LA-UR-09-02630

*Approved for public release;*
*distribution is unlimited.*

| | |
|---|---|
| *Title:* | Reduced-Dissipation Remapping of Velocity in Staggered Arbitrary Lagrangian-Eulerian Methods |
| *Author(s):* | David Bailey, Markus Berndt, Milan Kucharik, Mikhail Shashkov |
| *Intended for:* | Submitted to Journal of Computational Physics |




Form 836 (7/06)

# Reduced-Dissipation Remapping of Velocity in Staggered Arbitrary Lagrangian-Eulerian Methods

David Bailey [a] Markus Berndt [b] Milan Kucharik [c]
Mikhail Shashkov [d]

[a] *Lawrence Livermore National Laboratory, P.O. Box 808 L-016, Livermore, CA 94551, USA,* dsb@llnl.gov

[b] *Theoretical Division, T-5, Los Alamos National Laboratory MS-B284, Los Alamos, NM. 87545, USA,* berndt@lanl.gov

[c] *Theoretical Division, T-5, Los Alamos National Laboratory MS-B284, Los Alamos, NM. 87545, USA,* kucharik@lanl.gov, kucharik@karkulka.fjfi.cvut.cz

[d] *Theoretical Division, T-5, Los Alamos National Laboratory MS-B284, Los Alamos, NM. 87545, USA,* shashkov@lanl.gov

**Abstract**

Remapping is an essential part of most Arbitrary Lagrangian-Eulerian (ALE) methods. In this paper, we focus on the part of the remapping algorithm that performs the interpolation of the fluid velocity field from the Lagrangian to the rezoned computational mesh in the context of a staggered discretization. Standard remapping algorithms generate a discrepancy between the remapped kinetic energy, and the kinetic energy that is obtained from the remapped nodal velocities which conserves momentum. In most ALE codes, this discrepancy is redistributed to the internal energy of adjacent computational cells which allows for the conservation of total energy. This approach can introduce oscillations in the internal energy field, which may not be acceptable. We analyze the approach introduced in [1] which is not supposed to introduce dissipation. On a simple example, we demonstrate a situation in which this approach fails. A modification of this approach is described, which eliminates (when it is possible) or reduces the energy discrepancy.

*Key words:*
Conservative Interpolations, Staggered Discretization, Flux-Based Remap, Velocity Remap.

*Preprint submitted to Journal of Computational and Applied Mathematics16 April 2009*

## 1 Introduction

Arbitrary Lagrangian-Eulerian (ALE) methods introduced in [6] appear to be a reasonable compromise between Lagrangian and Eulerian approaches, allowing to solve a large variety of fluid problems. The standard ALE algorithm uses a Lagrangian solver to update fluid quantities and the computational mesh in the next time step, which can eventually tangle the mesh. To avoid such problems, mesh regularization (untangling or smoothing) is applied in the case of low mesh quality, followed by a remapping step that interpolates all fluid quantities from the Lagrangian to the smoothed mesh. Many authors have described ALE strategies to optimize accuracy, robustness, or computational efficiency, see for example [2,9,7,12].

It is possible to formulate the ALE scheme as a single algorithm [5] based on solving the equations in a moving coordinate frame. For fluid flows, it is common to separate the ALE scheme into three separate stages, 1) a Lagrangian stage in which the solution and computational mesh are updated; 2) a rezoning stage in which the nodes of the computational mesh are moved to a more optimal position; and 3) a remapping stage in which the Lagrangian solution is interpolated onto the rezoned mesh. Here, we focus on the last part of the ALE algorithm – remapping – in the case of a staggered discretization, where scalar quantities (density, pressure, specific internal energy) are defined inside mesh cells, and vector quantities (positions, velocities) are defined at mesh nodes [3]. A staggered discretization is used in most current ALE codes. Any proper remapping method must conserve mass, momentum, and total energy. Remapping of cell quantities in a flux form is described for example in [10,11,8], here, we focus on the remap of the nodal momenta/velocities. Generally, remapped nodal kinetic energy is not equal to nodal kinetic energy obtained from remapped velocities (usually obtained from momentum conservation equation in a flux form). This discrepancy leads to energy conservation violation and consequently to wrong shock speeds. Conservation of total energy is usually restored by redistributing the kinetic energy discrepancy to the internal energy of adjacent cells [2], which can violate smoothness of the internal energy field.

In an alternative approach introduced in [1], the remapped nodal kinetic energy is expressed in a flux form derived from the conservation of momentum and implies some constraints on momentum fluxes. Its conservation is thus enforced, and dissipation in the remapping process is eliminated. This approach requires the solution of a global system of coupled non-linear equations.

This paper has three main goals:

(1) illustrate that approach [1] does not always work;

(2) describe an alternative approach, which yields the same solution as [1] when it exists and reduces dissipation if it does not;
(3) highlight that this alternative approach can be used to get high-order fluxes in the context of FCT-like (flux corrected transport) remapping to improve accuracy but stay in bounds for velocity.

Only the 1D case and 1D examples are discussed in this paper. However, this approach is generalizable into multiple dimensions, and we implemented this in our Research Multi-Material ALE (RMALE) code.

## 2 Flux Form of Nodal Mass Remapping

In this paper, we use integer enumeration for mesh nodes, and half-integers for mesh cells, as shown in Figure 1. The nodal mass in node $i$ is remapped

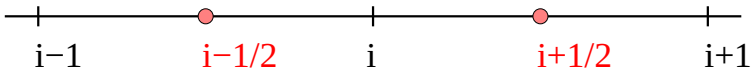



Fig. 1. Enumeration of nodes (black) and cells (red) of the 1D computational mesh. Coordinates of cell centers (red circles) computed by averaging of involved nodal coordinates.

in a standard flux form

$$\widetilde{m}_i = m_i + F^m_{i+1/2} - F^m_{i-1/2}\,. \tag{1}$$

where $F^m_{i+1/2}$ represents an oriented mass flux from node $i$ to a neighboring node $i+1$. The tilde denotes the remapped quantity (mass) in the new node.

The inter-nodal mass fluxes can be computed in several ways. The most natural way is based on intersecting the Lagrangian and rezoned nodal control volumes, and integrating the reconstructed cell density profile here to obtain the mass flux. This is simple in 1D but difficult to generalize to 2D, where it leads to intersections of similar, generally non-convex polygons. Another approach is based on the interpolation of inter-nodal mass fluxes from inter-cell mass fluxes, as described in [13]. When inter-nodal mass fluxes are computed, all nodal quantities can then be remapped in an analogous flux form, where the fluxes of a particular quantity are constructed by multiplying the mass fluxes by the value of the reconstructed quantity per unit mass. This is demonstrated in the next section for nodal momentum. For the purpose of this paper, the particular method for computation of inter-nodal mass flux $F^m$ is not important. Though, in real calculations, it can be complicated to compute the mass fluxes correctly (better than first order accurate).

## 3 Flux Form of Momentum Remapping

The remap of momentum can be performed in the flux form

$$\widetilde{\mu}_i = \widetilde{m}_i \, \widetilde{u}_i = m_i \, u_i + F^{\mu}_{i+1/2} - F^{\mu}_{i-1/2} \, , \tag{2}$$

defining the remapped nodal velocity $\widetilde{u}$. This formula guarantees global conservation of momentum.

In our approach, the momentum flux is obtained by multiplication of the mass fluxes by the flux velocities,

$$F^{\mu}_{i+1/2} = F^{m}_{i+1/2} \, u^{*}_{i+1/2} \, . \tag{3}$$

The flux velocities $u^{*}_{i+1/2}$ must be defined. The new nodal velocity is then computed as $\widetilde{u}_i = \widetilde{\mu}_i / \widetilde{m}_i$.

It is straightforward that this approach satisfies the DeBar condition [4,2], which is usually understood as a condition for self-consistency of a velocity remapping method. Suppose that we have a constant velocity field $u_n = \bar{u}$ and an arbitrary density field. After an arbitrary mesh movement, the remapping process must reproduce the constant velocity field. Any velocity reconstruction method will yield $u^{*} = \bar{u}$ for all flux velocities, so $\bar{u}$ can be factored from the whole right hand side of (2). The rest of the right hand side corresponds exactly to the new nodal mass (1), which cancels with the denominator in the expression of the new velocity formula. Thus, with the momentum flux in form (3), the remapping algorithm preserves the constant velocity field and is DeBar-consistent under the condition that the velocity reconstruction method preserves it also.

The only remaining question is how to define flux velocities $u^{*}$. Several methods exist for the low- or high-order definition of $u^{*}$. We focus here on a high-order velocity reconstruction method potentially conserving global nodal kinetic energy.

## 4 Kinetic Energy "Conserving" Remapping

In this section, we describe the high-order velocity definition algorithm that conserves global nodal kinetic energy, introduced in [1]. We will describe the derivation of the system and show a simple 1D example, for which the solution of this system does not exist. We will also suggest a modification of the system, which has the same solution as the solution of the original system if it exists.

This modification reduces the kinetic energy discrepancy, even in the case when the solution of the original system does not exist.

### *4.1 System Derivation*

As the original paper [1] was published in a not easily accessible journal, we repeat the derivation of the system here. We substitute the old nodal mass in the momentum update formula (2) by the nodal mass update formula (1), and we obtain

$$\widetilde{m}_i\,\widetilde{u}_i = \left(\widetilde{m}_i - F^m_{i+1/2} + F^m_{i-1/2}\right)\,u_i + F^m_{i+1/2}\,u^*_{i+1/2} - F^m_{i-1/2}\,u^*_{i-1/2}\,, \tag{4}$$

and after moving the first term to the left hand side, we can rewrite the expression as

$$\widetilde{m}_i\,\left(\widetilde{u}_i - u_i\right) = F^m_{i+1/2}\,\left(u^*_{i+1/2} - u_i\right) - F^m_{i-1/2}\,\left(u^*_{i-1/2} - u_i\right)\,. \tag{5}$$

Now, we multiply this equation with

$$\bar{u}_i = \frac{\widetilde{u}_i + u_i}{2} \tag{6}$$

and we obtain

$$\widetilde{m}_i\,\left(\frac{\widetilde{u}_i^2}{2} - \frac{u_i^2}{2}\right) = F^m_{i+1/2}\,\left(u^*_{i+1/2} - u_i\right)\,\bar{u}_i - F^m_{i-1/2}\,\left(u^*_{i-1/2} - u_i\right)\,\bar{u}_i\,. \tag{7}$$

To obtain the difference between new and old nodal kinetic energy on the left hand side, we add $(\widetilde{m}_i - m_i)\,u_i^2/2$ to the equation, and get

$$\begin{aligned}\widetilde{K}_i - K_i &= \frac{1}{2}\,\widetilde{m}_i\,\widetilde{u}_i^2 - \frac{1}{2}\,m_i\,u_i^2 \\ &= \frac{1}{2}\,\left(\widetilde{m}_i - m_i\right)\,u_i^2 + F^m_{i+1/2}\,\left(u^*_{i+1/2} - u_i\right)\,\bar{u}_i \\ &\qquad - F^m_{i-1/2}\,\left(u^*_{i-1/2} - u_i\right)\,\bar{u}_i\,.\end{aligned} \tag{8}$$

After substituting for $\widetilde{m}_i$ from (1), we can rewrite the expression as

$$\begin{aligned}\widetilde{K}_i - K_i = F^m_{i+1/2}\,&\left(\left(u^*_{i+1/2} - u_i\right)\,\bar{u}_i + \frac{u_i^2}{2}\right) \\ - F^m_{i-1/2}\,&\left(\left(u^*_{i-1/2} - u_i\right)\,\bar{u}_i + \frac{u_i^2}{2}\right)\,.\end{aligned} \tag{9}$$

We require the nodal kinetic energy in the flux form

$$\widetilde{K}_i = K_i + F^K_{i+1/2} - F^K_{i-1/2}\,. \tag{10}$$

To guarantee global conservation of the nodal kinetic energy, a particular flux viewed from both involved nodes must have the same value, which for example for flux $F^K_{i+1/2}$ means

$$\left(u^*_{i+1/2} - u_i\right) \bar{u}_i + \frac{u_i^2}{2} = \left(u^*_{i+1/2} - u_{i+1}\right) \bar{u}_{i+1} + \frac{u_{i+1}^2}{2}\,, \tag{11}$$

and, analogously for all other fluxes. After solving the equation for the flux velocity $u^*_{i+1/2}$, we obtain the final expression

$$u^*_{i+1/2} = \frac{u_{i+1}\,\bar{u}_{i+1} - u_i\,\bar{u}_i - \left(u_{i+1}^2 - u_i^2\right)/2}{\bar{u}_{i+1} - \bar{u}_i}\,. \tag{12}$$

Finally, we have a system of three types of equations (12), (6), and (2).This system can be solved for the set of unknowns $\{u^*,\, \bar{u},\, \widetilde{u}\}$ and its solution defines the flux velocities $u^*$.

A simple fixed point iteration process can be used as a solver. The initial guess for $u^*$ can be computed as an average of adjacent nodal velocities, for example

$$u^{*,\kappa=0}_{i+1/2} = \frac{1}{2}\,\left(u_i + u_{i+1}\right)\,, \tag{13}$$

where $\kappa$ represents the iteration index. The iterative process is then

$$\widetilde{u}_i^{\kappa} = \frac{1}{\widetilde{m}_i}\left(m_i\,u_i + F^m_{i+1/2}\,u^{*,\kappa-1}_{i+1/2} - F^m_{i-1/2}\,u^{*,\kappa-1}_{i-1/2}\right)\,, \tag{14}$$

$$\bar{u}_i^{\kappa} = \frac{\widetilde{u}_i^{\kappa} + u_i}{2}\,, \tag{15}$$

$$u^{*,\kappa}_{i+1/2} = \frac{u_{i+1}\,\bar{u}^{\kappa}_{i+1} - u_i\,\bar{u}^{\kappa}_i - \left(u_{i+1}^2 - u_i^2\right)/2}{\bar{u}_{i+1} - \bar{u}_i}\,. \tag{16}$$

In the first step of the iterative process, we use this initial guess for the update of nodal velocities using the momentum formula (14). In the second step, all $\bar{u}$ are updated as in (15). Finally, in the third step, the $u^{*,\kappa}$ are updated according to (16) (and similarly for other flux velocities), and we can start the first step of the next iteration. Due to the construction of the system, its solution must have the same kinetic energy as the old (Lagrangian) kinetic energy. This allows us to choose the stopping criteria in the form

$$\left|\frac{K^{\kappa} - K}{K}\right| < \epsilon\,, \tag{17}$$

where the tolerance for the kinetic energy discrepancy $\epsilon$ is chosen on the order

of $10^{-14} - 10^{-10}$, and the nodal kinetic energies are computed as

$$K = \sum_{\forall n} \frac{1}{2} m_n u_n^2 , \tag{18}$$

$$K^\kappa = \sum_{\forall n} \frac{1}{2} \widetilde{m}_n (\widetilde{u}_n^\kappa)^2 . \tag{19}$$

An alternative approach to solving this system is based on the construction of a vector function by moving the left hand side of (11) to the right hand side,

$$\vec{\mathcal{F}}_{i+1/2}(u^*, \bar{u}) = \left(u^*_{i+1/2} - u_{i+1}\right) \bar{u}_{i+1} + \frac{u_{i+1}^2}{2} - \left(u^*_{i+1/2} - u_i\right) \bar{u}_i + \frac{u_i^2}{2} . \tag{20}$$

After substituting for $\bar{u}$ from (6), and for $\widetilde{u}$ from (2), the function $\vec{\mathcal{F}}$ only depends on $u^*$. In (20), only one component of the vector function is shown, but similar expressions are constructed for all other fluxes. Even though this function has a local stencil, it is relatively large, especially in multiple dimensions. System (20) is basically system of coupled quadratic equations of the general form

$$\vec{\mathcal{F}}(\vec{u}^*) = \vec{0} \tag{21}$$

which can be solved using a Newton solver. We omit the explicit computation of the Jacobian of $\mathcal{F}$ as required by the classic Newton's method, and instead employ the Jacobian Free Newton Krylov (JFNK) method. In practice, we use the JFNK implementation in the NITSOL package [14].

### *4.2 Counter-Example – Non-Existent Solution*

In this section, we present a simple 1D example, for which the system (12), (6), (2) does not have a solution. The initial data are shown in Figure 2.

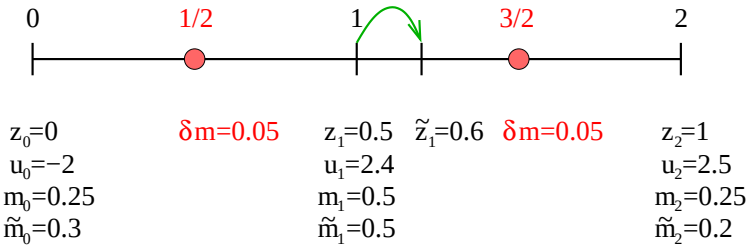



Fig. 2. Initial data of 1D example. Mesh nodes shown by black line segments, cell centers are shown by red circles. Movement of the central node is shown by green arrow, nodal positions (old position $z$ and new position $\widetilde{z}$), velocities ($u$), and masses (old mass $m$ and new mass $\widetilde{m}$) are written below the nodes, as well as inter-nodal mass fluxes ($\delta m$).

We have only two cells, positions of the surrounding nodes are $z_0 = 0$, $z_1 = 0.5$, and $z_2 = 1$. There is a constant density field $\rho = 1$ in the whole domain,

implying the values of nodal masses $m_0 = 0.25$, $m_1 = 0.5$, and $m_2 = 0.25$. Values of nodal velocities are $u_0 = -2$, $u_1 = 2.4$, and $u_2 = 2.5$. The rezoned mesh is obtained from the original mesh by moving the central node by 0.1, i.e. $\widetilde{z}_1 = 0.6$. This allows us to simply compute the inter-cell mass fluxes as $\delta m = \delta z\, \rho$, which means in our example $F^m_0 = 0$, $F^m_1 = 0.1$, and $F^m_2 = 0$. Inter-nodal mass fluxes are obtained by averaging of inter-cell fluxes, as in [13]. In our example, the fluxes are then $F^m_{1/2} = 0.05$ and $F^m_{3/2} = 0.05$. They have the same value and we will use a common symbol $F^m_{1/2} = F^m_{3/2} = \delta m$ for them. New nodal masses are obtained by the flux form remap

$$\widetilde{m}_0 = m_0 + \delta m = 0.3\,, \tag{22}$$
$$\widetilde{m}_1 = m_1 + \delta m - \delta m = m_1 = 0.5\,, \tag{23}$$
$$\widetilde{m}_2 = m_2 - \delta m = 0.2\,. \tag{24}$$

Similarly, velocity is remapped in the flux form (2),

$$\widetilde{u}_0 = \frac{1}{\widetilde{m}_0}\,\left(m_0\,u_0 + \delta m\,u^*_{1/2}\right)\,, \tag{25}$$
$$\widetilde{u}_1 = \frac{1}{\widetilde{m}_1}\,\left(m_1\,u_1 + \delta m\,u^*_{3/2} - \delta m\,u^*_{1/2}\right)\,, \tag{26}$$
$$\widetilde{u}_2 = \frac{1}{\widetilde{m}_2}\,\left(m_2\,u_2 - \delta m\,u^*_{3/2}\right)\,, \tag{27}$$

where $u^*_{1/2}$ and $u^*_{3/2}$ are unknown flux velocities which we want to find using equations

$$u^*_{1/2} = \frac{u_1\,\bar{u}_1 - u_0\,\bar{u}_0 - (u_1^2 - u_0^2)/2}{\bar{u}_1 - \bar{u}_0}\,, \tag{28}$$
$$u^*_{3/2} = \frac{u_2\,\bar{u}_2 - u_1\,\bar{u}_1 - (u_2^2 - u_1^2)/2}{\bar{u}_2 - \bar{u}_1}\,. \tag{29}$$

After multiplication by the denominators, substituting for all

$$\bar{u}_i = (u_i + \widetilde{u}_i)/2 \qquad \text{for all } i = 0..2\,, \tag{30}$$

and substituting for all new velocities from (25), (26), (27), we get the following system

$$u^*_{1/2}\left(u_1 + \frac{1}{\widetilde{m}_1}\,\left(m_1\,u_1 + \delta m\,u^*_{3/2} - \delta m\,u^*_{1/2}\right) - u_0 - \frac{1}{\widetilde{m}_0}\left(m_0\,u_0 + \delta m\,u^*_{1/2}\right)\right) = \frac{u_1}{\widetilde{m}_1}\,\left(m_1\,u_1 + \delta m\,u^*_{3/2} - \delta m\,u^*_{1/2}\right) - \frac{u_0}{\widetilde{m}_0}\,\left(m_0\,u_0 + \delta m\,u^*_{1/2}\right)\,,$$
$$u^*_{3/2}\left(u_2 + \frac{1}{\widetilde{m}_2}\,\left(m_2\,u_2 - \delta m\,u^*_{3/2}\right) - u_1 - \frac{1}{\widetilde{m}_1}\left(m_1\,u_1 + \delta m\,u^*_{3/2} - \delta m\,u^*_{1/2}\right)\right) = \frac{u_2}{\widetilde{m}_2}\,\left(m_2\,u_2 - \delta m\,u^*_{3/2}\right) - \frac{u_1}{\widetilde{m}_1}\,\left(m_1\,u_1 + \delta m\,u^*_{3/2} - \delta m\,u^*_{1/2}\right)\,.$$

We construct a vector of solutions $\vec{x} = [x_1, x_2] = [u^*_{1/2}, u^*_{3/2}]$. By subtracting the right hand side of the system, we can then rewrite the previous system in the form

$$\vec{\mathcal{F}}(\vec{x}) = \vec{0}\,, \tag{31}$$

where

$$\mathcal{F}_1(x_1, x_2) = C_1^1\, x_1^2 + C_1^2\, x_1\, x_2 + C_1^3\, x_1 + C_1^4\, x_2 + C_1^5\,, \tag{32}$$
$$\mathcal{F}_2(x_1, x_2) = C_2^1\, x_2^2 + C_2^2\, x_1\, x_2 + C_2^3\, x_1 + C_2^4\, x_2 + C_2^5\,, \tag{33}$$

and where the constants are

$$C_1^1 = -\frac{\delta m}{\widetilde{m}_1} - \frac{\delta m}{\widetilde{m}_0}\,, \tag{34}$$
$$C_1^2 = \frac{\delta m}{\widetilde{m}_1}\,, \tag{35}$$
$$C_1^3 = u_1 \left(1 + \frac{m_1}{\widetilde{m}_1} + \frac{\delta m}{\widetilde{m}_1}\right) - u_0 \left(1 + \frac{m_0}{\widetilde{m}_0} - \frac{\delta m}{\widetilde{m}_0}\right)\,, \tag{36}$$
$$C_1^4 = -\frac{\delta m}{\widetilde{m}_1}\, u_1\,, \tag{37}$$
$$C_1^5 = -\frac{m_1}{\widetilde{m}_1}\, u_1^2 + \frac{m_0}{\widetilde{m}_0}\, u_0^2\,, \tag{38}$$

and

$$C_2^1 = -\frac{\delta m}{\widetilde{m}_2} - \frac{\delta m}{\widetilde{m}_1}\,, \tag{39}$$
$$C_2^2 = \frac{\delta m}{\widetilde{m}_1}\,, \tag{40}$$
$$C_2^3 = -\frac{\delta m}{\widetilde{m}_1}\, u_1\,, \tag{41}$$
$$C_2^4 = u_2 \left(1 + \frac{m_2}{\widetilde{m}_2} + \frac{\delta m}{\widetilde{m}_2}\right) - u_1 \left(1 + \frac{m_1}{\widetilde{m}_1} - \frac{\delta m}{\widetilde{m}_1}\right)\,, \tag{42}$$
$$C_2^5 = -\frac{m_2}{\widetilde{m}_2}\, u_2^2 + \frac{m_1}{\widetilde{m}_1}\, u_1^2\,. \tag{43}$$

First, we attempted to solve the original system (28), (29) using the fixed point iteration but the iterative process did not converge. Next, we used NITSOL's JFNK [14] to solve the equivalent system (31) but it fails also, after 1000 iterations the solution jumps back and forth. We will show that the solution indeed does not exist by locating the minimum of $\|\vec{\mathcal{F}}(\vec{x})\|^2$ and showing that $\vec{\mathcal{F}} \neq \vec{0}$ there (let us note that the solution of the original system can exist when the remapping process is performed in several steps, known as subcycling).

We note that, for other examples, the solution may exist. For example, after changing the sign of the left velocity $u_0 = +2$, both mentioned approaches

(fixed point iterative process and JFNK solver) converge in several iterations to the correct solution with a zero kinetic energy discrepancy.

### *4.3 Modification of the System*

We construct a scalar function $\mathcal{G}$,

$$\mathcal{G}(\vec{u}^*) = \|\vec{\mathcal{F}}(\vec{u}^*)\|^2 \,. \tag{44}$$

Note, that both functions have the same solution $\mathcal{G}(\vec{u}^*) = 0 \Leftrightarrow \vec{\mathcal{F}}(\vec{u}^*) = \vec{0}$. While the components of the original function $\vec{\mathcal{F}}$ can change their sign, $\mathcal{G}$ is always positive. This means that $\mathcal{G}$ is equal to zero in its minimum, coinciding with the solution of $\vec{\mathcal{F}}$. Therefore, we are going to locate a minimum of $\mathcal{G}$. Both solving $\vec{\mathcal{F}} = \vec{0}$ and $\mathcal{G} = 0$ requires the inversion of the respective Jacobians, $J_F$ and $J_G$. Jacobian of G is better conditioned than $J_G$, but $J_G$ is symmetric whereas $J_F$ is not. Now we construct third function

$$\vec{\mathcal{H}}(\vec{u}^*) = \nabla\mathcal{G}(\vec{u}^*) \,, \tag{45}$$

which is equal to $\vec{0}$ in the minimum of $\mathcal{G}$. The system

$$\vec{\mathcal{H}}(\vec{u}^*) = \vec{0} \tag{46}$$

can again be solved by JFNK.

Particularly, for our 1D example, the scalar function $\mathcal{G}$ has the form

$$\mathcal{G}(x_1, x_2) = \mathcal{F}_1^2 + \mathcal{F}_2^2 \,, \tag{47}$$

and, consequently the vector function $\vec{\mathcal{H}}$ is

$$\vec{\mathcal{H}}(x_1, x_2) = \left[\frac{\partial\mathcal{G}(x_1, x_2)}{\partial x_1}, \frac{\partial\mathcal{G}(x_1, x_2)}{\partial x_2}\right] . \tag{48}$$

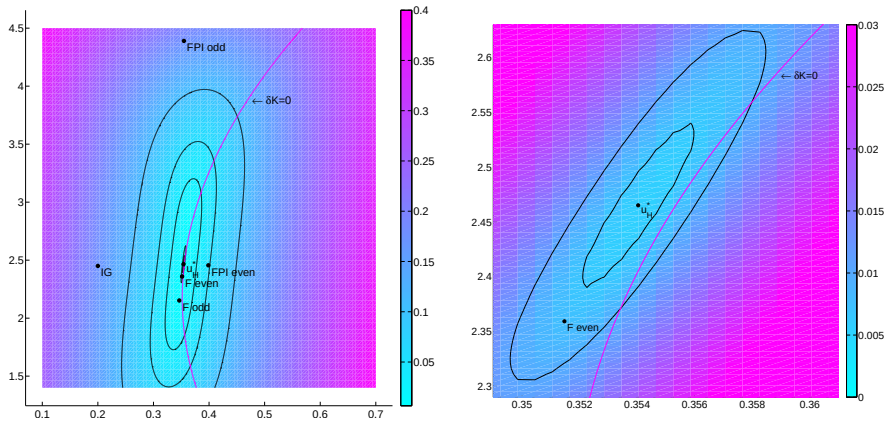
The solution of the system (46) is

$$\vec{x}^{\,\mathrm{sol}\,\mathcal{H}} = [0.354034363763449, 2.46508769340600] \,, \tag{49}$$

where $\|\vec{\mathcal{H}}(\vec{x}^{\,\mathrm{sol}\,\mathcal{H}})\| \sim 10^{-15}$. So, we have found a minimum of $G$ up to machine accuracy, and thus the point closest to the solution of system (31). In this point, the norm of $F$ is still relatively large, $\|\vec{\mathcal{F}}(\vec{x}^{\,\mathrm{sol}\,\mathcal{H}})\| \sim 5.51 \cdot 10^{-3}$. The energy discrepancy here is $\delta K = K - \widetilde{K} = -2.75 \cdot 10^{-4}$ and it is not possible to decrease it any more. For the initial guess $\vec{x}^{\,\mathrm{IG}} = [0.2, 2.45]$, the discrepancy is $\delta K = -3.28 \cdot 10^{-2}$. For comparison, we have tried to remap velocity using the donor approach (flux velocity is chosen from the nodal velocities according

to the mass flux sign, i.e. $\vec{x}^{\,\mathrm{donor}} = [2.4, 2.5]$ in our example). In this case, the energy discrepancy is $\delta K = 0.404$.

To clarify the situation, we demonstrate the situation in Figure 3. We have sampled $\|\vec{\mathcal{F}}(\vec{x})\|$ for $x_1 \in \langle 0.1, 0.7\rangle$ and $x_2 \in \langle 1.4, 4.5\rangle$. The magenta curve is




(a) (b)

Fig. 3. Colormap and several isolines of $\mathcal{G}(x_1, x_2) = \|\vec{\mathcal{F}}(x_1, x_2)\|^2$ sample over $x_1 \in \langle 0.1, 0.7\rangle$ and $x_2 \in \langle 1.4, 4.5\rangle$ (a), and zoom to the center of the sampling region. Horizontal axis represents $x_1 = u^*_{1/2}$, vertical one represents $x_2 = u^*_{3/2}$. Magenta line represent isoline of zero kinetic energy discrepancy $\delta K = 0$, the solution is expected to be located on this line. Points show initial guess (IG, average of adjacent nodal velocities), last odd and even iteration of fixed point iterative process (FPI odd and FPI even), last odd and even iteration of NITSOL's JFNK solver for $\vec{\mathcal{F}} = \vec{0}$ (F odd and F even), and solution of $\vec{\mathcal{H}} = \vec{0}$ ($u^*_{\mathcal{H}}$).

the isoline of $\widetilde{K} - K = 0$, so we expect the solution to be located on this curve. The initial guess (average of adjacent nodal velocities), and last odd and even iterations of the fixed point iterative process and the JFNK solver for $\vec{\mathcal{F}} = \vec{0}$ are shown to demonstrate divergence of the iterative process. Shown is also the point representing the JFNK solution of the modified $\vec{\mathcal{H}} = \vec{0}$ system ($\vec{x}^{\,\mathrm{sol}\,\mathcal{H}}$), located close, but not exactly on the zero discrepancy curve. Thus, we have reduced the value of the kinetic energy discrepancy, but have not eliminated it completely.

To conclude our 1D example, we have demonstrated, that the system (28), (29) has no solution in this case. Therefore, instead of looking for a solution of $\vec{F}(\vec{x}) = 0$ we find the minimum of $\|\vec{F}(\vec{x})\|^2$, which, if the solution of $\vec{F}(\vec{x}) = 0$ existed, would coincide with it. The minimum is found correctly, up to machine accuracy, the kinetic energy discrepancy is dramatically decreased (by the factor of $10^2$ when compared to the initial guess), but does not equal to zero.

We note that in 2D, the situation is similar. We can construct the $\mathcal{G}$ and $\vec{\mathcal{H}}$ functionals the analogous to 1D, but there will be a significantly larger number

of functional components and unknown flux velocities. The evaluation of the $\vec{\mathcal{H}}$ is more complex in the 2D case. The method has the same properties as in 1D – if the solution of the original system (31) exists, we find it by solving the modified system (46). If it does not exist, the solution of (46) decreases the kinetic energy discrepancy, but does not eliminate it completely.

## 5 Conclusion

In this paper, we have discussed a potentially kinetic-energy-conservative algorithm [1] for remapping nodal velocities in a staggered discretization. We have demonstrated that this approach is not bullet-proof – in some cases, the appropriate system might not have a solution. We have suggested a modification of this approach that is based on the minimization of $\|\vec{F}(\vec{x})\|$ instead of solving the original system $\vec{F}(\vec{x}) = \vec{0}$. This modification has the same solution as the original system, if it exists. If the solution of the original system does not exist, our modification decreases the kinetic energy discrepancy (dissipation) but does not generally eliminate it completely. This approach (as well as most other high-order methods) can introduce oscillations in the remapped nodal velocity field. Therefore, a combination of this approach with the low-order donor method by flux-corrected remap (FCR) is suggested.

Let us note that this (or a similar) approach is very promising as it eliminates problems with energy conservation in the remapping stage of the ALE algorithm without introducing disturbances into the internal energy field. The described process can be incorporated to a multi-dimensional, multi-material staggered remapper. Currently, it is implemented in the framework of our RMALE research code, and it will be described in a future paper.

## Acknowledgments

This work was performed under the auspices of the National Nuclear Security Administration of the US Department of Energy at Los Alamos National Laboratory under Contract DE-AC52-06NA25396. The authors acknowledge the partial support of the DOE Advance Simulation and Computing (ASC) Program and the DOE Office of Science ASCR Program, and the Laboratory Directed Research and Development program (LDRD) at the Los Alamos National Laboratory.

## References


[1] D. S. Bailey. Second-order monotonic advection in LASNEX. In *Laser Program Annual Report '84*, number UCRL-50021-84, pages 3–57–3–61, 1984.

[2] D. J. Benson. Computational methods in Lagrangian and Eulerian hydrocodes. *Computer Methods in Applied Mechanics and Engineering*, 99(2-3):235–394, 1992.

[3] E. J. Caramana, D. E. Burton, M. J. Shashkov, and P. P. Whalen. The construction of compatible hydrodynamics algorithms utilizing conservation of total energy. *Journal of Computational Physics*, 146(1):227–262, 1998.

[4] R. B. DeBar. Fundamentals of the KRAKEN code. Technical Report UCIR-760, Lawrence Livermore Laboratory, 1974.

[5] J. Donea, S. Guiliani, and J.P. Halleux. An arbitrary Lagrangian-Eulerian finite element method for transient fluid-structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 33(1-3):689–723, 1982.

[6] C. W. Hirt, A. A. Amsden, and J. L. Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 14(3):227–253, 1974.

[7] P. Kjellgren and J. Hyvarinen. An arbitrary Lagrangian-Eulerian finite element method. *Computational Mechanics*, 21(1):81–90, 1998.

[8] M. Kucharik, M. Shashkov, and B. Wendroff. An efficient linearity-and-bound-preserving remapping method. *Journal of Computational Physics*, 188(2):462–471, 2003.

[9] L. G. Margolin. Introduction to "An arbitrary Lagrangian-Eulerian computing method for all flow speeds". *Journal of Computational Physics*, 135(2):198–202, 1997.

[10] L. G. Margolin and M. Shashkov. Second-order sign-preserving conservative interpolation (remapping) on general grids. *Journal of Computational Physics*, 184(1):266–298, 2003.

[11] L.G. Margolin and M. Shashkov. Remapping, recovery and repair on staggered grid. *Computer Methods in Applied Mechanics and Engineering*, 193(39-41):4139–4155, 2004.

[12] J. S. Peery and D. E. Carroll. Multi-material ALE methods in unstructured grids. *Computer Methods in Applied Mechanics and Engineering*, 187(3-4):591–619, 2000.

[13] R. B. Pember and R. W. Anderson. A comparison of staggered-mesh Lagrange plus remap and cell-centered direct Eulerian Godunov schemes for Eulerian shock hydrodynamics. Technical report, LLNL, 2000. UCRL-JC-139820.

[14] M. Pernice and H. F. Walker. NITSOL: A Newton iterative solver for nonlinear systems. *SIAM Journal on Scientific Computing*, 19(1):302–318, 1998.