsilon^{-1}\,, \quad \text{in} \quad \Omega\,.$$

This implies:

$$\|\xi_x\|^{q+2} = O(\sigma^{-q} \alpha \epsilon^{-2})\,, \quad \text{in} \quad \Omega\,. \tag{2.15}$$

In $[0,1] \setminus \Omega$, since

$$\alpha u_x^2 = \alpha \epsilon^{2p} \to 0 \quad \text{as} \quad \epsilon \to 0\,, \tag{2.16}$$

https://doi.org/10.4208/nmtma.2016.m1508 Published online by Cambridge University Press

the intermediate value theorem implies the root of f_q is in $[\sigma^{-1}, 2\sigma^{-1}]$ for ϵ sufficiently small. This gives:

$$\|\xi_x\| = O(\sigma^{-1}), \quad \text{in} \quad [0,1] \setminus \Omega. \tag{2.17}$$

Next we obtain bounds on v_ξ in the two regions. Multiplying (2.13) by x_ξ^2, rearranging terms, and applying chain rule gives:

$$v_\xi^2 = u_x^2 x_\xi^2 = \frac{\sigma^q \xi_x^q - 1}{\alpha}. \tag{2.18}$$

Using (2.15) and (2.18), we have:

$$\begin{aligned} v_\xi^2 &\leq M\sigma^q \alpha^{-1} \left(\sigma^{-q}\alpha\epsilon^{-2}\right)^{q/(q+2)} \\ &\leq M \left(\sigma^{2q}\alpha^{-2}\epsilon^{-2q}\right)^{\frac{1}{(q+2)}}, \quad \text{in} \quad \Omega \end{aligned}$$

for some constant M. The lower bound is similar, and we have:

$$\|v_\xi\|^{q+2} = O(\sigma^q \alpha^{-1}\epsilon^{-q}), \quad \text{in} \quad \Omega. \tag{2.19}$$

Now using (2.17) and (2.18), we have:

$$\|v_\xi\| = O(\alpha^{-1/2}), \quad \text{in} \quad [0,1] \setminus \Omega. \tag{2.20}$$

Finally, we need an upper bound on the constant σ to obtain the result. Multiplying (2.12) by ξ_x and using the assumption (2.1), we have:

$$\sigma = \int_0^1 \sigma \xi_x dx = \int_0^1 (1 + \alpha v_\xi^2)^{1/q} dx.$$

This allows us to use (2.19)-(2.20) and estimate for $q \geq 1$:

$$\begin{aligned} \sigma &\leq 1 + \alpha^{1/q} \left(\int_\Omega v_\xi^{2/q} dx + \int_{[0,1]\setminus\Omega} v_\xi^{2/q} dx \right) \\ &\leq 1 + \alpha^{1/q}\, M_1 (\sigma^2\, \alpha^{-2/q}\, \epsilon^{-2})^{\frac{1}{q+2}}\, \text{meas}(\Omega) + \alpha^{1/q} M_2 \alpha^{-1/q} \\ &\leq \hat{M}_1 + \hat{M}_2 (\sigma^2 \alpha \epsilon^q)^{\frac{1}{q+2}} \leq \sigma^{\frac{2}{q+2}} \tilde{M}. \end{aligned}$$

The last inequality follows from the fact that $(\alpha\epsilon^q)^{\frac{1}{q+2}}$ is bounded and that $\sigma \geq 1$. The upper bound on σ follows. □

Note that in most of the literature, the monitor function is defined in terms of the derivatives of u_x and not v_ξ, as defined above. The above proof can be adapted to this case and it is noted that under the hypotheses of Theorem 2.1, for monitors of the form:

$$\omega = \left(1 + \alpha u_x^2\right)^{1/q}, \quad q \geq 2$$

https://doi.org/10.4208/nmtma.2016.m1508 Published online by Cambridge University Press

the bounds:

$$\|v_\xi\|^q = O\left(\alpha^{-1}\epsilon^{-q+2}\right) \quad \text{and} \quad \|\xi_x\|^q = O\left(\alpha\epsilon^{-2}\right),$$

hold for $q \geq 2$, constant α, and ϵ sufficiently small. This result is different from (2.11) and in the case $q = 2$, the balanced estimate (2.9) is no longer attained.

Since the bounds on v_ξ and ξ_x both depend on the intensity parameter, allowing α to depend on ϵ may allow for some flexibility in distributing the efficiency between the physical equation and the mesh equation. In order to use the result from the previous Theorem, we need ensure that in $[0,1] \setminus \Omega$, αu_x^2 is small and that $\alpha\epsilon^q$ is bounded as ϵ tends to zero. When $\alpha = \epsilon^{-r}$, the results from the Theorem hold as long as $r < 2p$ and $r \leq q$. The following Corollary addresses this case.

Corollary 2.1. *Let $u(x)$ be a physical solution that satisfies the hypotheses of Theorem* 2.1. *Then using moving mesh PDE* (2.3) *with the monitor function* (2.10) *and $\alpha = O(\epsilon^{-r})$, the transformed physical solution and the mesh transform satisfy the following bounds when $r < 2p$ and $r \leq q$:*

$$\|v_\xi\|^{q+2} = O\left(\epsilon^{r-q}\right) \quad \text{and} \quad \|\xi_x\|^{q+2} = O\left(\epsilon^{-r-2}\right).$$

2.2. Numerical examples

In this section, we illustrate the results from Section 2.1 with several numerical examples where we choose a particular form for the physical solution that satisfies the required hypotheses given by the theorem. We first consider the OI monitor (2.6) with α defined by (2.7) and show that for this example, the burden of resolving steep gradients that was present in the original problem is still present in the moving mesh system. A second example shows that by altering the choice for the intensity parameter, we can make the AL monitor behave in a manner similar to the OI monitor. In Section 3, it will be shown that this decrease in efficiency for the system will increase the accuracy in the computed physical solution by concentrating more mesh points in the region Ω.

Corollary 2.2. *Let $u(x)$ be a physical solution that satisfies conditions* (i)-(ii) *in Theorem* 2.1 *with $p > 3/2$. If the physical solution is transformed using* (2.3) *with the OI monitor defined by* (2.6)-(2.7) *then:*

$$\|v_\xi\| = O(1), \quad \text{and} \quad \|\xi_x\| = O(\epsilon^{-1}) \tag{2.21}$$

Proof. Using the OI monitor, the upper bound on σ is conveniently obtained by:

$$\begin{aligned}\sigma &= \int_0^1 \sigma\xi_x dx = \int_0^1 (1 + \alpha v_\xi^2)^{1/3} dx \\ &\leq 1 + \alpha^{1/3}\left(\int_0^1 v_\xi^{2/3} dx\right) \leq 2,\end{aligned}$$

where the last inequality follows directly from the definition of α in (2.7).

https://doi.org/10.4208/nmtma.2016.m1508 Published online by Cambridge University Press

In this particular case, the parameter α may depend on ϵ. For now, assume (2.16) holds so that the bound (2.17) can be written as:

$$\frac{1}{2} \leq \xi_x \leq \frac{a}{2}\,,$$

for $a > 1$. This allows us to estimate:

$$v_\xi^2 = \frac{\sigma^3 \xi_x^3 - 1}{\alpha} \leq 2^{-3}\alpha^{-1} \quad \text{in} \quad [0,1] \setminus \Omega\,,$$

where we have chosen a so that $a^3 - 1 = 2^{-3}$.

Using the results from Theorem 2.1 with $q = 3$, we estimate α by:

$$\begin{aligned}
\alpha^{-1/3} &= \int_\Omega v_\xi^{2/3} dx + \int_{[0,1]\setminus\Omega} v_\xi^{2/3} dx \\
&\leq \left(M\alpha^{-1/5}\epsilon^{-3/5}\right)^{2/3} \cdot \text{meas}(\Omega) + (2^{-3}\alpha^{-1})^{1/3} \\
&\leq M\alpha^{-2/15}\epsilon^{3/5} + \frac{1}{2}\alpha^{-1/3}\,.
\end{aligned}$$

This last equation implies:

$$\alpha^{-1} \leq M\epsilon^3\,,$$

and a similar argument shows that $\alpha^{-1} = O(\epsilon^3)$. We note that if the physical solution satisfies $\|u_x\| = O(\epsilon^p)$ in $[0,1] \setminus \Omega$ for $p > 3/2$, then our assumption (2.16) holds. Finally, the theorem results from substituting this estimate on α into (2.11) with $q = 3$. □

To illustrate Corollary 2.2 numerically, we assume that the solution of the physical partial differential equation is:

$$u(x) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{x - \frac{1}{2}}{4\epsilon}\right)\,, \tag{2.22}$$

which satisfies the hypotheses of Theorem 2.1. The profile for this function together with a redistributed mesh is illustrated in Fig. 1.

By applying the moving mesh PDE (2.4) with the OI monitor function for various values of ϵ, we obtain the results in Table 1 that verify the magnitude of the gradients of the transformed physical solution, v, and the mesh transform, ξ, given by Corollary 2.2. The column labeled Ratio gives the norm for the given derivative in the simulation with parameter value ϵ divided by the norm for parameter value $\epsilon/2$. Here we have discretized the moving mesh PDE by standard finite difference methods and we have estimated the parameter α by a simple Riemann sum. The computation used $N = 101$ mesh points with $\Delta t = 10^{-4}$ and the simulation ended when the difference in the solutions $x(\xi, t)$ for two consecutive time steps was measured as smaller than 10^{-10}.

https://doi.org/10.4208/nmtma.2016.m1508 Published online by Cambridge University Press

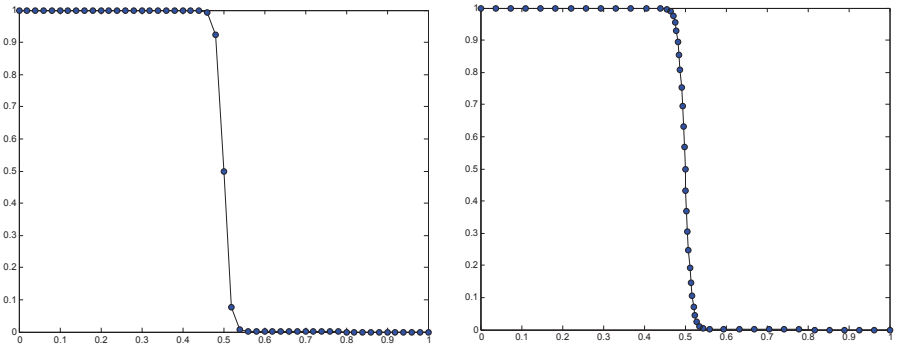


Figure 1: u in uniform mesh (left) and after mesh redistribution with OI monitor (right).

Table 1: Sup norms for the solution (2.22) using moving mesh and OI monitor.

ϵ	$\|v_\xi\|$	Ratio	$\|\xi_x\|$	Ratio	$\|u_x\|$	Ratio
0.008	3.102344	0.928618	5.030045	0.538545	15.604928	0.500103
0.004	3.340817	0.956375	9.340062	0.522881	31.203440	0.500070
0.002	3.493210	0.971506	17.862705	0.514715	62.398180	0.500049
0.001	3.595665	0.984354	34.704069	0.507976	124.784222	0.500028
0.0005	3.652818		68.318372		249.554596	

A similar result for the AL monitor can be found in [4], verifying (2.9) for constant α and showing that both $\|v_\xi\|$ and $\|\xi_x\|$ are of $O(\epsilon^{-1/2})$. Note that in this case, the magnitudes of the steep gradients present in both PDEs have been reduced and are in fact lower than the original PDE. While this does improve efficiency in the overall system, we will later see that it comes with the price of decreased accuracy when comparing with the OI monitor.

As a second example, we demonstrate that the transformed mesh solution with the AL monitor behaves just like the OI monitor from the last example by using Corollary 2.1 and letting $\alpha = \epsilon^{-2}/10$. Here, we have used the same parameters as the first simulation with the exception of a smaller $\Delta t = 10^{-6}$ for stability. The results are given in Table 2.

Table 2: Sup norms for the solution (2.22) using moving mesh and AL monitor with $\alpha = \epsilon^{-2}/10$.

ϵ	$\|v_\xi\|$	Ratio	$\|\xi_x\|$	Ratio	$\|u_x\|$	Ratio
0.008	1.719566	0.979855	9.083009	0.510288	15.618838	0.500008
0.004	1.754920	0.990204	17.799769	0.504950	31.237164	0.500004
0.002	1.772280	0.994337	35.250526	0.502850	62.473816	0.500002
0.001	1.782374	0.995264	70.101481	0.502381	124.947034	0.500002
0.0005	1.790856		139.538361		249.893058	

https://doi.org/10.4208/nmtma.2016.m1508 Published online by Cambridge University Press

3. Mesh spacing estimates

In order to derive estimates on the mesh spacing after the mesh has been redistributed, we first set up the discrete version of the moving mesh problem. Let ξ_j represent a discrete partition of the computational domain $[0,1]$ with $\xi_j = j\Delta\xi$ for $j = 0, 1, ...N$ and $\Delta\xi = 1/N$. Since x represents the mapping into the physical domain, at time t the location of the jth grid point is given by $x_j(t) = x(\xi_j, t)$. The spacing between grid points at time t is now defined by:

$$h_j(t) = x_j(t) - x_{j-1}(t)\,, \quad \text{for} \quad j = 1, ...N\,.$$

To update the mesh, we employ a discrete version of the equidistribution principle. It is not difficult to show that (2.2) is equivalent to:

$$h_j\omega_j = \frac{1}{N}\sigma_h \quad \text{for all} \quad j = 1, ...N \tag{3.1}$$

where ω_j is a discrete approximation of the monitor function at grid point x_j and $\sigma_h = \sum_{j=1}^{N}\omega_j h_j$. If we think of $\omega_j h_j$ as being some measure of error on the interval of length h_j, then (3.1) implies that the grid points are distributed so that the error on each interval is the same.

Partitioning the time interval $[0,T]$ into equal subintervals, we set $t^n = n\Delta t$ for $n = 0, 1, ...M$ with $\Delta t = T/M$. We denote the location of the jth grid point at time t^n by $x_j^n = x(\xi_j, t^n)$ and define the mesh spacing, h_j^n, in a similar manner. The solution to the physical PDE, $u(x,t)$, at time t^n and grid point x_j is denoted by $u_j^n = u(x_j, t^n)$. Starting with a uniform mesh in the physical domain, $x_j^0 = \xi_j$, the mesh is updated by solving a discrete approximation to (2.4). In the work that follows, (2.4) is solved using a standard, centered, finite difference approximation in space with Dirichlet boundary conditions, and explicit time stepping.

To approximate the monitor function at grid point x_j, it is convenient to define the backwards difference operator:

$$D_\xi v_j = \frac{v_j - v_{j-1}}{\Delta\xi}\,.$$

Here, $v_j = u(x(\xi_j))$. Then the discrete approximation to the monitor function (2.5) is given by:

$$\omega_j = \left(1 + \alpha(D_\xi v_j)^2\right)^{1/q}\,. \tag{3.2}$$

The AL monitor function results from substituting $q = 2$ in (3.2), and substituting $q = 3$ gives the OI monitor with:

$$\frac{1}{\alpha} = \left(\sum_{j=1}^{N}(D_\xi v_j)^{2/3}h_j\right)^3\,. \tag{3.3}$$

The mesh spacing estimate for monitor (3.2) with $q \geq 1$ is given below.

https://doi.org/10.4208/nmtma.2016.m1508 Published online by Cambridge University Press

Theorem 3.1. *Let $u(x)$ be the physical solution such that*

(i) $\|u_x\| = O(\epsilon^{-1})$ *in* Ω *with meas*$(\Omega) = O(\epsilon)$ *and*

(ii) $\|u_x\| = O(\epsilon^p)$ *in* $[0,1] \setminus \Omega$ *for* $p > 0$.

Then, when the mesh has been equidistributed with (2.3) *using the monitor function* (3.2) *with* $\alpha > 0$*, the following bounds on the mesh spacing hold for* $q \geq 1$*:*

$$\left(\frac{\Delta\xi}{h_j}\right)^{q+2} = O(\alpha\epsilon^{-2}) \quad \text{in } \Omega \qquad \text{and} \qquad \left(\frac{\Delta\xi}{h_j}\right) = O(1) \quad \text{in } [0,1] \setminus \Omega\,.$$

Proof. The proof is essentially a discrete version of the proof of Theorem 2.1, with $\frac{\Delta\xi}{h_j}$ replacing ξ_x and we include it only for completeness. We first establish a lower bound on $\sigma_h = \sum_{j=1}^{N} \omega_j h_j$. It follows from the fact that $\omega_j \geq 1$:

$$\sigma_h = \sum_{1}^{N} \omega_j h_j \geq \sum_{1}^{N} h_j = 1\,.$$

Next, we establish estimates analogous to (2.15)-(2.17). By manipulating (3.1) with ω_j replaced by (3.2), one observes that $\frac{\Delta\xi}{h_j}$ is a root of $f_q(z)$ given in (2.14) with σ replaced by its discrete counterpart, σ_h. This leads to

$$\left(\frac{\Delta\xi}{h_j}\right)^{q+2} = O(\sigma_h^{-q}\alpha\epsilon^{-2}) \quad \text{in} \quad \Omega\,,$$

and

$$\left(\frac{\Delta\xi}{h_j}\right) = O(\sigma_h^{-1}) \qquad \text{in} \quad [0,1] \setminus \Omega\,,$$

The result should follow from an upper bound on σ_h.

To bound σ_h from above, let I_j represent the interval (x_{j-1}, x_j). We use the bounds on $|v_\xi|$ obtained in Theorem 2.1 and the size of Ω to estimate:

$$\begin{aligned}
\sigma_h &= \sum_{1}^{N} \left(1 + \alpha(D_\xi v_j)^2\right)^{1/q} h_j \\
&\leq 1 + \alpha^{1/q} \left(\sum_{I_j \cap \Omega \neq \emptyset} (D_\xi v_j)^{2/q} h_j + \sum_{I_j \cap \Omega = \emptyset} (D_\xi v_j)^{2/q} h_j \right) \\
&\leq 1 + \alpha^{1/q} M_1 (\sigma_h^2 \alpha^{-2/q} \epsilon^{-2})^{\frac{1}{q+2}} \sum_{I_j \cap \Omega \neq \emptyset} h_j + \alpha^{1/q} M_2 \alpha^{-1/q} \\
&\leq \hat{M}_1 + \hat{M}_2 \left(\sigma_h^2 \alpha \epsilon^q\right)^{\frac{1}{q+2}} \leq M \sigma_h^{\frac{2}{q+2}}\,. \qquad (3.4)
\end{aligned}$$

Note that $\sum_{I_j \cap \Omega \neq \emptyset} h_j$ may be slightly bigger than meas(Ω), but can certainly be bounded by $K\epsilon$, since we expect the mesh spacing to be small for intervals I_j near Ω. Also, the

https://doi.org/10.4208/nmtma.2016.m1508 Published online by Cambridge University Press

last inequality in (3.4) follows from the fact that $\alpha\epsilon^q$ is bounded and $\sigma_h \geq 1$. With the bound on σ_h established, the result follows. □

The choice of q in the monitor function directly affects the mesh spacing in the steep gradient region, Ω. This implies that q can be used to control the level of accuracy in the physical solution.

Corollary 3.1. *Let $u(x)$ be a physical solution that satisfies conditions* (i)-(ii) *of Theorem* 3.1 *with $p > 3/2$. Then when the mesh has been redistributed using* (2.3) *with the OI monitor function* (3.2) *with $q = 3$ and α given by* (3.3)*, the following bounds hold:*

$$\frac{\Delta\xi}{h_j} = O(\epsilon^{-1}) \ \text{ in } \ \Omega \quad \text{ and } \quad \frac{\Delta\xi}{h_j} = O(1) \ \text{ in } \ [0,1] \setminus \Omega\,.$$

Corollary 3.1 is the discrete counterpart of Corollary 2.2. Since the proof is similar to the proof of Theorem 3.1, it is omitted.

To compare mesh spacing, we note that for the AL monitor function, the mesh spacing in Ω is given by $\Delta\xi/h_j = O(\epsilon^{-1/2})$. For small ϵ, the mesh spacing in the steep gradient region is more dense for the OI monitor than for the AL monitor. This is not surprising since the OI monitor specifically minimizes error when interpolating by piecewise constant functions. However, this gain in accuracy comes at the expense of a less efficient system to solve, as shown by Theorem 2.1.

3.1. Numerical results for mesh spacing

Next, we verify the mesh spacing estimates presented in the last section. If the physical solution has the form (2.22), then for $\epsilon = 0.002$, Fig. 2 shows plots of $N = 101$ mesh points after equidistribution. It is clear from the plots that the monitor OI moves more mesh points to the region where the gradient is steepest. Here, we have chosen $\Delta t = 10^{-4}$ and ended the simulation when the difference in the solutions $x(\xi, t)$ for two consecutive time steps was measured as smaller than 10^{-10}. The intensity parameter for the AL monitor was set at $\alpha = 1$.

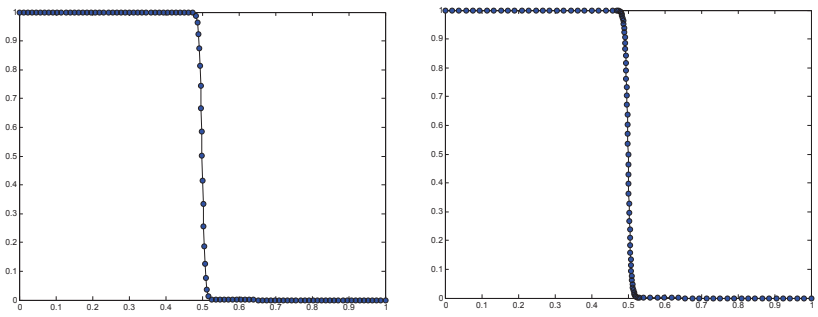


Figure 2: $N = 101$ mesh points after redistribution with AL monitor (left) and OI monitor (right) for $\epsilon = 0.002$.

https://doi.org/10.4208/nmtma.2016.m1508 Published online by Cambridge University Press

Table 3: Mesh spacing estimates at equidistribution when u has large gradient using the AL monitor and constant α.

	AL monitor, α constant			
ϵ	$\max h_j$	Ratio	$\min h_j$	Ratio
0.008	1.265423e-02	1.048	2.813595e-03	1.437
0.004	1.207829e-02	1.042	1.957793e-03	1.436
0.002	1.158592e-02	1.036	1.363325e-03	1.430
0.001	1.118764e-02	1.026	9.531910e-04	1.419
0.0005	1.089916e-02		6.718506e-04	

Table 4: Mesh spacing estimates at equidistribution when u has large gradient using the OI monitor.

	OI monitor			
ϵ	$\max h_j$	Ratio	$\min h_j$	Ratio
0.008	1.751353e-02	0.948	1.988054e-03	1.857
0.004	1.847074e-0	0.968	1.070657e-03	1.913
0.002	1.907786e-02	0.984	5.598256e-04	1.943
0.001	1.938621e-02	0.990	2.881507e-04	1.969
0.0005	1.958209e-02		1.463735e-04	

To verify the order of the grid spacing as predicted by Theorem 3.1 and Corollary 3.1 for the AL monitor and OI monitor respectively, we compute the maximal mesh spacing, $\max h_j$, and the minimal mesh spacing $\min h_j$ at steady state. The parameters were chosen in the same way as Fig. 2, with ϵ varying as shown in Tables 3-4 and N fixed. Note that the tables verify the correct order on the mesh spacing, since for the AL monitor, we expect fixed $\Delta\xi$ will give $\max h_j = O(1)$ and $\min h_j = O(\sqrt{\epsilon})$. Similarly, for the OI monitor, fixed $\Delta\xi$ implies $\max h_j = O(1)$ and $\min h_j = O(\epsilon)$.

In the next example, we show that the AL monitor can attain the same density of mesh spacing as the OI monitor if α is chosen appropriately. Using $N = 101$ points with

Table 5: Mesh spacing estimates at equidistribution when u has large gradient using the AL monitor with $\alpha = \epsilon^{-2}/10$.

	AL monitor, $\alpha = O(\epsilon^{-2})$			
ϵ	$\max h_j$	Ratio	$\min h_j$	Ratio
0.008	7.484215e-02	0.960	1.100957e-03	1.960
0.004	7.794605e-02	0.981	5.618050e-04	1.980
0.002	7.949496e-02	0.989	2.836837e-04	1.989
0.001	8.040300e-02	0.991	1.426503e-04	1.991
0.0005	8.117029e-02		7.166488e-05	

https://doi.org/10.4208/nmtma.2016.m1508 Published online by Cambridge University Press

$\Delta t = 10^{-6}$ and $\alpha = \epsilon^{-2}/10$ gives mesh spacing on the same order as the OI monitor with respect to ϵ. The results are given in Table 5. Note that the overall system is now less efficient, since we still need to resolve gradients of $O(\epsilon^{-1})$, but we have gained accuracy in the steep gradient region.

4. A Moving mesh system with burger's equation

The numerical examples presented in this paper so far have used a fixed function to represent the solution of the physical PDE. In this section, solutions of a coupled moving mesh system are computed using Burger's equation for the physical PDE:

$$u_t + \left(\frac{1}{2}u^2\right)_x = \nu u_{xx}\,, \quad 0 < x < 1\,, \quad t > 0\,, \tag{4.1}$$

with initial conditions:

$$u(x,0) = \begin{cases} 1 & 0 \le x < 0.25 \\ 2 - 4x & 0.25 \le x < 0.5 \\ 0 & 0.5 \le x \le 1\,, \end{cases} \tag{4.2}$$

and boundary conditions:

$$u(0,t) = 1\,, \qquad u(1,t) = 0\,. \tag{4.3}$$

Here $u(x,t)$ represents velocity and ν represents the viscosity parameter. For small ν, the physical solution develops a steep wave front that moves toward the right, which is difficult to resolve on a uniform grid.

Rewriting (4.1) in terms of the computational coordinate ξ and using (2.4) to control the mesh points, the corresponding moving mesh system is given by:

$$\begin{aligned} &u_t - \frac{u_\xi}{x_\xi}x_t + \frac{1}{x_\xi}\left(\frac{1}{2}u^2\right)_\xi = \frac{\nu}{x_\xi}\left(\frac{u_\xi}{x_\xi}\right)_\xi\,, \\ &x_t = \mu\left(\omega x_\xi\right)_\xi\,, \end{aligned}$$

for some monitor function ω, physical parameter ν, and moving mesh parameter μ. The initial and boundary conditions for u are given by (4.2)-(4.3). The boundary conditions for the mesh function are given by $x(0,t) = 0$ and $x(1,t) = 1$.

As in previous section, the domain $[0,1]$ is partitioned into N equal subintervals of length $\Delta\xi = 1/N$, the computational coordinate is discretized by letting $\xi_j = j\Delta\xi$ for $j = 0, 1, ...N$, and the location of the jth grid point at time t is $x(\xi_j, t) = x_j(t)$. The solution to Burger's equation at grid point x_j and time t is denoted by $u_j(t)$. By using standard, centered, finite difference approximations for the spatial derivatives,

https://doi.org/10.4208/nmtma.2016.m1508 Published online by Cambridge University Press

the semi-discrete form of the system at interior nodes labeled by $j = 1, ...N-1$ is:

$$\dot{u}_j = \dot{x}_j \left(\frac{u_{j+1} - u_{j-1}}{x_{j+1} - x_{j-1}} \right) - \frac{1}{2} \left(\frac{u_{j+1}^2 - u_{j-1}^2}{x_{j+1} - x_{j-1}} \right) + \frac{2\nu}{x_{j+1} - x_{j-1}} \left(\frac{u_{j+1} - u_j}{x_{j+1} - x_j} - \frac{u_j - u_{j-1}}{x_j - x_{j-1}} \right), \tag{4.4}$$

$$\dot{x}_j = \frac{\mu}{\Delta\xi^2} \Big(\omega_{j+1/2}(x_{j+1} - x_j) - \omega_{j-1/2}(x_j - x_{j-1}) \Big), \tag{4.5}$$

where the dot represents differentiation in time, $\omega_{j+1/2} = (\omega_{j+1} + \omega_j)/2$, and $\omega_{j-1/2}$ is defined similarly. The Dirichlet boundary conditions for both u and x are replaced by the ODEs:

$$\dot{u}_0 = 0\,, \quad \dot{u}_N = 0\,, \quad \dot{x}_0 = 0\,, \quad \dot{x}_N = 0\,. \tag{4.6}$$

Now, following [1], the system of ODEs given by (4.4)-(4.6) with initial conditions $u_j(0) = u(\xi_j, 0)$ as defined in (4.2) and $x_j(0) = \xi_j$ may be integrated using Matlab's implicit ODE solver of orders 1-5, ode15i. Note that in the computations that follow, the monitor function is smoothed using the weighted averaging technique described in [1] in order to prevent abrupt changes in the mesh trajectories.

Using 101 mesh points, the system is integrated with viscosity $\nu = 10^{-3}$ and moving mesh parameter $\mu = 1$ in order to compare the solutions for three separate cases: (1) using the AL monitor with constant $\alpha = 2$, (2) using the OI monitor with α defined by the integral in (2.7), and (3) using an AL monitor with $\alpha(t)$ updated based on the value of $\|u_x\|$ at time t. Relating the parameter ϵ from the previous sections to $\|u_x\|^{-1}$, we set $\alpha(t) = 0.5\|u_x\|^2$ in this last case. The solutions to the equations for all three cases are shown in Fig. 3 and some trajectories for the mesh points are shown in Fig. 4. In each of these figures, it is clear that the OI monitor moves more mesh points to the region where the gradient is steepest when compared to the AL monitor with constant α. However, the pictures for $\alpha(t) = 0.5\|u_x\|^2$ also appear to move a large number of mesh points toward the steep gradient region.

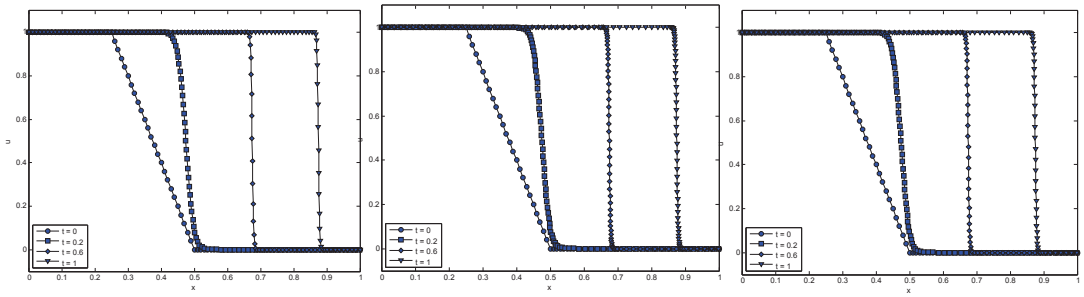


Figure 3: Solution to Burger's equation: AL monitor with constant α (left), OI monitor (center), AL monitor with $\alpha(t) = 0.5\|u_x\|^2$ (right).

As further evidence, Tables 6-8 show derivatives of the solutions along with the minimal mesh spacing at times indicated. In Table 6, as the derivative u_x becomes

https://doi.org/10.4208/nmtma.2016.m1508 Published online by Cambridge University Press

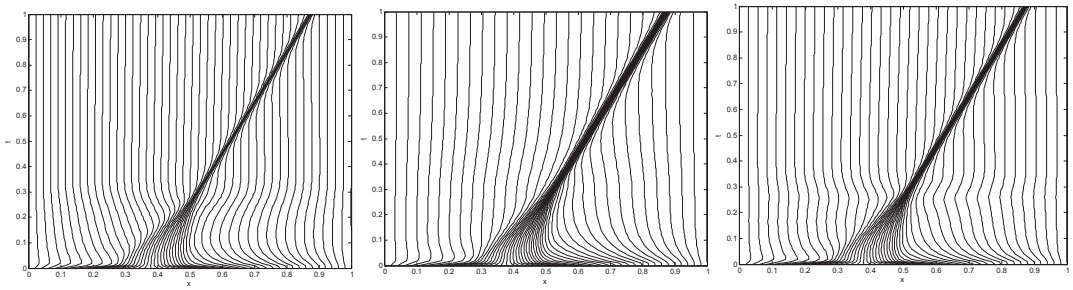


Figure 4: Mesh point trajectories: AL monitor with constant α (left), OI monitor (center), AL monitor with $\alpha(t) = 0.5\|u_x\|^2$ (right).

more steep, the AL monitor balances this steepness between the derivatives v_ξ and ξ_x, and we note that norms of both v_ξ and ξ_x are of lower order than u_x for this example. This is consistent with (2.9) and Theorem 2.1. With the OI monitor, in Table 7, one observes that the norms of v_ξ remain relatively constant, especially as u attains larger gradients in time. As a result, as the norm of u_x increases, the norm of ξ_x increases as expected from Theorem 2.1. Note that the minimal mesh spacing in Table 7 for $t \geq 0.6$ is less than half the values in Table 6. This is consistent with Theorem 3.1 and Corollary 3.1. Finally, note that the values in Table 8, for the modified AL monitor with $\alpha(t)$ behave more like the OI monitor, especially with regard to minimal mesh spacing.

Table 6: Derivatives of solution to Burgers equation at time t with AL monitor and constant $\alpha = 2$.

t	$\|v_\xi\|$	$\|\xi_x\|$	$\|u_x\|$	$\min h_j$
0.0	4.000000e+00	1.000000e+00	4.000000e+00	1.250000e-02
0.2	4.269413e+00	4.655121e+00	1.983713e+01	2.684054e-03
0.6	1.077445e+01	1.135972e+01	1.223947e+02	1.090397e-03
1.0	1.086150e+01	1.127936e+01	1.225108e+02	1.100204e-03

Table 7: Derivatives of solution to Burgers equation at time t with OI monitor.

t	$\|v_\xi\|$	$\|\xi_x\|$	$\|u_x\|$	$\min h_j$
0.0	4.000000e+00	1.000000e+00	4.000000e+00	1.250000e-02
0.2	3.135205e+00	6.373990e+00	1.994537e+01	1.960964e-03
0.6	3.924230e+00	3.175191e+01	1.244557e+02	3.931519e-04
1.0	3.934588e+00	3.166070e+01	1.245718e+02	3.947666e-04

5. Conclusion

In adaptive methods where the grid points are controlled by a moving mesh PDE with monitor function, it is not surprising that the choice of the monitor plays a critical

https://doi.org/10.4208/nmtma.2016.m1508 Published online by Cambridge University Press

Table 8: Derivatives of solution to Burgers equation at time t with AL monitor and $\alpha = 0.5\|u_x\|^2$.

t	$\|v_\xi\|$	$\|\xi_x\|$	$\|u_x\|$	$\min h_j$
0.0	4.000000e+00	1.000000e+00	4.000000e+00	1.000000e-02
0.2	3.215127e+00	6.189297e+00	1.987212e+01	1.615677e-03
0.6	4.917278e+00	2.533702e+01	1.244911e+02	3.938788e-04
1.0	4.959537e+00	2.512274e+01	1.244699e+02	3.975160e-04

role in the accuracy of the solution to the physical PDE. However, when we consider this advantage together with the computational efficiency of the overall moving mesh system, there is an evident trade off for this gain in accuracy. In problems with boundary layers, if we measure the efficiency of the system by the steepness of gradients that must be resolved, then the choice of monitor and intensity parameter both play a role in distributing the burden of these large gradients between the PDE for the transformed physical solution and the PDE that determine the location of the mesh points. This increase in computational efficiency has a direct impact on the spacing of mesh points in the physical domain, with more efficient systems having less densely spaced mesh points in regions of interest. While the AL monitor with constant intensity parameter leads to a more efficient system, the OI monitor gives more accurate solutions to the physical problem. Moreover, by an appropriate choice for the monitor and intensity parameter, we can specify the overall efficiency or the mesh spacing, as needed by the physical problem at hand.

Because the mesh transform adds a new level of complexity to solving the physical PDE, the analysis presented has only been for the one-dimensional case. While we expect these results to also hold in higher dimensions, this is certainly worthy of further investigation. The ideas presented in this paper may also extend naturally to systems of PDEs in one space dimension, for example in reaction-diffusion models that allow multiple substances to interact. In addition, we have only considered one form of monitor functions used in practice and further study of other forms is warranted. Most importantly, the results here illustrate the need for investigating properties of the whole moving mesh system, instead of focusing solely on one component of the system.

Acknowledgments The author would like to sincerely thank Qiang Du and both referees for their advice and helpful suggestions.

References

[1] W.-Z. Huang and R. D. Russell, *Adaptive Moving Mesh Methods*, Springer, Applied Mathematical Sciences, Vol. 174, 2011.

[2] C. J. Budd, W.-Z. Huang and R. D. Russell, *Adaptivity with moving grids*, Acta Numerica., **2** (2009), pp. 1–31.

[3] S.-T. Li, L. Petzold, and Y.-H. Ren, *Stability of moving mesh systems of partial differential equations*, SIAM J. Sci. Comp., Vol. 20, No. 2, (1998), pp. 719–738.

https://doi.org/10.4208/nmtma.2016.m1508 Published online by Cambridge University Press

[4] J. Remski, J. Zhang, and Q. Du, *On balanced moving mesh methods*, Journal of Computational and Applied Mathematics, vol. 265, (2014), pp. 255–263.
[5] W. Cao, W. Huang and R. D. Russell, A study of monitor functions for two dimensional adaptive mesh generation, SIAM Journal of Scientific Computing, 20 (1999), pp. 1978–1994.
[6] J. Mackenzie, M. L. Robertson, *A moving mesh method for the solution of the one-dimensional phase-field equations*, J. Comput. Phys., 181 (2002), pp. 526–544.
[7] P. Yu, L.-Q. Chen and Q. Du, *Applications of moving mesh methods to the Fourier Spectral Approximations of Phase-Field Equations*, in Recent Advances in Computational Sciences: Selected Papers from the International Workshop on Computational Sciences and its Education, (2008) pp. 80–99, edited by Jorgensen *et al.*, World Scientific.
[8] W.-M. Feng, P. Yu, S. Y. Hu, Zi-Kui Liu, Q. Du and L.-Q. Chen, *Spectral implementation of an adaptive moving mesh method for phase-field equations*, Journal of Computational Physics, 220 (2006), pp. 498–510.
[9] W.-M. Feng, P. Yu, S. Y. Hu, Zi-Kui Liu, Q. Du and L.-Q. Chen, *A Fourier Spectral Moving Mesh Method for the Cahn-Hilliard Equation with Elasticity*, Communications in Computational Phys., 5 (2009), pp. 582–599.
[10] W. Huang, Y. Ren and R. D. Russell, Moving mesh methods based on moving mesh partial differential equations, *J. Comput. Phys.*, 113 (1994), pp. 279–290.
[11] W. Huang. Practical Aspects of Formulation and Solution of Moving Mesh Partial Differential Equations, *Journal of Computational Physics*, 171 (2001), pp. 753–775.
[12] W. HUANG, Y. REN, AND R. D. RUSSELL; *Moving mesh partial differential equations (MMPDES) based on the equidistribution principle*, SIAM J. Numer. Anal. 31 (1994), no. 3, 709–30.
[13] Weiqiang Ren and Xiaoping Wang, *An Iterative Grid Redistribution Method for Singular Problems in Multiple Dimensions*, Journal of Computational Physics, 159 (2000), pp. 246–273.
[14] Hector D. Ceniceros and Thomas Y. Hou *An Efficient Dynamically Adaptive Mesh for Potentially Singular Solutions*, Journal of Computational Physics, 172 (2001) pp. 609–639.

https://doi.org/10.4208/nmtma.2016.m1508 Published online by Cambridge University Press