LA-UR-03-2230

Remapping, Recovery and Repair on a Staggered Grid

L. G. Margolin

Applied Physics Division, Los Alamos National Laboratory

MS-B218, Los Alamos, NM 87545

Mikhail Shashkov

Theoretical Division, T-7, Los Alamos National Laboratory

MS-B284, Los Alamos, NM 87545

Subject Classification Numbers: 65N05, 65N10, 65N15, 80A20

Keywords: Remapping, ALE Methods, High Speed Flows

Running Head: **Remapping, Recovery and Repair**

Send Proofs to:

Dr. Mikhail Shashkov

Mail Stop B284

Los Alamos National Laboratory

Los Alamos, NM 87545

Phone: (505) 667-4400

Fax: (505) 665-5757

E-mail: shashkov@lanl.gov

Abstract

An accurate remapping algorithm is an essential component of many Arbitrary Lagrangian-Eulerian (ALE) methods. In previous work, we have described a local remapping algorithm for a positive cell-centered scalar function that is second-order accurate, conservative, and sign preserving. However remapping in the context of high speed flow introduces new issues, which include the consistent treatment of the kinetic and internal energies, compatible remapping of mass and momentum on the staggered mesh, and the generalization from sign preservation to monotonicity preservation. We describe a remap strategy that deals with each of these issues. Although the theoretical development of this strategy is intricate, the resulting scheme is both simple to implement and efficient. We provide numerical examples to illustrate the individual steps of the remap and its overall performance.

1 Introduction

Arbitrary Lagrangian-Eulerian (ALE) methods were first developed for compressible fluid [4] and solid [3] dynamics. For flows with shocks or undergoing large compressions, possibly including multiple materials and (in the case of solids) complex constitutive relations, Lagrangian methods offer many advantages of accuracy over Eulerian methods. However in multidimensional calculations, Lagrangian meshes are prone to tangle, thereby sacrificing any advantage of accuracy. The original ALE methods were designed to preserve the Lagrangian nature of the mesh as much as possible; however, when mesh distortion threatens the accuracy, or perhaps even the survival of the calculation, a rezoning phase is invoked to reduce the undesirable distortion by improving the mesh (see e.g., [6], [15]). A new phase, remapping, is then required to interpolate the solution onto the improved mesh. For a more detailed review of ALE methodology, see [11].

The focus in this paper is on the remapping phase for ALE programs that use a staggered grid. We will restrict ourselves to local remappers, meaning that the rezoned mesh is constrained to be close enough to the original mesh that the remapping process can be viewed as the exchange of conserved quantities between a cell and its nearest neighbors. Our goal is to find remapped values of mass density ρ, velocity u, and specific internal energy I on the rezoned grid, given these fields on the original grid. In a recent paper [13], we described such a local remapping algorithm for a positive, scalar, cell-centered function that is second-order accurate, conservative, and sign preserving. Such an algorithm is suitable for remapping the mass density, but is not immediately applicable to fields located on the nodes of a staggered grid. In the next few paragraphs, we will describe several issues that must be confronted to develop a full function remapper for a compressible fluid dynamics ALE code.

To begin, we consider momentum. Momentum, unlike mass, is not a positive definite quantity. The physical property that is more appropriate to preserve in remapping momentum is monotonicity. Furthermore, most ALE codes applied to high speed flow problems use a staggered mesh, where velocity is stored at nodes (i.e., vertices) while the thermodynamic quantities like density and internal energy are stored at cell centers, [14]. This lack of collocation leads to several problems. First, the algorithm in [13] is not designed for nodal quantities. Further, the velocity is not a conserved quantity, and even though one can define a nodal mass and momentum, it is the cell-centered mass that is remapped. Thus, there is an incompatibility in the remapped values of cell mass and nodal momentum.

There are equally difficult issues concerning the total energy, which is the sum of the internal energy located in the cell centers and the kinetic energy centered on the nodes. In the high speed flows in which we are mainly interested, kinetic energy dominates the internal energy. We note that there is no dynamics in the remap phase of the simulation, and so in principle the kinetic, internal, and total energy are all individually conserved. However only one of these three is independent (note that velocity is recovered from the remap of momentum). In high speed flows, the kinetic energy dominates the internal energy. If one remaps the total energy, then the internal energy must be found by subtraction, and is the small difference of two large numbers. In this case large errors are possible and indeed there is no guarantee that the internal energy will remain positive. On the other hand, if one remaps the internal energy, then total energy conservation will be sacrificed.

In this paper we will describe a remap strategy to address these issues. In particular, we will devise our strategy to:

1) define and remap only cell-centered quantities — density, specific momentum and kinetic energy, specific internal energy. In addition, we will ensure that the forms of the mass, total energy, and momentum conserved during the remap are consistent with those conserved in the Lagrangian phase.

2) exactly conserve total energy, and will nearly conserve internal and kinetic energy. Further, we will guarantee the positivity of the internal energy by ensuring that the remap is locally dissipative.

3) allow the preservation of monotonicity of either $(\rho,\, \rho I,\, \rho u)$ or $(\rho,\, I,\, u)$.

There are several new ideas we have developed to accomplish these goals. The first is the remap of momentum as a cell-centered quantity. Here we have generalized previous work on the advection of nodal quantities by a method of moments [10]. Conceptually, this generalization replaces the velocity gradient by the kinetic energy as the supplemental cell-centered quantity, allowing us to simultaneously conserve momentum and kinetic energy. The second idea is the augmentation of the remapper by a new process that we term repair [7]. In fact, it is only in the repair stage that we enforce monotonicity constraints. We emphasize that preserving monotonicity of the velocity field is an essential feature as it guarantees the nonlinear stability of the remap process. The third idea is the use of coordinated fluxes, whereby the momentum and kinetic energy fluxes are expressed in terms of the mass flux and nodal velocity. This form ensures the realizability of the new velocities, and also leads to significant improvements in the overall efficiency and simplicity of the remap process.

The presentation in this paper involves one-dimensional descriptions and implementations. However each step of the methodology is readily extensible to two or more dimensions. An outline of the rest of this paper is as follows. In section 2, we will describe the individual steps of our remapping procedure, including repair. In section 3, we will offer numerical examples that illustrate the individual steps, and also show the results of repeated applications of the entire remapping package. In section 4 we will summarize our results in $1D$ and discuss future work, including the extension to multiple dimensions.

2 Remap

The remap stage that we describe in this section consists of several steps. First, density and specific internal energy are remapped in a conservative and sign-preserving manner, using the algorithm of [13]. Second, momentum and kinetic energy variables are constructed at the cell centers and remapped. This ensures that momentum and kinetic energy are also conserved; coupled with the conservation of internal energy, we consequently have conservation of total energy. Further, the cell-centered momentum and kinetic energy are consistent with the conserved forms of the Lagrangian phase. Third, the remapped cell-centered values of velocity are used to reconstruct the nodal velocity, essentially by inverting the construction of the second step. Each of the cells adjacent to a node may predict a different estimate of the velocity, and so an additional process of resolving these estimates to recover a single velocity is necessary. The recovery process is dissipative, and we explicitly assign the kinetic energy losses to the internal energy of these adjacent cells. Thus at the end of the third step, momentum and total energy are still conserved, and in a manner consistent with the Lagrangian dynamics. At this point, the first two elements of the remap strategy in section 1 have been achieved. We describe each of the steps in more detail in the following subsections.

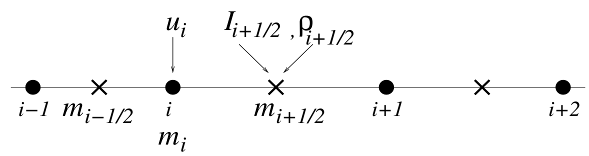


Figure 1: *Staggered grid. Nodes are marked by solid circles, and cell centers marked by* $\times$.

A one-dimensional staggered mesh data structure is shown in Fig. 1. Integer indices (i.e., subscripts) identify nodes while half-integer indices identify cell centers. All quantities at the beginning of the remap result from the Lagrangian phase, and will have no superscripts.

2.1 Density and Internal Energy

Density (ρ) and internal energy (ρI) are both naturally cell-centered variables. Each is constrained to be nonnegative on physical grounds. The methodology of [13] is directly applicable to these variables. We briefly review this methodology here, referring the reader to the original paper for additional discussion and details.

The remapper described in [13] provides conservative, positivity-preserving interpolation that is second-order accurate. The remapper is described for one- and two-dimensional meshes, but is readily extensible to three dimensions. The methodology is face-based, and so is equally applicable to structured and unstructured meshes. Examples of both are presented in [13].

The basic methodology involves two passes through the mesh. The first pass (DC) is first-order accurate and sign preserving, the analog of donor cell differencing in advection methods. In the second pass, we make a first-order estimate of the error of the first pass and then compensate it using the same DC algorithm. Thus the second step is also sign preserving. The two-pass process is patterned after a similar technique developed for advection, see [17, 9].

The two-pass structure leads to a further simplification for multidimensional applications. Effectively, we use an approximation in the DC that does not require detailed calculations of the intersections of the old and new meshes. Instead, the terms that (in the language of advection) are termed corner coupling are explicitly incorporated into the error estimates and are implemented in the second pass. This contributes to the efficiency of the algorithm, as the calculation of intersections is both expensive and delicate.

2.2 A Strategy for Cell-Centered Momentum Remapping

There have been many strategies for remapping momentum on a staggered mesh. In the original ALE code Yaqui [5], momentum was remapped on the dual mesh. There were many disadvantages to this approach, including lack of consistency, complexity, expense, etc. More recent approaches have focused on constructing a cell-centered variable representing momentum. A review of many nodal remapping schemes can be found in [2].

A simple strategy for cell-centered momentum remapping is found in the SALE code [1], where one averaged the nodal velocities to the center, multiplied by the cell mass, and remapped the resulting cell momentum. Dividing by the new cell mass leads to the new cell-centered velocity. The disadvantage to this approach appears when one uses these new centered velocities to reconstruct the nodal velocities, which is seen to be very diffusive. Effectively, too much information is lost in the averaging processes. Indeed, this process can be dissipative even when the grid is not rezoned.

An interesting approach to eliminate this diffusion is the method of moments, described in [10]. In one dimension ($1D$), one uses the nodal velocities at both edges, u_j and u_{j+1} to form an average velocity $U_{j+\frac{1}{2}} \equiv \frac{1}{2}(u_i + u_{i+1})$ and a velocity difference $\Delta U_{j+\frac{1}{2}} \equiv (u_{j+1} - u_j)$. These quantities are then multiplied by the cell mass, remapped in a conservative fashion, and then divided by the new cell mass. Finally, the new values of velocity and velocity difference are used to reconstruct the velocities at the vertices, effectively by Taylor series expansion about the cell center. Note that in $1D$, there will be two estimates of each new nodal velocity, one from each adjacent cell. These two values are averaged, which is a dissipative process. The entire process was shown to be second-order accurate (even when the underlying remapping scheme is first order), and to be consistent with the treatment of cell-centered quantities. However, the method of moments does not address the issues associated with preserving the partition of kinetic and internal energy.

To extend this method to two dimensions ($2D$), one needs to remap 4 independent quantities for each component of velocity. In [10], these were chosen to be the 4 cell-centered basis vectors (see [12]). This choice is effective, but expensive; e.g., in $2D$ one must advect a total of 8 quantities to recover 2 components of velocity. Perhaps of greater concern, there is no physical justification for conservation of any of these quantities except the averages.

Indeed, it is this last observation that has led us to the new method we describe here. Recall that the issues associated with the energy remapping (cf. section 1) arise from not being able to conserve internal energy and total energy simultaneously. We note that it is equivalent to conserve internal energy and kinetic energy separately, for then the total energy will also be conserved. So our strategy will be to choose kinetic energy as the second cell-centered variable (i.e., in $1D$) and conserve it as part of our treatment of the momentum equation.

2.3 Realizability and Implementation Issues

The discussion above is a strategy, but there are several issues of detail that must be discussed before we can write down explicit equations. First, we are not free to choose the form of the cell-centered kinetic energy, nor for that matter of the momentum. These must be consistent with the forms conserved in the Lagrangian phase of the ALE simulation. In the Lagrangian phase, the momentum equation is typically integrated on the dual mesh. The pressure gradient is approximated by surface integrals (i.e., using Gauss' theorem), which insures local conservation by detailed balance. Thus both momentum and kinetic energy are most naturally expressed as nodal quantities. This requires a definition of the nodal mass in terms of the masses of the adjacent cells. Usually a simple arithmetic average is used; in $1D$ (see Fig. 1),

$$m_j \equiv \frac{1}{2}(m_{j-\frac{1}{2}} + m_{j+\frac{1}{2}}) \tag{2.1}$$

We define the nodal momentum $M_j \equiv m_j u_j$ and the nodal kinetic energy $K_j \equiv \frac{1}{2} m_j (u_j)^2$.

Now we define the cell-centered momentum

$$M_{j+\frac{1}{2}} \equiv \frac{1}{2} m_{j+\frac{1}{2}} (u_j + u_{j+1}) = m_{j+\frac{1}{2}} U_{j+\frac{1}{2}} \tag{2.2}$$

and the cell-centered kinetic energy

$$K_{j+\frac{1}{2}} \equiv \frac{1}{4} m_{j+\frac{1}{2}} \left[(u_j)^2 + (u_{j+1})^2 \right] = m_{j+\frac{1}{2}} k_{j+\frac{1}{2}} \,, \tag{2.3}$$

where U is the specific momentum and k is the specific kinetic energy. With these definitions, the global nodal momentum and the global cell-centered momentum are identical; also the global nodal kinetic energy and the global cell-cell centered kinetic energy are identical. This can easily be seen by simply rewriting the sums over nodes as sums over cells, i.e.,

$$\begin{aligned} M_T &= \sum_{nodes} m_j u_j = \sum_{nodes} \frac{1}{2}(m_{j-\frac{1}{2}} + m_{j+\frac{1}{2}}) u_j \\ &= \sum_{cells} \frac{1}{2} m_{j+\frac{1}{2}} (u_j + u_{j+1}) = \sum_{cells} m_{j+\frac{1}{2}} U_{j+\frac{1}{2}} \,, \end{aligned} \tag{2.4}$$

and similarly for the global kinetic energy. This demonstrates the consistency of the (2.2) and (2.3) with the dynamics of the Lagrangian phase. It is clear these definitions of cell-centered momentum and kinetic energy are easily extended to $2D$ and $3D$.

A second issue concerns restrictions on the remapping schemes that can be employed. The kinetic energy is a quadratic function of the velocities, and so it will be necessary (but not sufficient) to insure that kinetic energy remains positive after remapping. As we will show, the process of resolving the cell-centered quantities to recover the nodal velocities will involve solving a quadratic equation. Thus, there will be an additional constraint to guarantee that the nodal velocities are real, as opposed to complex. Further, we will need a criterion to select the proper root.

In section 2.4, we will show that the aforementioned constraint has the form of an inequality involving the cell-centered momentum and kinetic energy, and that satisfying this inequality imposes restrictions on the algorithm for remapping these quantities.

2.4 Reconstruction and Recovery

Using the definitions (2.2) and (2.3), we can remap cell-centered momentum and kinetic energy. We will denote the remapped values $\tilde{M}$ and $\tilde{K}$, and the remapped cell mass $\tilde{m}$. The next task of the remapping phase is to recover the nodal velocities. This we accomplish by inverting (2.2) and (2.3). We have the equations

$$\tilde{u}^L_{j+\frac{1}{2}} + \tilde{u}^R_{j+\frac{1}{2}} = \frac{2\,\tilde{M}_{j+\frac{1}{2}}}{\tilde{m}_{j+\frac{1}{2}}} \tag{2.5}$$

and the cell-centered kinetic energy

$$(\tilde{u}^L_{j+\frac{1}{2}})^2 + (\tilde{u}^R_{j+\frac{1}{2}})^2 = \frac{4\,\tilde{K}_{j+\frac{1}{2}}}{\tilde{m}_{j+\frac{1}{2}}} \tag{2.6}$$

where now the right-hand-sides are known, and we seek $\tilde{u}^L_{j+\frac{1}{2}}$ and $\tilde{u}^R_{j+\frac{1}{2}}$. The superscripts L and R remind us that these are not the final nodal velocities; after these equations are inverted, we will still have two values of velocity at each node. Equations (2.5) and (2.6) guarantee the solutions $\tilde{u}^L_{j+\frac{1}{2}}$ and $\tilde{u}^R_{j+\frac{1}{2}}$ still exactly conserve momentum and kinetic energy.

The solution of these equations is easily found:

$$\tilde{u}^L_{j+\frac{1}{2}} = \frac{\tilde{M}_{j+\frac{1}{2}}}{\tilde{m}_{j+\frac{1}{2}}} \pm \sqrt{\frac{2\tilde{K}_{j+\frac{1}{2}}}{\tilde{m}_{j+\frac{1}{2}}} - \left(\frac{\tilde{M}_{j+\frac{1}{2}}}{\tilde{m}_{j+\frac{1}{2}}}\right)^2}$$

$$\tilde{u}^R_{j+\frac{1}{2}} = \frac{\tilde{M}_{j+\frac{1}{2}}}{\tilde{m}_{j+\frac{1}{2}}} \mp \sqrt{\frac{2\tilde{K}_{j+\frac{1}{2}}}{\tilde{m}_{j+\frac{1}{2}}} - \left(\frac{\tilde{M}_{j+\frac{1}{2}}}{\tilde{m}_{j+\frac{1}{2}}}\right)^2} \tag{2.7}$$

We note that, if the velocities are to be real, then we must have $\frac{2\tilde{K}_{j+\frac{1}{2}}}{\tilde{m}_{j+\frac{1}{2}}} \geq \left(\frac{\tilde{M}_{j+\frac{1}{2}}}{\tilde{m}_{j+\frac{1}{2}}}\right)^2$. This is the constraint mentioned in subsection 2.3 and to be studied in section 2.5.

The next question concerns the choice of sign in (2.7). The origin of the extra solution is readily seen from equations (2.5) and (2.6), which are symmetric in the superscripts L and R. That is, the interchange of $\tilde{u}^L_{j+\frac{1}{2}}$ and $\tilde{u}^R_{j+\frac{1}{2}}$ is also a solution. This recognition is the key to choosing the proper sign, for if $\tilde{M}_{j+\frac{1}{2}} = M_{j+\frac{1}{2}}$ and $\tilde{K}_{j+\frac{1}{2}} = K_{j+\frac{1}{2}}$, then $\tilde{u}^L_{j+\frac{1}{2}} = u_j$. The correct choice of sign depends on the quantity

$$s_{j+\frac{1}{2}} = sgn\left(u_{j+1} - u_j\right) .$$

Then the final recovered velocities are:

$$\begin{aligned}\tilde{u}^L_{j+\frac{1}{2}} &= \frac{\tilde{M}_{j+\frac{1}{2}}}{\tilde{m}_{j+\frac{1}{2}}} - s_{j+\frac{1}{2}}\sqrt{\frac{2\tilde{K}_{j+\frac{1}{2}}}{\tilde{m}_{j+\frac{1}{2}}} - \left(\frac{\tilde{M}_{j+\frac{1}{2}}}{\tilde{m}_{j+\frac{1}{2}}}\right)^2} \\ \tilde{u}^R_{j+\frac{1}{2}} &= \frac{\tilde{M}_{j+\frac{1}{2}}}{\tilde{m}_{j+\frac{1}{2}}} + s_{j+\frac{1}{2}}\sqrt{\frac{2\tilde{K}_{j+\frac{1}{2}}}{\tilde{m}_{j+\frac{1}{2}}} - \left(\frac{\tilde{M}_{j+\frac{1}{2}}}{\tilde{m}_{j+\frac{1}{2}}}\right)^2} .\end{aligned} \tag{2.8}$$

The final step is to resolve the differences between the estimates of nodal velocity from the left and from the right. In order to conserve momentum, we choose

$$\tilde{u}_j = \frac{\tilde{m}_{j+\frac{1}{2}}\tilde{u}^L_{j+\frac{1}{2}} + \tilde{m}_{j-\frac{1}{2}}\tilde{u}^R_{j-\frac{1}{2}}}{\tilde{m}_{j-\frac{1}{2}} + \tilde{m}_{j+\frac{1}{2}}} \tag{2.9}$$

We note that this averaging is always dissipative when the grid has been rezoned. We also note that the entire remapping process restores the original velocity values when there is no rezone. In order to conserve total energy, we explicitly calculate the loss of kinetic energy and distribute it to the internal energy of the neighboring cells. It is a simple calculation to show that kinetic energy is dissipated at the node:

$$\begin{aligned}\Delta K_j &= \left(\frac{\tilde{m}_{j+\frac{1}{2}}(\tilde{u}^L_{j+\frac{1}{2}})^2 + \tilde{m}_{j-\frac{1}{2}}(\tilde{u}^R_{j-\frac{1}{2}})^2}{4}\right) - \left(\frac{(\tilde{m}_{j+\frac{1}{2}} + \tilde{m}_{j-\frac{1}{2}})(\tilde{u}_j)^2}{4}\right) \\ &= -\frac{\tilde{m}_{j-\frac{1}{2}}\,\tilde{m}_{j+\frac{1}{2}}}{2(\tilde{m}_{j-\frac{1}{2}} + \tilde{m}_{j+\frac{1}{2}})}\,(\tilde{u}^R_{j-\frac{1}{2}} - \tilde{u}^L_{j+\frac{1}{2}})^2\end{aligned} \tag{2.10}$$

We add this dissipated energy to the adjacent cells in proportion to the internal energy already in the cell. If the internal energy in cell $j-\frac{1}{2}$ after remapping is $\tilde{I}_{j-\frac{1}{2}}$ and that in cell $j+\frac{1}{2}$ is $\tilde{I}_{j+\frac{1}{2}}$, then we define the fraction

$$\eta_j \equiv \frac{\tilde{I}_{j-\frac{1}{2}}}{\tilde{I}_{j-\frac{1}{2}} + \tilde{I}_{j+\frac{1}{2}} + \epsilon}$$

where ϵ is a small constant with dimensions of energy. Now we distribute ΔK_j

$$\begin{aligned} \bar{I}_{j+\frac{1}{2}} &= \tilde{I}_{j+\frac{1}{2}} - \eta_j \frac{\Delta K_j}{\tilde{m}_{j+\frac{1}{2}}} \\ \bar{I}_{j-\frac{1}{2}} &= \tilde{I}_{j-\frac{1}{2}} - (1-\eta_j) \frac{\Delta K_j}{\tilde{m}_{j-\frac{1}{2}}} \end{aligned} \tag{2.11}$$

In principle, there are two sources to the change in internal energy in each cell. These should be accumulated in separate storage, and only the final change added to cell, to avoid introducing a path dependence to the process. Another implementation detail concerns the case that the slope $s^n_{j+\frac{1}{2}} = 0$, either reflecting a local maximum or minimum, or a flat region. In this case we extend the stencil in an attempt to find a nonzero value of the slope.

2.5 Fluxes of Momentum and Kinetic Energy

The existence of real solutions to (2.8) requires that

$$\frac{2\tilde{K}}{\tilde{m}} - \left(\frac{\tilde{M}}{\tilde{m}}\right)^2 \geq 0 \tag{2.12}$$

in every cell after remapping. Our goal in this section is not to derive general criteria to ensure that a remapping algorithm will maintain this inequality, but more simply to exhibit that such algorithms exist in the context of our cell centered remapper [13]. In particular, we will exploit an idea advanced in [8] by coordinating the fluxes of momentum $\mathcal{F}^M$ and kinetic energy $\mathcal{F}^K$ with the mass flux $\mathcal{F}^m$. This approach simplifies the implementation and significantly improves the overall efficiency of the remapping. Here we will offer two examples of coordinated fluxes and explicitly demonstrate their consistency with (2.12).

To begin, let us observe that before remapping, $\frac{2K}{m} - \left(\frac{M}{m}\right)^2 \geq 0$ always. This follows directly from the definitions, and it is easy to verify from (2.2) and (2.3) that

$$\frac{2K_{j+\frac{1}{2}}}{m_{j+\frac{1}{2}}} - \left(\frac{M_{j+\frac{1}{2}}}{m_{j+\frac{1}{2}}}\right)^2 = \frac{(u_j - u_{j+1})^2}{4} \geq 0\,. \tag{2.13}$$

It is convenient to rewrite (2.13) in terms of specific kinetic energy k and specific momentum U,

$$2k_{j+\frac{1}{2}} - \left(U_{j+\frac{1}{2}}\right)^2 = \frac{(u_j - u_{j+1})^2}{4} \ \geq \ 0\,. \tag{2.14}$$

Now suppose that, as a result of remapping and repairing density, we have mass fluxes (Δ mass in $1D$) $\mathcal{F}^m_j$ at each node. We will say the mass flux is positive in $1D$ if the rezoner has displaced the point j to the left, or equivalently if the mass of the cell to the right of the point is increased. Note then that the remapped/repaired mass $\tilde{m}_{j+\frac{1}{2}} = m_{j+\frac{1}{2}} + \mathcal{F}^m_j - \mathcal{F}^m_{j+1}$. By coordinated fluxes, we mean that the fluxes of momentum and kinetic energy are proportional to the mass flux. As the first example, we choose

$$\begin{aligned}\mathcal{F}^M_j \ &= \ \mathcal{F}^m_j \left[\xi_j U_{j-\frac{1}{2}} + (1-\xi_j) U_{j+\frac{1}{2}}\right] \\ \mathcal{F}^K_j \ &= \ \mathcal{F}^m_j \left[\xi_j k_{j-\frac{1}{2}} + (1-\xi_j) k_{j+\frac{1}{2}}\right]\end{aligned} \tag{2.15}$$

where $\xi_j = 0.5[1 + sgn(\mathcal{F}^m_j)]$. This is analogous to donor cell where upwind is determined by the sign of the mass flux. Considering a cell in $1D$, there are four combinations of ξ_j and ξ_{j+1}. For example, when both mass fluxes are positive, it is straightforward to show

$$\begin{aligned}2\tilde{k}_{j+\frac{1}{2}} \ - \ \left(\tilde{U}_{j+\frac{1}{2}}\right)^2 \ &= \ \left[2k_{j+\frac{1}{2}} - (U_{j+\frac{1}{2}})^2\right]\,(1 - f_{j+\frac{1}{2}}) + \left[2k_{j-\frac{1}{2}} - (U_{j-\frac{1}{2}})^2\right]\, f_{j+\frac{1}{2}} \\ &+ \ \ f_{j+\frac{1}{2}}(1 - f_{j+\frac{1}{2}}) \left(U_{j+\frac{1}{2}} - U_{j-\frac{1}{2}})\right)^2 \ \geq \ 0\,,\end{aligned} \tag{2.16}$$

where the fraction

$$f_{j+\frac{1}{2}} \equiv \frac{\mathcal{F}^m_j}{m_{j+\frac{1}{2}} + \mathcal{F}^m_j - \mathcal{F}^m_{j+1}}$$

is assumed to lie in the interval $[0,1]$[1]. Similar results hold for the other three combinations of flux directions.

As a second example, we choose

$$\begin{aligned}\mathcal{F}^M_j \ &= \ \mathcal{F}^m_j\, u_j \\ \mathcal{F}^K_j \ &= \ \mathcal{F}^m_j\, \frac{(u_j)^2}{2}\,.\end{aligned} \tag{2.17}$$

This is more accurate than (2.15), and is, in fact, the form derived in [8]. Again it is straightforward to show

$$2\tilde{k}_{j+\frac{1}{2}} \ - \ \left(\tilde{U}_{j+\frac{1}{2}}\right)^2 = \alpha_{j+\frac{1}{2}}(1 - \alpha_{j+\frac{1}{2}})\,(u_{j+1} - u_j)^2 \ \geq \ 0\,. \tag{2.18}$$

[1]Note that this constitutes a restriction on the rezoner.

Here the fraction

$$\alpha_{j+\frac{1}{2}} \equiv \frac{m_{j+\frac{1}{2}} + \mathcal{F}_j^m}{2(m_{j+\frac{1}{2}} + \mathcal{F}_j^m - \mathcal{F}_{j+1}^m)} ,$$

where again we assume $\alpha_{j+\frac{1}{2}} \in [0,1]$. In the numerical examples of section 3, we use (2.17).

2.6 Repair

At this stage, we have remapped values of all variables — i.e., cell centered mass and internal energy, nodal velocities — in a conservative and second-order accurate manner. Further, all the local values of cell mass and internal energy are positive. However, none of these remapped fields preserve monotonicity (with respect to the fields before remapping). In order to enforce monotonicity, we further modify the fields in a process we call repair. Here we provide an overview of the repair process. The methodology of repair is described in detail in [7] for a cell-centered variable like density. The application of repair to nodal quantities, and to the internal energy, requires some extensions that we discuss here.

Conceptually, the repair process has three simple steps. First, the fields before remapping are inspected, and bounding values, maximum and minimum, are determined for the remapped fields. Second, the remapped fields are compared to their bounding values, and cells that require repair are identified. Third, one attempts to bring the outlying cells within their bounding values by borrowing or donating to neighboring cells in a conservative fashion, while ensuring that these neighboring cells continue to remain within their own bounds.

The repair process proceeds sequentially over the fields; first cell mass is repaired, then nodal velocity, and finally the internal energy. As in the recovery process described in the previous section, when velocity is repaired, momentum must be conserved and kinetic energy must be dissipated, with the dissipated energy apportioned to the internal energy fields of the adjacent cells. The fact that both the recovery [cf. equation (2.9)] and the repair processes for velocity can increase internal energy raises a question as to how to determine the local maximum bounds for I.

An important aspect of the repair process is that repair be a local process — i.e., that the exchange of conserved quantities is between neighboring cells. In this case, the repair process can be viewed as an adjustment of the remap fluxes and the relation to the underlying partial differential equations maintained. In this sense, repair would be similar to flux-corrected transport methods employed in advection schemes (see for example [16] and the references therein). However we note that FCT methodology is an *a priori* process, whereas repair is an *a posteriori* process. At present, there is no general proof that repair is always a local process; however in all of our numerical experiments, this has proven to be the case.

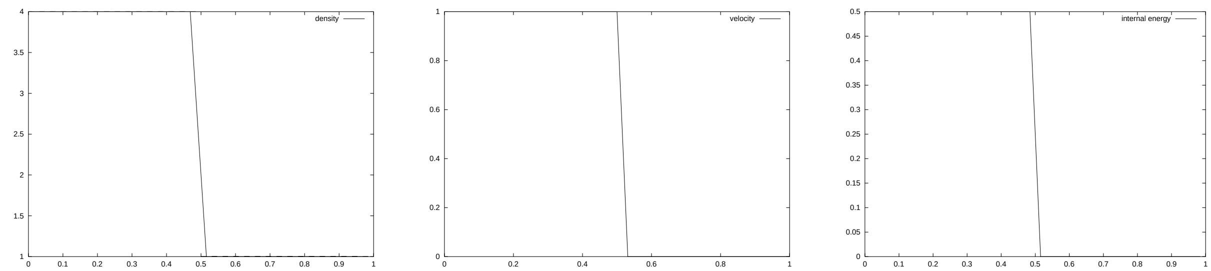
A practical issue requiring the locality of the repair is associated with the use of coordinated fluxes. The mass fluxes, for example as used in (2.17), must be evaluated after the repair of mass; hence the redistribution of mass must be consistent with some set of altered mass fluxes. It is possible to prove (by construction) that such fluxes always exist and are unique in $1D$. In $2D$, we have shown existence, but not uniqueness.

Finally, there is a question of which variables should be repaired — e.g., momentum density ρu or specific momentum u, and internal energy density ρI or specific internal energy I. Our experience so far is that using u and I produces consistently good results. However this is a question that requires further study.

3 Numerical Examples

Our remapping package is intended to be coupled with a rezoner, and to be used in the context of ALE simulations. However it is instructive to test it in a simpler environment where there are no additional coupled partial differential equations (e.g., no Lagrangian phase). As described in [13], we will test the remapping package in the context of interpolation. More precisely, we will specify the underlying functions for density, velocity and internal energy, and we will prescribe the grid motion — i.e., a sequence of displacements at each grid point. By choosing a grid motion that ultimately cycles back to the original grid, we can compare the exact (initial) values of these functions to the numerical simulations.

In all examples, we have used the nonmonotone, positivity-preserving method of [13] in combination with repair for density and internal energy. Velocity is remapped using the strategy of the previous section. In subsection 3.1, we exhibit the effects of each separate stage of remapping, comparing the profiles graphically for a single displacement of a node in an idealized shock. In subsection 3.2, we summarize the technique of cyclic remapping. In subsection 3.3, we show results and convergence studies for cyclic remapping of shocks and of smooth functions.



a) b) c)

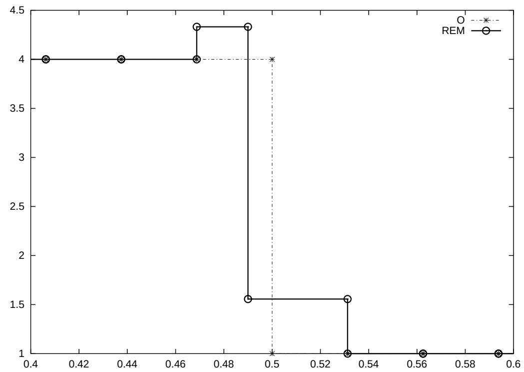
Figure 2: *Exact Shock Profiles: a) Density, b) Velocity, c) Specific Internal Energy.*

3.1 Numerical Illustration of the Main Stages of Remapping

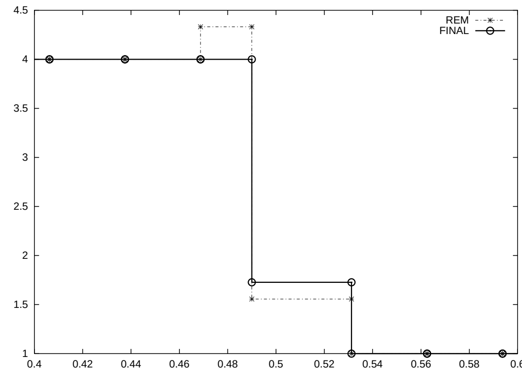
Here we illustrate the main stages of our algorithm by remapping the density ρ, velocity u, and specific internal energy I, corresponding to an infinite shock moving to the right — see Fig. 2. The problem domain is $[0, 1]$. The initial setup consists of two regions of 16 cells, each with constant values of $(\rho,\, u,\, I)$. The shock is the discontinuity between these regions, concentrated at one node, $x = 0.5$. We term the shock "infinite" because the field values on the left and right sides of the shock are related by the jump conditions for a shock in a perfect gas with infinite Mach number. We displace the shock node, that is the node between the two constant regions, to the left to $x = 0.49$. Hence relative displacement of this node to cell size is 32%. In each of the next three figures, we present a series of comparisons showing the changes in the profiles of ρ, u, and I at each stage of the remap; i.e., each intermediate stage appears twice. We plot only a fragment of the results, since only the immediate neighborhood of the shock is affected by the specified displacement.

In remapping density, our new algorithm consists only of two stages. In the first stage we use the method of [13]. The original density distribution and the result of the first stage are shown together in Fig. 3a). Because the algorithm of [13] preserves sign, but not monotonicity, the remapped density exhibits an undershoot at the foot of the shock front. In the second stage, we use the conservative repair algorithm of [7]. The remapped density after first stage, and result of repair are shown together in Fig. 3b). The repair eliminates the undershoot, which lies outside the prescribed bounds, while conserving mass.

In Fig. 4 we illustrate the main stages in remapping velocity. In first stage, we conservatively remap the cell-centered velocity and kinetic energy using (2.2) and (2.3). The original velocity and



a)



b)

Figure 3: *Stages of remap of density: a) Original and cell-centered remap; b) Cell-centered remap and result of repair.*

remapped cell-centered velocity are shown in Fig. 4a). The second stage is the reconstruction of left and right values of velocity in the cell using (2.8). The cell-centered velocity and the result of reconstruction are shown in Fig. 4b). In the third stage, we recover the velocity at the nodes using (2.9). The reconstructed velocity and the result of recovery are shown in Fig. 4c). Because the recovered velocity may not be monotone, it is repaired in the last stage. The recovered velocity and the result of repair are shown in Fig. 4d). This completes the remap of velocity.

In Fig. 5, we illustrate the stages in remapping internal energy. This process is complicated by the associated dissipation of energy, both in the recovery and in the repair stages of remapping velocity. In the first stage, we conservatively remap the internal energy. The original and the remapped internal energy are shown in Fig. 5a). After the recovery of nodal velocity from the left and right values, the loss of kinetic energy is contributed to neighboring cells. Internal energy after the first stage and second stage are shown in Fig. 5b). During repair of velocity we also may have loss of kinetic energy, which is contributed to internal energy of neighboring cells on third stage. This is shown in Fig. 5c). Finally, because in principle the first cell-centered remap can produce nonmonotone profiles of internal energy and further loss of monotonicity can occur in the succeeding stages, we repair internal energy itself. The repair is shown in Fig. 5d). This completes remap of internal energy.

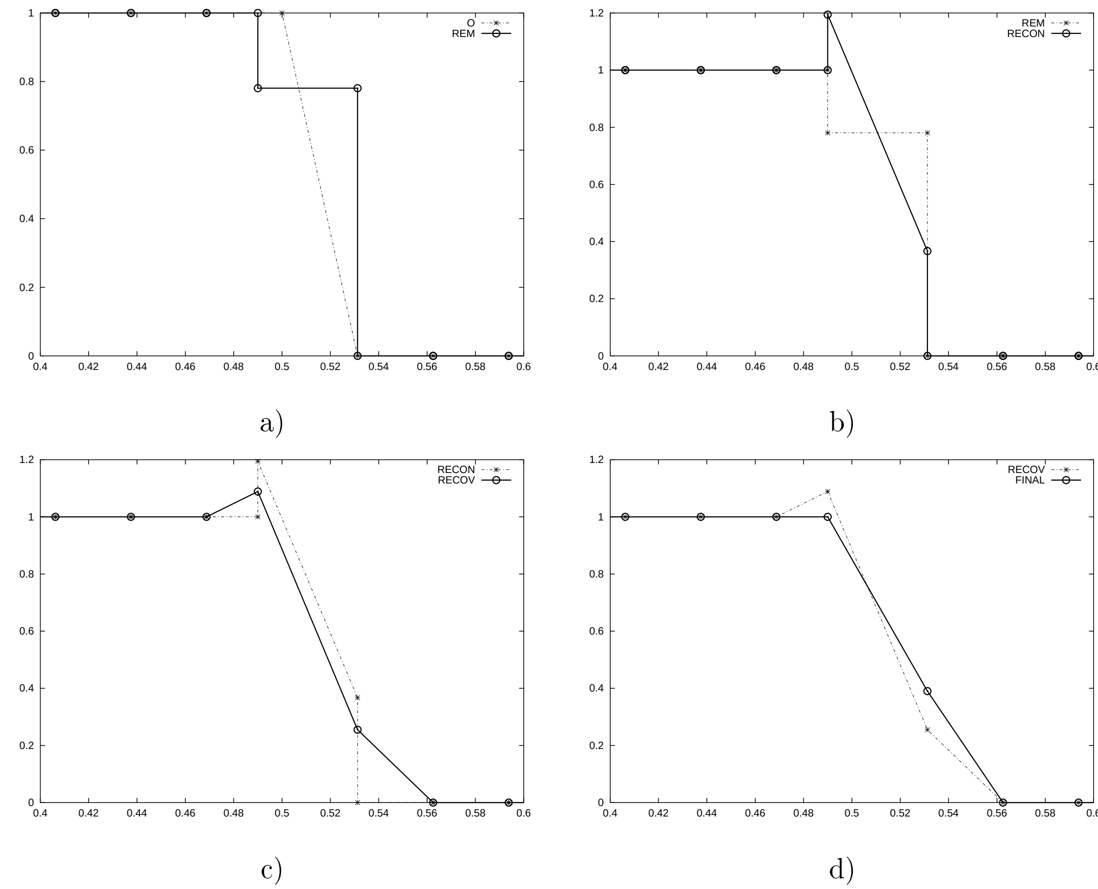
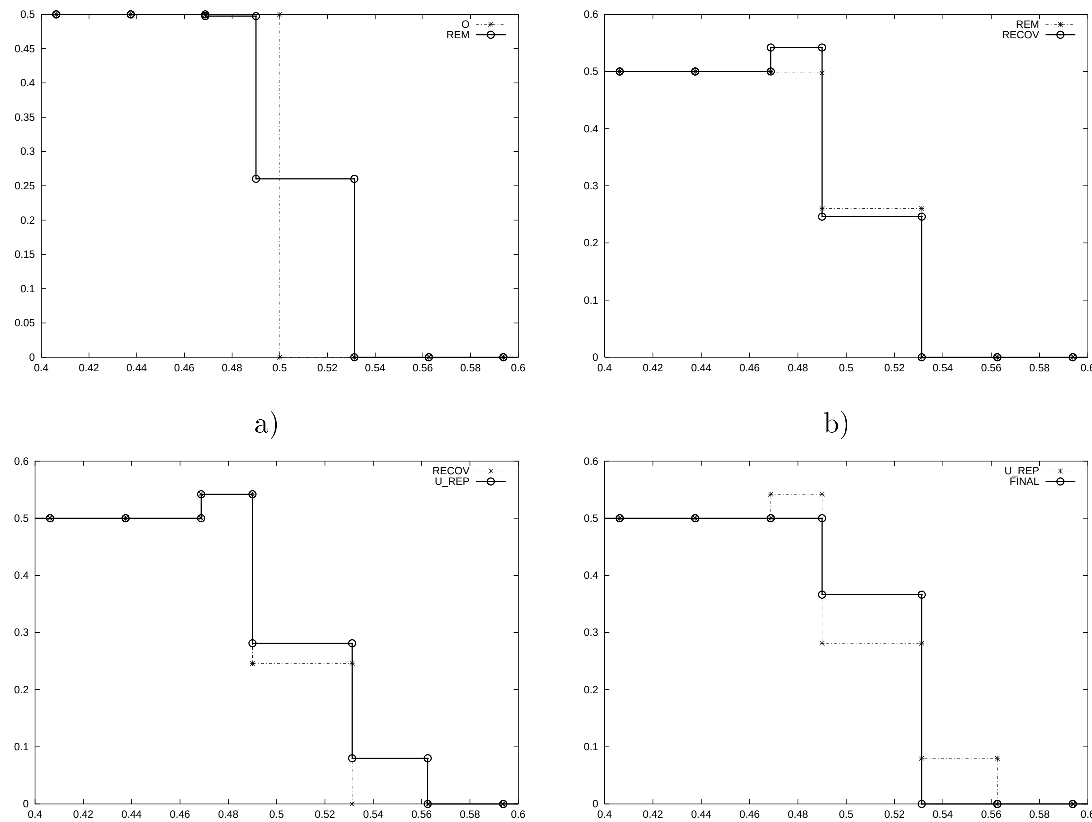
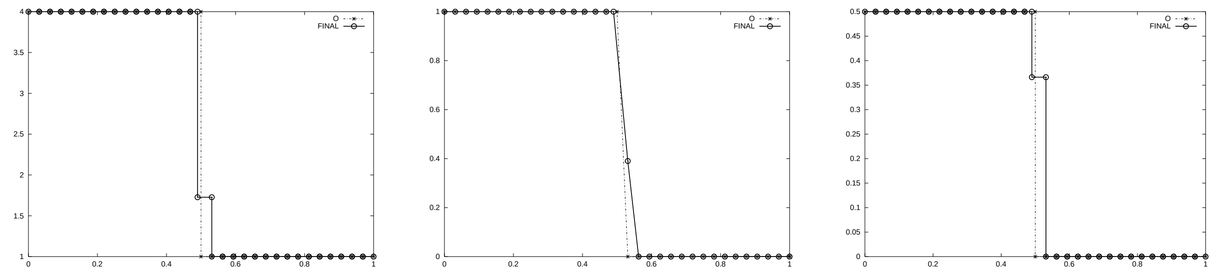


Figure 4: *Stages of remap of velocity: a) Original and cell-centered remap; b) Cell-centered remap and LR reconstruction; c) LR reconstruction and nodal recovery; d) Nodal recovery and final after repair.*



a) b) c) d)

Figure 5: *Stages of remap of specific internal energy: a) Original and cell-centered remap; b) Cell-centered remap and result of contributing of loss kinetic energy into internal energy due to recovery process; c) Internal energy after recovery of velocity and after repair of velocity; d) Internal energy after repair of velocity and after repair of the internal energy itself.*



a) b) c)

Figure 6: *Remapped Shock: a) Density, b) Velocity, c) Specific Internal Energy.*

A graphical summary of entire remap package is shown in Fig. 6. Here we directly compare the original and the final remapped profiles of density, velocity and internal energy.

3.2 Description of Cyclic Remapping

Here we describe a technique to study the cumulative effects of remapping. We generate a sequence of grids $\{x_i^k, i = 1, \ldots, n \;\; k = 0, \ldots k_{max}\}$ where the subscript i identifies a point on the grid and the superscript identifies a particular grid. It is useful to think of the superscript k as representing the discrete value of a fictitious time parameter t^k to define the grid motion. We begin with three test functions $\rho(x), u(x), I(x)$, and remap them from grid x_i^0 to grid x_i^1, and then remap the resulting functions from grid x_i^1 to grid x_i^2, etc. In particular if the sequence of grids is cyclic, returning to its initial configuration after some number of steps, then the difference between the initial values and the remapped values of any of the fields represents an estimate of the accumulated error of remapping.

In our tests, we generate a sequence of grids on the segment $[x_{min}, x_{max}]$ using the analytic function

$$
\begin{aligned}
& x(\xi, t) = x_{min} + (x_{max} - x_{min})\, \tilde{\xi}(\xi, t) \qquad (3.1)\\
& \tilde{\xi}(\xi, t) = (1 - \alpha(t))\, \xi + \alpha(t)\, \xi^2 \,, \\
& \alpha(t) = \frac{\sin(4\,\pi\, t)}{2} \,, \quad 0 \le \xi \le 1 \,; \; 0 \le t \le 1 \,.
\end{aligned}
$$

The grid $\{x_i^k\}$ is described by $x_i^k = x(\xi_i, t^k)$ where

$$
t^k = \frac{k}{k_{max}} \,, \quad k = 0, \ldots k_{max} \,, \quad \xi_i = (i-1)/(n-1) \,, \quad i = 1, \ldots n
$$

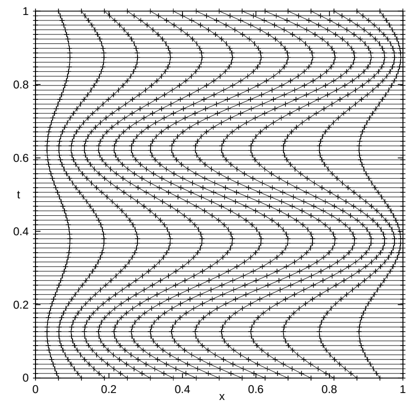


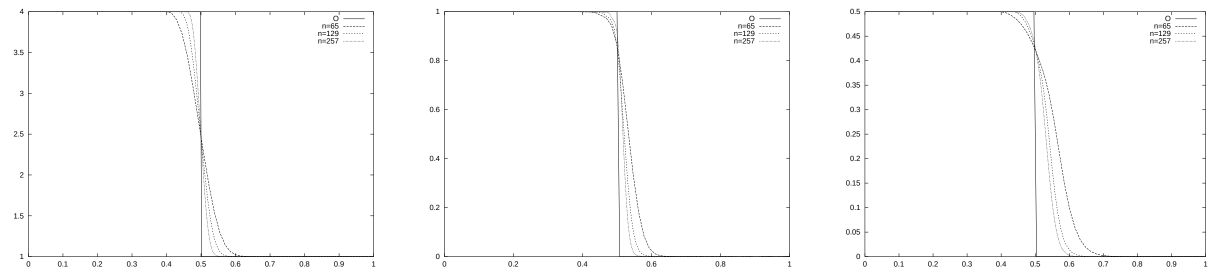
Figure 7: *Trajectories of grid points given by eq. (3.1) for* $k_{max} = 80$*,* $n = 17$*.*

One can prove that for any $0 \leq t \leq 1$, the Jacobian $\partial x/\partial\xi > 0$; i.e., each cell has positive length and so the corresponding grid is valid. These formulas produce a "smooth" displacement field. For $t^0 = 0$ and $t^k = 1$, α is zero; that is, the initial and final grids are uniform and identical. The grid evolution for $x_{min} = 0$ and $x_{max} = 1$ is shown in Fig. 7, where we show a space-time diagram. For fixed t, the diagram shows the grid corresponding to this time. For fixed i, the diagram shows the trajectory of corresponding point.

3.3 Numerical Results for Cyclic Remapping

In all tests we use the cyclic remapping described above. To study convergence properties, we do three sets of experiments in which we increase both the number of nodes and the number of "time" steps simultaneously by factors of two. We use the following pairs of resolution: ($n = 65$, $k_{max} = 320$), ($n = 127$, $k_{max} = 640$), ($n = 257$, $k_{max} = 1280$) . These pairs, coupled with the cyclic remapping, satisfy the constraints on displacement assumed after (2.18). In the first two tests, we investigate profiles associated with shocks and nonsmooth distributions. In the third test, we investigate smooth profiles.

For the first test, we remap the shock profiles shown in Fig. 2. Plots of the initial functions and corresponding remapped quantities for all three resolutions are shown in Fig. 8. The corresponding numerical data for errors and dissipation is summarized in Table 1. This table demonstrates an approximately first-order convergence rate in the L_1 norm for all quantities. Because the profiles



a) b) c)

Figure 8: *Convergence for "shock" problem: a) density, b) velocity, c) specific internal energy.*

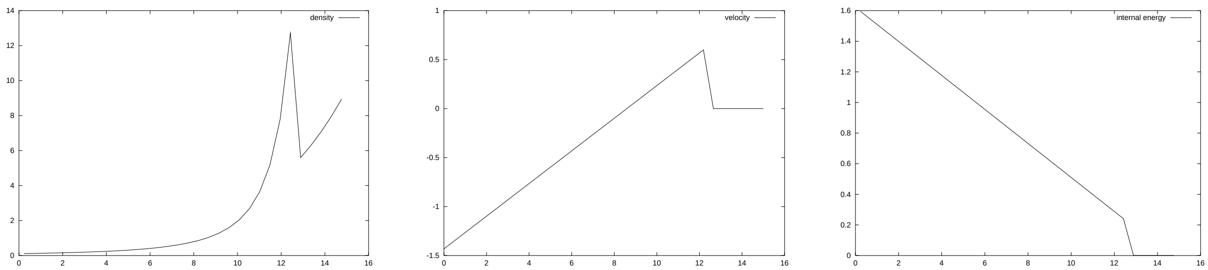
n/k_{max}	ρ	I	u	Rec-Dis.	Rep-Dis.
65/320	9.92E-2	3.35E-2	3.91E-2	0.41%	0.39%
129/640	6.28E-2	2.14E-2	2.47E-2	0.27%	0.26%
257/1280	3.97E-2	1.36E-2	1.57E-2	0.19%	0.16%

Table 1: *Convergence for the "shock" problem*

are not smooth, first-order convergence should be expected. Essentially all the error is localized to the immediate neighborhood of the shock.

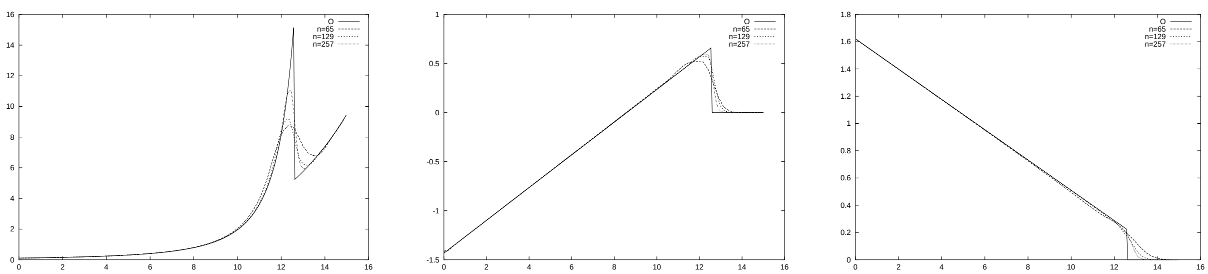
In the table, we also include the portion of the kinetic energy that is dissipated, normalized to the total energy and expressed as a percentage. Our algorithm conserves total energy to numerical roundoff. In principle, the kinetic energy should be conserved since there is no dynamics in the remap. Thus the dissipation represents an additional measure of the accuracy of the remap. Note that this dissipation even on the coarsest grid is less than one percent. We further breakdown the distributions into the separate contributions of the recovery and the repair processes of velocity. In this case, these two contributions are roughly equal. Finally, we remark that the kinetic energy dissipation is decreasing at essentially the same rate as the other errors.

The next example is the so-called exponential shock. In this problem, the computational domain is defined by $x_{min} = 0$ and $x_{max} = 15$ and the profiles are shown in Fig. 9. Plots of the exact functions and of the corresponding remapped quantities for the exponential shock test functions are shown in Fig. 10. This table demonstrates approximately first-order convergence rate in the L_1 norm for all quantities. The nonsmooth region in this problem is not as localized as in the



a) b) c)

Figure 9: *Exact exponential shock profiles: a) density, b) velocity, c) specific internal energy.*



a) b) c)

Figure 10: *Convergence for the exponential shock problem: a) density, b) velocity, c) specific internal energy.*

n/k_{max}	ρ	I	u	Rec-Dis.	Rep-Dis.
65/320	4.20	2.10E-1	2.87E-1	0.61%	2.71%
129/640	2.23	1.16E-1	1.78E-1	0.46%	1.50%
257/1280	1.53	6.57E-2	9.91E-2	0.20%	1.26%

Table 2: *Convergence for the exponential shock problem*

n/k_{max}	ρ	I	u	Rec-Dis.	Rep-Dis.
65/320	2.56E-1	3.53E-2	3.06E-2	0.02%	0.004%
129/640	6.59E-2	8.17E-3	9.57E-3	0.01%	0.0006%
257/1280	1.66E-2	1.65E-3	3.29E-3	0.005%	0.00009%

Table 3: *Convergence for the "smooth" test*

previous example, and the global error reflects this in its larger values. Likewise, the dissipation of kinetic energy is corresponding larger. In this example, the contribution from the repair process is roughly four times larger than that from the recovery process.

In the last test, we illustrate the performance of the remapping on a smooth function, We use a "piece" of the exponential shock profiles where the functions are smooth; more precisely, we cut the domain at approximately $x_{max} = 12.49$. We do not present the plots for this case because the remapped results are visually indistinguishable from the original profiles. The numerical data is presented in Table 3. The table shows that the cell-centered quantities, ρ and I exhibit second-order convergence as expected from [13]. Velocity u converges at less than second-order, with a rate ≈ 1.6. The dissipation from recovery here decreases at first order, while the dissipation from repair is much smaller. Indeed, total dissipation in this smooth case is negligible.

4 Discussion

In this paper we have described a remap strategy that can be used in Arbitrary Lagrangian-Eulerian (ALE) programs employing a straggered mesh to simulate high speed flows. The strategy is built on an existing framework of an accurate and efficient remap routine for cell-centered variables [13], and a post processing technique termed repair [7]. The use of repair allows us to separate the issues of conservation and accuracy from those of preserving monotonicity.

The novel aspect of this paper lies in our approach to remapping momentum when it is located on the nodes of the mesh. Our treatment simultaneously deals with several outstanding issues: it uses the cell-centered remapper; it conserves total energy exactly and almost conserves internal and kinetic energy separately while ensuring the positivity of the internal energy; it is consistent with the Lagrangian dynamics in the sense that it conserves the same forms of momentum and energy.

An additional innovation is the use of coordinated fluxes, a technique that significantly improves the simplicity and efficiency of the remap.

Our approach is described in sections 2.2 – 2.5, and consists of a sequence of steps in which we use the nodal velocities to define a cell-centered momentum and kinetic energy, remap these quantities, and then use the remapped values to recover new velocities at the nodes. Our description of these steps is detailed and contains several mathematical proofs that guarantee the existence and uniqueness of the nodal velocities. However we would emphasize that despite the intricacy of the construction, the resulting algorithm is both simple and efficient. Indeed, given the results of the remapped/repaired of density — i.e., the new cell masses $\tilde{m}_{j+\frac{1}{2}}$ and the mass fluxes $\mathcal{F}_j^m$ — it is possible to write down an explicit equation for the remapped velocities $\tilde{u}_j$ in terms of the original velocities u_j:

$$\begin{aligned}
\tilde{u}_j \;&=\; u_j\,\beta_j\left[1-\alpha_{j-\frac{1}{2}}+\sqrt{\alpha_{j-\frac{1}{2}}(1-\alpha_{j-\frac{1}{2}})}\right] \\
&+\; u_j\,(1-\beta_j)\left[\alpha_{j+\frac{1}{2}}+\sqrt{\alpha_{j+\frac{1}{2}}(1-\alpha_{j+\frac{1}{2}})}\right] \\
&+\; u_{j-1}\,\beta_j\left[\alpha_{j-\frac{1}{2}}-\sqrt{\alpha_{j-\frac{1}{2}}(1-\alpha_{j-\frac{1}{2}})}\right] \\
&+\; u_{j+1}\,(1-\beta_j)\left[1-\alpha_{j+\frac{1}{2}}-\sqrt{\alpha_{j+\frac{1}{2}}(1-\alpha_{j+\frac{1}{2}})}\right]
\end{aligned} \tag{4.1}$$

where

$$\beta_j \equiv \frac{\tilde{m}_{j-\frac{1}{2}}}{\tilde{m}_{j-\frac{1}{2}}+\tilde{m}_{j+\frac{1}{2}}}$$

and

$$\alpha_{j+\frac{1}{2}} \equiv \frac{\tilde{m}_{j+\frac{1}{2}}+\mathcal{F}^m_{j+1}}{2\tilde{m}_{j+\frac{1}{2}}}$$

In section 3, we offered some proof-of-principle simulations in $1D$, including both smooth and nonsmooth distributions of the three fields — density, velocity and internal energy. Using the technique of cyclic remapping, we presented and discussed the cumulative error of remapping for two problems with shocks and a third problem with smooth distributions. We remark that is difficult to characterize the general convergence properties of a remap package, which depend on both the smoothness of the functions to be remapped, and on the smoothness of the displacements. Nevertheless, even for nonsmooth functions, we found first-order convergence. In the case of smooth functions and displacements, we found second-order convergence of the density and internal energy, and nearly second-order convergence of the velocity. Further analytic and computational study of the total remap package will be required to more completely address questions of accuracy and to quantify efficiency.

An important next step is the generalization of our algorithm to multiple dimensions. In this regard, we have designed each step of the nodal remap to be extensible. In particular, we are testing a recovery process in multiple dimensions that reduces to the $1D$ algorithm where appropriate. In addition, there are issues in the repair concerning the choice of variables for which to preserve monotonicity and concerning the determination of the bounds for the internal energy.

We conclude by reemphasizing that the use of this remapping strategy places certain restrictions on the new mesh specified in the rezone phase. Similarly, the details of the error of the remapping phase will play a significant role in any rezoner based on *a posteriori* error analysis. *Caveat emptor.*

Acknowledgment

This work was performed under the auspices of the US Department of Energy at Los Alamos National Laboratory, under contract W-7405-ENG-36. The authors acknowledge the partial support of the DOE/BES Program in the Applied Mathematical Sciences and the Laboratory Directed Research and Development program (LDRD). M. Shashkov also acknowledges the partial support of DOE's Accelerated Strategic Computing Initiative (ASCI) . The authors thank R. B. Pember, P. Knupp, P. K. Smolarkiewicz and B. B. Wendroff for fruitful discussions and constructive comments.

References

[1] A.A. Amsden, H.M. Ruppel and C.W. Hirt, "SALE: A Simplified ALE Computer Program for Fluid Flow at All Speeds," Los Alamos National Laboratory Report LA-8095 (1980).

[2] D. J. Benson, "Momentum Advection on a Staggered Grid," J. Comp. Phys. **100** pp. 143-162 (1992).

[3] R.B. Demuth, L.G. Margolin, B.D. Nichols, T. F. Adams and B.W. Smith, "SHALE : a Computer Program for Solid Dynamics", Los Alamos National Laboratory Report LA-10236 (1985).

[4] C. W. Hirt, A. A. Amsden and J. Cook, "An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds", J. Comp. Phys. **14** pp. 227–253 (1974), and reprinted in **135** pp. 203–216 (1997).

[5] C.W. Hirt and A.A. Amsden, "YAQUI: An Arbitrary Lagrangian-Eulerian Computer Program for Fluid Flow at All Speeds," Los Alamos Scientific Laboratory Report LA-5100 (1973).

[6] P. Knupp, L.G. Margolin, and M. Shashkov, "Reference Jacobian Optimization-based Rezone Strategies for Arbitrary Lagrangian Eulerian Methods," J. Comp. Phys. **176** pp. 93–128 (2002).

[7] M. Kucharik, M. Shashkov and B. Wendroff, "An Efficient Linearity-and-Bound-Preserving Remapping Method", Report LA-UR-03-1766 of Los Alamos National Laboratory, Journal of Computational Physics, to appear.

[8] L.G. Margolin, M. Shashkov and P.K. Smolarkiewicz, "A Discrete Operator Calculus for Finite Difference Approximations," Comp. Meth. Appl. Sci. Eng. **187** pp. 365–383 (2000).

[9] L.G. Margolin and P.K. Smolarkiewicz, "Antidiffusive Velocities for Multipass Donor Cell Advection," SIAM J. Sci. Comp. **20** pp. 907–929 (1998).

[10] L.G. Margolin and C.W. Beason, "Remapping on the Staggered Mesh," Report UCRL-99682 of Lawrence Livermore National Laboratory (1988).

[11] L.G. Margolin, *Introduction to* "An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds", J. Comp. Phys. **135** pp. 198–202 (1997).

[12] L.G. Margolin and J. Pyun, "A Method for Treating Hourglass Patterns," Proc. 5th International Conference in Laminar and Turbulent Flow, C. Taylor, W. Habashi, M. Hafez, eds., pp. 149–160, Montreal, Canada (July, 1987).

[13] L.G. Margolin, and M. Shashkov, "Second-order sign-preserving conservative interpolation (remapping) on general grids," J. Comp. Phys. **184** pp. 266-298 (2003).

[14] M. Shashkov, **Conservative Finite-Difference Methods on General Grids**, CRC Press, Boca Raton, 1995.

[15] M. Shashkov and P. Knupp, "Optimization-Based Reference-Matrix Rezone Strategies for Arbitrary Lagrangian-Eulerian Methods on Unstructured Grids", Selcuk J. Appl. Math. 3(1) pp. 81–99 (2002).

[16] P.K. Smolarkiewicz and W.W. Grabowski, "The Multidimensional Positive Definite Advection Transport Algorithm: Nonoscillatory Option," J. Comp. Phys. **86** pp. 355–375 (1990).

[17] P. Smolarkiewicz and L. Margolin, "MPDATA: A Finite-Difference Solver for Geophysical Flows," J. Comp. Phys. **140** pp. 459–480 (1998).