LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-JRNL-639603

Arbitrary Lagrangian Eulerian Remap Treatments Consistent with Staggered Compatible Total Energy Conserving Lagrangian Methods

J. M. Owen, M. Shashkov

June 20, 2013

Journal of Computational Physics

Disclaimer

Arbitrary Lagrangian Eulerian Remap Treatments Consistent with Staggered Compatible Total Energy Conserving Lagrangian Methods

J. Michael Owen[a], M. J. Shashkov[b]

[a]*Lawrence Livermore National Laboratory, AX Division, M/S L-38, P.O. Box 808, Livermore, CA, 94550*
[b]*Los Alamos National Laboratory, XCP-4: Methods and Algorithms, M/S F644, Los Alamos, NM, 87545*

Abstract

We describe new methods of computing post-remap nodal and subzonal masses in Arbitrary Lagrangian Eulerian (ALE) calculations employing the staggered energy conserving Lagrangian hydrodynamics method of Caramana et al. (1998). An important feature of this Lagrangian algorithm is the distribution of the masses to subzonal corners within each zone, which is then used to difference the momentum and energy equations such that both linear momentum and total energy are conserved. Such algorithms present challenges when employed as part of an ALE application, however, as these subzonal masses must be treated consistently through the remap phase. In this work we develop new ideas to compute the post-remap corner masses and associated mass fluxes between the nodal control volumes, such that the new corner masses (and therefore zonal and nodal masses) are consistently defined and conservation of mass and momentum are preserved. We demonstrate applications of these ideas on examples including pure remapping and full ALE test cases.

Keywords: Remapping, ALE methods

Email addresses: mikeowen@llnl.gov (J. Michael Owen), shashkov@lanl.gov (M. J. Shashkov)

1. Introduction

Staggered grid Lagrangian hydrodynamics has a long history of successful application to modeling gas dynamics with multiple materials, complex flows, and strong shocks. In the late 90's a new formulation of staggered grid Lagrangian gas dynamics was proposed [8] that could simultaneously satisfy the conservation of mass, momentum, and energy to machine precision while still performing well on problems where maintaining accurate adiabatic evolution is important. The authors refer to this approach as a "compatible" hydrodynamics algorithm. Compatibility as used here means that the momentum and energy equations are discretized compatibly with one and other such that total energy is conserved, or more broadly the discrete forms of the hydrodynamic equations are "compatible" with their continuum counterparts for the global invariants which are reproduced (in this case the conservation of mass, linear momentum, and energy to machine precision).

However, the compatible Lagrangian discretization comes with its own challenges. An important property of this discretization is that it breaks up the zones of the mesh into discrete subvolumes called corners which are treated as Lagrangian – i.e., there is no mass flux in or out of these corners during a Lagrangian step. The corners represent the intersection of the primary mesh (consisting of the zonal control volumes) and the dual mesh (made up of the nodal control volumes). Fig. 1 shows one example of how these different volumes are defined for a 2D mesh consisting of hexagonal zones. The zonal and nodal control volumes are defined as the union of the appropriate set of corners, i.e., the corners making up the zone on the primary mesh or the node on the dual. The zonal and nodal masses are the sum of their constituent corner properties, so that

$$m_z = \sum_{\text{c(zone)}} m_c, \tag{1}$$

$$m_n = \sum_{\text{c(node)}} m_c, \tag{2}$$

where m_z, m_n, and m_c represent the masses of the zone, node, and corner respectively. Note we have used the shorthand c(zone) to represent "corners of the zone" and c(node) for "corners of the node". Requiring the corners to remain Lagrangian means that both the nodal and zonal volumes are Lagrangian as well.

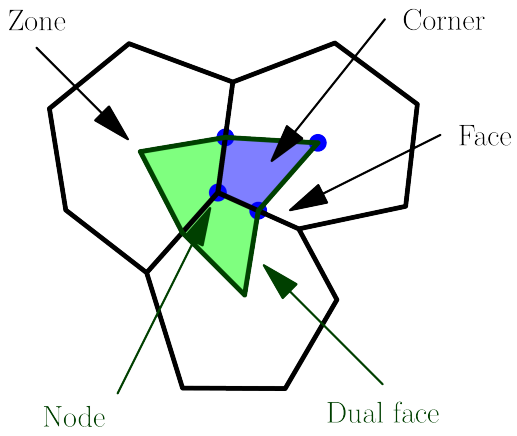


Figure 1: Cartoon view of the primary mesh components (black) and dual mesh (green). The intersection of the primary and dual zones represent the corners described in the text. We highlight a corner in light blue here, with the corner vertices marked.

While everything is nicely defined in terms of the Lagrangian corner volumes for the Lagrangian hydrodynamics algorithm, these definitions present us with certain difficulties in the context of ALE (Arbitrary Lagrangian Eulerian) calculations, wherein the mesh is allowed to move with a velocity other than the fluid velocity ([1], [2], [4], [14]). ALE is useful both because a Lagrangian calculation may result in an invalid mesh (such as when vorticity causes the mesh to cross over itself creating tangled control volumes), and because exerting some control over the mesh spacing can improve the accuracy in the solution. Standard ALE algorithms for staggered grid discretizations can be viewed as defining fluxes of conserved quantities between control volumes. Traditionally ALE based on a staggered grid Lagrangian method creates fluxes on the primary mesh for quantities defined on the zones (such as the thermal energy), and fluxes on the dual mesh for node centered quantities like the linear momentum. This is often implemented in the context of a HEMP Lagrangian step [21, 19], which is a staggered discretization on quadrilateral elements in 2D or hexahedral elements in 3D. In HEMP the nodal masses are constructed by evenly dividing the zonal mass up between its nodes: i.e., in 2D each node gets 1/4 of the mass from each of its surrounding zones. With these definitions simple averaging of the mass fluxes from the primary mesh to the dual is adequate: the post-ALE zonal and nodal

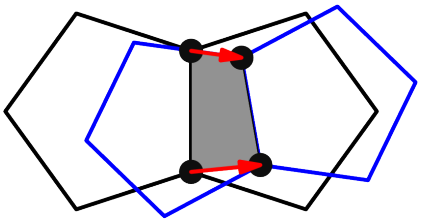


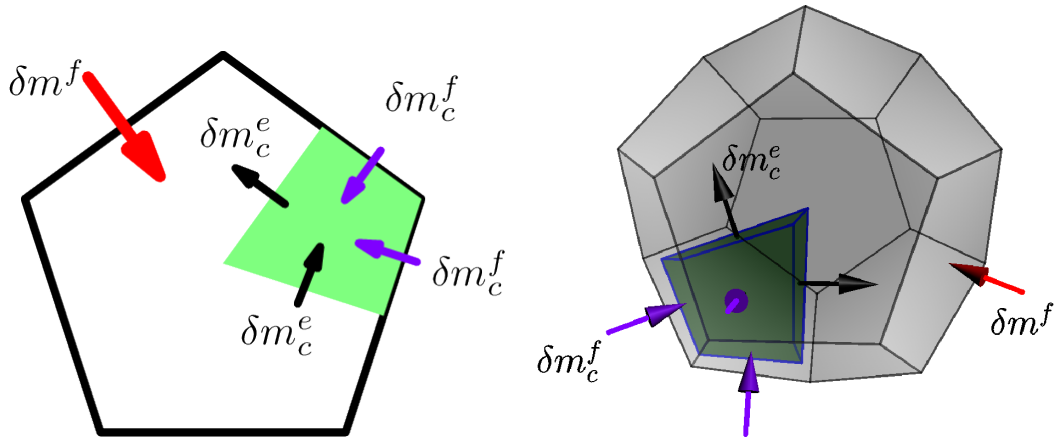
Figure 2: The swept volume flux approximation for the volume flux associated with moving a face from the original mesh configuration (black) to a new configuration (blue). The points highlight the face vertex positions in the old and new mesh positions, with the resulting swept area filled in grey.

masses will be consistent, linear momentum will be preserved, etc. However, the staggered compatible Lagrangian discretization adds the complication of defining the post-ALE corner masses, which are inconsistent with the HEMP definition during the Lagrange phase. We can view the staggered compatible discretization as requiring us to define consistent mass fluxes between corners such that the total post-ALE zonal and nodal masses are reproduced.

A standard approach to an ALE calculation is to split the time advancement algorithm into two distinct stages: a normal Lagrangian advance followed by a purely remapping stage, wherein the time is held fixed and the mesh moved over the material. This remapping stage will be our focus in this paper: we will assume that a compatible Lagrangian method is used for the first Lagrangian phase of the step, and our task is to define consistent remapped properties for the zonal, nodal, and corner control volumes during the remapping phase. See [16, 17, 18] for further discussion of these issues and the algorithms discussed in this paper.

2. Definition of the remapping fluxes

If we view the transition from the initial post-Lagrange mesh to the final relaxed mesh as a linear motion of the mesh vertices, the faces of the primary mesh sweep out a volume over the original material distribution during this transition. Fig. 2 shows a cartoon of such a procedure: the black mesh represents the initial (pre-remap) configuration, the blue the final mesh, the arrows the path of the vertices of the face, and the shaded area the area swept out by the moving face. A typical flux-based ALE implementation defines



(a) 2D (b) 3D

Figure 3: The various mass fluxes crossing contributing to a zone and one of its corners: the face flux δm^f, the corner-face fluxes δm_c^f, and the dual face flux components within the zone δm_c^e.

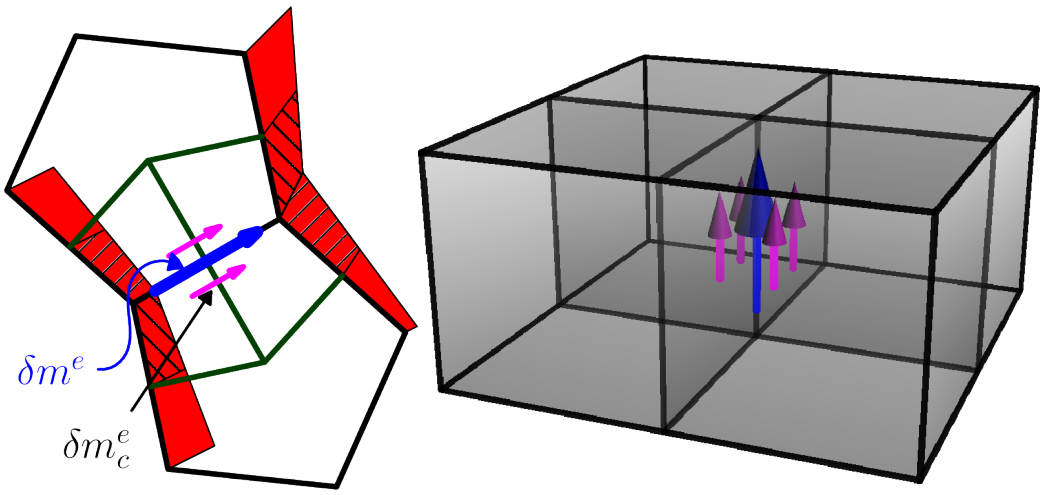
the value of some per-volume conserved quantity in this swept volume (such as the mass density ρ_f), which is then multiplied by the swept volume δV^f in order to define the mass contribution across the face from the donor to acceptor cell: $\delta m^f = \rho_f \delta V^f$. Our goal is to find the mass flux on the dual mesh and the new corner masses consistent with these primary mesh values of δm^f. There is a one to one mapping of the faces of the dual mesh and the mesh edges, so we use the edge index e to denote the dual face properties such as the dual mass flux δm^e. It is natural to extend these flux concepts to the corners as well, so the mass flux through a face can be broken into component fluxes for each corner that touches the face

$$\delta m^f = \sum_{\text{c(face)}} \delta m_c^f. \tag{3}$$

Similarly the the fluxes through the dual faces can be broken into the components between corners touching the dual face

$$\delta m^e = \sum_{\text{c(edge)}} \delta m_c^e. \tag{4}$$

Fig. 3 shows conceptual depictions of the centerings of these various fluxes.



(a) 2D (b) 3D

Figure 4: Example of partial dual face fluxes in zones contributing to the sum for the total dual face flux on the edge that shares the zones.

A standard approach to obtaining the dual mass fluxes δm^e is to average the primary face mass fluxes δm^f to the dual faces [3] – however, this algorithm needs to be generalized for polygonal/polyhedral meshes. If we treat the union of corners around an edge as that edges control volume, then there is a unique set of faces associated with each node of an edge that contribute to the mass fluxing along the edge between the nodes, as depicted in the first panel of Fig. 4 where we have colored the pertinent face swept-areas (i.e., δV^f) red. The hatched sections represent the portions of the face swept volumes that contribute for each corner for the edge (δV_c^f), obtained by sweeping the corner section of the face. We can apportion the mass flux over the face to each corner according to the volume flux as

$$\delta m_c^f = \frac{\delta V_c^f}{\delta V^f} \delta m^f , \tag{5}$$

which is equivalent to assuming the mass density of material crossing the face has a constant density ρ_f. For each zone that touches an edge there are two corners associated with that edge (one for each node), and a reasonable choice for the mass flux between these corners would be

$$\delta m_c^e = \frac{1}{2} \left(\delta m_{c\,1}^f + \delta m_{c\,2}^f \right) . \tag{6}$$

The total dual face mass flux is obtained by summing δm_c^e for the corners of the edge according to Eq. 4, yielding a definition of the dual mass flux consistent with the standard averaging used in HEMP like methods described in [3]. The magenta arrows in Fig. 4 depict the intrazonal fluxes δm_c^e, which sum to the net dual face flux δm^e in blue. Note that regardless of how they are defined, the fluxes δm_c^e represent purely intrazonal movement of the mass: they do not change the net mass in a zone, but rather redistribute mass within it.

Whatever algorithm is used to compute δm_c^e (of which Eq. 6 is simply one geometrically motivated choice), the final corner masses are related to these fluxes by conservation. Examining Fig. 3 we can relate these quantities by

$$\Delta m_c = m_c^1 - m_c^0 = \sum_{\text{f(corner)}} \delta m_c^f + \sum_{\text{e(corner)}} \psi_z^{ce} \delta m_c^e. \tag{7}$$

In this equation we use the convention that quantities superscripted with "0" represent the pre-remap state, while "1" is the post-remap state – thus the pre-remap corner mass is m_c^0 and post-remap value is m_c^1. ψ_z^{ce} is an integer sign matrix (with components $\psi_z^{ce} \in [-1, 0, 1]$) encoding the orientation of the fluxes δm_c^e. For a given corner and edge index (c and e), $\psi_z^{ce} = -1$ implies δm_c^e is oriented out of the corner, $\psi_z^{ce} = 1$ indicates δm_c^e points into the corner, and $\psi_z^{ce} = 0$ indicates the edge does not touch the corner. For face fluxes we adopt the convention that fluxes are positive for influx to the zone and negative for outflux. In §5.3 we work out examples of ψ_z^{ce} for particular zone types.

The closest analogue of the standard staggered ALE algorithm is to employ Eqs. 3–6 to define the mass fluxes, and use Eq. 7 to find the final corner masses. This approach does not guarantee monotonicity in the resulting corner and nodal masses, however, and as we will see in the examples below this leads to problems. The goal of our investigation is to define more successful algorithms for specifying the intrazonal mass fluxes δm_c^e and resulting post-remap corner masses m_c^1, but obeying the constraint of Eq. 7 to ensure conservation.

3. Goals

There are several principles we would like our dual mesh mass remapping algorithm to achieve:

Conservation: At a minimum we would like to enforce conservation and consistency of mass, implying the corner masses and fluxes are consistent with Eq. 7.

Monotonicity: This principle states that at the end of the remap step there should be no new extrema in any fields that have been remapped. Minimally we expect the nodal and zonal mass density fields should remain monotonic, though we are not as concerned about the corner mass densities themselves.

Logical consistency of the mass and volume fluxes: Physically we expect that the mass fluxes we compute on the dual mesh should relate with the volume fluxes determined by the mesh motion. As an example, if the mesh is relaxed strictly in the x direction we do not expect the mass fluxes on the dual mesh to suddenly be propagating in the y direction as well.

Smooth transition from the Lagrange state: This principle is related to the previous statement. We expect the amount of mass fluxing to be related to the magnitude of the mesh motion – if the mesh is relaxed by a tiny fraction of the local zone scale, we do not expect large displacements of material within the zone.

Accuracy: Finally we expect the remap of mass on the dual mesh to be as accurate as possible.

As we will see it is difficult to hit all of these goals simultaneously, but these principles provide useful metrics when comparing the strengths and weaknesses of each algorithm.

4. Algorithms for choosing Δm_c

4.1. Interpolated flux

We have already described one possible algorithm: interpolated flux, wherein Eqs. 3–6 define the mass fluxes and Eq. 7 determines the final corner masses. While this approach enforces conservation exactly, we will see that it leads to violations of monotonicity for remapped nodal quantities.

4.2. KULL Classic mode

The ASC code KULL [20] is our primary motivator in this work, and had an existing algorithm employed prior to this investigation. We refer to this mode as the KULL "classic" algorithm. Classic mode is a mixture of techniques which prioritizes monotonicity above all else. In this algorithm we use Eqs. 3–6 to define the mass flux between nodal control volumes, but simply set the corner mass densities in any remapped cell equal to the zonal density: $\rho_c^1 = \rho_z^1$. This choice yields the same final state in remapped zones that the traditional HEMP motivated ALE gives, and ensures monotonicity in both zonal and nodal remapping (which is the primary motivation in its choice). However, this choice violates Eq. 7: while global zonal and nodal masses will be conserved, the mass flux between nodal volumes does not reflect the final nodal masses obtained by taking $m_n^1 = \sum_{\mathrm{c(node)}} m_c^1$. This leads to loss of conservation of node centered quantities conserved by mass, most notably the linear momentum.

4.3. Flatten Total

The next algorithms we discuss work by choosing the desired corner mass change $\Delta m_c = m_c^1 - m_c^0$, and backing out mass fluxes that are consistent with these values. We defer until §5 the discussion of how to determine the required fluxes δm_c^e to in order to match the desired corner mass change.

The simplest choice for the new corner mass densities would be to set them equal to the zonal density: $\rho_c^1 = \rho_z^1$. This yields a net corner mass change of

$$\Delta m_c = m_c^1 - m_c^0 = \rho_c^1 V_c^1 - \rho_c^0 V_c^0 = \rho_z^1 V_c^1 - \rho_c^0 V_c^0. \tag{8}$$

Note this is similar to the classic algorithm described previously, except in this case we compute the mass fluxes δm_c^e such that conservation is preserved. We refer to this choice as the "flatten total" algorithm. This choice clearly achieves the monotonicity principle, but fails both the ideas of consistency with the ALE mesh motion and smooth transition from Lagrange state. Simple examples demonstrate why this is so: if we consider a zone that has built up corner mass density differences in the y direction and apply a purely x directional mesh motion to it, the requirement that the mass densities of the corners be equalized implies we will generate arbitrarily large fluxes orthogonally to the direction of the mesh motion in order to achieve this goal in a conservative manner.

4.4. Flatten Delta

Another idea we might try is to flux material out of corners proportionally to their initial masses (resulting in each corner hitting zero mass at the same time in the case of emptying the zone), while fluxing mass into each corner proportionally to their final volumes (implying that we raise each corner density by the same amount). These choices result from the philosophy that the in and out fluxes should smoothly bring the corner densities toward the same value, and can be expressed as

$$\Delta m_c = \frac{m_c^0}{m_z^0}\Delta m_z^{\text{outflux}} + \frac{V_c^1}{V_z^1}\Delta m_z^{\text{influx}}, \tag{9}$$

where the flux in and out of the zones is defined as

$$\Delta m_z^{\text{influx}} = \sum_f \max(0, \delta m^f) \geq 0, \tag{10}$$

$$\Delta m_z^{\text{outflux}} = \sum_f \min(0, \delta m^f) \leq 0. \tag{11}$$

We refer to the choice of Eq. 9 as the "flatten delta" algorithm. This idea meets the principle of smooth transition from the Lagrange state, but in general fails the idea that the mass flux should be consistent with the swept volume flux. Additionally, as we will see in the examples it does not meet the monotonicity requirement.

4.5. Variation diminishing

Motivated by our desire for monotonicity, we can employ the principle that all fluxing in, out, or within a zone should strictly move the corner densities closer to the average zonal value. In describing this option we will move through several steps defining intermediate values of the corner densities (denoted by various primes of ρ_c), until we arrive at the final value for ρ_c^1.

We begin by adjusting the corner densities to account for the zone volume change:

$$\rho_c' = \frac{V_z^0}{V_z^1}\rho_c. \tag{12}$$

We do not want to use the individual corner volume changes here since that may introduce non-monotonicities into ρ_c, but now we must rescale ρ_c' in

order to preserve the initial zone mass:

$$\rho_c'' = \frac{m_z^0}{\sum_{\mathrm{c(zone)}} V_c^1 \rho_c'} \rho_c'. \tag{13}$$

Next we drive the candidate corner densities toward the new average zonal value depending on the ratio of the mass fluxed and the initial mass in the zone. We define a fraction f_z representing the ratio of the fluxed mass to the original mass in the zone

$$f_z \equiv \min\left(1, \frac{\Delta m_z^{\mathrm{int}} + \Delta m_z^{\mathrm{influx}} - \Delta m_z^{\mathrm{outflux}}}{m_z^0}\right), \tag{14}$$

so that $f_z \in [0, 1]$, and we can now define

$$\rho_c''' = (1.0 - f_z)\rho_c'' + f_z \rho_z^1. \tag{15}$$

$\Delta m_z^{\mathrm{int}}$ represents the amount of material that is redistributed within the zone, rather than flowing in or out. We estimate this quantity as

$$\Delta m_z^{\mathrm{int}} = \min(V_z^0, \Delta V_z^{\mathrm{int}})\rho_z^0, \tag{16}$$

where $\Delta V_z^{\mathrm{int}}$ is the volume swept by the corner facets internal to the zone, i.e., the corner facets of the dual mesh faces in the zone. Note that we specify $\Delta V_z^{\mathrm{int}} \in [0, V_z^0]$, so $\Delta m_z^{\mathrm{int}} \in [0.0, m_z^0]$.

We now have to renormalize to the proper final zone mass in order to get the new corner mass density. We want to take care here such that we still only move corners toward the new zonal mass density. Begin by defining the mass discrepancy of the zonal mass and candidate corner masses,

$$\Delta m_z^{\mathrm{required}} \equiv \rho_z^1 V_z^1 - \sum_{\mathrm{c(zone)}} \rho_c''' V_c^1. \tag{17}$$

Then we define a correction weight per corner

$$w_c = \begin{cases} \max(0, (\rho_z^1 - \rho_c'''))V_c^1 & : \quad \Delta m_z^{\mathrm{required}} \geq 0, \\ \max(0, (\rho_c''' - \rho_z^1))V_c^1 & : \quad \Delta m_z^{\mathrm{required}} < 0, \end{cases} \tag{18}$$

and finally arrive at the corrected corner mass densities

$$\rho_c^1 = \rho_c''' + \frac{w_c}{\sum_c w_c} \frac{\Delta m_z^{\mathrm{required}}}{V_c^1}. \tag{19}$$

Note that the weights w_c have been constructed such that if the candidate corner masses underestimate the zonal mass, we only increase the densities of corners with $\rho_c''' < \rho_z^1$. Similarly if the candidate corner masses overestimate the final zonal mass, we decrease the densities of corners with $\rho_c''' > \rho_z^1$.

5. Computing consistent dual face mass fluxes.

Several of the algorithms in the previous section (such as flatten total, flatten delta, and variation diminishing) describe how to find the corner mass changes Δm_c, but have left the question of how to determine the dual fluxes δm_c^e consistent with these corner mass changes unresolved. In this section we describe a general algorithm for computing these intrazonal dual flux components based on a prescribed Δm_c.

5.1. Uniqueness of the dual fluxes.

While a given zone has N_e (the number of edges of the zone) values for δm_c^e, these values are not all independent. Eq. 7 imposes N_c (the number of corners of the zone) constraints that must be met in order to be consistent with the desired corner mass changes. In 2D we have the interesting case that each zone has the same number of corners as edges, $N_c = N_e$, implying we have the same number of constraints as unknowns. However, even in 2D there is not a unique solution for the fluxes δm_c^e. We can see why this is by examining the coupling of the fluxes to corners in Fig. 3. Each corner couples two values of the dual fluxes δm_c^e. For any solution of δm_c^e that meets the constraint of Eq. 7, adding a constant to each value of δm_c^e will also meet the corner constraints. Physically this constant represents a closed circulation pattern flowing around the zone, which is possible because the coupling of pairs of δm_c^e to each corner allows this arbitrary constant flux to pass through the corner. Therefore even though we have an equal number of constraints and unknown values of δm_c^e, there is in fact one more unknown represented by this arbitrary constant and the solution is not unique. In 3D there are generally more edges than corners and therefore we have more than one unconstrained unknown. In general the number of unconstrained or independent values N_I will be

$$N_I = N_e - N_c + 1. \tag{20}$$

5.2. Solving for the dual face fluxes.

There are a variety of methods we can use to solve for the dual face fluxes corresponding to a given choice for Δm_c. As mentioned in the previous section we need to impose additional constraints to obtain a solution. We have chosen to treat this as an optimization problem, where we define a set

of reference fluxes for each corner-dual face $\delta m_c^{e\,\mathrm{ref}}$ against which we will try to minimize the function

$$\chi(\delta m_c^e) = \sum_{\mathrm{e(zone)}} \left(\delta m_c^e - \delta m_c^{e\,\mathrm{ref}}\right)^2. \tag{21}$$

The choice of reference flux is arbitrary, but one physically plausible choice we can use the interpolated value in Eq. 6. This closely ties our reference dual flux to the known corner-face fluxes on the primary mesh, which seems reasonable.

We solve for δm_c^e as a constrained optimization problem: we wish to minimize $\chi(\delta m_c^e)$ (Eq. 21) subject to the constraints of Eq. 7, which suggests the technique of Lagrange multipliers. We recast the constraints as

$$B_c = \sum_{\mathrm{c(face)}} \delta m_c^f - \Delta m_c + \sum_{\mathrm{e(corner)}} \psi_z^{ce} \delta m_c^e \equiv A_c + \sum_{\mathrm{e(corner)}} \psi_z^{ce} \delta m_c^e = 0, \tag{22}$$

where we have gathered the known quantities into a single variable per corner

$$A_c \equiv \sum_{\mathrm{c(face)}} \delta m_c^f - \Delta m_c. \tag{23}$$

The Lagrangian function we are trying to minimize is

$$\begin{aligned} L(\delta m_c^e, \lambda_c) &= \chi(\delta m_c^e) + \sum_{\mathrm{c(zone)}} \lambda_c B_c, \qquad (24)\\ &= \sum_{\mathrm{e(zone)}} \left(\delta m_c^e - \delta m_c^{e\,\mathrm{ref}}\right)^2 + \sum_{\mathrm{c(zone)}} \lambda_c \left(A_c + \sum_{\mathrm{e(zone)}} \psi_z^{ce} \delta m_c^e \right). \end{aligned}$$

The gradient of the Lagrangian function is

$$\frac{\partial L}{\partial \delta m_c^e} = 2\left(\delta m_c^e - \delta m_c^{e\,\mathrm{ref}}\right) + \sum_{\mathrm{c(zone)}} \lambda_c \psi_z^{ce}, \tag{25}$$

$$\frac{\partial L}{\partial \lambda_c} = A_c + \sum_{\mathrm{e(zone)}} \psi_z^{ce} \delta m_c^e. \tag{26}$$

Setting $\partial L / \partial \delta m_c^e = 0$ we find

$$\delta m_c^e = \delta m_c^{e\,\mathrm{ref}} - \frac{1}{2} \sum_{\mathrm{c(zone)}} \lambda_c \psi_z^{ce}. \tag{27}$$

Setting $\partial L/\partial\lambda_c = 0$ and substituting in Eq. 27 we obtain an equation where the only unknowns are the Lagrange multipliers λ_c,

$$A_c + \sum_{\mathrm{e(zone)}} \psi_z^{ce} \left(\delta m_c^{e\,\mathrm{ref}} - \frac{1}{2} \sum_{\mathrm{c(zone)}} \lambda_c \psi_z^{ce} \right) = 0. \tag{28}$$

It is instructive to view Eq. 28 as a linear algebra problem, so

$$\boldsymbol{A} + \boldsymbol{\Psi} \left(\boldsymbol{\delta m}^{\mathrm{ref}} - \frac{\mathbf{1}}{\mathbf{2}} \boldsymbol{\Psi}^{\boldsymbol{T}} \boldsymbol{\lambda} \right) = \mathbf{0}, \tag{29}$$

where $\mathbf{A}$ and $\boldsymbol{\lambda}$ are N_c length vectors, $\boldsymbol{\delta m}^{\mathrm{ref}}$ is an N_e length vector, and $\boldsymbol{\Psi}$ is an $N_c \times N_e$ dimension matrix. Manipulating this equation we can reduce the problem to

$$\boldsymbol{\Psi^2 \lambda} = \boldsymbol{D}, \tag{30}$$

where

$$\boldsymbol{D} = \mathbf{2}(\boldsymbol{A} + \boldsymbol{\Psi \delta m}^{\mathrm{ref}}), \tag{31}$$

$$\boldsymbol{\Psi^2} = \boldsymbol{\Psi \Psi^T}. \tag{32}$$

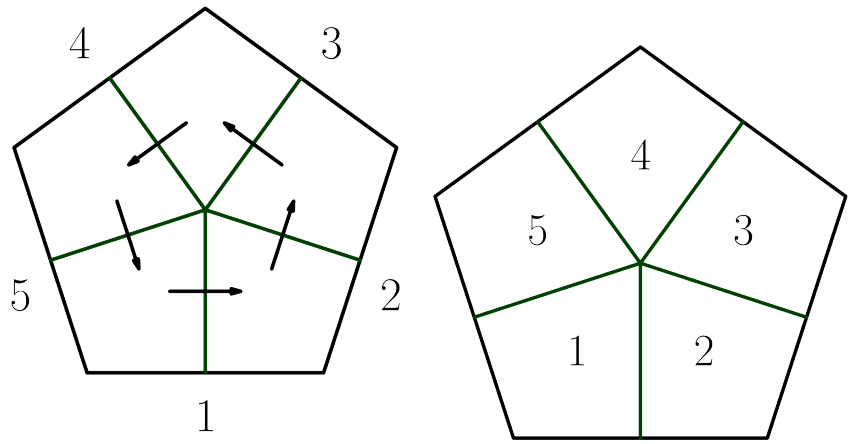
There is one wrinkle in solving Eq. 30 for λ_c: $\boldsymbol{\Psi^2}$ is singular. This is not surprising as we already know that the solution for δm_c^e is not unique. In fact, based on the discussion in §5.1 we know there will be one zero in the singular values of $\boldsymbol{\Psi^2}$. We apply singular value decomposition (SVD) to find the best-fit solution to Eq. 30. $\boldsymbol{\Psi^2}$ is decomposed into three components

$$\boldsymbol{\Psi^2} = \boldsymbol{U w V^T}, \tag{33}$$

where $\mathbf{w}$ is a diagonal matrix with the singular values of $\boldsymbol{\Psi^2}$ along the diagonal. We can then find the pseudoinverse of $\boldsymbol{\Psi^2}$ as

$$(\boldsymbol{\Psi^2})^+ = \boldsymbol{V w' U^T} \tag{34}$$

where $\mathbf{w}'$ is a diagonal matrix resulting by replacing the non-zero values of $\mathbf{w}$ with their inverse. The fact that the elements of $\boldsymbol{\Psi^2}$ are all small value integers makes this problem relatively easy: the non-singular values of $\mathbf{w}$ are also small value integers, easily distinguished from the zero of the singular element. We therefore do not have the usual difficulties with sensitivity to floating point tolerances in picking out the zero values of $\mathbf{w}$: we know how many zeros there should be and the non-zero values are integers.



(a) Edge ordering (b) Corner ordering

Figure 5: Example ordering for the edges and corners of a pentagon. The black arrows in panel (a) show the orientation of the intrazonal fluxes δm_c^e used in the discussion.

Once we have obtained the pseudoinverse $(\boldsymbol{\Psi^2})^+$ it is a simple matter to compute the values of λ_c as

$$\boldsymbol{\lambda} = (\boldsymbol{\Psi^2})^+ \boldsymbol{D}, \tag{35}$$

and we can use Eq. 27 to compute the desired constrained best-fit values for δm_c^e.

In Cartesian geometries an additional nicety of this approach is that ψ_z^{ce} (and therefore $(\boldsymbol{\Psi^2})^+$) are entirely defined by the connectivity of the edges and corners of the zone. This allows us to compute $(\boldsymbol{\Psi^2})^+$ once at problem setup and employ this value directly with Eq. 35 and Eq. 27 during the ALE step to immediately compute the values of δm_c^e – no iterative solutions are required during run time of the ALE simulation.

5.3. A few examples of computing $(\boldsymbol{\Psi^2})^+$.

To provide a few concrete examples, consider first the pentagonal zone shown in Fig. 5. The ψ_z^{ce} matrix we would derive for this zone based on the

numbering and orientation of the fluxes shown is

$$\psi^{ce}_{\text{pentagon}} = \begin{pmatrix} -1 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix}.$$

Note this matrix has some expected properties. Since in 2D every corner couples two edges, every row has two non-zero entries. Similarly since every edge couples two corners, every column has two non-zero values. Also, since every δm^e_c flux represents flux from one corner to another, conservation implies the sum of each column should equal zero. The corresponding $\mathbf{\Psi^2}$ for the pentagon is

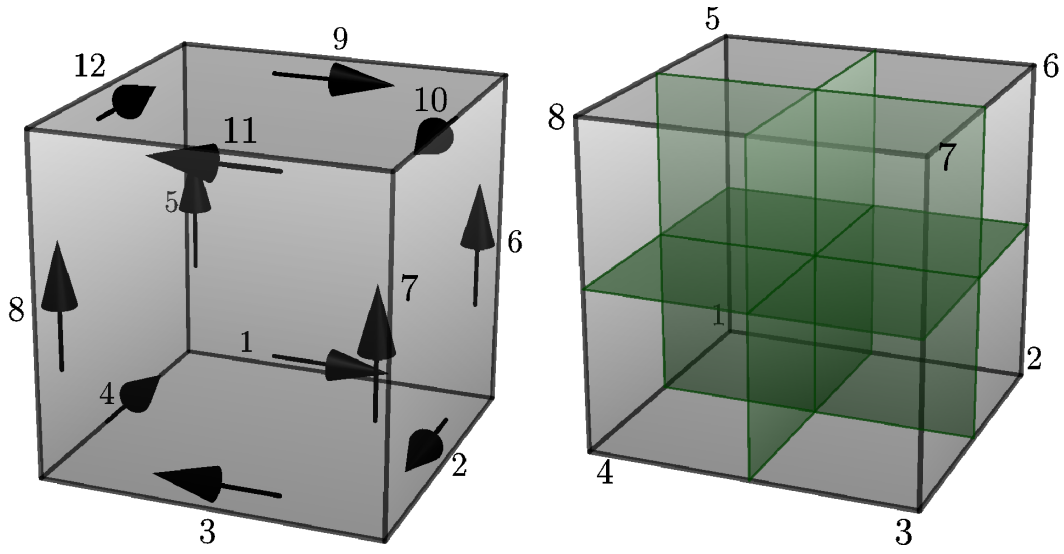
$$\mathbf{\Psi^2}_{\text{pentagon}} = \mathbf{\Psi}\mathbf{\Psi}^T = \begin{pmatrix} 2 & -1 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{pmatrix}.$$

Applying SVD to decompose this matrix yields the pseudoinverse

$$(\mathbf{\Psi^2})^+_{\text{pentagon}} = \frac{1}{5}\begin{pmatrix} 2 & 0 & -1 & -1 & 0 \\ 0 & 2 & 0 & -1 & -1 \\ -1 & 0 & 2 & 0 & -1 \\ -1 & -1 & 0 & 2 & 0 \\ 0 & -1 & -1 & 0 & 2 \end{pmatrix}.$$

Next consider the hexahedral zone in Fig. 6. The ψ^{ce}_z matrix we construct given these orderings is

$$\psi^{ce}_{z\ \text{hexahedron}} = \begin{pmatrix} -1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -1 \end{pmatrix}.$$



(a) Edge ordering (b) Corner ordering

Figure 6: Sample ordering for the edges and corners of a hexahedron. The black arrows in panel (a) show the orientation of the intrazonal fluxes δm_c^e used in the discussion. We have removed the internal corner faces in panel (a) for clarity.

In this case every corner of the hex couples three edges, so every row of ψ_z^{ce} has three non-zero values. Once again every edge couples two corners, so every column has two non-zero entries. Conservation of flux enforces the sum of each column is zero. The value of $\mathbf{\Psi^2}$ we get for this hexahedron is

$$\mathbf{\Psi^2}_{\text{hexahedron}} = \begin{pmatrix} 3 & -1 & 0 & -1 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 3 & -1 & 0 & 0 & -1 & 0 \\ -1 & 0 & -1 & 3 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 3 & -1 & 0 & -1 \\ 0 & -1 & 0 & 0 & -1 & 3 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & -1 & -1 & 0 & -1 & 3 \end{pmatrix}.$$

The pseudoinverse of $\mathbf{\Psi^2}_{\text{hexahedron}}$ is

$$(\mathbf{\Psi^2})^+_{\text{hexahedron}} = \frac{1}{96} \begin{pmatrix} 29 & 1 & -7 & 1 & 1 & -7 & -11 & -7 \\ 1 & 29 & 1 & -7 & -7 & 1 & -7 & -11 \\ -7 & 1 & 29 & 1 & -11 & -7 & 1 & -7 \\ 1 & -7 & 1 & 29 & -7 & -11 & -7 & 1 \\ 1 & -7 & -11 & -7 & 29 & 1 & -7 & 1 \\ -7 & 1 & -7 & -11 & 1 & 29 & 1 & -7 \\ -11 & -7 & 1 & -7 & -7 & 1 & 29 & 1 \\ -7 & -11 & -7 & 1 & 1 & -7 & 1 & 29 \end{pmatrix}.$$

5.4. Cylindrical geometry considerations.

In cylindrical (or rz) geometry it is not unusual to have to modify hydrodynamic methods to maintain symmetry. Most of our discussion is fine for rz case (such as the target corner mass definitions), but we find it necessary to adjust the derivation of dual face fluxes using an area weighted concept. This takes the form of solving for the fluxes per r: i.e., we work in quantities such as $\delta\tilde{m}_c^e \equiv \delta m_c^e / r_c^e$, $\delta\tilde{m}_c^{e\text{ref}} \equiv \delta m_c^{e\text{ref}} / r_c^e$, etc., where r_c^e is the radius of the centroid of the corresponding swept advection control volume. Repeating the analysis of §5.2 using this modification results in redefining Eqs. 31–32 to be

$$\boldsymbol{D} = \mathbf{2}(\boldsymbol{A} + \boldsymbol{\Psi}\boldsymbol{\delta}\tilde{\boldsymbol{m}}^{\text{ref}}), \tag{36}$$

$$\mathbf{\Psi^2} = \boldsymbol{\Psi}\boldsymbol{R}\boldsymbol{R}\boldsymbol{\Psi}^{\boldsymbol{T}}, \tag{37}$$

where $R_c^e \equiv \delta_{ee} r_c^e$ is a diagonal matrix of the r_c^e radii.

This modification is simple enough, but has an implementation consequence: the $\mathbf{\Psi^2}$ and $(\mathbf{\Psi^2})^+$ equations are no longer constant for a given zone since the radii r_c^e change each time the mesh is moved. So in the rz case (contrary to Cartesian examples discussed earlier) we cannot simply pre-compute the $(\mathbf{\Psi^2})^+$ matrix per zone at problem startup, but must instead re-evaluate it each ALE step. Fortunately these matrices are small, and we have found the performance impact of this extra solve per zone to be negligible in practice. For a more in depth description of our method in rz we refer the reader to [10].

6. Examples

We now consider a few examples comparing these different algorithms employing the KULL code [20].

6.1. 2D xy cyclic remapping of a cylindrical shell.

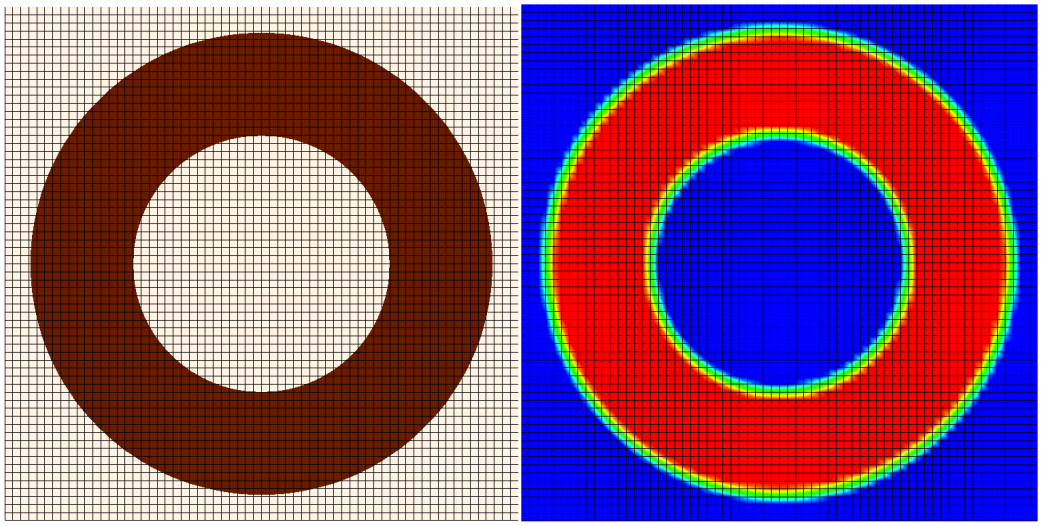
This is a two material remapping test (i.e., no hydrodynamics), based on similar tests in [12]. This example is performed in 2D assuming Cartesian (xy) geometry. We shape a cylindrical shell of material into a background ambient material with the properties $r_{\text{shell}} \in [0.25, 0.45]$, $\rho_{\text{shell}} = 1$, $v_{\text{shell}} = 1.0\ \hat{r}$, $\rho_{\text{ambiant}} = 0.1$, $v_{\text{ambiant}} = 0$, as shown in Fig. 7.

A prescribed cyclic mesh motion (Eq. 38) is employed such that the mesh goes through a periodic distortion, returning to its initial configuration every $k_{\max}$ iterations.

$$
\begin{aligned}
x_k(x_0, y_o, \tau) &= x_0 + F(\tau)\sin(2\pi\tau x_0)\sin(2\pi\tau y_0), \qquad (38)\\
y_k(x_0, y_o, \tau) &= y_0 + F(\tau)\sin(2\pi\tau x_0)\sin(2\pi\tau y_0),\\
F(\tau) &= \begin{cases} 0.2\ \tau & : \quad \tau \in [0, 0.5], \\ 0.2\ (1-\tau) & : \quad \tau \in (0.5, 1], \end{cases}\\
\tau &= \frac{k}{k_{\max}}, \qquad (39)
\end{aligned}
$$

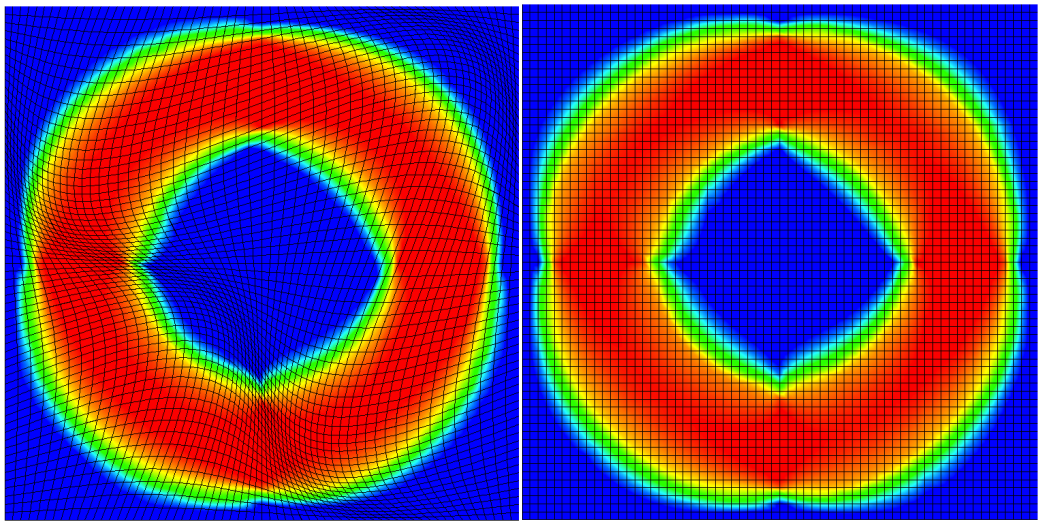
where k is the remapping step, (x_k, y_k) is the position of each mesh node on step k in terms of its initial position (x_0, y_0), and τ is the pseudo-time periodicity of the distortion.

Fig. 8 shows the magnitude of the velocity (a) at the point of maximum mesh distortion, and (b) after the mesh has returned to its original configuration. There is some diffusion evident, but the magnitude of the velocity



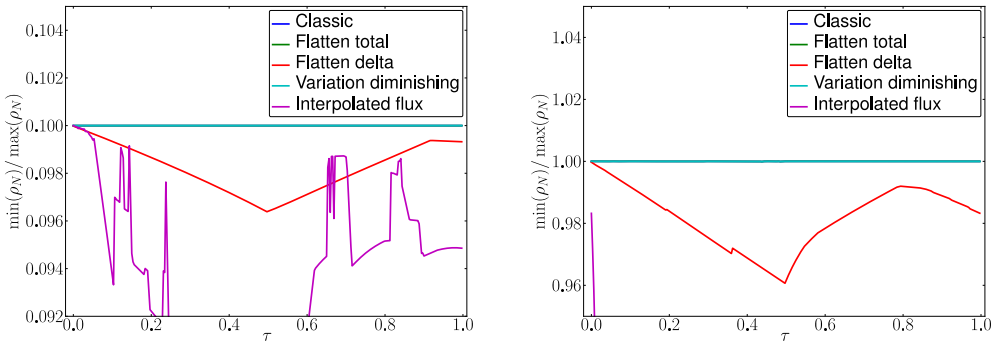
(a) Materials (b) Velocity magnitude

Figure 7: Initial configuration for the cyclic remapping test. The initial velocity is a unit vector radially outward in the cylindrical shell.



(a) $\tau = 1/2$ (b) $\tau = 1$

Figure 8: Velocity magnitude at maximum mesh distortion ($\tau = 1/2$) and after one full iteration through the mesh distortion cycle ($\tau = 1$) for the cyclic remapping test.



(a) All materials (b) Shell material only

Figure 9: Evolution of the ratio of the minimum to maximum nodal density in the cyclic remapping test.

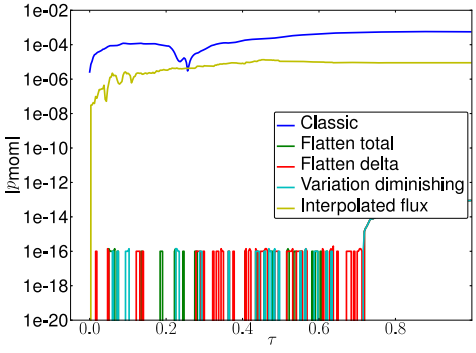
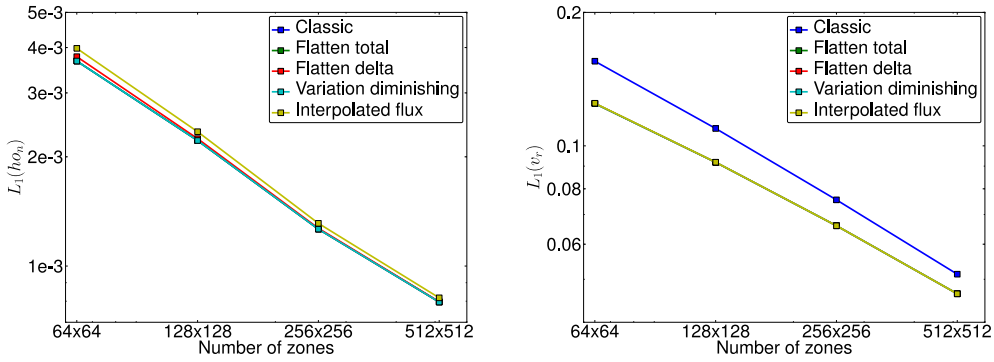


Figure 10: Evolution of the linear momentum magnitude in the cyclic remapping test.



(a) Nodal density

(b) Radial velocity

Figure 11: L_1 norms of the error in the cyclic remapping test.

remains monotonically bounded in all cases. However, the classic, flatten delta, and interpolated flux algorithms all show non-monotonic behavior in the nodal mass density in Fig. 9. In this figure we plot the ratio of the minimum to maximum nodal density as a function of time: (a) shows the total nodal densities summed over all materials, while (b) shows the case for just the shell material. Since the background material starts at 1/10 the density of the shell, we would expect Fig. 9a to stay at a constant ratio of 0.1, while the pure material result in panel (b) should remain at a constant value of unity. Only the classic, flatten total, and variation diminishing algorithms maintain monotonicity in the nodal densities.

Fig. 10 shows the time evolution of the linear momentum magnitude. Since this problem is established with pure radial outflow in the shell, by symmetry we expect the total linear momentum to be zero. As expected the newer algorithms (flatten total, flatten delta, and variation diminishing) maintain momentum conservation to numerical round-off. The interpolated flux algorithm should as well, but its non-monotonicities in the corner densities become severe enough that it loses some accuracy. As discussed previously, the classic method resulting from simply reusing a standard swept-volume nodal advection algorithm loses linear momentum conservation to the degree the corner mass sums for the nodal masses disagree with the swept volume prediction. We can see this error here, as the classic algorithm results in by far the largest errors in the total linear momentum.

We consider the convergence properties of these different algorithms in

Fig. 11. We employ the general definition for the L_p norm comparing two sequences a_i and b_i,

$$L_p \equiv \left(\frac{\sum_i (a_i - b_i)^p}{N} \right)^{1/p}. \tag{40}$$

We run each case at four different resolutions (64×64, 128×128, 256×256, & 512×512), in each case doubling the linear number of zones. Because the problem has discontinuities we expect first-order convergence. We fit the errors assuming $L_1 \propto N_z^m$ (where N_z is the number of zones and m the order of convergence), so first-order convergence corresponds to $m = -1$ and $m \geq 0$ implies non-convergence. We find that the algorithms converge similarly, with the densities converging at about $m = -0.74$ order, while the velocities are a bit slower at order $m \sim -0.5$. We note that the non-conservative classic algorithm shows the largest errors in the velocity, while all the newer consistent algorithms demonstrate similar accuracy in this case.

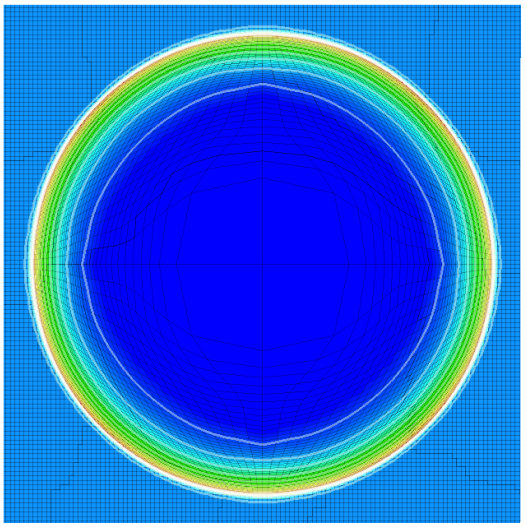
In this test we find that the variation diminishing algorithm is the most successful of our new options: it maintains conservation and monotonicity, while keeping the mass fluxes tied to the swept-volume fluxes through the optimization process. For this reason in the succeeding test cases we down select to comparing variation diminishing vs. the standard nodal swept volume classic algorithm.

6.2. Sedov-Taylor blastwave

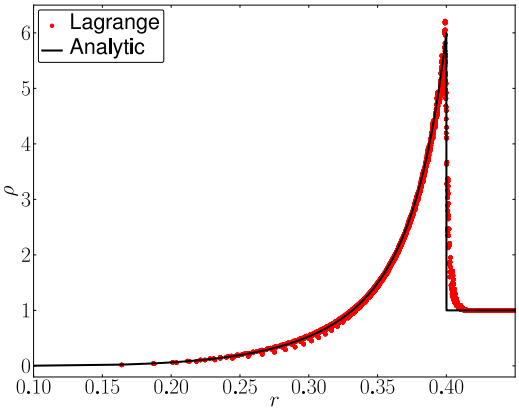
We now turn our attention to examples employing the full ALE algorithm: Lagrange + remap. We examine the Sedov-Taylor blast wave ([22], [23], [24]), which follows the self-similar expansion of a blast wave following the introduction of a high energy point source into an initially pressureless gas. Self-similar analytic solutions are available for 1D planar, 2D cylindrical, and 3D spherical geometries, allowing us to measure the simulation error against these solutions. In each case we consider a unit energy spike seeded into an initially pressureless ideal gas with $\gamma = 1.4$, which should yield a peak shock density of 6.

6.3. 2D xy cylindrical solution blastwave on quadrilateral zones

First we consider the 2D *xy* case on a quadrilateral mesh in a unit box. The initial energy spike is seeded in a shaped material in the central four cells, with a radius of a single cell width. This implies the central energy spike becomes narrower and more intense (i.e., larger specific energies) as

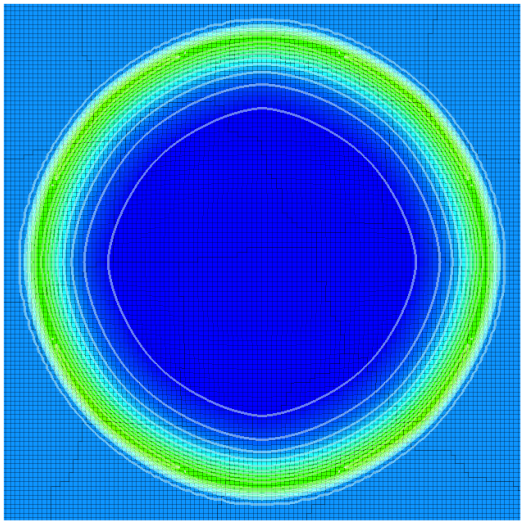


(a) Image with contours

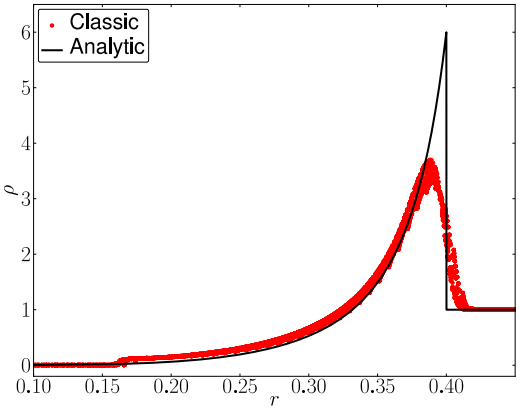


(b) Profile

Figure 12: Density (a) image and (b) profile for the 2D xy Sedov-Taylor blast wave for the 128×128 Lagrangian simulation. In (a) the color scales are the density (maxing at 6 in red), with density contours overplotted. Note for the profiles in (b) we have plotted the density in all cells as a function of radius, so the scatter gives some information as to the maintenance of symmetry.

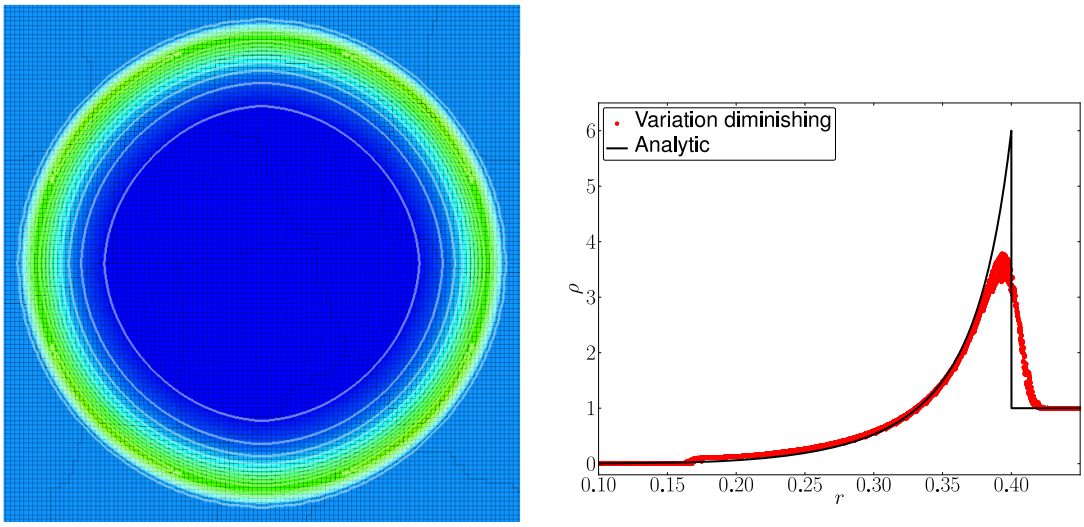


(a) Image with contours



(b) Profile

Figure 13: Same as Fig. 12 for the Classic ALE case.



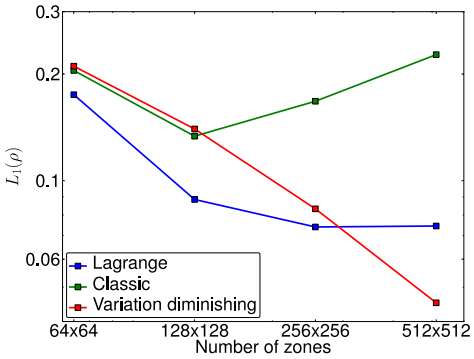
(a) Image with contours

(b) Profile

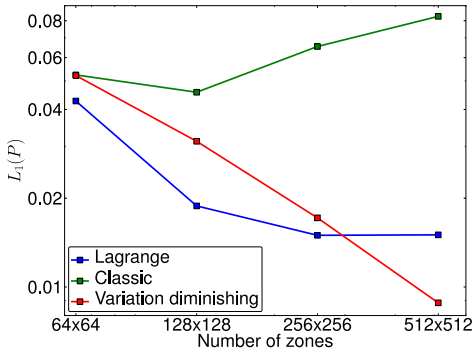
Figure 14: Same as Figs. 12–13 for the Variation diminishing ALE case.

	Lagrange	ALE classic	ALE variation diminishing
ρ	-0.40	0.077	-0.74
P	-0.48	0.25	-0.85
v_r	-0.54	-0.26	-0.84

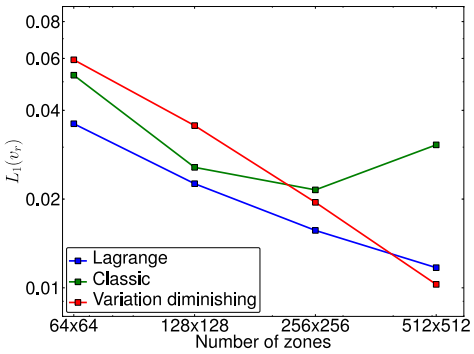
Table 1: Fitted order of convergence for L_1 norms in the 2D xy Sedov-Taylor simulations.



(a) ρ



(b) P



(c) v_r

Figure 15: L_1 errors in the 2D xy Sedov-Taylor blast wave as a function of resolution for the (a) density, (b) pressure, and (c) radial velocity.

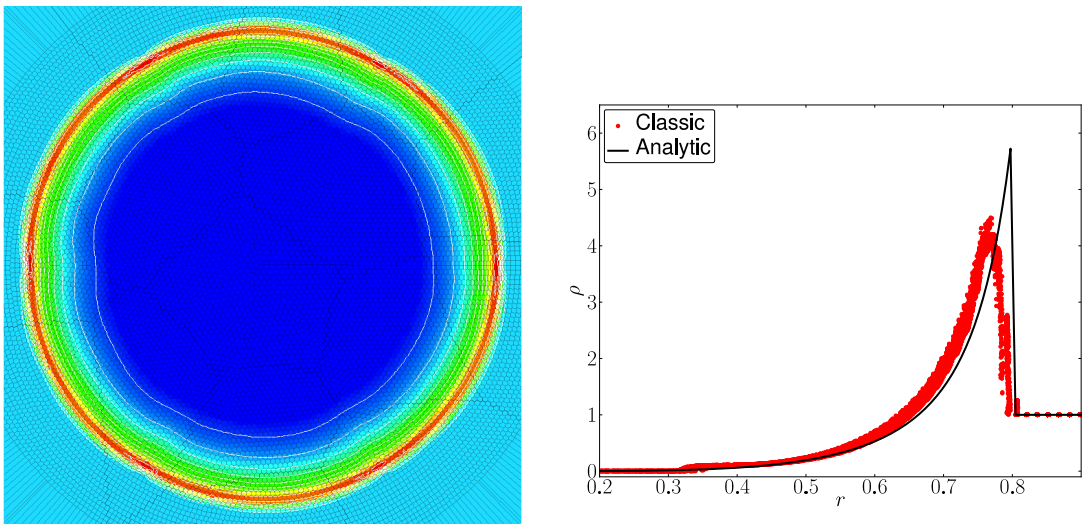
we go to higher resolutions/smaller cells, making this a very stressful test problem. We advance to the time corresponding to the analytic shock front advancing to a position of $r = 0.4$, occupying most of the simulated volume. We perform simulations at the same set of resolutions used in the cyclic remapping test case (64×64, 128×128, 256×256, & 512×512) allowing us to examine the convergence properties of each method.

Figs. 12–14 show snapshots of the mass density for the 128×128 simulations at the final time: panels (a) are images of the density, while panels (b) are scatter plots vs. radius of the densities. Note these radial profiles plot the values for all cells as a function of radius, so the scatter in these profiles gives some idea of how well symmetry is maintained. The Lagrange solution in Fig. 12 maintains the cylindrical symmetry very well, and resolves the expected peak density of $\rho_{\text{shock}} = 6$ with just a bit of overshoot. The ALE solutions don't resolve the peak density as well because these nearly Eulerian runs prevent the pileup of fine resolution near the shock front such as occurs in the Lagrange calculations. The classic ALE algorithm shows some evidence of distortion from cylindrical symmetry in the density contours, and perhaps more importantly is retarding the shock. The variation diminishing ALE results in Fig. 14 demonstrate visually improved symmetry and better shock position relative to the classic ALE results.

To be a bit more analytical in our comparison, Fig. 15 plots the L_1 errors for the density, pressure, and radial velocities at the final times as a function of resolution. As before we fit the errors and report the convergence rates for m in in table 1. Recall that first-order convergence corresponds to $m = -1$, while $m \geq 0$ implies non-convergence. It is evident that as we increase resolution the classic ALE case is failing to converge. This error is dominated by the convergence to the incorrect shock position, most likely because of the lack of conservation in the linear momentum in the classic formalism. By contrast the variation diminishing ALE algorithm demonstrates near first-order convergence, and in fact becomes more accurate than the Lagrange result at the highest resolution. This is due to plateauing of the accuracy of the Lagrange case – a subject for another discussion.

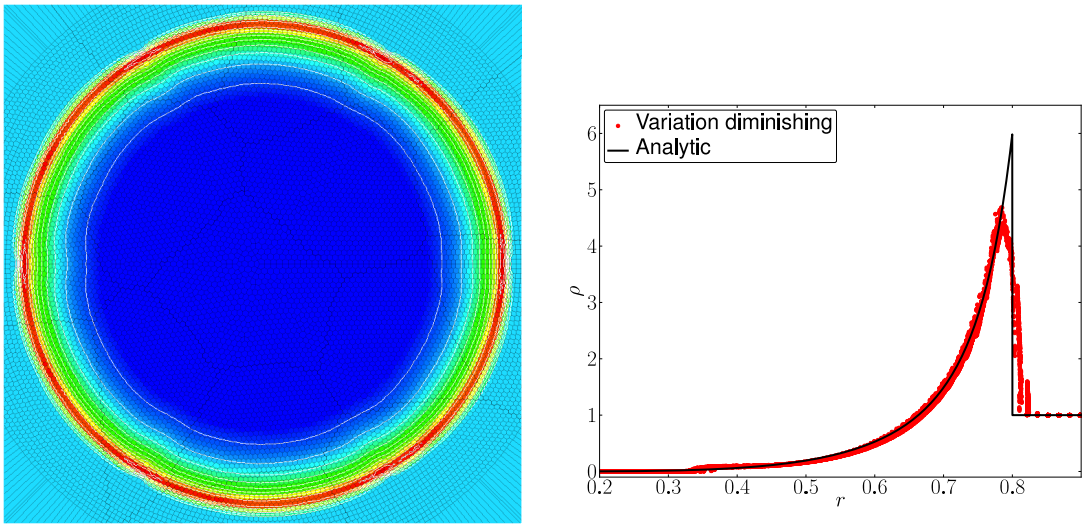
6.4. 2D xy cylindrical solution blastwave on arbitrary polygonal zones

This case is the same physical problem as we considered in §6.3, but now we employ meshes composed of arbitrary polygonal zones. These meshes are constructed using a Voronoi tessellation [15] based on seed or generator positions for each zone. The zone generators are seeded on N_r regular rings



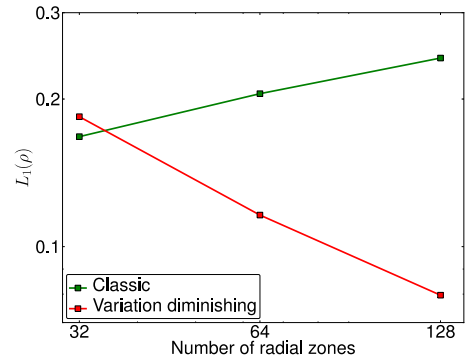
(a) Image with contours (b) Profile

Figure 16: Density (a) image with contours and (b) profile for the 2D xy Sedov-Taylor blast wave using the classic ALE algorithm on arbitrary polygonal zones, in this case the $N_r = 64$ zone case. In (a) the color scales are the density with density contours overplotted. As in previous figures the scatter plot profile of $\rho(r)$ in (b) plots results for all cells.

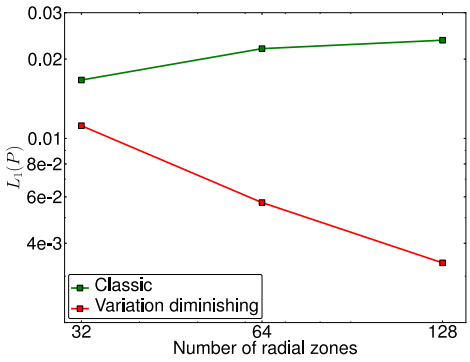


(a) Image with contours (b) Profile

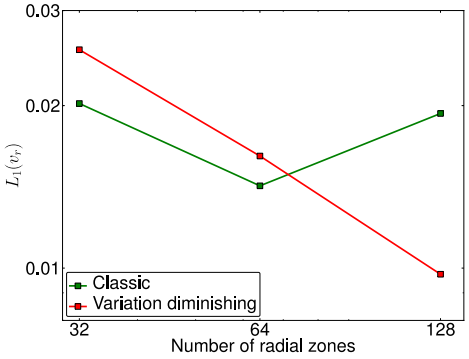
Figure 17: Same as Fig. 16 for the variation diminishing ALE case.



(a) ρ



(b) P



(c) v_r

Figure 18: L_1 errors in the 2D xy polygonal Sedov-Taylor blast wave as a function of resolution for the (a) density, (b) pressure, and (c) radial velocity.

	ALE classic	ALE variation diminishing
ρ	0.27	-0.61
P	0.25	-0.86
v_r	-0.03	-0.55

Table 2: Fitted order of convergence for L_1 norms in the 2D xy Sedov-Taylor simulations on arbitrary polygons.

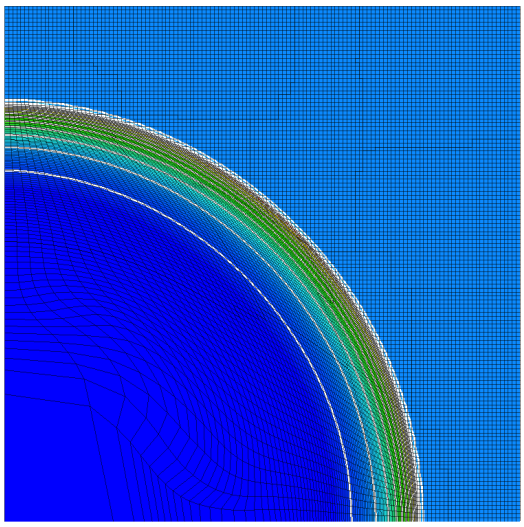
	Lagrange	ALE classic	ALE variation diminishing
ρ	-0.58	-0.60	-0.54
P	-0.59	-0.52	-0.68
v_r	-0.74	-0.87	-0.74

Table 3: Fitted order of convergence for L_1 norms in the 2D rz Sedov-Taylor simulations.

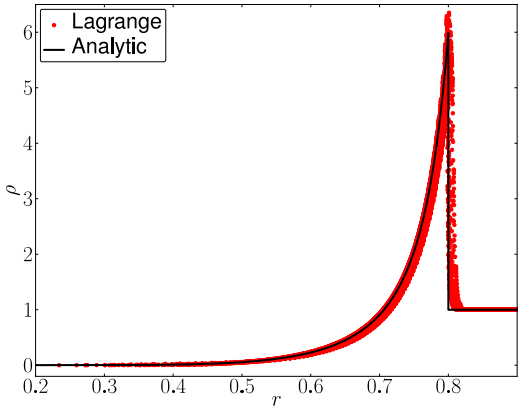
surrounding the origin with a radial step size of $1/N_r$ between rings. Each ring consists of an integer number of zones chosen such that the circumferential spacing is as close as possible to the radial step size. We perform tests using meshes with radial resolutions of $N_r \in [32, 64, 128]$. We only show ALE result for this test case, as the Lagrange calculations are prohibitive on these meshes due to small edges. Figs. 16–17 show images and profiles of the mass density for the $N_r = 64$ cases when the analytic solution predicts the shock to be at $r = 0.8$. The classic ALE algorithm once again retards the shock position relative to the analytic solution, and shows more scatter as compared with the variation diminishing result. This difference is reflected in the error analysis of Fig. 18 and table 2, where we find the classic algorithm is not converging to the correct solution whereas the variation diminishing ALE approach is. As intended, the variation diminishing formalism works reliably on these unstructured meshes of arbitrary polygons, yielding similar results to the structured case considered in §6.3.

6.5. 3D rz spherical solution blastwave on quadrilateral zones

We next consider the Sedov-Taylor problem in rz geometry. In this case we need to employ the r weighting scheme described in §5.4 when solving for the corner mass fluxes. We model one quadrant of this problem in a unit rz box until the predicted shock position reaches a radius of 0.8, employing mesh resolutions of 32×32, 64×64, 128×128, & 256×256. Figs. 19–21 show snapshots of the density at this final time for the 128×128 models. In the Lagrange image we see some deviation from perfect radial evolution near the origin due to unphysical post-shock vorticity (evident in the mesh geometry near the origin). However, this error is indistinguishable in the profile plot of Fig. 19b. Comparing the classic vs. variation diminishing ALE results in Figs. 20–21, we see the error in the shock position is much reduced for the classic ALE calculation relative to the earlier xy calculations. This is born out by the error analysis in Fig. 22 and table 3, where we find little difference

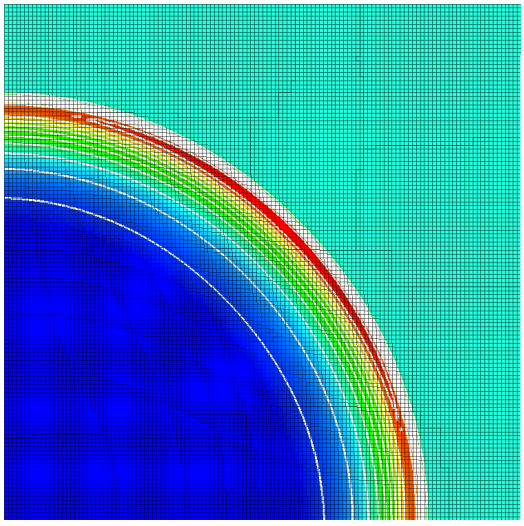


(a) Image with contours

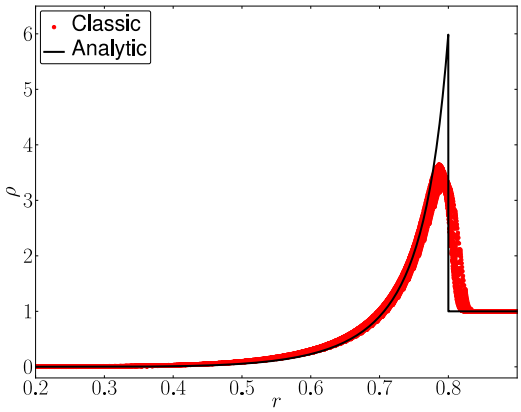


(b) Profile

Figure 19: Density (a) image and (b) profile for the 3D rz Sedov-Taylor blast wave for the 128×128 Lagrangian simulation. In (a) we overplot the density contours on the image of the density. In (b) we have plotted the density in all cells as a function of radius, so the scatter gives some information as to the maintenance of symmetry.

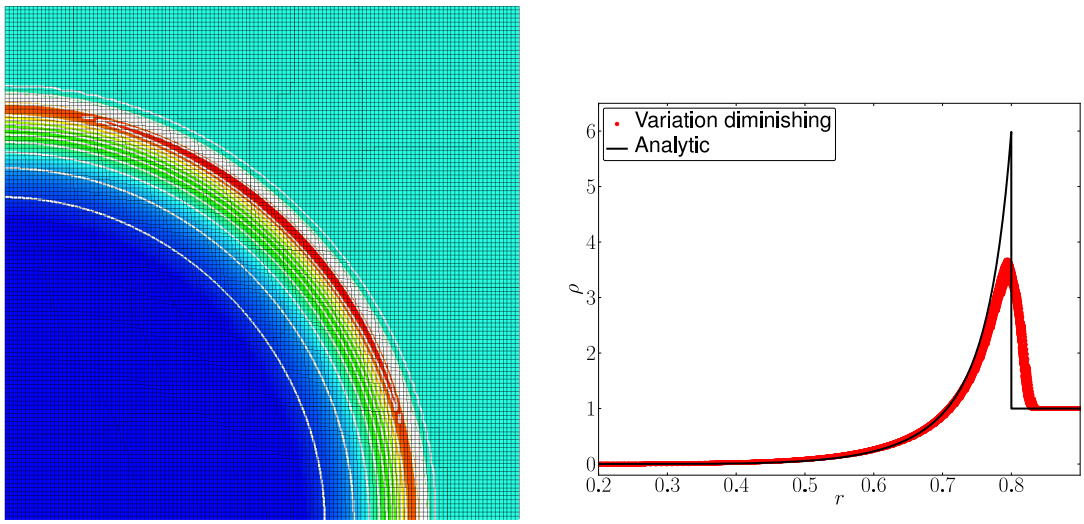


(a) Image with contours



(b) Profile

Figure 20: Same as Fig. 19 for the Classic ALE case.



(a) Image with contours (b) Profile

Figure 21: Same as Figs. 19–20 for the variation diminishing ALE case.

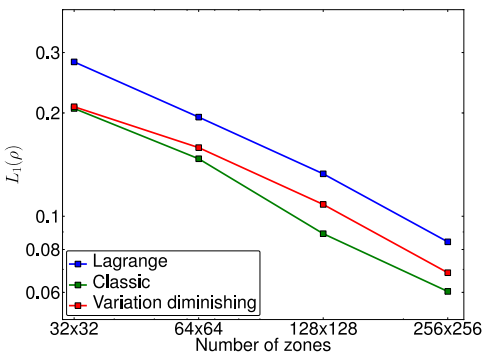
	Lagrange	ALE classic	ALE variation diminishing
ρ	-0.47	-0.26	-0.50
P	-0.53	-0.66	-0.99
v_r	-0.56	-0.50	-0.74

Table 4: Fitted order of convergence for L_1 norms in the 3D xyz Sedov-Taylor simulations.

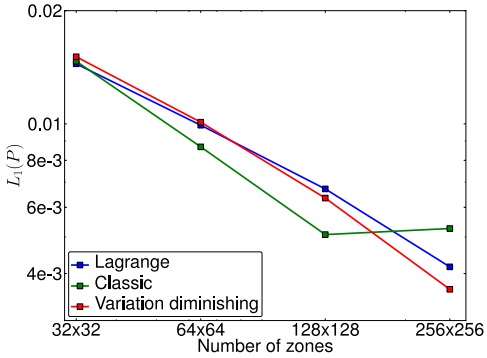
in the accuracy of the two ALE approaches (other than in the pressure, where the highest resolution classic ALE calculation shows more error). Regardless, it is evident that the newer variation diminishing algorithm maintains the expected symmetry in this case with the radial weighting approach discussed in §5.4: without these modification calculations show strong deviations from the proper solution near the axis in the form of jetting.

6.6. 3D xyz spherical solution blastwave on hexahedral zones

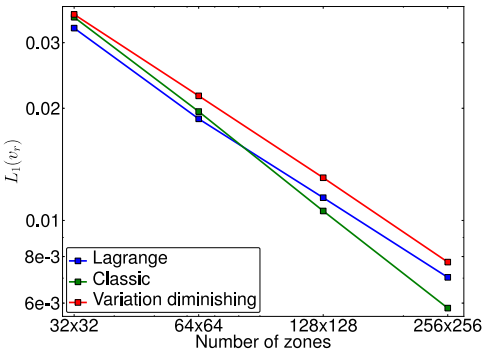
For our final example we run an octant of the 3D Sedov-Taylor blastwave on a 3D mesh consisting of hexahedra. We consider four different mesh resolutions for the unit cube, with $N_{\text{zones}} \in (16^3, 32^3, 64^3, 128^3)$. Figs. 23–25 show the results for the 128^3 calculations at the final time, when the analytic prediction of the shock position is at $r = 0.8$ in our unit box. The



(a) ρ

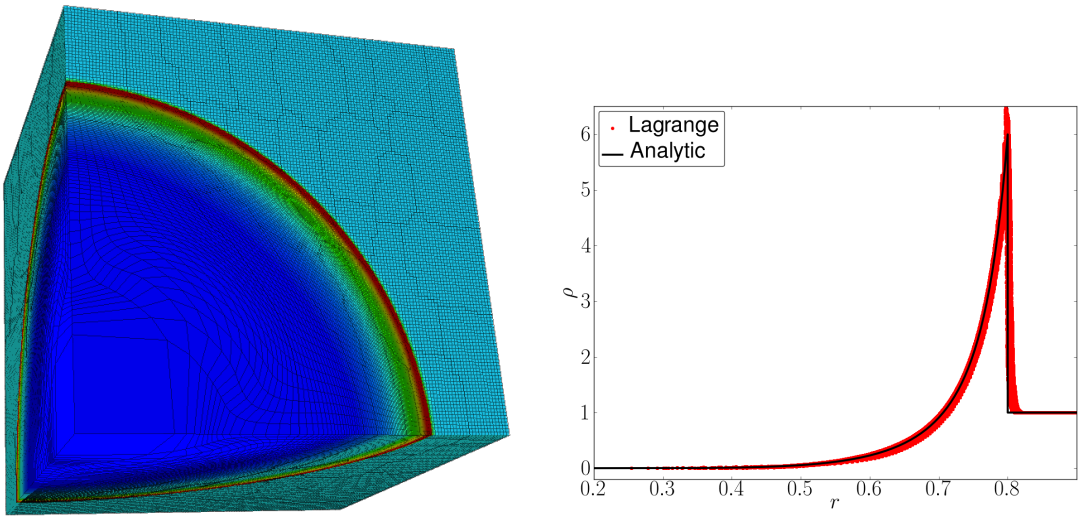


(b) P



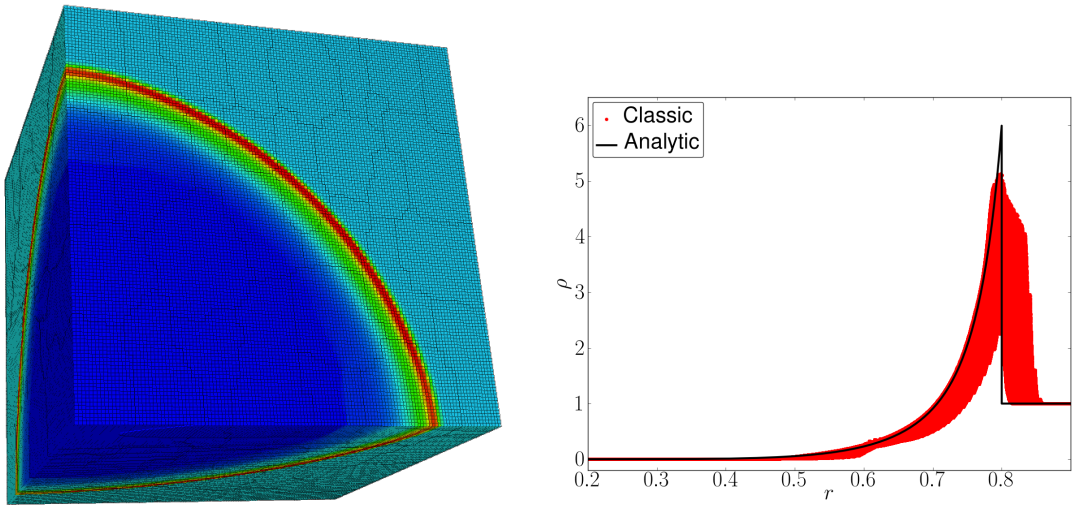
(c) v_r

Figure 22: L_1 errors in the 2D rz Sedov-Taylor blast wave as a function of resolution for the (a) density, (b) pressure, and (c) radial velocity.



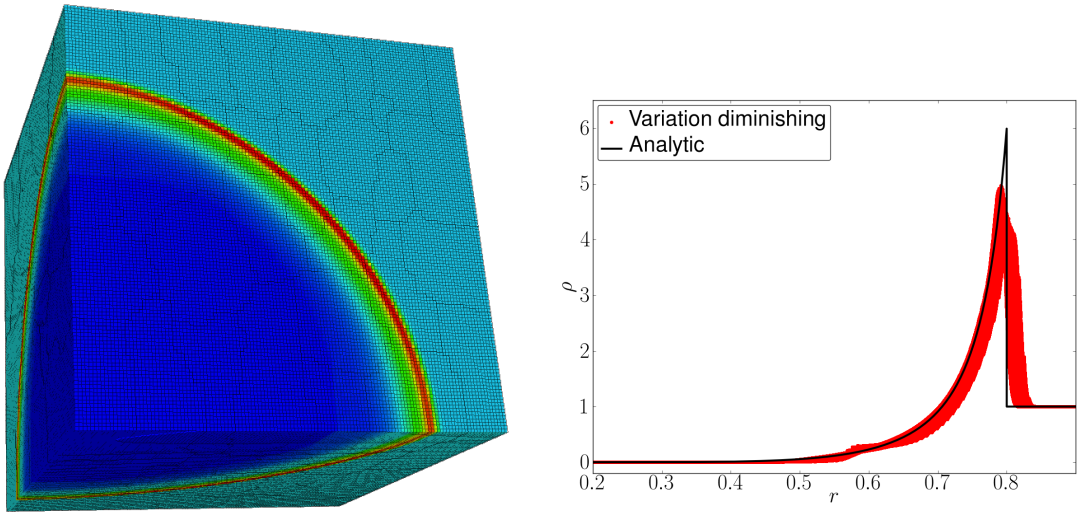
(a) Density image (b) Profile

Figure 23: Density (a) image and (b) profile for the 3D xyz Sedov-Taylor blast wave for the 128^3 Lagrangian simulation. The pseudo-color plot in (a) shows the full octant at the time the shock front should be at $r = 0.8$, while (b) plots the zone centered density values vs. radius.



(a) Density image (b) Profile

Figure 24: Same as Fig. 23 for the classic ALE case.



(a) Density image

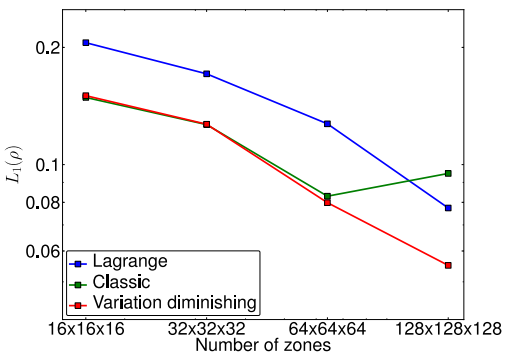
(b) Profile

Figure 25: Same as Figs. 23–24 for the variation diminishing ALE case.

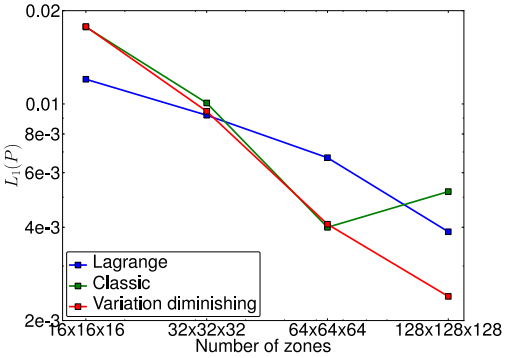
Lagrangian calculation in Fig. 23 maintains excellent symmetry: the two ALE calculations see somewhat more spread in the $\rho(r)$ density profiles, though consistent with our findings in 2D the variation diminishing ALE calculation maintains symmetry more effectively. Examining the behavior of the L_1 error norms in Fig. 26, we see that the two ALE calculations track very closely in terms of accuracy for the lower resolution calculations. However, at the highest resolution the classic ALE calculation stops converging, whereas the variation diminishing algorithm continues to improve. This is reflected in the convergence rates tabulated in table 4, where we see the convergence rate of the classic ALE results falling off while the variation diminishing ALE calculations show the most reliable convergence. Overall the 3D results are consistent with our prior 2D calculations.

7. Conclusions

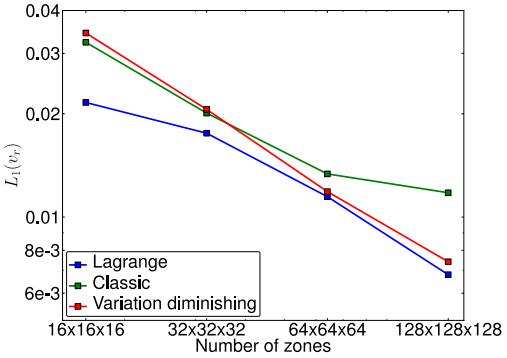
In this paper we have considered several different algorithms for handling the subzonal corner masses of the staggered compatible Lagrangian discretization through an ALE remap, and demonstrated that the variation diminishing idea of §4.5 best meets our criteria of conservation, monotonicity, consistency, and accuracy. This algorithm works well for a variety of



(a) ρ



(b) P



(c) v_r

Figure 26: L_1 errors in the 3D xyz Sedov-Taylor blast wave as a function of resolution for the (a) density, (b) pressure, and (c) radial velocity.

meshes and geometries: 2D xy & rz geometries, 3D xyz, and both structured and unstructured cases. The major advantages of this approach are that it works well with a standard ALE remapping approach, being based on flux-based algorithms on the primary (zonal) and dual (nodal) meshes, and does not add significantly to the expense of such an ALE algorithm. The algorithms described here represent a simple extension of standard staggered ALE algorithms, and integrate easily into codes such as KULL [20] and FLAG [5, 6, 7, 11] which had existing staggered ALE implementations and now employ these ideas.

Other approaches to the problem of consistent ALE remapping for use with the staggered compatible Lagrange step have been suggested. Since the zonal and nodal control volumes are built up from the subzonal corner volumes, one natural approach to solve the remapping problem might be to scatter the conserved properties to the corners, remap on this finer discretization, and then simply sum up the post-remap conserved properties once again to define the new nodal and zonal values via Eqs. 2 (this idea is explored in [13]). This is conceptually an elegant method of handling the remap procedure, but there are two important issues to consider. First, as a practical matter such an approach greatly increases the cost of the remap – in 3D a mesh consisting of hexahedra has 8 times the number of corners as zones. Secondly, how to handle multiple materials is somewhat problematic. Modern ALE methods employ interface reconstruction within zones to isolate multiple materials, and it does not make sense in the Lagrangian step to have different sets of materials in the various corners of a given zone. Rather the compatible Lagrange step would need to homogenize this subzonal material information. This points out the fundamental issue with this approach: the hydrodynamic equations are differenced on the level of the staggered zonal and nodal volumes, not the corners. The extra effort of remapping on corners is wasted, as that effectively much greater resolution (and associated numerical work for the remap) is ignored by the hydrodynamic differencing. Therefore we believe an approach such as described in this paper is more appropriate for utilizing the staggered compatible Lagrange method as part of an ALE algorithm.

It is also possible to extend these ideas to more general ALE applications. In [9], the authors use an overlay or intersection based remap for the remapping stage of an ALE algorithm. By casting the intersection volumes as fluxes from donor to acceptor volumes, the authors are able to adapt the algorithms described in this paper in order to define the subzonal corner

properties. The only difference is that there are more fluxes between elements, since the intersection based remap defines donor & acceptors between all overlapping elements, not just those connected by faces as in our examples here.

8. Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. The work of M. Shashkov was performed under the auspices of the National Nuclear Security Administration of the US Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396 and was partially supported by the DOE Advanced Simulation and Computing (ASC) program and the DOE Office of Science ASCR Program.

[1] D. J. Benson, An efficient, accurate, simple ALE method for nonlinear finite element programs, Comput. Meth. Appl. Mech. Eng. v. 72, pp. 305-350, 1989.

[2] D. J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Comput. Meth. Appl. Mech. Eng. v. 99, pp. 235-394, 1992.

[3] D. J. Benson, Momentum advection on a staggered mesh, J. Comp. Phys. v. 100, pp. 143-162, 1992.

[4] R. L. Bowers & J. R. Wilson, Numerical modeling in applied physics and astrophysics, Jones and Bartlett, Boston, 1991.

[5] D. E. Burton, Exact conservation of energy and momentum in staggered-grid hydrodynamics with arbitrary connectivity, in: Advances in the Free Lagrange Method, Springer-Verlag, New York, 1990.

[6] D. E. Burton, Multidimensional discretization of conservation laws for unstructured polyhedral meshes, Report of Lawrence Livermore National Laboratory, UCRL-JC-118306, 1994.

[7] D. E. Burton, Multidimensional discretization of conservation laws for unstructured grids, in: Second International Workshop on Analytical

Methods and Process Optimization in Fluid and Gas Mechanics (SAMGOP), Arzamas-16, Russia (also report UCRL-JC-118306, Lawrence Livermore National Laboratory), 1994.

[8] 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. Comp. Phys. v. 146, pp. 227–262, 1998.

[9] M. Kucharik & M. Shashkov, Conservative Multi-Material Remap for Staggered Multi-Material Arbitrary Lagrangian-Eulerian Methods, LANL Report, LA-UR-1322164, 2013, Submitted to Journal of Computational Physics.

[10] M. Kenamond, M. Bement, & M. Shashkov, Compatible, Total Energy Conserving and Symmetry Preserving Arbitrary Lagrangian-Eulerian Hydrodynamics in 2D rz - Cylindrical Coordinates, LANL Report, LA-UR-112-26781, 2012, Submitted to Journal of Computational Physics.

[11] K. Lipnikov & M. Shashkov, A framework for developing a mimetic tensor artificial viscosity for Lagrangian hydrocodes on arbitrary polygonal meshes, J. Comp. Phys. v. 229, pp. 7911-7941, 2010.

[12] R. Liska, M. Shashkov, P. Váchal, and B. Wendroff, Optimization-based synchronized flux-corrected conservative interpolation (remapping) of mass and momentum for Arbitrary Lagrangian-Eulerian methods, J. Comp. Phys. v. 229, pp. 1467–1497, 2010.

[13] R. Loubére & M. Shashkov, A subcell remapping method on staggered polygonal grids for arbitrary-LagrangianEulerian methods, J. Comp. Phys. v. 209, pp. 105–138, 2005.

[14] L. G. Margolin, Introduction to an arbitrary Lagrangian Eulerian computing method for all speeds, J. Comp. Phys. v. 135, pp. 198-202, 1997.

[15] A. Okabe, B. Boots, K. Sugihara, & S. Chiu, Spatial Tessellations, Concepts and Applications of Voronoi Diagrams, second ed., John Wiley & Sons Ltd., Chichester, 2000.

[16] J. M. Owen, Subzonal corner mass treatments during ALE remap, LLNL-PRES-413307, presented at Numerical Methods for Multimaterial Fluids and Structures, Pavia, Italy, Sep. 21–25, 2009.

[17] J. M. Owen & M. Shashkov, ALE remap treatments consistent with corner based compatible total energy conserving Lagrangian methods, LLNL-PRES-457355, presented at the Nuclear Explosives Code Developers Conference, Los Alamos, NM, 2010.

[18] J. M. Owen & M. Shashkov, A corner and dual mesh ALE remapping algorithm for use with the compatible energy Lagrangian discretization, presented at the SIAM Conference on Computation Science and Engineering, Reno, NV, 2011.

[19] R. B. Pember & R. W. Anderson, A comparison of staggered-mesh Lagrange plus remap and cell-centered direct Eulerian Godunov schemes for Eulerian shock hydrodynamics, UCRL-JC-139820, presented at the Nuclear Explosives Code Developers Conference, Oakland, CA, 2000.

[20] J. A. Rathkopf, et. al., KULL: LLNL's ASCI Inertial Confinement Fusion Simulation Code, Article submitted to Physor 2000 American Nuclear Society Topical Meeting, Pittsburgh, PA, UCRL-JC-137053, 2000.

[21] M. L. Wilkins, Calculations of Elastic-Plastic Flow, LLNL report, UCRL-7322, 1963.

[22] L. I. Sedov, Propagation of strong shock waves, Journal of Applied Mathematics and Mechanics v. 10, pp. 241–250, 1946.

[23] L. I. Sedov, Similarity and Dimensional Methods in Mechanics, Academic Press, New York, 1959.

[24] J. Lockwood-Taylor, An exact solution of the spherical blast wave problem, Philosophical Magazine v. 46, pp. 317–320, 1955.