

# The Error-Minimization-based Rezone Strategy for Arbitrary Lagrangian-Eulerian Methods

Konstantin Lipnikov, Mikhail Shashkov

Theoretical Division, Los Alamos National Laboratory, MS B284, Los Alamos, New Mexico 87545

Received 22 December 2004; accepted 12 June 2005

Published online in Wiley InterScience (www.interscience.wiley.com).

DOI 10.1002/num.20113\*

The objective of the Arbitrary Lagrangian-Eulerian (ALE) methodology for solving multidimensional fluid flow problems is to move the computational mesh, using the flow as a guide, to improve the robustness, accuracy and efficiency of a simulation. The main elements in the ALE simulation are an explicit Lagrangian phase, a rezone phase in which a new mesh is defined, and a remapping (conservative interpolation) phase, in which the Lagrangian solution is transferred to the new mesh. In most ALE codes, the main goal of the rezone phase is to maintain high quality of the rezoned mesh. In this article, we describe a new rezone strategy which minimizes the  $L_2$  norm of the solution error and maintains smoothness of the mesh. The efficiency of the new method is demonstrated with numerical experiments. © 2005 Wiley Periodicals, Inc. Numer Methods Partial Differential Eq 21: 000–000, 2005

*Keywords:* ALE methods; moving meshes; hyperbolic conservation laws

## I. INTRODUCTION

In a numerical simulation of a fluid flow, the relationship of the motion of a computational mesh to the motion of the fluid is an important issue. There are two choices that are typically made; one represents the Lagrangian methodology and the other represents the Eulerian methodology. In the Lagrangian methodology, the mesh moves with a local fluid velocity, while in the Eulerian methodology the fluid flows through a mesh fixed in space.

In general, the motion of the mesh can be chosen arbitrarily. The objective of the Arbitrary Lagrangian-Eulerian (ALE) methodology [1–6] is to exploit this freedom to improve the robustness, accuracy and efficiency of the simulation. The main elements in an ALE simulation are an explicit Lagrangian phase in which mesh moves with the fluid, a rezone phase in which

*Correspondence to:* Konstantin Lipnikov, Theoretical Division T-7, Los Alamos National Laboratory, MS B284, Los Alamos, NM 87545 (e-mail: lipnikov@lanl.gov)

Contract grant sponsor: DOE/ASCR Program in the Applied Mathematical Sciences

Contract grant sponsor: Laboratory Directed Research and Development program (LDRD)

Contract grant sponsor: DOE's Accelerated Strategic Computing Initiative (ASCI)

© 2005 Wiley Periodicals, Inc. \*This article is a US Government work and, as such, is in the public domain in the United States of America.

a new mesh is defined, and a remapping phase in which the Lagrangian solution is transferred to the rezoned mesh.

The Lagrangian phase of the ALE methodology is well developed [7–9]. It conserves the mass, momentum, and total energy. This property is very important for modeling high-speed compressible flows with shocks, which is our primary interest. The remapping algorithms are described in [10, 11]. There are two principally different rezone strategies. The first changes the mesh only when it becomes tangled. Its main drawback is that the errors due to a mesh distortion are accumulated with time. This may nullify possible advantages of the “natural adaptivity” of the Lagrangian mesh and require global re-meshing. The second strategy performs rezoning regularly, say after  $k$  (usually,  $k \leq 10$ ) Lagrangian steps. It maintains high mesh quality which results in reduction of the discretization error. An example of such a strategy is the Reference Jacobian Matrix (RJM) method [12].

It is intuitively clear that we may improve accuracy of an ALE simulation even more if the rezoned mesh has high quality and at the same time minimizes some measure of a solution error. In this article, we developed a new Error-Minimization-Based (EMB) rezone strategy. In the EMB method, the mesh is changed at time moment  $t^n$  in such a way that the  $L_2$  norm of the error at time  $t^{n+1} = t^n + \Delta t^n$  is minimized. This error is a superposition of the interpolation error, an error due to a time integration method and the space discretization error. The rigorous numerical analysis for a model problem shows that under reasonable assumptions of the mesh smoothness, accuracy of interpolation and time step, the leading term in the error depends on accuracy with which the solution at time moment  $t^n$  is represented by its exact mean values. A minimization of the leading term is the well-known interpolation problem of the best piecewise constant fit with adjustable nodes [13].

Developing an error-based rezone strategy is a challenging task for the system of gas dynamics equations in 2D and 3D. In this article, we focus mostly on a simpler problem, the 1D viscous Burgers’ equation. This equation has many important features of gas dynamics equations: it expresses conservation law, can develop shock-like structures, and its Lagrangian form resembles one for a full system of gas dynamics equations. In Section VI, we explain how the developed rezone strategy can be applied to the system of 1D gas dynamics equations.

The outline of our article is as follows. In the Section II, we describe the conservative Lagrangian form of Burgers’ equation. In Section III, we describe the ALE method. The EMB rezone strategy is introduced in Section IIIB. In Section IV, we present the analysis of the EMB rezone strategy. In Section V, we describe a numerical algorithm which minimizes the error and produces smooth meshes at the same time. In Section VI, we revise the EMB rezone strategy for the system of 1D gas dynamics equations. The results of numerical experiments are presented in Section VII.

## II. VISCIOUS BURGERS’ EQUATION

The standard Eulerian form of 1D viscous Burgers’ equation is

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \varepsilon \frac{\partial^2 u}{\partial x^2}, \quad t \in (0, T), \quad (2.1)$$

where  $\varepsilon \ll 1$  is a given constant. The solution to (2.1) is a smooth function which may have sharp gradients for small  $\varepsilon$ .

In the Lagrangian framework, the trajectory of a point with the Lagrangian coordinate  $\xi$  is given by

$$\frac{d}{dt} x(\xi, t) = u(x(\xi, t), t), \quad x(\xi, 0) = x(0). \quad (2.2)$$

Let  $\mathcal{J}(\xi)$  be Jacobian of the transformation from Eulerian to Lagrangian coordinates,

$$\mathcal{J} = \frac{\partial x}{\partial \xi}.$$

Since the characteristics do not cross, the Jacobian  $\mathcal{J}$  is always positive.

Using the definition of material derivative (also called the substantial derivative or the total derivative), we can write Burgers' equation in the following form:

$$\frac{d}{dt} (\mathcal{J}(\xi)u) - \frac{\partial}{\partial \xi} \left( \frac{u^2}{2} \right) = \varepsilon \frac{\partial}{\partial \xi} \left( \frac{1}{\mathcal{J}(\xi)} \frac{\partial u}{\partial \xi} \right) = \varepsilon \frac{\partial}{\partial \xi} \left( \frac{\partial u}{\partial x} \right), \quad (2.3)$$

which we shall refer to as the conservative Lagrangian form of Burgers' equation. The details of derivation of (2.3) can be found in [14, Chapter 37].

We solve Burgers' equation on a finite interval with boundary points moving with the prescribed velocities and the initial condition

$$u(x, 0) = U(x),$$

where  $U(x)$  is a given function.

### III. THE ALE METHOD

#### A. The Lagrangian Phase

In the Lagrangian phase of the ALE method, we use discrete forms of Eqs. (2.2) and (2.3). Let us introduce the superscript  $n$  which specifies the time moment for all discrete quantities. At time moment  $t = t^n$ , a mesh  $\{x_i^n\}$  is given by

$$x_0^n < x_1^n < \dots < x_{M+1}^n.$$

Let  $h_{i+1/2}^n$  be the mesh step,  $h_{i+1/2}^n = x_{i+1}^n - x_i^n$ , and  $x_{i+1/2}^n$  be the middle points of mesh cell  $[x_i, x_{i+1}]$ . The continuous function  $u(x(\xi, t), t)$  is represented by cell-based values  $\{\bar{u}_{i+1/2}^n\}$  where

$$\bar{u}_{i+1/2}^n \approx \bar{u}([x_i^n, x_{i+1}^n], t^n) \stackrel{\text{def}}{=} \frac{1}{h_{i+1/2}^n} \int_{x_i^n}^{x_{i+1}^n} u(x, t^n) dx.$$

Hereafter, we use  $\{\cdot\}$  to denote a set of either mesh point or values of a piecewise constant function leaving no room for confusion.

#### 4 LIPNIKOV AND SHASHKOV

Let us consider the time-explicit discretization of (2.2), (2.3):

$$\frac{h_{i+1/2}^{n+1}\bar{u}_{i+1/2}^{n+1} - h_{i+1/2}^n\bar{u}_{i+1/2}^n}{\Delta t^n} - \frac{1}{2}((u_{i+1}^n)^2 - (u_i^n)^2) = \varepsilon \left( \left[ \frac{\delta \bar{u}^n}{\delta x} \right]_{i+1} - \left[ \frac{\delta \bar{u}^n}{\delta x} \right]_i \right) \quad (3.1)$$

$$x_i^{n+1} = x_i^n + \Delta t^n u_i^n,$$

where  $\Delta t^n = t^{n+1} - t^n$  denotes the time step,  $h$  plays the role of  $\mathcal{J}$ ,  $u_i^n$  denotes nodal velocity obtained by linear interpolation of cell-centered values,

$$u_i^n = \frac{h_{i-1/2}^n \bar{u}_{i+1/2}^n + h_{i+1/2}^n \bar{u}_{i-1/2}^n}{h_{i-1/2}^n + h_{i+1/2}^n}, \quad (3.2)$$

and

$$\left[ \frac{\delta \bar{u}^n}{\delta x} \right]_i = \frac{\bar{u}_{i+1/2}^n - \bar{u}_{i-1/2}^n}{0.5(h_{i+1/2}^n + h_{i-1/2}^n)} \approx \frac{\partial u}{\partial x}(x_i^n, t^n).$$

The sequence of computations in the Lagrangian phase consists of three steps. First, we compute nodal velocities according to (3.2). Then, we compute new coordinates of mesh nodes using the second formula in (3.1). Finally, we update cell-centered values  $\{\bar{u}_{i+1/2}^{n+1}\}$  using the first formula in (3.1). The time step  $\Delta t^n$  is chosen according to the following practical stability condition

$$\Delta t^n \leq \min_i \left( \frac{|\bar{u}_{i+1/2}^n|}{h_{i+1/2}^n} + \frac{2\varepsilon}{(h_{i+1/2}^n)^2} \right)^{-1}, \quad (3.3)$$

which can be found, for example, in [15, Section 1.3.2].

#### B. The EMB Rezone Phase

Let us consider the time moment  $t^0$ . We assume that the initial mesh  $\{x_i^0\}$  is such that all features of function  $U(x)$  are well resolved. In order to achieve this, we may use the concept of the best piecewise constant fit described in [13].

Now, let us consider a time moment  $t^n$  and choose a mesh for the next Lagrangian step. Besides the computational mesh  $\{x_i^n\}$ , we know the mean values  $\{\bar{u}_{i+1/2}^n\}$ . We can start the Lagrangian step using these data or can use a new mesh  $\{\tilde{x}_i^n\}$  and the corresponding conservatively interpolated mean values  $\{\bar{u}_{i+1/2}^n\}$ . In this article, we follow the second approach.

In general, the new mesh  $\{\tilde{x}_i^n\}$  can be chosen independently of the Lagrangian mesh  $\{x_i^n\}$ ; however, there are some advantages of keeping it close to the Lagrangian one [5]. The ALE method uses this freedom to improve accuracy and stability of the simulation. Since the computational mesh is changed, the Lagrangian solution should be transferred (interpolated) onto the rezoned mesh. We denote the interpolated solution by  $\{\bar{u}_{i+1/2}^n\}$  and use the time integration scheme (3.1) with  $\bar{u}_{i+1/2}^n$  instead of  $\bar{u}_{i+1/2}^n$  and  $\tilde{x}_i^n$  instead of  $x_i^n$ .

The new (rezoned) mesh is defined as a solution of the minimization problem

$$\min_{\tilde{x}_1^n, \dots, \tilde{x}_M^n} \Phi(\{\tilde{x}_i^n\}), \quad \Phi(\{\tilde{x}_i^n\}) = \sum_{i=0}^M \int_{x_i^{n+1}}^{x_{i+1}^{n+1}} (u(x, t^{n+1}) - \bar{u}_{i+1/2}^{n+1})^2 dx, \quad (3.4)$$

posed over a class of smooth meshes  $\{\tilde{x}_i^n\}$ . Note that both the solution values  $\bar{u}_{i+1/2}^{n+1}$  and the integration limits  $x_i^{n+1}$  are functions of  $\tilde{x}_i^n$ . The notion of a mesh smoothness is introduced in the next section. The numerical algorithm minimizing (3.4) and resulting in a smooth mesh is presented in Section V.

The error functional  $\Phi$  represents the  $L_2$  norm of error at time moment  $t^{n+1}$ . This error is a superposition of a remapping error, the error due to the time advancing method (3.1), and the space discretization error. It is clear that if  $\{\tilde{x}_i^n\} \equiv \{x_i^n\}$  (no rezone phase), then the interpolation error is zero; however, the mesh at time moment  $t^{n+1}$  may not be the best mesh to represent new solution features. The goal of minimization problem (3.4) is to achieve balance between the errors (if possible) which minimizes the overall error at time moment  $t^{n+1}$ .

We shall refer to the above rezone strategy as the EMB rezone method. The theoretical analysis of the EMB method is given in Section IV.

### C. The Remapping Phase

In a remapping phase of the ALE method, we use a conservative second-order accurate interpolation algorithm based on a piecewise linear reconstruction [10]:

$$\tilde{u}(x) = \bar{u}_{i+1/2}^n + s_{i+1/2}^n(x - x_{i+1/2}^n), \quad x \in [x_i^n, x_{i+1}^n],$$

where  $s_{i+1/2}^n$  is a limited slope. More precisely, we use the minmod limiter in Section VIIC and the Barth-Jespersen limiter in Section VIID [16]. For a given mesh  $\{\tilde{x}_i^n\}$ , the remapped function is computed by integrating  $\tilde{u}(x)$ :

$$\bar{u}_{i+1/2}^n = \frac{1}{\tilde{h}_{i+1/2}^n} \int_{\tilde{x}_i^n}^{\tilde{x}_{i+1}^n} \tilde{u}(x) dx. \quad (3.5)$$

This remapping algorithm is exact for linear functions and conservative, i.e.

$$\sum_{i=0}^M \bar{u}_{i+1/2}^n \tilde{h}_{i+1/2}^n = \sum_{i=0}^M \bar{u}_{i+1/2}^n h_{i+1/2}^n.$$

The proof follows easily from the fact that the minmod limiter does not change slopes of linear functions.

Note that higher order reconstruction methods can be used for smooth solutions. The analysis presented in the next section does not rely on a particular method but rather assumes that the remapping is at least second-order accurate. In two dimensions, the second-order accurate remapping method has been proposed in [11].

## IV. ERROR ANALYSIS OF THE EMB METHOD

Let us introduce a small parameter  $h = 1/(M + 1)$ . All accuracy estimates will be formulated with respect to this small parameter. In order to simplify the analysis, we make a few assumptions.

**A1** We assume that solution  $u(x, t)$  has bounded second derivatives and data associated with mesh  $\{x_i^n\}$  are exact.

**A2** We assume that there are constants  $c_h$ ,  $C_h$  and  $C_{\Delta h}$ , independent of  $h$ , such that

$$c_h h \leq \tilde{h}_{i+1/2}^n \leq C_h h \quad \text{and} \quad |\tilde{h}_{i+1/2}^n - \tilde{h}_{i-1/2}^n| \leq C_{\Delta h} h^2. \quad (4.1)$$

**A3** We assume that the remapping algorithm from mesh  $\{x_i^n\}$  to mesh  $\{\tilde{x}_i^n\}$  is second-order accurate. In other words, the interpolated mean values are second-order approximations of the exact ones:

$$\bar{u}_{i+1/2}^n = \bar{u}([\tilde{x}_i^n, \tilde{x}_{i+1}^n], t^n) + O(h^2). \quad (4.2)$$

Our numerical experiments confirm that these are valid assumptions for problems with smooth solutions. The problem of building meshes satisfying (4.1) is addressed in Section V.

Let us define the following function:

$$e_{i+1/2}(x, t^{n+1}) = u(x, t^{n+1}) - \bar{u}_{i+1/2}^{n+1}, \quad x \in [x_i^{n+1}, x_{i+1}^{n+1}]. \quad (4.3)$$

This function represents the error between the continuous solution and its piecewise constant approximation at time moment  $t^{n+1}$ .

We begin with finding useful estimates for  $u(x, t^{n+1}) = u(x(\xi, t^{n+1}), t^{n+1})$  when  $x \in [x_i^{n+1}, x_{i+1}^{n+1}]$  and correspondingly  $\xi \in [\xi_i, \xi_{i+1}]$ . For our analysis, it will be enough to use the Taylor expansion of the first-order at point  $(\xi, t^n)$ :

$$u(x(\xi, t^{n+1}), t^{n+1}) = u(x(\xi, t^n), t^n) + \Delta t^n \frac{du}{dt}(x(\xi, t^n), t^n) + O((\Delta t^n)^2). \quad (4.4)$$

Using definition of the material derivative and Burgers' equation, we rewrite Eq. (4.4) as follows:

$$u(x(\xi, t^{n+1}), t^{n+1}) = u(x(\xi, t^n), t^n) + \varepsilon \Delta t^n \frac{\partial^2 u}{\partial x^2}(x(\xi, t^n), t^n) + O((\Delta t^n)^2).$$

The stability conditions (3.3) implies that  $\varepsilon \Delta t^n < (h_{i+1/2}^n)^2/2$ . Thus, the  $\varepsilon$ -term in the last equation is  $O(h^2)$  and therefore

$$u(x(\xi, t^{n+1}), t^{n+1}) = u(x(\xi, t^n), t^n) + O(h^2 + (\Delta t^n)^2). \quad (4.5)$$

Now, the first equation in (3.1) yields

$$\bar{u}_{i+1/2}^{n+1} = \frac{\tilde{h}_{i+1/2}^n}{h_{i+1/2}^{n+1}} \bar{u}_{i+1/2}^n + \frac{\Delta t^n}{h_{i+1/2}^{n+1}} \frac{(\tilde{u}_{i+1}^n)^2 - (\tilde{u}_i^n)^2}{2} + \frac{\varepsilon \Delta t^n}{h_{i+1/2}^{n+1}} \left( \left[ \frac{\delta \bar{u}^n}{\delta x} \right]_{i+1} - \left[ \frac{\delta \bar{u}^n}{\delta x} \right]_i \right). \quad (4.6)$$

The equation of mesh movement [second equation in (3.1)] implies that

$$h_{i+1/2}^{n+1} = \tilde{h}_{i+1/2}^n + \Delta t^n (\tilde{u}_{i+1}^n - \tilde{u}_i^n)$$

and therefore,

$$\frac{\tilde{h}_{i+1/2}^n}{h_{i+1/2}^{n+1}} = 1 - \Delta t^n \frac{\partial u}{\partial x}(\tilde{x}_{i+1/2}^n, t^n) + O(h\Delta t^n). \quad (4.7)$$

Let us show that the last term in the right-hand side of (4.6) is  $O(h^2)$ . Using (4.2) and the fact that exact mean value  $\bar{u}([\tilde{x}_i^n, \tilde{x}_{i+1}^n], t^n)$  differs from the function value at the middle point  $\tilde{x}_{i+1/2}^n$  by  $O(h^2)$ , we can prove that

$$\left[ \frac{\delta \bar{u}^n}{\delta x} \right]_i = \frac{\partial u}{\partial x}(\tilde{x}_i^n, t^n) + O(h).$$

This equation, the previously used estimate for  $\varepsilon \Delta t^n$ , and (4.7) imply that the  $\varepsilon$ -term in (4.6) is  $O(h^2)$ , i.e.,

$$\frac{\varepsilon \Delta t^n}{h_{i+1/2}^{n+1}} \left( \left[ \frac{\delta \bar{u}^n}{\delta x} \right]_{i+1} - \left[ \frac{\delta \bar{u}^n}{\delta x} \right]_i \right) = O(h^2). \quad (4.8)$$

To estimate the first term in the right-hand side of (4.6), we use again (4.7):

$$\frac{\tilde{h}_{i+1/2}^n}{h_{i+1/2}^{n+1}} \tilde{u}_{i+1/2}^n = \bar{u}_{i+1/2}^n - \Delta t^n u(\tilde{x}_{i+1/2}^n, t^n) \frac{\partial u}{\partial x}(\tilde{x}_{i+1/2}^n, t^n) + O(h\Delta t^n). \quad (4.9)$$

Let us analyze the second term in the right-hand side of (4.6). Using Assumptions A2 and A3, one can prove that

$$\frac{\Delta t^n}{h_{i+1/2}^{n+1}} \frac{(\tilde{u}_{i+1}^n)^2 - (\tilde{u}_i^n)^2}{2} = \Delta t^n u(\tilde{x}_{i+1/2}^n, t^n) \frac{\partial u}{\partial x}(\tilde{x}_{i+1/2}^n, t^n) + O(h\Delta t^n). \quad (4.10)$$

Substituting (4.8), (4.9) and (4.10) into (4.6) we conclude that

$$\bar{u}_{i+1/2}^{n+1} = \bar{u}_{i+1/2}^n + O(h^2 + h\Delta t^n). \quad (4.11)$$

Finally, definition (4.3) and Eqs. (4.5) and (4.11) imply that

$$\begin{aligned} e_{i+1/2}(x(\xi, t^{n+1}), t^{n+1}) &= u(x(\xi, t^{n+1}), t^{n+1}) - \bar{u}_{i+1/2}^{n+1} \\ &= u(x(\xi, t^n), t^n) - \bar{u}([\tilde{x}_i^n, \tilde{x}_{i+1}^n], t^n) + O((h + \Delta t^n)^2). \end{aligned}$$

Thus, we proved the following result.

## 8 LIPNIKOV AND SHASHKOV

**Theorem 4.1.** *Under Assumption A1–A3, the leading term in the error functional  $\Phi(\{\tilde{x}_i^n\})$  depends only on the accuracy of representation of function  $u(x, t)$  at time moment  $t^n$  by its exact mean values, i.e.,*

$$\Phi(\{\tilde{x}_i^n\}) = \Phi_0(\{\tilde{x}_i^n\}) + O((h + \Delta t^n)^3),$$

where

$$\Phi_0(\{\tilde{x}_i^n\}) = \sum_{i=0}^M \int_{\tilde{x}_i^n}^{\tilde{x}_{i+1}^n} (u(x, t^n) - \bar{u}([\tilde{x}_i^n, \tilde{x}_{i+1}^n], t^n))^2 dx. \quad (4.12)$$

What is remarkable in Theorem 4.1 is that problem of minimizing functional (4.12) is nothing else but the standard interpolation problem of the best piecewise constant fit with adjustable nodes. This problem has been studied, for instance, in [13].

The functional  $\Phi_0$  is suitable for the theoretical analysis but cannot be used in numerical algorithms. Using the mesh regularity assumption A2, it is easy to show that

$$\Phi_0(\{\tilde{x}_i\}) = \Phi_1(\{\tilde{x}_i\}) + O(h^3), \quad \Phi_1(\{\tilde{x}_i\}) = \frac{1}{12} \sum_{i=0}^M \left[ \frac{\delta \bar{u}^n}{\delta x} \right]_{i+1/2}^2 \tilde{h}_{i+1/2}^3, \quad (4.13)$$

where  $[\delta \bar{u}^n / \delta x]_{i+1/2}$  is a first-order approximation of derivative of  $u$  at point  $\tilde{x}_{i+1/2}^n$ . Thus, functional  $\Phi_1$  is close to functional  $\Phi_0$  and is suitable for numerical algorithms.

In general, the evaluation of the derivative  $\partial u / \partial x$  is based on *a posteriori* error estimates. In one dimension, these estimates are quite simple. First, we approximate  $\partial u / \partial x$  on the mesh  $\{x_i^n\}$  by a piecewise constant function  $\{[\delta \bar{u}^n / \delta x]_{i+1/2}\}$ ,

$$\left[ \frac{\delta \bar{u}^n}{\delta x} \right]_{i+1/2} = \alpha_{i+1/2} \left[ \frac{\delta \bar{u}^n}{\delta x} \right]_{i+1} + (1 - \alpha_{i+1/2}) \left[ \frac{\delta \bar{u}^n}{\delta x} \right]_i, \quad (4.14)$$

where

$$\alpha_{i+1/2} = \frac{h_{i-1/2}^n + h_{i+1/2}^n / 2}{h_{i-1/2}^n + h_{i+1/2}^n + h_{i+3/2}^n}.$$

Second, we interpolate this discrete function onto the mesh  $\{\tilde{x}_i^n\}$  with the remapping algorithm described in Section III C and use the result to compute functional  $\Phi_1$ . Since formula (4.14) is second-order accurate, the estimate for  $\partial u / \partial x$  on the rezoned mesh is second-order accurate too.

## V. ACHIEVING MESH REGULARITY

The error analysis performed in the previous section assumes that all meshes are smooth. Therefore, we need an algorithm which minimizes functional (4.13) over a class of smooth meshes. This is especially important for the regions where solution  $u$  is constant, i.e., the local



error is zero for any mesh. In this section, we propose an algorithm which solves this problem. For the sake of simplicity, we shall omit the superscript “ $n$ ” throughout this section.

We employ the idea of the guaranteed smoothing proposed in [17]. Let  $\alpha$  be a positive number (in practice,  $\alpha$  is greater than 1). The mesh  $\{\tilde{x}_i\}$  is called smooth if the mesh steps satisfy the following condition:

$$\frac{\alpha}{\alpha + 1} \leq \frac{\tilde{h}_{i-1/2}}{\tilde{h}_{i+1/2}} \leq \frac{\alpha + 1}{\alpha}, \quad i = 1, \dots, M. \quad (5.1)$$

For a fixed number  $M$ , the smooth mesh satisfying (5.1) does not have very small space intervals. This guarantees that the time step,  $\Delta t$ , is bounded from below. The argument is to show that in the extreme case when the mesh is geometrically refined to point  $\tilde{x}_{M+1}$ , the minimal mesh step is given by

$$\tilde{h}_{M+1/2} = (\tilde{x}_{M+1} - \tilde{x}_0) \frac{\alpha^M}{(1 + \alpha)^{M+1} - \alpha^{M+1}}.$$

One way to build a smooth mesh is to apply algorithms from [17] to the minimizer of functional  $\Phi_1$ . Another approach is to modify the minimization problem in such a way that its solution will be a smooth mesh. The second approach requires less computational resources than the first one. This is due to better properties of the modified error functional and the faster convergence of nonlinear optimization algorithms.

We begin with deriving the necessary conditions for extrema of functional  $\Phi_0$ . The straightforward differentiation of  $\Phi_0$  with respect to  $\tilde{x}_i$  gives

$$\frac{\partial \Phi_0}{\partial \tilde{x}_i} = 2u(\tilde{x}_i) - \bar{u}([\tilde{x}_{i-1}, \tilde{x}_i], t) - \bar{u}([\tilde{x}_i, \tilde{x}_{i+1}], t) = 0. \quad (5.2)$$

Using the Taylor expansions at point  $\tilde{x}_i$  and neglecting terms of third order and higher, we get

$$\frac{\partial u}{\partial x} \Big|_{\tilde{x}_i} (\tilde{h}_{i+1/2} - \tilde{h}_{i-1/2}) + \frac{1}{3} \frac{\partial^2 u}{\partial x^2} \Big|_{\tilde{x}_i} (\tilde{h}_{i+1/2}^2 + \tilde{h}_{i-1/2}^2) = 0. \quad (5.3)$$

Now we assume that the mesh satisfying (5.3) is described by a smooth function  $\tilde{h}(x)$ , i.e.,  $\tilde{h}_{i+1/2} = \tilde{h}(x_{i+1/2})$ . Then

$$\frac{\tilde{h}_{i+1/2} - \tilde{h}_{i-1/2}}{(\tilde{h}_{i+1/2} + \tilde{h}_{i-1/2})/2} \approx \frac{\partial \tilde{h}}{\partial x} \Big|_{\tilde{x}_i} \quad \text{and} \quad \tilde{h}_{i+1/2}^2 + \tilde{h}_{i-1/2}^2 \approx 2[\tilde{h}(x_i)]^2.$$

Substituting these estimates into (5.3), we get the following equation at point  $\tilde{x}_i$ :

$$\frac{\partial}{\partial x} \left( \ln \left| \frac{\partial u}{\partial x} \right|^{2/3} \right) = - \frac{\partial}{\partial x} (\ln \tilde{h}(x)).$$

The derivation of this equation assumes that  $\partial u/\partial x \neq 0$ . The opposite case means that any mesh can be used in regions where the solution is constant. In actual computations, we avoid this problem by modifying functional  $\Phi_1$ .

It can be shown that the same equation characterizes the minimizer of functional  $\Phi_1$ . Thus, the mesh  $\{\tilde{x}_i\}$  minimizing functional  $\Phi_1$  satisfies approximately the following condition:

$$\left[ \frac{\delta \bar{u}}{\delta x} \right]_{i+1/2}^{2/3} \tilde{h}_{i+1/2} = \text{constant}. \quad (5.4)$$

In other words, in order to build a smooth mesh, we may apply algorithms of the guaranteed smoothing to the piecewise constant function  $\{[\delta \bar{u}/\delta x]_{i+1/2}^{2/3}\}$ .

**Lemma 5.1.** *Let  $\{\omega_{i+1/2}\}$  be a given function such that  $\omega_{i+1/2} \geq 0$ . Consider the system of  $M + 1$  linear equations:*

$$\tilde{\omega}_{i+1/2} - \alpha(\alpha + 1)(\tilde{\omega}_{i+3/2} - 2\tilde{\omega}_{i+1/2} + \tilde{\omega}_{i-1/2}) = \omega_{i+1/2}, \quad (5.5)$$

where  $\tilde{\omega}_{-1/2} = \tilde{\omega}_{1/2}$  and  $\tilde{\omega}_{M+3/2} = \tilde{\omega}_{M+1/2}$ . Then, solution  $\{\tilde{\omega}_{i+1/2}\}$  satisfies

$$\frac{\alpha}{\alpha + 1} \leq \frac{\tilde{\omega}_{i+1/2}}{\tilde{\omega}_{i-1/2}} \leq \frac{\alpha + 1}{\alpha}, \quad i = 1, \dots, M.$$

The proof of Lemma 5.1 can be found in [18, 19]. The lemma provides a simple way for merging the functional minimization with the mesh smoothing. It is pertinent to note that relaxation schemes based on the Gauss-Seidel sweeps (see e.g., [20]) do not have the smoothing property of Lemma 5.1.

For a sufficiently smooth solution, the second term in the left-hand side of (5.5) is  $O(h^2)$ . Thus, the operator acting on  $\{\tilde{\omega}_{i+1/2}\}$  is close to the identity operator. This consideration gives a simple explanation of why the solution to (5.5) preserves main features of the right-hand side.

The relationship between  $\tilde{\omega}_{i+1/2}$  from (5.5) and the right-hand side  $\{\omega_{i+1/2}\}$  can be formalized as follows:

$$\tilde{\omega}_{i+1/2} = \mathcal{S}_{\alpha, i+1/2}(\{\omega_{i+1/2}\}),$$

where  $\mathcal{S}_{\alpha, i+1/2}$  is a linear operator. Then, the final minimization problem used in our numerical experiments reads:

$$\min_{\tilde{x}_1, \dots, \tilde{x}_M} \Phi_2(\{\tilde{x}_{ij}\}), \quad \Phi_2(\{\tilde{x}_{ij}\}) = \sum_{i=0}^M [\mathcal{S}_{\alpha, i+1/2}(\{[\delta \bar{u}/\delta x]_{i+1/2}^{2/3}\})]^3 \tilde{h}_{i+1/2}^3. \quad (5.6)$$

One of the interesting problems is to show that a minimizer of functional  $\Phi_2$  results in the reduction of functionals  $\Phi_0$  and  $\Phi_1$ . We shall address this topic in the future research. In this article, we present the results of numerical experiments showing such a connection.

## VI. GAS DYNAMICS EQUATIONS

In this section we revise the EMB rezone strategy for the system of 1D gas dynamics equations written in the Lagrangian form:

$$\begin{aligned} \frac{1}{\rho} \frac{d\rho}{dt} + \frac{\partial u}{\partial x} &= 0, \\ \rho \frac{du}{dt} + \frac{\partial p}{\partial x} &= 0, \\ \rho \frac{d\epsilon}{dt} + p \frac{\partial u}{\partial x} &= 0, \end{aligned} \tag{6.1}$$

where  $\rho$  is the density,  $u$  is the velocity,  $\epsilon$  is the specific internal energy, and  $p$  is the pressure. The pressure is computed from the equation of state,  $p = (\gamma - 1)\rho\epsilon$ , where  $\gamma$  is the ratio of specific heats. The system (6.1) is closed by imposing some initial conditions and boundary conditions for either the velocity or pressure.

In general, the solution to (6.1) may possess contact discontinuities and shock waves. But in this section, we restrict our analysis to problems satisfying assumptions A1–A3. The goal is to show that an analog of Theorem 4.1 is held for problems with smooth solutions.

In our analysis, we shall follow the approach described in previous sections. Let  $\{\bar{x}_i^n\}$  and  $\{\tilde{x}_i^n\}$  be the Lagrangian and rezoned meshes, respectively. We assume that the meshes satisfy assumption A2. The discrete pressure, density and specific internal energy are defined at mesh cells. We assume that assumptions A1 and A3 hold for piecewise constant pressures  $\{\bar{p}_{i+1/2}^n\}$  and  $\{\tilde{p}_{i+1/2}^n\}$ , piecewise constant densities  $\{\bar{\rho}_{i+1/2}^n\}$  and  $\{\tilde{\rho}_{i+1/2}^n\}$ , and piecewise constant internal energies  $\{\bar{\epsilon}_{i+1/2}^n\}$  and  $\{\tilde{\epsilon}_{i+1/2}^n\}$ . The discrete velocities  $\{u_i^n\}$  and  $\{\tilde{u}_i^n\}$  are defined at mesh nodes and satisfy assumption A1 and a slightly modified version of assumption A3. Instead of (4.2), we assume that

$$\tilde{u}_i^n = u(\bar{x}_i^n, t^n) + O(h^2). \tag{6.2}$$

Let  $M_{i+1/2}$  and  $M_i$  be the cell and nodal masses, respectively,

$$M_{i+1/2} = \bar{\rho}_{i+1/2}^n \tilde{h}_{i+1/2}^n \quad \text{and} \quad M_i = (M_{i+1/2} + M_{i-1/2})/2.$$

We consider the time-explicit discretization of (6.1) described in [21]:

$$\frac{u_i^{n+1} - \tilde{u}_i^n}{\Delta t^n} = - \frac{1}{M_i} (\bar{p}_{i+1/2}^n - \bar{p}_{i-1/2}^n),$$

$$u_i^{n+1/2} = \frac{1}{2} (u_i^{n+1} + \tilde{u}_i^n),$$

$$\frac{\bar{\epsilon}_i^{n+1} - \bar{\epsilon}_i^n}{\Delta t^n} = -\frac{\bar{P}_{i+1/2}^n}{M_{i+1/2}}(u_{i+1}^{n+1/2} - u_i^{n+1/2}),$$

$$x_i^{n+1} = \bar{x}_i^n + \Delta t^n u_i^{n+1/2},$$

$$\bar{\rho}_{i+1/2}^{n+1} = \frac{M_{i+1/2}}{h_{i+1/2}^{n+1}}. \quad (6.3)$$

The pressure unknown  $\bar{p}_{i+1/2}^{n+1}$  is computed from the equation of state.

The error at time moment  $t^{n+1}$  consists of errors in the density, velocity, and internal energy. In this section, we analyze the error in the density. The analysis of other errors uses similar arguments and therefore is omitted. Let

$$e_{i+1/2}(x, t^{n+1}) = \rho(x, t^{n+1}) - \bar{\rho}_{i+1/2}^{n+1}, \quad x \in [x_i^{n+1}, x_{i+1}^{n+1}].$$

We begin with deriving a useful estimate for  $\rho(x, t^{n+1}) = \rho(x(\xi, t^{n+1}), t^{n+1})$  when  $x \in [x_i^{n+1}, x_{i+1}^{n+1}]$  and correspondingly  $\xi \in [\xi_i, \xi_{i+1}]$ . The Taylor expansion at point  $(\xi, t^n)$  yields

$$\rho(x(\xi, t^{n+1}), t^{n+1}) = \rho(x(\xi, t^n), t^n) + \Delta t^n \frac{d\rho}{dt}(x(\xi, t^n), t^n) + O((\Delta t^n)^2).$$

The first equation in (6.1) implies that

$$\begin{aligned} \rho(x(\xi, t^{n+1}), t^{n+1}) &= \rho(x(\xi, t^n), t^n) \left( 1 - \Delta t^n \frac{\partial u}{\partial x}(x(\xi, t^n), t^n) \right) + O((\Delta t^n)^2) \\ &= \rho(x(\xi, t^n), t^n) \left( 1 - \Delta t^n \frac{\partial u}{\partial x}(\bar{x}_{i+1/2}^n, t^n) \right) + O(h\Delta t^n + (\Delta t^n)^2). \end{aligned} \quad (6.4)$$

We proceed with deriving an estimate for  $\bar{p}_{i+1/2}^{n+1}$ . Combining the last two equations in (6.3) and using the cell mass definition, we get

$$\begin{aligned} \bar{\rho}_{i+1/2}^{n+1} &= \frac{M_{i+1/2}}{h_{i+1/2}^{n+1}} = \frac{M_{i+1/2}}{\tilde{h}_{i+1/2}^n + \Delta t^n (u_{i+1}^{n+1/2} - u_i^{n+1/2})} \\ &= \bar{\rho}_{i+1/2}^n \frac{\tilde{h}_{i+1/2}^n}{\tilde{h}_{i+1/2}^n + \Delta t^n (u_{i+1}^{n+1/2} - u_i^{n+1/2})}. \end{aligned} \quad (6.5)$$

The first two formulas in (6.3) give the following estimate for velocities:

$$u_{i+1}^{n+1/2} - u_i^{n+1/2} = \tilde{u}_{i+1}^n - \tilde{u}_i^n - \frac{\Delta t}{2} \left[ \frac{1}{M_{i+1}} (\bar{p}_{i+3/2}^n - \bar{p}_{i+1/2}^n) - \frac{1}{M_i} (\bar{p}_{i+1/2}^n - \bar{p}_{i-1/2}^n) \right].$$

The assumptions A2 and (6.2) imply that

$$\tilde{u}_{i+1}^n - \tilde{u}_i^n = \tilde{h}_{i+1/2}^n \frac{u(\tilde{x}_{i+1}^n, t^n) - u(\tilde{x}_i^n, t^n)}{\tilde{h}_{i+1/2}^n} + O(h^2) = \tilde{h}_{i+1/2}^n \frac{\partial u}{\partial x}(\tilde{x}_{i+1/2}^n, t^n) + O(h^2).$$

Using similar arguments, we can easily show that

$$\frac{1}{M_{i+1}} (\bar{p}_{i+3/2}^n - \bar{p}_{i+1/2}^n) = \frac{1}{\rho(\tilde{x}_{i+1}^n, t^n)} \frac{\partial p}{\partial x}(\tilde{x}_{i+1}^n, t^n) + O(h)$$

and

$$\frac{1}{M_{i+1}} (\bar{p}_{i+3/2}^n - \bar{p}_{i+1/2}^n) - \frac{1}{M_i} (\bar{p}_{i+1/2}^n - \bar{p}_{i-1/2}^n) = O(h).$$

Substituting the last estimates in (6.5) and using the Taylor expansion for  $1/(1 + O(\Delta t^n))$ , we get

$$\bar{\rho}_{i+1/2}^{n+1} = \bar{\rho}_{i+1/2}^n \left( 1 - \Delta t^n \frac{\partial u}{\partial x}(\tilde{x}_{i+1/2}^n, t^n) \right) + O(h\Delta t^n + (\Delta t^n)^2). \quad (6.6)$$

Now, estimates (6.4), (6.6), and Assumption A3 imply that

$$\begin{aligned} e_{i+1/2}(x(\xi, t^{n+1}), t^{n+1}) &= \rho(x(\xi, t^{n+1}), t^{n+1}) - \bar{\rho}_{i+1/2}^{n+1} \\ &= \rho(x(\xi, t^n), t^n) - \bar{\rho}([\tilde{x}_i^n, \tilde{x}_{i+1}^n], t^n) + O((h + \Delta t^n)^2). \end{aligned}$$

The last estimate is similar to the error estimate derived for Burgers' equation. Thus, an analog of Theorem 4.1 holds for the system of 1D gas dynamics equations with sufficiently smooth solutions.

## VII. NUMERICAL EXPERIMENTS

In this section we analyze numerically the robustness and efficiency of the EMB rezone strategy first for viscous Burgers' equation and then for the system of 1D gas dynamics equations.

### A. Implementation Issues

We use the nonlinear conjugate gradient method for solving minimization problem (5.6):

$$\begin{aligned} x_i^{(k+1)} &= x_i^{(k)} + \lambda^k d_i^{(k)}, \\ d_i^{(k)} &= -\frac{\delta\Phi_2}{\delta\tilde{x}_i}(\{x_i^{(k)}\}) + \beta^k d_i^{(k-1)}, \quad k = 1, 2, \dots, \end{aligned} \quad (7.1)$$

where  $\{x_i^{(0)}\}$  is the given mesh,  $x_i^{(0)} = x_i$ ,  $d^{(k)}$  is the descent direction, and  $\delta\Phi_2/\delta\tilde{x}_i$  is a finite difference approximation of the continuous derivative  $\partial\Phi_2/\partial\tilde{x}_i$ . In numerical experiments we

use Polak-Ribière's choice for parameter  $\beta^k$  [22]. The parameter  $\lambda^k$  is specified by a line search algorithm:

$$\lambda^k = \underset{\lambda}{\operatorname{argmin}} \Phi_2(\{x_i^{(k)} + \lambda d_i^{(k)}\}). \quad (7.2)$$

The quality of the line search algorithm is crucial to preserve the mutual conjugacy property of the descent directions. The details of efficient numerical methods for solving (7.2) can be found in [22].

The functional  $\Phi_2$  requires evaluation of the solution derivative on the rezoned mesh. We use the conservative interpolation algorithms from Section III C, which preserve the following quantity:

$$E = \sum_{i=0}^M \left[ \frac{\delta \tilde{u}}{\delta x} \right]_{i+1/2}^{2/3} \tilde{h}_{i+1/2}. \quad (7.3)$$

Note that the smoothing is the most time-consuming step in evaluation of  $\Phi_2$ . Therefore, we perform it only once per each conjugate gradient iteration. The smoothed values then used by the remapping method from Section III C to evaluate approximately functional  $\Phi_2$  in the line search algorithm and its derivative in the descent direction computation.

We terminate the nonlinear iterations (7.1) when either  $d^{(k)}$  is no longer a descent direction or the relative change in the position of mesh nodes is less than the user prescribed tolerance  $TOL$ , i.e.,

$$\max_{1 \leq i \leq M} \frac{|x_i^{(k)} - x_i^{(k+1)}|}{x_{i+1}^{(k)} - x_{i-1}^{(k)}} \leq TOL \quad \text{for some } k.$$

In general, the rezoned meshes at time moments  $t^n$  and  $t^{n+1}$  may differ significantly. The main reason is that functional  $\Phi_2$  may have many local minima those locations are not smooth functions of time. It may result in nonphysical time oscillations of mesh points. In order to dump these oscillations, we use a simple time smoothing. A new rezoned mesh is defined as a linear combination of the original mesh and the mesh minimizing functional  $\Phi_2$ :

$$\tilde{x}_i := (x_i + \tilde{x}_i)/2. \quad (7.4)$$

As we mentioned in the introduction, the EMB rezone strategy can be employed once for every  $k$  Lagrangian steps. We shall denote this strategy as EMB( $k$ ).

For the problems with shock-like structures, the gradient of the solution cannot be correctly estimated at the shock location. It may result in very small mesh steps even for smooth meshes. In the numerical experiments we choose dynamically the smoothing parameter  $\alpha$  such that the minimal mesh step is always bigger than  $h/10$ .

## B. The Effect of Mesh Smoothing

Let us consider the viscous Burgers' equation with the exact solution from [23]:

$$u(x, t) = 1 - \frac{0.9r_1(x, t) + 0.5r_2(x, t)}{r_1(x, t) + r_2(x, t) + r_3(x, t)}, \quad (7.5)$$

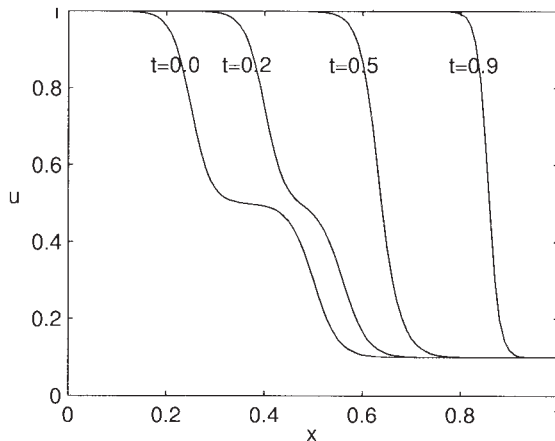


FIG. 1. Viscous Burgers' equation with  $\varepsilon = 0.005$ . Exact solution for different  $t$ .

where

$$r_1 = \exp\left(-\frac{x-0.5}{20\varepsilon} - \frac{99t}{400\varepsilon}\right), \quad r_2 = \exp\left(-\frac{x-0.5}{4\varepsilon} - \frac{3t}{16\varepsilon}\right), \quad r_3 = \exp\left(-\frac{x-3/8}{2\varepsilon}\right).$$

At time moment  $t = 0$ , this function consists of two shock-like structures, which then move with different speeds and finally merge at  $t \approx 0.5$  (see Fig. 1).

We analyze the effect of mesh smoothing on accuracy of representation of initial data by a piecewise constant function. We assume that the exact initial data  $\{u_{i+1/2}^0\}$  are given on the uniform mesh and the smoothing parameter  $\alpha$  is set to 1. Let  $\{\tilde{x}_i^\sigma\}$  be the mesh minimizing functional  $\Phi_\sigma$ ,  $\sigma = 0, 1, 2$ , and  $\{x_i^{mn}\}$  be the uniform mesh. The results of numerical experiments are shown in Table I. We use the square root of functional  $\Phi_0$  as the quality measure of the rezoned mesh.

On the finest mesh, the loss in the solution accuracy due to smoothing is about 3%. On left panel in Fig. 2, we plot functions  $\{[\delta\tilde{u}/\delta x]_{i+1/2}^{2/3}\}$  and  $\{S_{\alpha,i+1/2}(\{[\delta\tilde{u}/\delta x]_{i+1/2}^{2/3}\})\}$ . Observe that the smoothed function preserves the main features of the original function. On the right panel in Fig. 2, we plot the ratio of neighboring mesh steps in mesh  $\{\tilde{x}_i^2\}$ . They satisfy inequality (5.1) with  $\alpha = 1$ .

### C. Viscous Burgers' Equation

In this section, we compare the pure Lagrangian method with two ALE methods using the RJM [12] and EMB rezone strategies. We consider a model problem with the exact solution (7.5).

TABLE I. Accuracy analysis for Burgers' equation:  $t = 0$  and  $\alpha = 1$ .

$M$	$\sqrt{\Phi_0(\{\tilde{x}_i^{mn}\})}$	$\sqrt{\Phi_0(\{\tilde{x}_i^0\})}$	$\sqrt{\Phi_0(\{\tilde{x}_i^1\})}$	$\sqrt{\Phi_0(\{\tilde{x}_i^2\})}$
16	2.99e-2	1.01e-2	1.19e-2	1.75e-2
32	1.59e-2	4.99e-3	5.18e-3	6.28e-3
64	7.99e-3	2.48e-3	2.50e-3	2.70e-3
128	4.00e-3	1.24e-3	1.24e-3	1.28e-3

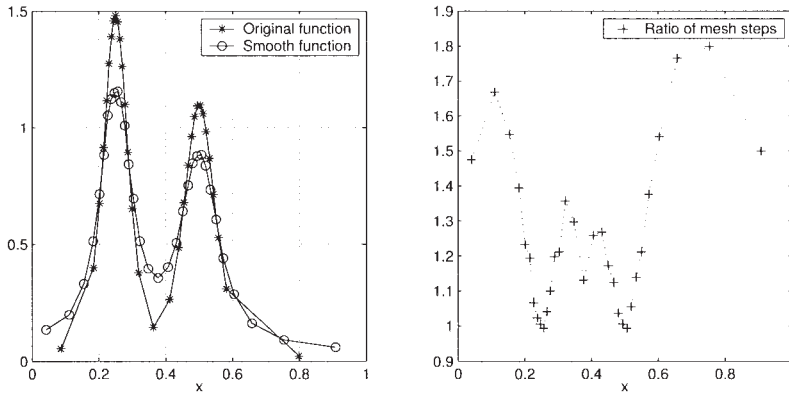


FIG. 2. Effect of smoothing for Burgers' equation:  $M = 32$ ,  $t = 0$  and  $\alpha = 1$ .

In all experiments, the CFL number is equal to 0.5, i.e., we use a time step that is half the size recommended by the stability condition (3.3). We set  $K_{\max} = 50$ ,  $TOL = 10^{-3}$  and do not impose any restrictions on the minimal mesh step. Indeed, in the case of smooth solutions, the minimal mesh step is controlled by the solution gradient, which is bounded.

The trajectories of mesh nodes and discrete solution at the initial and final time moments are presented in Fig. 3. The final time for the ALE simulations is  $t = 0.9$ . We terminated the pure

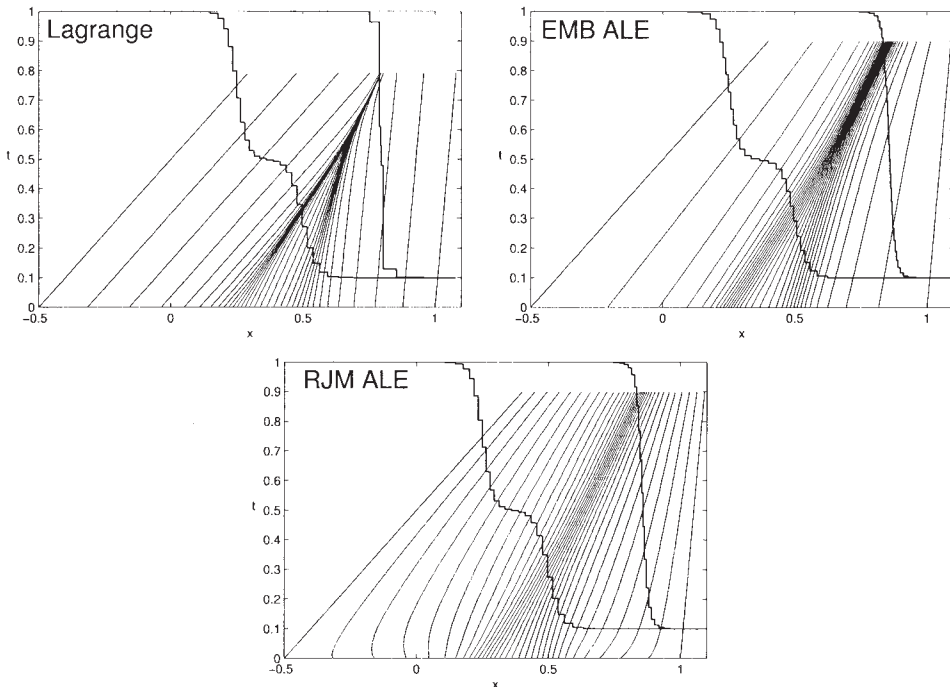


FIG. 3. Viscous Burgers' equation with  $\varepsilon = 0.005$ : trajectories of 32 mesh points and discrete solutions at initial and final time moments.



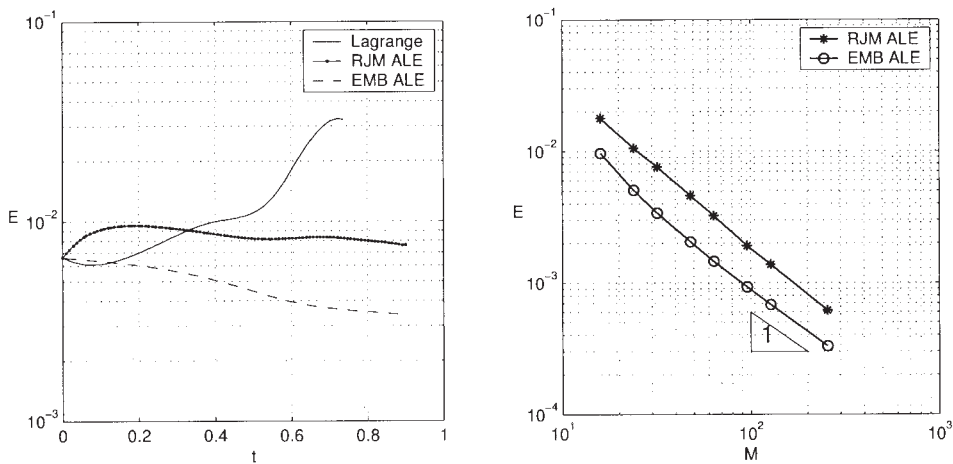


FIG. 4. Viscous Burgers’ equation with  $\varepsilon = 0.005$ . The panel on the left shows the accuracy of the simulations ( $L_2$  norm of error denoted by  $E$ ) versus the time for the case  $M = 32$ . The panel on the right shows convergence rates versus the number of mesh points,  $M$ .

Lagrangian simulation at  $t \approx 0.8$ . Each horizontal slice represents the 1D mesh at the corresponding time moment.

As one can see from Fig. 3, in the pure Lagrangian simulation, the nodes get progressively closer to each other and the calculation practically stalls at  $t \approx 0.8$ . This confirms the statement from [14, Section 37.2.2] that the method of characteristics is not well suited for general hyperbolic partial differential equations and in particular for Burgers’ equation. In the ALE methods, the mesh is smooth and resolves features of the solution at each time moment. It is clear that the trajectories of mesh nodes follow roughly the Lagrangian trajectories.

On the left picture in Fig. 4, we plot the error as a function of time for the case  $M = 32$ . The pure Lagrangian method is accurate up to  $t \approx 0.1$ . After this moment, dynamics of mesh cells does not correspond to dynamics of solution features and accuracy starts to degrade. For the RJM ALE method, the error first starts to grow because the RJM method makes the mesh smoother compared to the original mesh (see Fig. 3) and by doing this, it reduces resolution of solution features. Then, the error stays approximately constant. After  $t \approx 0.3$  the accuracy of the RJM ALE method is significantly better than accuracy of the pure Lagrangian method. For the EMB ALE method, the error is about the same as for the pure Lagrangian method up to  $t \approx 0.15$ , but after this moment, it is significantly less than the errors for the other methods.

The right panel in Fig. 4 demonstrates the linear dependence of  $L_2$ -norm of the error on the number of mesh cells in the ALE methods. However, for the given accuracy, e.g.,  $2 \cdot 10^{-3}$ , we need about 2 times less mesh cells for the EMB ALE method than for the RJM ALE method. We observed in other experiments that this difference was bigger for smaller values of  $\varepsilon$ .

**D. System of Gas Dynamics Equations**

We consider the Sod shock tube problem. The problem consists of two regions of gas with  $\gamma = 1.4$  initially separated by a membrane at  $x = 0.5$ . The gas to the left of the membrane is more dense ( $\rho_L = 1$  and  $\rho_R = 0.125$ ) and at a higher pressure ( $p_L = 1$  and  $p_R = 0.1$ ) compared to the one on the right. We impose the reflecting boundary conditions at  $x = 0$  and  $x = 1$ . The final time is  $t = 1$  so that we can study the interaction of waves generated after removing the membrane.

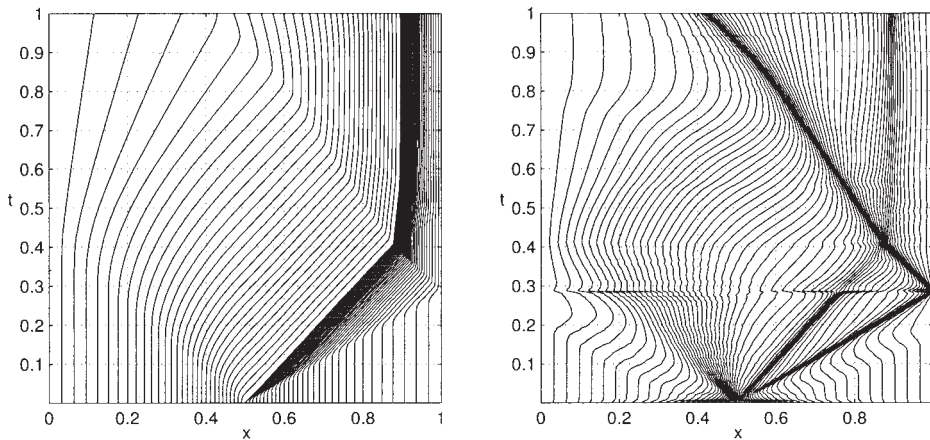


FIG. 5. The Sod problem: the trajectories of the mesh points in the pure Lagrangian (left) and the EMB ALE (right) simulations.

The solution to Sod's problem has the constant discontinuity and develops a shock wave. The discontinuous solutions are not covered by the theory developed in previous sections. However, in the rest of this section, we use formally functional  $\Phi_2$  for the discontinuous density  $\rho$ .

In order to prevent an excessive mesh refinement around solution discontinuities, the smoothing parameter  $\alpha$  is chosen dynamically to keep the minimal mesh step bigger than  $h/10$ .

In all experiments the CFL number is equal to 0.5 and  $M = 120$ . We use the quadratic artificial viscosity to stabilize the numerical scheme. We use the time smoothing strategy (7.4) and set the default value of mesh smoothing parameter  $\alpha = 2$ . In Fig. 5, we show trajectories of mesh nodes with odd indices. We begin simulation with the mesh adapted to a modified density function:

$$\hat{\rho}(x) = \rho(x) + \frac{\rho_R - \rho_L}{2} \left( 1 + \frac{2}{\pi} \tan^{-1} \left( \frac{x - 0.5}{0.005} \right) \right),$$

where  $\rho(x)$  is the initial discontinuous density function.

The left panel in Fig. 5 corresponds to the pure Lagrangian simulation. When the reflected shock waves passes the contact discontinuity, the Lagrangian velocities of the majority of mesh nodes are very small. Therefore, when the shock reaches point  $x = 0.5$ , its resolution becomes very poor. The RJM ALE method makes mesh smoother; however, the qualitative behavior of mesh trajectories resembles one in the pure Lagrangian method. As the result, it practically does not improve accuracy of the final solution. It is pertinent to note that the RJM rezone may be crucial for success of 2D ALE simulations [12].

The right panel in Fig. 5 corresponds to the ALE method with the EMB(1) rezone strategy. The mesh is adapted to the density function. Note that the rarefaction wave, the contact discontinuity and the shock wave are well resolved at all time moments. We note the unphysical mesh movement at time moment  $t \approx 0.28$ . Similar mesh movement has been reported in [17]. It may be reduced by a stronger time smoothing.

In Fig. 6, we plot the density, internal energy and velocity at the final time moment for the pure Lagrangian and EMB ALE methods. It is obvious that the solution obtained with the ALE method is more accurate than the Lagrangian solution. It is pertinent to note one interesting

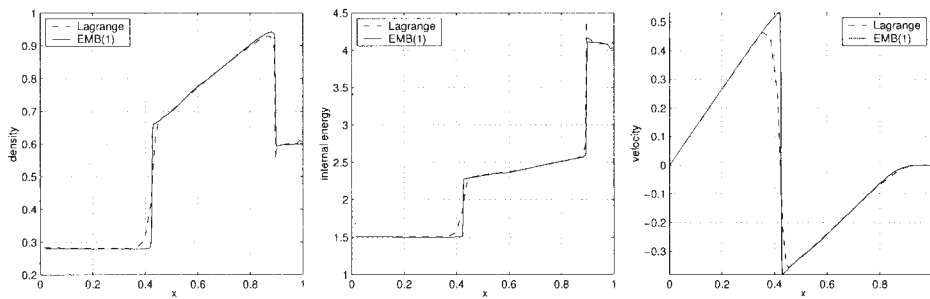


FIG. 6. The Sod problem: comparison of the density, internal energy, and velocity in the pure Lagrangian and EMB ALE methods.

aspect of the ALE method. A natural diffusion presented in the remapping algorithm smears overshoots and undershoots in the Lagrangian solution.

In Figs. 7 and 8, we compare efficiency and accuracy of the EMB ALE and pure Lagrangian methods. The exact solution at time moment  $t = 1$  has been replaced by a discrete solution computed on a mesh with 10,000 mesh cells. We plot the  $L_1$  norm of the error because it is the typical measure of accuracy of discontinuous solutions in the literature. The Hölder inequality implies that the functional  $\Phi_0$  provides an upper estimate for the  $L_1$  norm of the error. Hence, by minimizing this functional, we reduce the  $L_1$  norm of the error as well.

The left panel in Fig. 7 shows that the ALE methods are more efficient than the pure Lagrangian method. For instance, for the given CPU, e.g., 10s, the errors are approximately 0.0035 and 0.0021 in the Lagrangian and ALE methods, respectively. Moreover, the ALE method requires about 3 times less mesh cells. Since the complexity of the nonlinear conjugate gradient method grows super-linear with respect to the problem size, the pure Lagrangian method may begin to dominate. For the particular example, the Lagrangian mesh must have more than 1000 mesh cells.

On the right panel in Fig. 7, we demonstrate convergence rates for the pure Lagrangian and EMB ALE methods. The pure Lagrangian method converges linearly with the number of mesh

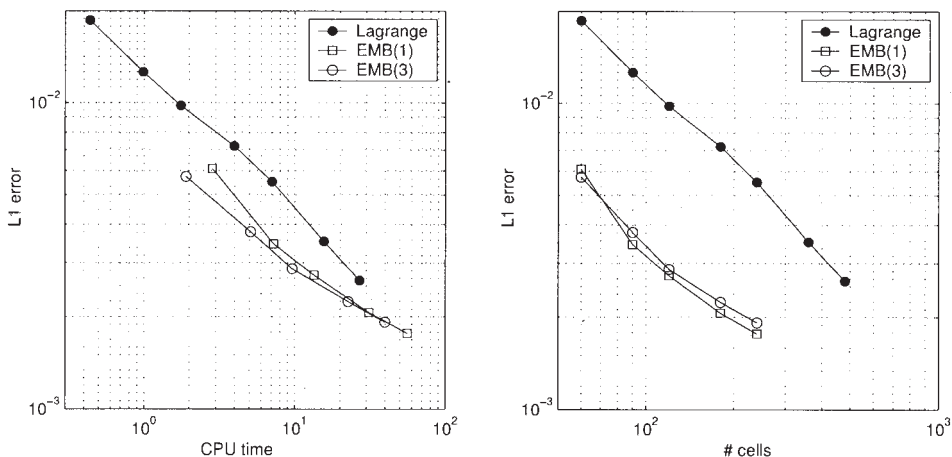


FIG. 7. The Sod problem: the efficiency (left) and convergence (right) for the pure Lagrangian, EMB(1) and EMB(3) ALE methods for initially adapted mesh.

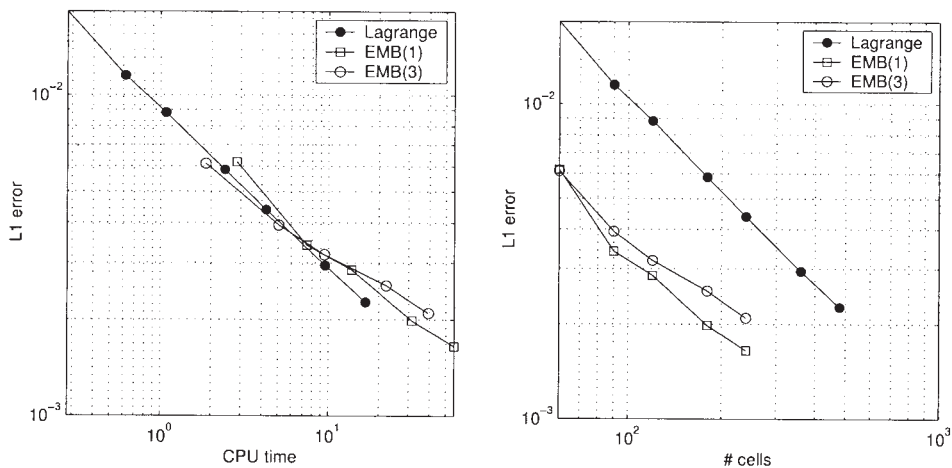


FIG. 8. The Sod problem: the efficiency (left) and convergence (right) for the pure Lagrangian, EMB(1) and EMB(3) ALE methods for initially uniform mesh.

cells. The sublinear convergence rate (in  $L_1$  norm) for the EMB ALE methods is due to minimizing the functional  $\Phi_0$ , which represents the  $L_2$  norm of the error. Nevertheless, the ALE methods are much more accurate on all meshes.

In Fig. 8, we change the setup of the previous experiment by starting the simulations with a uniform mesh. Now, the computational efficiency of the EMB ALE method can be obtained by tuning some of the simulation parameters, e.g., the number of Lagrangian steps between rezone steps. However, the convergence rates are roughly the same as in the previous experiment.

## VIII. CONCLUSION

We have developed a new error-minimization-based rezone strategy for 1D viscous Burgers' equation in the Lagrangian framework. The goal of the rezone strategy is to change the mesh at time moment  $t^n$  in such a way that  $L_2$  norm of the error at time moment  $t^{n+1}$  is minimized. The developed rezone strategy has been successfully applied to the problem with discontinuous solutions. In the future we plan to extend the EMB ALE method to the full system of gas dynamics equations in 2D.

The authors thank B. Wendroff, L. Margolin, M. Hyman, S. Li and R. Garimella (LANL) and Yu. Vassilevski (INM) for many valuable comments. The work was performed at Los Alamos National Laboratory operated by the University of California for the US Department of Energy under contract W-7405-ENG-36, LA-UR-04-7660.

## References

1. D. J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, *Comput Methods Appl Mech Engng* 99 (1992), 235–394.
2. C. W. Hirt, A. Amsden, and J. Cook, An Arbitrary Lagrangian-Eulerian computing method for all flow speeds, *J Comput Phys* 14 (1974), 227–253.
3. D. S. Kershaw, M. K. Prasad, M. J. Shaw, and J. L. Milovich, 3D unstructured mesh ALE hydrodynamics with the upwind discontinuous finite element method, *Comput Methods Appl Mech Engng* 158 (1998), 81–116.

4. P. Kjellgren and J. Hyvarien, An arbitrary Lagrangian-Eulerian finite element method, *Comput Mech* 21 (1998), 81–90.
5. L. G. Margolin, Introduction to “an arbitrary Lagrangian-Eulerian computing method for all flow speeds,” *J Comput Phys* 135 (1997), 202.
6. J. S. Peery and D. E. Carroll, Multi-material ALE methods in unstructured grids, *Comput Methods Appl Mech Engng* 187 (2000), 591–619.
7. E. J. Caramana, D. E. Burton, M. J. Shashkov, and P. P. Whalen, The construction of compatible hydrodynamics algorithms utilizing conservation of total energy, *J Comput Phys* 146 (1998), 227–262.
8. E. J. Caramana and M. J. Shashkov, Elimination of artificial grid distortion and hourglass-type motions by means of Lagrangian subzonal masses and pressures, *J Comput Phys* 142 (1998), 521–561.
9. E. J. Caramana, M. J. Shashkov, and P. P. Whalen, Formulations of artificial viscosity for multidimensional shock wave computations, *J Comput Phys* 144 (1998), 70–97.
10. M. Kucharik, M. Shashkov, and B. Wendroff, An efficient linearity-and-bound-preserving remapping methods, *J Comput Phys* 188 (2003), 462–471.
11. L. G. Margolin and M. Shashkov, Second-order sign-preserving conservative interpolation (remapping) on general grids, *J Comput Phys* 184 (2003), 226–298.
12. P. Knupp, L. G. Margolin, and M. Shashkov, Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods, *J Comput Phys* 176 (2002), 93–128.
13. M. J. Baines, *Moving finite elements*, Oxford University Press, New York, 1994.
14. J. F. Thompson, B. K. Soni, and N. P. Weatherill, editors, *Handbook of grid generation*, CRC Press, Boca Raton, FL, 1999.
15. K. Morton, *Numerical solution of convection-diffusion problems*, Chapman & Hall, London, 1996.
16. C. B. Laney, *Computational gasdynamics*, Cambridge University Press, 1998.
17. E. Dorfi and L. Drury, Simple adaptive grids for 1-D initial value problems, *J Comput Phys* 69 (1987), 175–195.
18. K. Lipnikov and M. Shashkov, Moving meshes for the Burgers equation, Report LA-UR-03-7605, Los Alamos National Laboratory, 2003.
19. P. A. Zegeling, Moving-grid methods for time-dependent partial differential equations, *Centrum voor Wiskunde en Informatica*, Amsterdam, 1992.
20. H. Tang and T. Tang, Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws, *SIAM J Numer Anal* 41 (2003), 487–515.
21. M. Shashkov, *Conservative finite-difference methods on general grids*, CRC Press, Boca Raton, FL, 1996.
22. J. Nocedal and S. Wright, *Numerical optimization*, Springer, Berlin, 1999.
23. L. S. Mulholland, Y. Qui, and D. M. Sloan, Solution of evolutionary partial differential equations using adaptive finite differences with pseudospectral post-processing, *J Comput Phys* 131 (1997), 280–298.