Habilitation Thesis – Habilitační práce

Czech Technical University in Prague

F4

Faculty of Nuclear Sciences and Physical Engineering
Department of Physical Electronics

Numerical Methods for Quantity Remapping in the Context of Indirect Arbitrary Lagrangian-Eulerian (ALE) Hydrodynamics

Numerické metody pro remapování veličin v kontextu nepřímé lagrangeovsko-eulerovské hydrodynamiky

Milan Kuchařík

Field of study: Applied Mathematics – Aplikovaná matematika
November 2017

Acknowledgements

In the first place I would especially like to thank my family for love, support, and understanding. I would also like to thank my former supervisor Richard Liska for help, support, a lot of good advice, and opportunity to work on interesting topics. Thanks also to my colleagues from the Department of Physical Electronics for a friendly environment and teaching me many things, namely Pavel Váchal, Jiří Limpouch, and Ondřej Klimo. Special thanks belong to my postdoc advisor Mikhail Shashkov for a great opportunity to work at LANL, for bringing me to the topic of this Habilitation, and for a great inspiration in many interesting subjects. Finally, I would like to thank my foreign colleagues who influenced me a lot, and from whom I learned many important points, namely Hyung Ahn, Jan Badziak, Markus Berndt, Jérôme Breil, Vadim Dyadechko, Stéphane Galera, Rao Garimella, Raphaël Loubère, Pierre-Henri Maire, Daniele Margarone, Sam Scofield, Guglielmo Scovazzi, Stanly Steinberg, Stefan Weber, and Burton Wendroff.

Abstract

For hydrodynamic simulations of fast flows, the indirect Arbitrary Lagrangian-Eulerian (ALE) methods represent one of the few state of the art numerical approaches, which are efficient, accurate, and robust enough for realistic calculations. In this approach, a Lagrangian solver is used to advance the solution along with the computational mesh in time, while its robustness is achieved by smoothing (regularization) of the mesh. Remapping is one of the essential parts of the ALE algorithm, conservatively interpolating the fluid quantities between different computational meshes. This habilitation thesis summarizes the contribution of the author in the field of remapping methods. After a brief description of the ALE algorithm, it focuses on the description of the remapping approaches and emphasizes the input of the author in the form of his commented articles. It mainly includes his work related to a combination of intersection- and swept-based remapping approaches, the remap in multi-material problems, and the development of compatible algorithms for the remap of all fluid quantities. Finally, several applications are presented, especially from the field of hydrodynamic laser-plasma simulations.

Keywords: Arbitrary Lagrangian-Eulerian methods, Conservative Interpolations, Multi-Material ALE, Flux-Based Remap.

Abstrakt

Jednou z mála numerických metod pro hydrodynamické simulace rychlého prodění, které jsou v současné době považovány za dostatečně efektivní, přesné a robustní, jsou nepřímé metody lagrangeovsko-eulerovské (ALE). Tento přístup využívá lagrangeovského řešiče, který posouvá řešení v čase spolu s výpočetní sítí, zatímco robustnosti metody je dosaženo vyhlazováním sítě. Jednou z nezbytných částí ALE algoritmu je remapování sloužící ke konzervativní interpolaci jednotlivých veličin v tekutině mezi výpočetními sítěmi. V této habilitační práci jsou popsány příspěvky autora v oblasti metod pro remapování stavových veličin. Po krátkém popisu algoritmu ALE se práce zaměřuje na představení jednotlivých metod pro remapování veličin se zdůrazněním přínosu autora ve formě jeho komentovaných článků. Ty zahrnují jeho práci především v oblastech kombinování remapovacích algoritmů založených na průnicích a na oblastech posunutí, remapu v případě vícemateriálových problémů a vývoje metod pro kompatibilní remapování všech stavových veličin. Závěrem práce jsou zmíněny některé aplikace vyvinutých metod z oblasti hydrodynamických simulací laserového plazmatu.

Klíčová slova: Lagrangeovsko-eulerovské metody, konzervativní interpolace, multi-materiálové ALE, remap v tokovém tvaru.

Contents

Chapter 1

Introduction

Numerical methods were developed during the last centuries and their names still resemble the names of great mathematicians of that time, such as Newton, Lagrange, Gauss, or Euler. The complex physical phenomena started to be modeled long before the invention of modern computers, by hand. The first note about the numerical solving of partial differential equations comes from 1922 [171], which is considered the beginning of the scientific discipline nowadays called the Computational Fluid Dynamics (CFD). With the development of modern computers, the computer simulations grew rapidly, and the first large-scale simulations were deployed during the Manhattan Project in World War II. Since that, many computer simulation codes intended for different problems of physics have been developed.

The field of Computational Fluid Dynamics has a wide range of applications. The first applications were driven by the need of an accurate weather forecast for the navy, which is still an important part of CFD used by most people on a daily basis. Many industrial applications are of great importance, such as various aerodynamic simulations for designing the shape of cars, airplanes, missiles, or wind turbines, optimal placement of oil wells in the petroleum industry, blood flow in the veins or heart for the bioapplications, and many other areas. There exists a similarly broad range of scientific applications, related to astronomy or plasma physics, which is of main interest in this thesis.

The research in plasma physics is significantly motivated by the goal of the inertial confinement fusion (ICF), an infinite source of clean energy. In this concept, the energy is released due to the compression of light atoms (typically Hydrogen, Deuterium, and Tritium) in the form of micropelets, ignited (directly or indirectly) by a strong laser pulse. In this problem, many partial issues are being simulated by production ICF codes, such as laser absorption, shock wave formation, collision of shocks, degeneration of the target symmetry due to instabilities, etc.

In this thesis, the contributions of the author to the development of numerical methods for hydrodynamic simulation codes are summarized. In particular, the main topic of the thesis is the improvement of the numerical

methods for conservative interpolations (remapping) of fluid quantities, which is a crucial part of most Arbitrary Lagrangian-Eulerian (ALE) algorithms, representing one of few current technologies considered as efficient, accurate, and robust enough for realistic hydrodynamic calculations. The newly developed numerical methods have been tested in the framework of a research multi-material ALE code (RMALE), and selected methods have been implemented in the Prague ALE (PALE) code, which has been developed and is routinely used at the Department of Physical Electronics for realistic hydrodynamic simulations of laser-plasma interactions. Several applications from the ICF-oriented computer simulations are referenced to demonstrate ability of the developed techniques to be employed for complex physics computations.

This thesis is arranged as a commented collection of author's publications and is organized as follows. The concept of ALE methods is briefly overviewed in Chapter 2, describing all its steps and the position of remap in the algorithm. The remapping step is described in more detail in Chapter 3, paying attention to various aspects of quantity remap in the context of an ALE algorithm. The main part of the thesis is Chapter 4, containing comments on selected publications of the author related to quantity remap. Each of these papers is included as a separate Appendix at the end of the thesis, its content is reviewed, and the contributions of the author are summarized. In Chapter 5, the applications of the developed numerical methods in the field of plasma physics are mentioned, not only demonstrating their applicability in demanding realistic computer simulations, but also bringing important new results from the physics of laser/plasma interactions. The Thesis is concluded in Chapter 6. The name of the author is emphasized in bold in the following Bibliography, to allow the reader to simply see, which papers are coauthored.

Chapter 2

Arbitrary Lagrangian-Eulerian Methods

In the numerical hydrodynamics simulations, the choice of the computational mesh is crucial. Traditionally, there have been two viewpoints utilizing the Lagrangian or the Eulerian framework, each with its own advantages and disadvantages.

The class of Eulerian methods employs a static computational mesh. This mesh is typically constructed at the initialization stage and remains unchanged during the whole simulation. The fluid moves through the edges (faces) of such mesh in the form of mass fluxes. The Eulerian methods are typically simpler to analyze and implement, there exists a vast amount of related literature, for example [189, 130, 128]. Unfortunately, this approach is not well suited for certain types of simulations, typically involving strong material compressions or expansions, which is a common situation in the field of plasma hydrodynamics.

In the Lagrangian methods, the computational mesh moves with the fluid and no mass fluxes between the computational cells are present [56, 178, 36, 70, 194, 145, 50, 52, 51, 48, 60, 61, 63, 64]. Typically, fluid is described by the set of Euler equations,

$$\frac{1}{\rho}\,\dot{\rho} + \nabla \cdot \vec{u} = 0\,, \tag{2.1}$$

$$\rho\,\dot{\vec{u}} + \nabla p = \vec{0}\,, \tag{2.2}$$

$$\rho\,\dot{\varepsilon} + p\,\nabla \cdot \vec{u} = 0\,, \tag{2.3}$$

representing conservation of mass, momentum, and energy, where the symbols ρ, $\vec{u}$, p, and ε stand for fluid density, velocity, pressure, and specific internal energy respectively. The system is closed by an equation of state (EOS) in the form $p = P(\rho, \varepsilon)$. The motion of the mesh naturally follows the fluid motion and is described by an ordinary differential equation $\dot{\vec{x}} = \vec{u}$, so the computational domain changes adaptively as necessary even for strong material deformations. This is the main reason, why mostly methods based on the Lagrangian concept are used in laser-plasma hydrodynamics. The main disadvantage of the Lagrangian methods results from the mesh motion – the computational mesh can distort and invalid (non-convex, flipped, or negative-

volume) cells can appear, which typically results in simulation failure. The source of the problem is either physical (such as presence of strong shears or vortexes) or numerical (resulting typically from a non-physically high degree of freedom of the quadrilateral cell), which is often termed as hourglass. Although there exist methods for treating such patterns [31, 52, 179, 119], this problem is not solved universally and more advanced methods must be used.

In a pioneering paper [91], Hirt et al. developed the formalism for a mesh whose motion could be determined as an independent degree of freedom, and showed that this general framework could be used to combine the best properties of the Lagrangian and Eulerian methods. This class of methods has been termed Arbitrary Lagrangian-Eulerian or ALE. Many authors have described the ALE strategies to optimize accuracy, robustness, or computational efficiency, see for example [32, 154, 102, 165, 6, 66, 137, 136, 81, 44, 144, 27, 113, 132].

The ALE methods can be split in two families, often called as direct and indirect ALE. The direct ALE methods are characterized by an a priori specified velocity field, defining the motion of the computational mesh. The fluid in the simulation then evolves with the numerical fluid velocity, independent of the mesh velocity. Such approach is typically useful when the evolution of the problem is approximately known, so the velocity field can be chosen close to the real fluid motion [65, 79, 43]. In current literature, the numerical fluid velocity is often used with vortex or shear velocity modes filtered out by its decomposition in several components [55, 55].

In this thesis, we primarily focus on the indirect ALE methods [26, 99, 135, 137, 136, 148, 93, 27, 132], in which the numerical fluid velocity is used for the mesh motion, which is repaired a posteriori by a special mesh smoothing technique. The indirect ALE algorithm can be typically split in three distinct steps:

1. Lagrangian solver, advancing the fluid quantities and the computational mesh to the next time level;
2. mesh rezoner, untangling and smoothing the mesh when its geometric quality becomes low;
3. remapper, conservatively interpolating (transferring) all fluid quantities from the Lagrangian to the rezoned mesh.

The last step of the ALE algorithm is the main topic of this thesis. In general, the rezoning and remapping steps are jointly called as the Eulerian part of the ALE algorithm, because the computational mesh is moved to different (non-Lagrangian) positions and mass fluxes across the mesh edges appear during the remapping step.

The Lagrangian solvers can be derived in different formulations, with the cell-centered and staggered discretization being the most popular. In the

cell-centered discretization, all fluid quantities are located in the center of computational cells, and nodal velocity needed for the mesh motion must be further computed using an approximate Riemann solver. This approach has become very popular in recent years [61, 147, 143, 149, 145, 146] and when employed in the full ALE algorithm, remapping becomes simple due to its cell-based nature. We focus here on the other option – the staggered discretization [50, 53, 52] based on the compatible mimetic approach [180], in which the thermodynamic fluid quantities (density, pressure, energy) are defined in the cell centers, while the kinematic quantities (such as velocity) are defined at the mesh nodes. This formulation has traditionally been used in many hydrodynamic codes [88, 84, 46, 78, 173, 8, 9, 160, 135, 132] and its advantage is the direct knowledge of the nodal velocity advancing the computational mesh. On the other hand, when incorporated in the full ALE code, remap becomes more complicated.

From the point of view of the ALE algorithm, the mesh rezoning procedure should move only the nodes, which need to be moved, and move them as little as possible to avoid excessive diffusion of the following remapping step. There exists a large number of rezoning techniques, which can be used for fixing the computational mesh when it becomes distorted, for an overview see for example [107]. Traditionally, simpler methods are being used due to their computational performance, such as techniques based on Laplace smoothing [89] or the classical Winslow mesh smoothing method [197, 187, 109]. Such methods work acceptably in most situations, but can produce too strong mesh motion even in places, where no significant mesh fix has to be performed. To avoid this problem, more elaborate methods have been developed, reducing mesh motion in regions where no significant fix is necessary. Methods based on condition number minimization [108, 191], or reference-Jacobian method [110, 182, 71] can be mentioned as an example. Different approaches need to be used when mesh untangling needs to be performed, because the previously mentioned methods do not guarantee validity of the smoothed mesh. There exist approaches based on modification of the condition number functional [74], or pure geometric approaches [4, 190, 1, 35] based on an explicit construction of a set of valid nodal positions. The last topic we mention here is smoothing of external and internal domain boundaries. The treatment of internal boundaries can be avoided in some situations by the introduction of slide lines [30, 196, 49, 118]. Smoothing of external domain boundaries is typically treated by their shifting along approximated curves (linear or more complex) [198, 77], or by a volume-preserving boundary approximation [126, 101].

The last step of the ALE algorithm is remap, transferring conservatively all fluid quantities from the Lagrangian to the rezoned mesh. There exist several possible approaches for remapping, for a review see the seminal paper by Benson [32]. The most natural approach is based on employing zonal intersections between both meshes and integration of the reconstructed quantities over the intersections [67, 69, 86]. Due to the necessity of the intersection construction, this approach is computationally expensive and its

implementation in a robust way is difficult (the intersections of close meshes often involve nearly-parallel edges). Therefore, simpler approaches have been developed, from which the swept-region-based method [68, 155, 156, 125] is most popular. In this approach, remap is formulated in a flux form, and fluxes are constructed by integration of the quantity reconstruction over the approximate region defined by the imaginary motion of the Lagrangian cell edge to its new position. This approach is significantly faster than intersections, but can suffer from problems related to monotonicity- and bound-preservation. For a detailed analysis and comparison of both approaches, one can see [155, 129, 105]. The remap stage of the ALE algorithm is described in more detail in Chapter 3.

A large amount of work has been performed in the field of multi-material ALE during recent years [7, 165, 138, 26, 5, 80, 81, 172, 45, 44, 27]. In this case, multiple materials are allowed in each computational cell. Their amount is typically specified by their relative volumes or masses, usually called as volume/mass fractions. The approximate position of the particular material in a cell can be specified by its centroid, i.e. the geometric center in the integral sense. This situation is unavoidable in ALE simulations if multiple materials are present, even in cases when only single-material cells are constructed during initialization, because mesh rezoning does not respect material interfaces and mixed cells are created. Therefore, treatment of multiple materials in a single cells is necessary. From the point of view of the Lagrangian solver, a closure model [184, 188, 25, 39, 59, 62, 184, 158, 161, 98, 97, 181, 24] is required, defining the material interaction, or in other words, specifying the evolution of the material amount and position during the Lagrangian step and the average cell pressure required for the pressure forces construction. Simple closure models are based on an equal-strain assumption or on a slower relaxation to an average pressure, while the advanced models fully simulate the motion of the material interfaces via approximate Riemann solvers. In the remapping step, distribution of each material in the cell needs to be explicitly known, therefore a method for material reconstruction [162, 186, 29, 85, 163, 11, 40, 22, 195, 176, 177, 175, 115] has to be employed. Historically, the Volume of Fluid (VOF) approach [92, 199, 200, 159, 185, 33, 94, 57, 183, 174] became popular, recovering the material polygons (polyhedra) from the volume fraction data in the particular cell and its neighbors. There exist several variants of the VOF material reconstruction method, based on reconstruction of the volume fraction field, or minimization of the error in all the surrounding cells. In recent years, a modern Moment of Fluid (MOF) method [72, 73, 2, 3, 90] has been developed, which requires an approximate knowledge of material centroids together with the volume fraction data. The material polygons are then constructed by minimizing the error between the provided centroids from the reference data. This approach is able to recover smaller artifacts in the interface profile, it is local (requires information from one cell only), and naturally allows to choose the best material ordering when more than two materials are present, which is a big problem for the VOF family of methods.

Multi-material ALE is a key requirement for physically relevant numerical simulations. In cases, when involved materials have significantly different properties (such as air and metal, for example), a standard numerical ALE simulation leads to numerical material mixing and incorrect treatment of the mixture by a meaningless EOS. On the other hand, the multi-material ALE methods handle the materials separately and different EOSes are used for each component. Then, the simulation corresponds to a physical reality much better, so the development of multi-material ALE methods is a significant contribution to the reliability of numerical predictions and modeling of physical processes.

Chapter 3

Conservative Remapping

Remapping is one of the key steps of a typical ALE algorithm, transferring conservatively all fluid quantities from the Lagrangian mesh to the rezoned one. For an example of a Lagrangian and a rezoned mesh (obtained from the Lagrangian mesh by the application of the Winslow rezoning algorithm [197, 187, 109]), see Figure 3.1. As we can see, both meshes are very close to each

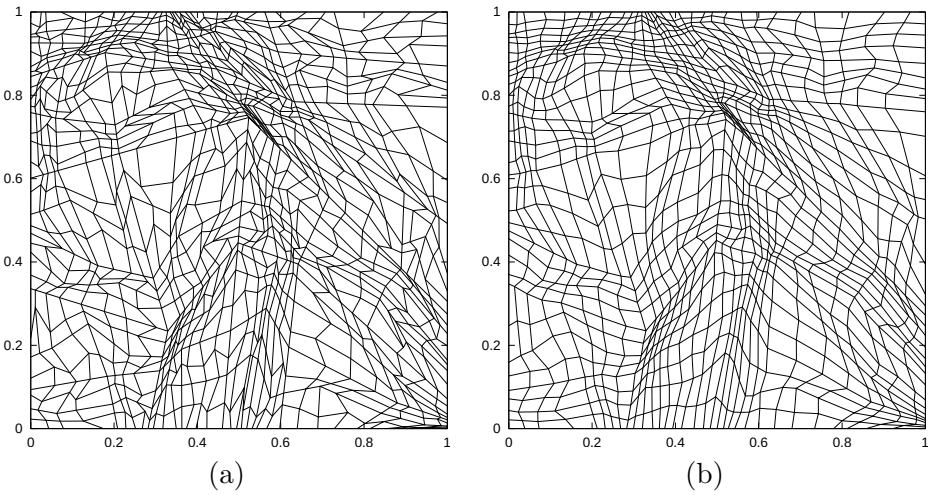


Figure 3.1: (a) Old (Lagrangian) mesh and (b) new mesh obtained from the old one with the Winslow rezoning algorithm.

other and contain the same features. However, the rezoned mesh is even visually smoother and more suitable for the following Lagrangian step.

We assume, that (as a result of the Lagrangian step) we know the geometry of the old and new meshes, and all Lagrangian fluid quantities inside the old cells as well as the nodal velocities. In the multi-material case, the material quantities need to be known also, in particular the knowledge of material volume fractions, eventually material centroids, is necessary. As a result, we want to compute the same quantities on the new mesh so that the remapping process satisfies the following properties:

- accuracy – at least second-order of accuracy is necessary to avoid excessive diffusion of the solution;
- continuity – if the computational mesh does not change, no quantity is supposed to change;
- conservation – fluid mass, volume, and internal energy have to be conserved (per material in case of multi-material remap);
- conservation of nodal quantities – nodal mass, momenta, and total energy have to be conserved;
- efficiency – all geometrical calculations are done only once, at the beginning of the remapping process.

In the following sections, the basic numerical approaches for remapping are summarized, with a special focus on multi-material remap in the context of multi-material ALE algorithms.

3.1 Function reconstruction

In order to achieve second-order accurate remapping scheme, function reconstruction must be performed to approximate the unknown function profile from the discrete data. Let us assume that there exist an unknown density function $\rho(x, y)$, and mass of each computational cell c is obtained as

$$m_c = \int_c \rho(x, y)\, dx\, dy \tag{3.1}$$

or

$$m_c = I_c^{\rho(x,y)}\,, \tag{3.2}$$

where the I symbol is used for denoting an integral of an arbitrary function over an arbitrary polygon. From the cell geometry, its volume can be computed as $V_c = I_c^1$, and the density mean value in cell c is than $\rho_c = m_c/V_c$ in the integral sense.

To approximate the density function in the mesh cells, a piece-wise linear function in the form

$$\rho_c(x, y) = \rho_c + \left(\frac{\partial \rho}{\partial x}\right)_c (x - x_c) + \left(\frac{\partial \rho}{\partial y}\right)_c (y - y_c) \tag{3.3}$$

is typically used, where $x_c = I_c^x / I_c^1$ and $y_c = I_c^y / I_c^1$ are the coordinates of the cell centroid. All necessary I integrals are just integrals of simple polynomials over a polygon and can be computed analytically [123]. For this construction, it can be simply shown that the definition of the cell mass (3.1) holds also for the approximate density function $\rho_c(x, y)$ and the method is self-consistent. The main task of the reconstruction method is the definition of the density slopes, $S_c^x = (\partial\rho/\partial x)_c$ and $S_c^y = (\partial\rho/\partial y)_c$ from the discrete density mean

values ρ_c. One popular possibility is their computation by minimization of an error functional, measuring the difference (in the least-squares sense) between the particular linear function evaluated in the centroids of the neighboring cells and the corresponding discrete mean values [157, 123]. This minimization process can be performed analytically, without the numerical calculation [123], and can be naturally extended to polygonal meshes [120] and to 3D [82]. To avoid problems with oscillations and overshoots, limiting is usually used, with the Barth-Jespersen limiter [28] being one of the most popular. Eventually, an a posteriori repair [125, 156, 140] can be applied, fixing the generated overshoots by a conservative mass redistribution.

3.2 Intersection-based remap

The intersection-based remap (reviewed in [155]) is the most natural approach for conservative transfer of conservative quantities between general computational meshes. Let us assume that we have a Lagrangian mesh containing cells $\{c\}$ and a rezoned mesh with cells $\{\tilde{c}\}$. For an example of a small Lagrangian and rezoned mesh, see Figure 3.2 (a) and (b). Both meshes in one picture are shown in Figure 3.2 (c). The central cell in the original mesh is denoted by the symbol c, and the tilde $\tilde{}$ symbol is used for the came cell in the new mesh.

The new cell $\tilde{c}$ can be composed from its intersections with all cells of the original mesh,

$$\tilde{c} = \bigcup_{c' \in \{c\}} \tilde{c} \cap c' \,, \tag{3.4}$$

as shown in Figure 3.2 (d), with the intersections shown by different colors. Following (3.1), remap of mass can be written in the form

$$m_{\tilde{c}} = I_{\tilde{c}}^{\rho(x,y)} = I_{\cup_{c' \in \{c\}} \tilde{c} \cap c'}^{\rho(x,y)} = \sum_{c' \in \{c\}} I_{\tilde{c} \cap c'}^{\rho(x,y)} \approx \sum_{c' \in \{c\}} I_{\tilde{c} \cap c'}^{\rho_{c'}(x,y)} \,. \tag{3.5}$$

This approach satisfies all our requirements except efficiency and is applicable for general arbitrary meshes. On the other hand, in case of same-connectivity meshes and small nodal motion during the rezoning step, such as shown in Figure 3.1, only neighboring cells can be employed instead of the entire mesh, which significantly reduces its computations cost.

This approach can be formulated in a flux form derived in [68, 155], which is based on adding and removing of pieces of the Lagrangian cell in order to obtain the rezoned one,

$$\tilde{c} = c \cup \bigcup_{c' \in C(c)} \Big(\big(\tilde{c} \cap c' \big) \setminus \big(c \cap \tilde{c}' \big) \Big) \,, \tag{3.6}$$

where $C(c)$ represents the set of cells neighboring with c (including the corner neighbors). This formula is demonstrated in Figure 3.2 (e), where different

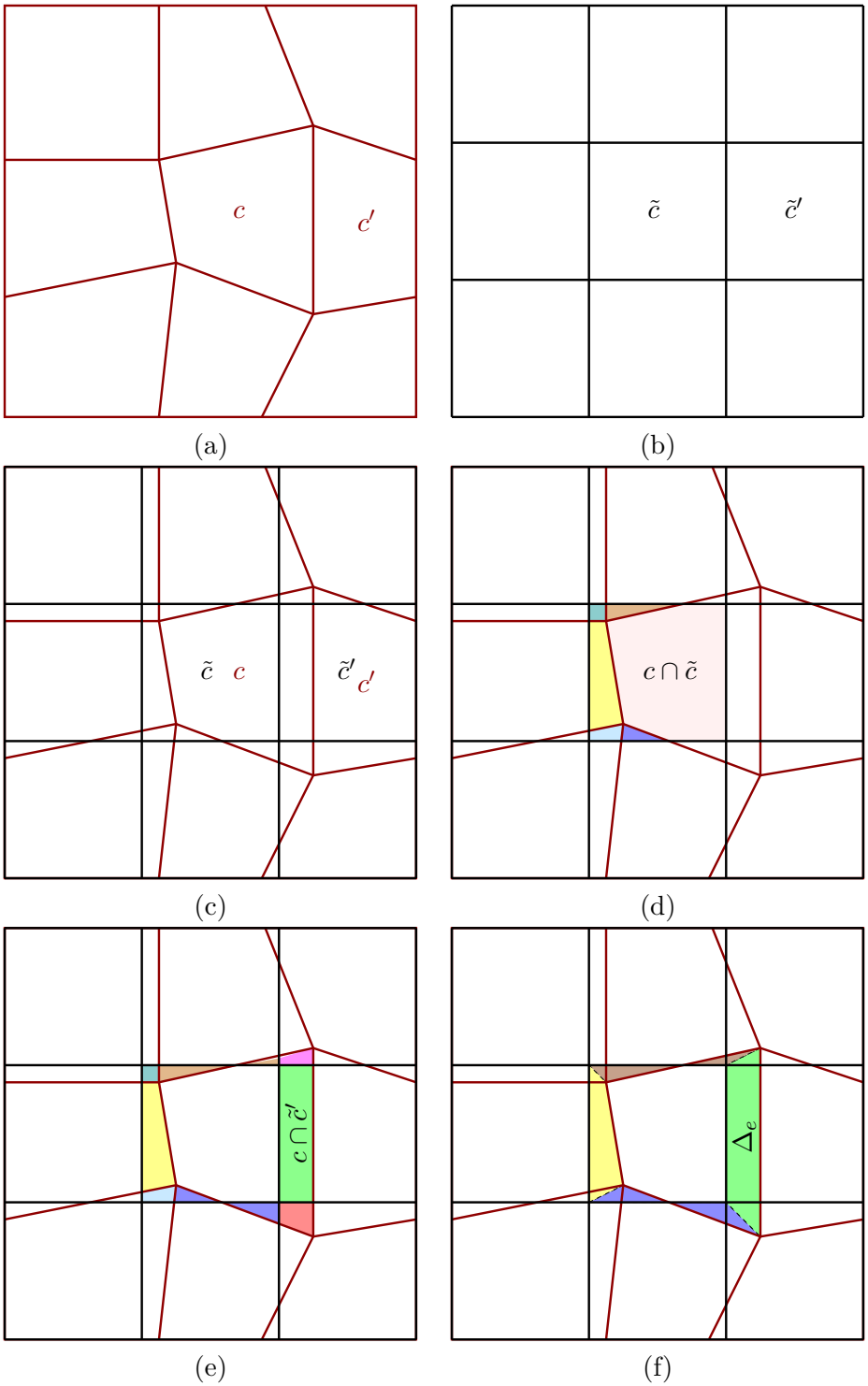


Figure 3.2: Comparison of intersection- and swept-based remapping methods: (a) Lagrangian mesh; (b) rezoned mesh; (c) both meshes over each other; (d) intersection-based remap; (e) intersection-based remap in a flux form; (f) swept-based remap.

colors show the intersections. The mass remap (3.5) can be formulated in an equivalent flux form

$$m_{\tilde{c}} = m_c + \sum_{c' \in C(c)} F^m_{c,c'} \,, \tag{3.7}$$

where the mass fluxes F^m are composed from their positive and negative contributions,

$$F^m_{c,c'} = I^{\rho_{c'}(x,y)}_{\tilde{c} \cap c'} - I^{\rho_c(x,y)}_{c \cap \tilde{c}'} \,. \tag{3.8}$$

These integrals represent the masses of all the intersections, and in practice, by using (3.3), they can be composed from the exchange integrals I^1, I^x, I^y, I^{xx}, I^{xy}, I^{yy} (i.e. integrals of simple polynomials up to the second order, pre-computed at the beginning of the remapping step from the geometry of the intersections) as

$$\begin{aligned} I^{\rho_{c'}(x,y)}_{\tilde{c} \cap c'} = \rho_{c'} \, I^1_{\tilde{c} \cap c'} &+ \left(\frac{\partial \rho}{\partial x}\right)_{c'} \left(I^x_{\tilde{c} \cap c'} - x_{c'} \, I^1_{\tilde{c} \cap c'}\right) \\ &+ \left(\frac{\partial \rho}{\partial y}\right)_{c'} \left(I^y_{\tilde{c} \cap c'} - y_{c'} \, I^1_{\tilde{c} \cap c'}\right) \,. \end{aligned} \tag{3.9}$$

Similar formula can be used for other quantities, for examples of nodal momenta or cell internal energy, see [123, 121].

3.3 Swept-region-based remap

The notion of swept regions was described and used for example in [54, 68, 155, 125]. A swept region Δ_e is defined by the motion of a particular edge e between cells c and c' into its new position $\tilde{e}$ during the rezoning phase of the ALE algorithm. There always exist one quadrilateral swept region corresponding to each cell edge, no explicit corner fluxes are involved in this method, see Figure 3.2 (f).

Remap of cell mass can be written as

$$m_{\tilde{c}} = m_c + \sum_{e \in E(c)} F^m_e \,, \tag{3.10}$$

where $E(c)$ stands for a set of all edges of cell c, and where the mass fluxes are computed as integrals of the reconstructed density over the swept regions Δ_e,

$$F^m_e = \int_{\Delta_e} \rho_{c^*}(x,y) \, dx \, dy \,. \tag{3.11}$$

The reconstruction is taken from the cell c^*, which is either the original cell c, or its neighbor c' over the edge e, depending on the sign of algebraic area of the swept region,

$$c^* = \begin{cases} c & \text{if} \quad I^1_{\Delta_e} < 0 \\ c' & \text{if} \quad I^1_{\Delta_e} > 0 \end{cases} \,. \tag{3.12}$$

The integration can be again performed by composing the swept mass F_e^m from the pre-computed integrals $I^1_{\Delta_e}$, $I^x_{\Delta_e}$, and $I^y_{\Delta_e}$.

As no intersections are used in the remapper, the swept region approach is significantly faster than the intersection-based approaches. Therefore, the swept region methods are more popular in real ALE codes, allowing to get the problem solution in a shorter time. The swept region approach can be naturally extended to 3D [82], and even for certain special cases of meshes with changing connectivity [120] (included as Appendix 4.1).

3.4 Multi-material remap

In multi-material ALE, more than one material is permitted in each computational cell. This approach allows to treat materials properly, without their artificial mixing, and leads to a significant improvement of the simulation reliability. Many authors have been working on different aspects of multi-material ALE in recent years [165, 8, 195, 112, 26, 173, 115, 80, 81, 172, 44, 76, 77, 27].

Although there exist methods which do not require an explicit knowledge of the material location in the computational cells (such as the method of concentrations [81]), in most modern approaches the polygons of pure materials in the cell need to be reconstructed using a VOF [92] or MOF [73] material-reconstruction method. For a comparison of most popular material-reconstruction methods, see [115] (included as Appendix 4.3). Let us denote the polygon of material k in cell c by the symbol c_k. The volume of the material $V_{c,k}$ can be normalized and represented in the form of material volume fractions $\alpha_{c,k} = V_{c,k}/V_c$, its centroid $x_{c,k} = I^x_{c_k}/I^1_{c_k}$, $y_{c,k} = I^y_{c_k}/I^1_{c_k}$ is typically used to represent the approximate position in the cell. The evolution of these material quantities in the Lagrangian stage is performed by the multi-material closure model [97] and need to be remapped together with the fluid quantities during the remapping step. In the staggered discretization, the cell-centered quantities are treated separately for each material, while the nodal velocity is joint for all materials.

From the point of view of the basic remapping methods, intersection-based remap is used almost exclusively for multi-material problems. Its generalization for multiple materials is straightforward – intersections of the new cell must be performed with the pure material polygons instead of the whole original cell, as shown in Figure 3.3 (a), (b). For example, the mass flux between the central cell c and the right neighboring cell c' is split in two materials. The material mass remap can be then formulated in the form

$$m_{\tilde{c},k} = m_{c,k} + \sum_{c' \in C(c)} F^m_{c,c',k}\,, \tag{3.13}$$

and the material mass fluxes are

$$F^m_{c,c',k} = I^{\rho_{c',k}(x,y)}_{\tilde{c}\cap c'_k} - I^{\rho_{c,k}(x,y)}_{c_k \cap \tilde{c}'}\,. \tag{3.14}$$

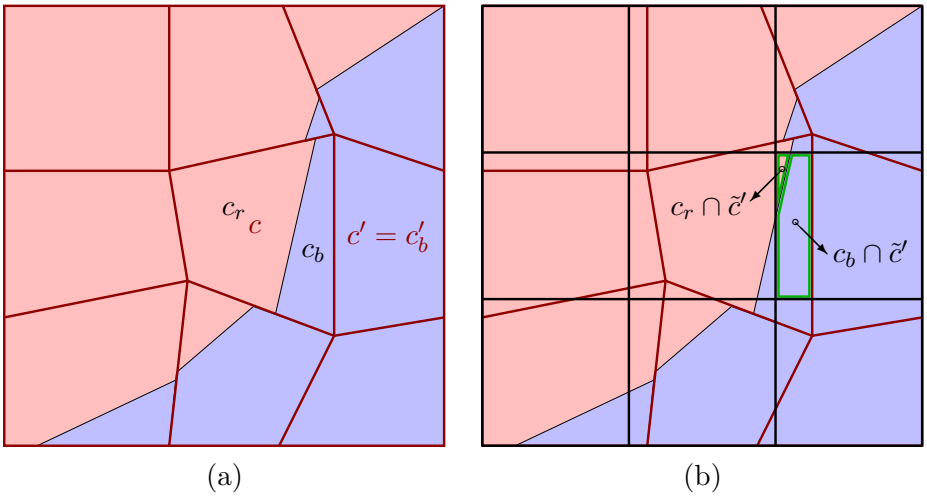


Figure 3.3: Multi-material remap: (a) two materials (red r and blue b) in the Lagrangian mesh; (b) flux between cells c and c' split in two materials (green polygons).

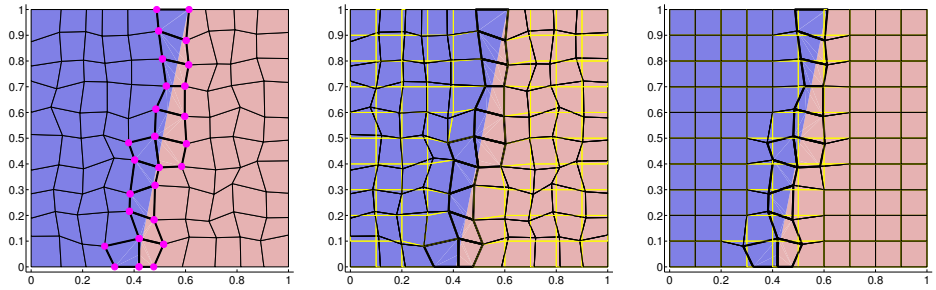
The density reconstructions $\rho_{c,k}(x, y)$ are performed in the same piece-wise linear manner as (3.3), for each material separately.

Let us note that there exist approaches trying to avoid expensive intersections by adapting the swept-based remap for multiple materials. As far as we know, all these approaches suffer from inconsistencies, and necessary approximate distribution of mass flux to materials generally leads to negative masses, density oscillations, or material fragmentation.

3.5 Hybrid remap

One possible approach for efficiency improvement of multi-material remapping is the hybrid remapping concept introduced in a series of papers [114, 34, 122]. In this approach, both intersection- and swept-based methods are combined in such a way, that the expensive intersections are used only in the vicinity of material interfaces, while cheap swept-regions are employed inside pure material regions covering most of the computational domain. For the remapping method to be consistent, a special treatment compatible with both approaches needs to be performed at the buffer region, where both methods meet.

In [114] (included as Appendix 4.4), the basic concept of hybrid remapping method was introduced and situation for logically-rectangular meshes is analyzed. A typical situation is shown in Figure 3.4 (a), where a Lagrangian mesh containing two materials is shown, with mixed cells and mixed nodes highlighted. All fluxes of mixed cells (surrounded by thick edges), including the corner fluxes, are computed by intersections, while in pure regions, they



(a) (b) (c)

Figure 3.4: Hybrid remapping: (a) two materials in a Lagrangian mesh with highlighted mixed cells (bold) and mixed nodes (magenta); (b) treatment of pure regions by swept regions, new mesh shown by yellow edges; (c) treatment of material interfaces by intersections, new mesh shown by yellow edges.

are computed by swept regions. For edges in the buffer regions (thin edges connected to mixed nodes), swept regions are decomposed into triangles, treated separately.

To be able to use the hybrid remap in a general polygonal mesh, the whole process has been separated in two distinct steps in [34] (included as Appendix 4.5). In the first step shown in Figure 3.4 (b), only nodes belonging to single-material cells are moved during rezoning and remap is performed with fluxes computed by swept regions. In the second step shown in Figure 3.4 (c), the remaining mixed nodes are rezoned and remap is done with intersections.

In [122] (included as Appendix 4.6), a complex one-step approach has been developed, treating all fluxes in the same swept-like manner. Swept regions at material interfaces are intersected with pure material polygons to construct the material fluxes, with paying a special attention to non-convex swept regions by splitting them to triangles. This approach avoids certain symmetry problems arising from a different treatment of mixed and pure nodes in the two-step method.

The concept of hybrid remapping can be used also in the single-material case [103]. The main motivation is the accuracy of the remapping scheme, especially from the point of view of symmetry violations caused by the swept-based remapping scheme, which was observed previously [47, 100]. Following preliminary works from [155] and [129], a full analysis of the local error of both methods has been performed in [105] (included as Appendix 4.9), identifying function properties and mesh motion patterns, for which is each method more accurate. Based on this analysis, several switches has been designed to switch between the methods in [104] (included as Appendix 4.10). This method allows to perform remapping efficiently while keeping low numerical error of the scheme and preserving function symmetry for non-conformal meshes.

3.6 Remap of all quantities

Up to now, remap of a single conservative quantity (fluid mass) has been considered. However, for the following Lagrangian solver, the complete set of fluid quantities has to be remapped conservatively in a consistent way. An extensive review of various possibilities for remapping of zonal quantities is included in the seminal paper [32].

In the cell-centered discretization, remapping is relatively simple. It is possible to compute the density of all cell-centered quantities, perform their reconstruction (3.3) and remapping (3.7) in the same manner [81, 80]. However, in the staggered discretization, which is practically used in many hydrodynamic codes, the situation is more complicated due to different location of various quantities in the mesh.

In [139], a novel approach for staggered remap has been developed, distributing all cell-based and nodal quantities to cell corners (subzones) and remapping all quantities on a double-fine mesh in the same manner as in the cell-centered discretization. In a series of papers [7, 8, 9, 10], a multi-material intersection-based remapping with VOF material reconstruction method has been described. Another multi-material remapping scheme with a similar spirit is presented in [141]. There also exist other intersection-based remapping schemes, one example is the approach of an explicit construction of a supermesh [75], containing all intersections in the form of separate cells. In [134], remapping of staggered quantities with a special focus on bound-preserving is described. A similar FCT-based approach has been developed in [37, 38].

In the preliminary work [121] and the full paper [123] (included as Appendix 4.7), a new approach for remapping of multi-material quantities in staggered discretization has been introduced. This scheme has been further studied with respect to its bound-preservation in [124]. This intersection-based approach employs the MOF material reconstruction method. All quantities are remapped in a flux form – material quantities (volume fractions and centroids) and material mass are remapped in the form (3.13). For the internal energy, a more complex flux structure must be used [68] to achieve consistency with mass remap. While material pressures are computed from the equation of state, a similar flux form has been suggested for remap of $(p\,V)_c$ in order to obtain new average cell pressure needed for the construction of the pressure forces in the following Lagrangian step.

A special attention is paid to remap of nodal quantities. Nodal mass is remapped in a similar flux-form

$$m_{\tilde{n}} = m_n + \sum_{n' \in N(n)} F^m_{n,n'}, \tag{3.15}$$

where the inter-nodal mass fluxes are constructed either by their interpolation from the inter-cell fluxes computed previously (3.14) as in [166], or a more

advanced minimization-based approach inspired by [164] is applied. Remap of the remaining nodal quantities (momentum components and kinetic energy) is performed in the same flux form, where the construction of the appropriate fluxes is performed by the attachment of the particular reconstructed quantity to the inter-nodal mass fluxes,

$$F^{\mu}_{n,n'} = u^*_{n,n'}\, F^m_{n,n'}\,, \tag{3.16}$$

$$F^{\nu}_{n,n'} = v^*_{n,n'}\, F^m_{n,n'}\,, \tag{3.17}$$

$$F^{K}_{n,n'} = \frac{1}{2}\left((u^*_{n,n'})^2 + (v^*_{n,n'})^2\right) F^m_{n,n'}\,, \tag{3.18}$$

where u^* and v^* are the reconstructions of the velocity components. These can be defined in different ways – low-order piece-wise constant values guaranteeing preservation of the local bounds, piece-wise linear reconstruction in a dual cell, bilinear interpolation in the particular primary cell, or combination of the previously mentioned approaches via the Flux-Corrected Transport (FCT) scheme [201, 127]. In [192] (included as Appendix 4.8) and [193], this FCT approach has been redesigned with a special focus on the symmetry of the resulting velocity field. It has been shown that the momentum remap in the presented flux form satisfies the DeBar consistency condition [58].

After the nodal remap, the kinetic energy is computed from the remapped nodal velocities,

$$\bar{K}_{\tilde{n}} = \frac{1}{2}\, m_{\tilde{n}} \left(u^2_{\tilde{n}} + v^2_{\tilde{n}}\right). \tag{3.19}$$

Due to the non-linear dependence of kinetic energy on velocity, conservation of energy is violated. This known problem is usually solved by the standard energy fix – a conservative kinetic energy $K_{\tilde{n}}$ is obtained by its remapping using the previously defined flux (3.18), and the obtained kinetic energy discrepancy $\delta K_{\tilde{n}} = K_{\tilde{n}} - \bar{K}_{\tilde{n}}$ is distributed to the internal energy of the adjacent computational cells. This approach fixes the energy conservation problem, however it can create disturbances in the internal energy field. To avoid this situation, a new approach inspired by [23] has been designed [21] (included as Appendix 4.2), creating the reconstructions of velocity $u^*_{n,n'}$, $v^*_{n,n'}$ in such a way that the kinetic energy discrepancy $\delta K_{\tilde{n}}$ is minimized.

This remapping approach is consistent, second-order accurate for all quantities, respects their continuity and conservation, and keeps their local bounds. Its efficiency is achieved by construction of all fluxes from the pre-computed exchange integrals. It has been demonstrated that it is applicable to a broad range of multi-material high-accuracy ALE simulations.

Chapter 4

Comments on Selected Publications

In the appendices of this thesis, ten selected representative publications of the author are included, all of them related to the topic of quantity remapping in the context of ALE methods. In this chapter, we briefly summarize the main results of each paper and the contributions of the author to the topic.

4.1 Paper 1: Extension of efficient, swept-integration-based conservative remapping method for meshes with changing connectivity

Reference:
M. Kucharik and M. Shashkov. Extension of efficient, swept-integration-based conservative remapping method for meshes with changing connectivity, *International Journal for Numerical Methods in Fluids*, **56**(8):1359-1365, 2008.
Attached as Appendix A.

Topic and content:
In [125], a new remapping method has been developed, combining the computationally efficient integration by swept regions with an a posteriori repair, fixing local-bound violations by a conservative mass redistribution. In this short paper, we have extended this method for meshes changing their connectivity in a Voronoi-like manner [12]. In case of Voronoi meshes with slightly different generators, mesh connectivity can change only in such a way that a particular edge disappears and is replaced by a different edge such that neighbors of an adjacent cell change. In this situation, we define a center (point) of reconnection and replace the swept-region flux approximation by a two-step procedure, shrinking imaginably the original edge to the center and extending it to the new edge. By doing this, the quadrilateral swept regions are replaced by two sets of triangles in a way consistent with the rest of the mesh, where no reconnections appear. This allows to perform quantity remapping in a computational efficient swept-like manner even for methods using dynamically changing meshes, such as for example the ReALE

method [137]. Properties of the new method have been demonstrated on a set of static-remapping tests.

Contribution of the author:

- Design of the method (together with M. Shashkov).
- Development of new remapping code based on the MSTK unstructured mesh environment [83].
- Implementation of the new changing-connectivity remapping method.
- Performing the numerical tests.
- Main and corresponding author.

4.2 Paper 2: Reduced-dissipation remapping of velocity in staggered arbitrary Lagrangian-Eulerian methods

Reference:
D. Bailey, M. Berndt, M. Kucharik, and M. Shashkov. Reduced-dissipation remapping of velocity in staggered arbitrary Lagrangian-Eulerian methods, *Journal of Computational and Applied Mathematics*, **233**(12):3148-3156, 2010. Attached as Appendix B.

Topic and content:
Remapping is a conservative process and conservation is exact for the primary remapped quantities, typically fluid mass, momentum, and internal energy. However, kinetic energy is a compound quantity involving fluid mass and velocity (in a non-linear way), so it is not generally conserved during the remapping process and consequently conservation of total energy is violated. This problem is typically treated by the standard energy fix [32] – kinetic energy is remapped separately and the discrepancy between remapped (conservative) and real (obtained from remapped velocity) kinetic energy is distributed (dissipated) to the internal energy of the neighboring cells. This approach is inspired by [23] and guarantees conservation, however, it can introduce disturbances of the internal energy field, especially for cold and fast moving fluids. To fix this problem, we have developed a velocity remapping algorithm, which tries to avoid this problem by a clever reconstruction of the velocity field during the momentum flux construction in such a way that the discrepancy is minimized. A functional describing the discrepancy as a function of velocity reconstruction in each flux has been designed and a global minimization problem is solved to achieve the smallest possible value. For the minimization, a global Jacobian-Free Newton-Krylov (JFNK) [167] solver has been employed. On selected numerical tests, it has been demonstrated that if the solution exist (zero energy discrepancy can be achieved), the algorithm finds it and no fix is needed at all. In the opposite case, the algorithm

converges to the global minimum, finding the velocity reconstruction in such a way that the discrepancy is as low as possible.

Contribution of the author:

- Generalization of the approach from [23] for staggered remapping (together with M. Shashkov and D. Bailey).
- Design of the modified functional for minimization (together with M. Shashkov).
- Design of the global minimization problem, incorporation of the JFNK minimization solver from the NITSOL [167] package (together with M. Berndt).
- Implementation of the new method in the context of the research RMALE code.
- Performing the numerical tests.
- Main and corresponding author.

4.3 Paper 3: A comparative study of interface reconstruction methods for multi-material ALE simulations

Reference:
M. Kucharik, R.V. Garimella, S.P. Schofield, and M.J. Shashkov. A comparative study of interface reconstruction methods for multi-material ALE simulations, *Journal of Computational Physics*, **229**(7):2432-2452, 2010. Attached as Appendix C.

Topic and content:
To perform multi-material ALE simulations, a multi-material remapping approach is needed, transferring the fluid quantity and the material information (volume fractions and material centroids) to the rezoned mesh. To avoid any approximations in material distribution (which can lead to possibly negative material masses, energies, etc.), an explicit knowledge of pure material polygons is necessary. These polygons are typically recovered by a particular material reconstruction method. There exist several types of methods for material reconstruction. In this study, we compare the standard Youngs' order-dependent VOF method [199] using several material orderings, an alternative VOF-PD method [176] based on weighted Voronoi mesh (power diagram) partitioning the cell, and the modern MOF reconstruction method [73] benefiting from the knowledge of material centroids. The comparison is performed on a broad range of typical multi-material hydrodynamics problems modeled in the staggered research RMALE code. It has been demonstrated, that the classical VOF method works remarkably

well when correct material ordering is used, however, the ordering cannot be simply determined for a general problem. The MOF method performs best, distinguishing correctly small interface features, but is computationally more expensive due to testing of all possible material orderings. The VOF-PD method does not resolve the interface very accurately, however, material topology is typically correct and the reconstruction is therefore better than for VOF with wrong ordering. Next to this comparison, the paper contains an appendix with a new algorithm for advancing of the material centroids during the staggered Lagrangian step, based on a bilinear parametrization of the nodal velocity field.

Contribution of the author:

- Design of numerical tests suitable for the comparison (together with all authors).
- Incorporation of the MOF, VOF (together with R. Garimella), and VOF-PD (together with S. Schofield and R. Garimella) reconstruction methods in the RMALE code.
- Adaptation of the quantity remapping algorithm for compatibility with multiple material reconstruction methods (with M. Shashkov).
- Development of algorithm for approximate motion of material centroids in the Lagrangian solver (together with S. Schofield).
- Running the hydrodynamic simulations.
- Analysis of the results, comparison of the methods (together with all authors).
- Main and corresponding author.

4.4 Paper 4: Hybrid remap for multi-material ALE

Reference:
M. Kucharik, J. Breil, S. Galera, P.-H. Maire, M. Berndt, and M. Shashkov. Hybrid remap for multi-Material ALE, *Computers & Fluids*, **46**(1):293-297, 2011.
Attached as Appendix D.

Topic and content:
For remapping, two basic approaches can be used. The approach of approximate fluxes based on swept regions [68, 155, 156, 125] is commonly used in many codes, however, its generalization for multi-material problems is complicated and one typically needs to switch to full intersections and exact integration [67, 69, 86] of the reconstructed quantities in order to remap the material data properly. While the swept-based approach is computationally very efficient, the intersection-based method can become rather expensive in

2D and almost impossible in 3D. This paper is the first from the series of three papers, in which a new concept of hybrid remapping is introduced. In this approach, both swept- and intersection-based approaches are combined in such a way that the expensive intersections need to be performed at the material interfaces only, while the simple and fast swept regions can be used in most of the domain containing pure materials. The difficult part is how to interconnect flux construction for both types of methods in a compatible way. In the buffer region where both types of fluxes meet, a special treatment is required due to the existence of the corner fluxes in the intersection-based method, while they are missing in the swept region approach. In this paper, the situation in case of a logically-rectangular computational mesh is analyzed and joining of the methods is achieved by the decomposition of the flux regions into a set of triangles in the buffer region between both methods, which can be treated uniformly in a single step. On selected numerical tests it has been shown, that the new approach compatibly connects both methods and global linear function is remapped exactly, while for other functions, the numerical error is comparable to the original methods. It has been demonstrated that the computational cost of the new method is close to the cost of the swept-based approach, while the ability of the intersection-based approach of handling multi-material situations is preserved.

Contribution of the author:

- Design of the new hybrid remapping method (together with M. Berndt and M. Shashkov), its implementation in the RMALE code.
- Adaptation of the main logic of the staggered RMALE code to handle the hybrid remapping concept.
- Comparison with results from a cell-centered code CHIC (with J. Breil, S. Galera, and P.-H. Maire).
- Performing the numerical tests.
- Main and corresponding author.

4.5 Paper 5: Two-step hybrid conservative remapping for multimaterial arbitrary Lagrangian-Eulerian methods

Reference:
M. Berndt, J. Breil, S. Galera, M. Kucharik, P.-H. Maire, and M. Shashkov. Two-step hybrid conservative remapping for multimaterial arbitrary Lagrangian-Eulerian methods, *Journal of Computational Physics*, **230**(17):6664-6687, 2011.
Attached as Appendix E.

Topic and content:
This paper is the second paper on the hybrid remapping concept. It describes a new two-step approach, treating mixed and pure-material regions consequently. In the first step, only nodes belonging to single-material cells are moved during rezoning and computationally cheap swept-regions are used for quantity remapping. As no mixed cell is affected by the rezoner, there is no problem with material interfaces during the swept-based remap. In the second step, the rezoner fixes the rest of the nodes, typically along material interfaces, and the expensive intersection-based remap is used, which is able to handle the intersections naturally. A small additional computational cost is necessary because the cells in the buffer zone between pure and mixed nodes are treated twice. This approach has been implemented in the framework of two different multi-material ALE codes – the structured staggered RMALE code and the unstructured cell-centered CHIC code. The numerical examples then demonstrate performance of the new method in both codes for stand-alone cyclic remapping and also for realistic multi-material hydrodynamic tests. It is shown that the new method can dramatically decrease the computational cost of the multi-material remapping, however, this gain is strongly implementation-dependent. The accuracy of the new method is almost the same as for intersections.

Contribution of the author:

- Design of the two-step hybrid remapping method (together with M. Berndt and M. Shashkov).
- Implementation of the new method in the RMALE code.
- Adaptation of the main logic of the RMALE code for the hybrid remap.
- Performing the numerical tests in the structured staggered RMALE code.
- Comparison of the performance of the method implemented in RMALE and CHIC codes (together with M. Shashkov, J. Breil, S. Galera, and P.-H. Maire).
- Analysis of the numerical results (together with all authors).

4.6 Paper 6: One-step hybrid remapping algorithm for multi-material arbitrary Lagrangian-Eulerian methods

Reference:
M. Kucharik and M. Shashkov. One-step hybrid remapping algorithm for multi-material arbitrary Lagrangian-Eulerian methods, *Journal of Computational Physics*, **231**(7):2851-2864, 2012.
Attached as Appendix F.

Topic and content:
This paper is the third paper on the hybrid remapping concept. A new one-step hybrid remapping approach is developed here, applicable to general polygonal meshes (unlike the one-step approach described in Section 4.4) and avoiding symmetry problems arising from a different treatment of mixed and pure mesh nodes in the two-step method described in Section 4.5. The new method explicitly combines the swept fluxes in pure material regions with the intersection-based fluxes close to the material interfaces in a single step. The swept regions are constructed even for multi-material cells and the corresponding multi-material fluxes are constructed by intersecting the swept regions with the material polygons of the involved cells. In case of a self-overlapping (or non-convex) swept region, it is split in triangles which are then treated separately. The fluxes in pure-material and mixed regions are then consistent, no buffer region is needed, just the internal structure of the flux is different. The numerical tests show that the computational time is reduced significantly when compared with the pure intersections, and the new method is simple, easy to implement to an existing code, applicable for general polygonal meshes and preserves symmetry of the problem.

Contribution of the author:

- Design of the new one-step hybrid remapping method (together with M. Shashkov).
- Implementation of the new method in the context of a research remapping code.
- Performing the numerical tests.
- Analysis of the results, comparison with previously developed hybrid methods (together with M. Shashkov).
- Main and corresponding author.

4.7 Paper 7: Conservative multi-material remap for staggered multi-material arbitrary Lagrangian-Eulerian methods

Reference:
M. Kucharik and M. Shashkov. Conservative multi-material remap for staggered multi-material arbitrary Lagrangian-Eulerian methods, *Journal of Computational Physics*, **258**:268-304, 2014.
Attached as Appendix G.

Topic and content:
This detailed paper summarizes the long-term work on the development of a full multi-material remapping method for all fluid quantities in the staggered discretization. It is the most significant paper from the list, containing

most of the work of the author in the field of quantity remapping performed from 2006 to 2014. In this paper, we present a new remapping approach in the framework of 2D staggered multi-material ALE on logically rectangular meshes. It is based on the computation of the exchange integrals (using intersections/overlays), which are integrals of simple polynomials (up to the second degree) over the intersections. Fluxes to all neighboring cells (including the corner neighbors) of all quantities are composed from these exchange integrals. Fluid mass is then remapped in a flux form. Internal energy is remapped similarly, with the energy fluxes constructed by integration of energy density in the flux regions [68]. To achieve continuity, a method for remap of common cell pressure has been developed, with the pressure-volume fluxes constructed in a similar flux form and composed from the exchange integrals. Similarly, material quantities (volume fractions, centroids) are remapped as integral quantities, allowing to compute volume fractions and approximate positions of the materials in the new mesh, which are later used for the MOF material reconstruction [73]. We pay a special attention to the remap of nodal quantities, performed also in a flux form. An optimization-based approach [164] is used for the construction of the nodal mass fluxes. The flux-corrected remap (FCR) approach [127] for the flux limiting is employed for the nodal velocity remap, which enforces bound preservation of the remapped constructed velocity field. Nodal kinetic energy is remapped similarly and used for the standard kinetic energy fix [32]. Properties of the new remapping algorithm are demonstrated on a suite of realistic hydrodynamic examples. In appendices, several practical issues are presented: formulas for the computation of the exchange integrals, a robust algorithm for intersections of close polygons, and demonstration of consistency of the internal energy remap.

Contribution of the author:

- Primary developer of the RMALE code, implementation of all necessary multi-material methods (MM Lagrangian solver with several closure models, several material reconstruction methods, adaptation of an hourglass-control mechanism for MM case, advancing of material centroids in the Lagrangian phase, etc).
- Design of the new flux-based multi-material remapping method (together with M. Shashkov) and its implementation in the RMALE code, in particular:
  - Robust intersection approach based on halfplane intersections, employing bisection for close-to-parallel edges.
  - Construction of exchange integrals by exact integration of simple polynomials over the intersections.
  - Construction of inter-cell fluxes, remap of mass, internal energy, pressure, and material quantities in a flux form.
  - Construction of inter-nodal mass fluxes by local optimization, remap of nodal mass.

  - Construction of momentum and kinetic energy fluxes by attaching the reconstructions to mass fluxes, remap of momentum and kinetic energy.
  - Energy fix by local distribution of kinetic energy discrepancy to internal energy.

- Performing the numerical tests.
- Analysis of the results (together with M. Shashkov).
- Main and corresponding author.

4.8 Paper 8: Symmetry- and essentially-bound-preserving flux-corrected remapping of momentum in staggered ALE hydrodynamics

Reference:
J. Velechovsky, M. Kucharik, R. Liska, M. Shashkov, and P. Vachal. Symmetry- and essentially-bound-preserving flux-corrected remapping of momentum in staggered ALE hydrodynamics, *Journal of Computational Physics*, **255**:590-611, 2013.
Attached as Appendix H.

Topic and content:
Remapping of nodal velocity is the most delicate part of staggered remapping algorithms. In this paper, we focus on symmetry of the remapped velocity field next to its accuracy, conservation, and consistency (in the deBar sense [58]), which is a crucial property in certain physical applications, such as the Inertial Confinement Fusion (ICF) simulations. To achieve bound preservation and avoid generation of new local extrema, some sort of vector limiting must be performed. In this work, we adapt the Flux Corrected Transport [127] approach for symmetric velocity fields to achieve preservation of coordinate invariant local bounds, while the radial symmetry of the flow is preserved as well. Proof of symmetry of the new method is provided. The properties of the method are demonstrated on a set of static remapping and full hydrodynamic tests.

Contribution of the author:

- Participation on the design of the method (together with all authors).
- Implementation of the method in the RMALE hydrodynamic code.
- Performing the hydrodynamic simulations, analysis of the results.

4.9 Paper 9: Local error analysis and comparison of the swept- and intersection-based remapping methods

Reference:
M. Klima, M. Kucharik, and M. Shashkov. Local error analysis and comparison of the swept- and intersection-based remapping methods, *Communications in Computational Physics*, **21**(2):526-558, 2017.
Attached as Appendix I.

Topic and content:
For remapping, two basic classes of methods are used, which are methods based on intersections and on swept regions. Both approaches are known to be second order accurate and in practice, the swept approach is typically preferred due to its lower computational cost. However, it has been observed that this method can perform worse in some tests in terms of solution symmetry [47, 100]. Following preliminary works from [155] and [129], a full analysis of the local error of both methods is performed in a general case and for several typical mesh motion patterns, such as cell-corner movement, hourglass edge rotation, shear cell deformation, and cell rotation. It is shown, that each method can be more suitable for different mesh motion patterns and different quantity profiles, especially depending on its second derivatives. The validity of the analysis and properties of both approaches are demonstrated on a set of cyclic remapping and hydrodynamic numerical tests.

Contribution of the author:

- Supervising Matěj Klíma (Ph.D. student and the main author).
- Participation on the analysis of the remapping methods (together with all authors).
- Analysis of the results, comparison of both methods (together with M. Klíma).
- Corresponding author.

4.10 Paper 10: Combined swept region and intersection-based single-material remapping method

Reference:
M. Klima, M. Kucharik, and M. Shashkov. Combined swept region and intersection-based single-material remapping method, *International Journal for Numerical Methods in Fluids*, 2017. In press.
Attached as Appendix J.

Topic and content:
In the hybrid remapping concept [114, 34, 122] (attached in Appendices D-F), the exact and swept remapping approaches are used in different parts of the computational domain, depending on the presence of material interfaces. In the previous paper [105] (attached as Appendix I), a local-error analysis of both approaches has been conducted, identifying, in which region is each method more accurate for a particular discrete function. This analysis has been used for the construction of several switching functions, combining the remapping methods in a two-step hybrid way [34], such that each method is used when it is more suitable (depending on function profile and mesh motion pattern). We call this approach pseudo-hybrid or combined remap. Simple gradient and Hessian based switches are used to detect shocks/peaks and to demonstrate viability of the method. For minimization of the numerical error, the directional second derivative ratio switch has been designed, shown to be able to produce remapper with lower numerical error than each of the standard approaches. The diagonal second derivative switch emphasizes symmetry of the solution, and is demonstrated to preserve symmetry similarly to intersection-based remap, while keeping computational cost significantly lower. Behavior of the basic concept and all switches has been verified on both cyclic remapping and full hydrodynamic numerical tests.

Contribution of the author:

- Supervising Matěj Klíma (Ph.D. student, the main and corresponding author).
- Design of the basic concept of single-material pseudo-hybrid remapping.
- Participation on the construction of the switching functions (together with M. Klíma).
- Analysis of the results, comparison of the switches (together with M. Klíma).

4.11 Main contributions of the listed publications

Here, we summarize the main scientific contributions from the presented papers in the three main sub-categories:
Analysis and improvement of basic remapping approaches:

- Approximate swept-based remapping approach was extended to meshes with changing connectivity in a Voronoi-like manner.
- Rigorous error analysis of swept- and intersection-based remapping methods for different mesh motion patterns was performed.
- New remapping approach was designed, combining swept- and intersection-based approaches in different parts of the domain, minimizing the numerical error and computational cost, while preserving solution symmetry.

Development of hybrid remapping concept:

- Concept of hybrid remapping for logically rectangular meshes was developed.
- Hybrid remapping was adapted to a two-step procedure, its applicability was demonstrated in a staggered and a cell-centered hydrodynamic ALE codes.
- Unification of the hybrid remapping concept into a swept-intersection-based one-step algorithm was performed, eliminating drawbacks of previous hybrid methods.

Contributions in multi-material remapping:

- New approach for velocity reconstruction was designed, taking into account kinetic energy conservation and therefore reducing dissipation.
- Comparison of most popular material reconstruction methods was performed during remap in the context of multi-material ALE hydrodynamic simulations.
- A novel flux-based remapping algorithm for all fluid quantities was developed in the staggered ALE framework.
- A new FCT-based approach for limiting of velocity field with a special focus on its symmetry was developed.

All these contributions improve the existing methods or introduce new methods for remapping of staggered fluid quantities in the context of multi-material ALE codes. As a result, the accuracy, reliability, efficiency, and robustness of the realistic physical simulations has been significantly improved.

Chapter 5

Applications in Laser-Plasma Hydrodynamics

The research on ALE methods at the Department of Physical Electronics resulted in the development of the PALE (Prague ALE) hydrodynamic code, intended primarily for the simulations of laser/plasma interactions. This code solves the Euler equations (2.1)-(2.3) in the modified form

$$\frac{1}{\rho}\,\dot{\rho} + \nabla \cdot \vec{u} = 0\,, \tag{5.1}$$

$$\rho\,\dot{\vec{u}} + \nabla p = \vec{0}\,, \tag{5.2}$$

$$\rho\,\dot{\varepsilon} + p\,\nabla \cdot \vec{u} = \nabla \cdot (\kappa\,\nabla T) - \nabla \cdot \vec{I}\,, \tag{5.3}$$

where the first term on the right hand side of the energy equation represents heat conductivity, and the second term is responsible for laser absorption. Here, κ is the thermal conductivity coefficient, T is fluid temperature, and $\vec{I}$ is the laser beam energy flux (Poynting vector). The basic hydrodynamic model has been described in [116], a simple laser absorption and heat conductivity models were presented in [131]. The main part of the PALE code has been developed during the disertation of the author [113] and is under the continuing development until now. Many of the described numerical methods have been incorporated in the PALE code, increasing its accuracy, robustness, and efficiency. It is routinely used for simulations of experiments performed at the PALS laser facility [95], for a set of representative examples, see for example [121]. Here, we briefly overview several main types of hydrodynamic laser/plasma simulations performed by the PALE code.

In a series of papers [42, 41, 87, 96], results of experiments performed at the PALS laser facility have been described. In these experiments, laser interaction with a double target in different configurations has been investigated. An intense laser beam (hundreds of J) irradiates a thin Aluminum disc, which is evaporated and ablatively accelerated up to very high velocity (tens to hundreds of km/s). Such projectile impacts a massive metal target, generates a shock wave melting and evaporating the material, and results in the development of a crater. This process has been modeled in two phases [113, 133, 117]. First, the process of laser absorption in the flyer and its acceleration has

been simulated in a pure Lagrangian way. The obtained parameters profile of density, temperature, and velocity has been used for the construction of the initial data for the second phase – the impact simulation. Due to strong shears in the material, such simulation has to be performed with the full ALE machinery. As a result, impacting velocities and resulting crater depths and diameters have been measured and compared with the experimental data, confirming validity of the models and demonstrating reliability of the hydrodynamic predictions.

Study of laser interactions with different types of targets is performed in [133, 132]. Next to the double targets discussed previously, interactions with oblique foils, double foils, and foams, were studied. It has been demonstrated that for the oblique foil targets, the produced plasma moves in the direction perpendicular to the target, which has been observed experimentally before. For the double foil target, burn through one foil is modeled and collision of interacting plasma plumes resulting in high density/pressure region development is investigated. Finally, the speed of burn through the foam targets (using structured and uniform foam models) has been investigated, showing good agreement between the structured model and experimental measurements.

Another suite of simulations has been performed in the field of strong shock generation using the laser-induced cavity pressure acceleration (LICPA) scheme [14]. This scheme resembles the classical cannon ball setup – the laser beam enters small cavity, evaporates material of a small projectile, which is ablatively accelerated and hits the massive target. Due to the cavity, only small portion of the laser beam energy is allowed to escape and its most significant part is converted into the energy of the moving projectile and eventually into the energy of the spreading shock wave after the impact. Efficiency of this scheme is significantly higher than that of the standard ablative acceleration. In [16, 15], the basic simulations of projectile acceleration and its impact on the massive target have been performed and the results were compared to the experimental data. In [19], mostly the acceleration phase has been studied in detail and the structure of the impacting heavy macroparticle was described. This study was further extended in [20], where the situation at the moment of the impact and shock wave formation was examined. The later stages of the shock wave formation and crater development were explored in [13] and the evolution of crates for a long period of time in [17]. Finally in [18], the whole process of hydrodynamic simulations is reviewed and the results are presented for all phases of the problem. It has been demonstrated that the results of the hydrodynamic simulations reproduce well the experimental data and help to understand behavior inside the problem, which is difficult or impossible to be measured with the current detection devices. For example, the pressure at the shock wave was estimated from the numerical data, which is a crucial parameter for the inertial fusion ignition.

The formation of the shock wave and its parameters have been further

studied in [111, 142]. The effect of fast electrons on the generated shock wave formation have been studied experimentally and numerically. The laser-plasma coupling in a shock ignition relevant regime was investigated, with a strong shock generated by a secondary laser beam at various delays with respect to the first laser beam. It has been shown by the numerical simulation that the shock pressure is lower than expected and the 2D effects play a major role in these experiments.

A series of experiments has been described and supported by the hydrodynamic and particle-in-cell numerical simulations in [169, 150, 151]. Various types of Silicone targets with Boron dopants have been used and target optimization was performed by the hydrodynamic simulation – Boron-implanted layer in a Silicone substrate, diffused Boron in the Silicone substrate, and a Boron layer deposited on the surface of the Silicone substrate. The ultra-clean proton-Boron fusion reaction producing energetic α-particles was studied. The first target configuration was shown to have the highest probability of inducing the nuclear reaction. These highly cited results are expected to significantly influence the future applications, such as nuclear fusion without production of neutron-induced radioactivity.

Especially demanding hydrodynamic simulations have been performed in order to model the interaction of an intense laser beam with a cryogenic solid-Hydrogen target in [152], producing a collimated stream of accelerated photons, carrying a very high energy and charge. Very low temperatures (about 10 K) of the target material required significant improvements in the EOS model, the Lagrangian solver, and the accurate limiting in the remapping phase of the ALE algorithm. It has been demonstrated that near-critical plasma is created in the target, continuously irradiated by the laser beam, which suggests a potential enhancement of the laser absorption at the highest intensities. The experimental results delivered at the PALS laser facility and the numerical simulations are presented and discussed along with potential multidisciplinary applications.

The last application described in [168] is mainly related to the fabrication of microscopic targets for laser-target interaction experiments. To achieve the requested plasma density and energy profiles, a series of hydrodynamic simulations have been performed. A comparison with experimental data is presented, showing its relatively good agreement with the numerical model.

To summarize, selected developed numerical methods have been implemented and incorporated in the in-house PALE hydrodynamic code. In the publications describing the particular methods, static tests or simple fluid simulations have been presented, demonstrating their basic numerical properties. Over the years, several types of complex hydrodynamic simulations have been performed, involving additional physics models (laser absorption, realistic equation of state, heat conductivity, separate treatment of electron and ion temperature, etc.) as well as certain features, which are very difficult to model by the standard ALE framework (high aspect-ratio cell, close-to-zero densities, energies and temperatures, extremely high velocities, large

gradients, vortexes, or shear flows). The developed methods have shown their viability in such difficult situations, and their applicability in a laser/plasma hydrodynamic code. The robustness, accuracy, efficiency, and reliability of the new methods allowed the accomplishment of state of the art simulations from the field of extreme laser/plasma physic.

Chapter 6

Conclusions

In this thesis, the work of the author in the field of conservative interpolations (remapping) for indirect Arbitrary Lagrangian-Eulerian methods in recent years has been overviewed. The main contributions have been summarized in Section 4, where the content of 10 selected papers is briefly described. These papers are included as appendices of this thesis. The described numerical methods have been tested in the context of a research RMALE code and selected methods were implemented in the code PALE, which is under development in our research group, helping to improve its reliability, accuracy, robustness, stability, and efficiency. In Section 5, several papers co-authored by the author are referenced, showing particular simulations of laser/plasma interactions performed by the PALE code, employing the developed numerical methods. Additional simulations have been performed in the PALE code by other members of the Computational Physics Group, exploring different phenomena in laser generated plasma. Therefore, it has been demonstrated that the numerical methods are applicable in realistic calculations, leading to substantial improvements of the solution. At the same time, the mentioned simulations contributed to the research on the physics of laser/plasma interactions, mostly motivated by the study of inertial confinement fusion physics, which is a crucial and ambitious branch of physics leading to a potential development of a practically infinite and clean energy source.

6.1 Future work

There exist many unsolved problems in the theory of conservative interpolations, providing a large room for future research in this field.

The first topic is the problem of energy conservation due to the nonlinear dependence of the kinetic energy on the remapped velocity. This discrepancy is usually treated by the standard energy fix, distributing the discrepancy to the internal energy field in order to keep the conservation, which can, however, violate its smoothness for cold and fast fluids and result in negative internal energy and simulations breakdown. In [21], this issue

has been investigated systematically and a new method based on clever minimization-based construction of momentum fluxes has been developed, which helps significantly in decreasing the discrepancy but does not solve this problem completely. Especially important is the situation when the energy discrepancy is negative and needs to be subtracted from the internal energy, which contradicts the second law of thermodynamics. This issue needs to be investigated in detail and a new consistent, fully minimization-based remapping approach needs to be designed.

The second future topic is related to elastic/plastic simulations. The PALE simulation code has been extended from fluids to continuum dynamics of fluids and solids during recent years, allowing simulations of deforming materials in the Lagrangian manner. In order to be able to perform full ALE continuum simulations, the remap of deviatoric stress tensor (and the remap of tensor quantities in general) needs to be investigated. Particularly, limiting of tensor quantities is especially interesting, because the notion of monotonicity is not clear for tensors, and special internal tensor relations cannot be violated (tensor invariants need to be preserved, for example). Preliminary investigation in this direction has already been conducted [106], however, this topic needs a detailed research in order to design a consistent remapping approach for tensors.

The last topic mentioned here is remap in case of curvilinear meshes [153, 170, 63, 64]. This modern and fast evolving concept incorporates high-order computational cells, deforming with the fluid motion, and allowing significantly larger set of problems to be modeled in the Lagrangian setup. However, when switching from Lagrangian to the ALE framework, intersections and integration along curved edges need to be performed in order to remap the fluid quantities conservatively. Not much work has been done in this direction so far, and we expect an increasing demand for a curvilinear remap in the near future.

6.2 Collaboration with students

The scientific work at the university is inseparably connected with the education of students and a joint scientific collaboration. Many topics mentioned in this thesis have been investigated together with the students of the faculty.

A significant contribution has been brought by Ing. Matěj Klíma during the work on his diploma thesis and it continues now during his Ph.D. study. This work is related mainly to the analysis of remapping methods, development of the combined remapping strategy, and remapping of tensors. Work related to the material reconstruction methods has been done by Ing. Kateřina Deriánová during her diploma thesis, which contributed to the remap of multi-material quantities. Finally, Ing. Jan Nikl contributed mainly in applications of the remapping concept in laser/plasma simulations during his diploma thesis, and this collaboration still continues during his Ph.D. study.

Bibliography

[1] H. Abhold, M. Berndt, **M. Kucharik**, M. Shashkov, and B. Powers. Feasible set mesh untangling and smoothing. Technical Report LA-UR-08-1207, Los Alamos National Laboratory, 2008.

[2] H. T. Ahn and M. Shashkov. Multi-material interface reconstruction on generalized polyhedral meshes. *Journal of Computational Physics*, 226(2):2096–2132, 2007.

[3] H. T. Ahn and M. Shashkov. Adaptive moment-of-fluid method. *Journal of Computational Physics*, 228(8):2792–2821, 2009.

[4] N. Amenta, M. Bern, and D. Eppstein. Optimal point placement for mesh smoothing. *Journal of Algorithms*, 30(2):302–322, 1999.

[5] H. R. Anbarlooei and K. Mazaheri. Moment of fluid interface reconstruction method in multi-material arbitrary Lagrangian Eulerian (MMALE) algorithms. *Computer Methods in Applied Mechanics and Engineering*, 198(47-48):3782–3794, 2009.

[6] R. W. Anderson, N. S. Elliott, and R. B. Pember. An arbitrary Lagrangian-Eulerian method with adaptive mesh refinement for the solution of the Euler equations. *Journal of Computational Physics*, 199(2):598–617, 2004.

[7] P. Anninos. New VOF interface capturing and reconstruction algorithms. Technical Report UCRL-ID-135084, Lawrence Livermore National Laboratory, 1999.

[8] P. Anninos. Kull ALE: I. Unstructured mesh advection, interface capturing, and multiphase 2T RHD with material interfaces. Technical Report UCRL-ID-147297-PT-1, Lawrence Livermore National Laboratory, 2002.

[9] P. Anninos. Kull ALE: II. Grid motion on unstructured arbitrary polyhedral meshes. Technical Report UCRL-ID-147297-PT-2, Lawrence Livermore National Laboratory, 2002.

[10] P. Anninos. Multiphase advection and radiation diffusion with material

interfaces on unstructured meshes. Technical Report UCRL-JC-150129, Lawrence Livermore National Laboratory, 2002.

[11] E. Aulisa, S. Manservisi, and R. Scardovelli. A surface marker algorithm coupled to an area-preserving marker redistribution method for three dimensional interface tracking. *Journal of Computational Physics*, 197(2):555–584, 2004.

[12] F. Aurenhammer. Voronoi diagrams – a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, 1991.

[13] J. Badziak, L. Antonelli, F. Baffigi, D. Batani, T. Chodukowski, G. Cristoforetti, R. Dudzak, L. A. Gizzi, G. Folpini, F. Hall, Z. Kalinowska, P. Koester, E. Krousky, **M. Kucharik**, L. Labate, R. Liska, G. Malka, Y. Maheut, P. Parys, M. Pfeifer, T. Pisarczyk, O. Renner, M. Rosinski, L. Ryc, J. Skala, M. Smid, C. Spindloe, J. Ullschmied, and A. Zaras-Szydlowska. Studies of ablated plasma and shocks produced in a planar target by a sub-nanosecond laser pulse of intensity relevant to shock ignition. *Laser and Particle Beams*, 33(3):561–575, 2015.

[14] J. Badziak, S. Borodziuk, T. Pisarczyk, T. Chodukowski, K. Masek, J. Skala, J. Ullschmied, and Y.-J. Rhee. Highly efficient acceleration and collimation of high-density plasma using laser-induced cavity pressure. *Applied Physics Letters*, 96(25):251502, 2010.

[15] J. Badziak, S. Jablonski, T. Pisarczyk, T. Chodukowski, P. Parys, P. Raczka, M. Rosinski, E. Krousky, J. Ullschmied, R. Liska, **M. Kucharik**, and L. Torrisi. The LICPA accelerator of dense plasma and ion beams. *Journal of Physics: Conference Series*, 508(1):012006, 2014.

[16] J. Badziak, S. Jablonski, T. Pisarczyk, P. Raczka, E. Krousky, R. Liska, **M. Kucharik**, T. Chodukowski, Z. Kalinowska, P. Parys, M. Rosinski, S. Borodziuk, and J. Ullschmied. Highly efficient accelerator of dense matter using laser-induced cavity pressure acceleration. *Physics of Plasmas*, 19(5):053105, 2012.

[17] J. Badziak, E. Krousky, **M. Kucharik**, and R. Liska. The LICPA-driven collider – a novel efficient tool for the production of ultra-high pressures in condensed media. *Journal of Instrumentation*, 11(3):C03043, 2016.

[18] J. Badziak, **M. Kucharik**, and R. Liska. Production of sub-gigabar pressures by a hyper-velocity impact in the collider using laser-induced cavity pressure acceleration. *Laser and Particle Beams*, 2017. Accepted.

[19] J. Badziak, M. Rosinski, S. Jablonski, T. Pisarczyk, T. Chodukowski, P. Parys, P. Raczka, E. Krousky, J. Ullschmied, R. Liska, and **M. Kucharik**. Enhanced efficiency of plasma acceleration in the laser-induced cavity pressure acceleration scheme. *Plasma Physics and Controlled Fusion*, 57:014007, 2014.

[20] J. Badziak, M. Rosinski, E. Krousky, **M. Kucharik**, R. Liska, and J. Ullschmied. Generation of ultra-high-pressure shocks by collision of a fast plasma projectile driven in the laser-induced cavity pressure acceleration scheme with a solid target. *Physics of Plasmas*, 22(3):032709, 2015.

[21] D. Bailey, M. Berndt, **M. Kucharik**, and M. Shashkov. Reduced-dissipation remapping of velocity in staggered arbitrary Lagrangian-Eulerian methods. *Journal of Computational and Applied Mathematics*, 233(12):3148–3156, 2010.

[22] D. A. Bailey, P. K. Sweby, and P. Glaister. A ghost fluid, moving finite volume plus continuous remap method for compressible Euler flow. *International Journal for Numerical Methods in Fluids*, 47:833–840, 2005.

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

[24] A. Barlow, R. Hill, and M. Shashkov. Constrained optimization framework for interface-aware sub-scale dynamics closure model for multimaterial cells in Lagrangian and arbitrary Lagrangian-Eulerian hydrodynamics. *Journal of Computational Physics*, 276:92–135, 2014.

[25] A. J. Barlow. A new Lagrangian scheme for multi-material cells. In *Proceeding of European Congress on Computational Methods in Applied Sciences and Engineering - ECCOMAS Computational Fluid Dynamics Conference 2001, Swansea, Wales, UK, 4-7 September 2001*, 2001.

[26] A. J. Barlow. A compatible finite element multi-material ALE hydrodynamics algorithm. *International Journal for Numerical Methods in Fluids*, 56(8):953–964, 2008.

[27] A. J. Barlow, P.-H. Maire, W. J. Rider, R. N. Rieben, and M. J. Shashkov. Arbitrary Lagrangian-Eulerian methods for modeling high-speed compressible multimaterial flows. *Journal of Computational Physics*, 322:603–665, 2016.

[28] T. J. Barth and D. C. Jespersen. The design and application of upwind schemes on unstructured meshes. In *AIAA-89-0366*, 1989. 27th Aerospace Sciences Meeting, January 9-12, Reno, Nevada.

[29] R. Bell and E. Hertel. An improved material interface reconstruction algorithm for eulerian codes. Technical Report SAND 92-1716, Sandia National Laboratories, 1992.

[30] T. Belytschko, W. K. Liu, and M. O. Neal. Compact-impact algorithms for penetration studies. Technical report, Northwestern University, 1990. DAAL03-87-K-0035.

[31] T. Belytschko and J. S.-J. Ong. Hourglass control in linear and nonlinear problems. *Computer Methods in Applied Mechanics and Engineering*, 43(3):251–276, 1984.

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

[33] D. J. Benson. Volume of fluid interface reconstruction methods for multi-material problems. *Applied Mechanics Reviews*, 55(2):151–165, 2002.

[34] M. Berndt, J. Breil, S. Galera, **M. Kucharik**, P.-H. Maire, and M. Shashkov. Two-step hybrid conservative remapping for multimaterial arbitrary Lagrangian-Eulerian methods. *Journal of Computational Physics*, 230(17):6664–6687, 2011.

[35] M. Berndt, **M. Kucharik**, and M. Shashkov. Using the feasible set method for rezoning in ALE. *Procedia Computer Science*, 1(1):1879–1886, 2010.

[36] L. Bilbao. A three-dimensional Lagrangian method for fluid dynamics. *Journal of Computational Physics*, 91(2):361–380, 1990.

[37] P. Bochev, D. Ridzal, G. Scovazzi, and M. Shashkov. Formulation, analysis and numerical study of an optimization-based conservative interpolation (remap) of scalar fields for arbitrary Lagrangian-Eulerian methods. *Journal of Computational Physics*, 230(13):5199–5225, 2011.

[38] P. Bochev, D. Ridzal, and M. Shashkov. Fast optimization-based conservative remap of scalar fields through aggregate mass transfer. *Journal of Computational Physics*, 246:37–57, 2013.

[39] Yu. Bondarenko and Yu. Yanilkin. Computation of the thermodynamic parameters in the mixed cells in gas dynamics. *Mathematical Modeling*, 14(6):63–81, 2002.

[40] K. S. Bonnel, M. A. Ducheineau, D. R. Schikore, B. Hamann, and K. I. Joy. Material interface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 9(4):500–511, 2003.

[41] S. Borodziuk, A. Kasperczuk, T. Pisarczyk, M. Kalal, J. Ullschmied, J. Limpouch, K. Rohlena, N. N. Demchenko, S. Yu. Gus'kov, V. B. Rozanov, V. N. Kondrashov, and P. Pisarczyk. Application of the three frame interferometric and crater replica methods for investigation of a laser accelerated macroparticles – massive target interaction in the PALS experiment. *Optica Applicata*, 34(3):385–403, 2004.

[42] S. Borodziuk, A. Kasperczuk, T. Pisarczyk, K. Rohlena, J. Ullschmied, M. Kalal, J. Limpouch, and P. Pisarczyk. Application of laser simulation method for the analysis of crater formation experiment on PALS laser. *Czechoslovak Journal of Physics*, 53(9):799–810, 2003.

[43] W. Boscheri. High order direct Arbitrary-Lagrangian–Eulerian (ALE) finite volume schemes for hyperbolic systems on unstructured meshes. *Archives of Computational Methods in Engineering*, 2016. In press.

[44] J. Breil, S. Galera, and P.-H. Maire. Multi-material ALE computation in inertial confinement fusion code CHIC. *Computers & Fluids*, 46(1):161–167, 2011.

[45] J. Breil, S. Galera, and P.-H. Maire. A two-dimensional VOF interface reconstruction in a multi-material cell-centered ALE scheme. *International Journal for Numerical Methods in Fluids*, 65(11-12):1351–1364, 2011.

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

[47] D. E. Burton, M. A. Kenamond, N. R. Morgan, T. C. Carney, and M. J. Shashkov. An intersection based ALE scheme (xALE) for cell centered hydrodynamics (CCH). Talk at Multimat 2013, International Conference on Numerical Methods for Multi-Material Fluid Flows, San Francisco, September 2-6, 2013. LA-UR-13-26756.2.

[48] J. C. Campbell and M. J. Shashkov. A compatible Lagrangian hydrodynamics algorithm for unstructured grids. Technical Report LA-UR-00-3231, Los Alamos National Laboratory, 2000.

[49] E. J. Caramana. The implementation of slide lines as a combined force and velocity boundary condition. *Journal of Computational Physics*, 228(11):3911–3916, 2009.

[50] 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.

[51] E. J. Caramana, C. L. Rousculp, and D. E. Burton. A compatible, energy and symmetry preserving Lagrangian hydrodynamics algorithm in three-dimensional Cartesian geometry. *Journal of Computational Physics*, 157(1):89–119, 2000.

[52] E. J. Caramana and M. J. Shashkov. Elimination of artificial grid distortion and hourglass-type motions by means of Lagrangian subzonal masses and pressures. *Journal of Computational Physics*, 142(2):521–561, 1998.

[53] E. J. Caramana, M. J. Shashkov, and P. P. Whalen. Formulations of artificial viscosity for muti-dimensional shock wave computations. *Journal of Computational Physics*, 144(2):70–97, 1998.

[54] Ph. Colella. Multidimensional upwind methods for hyperbolic conservation laws. *Journal of Computational Physics*, 87(1):171–200, 1990.

[55] J. Costes, J.-M. Ghidaglia, and J. Breil. Mesh regularization for an ALE code based on the limitation of the Lagrangian mesh velocity. *International Journal for Numerical Methods in Fluids*, 2017. In press.

[56] R. Courant and K. O. Friedrichs. *Supersonic Flow and Shock Waves*, volume 21 of *Applied Mathematical Sciences*. Springer Verlag, 1999. Reprint of 1st ed. Interscience Publishers, New York 1948. ISBN 0-387-90232-5.

[57] D. de Niem, E. Kuhrt, and U. Motschmann. A volume-of-fluid method for simulation of compressible axisymmetric multi-material flow. *Computer Physics Communications*, 76:170–190, 2007.

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

[59] V. I. Delov and V. V. Sadchikov. Comparison of several models for computation of thermodynamical parameters for heterogeneous Lagrangian cells. *VANT, Mathematical Modeling of Physical Processes*, 1:57–70, 2005.

[60] B. Despres. Lagrangian systems of conservation laws and approximate Riemann solvers. In E. F. Toro, editor, *Godunov methods: Theory and Applications*, pages 233–245. Kluwer/Plenum, New York, 2001.

[61] B. Despres and E. Labourasse. Stabilization of cell-centered compressible Lagrangian methods using subzonal entropy. *Journal of Computational Physics*, 231(20):6559–6595, 2012.

[62] B. Despres and F. Lagoutiere. Numerical resolution of a two-component compressible fluid model with interfaces. *Progress in Computational Fluid Dynamics, An International Journal*, 7(6):295–310, 2007.

[63] V. A. Dobrev, T. E. Ellis, Tz. V. Kolev, and R. N. Rieben. Curvilinear finite elements for Lagrangian hydrodynamics. *International Journal for Numerical Methods in Fluids*, 65(11-12):1295–1310, 2011.

[64] V. A. Dobrev, Tz. V. Kolev, and R. N. Rieben. High-order curvilinear finite element methods for Lagrangian hydrodynamics. *SIAM Journal on Scientific Computing*, 34(5):B606–B641, 2012.

[65] 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.

[66] J. Donea, A. Huerta, J.-Ph. Ponthot, and A. Rodriguez-Ferran. Arbitrary Lagrangian-Eulerian methods. In E. Stein, R. de Borst, and T. Hughes, editors, *The Encyclopedia of Computational Mechanics*, chapter 14, pages 413–437. Wiley, 2004.

[67] J. K. Dukowicz. Conservative rezoning (remapping) for general quadrilateral meshes. *Journal of Computational Physics*, 54(3):411–424, 1984.

[68] J. K. Dukowicz and J. R. Baumgardner. Incremental remapping as a transport/advection algorithm. *Journal of Computational Physics*, 160(1):318–335, 2000.

[69] J. K. Dukowicz and J. W. Kodis. Accurate conservative remapping (rezoning) for arbitrary Lagrangian-Eulerian computations. *SIAM Journal of Scientific and Statistical Computing*, 8(3):305–321, 1987.

[70] J. K. Dukowicz and B. J. A. Meltz. Vorticity errors in multidimensional Lagrangian codes. *Journal of Computational Physics*, 99(1):115–134, 1992.

[71] V. Dyadechko, R. Garimella, and M. Shashkov. Reference Jacobian rezoning strategy for arbitrary Lagrangian-Eulerian methods on polyhedral grids. In *Proceedings of 13th International Meshing Roundtable*, pages 459–470, 2004.

[72] V. Dyadechko and M. Shashkov. Moment-of-fluid interface reconstruction. Technical Report LA-UR-05-7571, Los Alamos National Laboratory, 2005.

[73] V. Dyadechko and M. Shashkov. Reconstruction of multi-material interfaces from moment data. *Journal of Computational Physics*, 227(11):5361–5384, 2008.

[74] J. M. Escobar, E. Rodriguez, R. Montenegro, G. Montero, and J. M. Gonzalez-Yuste. Simultaneous untangling and smoothing of tetrahedral meshes. *Computer Methods in Applied Mechanics and Engineering*, 192(25):2775–2787, 2003.

[75] P. E. Farrell, M. D. Piggott, C. C. Pain, G. J. Gorman, and C. R. Wilson. Conservative interpolation between unstructured meshes via supermesh construction. *Computer Methods in Applied Mechanics and Engineering*, 198(33-36):2632–2642, 2009.

[76] C. Fochesato, R. Loubere, R. Motte, and J. Ovadia. Adaptive subdivision piecewise linear interface calculation (ASPLIC) for 2D multimaterial hydrodynamic simulation codes. *International Journal for Numerical Methods in Fluids*, 77(7):418–439, 2014.

[77] M. B. Friess, J. Breil, S. Galera, P.-H. Maire, and M. Shashkov. A multi-material CCALE-MOF approach in cylindrical geometry. *Communications in Computational Physics*, 15(2):330–364, 2014.

[78] J. Fung, M. Francois, E. Dendy, M. Kenamond, and R. Lowrie. Calculations of the Rayleigh-Taylor instability: RAGE and FLAG hydrocode comparisons. In *Proceedings of NECDC06*, 2006.

[79] E. Gaburro, M. Dumbser, and M. J. Castro. Direct Arbitrary-Lagrangian-Eulerian finite volume schemes on moving nonconforming unstructured meshes. Technical report, 2016. arXiv:1602.01703.

[80] S. Galera, J. Breil, and P.-H. Maire. A 2D unstructured multi-material cell-centered arbitrary Lagrangian–Eulerian (CCALE) scheme using MOF interface reconstruction. *Computers & Fluids*, 46(1):237–244, 2011.

[81] S. Galera, P.-H. Maire, and J. Breil. A two-dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction. *Journal of Computational Physics*, 229(16):5755–5787, 2010.

[82] R. Garimella, **M. Kucharik**, and M. Shashkov. An efficient linearity and bound preserving conservative interpolation (remapping) on polyhedral meshes. *Computers & Fluids*, 36(2):224–237, 2007.

[83] R. V. Garimella. MSTK – a flexible infrastructure library for developing mesh based applications. In *Proceedings of 13th International Meshing Roundtable*, 2004. September 19-22, 2004, Williamsburg, Virginia, USA.

[84] M. Gittings, R. Weaver, M. Clover, T. Betlach, N. Byrne, R. Coker, E. Dendy, R. Hueckstaedt, K. New, W. R. Oakes, D. Ranta, and R. Stefan. The RAGE radiation-hydrodynamic code. *Computational Science & Discovery*, 1:015005, 2008.

[85] J. Glimm, M. J. Graham, J. Grove, X. L. Li, T. M. Smith, D. Tan, F. Tangerman, and Q. Zhang. Front tracking in two and three dimensions. *Computers & Mathematics with Applications*, 35(7):1–11, 1998.

[86] J. Grandy. Conservative remapping and region overlays by intersecting arbitrary polyhedra. *Journal of Computational Physics*, 148(2):433–466, 1999.

[87] S. Yu. Guskov, S. Borodziuk, M. Kalal, A. Kasperczuk, V. N. Kondrashov, J. Limpouch, P. Pisarczyk, T. Pisarczyk, K. Rohlena, J. Skala, and J. Ullschmied. Investigation of shock wave loading and crater creation by means of single and double targets in the PALS-laser experiment. *Journal of Russian Laser Research*, 26(3):228–244, 2005.

[88] J. A. Harte, W. E. Alley, D. S. Bailey, J. L. Eddleman, and G. B. Zimmerman. LASNEX – A 2D physics code for modeling ICF. *ICF Quarterly Report*, 6(4):150–164, 1997. Lawrence Livermore National Laboratory, UCRL-LR-105821-96-4.

[89] L. R. Herrmann. Laplacian-isoparametric grid generation scheme. *Journal of the Engineering Mechanics Division*, 102(5):749–907, 1976.

[90] R. N. Hill and M. Shashkov. The symmetric moment-of-fluid interface reconstruction algorithm. *Journal of Computational Physics*, 249:180–184, 2013.

[91] 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.

[92] C. W. Hirt and B. D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39(1):201–225, 1981.

[93] Ph. Hoch. An arbitrary Lagrangian-Eulerian strategy to solve com-

pressible fluid flows. Technical report, CEA, 2009. HAL: hal-00366858. Available at `http://hal.archives-ouvertes.fr/docs/00/36/68/58/PDF/ale2d.pdf`.

[94] J. E. Pilliod Jr. and E. G. Puckett. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *Journal of Computational Physics*, 199(2):465–502, 2004.

[95] K. Jungwirth, A. Cejnarova, L. Juha, B. Kralikova, J. Krasa, P. Krupickova E. Krousky, L. Laska, K. Masek, T. Mocek, M. Pfeifer, A. Prag, O. Renner, K. Rohlena, B. Rus, J. Skala, P. Straka, and J. Ullschmied. The Prague Asterix laser system. *Physics of Plasmas*, 8(5):2495–2501, 2001.

[96] M. Kalal, S. Borodziuk, N. N. Demchenko, S. Yu. Guskov, K. Jungwirth, A. Kasperczuk, V. N. Kondrashov, B. Kralikova, E. Krousky, J. Limpouch, K. Masek, P. Pisarczyk, T. Pisarczyk, M. Pfeifer, K. Rohlena, V. B. Rozanov, J. Skala, and J. Ullschmied. High power laser interaction with single and double layer targets. In *Proceedings of XXVIII ECLIM*, pages 249–260, 2004.

[97] J. R. Kamm, M. J. Shashkov, J. Fung, A. K. Harrison, and T. R. Canfield. A comparative study of various pressure relaxation closure models for one-dimensional two-material Lagrangian hydrodynamics. *International Journal for Numerical Methods in Fluids*, 65(11-12):1311–1324, 2011.

[98] J. R. Kamm, M. J. Shashkov, and W. J. Rider. A new pressure relaxation closure model for one-dimensional two-material Lagrangian hydrodynamics. In *New Models and Hydrocodes for Shock Wave Processes in Condensed Matter*, volume 10 of *EPJ Web of Conferences*, page 00038, 2010.

[99] M. Kenamond, M. Bement, and M. Shashkov. Compatible, total energy conserving and symmetry preserving arbitrary Lagrangian-Eulerian hydrodynamics in 2D rz-cylindrical coordinates. *Journal of Computational Physics*, 268:154–185, 2014.

[100] M. A. Kenamond and D. E. Burton. Exact intersection remapping of multi-material domain-decomposed polygonal meshes. Talk at Multimat 2013, International Conference on Numerical Methods for Multi-Material Fluid Flows, San Francisco, September 2-6, 2013. LA-UR-13-26794.

[101] A. Khamayseh and A. Kuprat. Hybrid curve distribution algorithms. *SIAM Journal on Scientific Computing*, 23(5):1464–1484, 2002.

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

[103] K. Klima, **M. Kucharik**, and M. Shashkov. Combination of intersection- and swept-based methods for single-material remap. In

E. Onate, X. Oliver, and A. Huerta, editors, *Proceedings of 6th European Congress on Computational Fluid Dynamics*, pages 5977–5988. International Center for Numerical Methods in Engineering, 2014. ISBN: 978-84-942844-7-2.

[104] M. Klima, **M. Kucharik**, and M. Shashkov. Combined swept region and intersection-based single-material remapping method. *International Journal for Numerical Methods in Fluids*, 85(6):363–382, 2017.

[105] M. Klima, **M. Kucharik**, and M. Shashkov. Local error analysis and comparison of the swept- and intersection-based remapping methods. *Communications in Computational Physics*, 21(2):526–558, 2017.

[106] M. Klima, **M. Kucharik**, M. Shashkov, and J. Velechovsky. Bound-preserving reconstruction of tensor quantities for remap in ALE fluid dynamics. In *Springer Proceedings in Mathematics and Statistics*, 2017. Proceedings of HYP 2016, XVI International Conference on Hyperbolic Problems - Theory, Numerics, Applications, Aachen, Germany. Accepted.

[107] P. Knupp and S. Steinberg. *Fundamentals of Grid Generation*. CRC Press, Boca Raton, 1993.

[108] P. M. Knupp. Matrix norms and the condition number: A general framework to improve mesh quality via node-movement. In *Proceedings of 8th International Meshing Roundtable*, 1999.

[109] P. M. Knupp. Winslow smoothing on two-dimensional unstructured meshes. *Engineering with Computers*, 15:263–268, 1999.

[110] P. M. Knupp, L. G. Margolin, and M. J. Shashkov. Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods. *Journal of Computational Physics*, 176(1):93–128, 2002.

[111] P. Koester, L. Antonelli, S. Atzeni, J. Badziak, F. Baffigi, D. Batani, C. Cecchetti, T. Chodukowski, F. Consoli, G. Cristoforetti, R. De Angelis, G. Folpini, L. Gizzi, Z. Kalinowska, E. Krousky, **M. Kucharik**, L. Labate, T. Levato, R. Liska, G. Malka, Y. Maheut, A. Marocchino, P. Nicolai, T. O'Dell, P. Parys, T. Pisarczyk, P. Raczka, O. Renner, Y. Rhee, X. Ribeyre, M. Richetta, M. Rosinski, L. Ryc, J. Skala, A. Schiavi, G. Schurtz, M. Smid, C. Spindloe, J. Ullschmied, J. Wolowski, and A. Zaras. Recent results from experimental studies on laser-plasma coupling in a shock ignition relevant regime. *Plasma Physics and Controlled Fusion*, 55(12):124045, 2013.

[112] N. I. Kolev. *Multiphase Flow Dynamics*, chapter Numerical solution methods for multi-phase flow problems, pages 481–585. Springer Berlin Heidelberg, 2007.

[113] **M. Kucharik**. *Arbitrary Lagrangian-Eulerian (ALE) Methods in Plasma Physics*. PhD thesis, Czech Technical University in Prague, 2006.

[114] **M. Kucharik**, J. Breil, S. Galera, P.-H. Maire, M. Berndt, and M. Shashkov. Hybrid remap for multi-material ALE. *Computers & Fluids*, 46(1):293–297, 2011.

[115] **M. Kucharik**, R. V. Garimella, S. P. Schofield, and M. J. Shashkov. A comparative study of interface reconstruction methods for multi-material ALE simulations. *Journal of Computational Physics*, 229(7):2432–2452, 2010.

[116] **M. Kucharik** and R. Liska. Arbitrary Lagrangian-Eulerian (ALE) code for plasma simulations. In M. Benes, J. Mikyska, and T. Oberhuber, editors, *Proceedings of Czech-Japanese Seminar in Applied Mathematics*, pages 96–105. Czech Technical University in Prague, 2005. ISBN 80-01-03181-0.

[117] **M. Kucharik**, R. Liska, R. Loubere, and M. Shashkov. Arbitrary Lagrangian-Eulerian (ALE) method in cylindrical coordinates for laser plasma simulations. In S. Benzoni-Gavage and D. Serre, editors, *Hyperbolic Problems: Theory, Numerics, Applications - Proceedings of the Eleventh International Conference on Hyperbolic Problems*, pages 687–694. Springer Verlag, 2008.

[118] **M. Kucharik**, R. Loubere, L. Bednarik, and R. Liska. Enhancement of Lagrangian slide lines as a combined force and velocity boundary condition. *Computers & Fluids*, 83:3–14, 2013.

[119] **M. Kucharik**, G. Scovazzi, M. Shashkov, and R. Loubere. A multi-scale residual-based anti-hourglass control for compatible staggered Lagrangian hydrodynamics. *Journal of Computational Physics*, 2017. Accepted.

[120] **M. Kucharik** and M. Shashkov. Extension of efficient, swept-integration based conservative remapping method for meshes with changing connectivity. *International Journal for Numerical Methods in Fluids*, 56(8):1359–1365, 2008.

[121] **M. Kucharik** and M. Shashkov. Flux-based approach for conservative remap of multi-material quantities in 2D arbitrary Lagrangian-Eulerian simulations. In Jaroslav Fořt, Jiří Fürst, Jan Halama, Raphaèl Herbin, and Florence Hubert, editors, *Finite Volumes for Complex Applications VI Problems & Perspectives*, volume 1 of *Springer Proceedings in Mathematics*, pages 623–631. Springer, 2011.

[122] **M. Kucharik** and M. Shashkov. One-step hybrid remapping algorithm for multi-material arbitrary Lagrangian-Eulerian methods. *Journal of Computational Physics*, 231(7):2851–2864, 2012.

[123] **M. Kucharik** and M. Shashkov. Conservative multi-material remap for staggered multi-material arbitrary Lagrangian-Eulerian methods. *Journal of Computational Physics*, 258:268–304, 2014.

[124] **M. Kucharik** and M. Shashkov. Bound-preserving remapping of

staggered quantities for multi-material ALE methods. *AIP Conference Proceedings*, 1863(1):030025, 2017.

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

[126] A. Kuprat, A. Khamayseh, D. George, and L. Larkey. Volume conserving smoothing for piecewise linear curves, surfaces, and triple lines. *Journal of Computational Physics*, 172:99–118, 2001.

[127] D. Kuzmin, R. Lohner, and S. Turek, editors. *Flux Corrected Transport: Principles, Algorithms, and Applications*. Scientific Computation Series. Springer-Verlag New York, LLC, 2005.

[128] C. B. Laney. *Computational Gasdynamics*. Cambridge University Press, Cambridge, UK, 1998. ISBN 0-521-57069-7.

[129] P. H. Lauritzen, Ch. Erath, and R. Mittal. On simplifying 'incremental remap'-based transport schemes. *Journal of Computational Physics*, 230(22):7957–7963, 2011.

[130] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002. ISBN 0-521-81087-6.

[131] R. Liska and **M. Kucharik**. Arbitrary Lagrangian Eulerian method for compressible plasma simulations. In M. Fila, A. Handlovicova, K. Mikula, M. Medved, P. Quittner, and D. Sevcovic, editors, *Proceedings of EQUADIFF 11, International Conference on Differential Equations*, pages 213–222. STU, 2007. ISBN 978-80-227-2624-5.

[132] R. Liska, **M. Kucharik**, J. Limpouch, O. Renner, P. Vachal, L. Bednarik, and J. Velechovsky. ALE methods for simulations of laser-produced plasmas. In Jaroslav Fořt, Jiří Fürst, Jan Halama, Raphaèl Herbin, and Florence Hubert, editors, *Finite Volumes for Complex Applications VI Problems & Perspectives*, volume 4 of *Springer Proceedings in Mathematics*, pages 857–873. Springer, 2011.

[133] R. Liska, J. Limpouch, **M. Kucharik**, and O. Renner. Selected laser plasma simulations by ALE method. *Journal of Physics: Conference Series*, 112(022009):1–4, 2008.

[134] R. Liska, M. Shashkov, P. Vachal, and B. Wendroff. Optimization-based synchronized flux-corrected conservative interpolation (remapping) of mass and momentum for arbitrary Lagrangian-Eulerian methods. *Journal of Computational Physics*, 229(5):1467–1497, 2010.

[135] R. Loubere. First steps into ALE INC(ubator): A 2D arbitrary-Lagrangian-Eulerian code on general polygonal mesh for compressible flows. Technical Report LA-UR-04-8840, Los Alamos National Laboratory, 2005.

[136] R. Loubere. Contribution to the Lagrangian and ALE numerical meth-

ods, 2013. Habilitation, University of Toulouse.

[137] R. Loubere, P.-H. Maire, M. Shashkov, J. Breil, and S. Galera. ReALE: A reconnection-based arbitrary-Lagrangian–Eulerian method. *Journal of Computational Physics*, 229(12):4724–4761, 2010.

[138] R. Loubere and M. Shashkov. A mixing model for multimaterial computations in fluid dynamics: Remapping and ALE. Technical Report LAUR-05-7571, Los Alamos National Laboratory, 2005.

[139] R. Loubere and M. Shashkov. A subcell remapping method on staggered polygonal grids for arbitrary-Lagrangian-Eulerian methods. *Journal of Computational Physics*, 209(1):105–138, 2005.

[140] R. Loubere, M. Staley, and B. Wendroff. The repair paradigm: New algorithms and applications to compressible flow. *Journal of Computational Physics*, 211(2):385–404, 2006.

[141] G. Luttwak. Second order discrete rezoning. *AIP Conference Proceedings*, 309:1777–1780, 1994. High-pressure science and technology, Schmidt S.C. et al (editors).

[142] Y. Maheut, L. Antonelli, S. Atzeni, J. Badziak, F. Baffigi, D. Batani, C. Cecchetti, T. Chodukowski, F. Consoli, G. Cristoforetti, R. De Angelis, G. Folpini, L.A. Gizzi, Z. Kalinowska, **M. Kucharik**, P. Koster, E. Krousky, L. Labate, T. Levato, R. Liska, G. Malka, A. Marocchino, P. Nicolai, T. O'Dell, P. Parys, T. Pisarczyk, P. Raczka, O. Renner, Y. J. Rhee, X.Ribeyre, M. Richetta, M. Rosinski, L. Ryc, J. Skala, A. Schiavi, G. Schurtz, M. Smid, C. Spindloe, J. Ullschmied, J. Wolowski, and A. Zaras. Experiment on laser interaction with a planar target for conditions relevant to shock ignition. *Physica Scripta*, 2014(T161):014017, 2014.

[143] P.-H. Maire. A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flow on unstructured meshes. *Journal of Computational Physics*, 228(7):2391–2425, 2009.

[144] P.-H. Maire. Contribution to the numerical modeling of inertial confinement fusion, 2011. Habilitation, Université Bordeaux I.

[145] P.-H. Maire. A high-order one-step sub-cell force-based discretization for cell-centered Lagrangian hydrodynamics on polygonal grids. *Computers & Fluids*, 46(1):341–347, 2011.

[146] P.-H. Maire, R. Abgrall, J. Breil, R. Loubere, and B. Rebourcet. A nominally second-order cell-centered Lagrangian scheme for simulating elastic–plastic flows on two-dimensional unstructured grids. *Journal of Computational Physics*, 235:626–665, 2013.

[147] P.-H. Maire, R. Abgrall, J. Breil, and J. Ovadia. A cell-centered Lagrangian scheme for two-dimensional compressible flow problems. *SIAM Journal on Scientific Computing*, 29(4):1781–1824, 2007.

[148] P.-H. Maire, J. Breil, and S. Galera. A cell-centered arbitrary Lagrangian-Eulerian (ALE) method. *International Journal for Numerical Methods in Fluids*, 56(8):1161–1166, 2008.

[149] P.-H. Maire and B. Nkonga. Multi-scale Godunov-type method for cell-centered discrete Lagrangian hydrodynamics. *Journal of Computational Physics*, 228(3):799–821, 2009.

[150] D. Margarone, A. Picciotto, A. Velyhan, J. Krasa, **M. Kucharik**, A. Mangione, A. Szydlowsky, A. Malinowska, G. Bertuccio, Y. Shi, M. Crivellari, J. Ullschmied, P. Bellutti, and G. Korn. Advanced scheme for high-yield laser driven nuclear reactions. *Plasma Physics and Controlled Fusion*, 57:014030, 2015.

[151] D. Margarone, A. Picciotto, A. Velyhan, J. Krasa, **M. Kucharik**, M. Morrissey, A. Mangione, A. Szydlowsky, A. Malinowska, G. Bertuccio, Y. Shi, M. Crivellari, J. Ullschmied, P. Bellutti, and G. Korn. Advanced scheme for high-yield laser driven proton-boron fusion reaction. *SPIE Proceedings, High Power Lasers for Fusion Research III*, 9345:93450F, 2015.

[152] D. Margarone, A. Velyhan, J. Dostal, J. Ullschmied, J. P. Perin, D. Chatain, S. Garcia, P. Bonnay, T. Pisarczyk, R. Dudzak, M. Rosinski, J. Krasa, L. Giuffrida, J. Prokupek, V. Scuderi, J. Psikal, **M. Kucharik**, M. De Marco, J. Cikhardt, E. Krousky, Z. Kalinowska, T. Chodukowski, G. A. P. Cirrone, and G. Korn. Proton acceleration driven by a nanosecond laser from a cryogenic thin solid-hydrogen ribbon. *Physical Review X*, 6:041030, 2016.

[153] L. Margolin and M. Shashkov. Using a curvilinear grid to construct symmetry-preserving discretizations for Lagrangian gas dynamics. *Journal of Computational Physics*, 149(2):389–417, 1999.

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

[155] 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.

[156] L. G. Margolin and M. Shashkov. Remapping, recovery and repair on staggered grid. *Comput. Methods Appl. Mech. Engrg.*, 193:4139–4155, 2004.

[157] D. J. Mavriplis. Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes. In *AIAA 2003-3986*, 2003. 16th AIAA Computational Fluid Dynamics Conference, June 23-26, Orlando, Florida.

[158] D. S. Miller and G. B. Zimmerman. An algorithm for time evolving volume fractions in mixed zones in Lagrangian hydrodynamics calcu-

lations. Technical Report UCRL-PRES-223908, Lawrence Livermore National Laboratory, 2006.

[159] S. J. Mosso, B. K. Swartz, D. B. Kothe, and R. C. Ferrell. A parallel, volume-tracking algorithm for unstructured meshes. In P. Schiano, A. Ecer, J. Periaux, and N. Satofuka, editors, *Parallel Computational Fluid Dynamics: Algorithms and Results using Advanced Computers*, pages 368–375. Elsevier, 1997.

[160] J. W. Murphy and A. Burrows. BETHE-hydro: An arbitrary Lagrangian-Eulerian multidimensional hydrodynamics code for astrophysical simulations. *The Astrophysical Journal Supplement Series*, 179(1):209–241, 2008.

[161] A. Murrone and H. Guillard. A five equation reduced model for compressible two phase flow problems. *Journal of Computational Physics*, 202:664–698, 2005.

[162] W. F. Noh and P. Woodward. SLIC (simple line interface calculation). Technical Report UCRL-77651, Lawrence Livermore Laboratory, 1976. Appeared in A.I. van der Vooren, P.J. Zandbergen (Eds.), Proceedings of Fifth International Conference on Numerical Methods in Fluid Dynamics, Springer-Verlag, pp. 330–340.

[163] S. Osher and R. P. Fedkiw. Level set methods: an overview and some recent results. *Journal of Computational Physics*, 169(2):463–502, 2001.

[164] M. J. Owen and M. J. Shashkov. Arbitrary Lagrangian Eulerian remap treatments consistent with staggered compatible total energy conserving Lagrangian methods. *Journal of Computational Physics*, 273:520–547, 2014.

[165] 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.

[166] 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.

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

[168] A. Picciotto, D. Margarone, M. Crivellari, P. Bellutti, **M. Kucharik**, J. Krasa, M. Barozzi, A. Swidlosky, A. Malinowska, J. Ullschmied, and A. Velyhan. Advanced targets realization for laser driven fusion reactions by standard microelectronics technological procedures. *Journal of Instrumentation*, 2017. Accepted.

[169] A. Picciotto, D. Margarone, A. Velyhan, P. Bellutti, J. Krasa, A. Szydlowsky, G. Bertuccio, Y. Shi, A. Mangione, J. Prokupek, A. Malinowska,

E. Krousky, J. Ullschmied, L. Laska, **M. Kucharik**, and G. Korn. Boron-proton nuclear-fusion enhancement induced in boron-doped silicon targets by low-contrast pulsed laser. *Physical Review X*, 4(3):031030, 2014.

[170] S. Del Pino. A curvilinear finite-volume method to solve compressible gas dynamics in semi-Lagrangian coordinates. *Comptes Rendus Mathematique*, 348(17-18):1027–1032, 2010.

[171] L. F. Richardson. *Weather Prediction by Numerical Process*. Cambridge University Press, 1922.

[172] W. J. Rider, E. Love, M. K. Wong, O. E. Strack, S. V. Petney, and D. A. Labreche. Adaptive methods for multi-material ALE hydrodynamics. *International Journal for Numerical Methods in Fluids*, 65(11-12):1325–1337, 2011.

[173] A. C. Robinson, T. A. Brunner, S. Carroll, R. Drake, C. J. Garasi, T. Gardiner, T. Haill, H. Hanshaw, D. Hensinger, D. Labreche, R. Lemke, E. Love, C. Luchini, S. Mosso, J. Niederhaus, C. C. Ober, S. Petney, W. J. Rider, G. Scovazzi, O. E. Strack, R. Summers, T. Trucano, V. G. Weirs, M. Wong, and T. Voth. ALEGRA: An arbitrary Lagrangian-Eulerian multimaterial, multiphysics code. In *Proceedings of the 46th AIAA Aerospace Sciences Meeting*, Reno, NV, 2008.

[174] S. P. Schofield and M. A. Christon. Effects of element order and interface reconstruction in FEM/volume-of-fluid incompressible flow simulation. *International Journal for Numerical Methods in Fluids*, 68(11):1422–1437, 2012.

[175] S. P. Schofield, M. A. Christon, V. Dyadechko, R. V. Garimella, R. B. Lowrie, and B. K. Swartz. Multi-material incompressible flow simulation using the Moment-of-Fluid method. *International Journal for Numerical Methods in Fluids*, 63(8):931–952, 2010.

[176] S. P. Schofield, R. V. Garimella, M. M. Francois, and R. Loubere. Material order-independent interface reconstruction using power diagrams. *International Journal for Numerical Methods in Fluids*, 56:643–659, 2008.

[177] S. P. Schofield, R. V. Garimella, M. M. Francois, and R. Loubere. A second-order accurate material-order-independent interface reconstruction technique for multi-material flow simulations. *Journal of Computational Physics*, 228(3):731–745, 2009.

[178] W. D. Schulz. Two-dimensional Lagrangian hydrodynamic difference equations. *Methods of Computational Physics*, 3:1–45, 1964.

[179] G. Scovazzi, E. Love, and M. Shashkov. Multi-scale Lagrangian shock hydrodynamics on Q1/P0 finite elements: Theoretical framework and two-dimensional computations. *Computer Methods in Applied Mechanics and Engineering*, 197(9-12):1056–1079, 2008.

[180] M. Shashkov. *Conservative Finite-Difference Methods on General Grids*. CRC Press, Boca Raton, Florida, 1996. ISBN 0-8493-7375-1.

[181] M. Shashkov. Closure models for multimaterial cells in arbitrary Lagrangian-Eulerian hydrocodes. *International Journal for Numerical Methods in Fluids*, 56(8):1497–1504, 2008.

[182] M. Shashkov and P. Knupp. Optimization-based reference-matrix rezone strategies for arbitrary Lagrangian-Eulerian methods on unstructured meshes. In *Proceedings of 10th International Meshing Roundtable*, Newport Beach, California, USA, 2001.

[183] C. D. Sijoy and S. Chaturvedi. Volume-of-fluid algorithm with different modified dynamic material ordering methods and their comparisons. *Journal of Computational Physics*, 229(10):3848–3863, 2010.

[184] H. Stewart and B. Wendroff. Two-phase flow: Models and methods. *Journal of Computational Physics*, 56:363–409, 1984.

[185] M. Sussman and E. G. Puckett. A coupled level set and volume-of-fluid method for computing 3D and axi-symmetric incompressible two-phase flows. *Journal of Computational Physics*, 162(2):301–337, 2000.

[186] B. Swartz. The second-order sharpening of blurred smooth borders. *Mathematics of Computation*, 52(186):675–714, 1989.

[187] J. F. Thompson, F. C. Thames, and C. W. Mastin. Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies. *Journal of Computational Physics*, 15(3):299–427, 1974.

[188] R. E. Tipton. CALE mixed zone pressure relaxation model. Private communication, 1989.

[189] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer Verlag, Berlin, Heidelberg, 1997. ISBN 3-540-61676-4.

[190] P. Vachal, R. Garimella, and M. Shashkov. Untangling of 2D meshes in ALE simulations. *Journal of Computational Physics*, 196(2):627–644, 2004.

[191] P. Vachal and P.-H. Maire. Discretizations for weighted condition number smoothing on general unstructured meshes. *Computers & Fluids*, 46(1):479–485, 2011.

[192] J. Velechovsky, **M. Kucharik**, R. Liska, M. Shashkov, and P. Vachal. Symmetry- and essentially-bound-preserving flux-corrected remapping of momentum in staggered ALE hydrodynamics. *Journal of Computational Physics*, 255:590–611, 2013.

[193] J. Velechovsky, **M. Kucharik**, R. Liska, M. Shashkov, and P. Vachal. Symmetry-preserving momentum remap for ALE hydrodynamics. *Journal of Physics: Conference Series*, 454(1):012003, 2013.

[194] C. S. Venkatasubban. A new finite element formulation for ALE (arbitrary Lagrangian Eulerian) compatible fluid mechanics. 33:1743–1762, 1995.

[195] E. Vitali and D. J. Benson. An extended finite element formulation for contact in multi-material arbitrary Lagrangian-Eulerian calculations. *International Journal for Numerical Methods in Engineering*, 67(10):1420–1444, 2006.

[196] M. L. Wilkins. *Computer Simulation of Dynamic Phenomena*. Scientific Computation. Springer, 1999.

[197] A. M. Winslow. Equipotential zoning of two-dimensional meshes. Technical Report UCRL-7312, Lawrence Livermore National Laboratory, 1963.

[198] J. Yin and C. Teodosiu. Constrained mesh optimization on boundary. *Engineering with Computers*, 24:231–240, 2008.

[199] D. L. Youngs. Time dependent multi-material flow with large fluid distortion. In K. W. Morton and M. J. Baines, editors, *Numerical Methods for Fluid Dynamics*, pages 273–285. Academic Press, 1982.

[200] D. L. Youngs. An interface tracking method for a 3D Eulerian hydrodynamics code. Technical Report AWE/44/92/35, AWRE Design and Math Division, 1984.

[201] S. T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of Computational Physics*, 31(3):335–362, 1979.

Appendices

INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2008; **56**:1359–1365
Published online 7 August 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1577

Extension of efficient, swept-integration-based conservative remapping method for meshes with changing connectivity‡

M. Kucharik*,† and M. Shashkov

T-7 Group, MS B284, Los Alamos National Laboratory, P.O. Box 1663, Los Alamos, NM 87545, U.S.A.

SUMMARY

Remapping is one of the essential parts of most arbitrary Lagrangian–Eulerian methods. Here, we extend the idea of swept integration introduced in (*J. Comput. Phys.* 2003; **184**(1):266–298) to meshes with connectivity changing in Voronoi-like manner. To demonstrate properties of the developed method, we present several numerical examples. Published in 2007 by John Wiley & Sons, Ltd.

Received 10 April 2007; Revised 19 June 2007; Accepted 22 June 2007

KEY WORDS: ALE; conservative interpolations; Voronoi meshes

1. INTRODUCTION

In numerical simulations of fluid flow, the choice of the computational mesh is crucial. Traditionally, there have been two viewpoints, utilizing the Lagrangian or the Eulerian framework, each with its own advantages and disadvantages. In a pioneering paper [1], Hirt *et al*. developed the formalism for a mesh whose motion could be determined as an independent degree of freedom, and showed that this general framework could be used to combine the best properties of Lagrangian and Eulerian methods. This class of methods has been termed Arbitrary Lagrangian–Eulerian or ALE.

It is most usual to separate the ALE algorithm into three individual phases. These are the following: (1) a Lagrangian phase, in which the solution and mesh are updated, (2) a rezoning phase, in which the nodes of the computational mesh are moved to a more optimal position, and (3) a remapping phase, in which the Lagrangian solution is interpolated onto the rezoned mesh. We are interested in the development of staggered ALE methods for meshes whose connectivity may

*Correspondence to: M. Kucharik, T-7 Group, MS B284, Los Alamos National Laboratory, P.O. Box 1663, Los Alamos, NM 87545, U.S.A.
†E-mail: kucharik@lanl.gov

Contract/grant sponsor: U.S. Department of Energy; contract/grant number: DE-AC52-06NA25396

change during the calculation. In such methods, the total number of cells remains fixed, but the number of edges bounding each cell may change with time. Changing connectivity adds another degree of freedom to the method—in the case of shear flows, initially close cells may not be close to each other in later stages, and methods without reconnections fail due to mesh tangling. We focus here on the Voronoi meshes [2] constructed from arbitrary set of points (Voronoi generators) in the computational domain. Each cell of the Voronoi mesh corresponds to one of the Voronoi generators, and is defined as a set of points, which are closer to the particular generator than to all the other ones. Generally, each node of the Voronoi mesh connects three different edges. However, quadrilateral meshes (with four edges in each mesh node) can be constructed by degeneration of two nodes to one physical location. By movement of the Voronoi generators, the mesh nodes and edges move, and reconnection in the particular node can appear. We allow the topology changes caused by the prescribed movement of the Voronoi generators.

This paper focuses primarily on the last phase of the ALE algorithm—remapping. We are looking for the remapping algorithm, which satisfies several important conditions: (1) conservation, (2) local-bound preservation, (3) linearity preservation, and (4) efficiency. The complete remapping algorithm based on approximate swept integration was presented in [3] for the case of 2D logically orthogonal computational meshes with the same connectivity. This algorithm does not require finding the cell intersections, and is face-based and thus more efficient than the natural exact integration method. In this paper, we describe the process of dealing with Voronoi-like intersections, and extending the swept-integration-based remapping algorithm to similar meshes with changing connectivity.

2. SWEPT-INTEGRATION-BASED METHOD FOR MESHES WITH IDENTICAL TOPOLOGY

Our remapping algorithm [3] includes three steps: (1) piecewise-linear reconstruction of the unknown quantity function (density of mass, total energy, total momenta in both directions) in each cell from the mean values of the particular quantity in the neighboring cells; (2) approximate integration of the reconstructed function interpolating new mean values on the new mesh; and (3) repair, mass redistribution procedure enforcing local-bound preservation. The reconstruction and repair stages are performed exactly the same way, as described in [3] for 2D and in [4] for 3D.

Let us focus on the second step, the approximate swept integration. Suppose, we know the mean values $\bar{g}_c$ (and masses $m_c = \bar{g}_c V_c$) in cells of the original mesh $\{c\}$, and during the reconstruction stage, we have computed slopes in each cell c. Here, V_c stands for volume of cell c, and is computed analytically as integral of 1 over the particular cell. Our goal is to compute new mean values $\bar{g}_{\tilde{c}}$ (and masses $m_{\tilde{c}}$) in the cells of the new mesh $\{\tilde{c}\}$. The same cell c in original mesh and new mesh $\tilde{c}$ are shown in Figure 1(a) and (b). The swept integration method is based on the idea that the mass in the new cell $m_{\tilde{c}}$ can be written in the flux form as the original mass m_c plus masses of swept regions δm_e (mass fluxes) corresponding to each edge e of the cell

$$\tilde{m}_{\tilde{c}} = m_c + \sum_{e \in \partial c} \Omega_c(e)\, \delta m_e \tag{1}$$

where $\Omega_c(e)$ is equal to 1 if edge e is in counter-clockwise orientation according to cell c and -1 otherwise. Swept masses $\delta m_e = \int_{\delta V_e} g_{c^*}(\mathbf{r})\, \mathrm{d}S$ are computed as integrals over swept region δV_e

Published in 2007 by John Wiley & Sons, Ltd. *Int. J. Numer. Meth. Fluids* 2008; **56**:1359–1365
DOI: 10.1002/fld

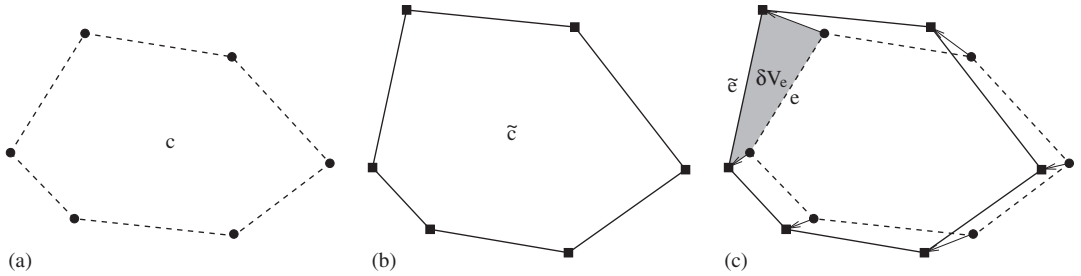


Figure 1. One cell (a) c in original dashed mesh; (b) $\tilde{c}$ in new solid mesh; and (c) swept region δV_e corresponding to old edge e and new edge $\tilde{e}$.

attached to edge e (see Figure 1(c)). The reconstruction is taken from cell c^*, which is selected according to the sign of the volume of swept region δV_e. If the edge e moves inwards the cell c, the swept volume is negative and we take the reconstruction from cell $c^* = c$. In the opposite case, the reconstruction is taken from the cell neighboring with c over edge e. The described swept integration algorithm is approximate, and does not guarantee preservation of local bounds. Therefore, repair stage [5] enforcing this property must be added. In [3] we have shown that these algorithms followed by the repair stage satisfy all our conditions stated in Section 1.

3. SWEPT-INTEGRATION-BASED METHOD FOR MESHES WITH CHANGING TOPOLOGY

After detecting, that there is no connectivity change around the particular cell (its neighborhood remains the same), we perform the algorithm as described in Section 2. Now, let us discuss the possibility that the connectivity changes (the neighborhood of the particular cell is different in old and new meshes). The typical situation is shown in Figure 2, showing four cells in original mesh (a), new mesh (b), and both meshes plotted over each other (c). As we can see, the original bottom-left (BL) cell was neighbor of the upper-right (UR) cell. In the new mesh, their common edge disappeared, and new edge was added, causing the upper-left (UL) and bottom-right (BR) cells to be neighbors in the new mesh. We call this type of reconnection (one removed and one added edge) Voronoi-like reconnection.

When such a reconnection is detected, we find the center of reconnection C by averaging the coordinates of vertices of removed and added edges, as shown in Figure 3(a). Then, we follow the swept integration algorithm, as described before, in two steps. At first, we shrink the removed edge to this central point and perform swept integrations corresponding to all involved edges—the removed edge and four edges connected to it (see Figure 3(b)). In the second step, similar five swept integrations are performed by extending the central point to the created edge (see Figure 3(c)).

For correct functioning of the described algorithm, we must ensure that each mesh edge is involved in at most one reconnection. This is not generally fulfilled, but it can be achieved in the case of Voronoi meshes obtained by generators movement (as in our case). When more (two) reconnections at one edge are detected, the generators movement can be reduced and performed in two or more steps. Time-step reduction causes higher total number of remappings, but it guarantees satisfaction of the single reconnection condition.

Published in 2007 by John Wiley & Sons, Ltd. *Int. J. Numer. Meth. Fluids* 2008; **56**:1359–1365
DOI: 10.1002/fld

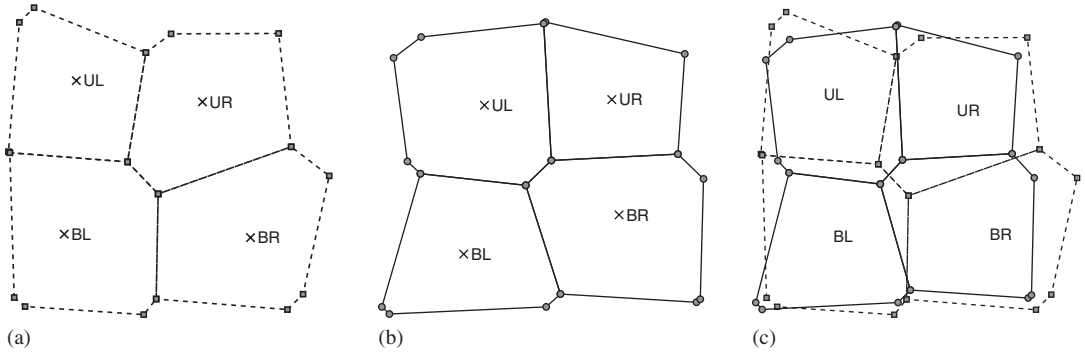


Figure 2. Reconnection in Voronoi mesh: (a) piece of original; (b) new mesh; and (c) both meshes over each other.

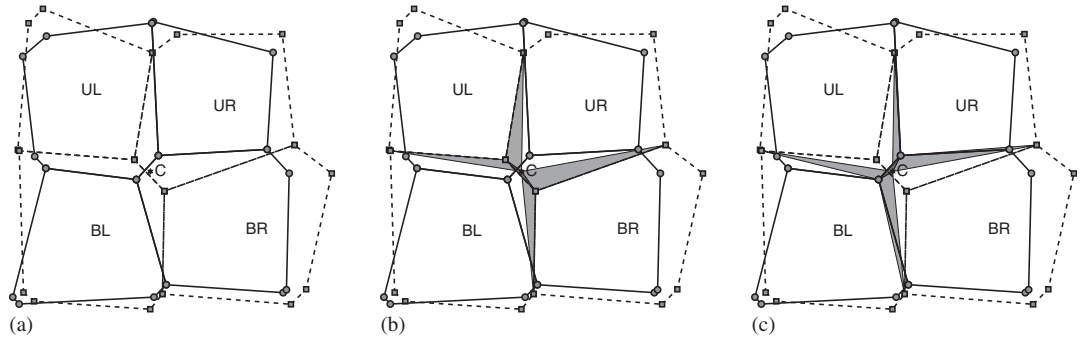


Figure 3. Three steps for handling of reconnection: (a) location of center of reconnection C; (b) shrinking of removed edge to center C; and (c) expanding of center C to new edge.

4. NUMERICAL EXAMPLES

In this section, we present several numerical tests to demonstrate properties of our remapping algorithm. The complete method was implemented in C programing language using MSTK [6] environment for mesh representation.

As the testing mesh, we present here the Voronoi mesh generated by the uniformly spread 32^2 generators in the $\langle 0, 1\rangle^2$ computational domain. The generated Voronoi mesh is the regular logically orthogonal mesh, in fact each vertex is degenerated from two Voronoi vertices. To look at the cumulative effects of many remappings, we remap the function over a sequence of meshes (usually called cyclic remapping [7]), generated as Voronoi meshes from the moved generators. The velocity of the Voronoi generator of cell c in time t^{n+1} is described by the stream-like formula:

$$u^{n+1} = -\alpha(t^{n+1})(\sin(\pi x_c^n))^2 \sin(2\pi y_c^n), \quad v^{n+1} = +\alpha(t^{n+1})(\sin(\pi y_c^n))^2 \sin(2\pi x_c^n) \tag{2}$$

where the parameter $\alpha(t) = \cos(\pi t)$, and x_c^n, y_c^n are coordinates of the cell generator in time level t^n. Computation ends in final time $t = 3$. The generated sequence of meshes includes many connectivity changes, testing all capabilities of our algorithm.

Published in 2007 by John Wiley & Sons, Ltd. *Int. J. Numer. Meth. Fluids* 2008; **56**:1359–1365
DOI: 10.1002/fld

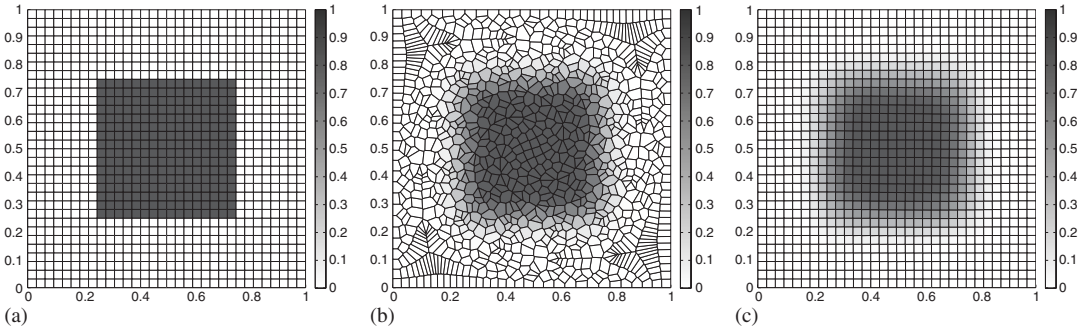


Figure 4. Square color function in 1024 cells of (a) initial; (b) middle; and (c) final computational mesh.

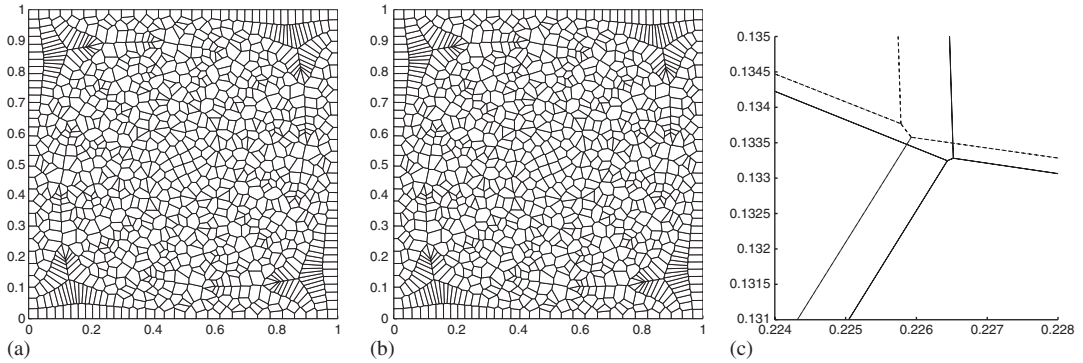


Figure 5. Two consecutive meshes in the middle of the stream-like mesh movement: (a) previous mesh in time $t = 1.4352$; (b) following mesh in time $t = 1.5000$; and (c) zoom to overlap of both meshes in lower-left region of the computational domain including one typical reconnection.

As the first test function, we use a linear function $g(x, y) = 1 + x + 2y$. The numerical error is zero (up to the round off error) in all simulations (with different initial meshes and different generators movements), which confirms linearity preservation of our algorithm.

The mean values of the second test function in the cells of the initial mesh are shown in Figure 4(a). This color square function is equal to 1 inside the square of edge length $\frac{1}{2}$ located in the center of the computational domain, and 0 otherwise. In Figure 4(b), we can see the situation in the middle of the remapping process in time $t = \frac{3}{2}$. As we can see, the mesh topology is completely different from the initial one. In Figure 5, we can see two consecutive meshes in the middle of the simulation, in times $t = 1.4352$ and 1.5000. As we can see, both meshes are very similar to each other. To demonstrate that the meshes have different topologies, we zoomed one typical reconnection in the lower-left part of the computational domain in Figure 5(c).

The remapped function mean values in the final time are shown in Figure 4(c). We can observe the dissipation around the square edge, accumulated from many remappings. In Table I, we present a comparison of numerical relative L_1 errors and times of computation for our swept-integration-based method and the natural method based on exact integration. As we can see, the new method is more than three times faster than the natural method, which is caused by the fact that exact

Published in 2007 by John Wiley & Sons, Ltd. *Int. J. Numer. Meth. Fluids* 2008; **56**:1359–1365
DOI: 10.1002/fld

Table I. Comparison of relative L_1 error and time of computation T in seconds for standard method based on exact integration and new method using approximate swept integrations.

$\|\{c\}\|$	L_1^{exact}	L_1^{swept}	T^{exact} (s)	T^{swept} (s)
16	3.6×10^{-1}	3.7×10^{-1}	1.1×10^{0}	1.0×10^{0}
64	2.1×10^{-1}	2.1×10^{-1}	9.0×10^{0}	5.4×10^{0}
256	1.3×10^{-1}	1.3×10^{-1}	1.2×10^{2}	4.7×10^{1}
1024	7.8×10^{-2}	7.8×10^{-2}	1.9×10^{3}	6.0×10^{2}
4096	4.8×10^{-2}	4.8×10^{-2}	3.1×10^{4}	9.7×10^{3}
16 384	3.0×10^{-2}	3.0×10^{-2}	4.9×10^{5}	1.5×10^{5}

Note: Comparison is shown for several initially uniform meshes with $|\{c\}|$ cells. Simulations performed on standard PC with 2.0 GHz AMD Opteron processor.

integration method requires finding all intersections of both meshes. The numerical errors of both methods are almost the same, the order of convergence is close to the first order for the non-smooth function. The local extrema were not overshot in any cell of the computational mesh.

We have also performed several tests with smooth 2D sine function. As in the presented color square function example, numerical errors of swept and exact integration methods are almost identical. For smooth functions, we numerically achieved second order of convergence of our algorithm in L_1 error. This is the consequence of the linearity preservation property of our algorithm.

5. CONCLUSION

In this paper, we have described the extension of the swept-integration-based remapping algorithm proposed in [3] to the general polygonal meshes with connectivity changing in Voronoi-like manner. We have also presented several numerical examples including numerical errors, which verifies linearity and local-bound preservation, conservation, and applicability to general 2D polygonal meshes. Comparison of our algorithm with the classical exact integration algorithm was performed, showing comparable numerical errors and higher efficiency of new method.

ACKNOWLEDGEMENTS

This work was carried out under the auspices of the National Nuclear Security Administration of the U.S. Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396. The authors thank R. Garimella, L. Margolin, B. Wendroff, B. Swartz, R. Liska, M. Berndt and V. Dyadechko for fruitful discussions and constructive comments.

REFERENCES

1. Hirt CW, Amsden AA, Cook JL. An arbitrary Lagrangian–Eulerian computing method for all flow speeds. *Journal of Computational Physics* 1974; **14**(3):227–253.
2. Aurenhammer F, Klein R. Voronoi diagrams. In *Handbook of Computational Geometry*, Chapter 5. Elsevier: Amsterdam, The Netherlands, 2000; 201–290.
3. Kucharik M, Shashkov M, Wendroff B. An efficient linearity-and-bound-preserving remapping method. *Journal of Computational Physics* 2003; **188**(2):462–471.

Published in 2007 by John Wiley & Sons, Ltd. *Int. J. Numer. Meth. Fluids* 2008; **56**:1359–1365
DOI: 10.1002/fld

4. Garimella R, Kucharik M, Shashkov M. An efficient linearity and bound preserving conservative interpolation (remapping) on polyhedral meshes. *Computers & Fluids* 2007; **36**(2):224–237.
5. Shashkov M, Wendroff B. The repair paradigm and application to conservation laws. *Journal of Computational Physics* 2004; **198**(1):265–277.
6. Garimella R. Mesh data structure selection for mesh generation and FEA applications. *International Journal for Numerical Methods in Engineering* 2002; **55**(4):451–478.
7. Margolin LG, Shashkov M. Second-order sign-preserving conservative interpolation (remapping) on general grids. *Journal of Computational Physics* 2003; **184**(1):266–298.

Published in 2007 by John Wiley & Sons, Ltd.

Int. J. Numer. Meth. Fluids 2008; **56**:1359–1365
DOI: 10.1002/fld

Journal of Computational and Applied Mathematics 233 (2010) 3148–3156

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Reduced-dissipation remapping of velocity in staggered arbitrary Lagrangian–Eulerian methods

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

[a] Lawrence Livermore National Laboratory, P.O. Box 808 L-016, Livermore, CA 94551, USA

[b] Theoretical Division, T-5, Los Alamos National Laboratory MS-B284, Los Alamos, NM 87545, USA

[c] Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Brehova 7, Praha 1, 115 19, Czech Republic

ARTICLE INFO

Article history:
Received 16 April 2009
Received in revised form 2 September 2009

Keywords:
Conservative interpolations
Staggered discretization
Flux-based remap
Velocity remap

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 Bailey (1984) [11] 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.

1. Introduction

Arbitrary Lagrangian–Eulerian (ALE) methods introduced in [1] appear to be a reasonable compromise between Lagrangian and Eulerian approaches, allowing the solution of 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 [2–5].

It is possible to formulate the ALE scheme as a single algorithm [6] 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 [7]. 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

* Corresponding author at: Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Brehova 7, Praha 1, 115 19, Czech Republic. Tel.: +420 22435 8637.

E-mail addresses: dsb@llnl.gov (D. Bailey), berndt@lanl.gov (M. Berndt), kucharik@lanl.gov, kucharik@newton.fjfi.cvut.cz (M. Kucharik), shashkov@lanl.gov (M. Shashkov).

0377-0427/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2009.09.008

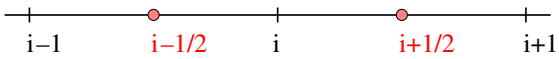


Fig. 1. Enumeration of nodes (black) and cells (red) of the 1D computational mesh. Coordinates of cell centers (red circles) computed by averaging of involved nodal coordinates.

example in [8–10], 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 [11], 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. Conservation of the kinetic energy also guarantees that the remapped velocities cannot grow without bound. This approach requires the solution of a global system of coupled non-linear equations. This method has been used successfully for many years in complex ICF simulations and the cost is nominal; much less than that to remap all the many other state variables.

This paper has three main goals:

(1) illustrates that approach [11] does not always work;
(2) describes an alternative approach, which yields the same solution as [11] when it exists and reduces dissipation if it does not;
(3) highlights 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, (in [11], a 2D extension of the original algorithm is presented) and we have implemented a 2D extension of our modified method 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 Fig. 1. The nodal mass in node i is remapped 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 [12]. 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. Although in real calculations it can be complicated to compute the inter-nodal mass flux F^m to better than first order of accuracy, for the purposes of this paper, the particular method for computation of the mass fluxes is not important.

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 [13,2], 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 which in certain circumstance will exactly conserve the 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 [11]. 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 [11] 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

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

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

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

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

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

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 Eqs. (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 κ represents the iteration index. The iterative process is then

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

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

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

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 ϵ is chosen on the order of 10^{-14}–10^{-10}, and the nodal kinetic energies are computed as

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

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

An alternative approach to solving this system is based on the construction of a flux discrepancy function for each index $i + 1/2$ by moving the left-hand side of (11) to the right-hand side,

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

After substituting for $\bar{u}$s from (6), and for $\widetilde{u}$s from (2), the function $\mathcal{F}_{i+1/2}$ only depends on u^*s. To avoid overwhelming the reader with the nodal and flux indices, we consider $\mathcal{F}_{i+1/2}$ to be just one component of a vector function $\vec{\mathcal{F}}$, and similarly for $\vec{u}^*$. 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 a system of coupled quadratic equations of the general form

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

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 Fig. 2.

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 [12]. In

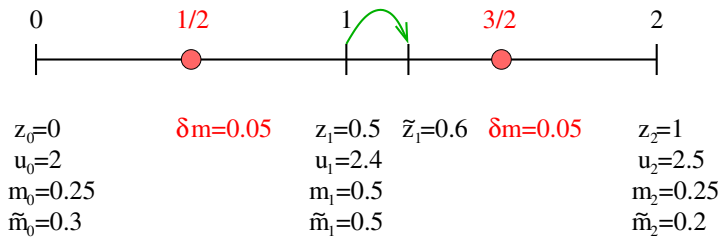


Fig. 2. Initial data of 1D example. Mesh nodes shown by black line segments, cell centers are shown by red circles. Movement of the central node is shown by green arrow, nodal positions (old position z and new position $\widetilde{z}$), velocities (u), and masses (old mass m and new mass $\widetilde{m}$)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

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

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

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

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

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

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

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 $\|\vec{\mathcal{F}}(\vec{x})\|^2 = \mathcal{F}_1^2(\vec{x}) + \mathcal{F}_2^2(\vec{x})$ 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 of both functions, $J_{\mathcal{F}}$ and $J_{\mathcal{G}}$. Jacobian of $\mathcal{F}$ is better conditioned than $J_{\mathcal{G}}$, but $J_{\mathcal{G}}$ is symmetric whereas $J_{\mathcal{F}}$ is not. Now we construct a 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

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

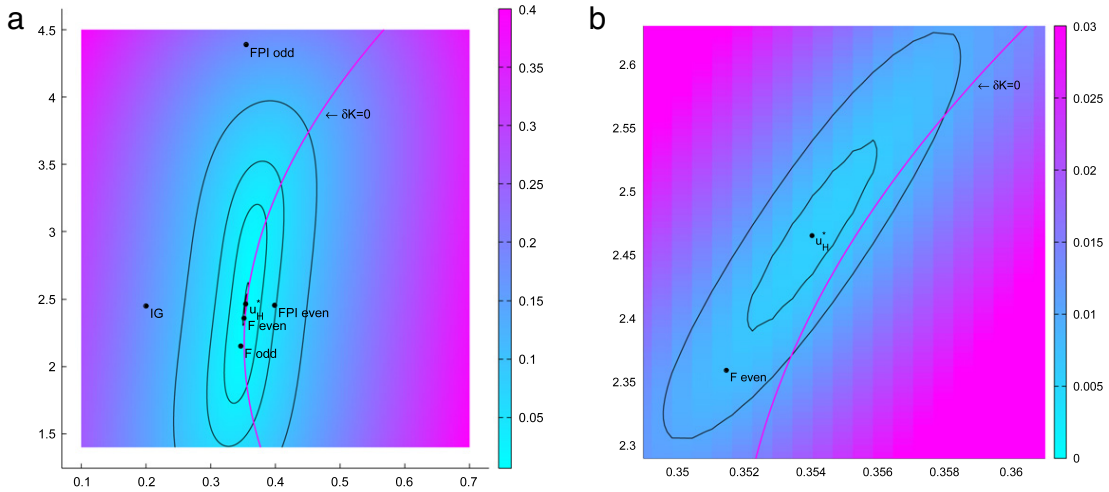


Fig. 3. Colormap and several isolines of $\mathcal{G}(x_1, x_2) = \|\vec{\mathcal{F}}(x_1, x_2)\|^2$ sampled over $x_1 \in \langle 0.1, 0.7\rangle$ and $x_2 \in \langle 1.4, 4.5\rangle$ (a), and zoomed to the center of the sampling region (b). Horizontal axis represents $x_1 = u^*_{1/2}$, vertical one represents $x_2 = u^*_{3/2}$. Magenta line represents 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 $\vec{\mathcal{H}} = \vec{0}$ ($u^*_{\mathcal{H}} = \vec{x}^{\mathrm{sol}\,\mathcal{H}}$). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

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 $\mathcal{G}$ up to machine accuracy, and thus the point closest to the solution of system (31). In this point, the norm of $\vec{\mathcal{F}}$ is still relatively large, $\|\vec{\mathcal{F}}(\vec{x}^{\mathrm{sol}\,\mathcal{H}})\| \sim 5.51 \times 10^{-3}$. The energy discrepancy here is $\delta K = K - \widetilde{K} = -2.75 \times 10^{-4}$ 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 $\delta K = -3.28 \times 10^{-2}$. 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 Fig. 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 the isoline of $K - \widetilde{K} = 0$, 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 $\vec{\mathcal{F}}(\vec{x}) = 0$ we find the minimum of $\|\vec{\mathcal{F}}(\vec{x})\|^2$, which, if the solution of $\vec{\mathcal{F}}(\vec{x}) = 0$ 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.

4.4. Flux-corrected remap (FCR)

As we pointed out above, any high-order methods (for example, the described potentially kinetic-energy-conservative algorithm) can introduce oscillations in the velocity field that exceed the local extrema. To avoid this problem, the flux-corrected remap (FCR) based on the flux-corrected transport (FCT) approach [15] can be used. In this approach, the final momentum flux is decomposed into a low-order flux and an anti-diffusive flux (constructed from high-order and low-order fluxes)

$$F^{\mu}_{i+1/2} = F^{\mu,L}_{i+1/2} + C_{i+1/2}\left(F^{\mu,H}_{i+1/2} - F^{\mu,L}_{i+1/2}\right). \tag{50}$$

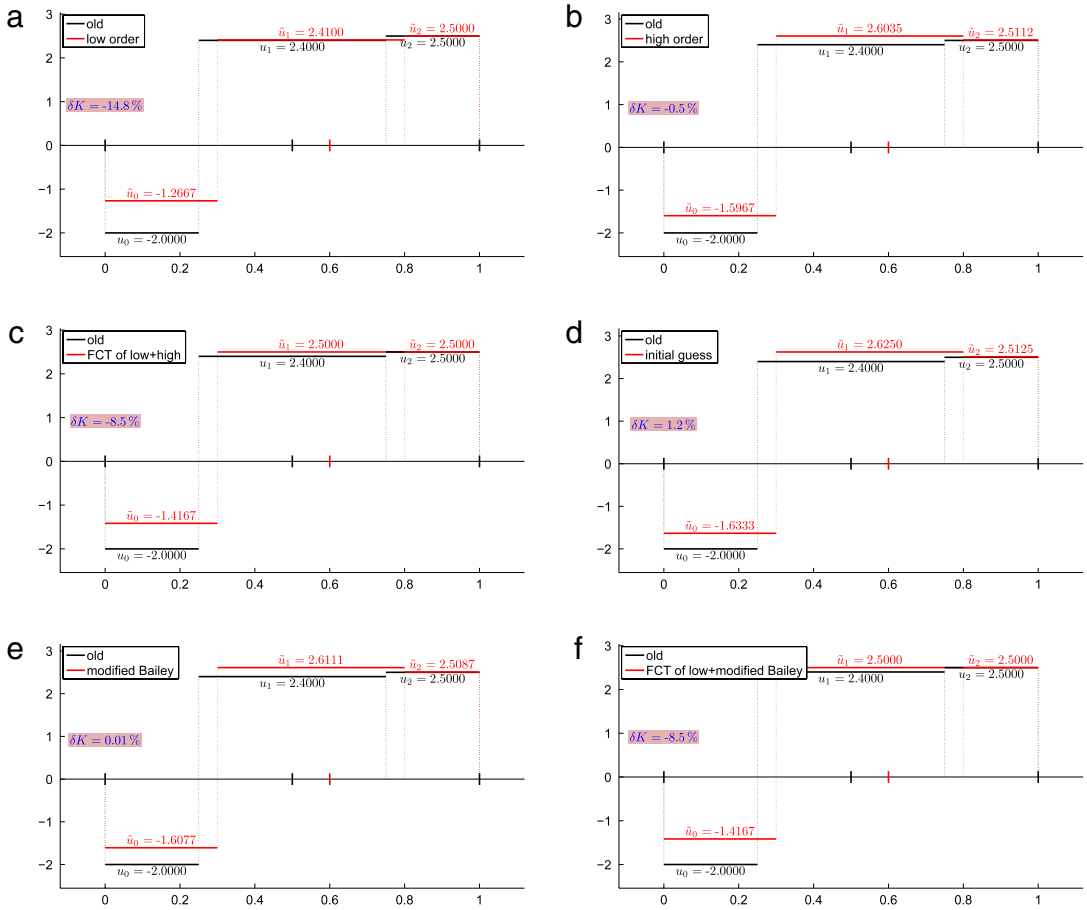


Fig. 4. Velocity profiles before and after remap in the counter-example. Black line shows the initial profile, the new profiles are shown in red: (a) low-order (donor) velocity reconstruction; (b) high-order (piece-wise linear) velocity reconstruction; (c) FCT of low- and high-order approaches; (d) initial guess from the potentially kinetic-energy-conservative algorithm (13); (e) potentially kinetic-energy-conservative algorithm; and (f) FCT of low order and potentially kinetic-energy-conservative approaches. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The coefficients $C_{i+1/2}$ are computed using the standard FCT approach to be as close to one as possible (to maintain high order for fluxes), but to respect the local bound preservation condition. The momentum fluxes can be written as high- or low-order reconstructed velocity that is multiplied by the same mass flux, thus, yielding the following formula

$$F^{\mu}_{i+1/2} = u^{*}_{i+1/2} F^{m}_{i+1/2}, \quad u^{*}_{i+1/2} = u^{*,L}_{i+1/2} + C_{i+1/2}\left(u^{*,H}_{i+1/2} - u^{*,L}_{i+1/2}\right), \tag{51}$$

where the low-order flux velocity $u^{*,L}_{i+1/2}$ is typically computed from the piece-wise constant (donor) velocity reconstruction, and the high-order flux velocity $u^{*,H}_{i+1/2}$ can be computed from a piece-wise linear velocity reconstruction or from our potentially kinetic-energy-conservative method. In this context, we take the donor concept to be the mass flux dependent reconstruction

$$u^{*,L}_{i+1/2} = \begin{cases} u_{i+1} & \text{if } F^{m}_{i+1/2} \geq 0 \\ u_{i} & \text{if } F^{m}_{i+1/2} < 0. \end{cases} \tag{52}$$

We now demonstrate the behavior of the methods in Fig. 4. We observe that the high-order methods (b), (d), and (e) produce velocity overshoots in the central and right cells. The FCT correction alleviates these overshoots with corrected velocity values between the local extrema. We note that, in this example, when the FCT mechanism is applied to the original high-order non-monotone methods (including our new method), it significantly increases the kinetic energy discrepancy. The kinetic energy discrepancy for the low-order (donor) velocity reconstruction is the worst with a relative discrepancy of -14.8% when compared to the initial kinetic energy. The high-order (piece-wise linear) reconstruction is significantly better with a relative discrepancy to the initial kinetic energy of -0.5%. The discrepancy of our proposed method and the initial kinetic energy is about 0.01%. The FCT mechanism when applied to high-order methods (b) or (e) increases the relative

discrepancy from −0.5% or 0.01%, respectively, to a fairly high value of −8.5%. The FCT correction affects only regions where oscillations were introduced (e.g., along discontinuities in density), thus, its impact on the overall simulation is not as dramatic as our example suggests.

5. Conclusion

In this paper, we have discussed a potentially kinetic-energy-conservative algorithm [11] 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 $\|\vec{\mathcal{F}}(\vec{x})\|$ instead of solving the original system $\vec{\mathcal{F}}(\vec{x}) = \vec{0}$. 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. Lawrence Livermore National Laboratory is operated by Lawrence Livermore National Security, LLC, for the U.S. Department of Energy, National Nuclear Security Administration under Contract DE-AC52-07NA27344. M. Kucharik has been supported in part by the Czech Ministry of Education grant MSM 6840770022.

References

[1] C.W. Hirt, A.A. Amsden, J.L. Cook, An arbitrary Lagrangian-Eulerian computing method for all flow speeds, Journal of Computational Physics 14 (3) (1974) 227–253.
[2] D.J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Computer Methods in Applied Mechanics and Engineering 99 (2–3) (1992) 235–394.
[3] L.G. Margolin, Introduction to "An arbitrary Lagrangian-Eulerian computing method for all flow speeds", Journal of Computational Physics 135 (2) (1997) 198–202.
[4] P. Kjellgren, J. Hyvarinen, An arbitrary Lagrangian-Eulerian finite element method, Computational Mechanics 21 (1) (1998) 81–90.
[5] J.S. Peery, D.E. Carroll, Multi-material ALE methods in unstructured grids, Computer Methods in Applied Mechanics and Engineering 187 (3-4) (2000) 591–619.
[6] J. Donea, S. Guiliani, 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) (1982) 689–723.
[7] E.J. Caramana, D.E. Burton, M.J. Shashkov, P.P. Whalen, The construction of compatible hydrodynamics algorithms utilizing conservation of total energy, Journal of Computational Physics 146 (1) (1998) 227–262.
[8] L.G. Margolin, M. Shashkov, Second-order sign-preserving conservative interpolation (remapping) on general grids, Journal of Computational Physics 184 (1) (2003) 266–298.
[9] L.G. Margolin, M. Shashkov, Remapping, recovery and repair on staggered grid, Computer Methods in Applied Mechanics and Engineering 193 (39–41) (2004) 4139–4155.
[10] M. Kucharik, M. Shashkov, B. Wendroff, An efficient linearity-and-bound-preserving remapping method, Journal of Computational Physics 188 (2) (2003) 462–471.
[11] D.S. Bailey, Second-order monotonic advection in LASNEX, in: Laser Program Annual Report'84, number UCRL-50021-84, 1984, pp. 3-57–3-61.
[12] 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, Technical report, LLNL, 2000. UCRL-JC-139820.
[13] R.B. DeBar, Fundamentals of the KRAKEN code, Technical Report UCIR-760, Lawrence Livermore Laboratory, 1974.
[14] M. Pernice, H.F. Walker, NITSOL: A Newton iterative solver for nonlinear systems, SIAM Journal on Scientific Computing 19 (1) (1998) 302–318.
[15] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, Journal of Computational Physics 31 (3) (1979) 335–362.

Journal of Computational Physics 229 (2010) 2432–2452

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A comparative study of interface reconstruction methods for multi-material ALE simulations

Milan Kucharik [a,b,*], Rao V. Garimella [a], Samuel P. Schofield [a], Mikhail J. Shashkov [a]

[a] Applied Math and Plasma Physics (T-5), Los Alamos National Laboratory, P.O. Box 1663, Los Almos, NM 87545, United States
[b] Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University, Prague, Brehova 7 Praha1, 115 19 Prague, Czech Republic

ARTICLE INFO

Article history:
Received 9 March 2009
Received in revised form 3 July 2009
Accepted 13 July 2009
Available online 23 July 2009

Keywords:
Interface reconstruction
Moment-of-fluid method
Compressible flow

ABSTRACT

In this paper we compare the performance of different methods for reconstructing interfaces in multi-material compressible flow simulations. The methods compared are a material-order-dependent Volume-of-Fluid (VOF) method, a material-order-independent VOF method based on power diagram partitioning of cells and the Moment-of-Fluid method (MOF). We demonstrate that the MOF method provides the most accurate tracking of interfaces, followed by the VOF method with the right material ordering. The material-order-independent VOF method performs somewhat worse than the above two while the solutions with VOF using the wrong material order are considerably worse.

1. Introduction

Accurate simulation of multi-material and multi-phase flows requires effective tracking and management of material interfaces. Due to their ability to strictly conserve the mass of different materials, volume-of-fluid (VOF) methods using interface reconstruction are widely used in such simulations [1–4]. Originally developed by Hirt and Nichols [5], VOF methods do not explicitly track interfaces but rather track the volume of each material. The interface between materials is first reconstructed in cells based on the material volume fractions. Then the volume fluxes of each material between cells are estimated from the geometric reconstruction and finally, the fluxes are used to compute new volume fractions in each cell, in preparation for the next time step.

More recently, an interface tracking method has been devised based on tracking both the volume (zeroth moment) and centroid (ratio of first and zeroth moment) of the materials in mesh cells. This new method, called the Moment-of-Fluid (MOF) method [6], reconstructs interfaces more accurately than VOF methods and is able to resolve interfacial features on the order of the local mesh size whereas VOF methods do poorly in resolving features smaller than 3–4 times the local mesh size. In this paper, we present a comparative study of different VOF methods and the MOF method for complex compressible flow simulations involving more than two materials. It is organized as follows: in Section 2, we present a brief overview of the common material order-dependent VOF methods. We describe the basic principle of each method and focus mainly on the Youngs' VOF method, which is implemented in most multi-material codes. We describe the problems with choosing the correct material ordering for such methods. In Section 3, we describe the order independent VOF method based on the power diagrams. In Section 4, the MOF material reconstruction method is described. The slope of the material interface is not determined from the volume fractions of the neighboring cells, but from the material centroids of the particular

* Corresponding author. Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University, Prague, Brehova 7 Praha1, 115 19 Prague, Czech Republic.
E-mail addresses: kucharik@lanl.gov, kucharik@newton.fjfi.cvut.cz (M. Kucharik), rao@lanl.gov (R.V. Garimella), sams@lanl.gov (S.P. Schofield), shashkov@lanl.gov (M.J. Shashkov).

0021-9991/$ - see front matter © 2009 Published by Elsevier Inc.
doi:10.1016/j.jcp.2009.07.009

cell. In Section 5, we briefly describe all steps of the ALE algorithm implemented in our research multi-material code. We focus mainly on the propagation of the material centroids needed for the MOF material reconstruction during the Lagrangian and remapping steps of the algorithm. Coupling of the material reconstruction methods with a multi-material ALE code is described. Section 6 is the key part of the paper. It includes comparison of the described material reconstruction methods in the context of particular multi-material hydrodynamic simulations including typical phenomena appearing in real problems – vortex, explosion, and a shock wave-material interaction. All numerical examples include more than 2 materials to emphasize key properties of each method. Finally, we conclude the paper and review the material reconstruction methods in Section 7.

2. VOF methods with nested dissection (VOF-PLIC)

Early VOF methods used a straight line aligned with a coordinate axis to partition the cell according to the material volume fractions. This is often referred to as the simple line interface calculation (SLIC) originally due to Noh and Woodward [7]. Youngs [8,9] extended the method to permit the material interface to have an arbitrary orientation within the cell (called PLIC or Piecewise Linear Interface Calculation by Rider and Kothe [3]). In Youngs' method, the outward normal of the interface separating a material from the rest of the cell is taken to be the negative gradient of the "volume fraction function". The "volume fraction function" is treated as a smooth function whose cell-centered values are given by the cell-wise material volume fractions. The interface is then defined by locating a line with the prescribed normal that cuts off the correct volume of material from the computational cell.

Gradient based methods are in general first order accurate although they may exhibit near second order accuracy on regular Cartesian grids. However, there are extensions that make the reconstruction second-order accurate for general grids. The LVIRA technique by Pilliod and Puckett [10] tries to find an extended straight line interface that cuts off the exact volume fraction in the cell of interest and minimizes the error in matching the volume fractions in the surrounding cells. LVIRA uses a minimization procedure with a gradient-based normal as the initial guess. An alternative is the interface smoothing procedure based on Swartz's quadratically convergent procedure [11,14] for finding a straight line that cuts off the right volume fractions from two arbitrary planar shapes.[1] Mosso et al. [14] and Garimella et al. [15] have used this procedure in slightly different ways to devise interface smoothing procedures. For a given mixed cell, Garimella et al. compute a straight line cutting off the right volume fractions from the cell and each of its mixed cell neighbors by the Swartz method. The normals of these different straight lines are then averaged to give a smoothed interface normal for the cell.

VOF-PLIC techniques have been successfully used to accurately simulate two-phase (or two-material) flows and free-surface flows in two and three dimensions. However, their application to flows involving three or more materials that come closer than the mesh spacing and even form junctions has been mostly *ad hoc*. Examples of such phenomena are flows of immiscible fluids (e.g. oil–water–gas), inertial confinement fusion, armor–antiarmor penetration and powder metallurgical simulation of multiple materials.

The most common extensions of PLIC to cells with more than two materials (multi-material cells)[2], is to process materials one by one leading to a reconstruction that is strongly dependent on the order in which the materials are processed. Of the different ways to sequentially partition a cell, one of the most general and accurate ways is called the "nested dissection" method [6], where each material is separated from the others in a specified order. In the method, a pure polygon (or polyhedron) representing the first material is marked out from the cell, leaving a mixed polygon for the remaining materials. Then, a polygon representing the second material is marked out from the mixed polygon and the process continues until the last material is processed. This method is illustrated in Fig. 1 and described in detail in [6,16,17]. Clearly, such an order dependent method can easily place materials in wrong locations in the cells if the chosen order of processing is incorrect. Even if the order of the materials is right, the computation of the interface normals in multi-material cells is ambiguous. In computing the normal as the negative gradient of the volume fraction function of a material, it is unclear whether one should use the volume fractions with respect to original cells or the part of the cells remaining after the earlier materials have been removed. It is also not clear where these function values should be centered – at the center of the original cell or the center of the unprocessed part of the cell.

The most significant adverse effect of these incorrect reconstructions, however, is in material advection in flow simulations. An improper material ordering may result in materials being advected prematurely (or belatedly) into neighboring cells. This can further lead to small pieces of the material getting separated and drifting away from the bulk of the material (sometimes known as "flotsam and jetsam"). The effect of material ordering is illustrated clearly in an example from [18] in which a four-material disk (with each material occupying one quadrant of the disk) is advected diagonally for 30 time steps. The results in Fig. 2 show dramatically different results with different material orderings and a complete loss of the cross-shaped interface.

The most common and trivial way to deal with the material order dependency is to select the "correct" global ordering for a problem. However, this is obviously problematic if the same materials must be processed differently in different parts of the mesh or if the material configurations change as the problem advances in time. Also, some interface configurations may

[1] This is commonly known as the "ham-sandwich" or Steinhaus problem [12,13].

[2] In a strict sense, any cell with more than one material is a multi-material cell. However, we choose to distinguish two material cells from cells with more than two materials by calling the latter multi-material cells. The reason for this distinction is that interface reconstruction for one material is (in the case of VOF methods) complementary to the second in a cell with two-materials while it is not for more than two materials.

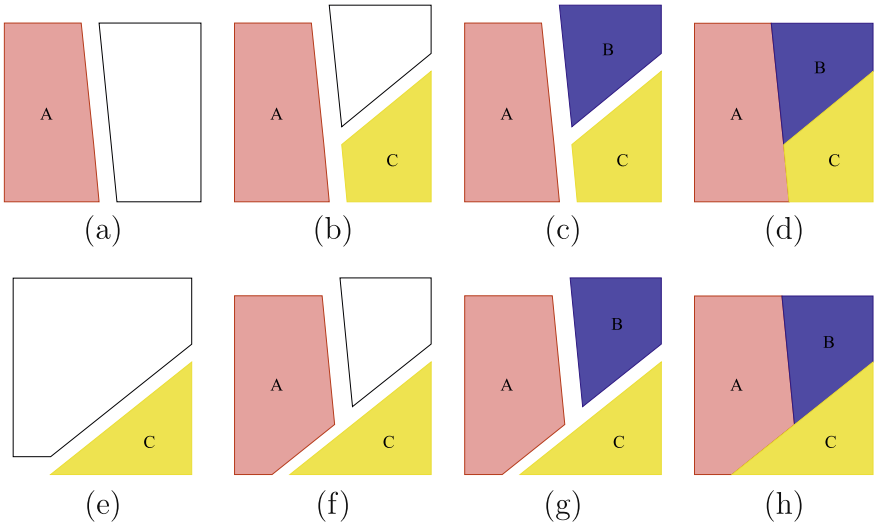


Fig. 1. Nested dissection interface reconstruction for three materials in the order ACB: (a) the first (A) material is removed leaving a smaller available polygon, (b) the second (C) material is removed from the available polygon, (c) the remaining available polygon is assigned to material B, (d) the resulting partitioning of the computational cell. (e)–(g) show the same procedure but the materials are processed in a different (CAB) order leading to a different reconstruction (h).

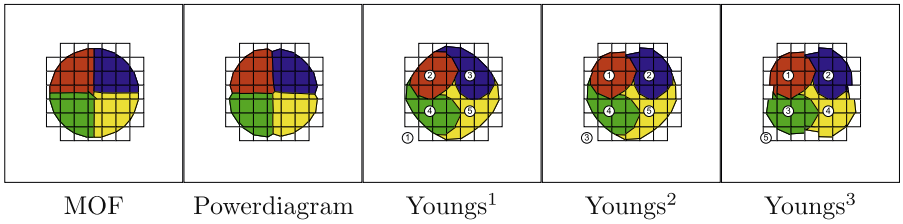


Fig. 2. Four material disk at time $T = 0.5$ translated from the initial position $(0.2, 0.2)$ with 30 time steps at a velocity of $(1, 1)$ on 32×32 mesh of the $[0, 1]^2$ domain. Material reconstruction done by several methods – MOF, VOF with power diagrams, and Youngs' VOF. The material orderings for Youngs' are indicated in the figure.

not be reproducible by any particular order, such as the four material example referred to above. While there has been some work on automatically deriving material order, most of these attempts assume a layered structure for the interface [14,19] and cannot handle multiple materials coming together at a point very well.

3. VOF methods with power diagram reconstruction (VOF-PD)

Recently, Schofield et al. [18] developed a new VOF-based reconstruction method that is completely material order independent. This method, called the Power Diagram method for Interface Reconstruction, does not sequentially carve off materials from a cell using straight lines. Rather it first locates materials approximately in multi-material cells and then partitions the cell simultaneously into multiple material regions using a weighted Voronoi decomposition thereby avoiding the order dependence problem. We describe this procedure below referring to it as the VOF-PD method.

In the first step of the VOF-PD method, approximate locations or "centroids" of the materials in a cell are determined using the volume fractions of the materials in the cell and its neighbors. This is accomplished by treating the volume fractions of each material in the cell and its neighbors as pointwise values of a pseudo-density function. The pointwise values of this pseudo-density function are then used to obtain a linear reconstruction of the function along with application of a limiter restricting the minimum and maximum values to 0 and 1, respectively. Then the linear approximation of this pseudo-density function is used to derive an approximate centroid for the material in the cell. While this method does not locate the material centroids very accurately in an absolute sense, it does locate the materials quite well relative to each other.

In the second step of the procedure, the approximate centroids of the materials are used as generators for a weighted Voronoi or Power Diagram subdivision [20,21] of the cell. The weights of the different generators are chosen iteratively such that the volume fractions of the different Voronoi polygons truncated by the cell boundary match the specified material volume fractions exactly.

The authors have shown that this procedure is in general first-order accurate and for two materials, exactly reproduces a gradient-based subdivision of the cell. They have also presented a smoothing procedure for the power diagram-based subdivision which results in a second-order accurate reconstruction but slows the procedure down considerably unless applied only to cells with more than two materials.

4. Moment-of-fluid (MOF) method

While VOF methods track only volume fractions of the individual materials in mesh cells, the recently developed Moment-of-fluid (MOF) method [6] tracks both the volume (zeroth moment) and centroid (ratio of first and zeroth moment) of the materials in the cells. By tracking both moments the MOF method reconstructs the material interface with higher accuracy than VOF methods and is able to resolve interfacial details on the order of the local mesh size. In contrast, VOF methods can only resolve details on the order of 3–4 times the local mesh size. Also, since a line can be determined by only two parameters (an intercept and a slope), the linear interface in a cell is actually over-determined by specifying the volume fraction and centroid. This implies that MOF can perform an exact reconstruction of a linear interface and a second-order reconstruction of a smoothly curved interface in a cell without the need for information from neighboring cells.

Given the volume fraction and centroid of a material in a cell, the MOF reconstruction method computes a linear interface such that the volume fraction of the material is exactly matched and the discrepancy between the specified centroid and the centroid of the polygon or polyhedron behind interface is minimized. This is done by an optimization process with the slope of the linear interface (or its angle with respect to the x-direction) as the primary variable. For any given slope, the intercept of the line is determined uniquely by matching the specified material volume fraction.

The MOF reconstruction is also typically implemented as a nested dissection method where materials are carved off from a cell sequentially thereby making it an order-dependent problem. However, it is possible to combinatorially determine the correct sequence of material reconstructions in MOF by reconstructing with all possible sequences and choosing the sequence which leads to the least discrepancy between the reconstruction and specified centroids. Although the number of possible sequences grows as the factorial of the number of materials, the computational overhead of this approach is tolerable as each cell contains only a small number of materials for most problems. Also, more complex configurations such as 4 materials coming together at a point can be reconstructed by recursively reconstructing the interface between groups of materials first and then resolving the interfaces between materials in each group. Again, due to the small number of materials in a cell, this does not impose a significant computational penalty. Such a technique has proved very effective in accurately reconstructing multi-material interfaces.

Further details of the MOF technique of interface reconstruction are given in [6,17].

5. Compressible flow simulation with VOF and MOF reconstructions

Here we briefly describe an arbitrary-Eulerian–Lagrangian (ALE) compressible flow simulation algorithm used to compare the effects of the VOF and MOF reconstruction techniques. Since the purpose of this paper is to compare the different interface reconstruction methods, we deliberately do not provide many details of the ALE code to avoid overwhelming the discussion. We believe the general conclusions of this comparative study will hold regardless of the ALE code used.

Our 2D research multi-material ALE code (RMALE) has a standard structure shown in Fig. 3.

It consists of three main components – multi-material Lagrangian solver, mesh untangling and smoothing method, and a flux-based multi-material remapper. The Lagrangian step is repeated, until the mesh smoothing becomes necessary (for example, due to poor mesh quality, or a given number of hydro steps being completed). When mesh smoothing is applied to improving the mesh quality it is followed by a remapping step conservatively interpolating all quantities on the new mesh. Then, a new Lagrangian cycle can begin. The entire code employs a staggered Mimetic Finite Difference discretization [22], where scalar fluid quantities (density, mass, pressure, internal energy) are located inside mesh cells, and vector quantities (positions, velocities) on mesh nodes. The multi-material ALE framework allows more than one material inside one computational cell, where the amount of each material is defined by its volume and mass fractions, and if we use MOF, the relative location of each material is defined by the material centroid. In each multi-material cell, scalar quantities are defined separately for every material, but the variables in the primary equations are the average cell quantities. Contrary to a single-material approach, our multi-material Lagrangian step and remapper must update not only all fluid quantities, but also material volume and mass fractions, and material centroids.

The Lagrangian solver solves the following set of hydrodynamic equations

$$\frac{1}{\rho}\frac{d\rho}{dt} = -\nabla\cdot\mathbf{w}, \quad \rho\frac{d\mathbf{w}}{dt} = -\nabla\cdot p, \quad \rho\frac{d\varepsilon}{dt} = -p\nabla\cdot\mathbf{w} \tag{1}$$

representing conservation of mass, momenta in both directions, and total energy, completed by the ideal gas equation of state $p = (\gamma - 1)\rho\varepsilon$. Here, ρ is the fluid density, $\mathbf{w}$ is the vector of velocities, p is the fluid pressure, ε is the specific internal energy, and γ is the ratio of specific heats. The solver is based on evaluation of several types of forces affecting each mesh node [22] – zonal pressure force representing forces due to the pressure in all neighboring zones, artificial viscosity force

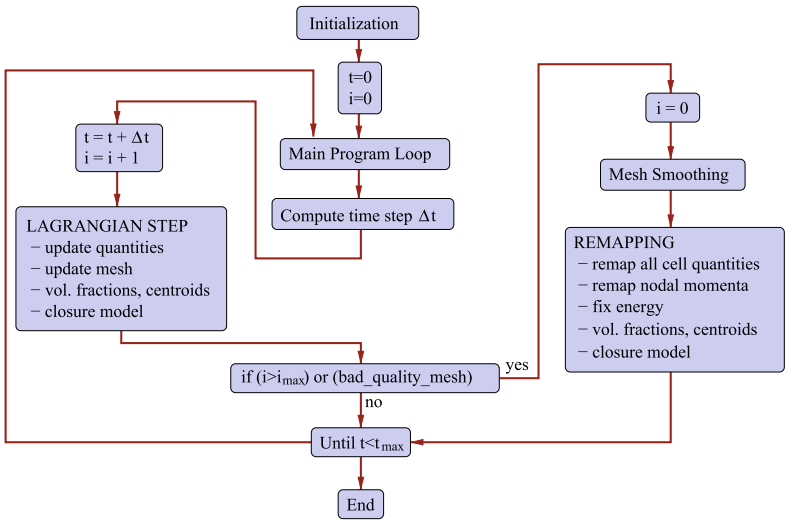


Fig. 3. Flowchart of our research multi-material code. Material reconstruction is hidden in the update of material centroids at the end of the Lagrangian step.

Fig. 4. Initial conditions for static triple point problem. Materials are shown in different colors, and values of ratio of specific heats γ, density ρ, pressure p, and velocity u are listed.

(edge viscosity [23] is used in the examples), and anti-hourglass stabilization force introduced in [24], suppressing some unphysical modes in the mesh motion. The viscosity forces in the mixed cells are computed from the average fluid quantities, and the appropriate heating is redistributed among the particular materials according to their mass fractions. For volume fraction update and common pressure construction, a multi-material closure model is applied [25]. In our numerical examples, the simplest model employing the constant volume fractions (equal strain model [1]), is used. The last part of the Lagrangian step is a method for updating the material centroids. In the first step, we advect them by keeping their parametric coordinates constant. Appendix A shows that this method reproduces the Lagrangian motion of the centroid for compressible flows with second-order accuracy. These centroids are then used (together with updated volume fractions) as reference centroids for the next material reconstruction step. The final material centroids are then set to the centroids of the reconstructed polygons.

Our code incorporates several mesh-untangling and mesh-smoothing methods. All ALE examples in this paper use one iteration of the classical Winslow mesh smoothing algorithm [26] performed in a Jacobi manner to avoid breaking the problem symmetry.

The last essential part of the ALE code is a remapping technique interpolating all fluid and material quantities between Lagrangian and smoothed computational meshes. Our remapper employs the cell-cell or pure polygon-cell intersections and exact integration in the entire mesh, performed in a flux form.

This flux-based remapper represents the multi-material extension of the technique described in [27] – it constructs inward and outward fluxes of integrals of 1, x, y, and some higher order polynomials using overlays (intersections) of Lagrangian cells (or pure material polygons in the case of mixed cells) with their neighbors in the smoothed mesh, and vice versa. Note that these integrals of polynomials over polygons can be computed analytically. Fluxes of all cell- and

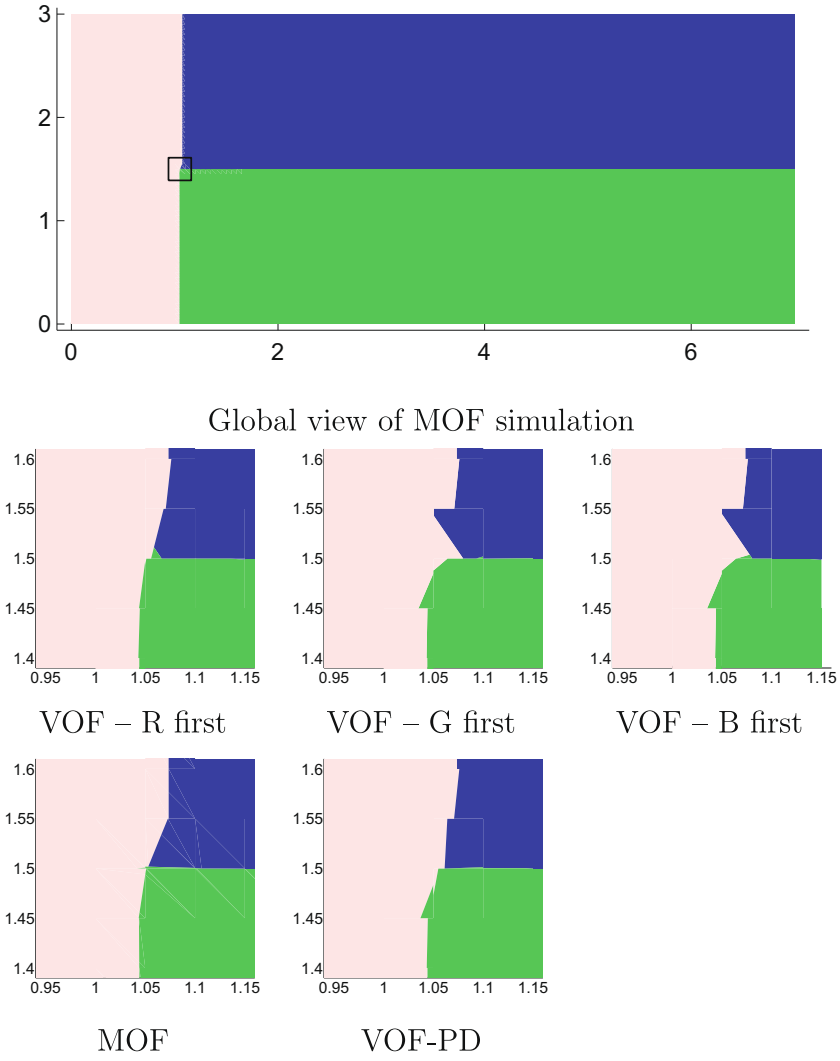


Fig. 5. Materials of triple point problem simulation, time $T = 0.1$. Eulerian runs (as Lagrangian step and remap to the initial orthogonal mesh) using different methods for material reconstruction are shown: global view on the entire computational domain for MOF method, and zooms to the three material junction for Youngs' VOF method (with different material orderings), MOF, and Power Diagram based methods are shown.

material-centered quantities are then constructed from these pre-computed exchange integrals, the material quantities (mass, internal energy) are remapped in a material-by-material way. They are also used for remapping material volumes (and consequently volume fractions) and centroids in a flux form. For remapping nodal mass, we need to construct inter-nodal mass fluxes, which we interpolate from inter-cell mass fluxes as described in [28], extended by split side fluxes for adjacent cells and corner fluxes. All nodal quantities are then remapped by attaching them to these inter-nodal mass fluxes (for example, the momentum fluxes are obtained by multiplication of the mass fluxes by an interpolated flux velocity). This approach allows us to construct two kinetic energies at each node – conservative kinetic energy obtained by its remap, and non-conservative kinetic energy obtained from remapped velocities. This kinetic energy discrepancy is resolved by a standard energy fix [1], it is redistributed into the remapped internal energy of adjacent materials, and thus global energy conservation is guaranteed. For a complete detailed description of our multi-material remapping method, see [29].

The material reconstruction method is performed at the end of the Lagrangian stage, during the centroid update process. This whole step can be avoided when VOF type of method is used, and no centroid information is required. The second part of the ALE algorithm employing the material reconstruction method is the beginning of the remapping stage, during the computation of the material exchange integrals, and can also be reused during the slope (of density or internal energy) limiting. This material reconstruction must be performed in every remapping step, independent of the reconstruction method used, or the data from the Lagrangian step reconstruction can be reused, if it was performed.

6. Numerical examples

We demonstrate the properties of the described material reconstruction methods in the context of multi-material ALE hydrocode for three types of problems. These are: a triple point problem containing a strong vortex in its solution, a multi-material modification of the Sedov problem representing a material expansion (and thus its narrowing) due to a point explosion, and finally a multi-material modification of Saltzman problem employing the interaction of the piston-generated shock wave with a multi-material structure. These three problems represent a wide range of processes involved in real complex numerical hydro simulations. In our comparison, we focus especially on the material topology (relative position of the materials) and on how well the thin material filaments are resolved.

6.1. Triple point problem

The initial data for the triple point problem [30] is shown in Fig. 4. The computational domain has a rectangular shape with 7×3 edge ratio. In all simulations, we use an equispaced orthogonal initial computational mesh with 140×60 cells. It includes three materials at rest, initially forming a T-junction. The high-pressure material (in light red or white) creates a shock wave moving to the right, through the low pressure blue (or darkest gray) and green (medium gray) materials. Due to different material properties, it moves faster in the blue or dark gray (lower density) material, and therefore a vortex evolves around the triple point. In the later stages of the simulation (final time $T = 5$), we can observe thin filaments of materials

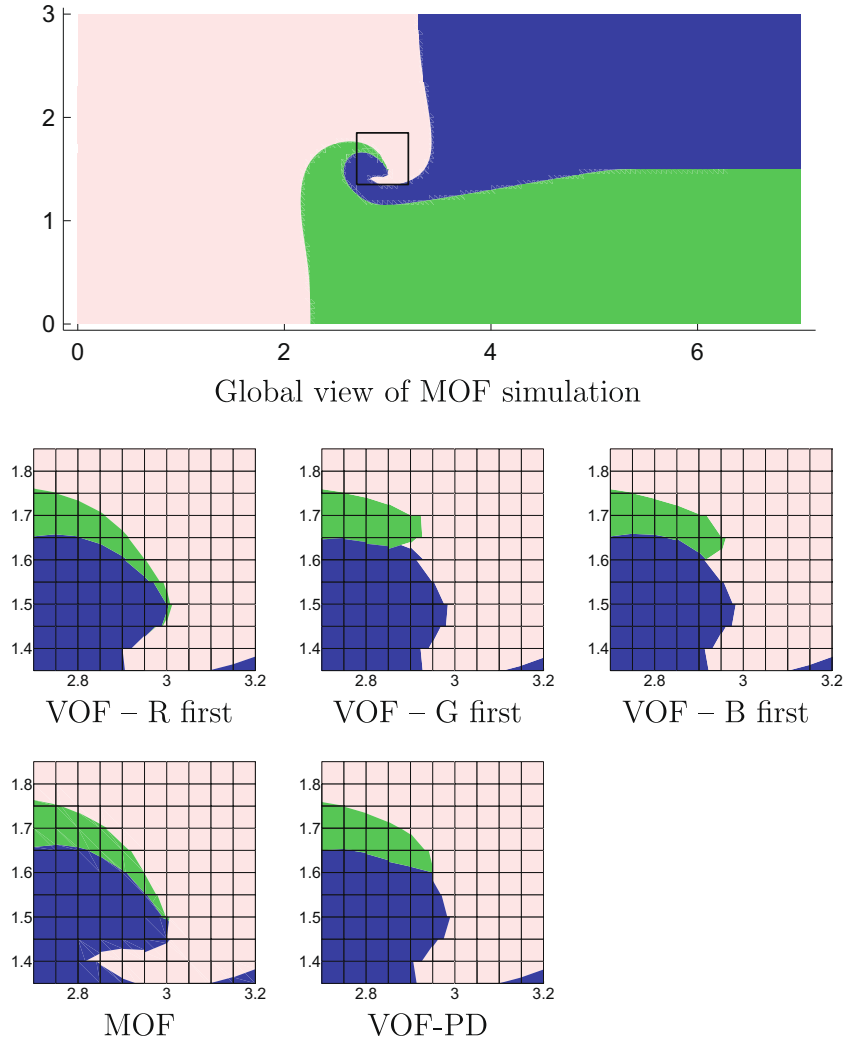


Fig. 6. Materials of triple point problem simulation, time $T = 2.5$. Eulerian runs (as Lagrangian step and remap to the initial orthogonal mesh) using different methods for material reconstruction are shown: global view on the entire computational domain for MOF method, and zooms to the three material junction for Youngs' VOF method (with different material orderings), MOF, and Power Diagram based methods are shown.

rotating around the vortex. It is to be noted that no mixed cells are present at the beginning of the simulations, however, they appear during the first remap.

Here, we compare a traditional gradient-based VOF method with different orderings, the MOF method, and a VOF method based on power diagrams (VOF-PD). We perform the comparison for two types of simulations: Eulerian and full ALE. In the Eulerian approach, the solution is remapped back to the orthogonal initial mesh after each Lagrangian step, while in the ALE approach, Winslow mesh smoothing and consecutive remapping is performed after every 20 Lagrangian steps.

In Fig. 5, we can see the first snapshot of the Eulerian simulation, corresponding to time $T = 0.1$. In this early moment, the white–blue interface is shifted more to the right than the white–green one. As we can see, smooth interfaces are preserved when using VOF starting with white material, which is the correct local material ordering for this particular problem, and when using the MOF method. The VOF with Power diagrams still provide acceptable results, while VOF methods using wrong orderings created very distorted interfaces leading to problems in later stages of the simulation.

A snapshot in the middle of the simulation ($T = 2.5$) is shown in Fig. 6. A thin filament of green material is starting to develop, which is reasonably resolved using MOF and VOF with the correct ordering. VOF with power diagrams keeps the correct topology of materials, but starts to have problems with resolving the thin filament. VOF with the wrong material orderings provides the worst results – the filament starts to separate from the heavy blue material, and there are small pieces of white material between green and blue that are not easily visible at this scale.

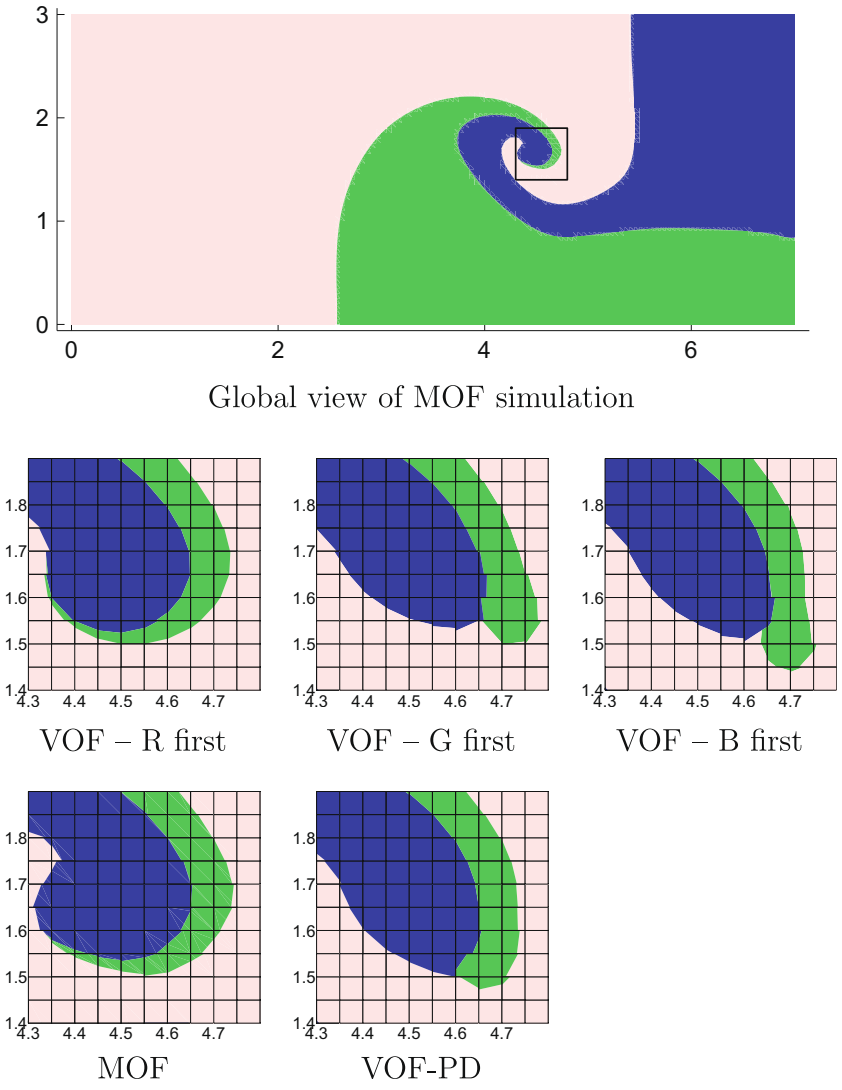


Fig. 7. Materials of triple point problem simulation, time $T = 5.0$. Eulerian runs (as Lagrangian step and remap to the initial orthogonal mesh) using different methods for material reconstruction are shown: global view on the entire computational domain for MOF method, and zooms to the three material junction for Youngs' VOF method (with different material orderings), MOF, and Power Diagram based methods are shown.

In Fig. 7, we can see the final snapshot of the Eulerian simulation corresponding to time $T = 5$. Again, MOF and VOF in the correct ordering resolve the thin part of the green filament reasonably well. VOF with the wrong material orderings give us unacceptable results – filament transforms into a drip separating from the blue material, and there are many tiny droplets of white material between the blue and green materials VOF with power diagrams also do not succeed in resolving the thin part of the filament, but the result is qualitatively better: the material topology is correct, no droplets appear, and green material stays attached to the blue one.

In the next set of figures, the results of the same problem obtained by ALE approach are presented. Generally, the results are worse than for the Eulerian simulations due to the distorted computational mesh.

In Fig. 8, the early stages of an ALE simulation at time $T = 0.1$ are presented for the same example. As we can see, the MOF results are best of all methods being compared, the multi-material interface smoothly transitions from the white–blue to the white–green interface and no major jumps appear. The results of VOF in correct ordering are comparable to the results of VOF with power diagrams at this early stage. We can observe minor material jumps and smoothness of the interface is violated. The worst results are clearly obtained by VOF methods using the wrong material orderings. The T-shape of the interface is completely violated and an unphysical wedge of white material starts to separate blue and green materials, leading to more severe problems in later stages of the simulations.

Fig. 9 presents results in the middle of the simulation ($T = 2.5$). In this time moment, the (initially orthogonal) computational mesh is already relatively distorted. As we can see, VOF in correct ordering resolves the longest green filament. Filament resolved by MOF is shorter, compact, with a relatively smooth interface. Power diagrams and VOF with wrong

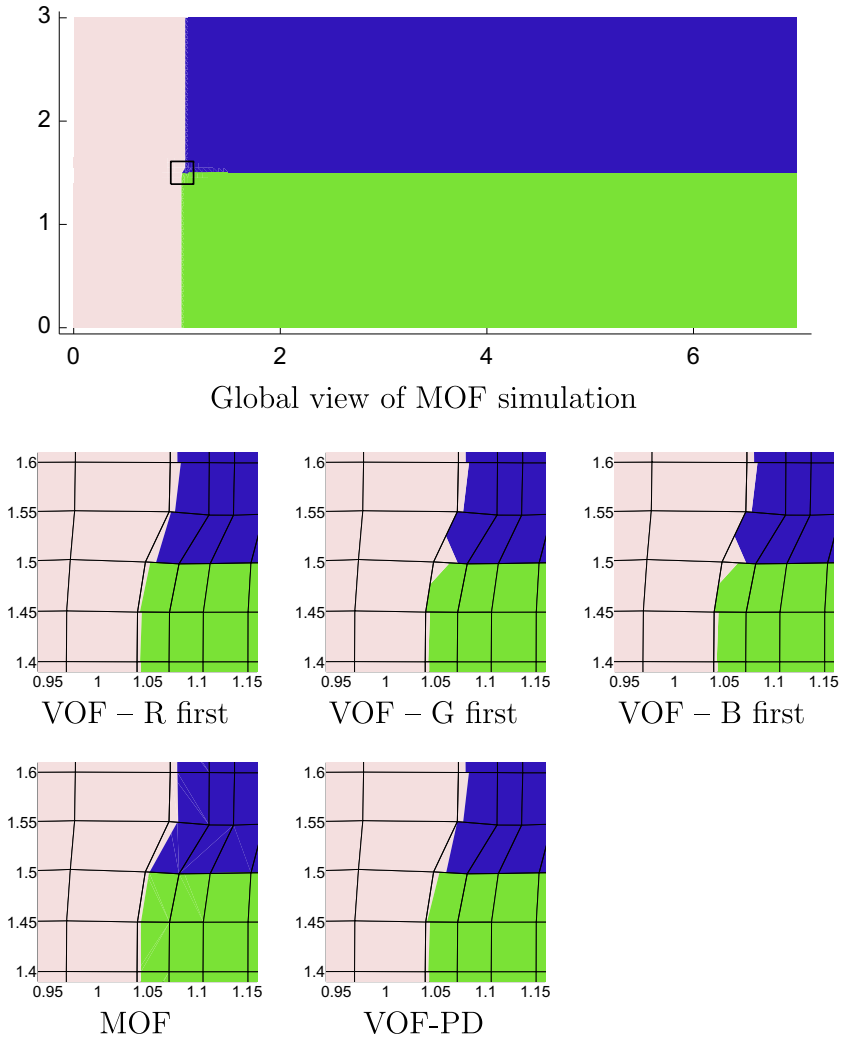


Fig. 8. Materials of triple point problem simulation, time $T = 0.1$. ALE runs (as Lagrangian step and remap to the Winslow smoothed mesh after every 20 Lagrangian steps) using different methods for material reconstruction are shown: global view on the entire computational domain for MOF method, and zooms to the three material junction for Youngs' VOF method (with different material orderings), MOF, and Power Diagram based methods are shown.

material orderings do not resolve the filament very well, but power diagrams surpass VOF with incorrect material order in material topology – no fragment of white and blue material appear on the other side of the green filament.

In Fig. 10, we can see the last moment ($T = 5$) of the ALE simulation.

MOF provides best result again – the filament is compact, relatively smooth, no separated tiny droplets are present. We can observe such small pieces for all VOF methods, even for correct ordering, where a tiny thin fiber of green material separates white–blue interface upto the picture boundary (zoomed in the last image of Fig. 10). As for power diagrams, no droplets appear, but we can see that the green filament has broken into two parts.

6.2. Multi-material Sedov problem

The second numerical problem we present here is a multi-material generalization of the well known Sedov problem [31].

Typically, only one quarter of the Sedov problem is solved in the domain $\langle 0, 1.1\rangle^2$, final time of the simulation is $T = 1$. The standard Sedov problem has a uniform density $\rho = 1$, pressure $p = 10^{-6}$, and ratio of specific heats $\gamma = 1.4$, the fluid is static. A high energy cell in the domain origin is set, causing an explosion generating a strong circular shock wave spreading from the origin.

In our modification, we paint 4 materials over the Cartesian computational mesh containing 32^2 cells in the domain, as shown in Fig. 11. The material interfaces are placed at radiuses $r = 0.1, r = 0.2$, and $r = 0.3$. The central material **A** represents

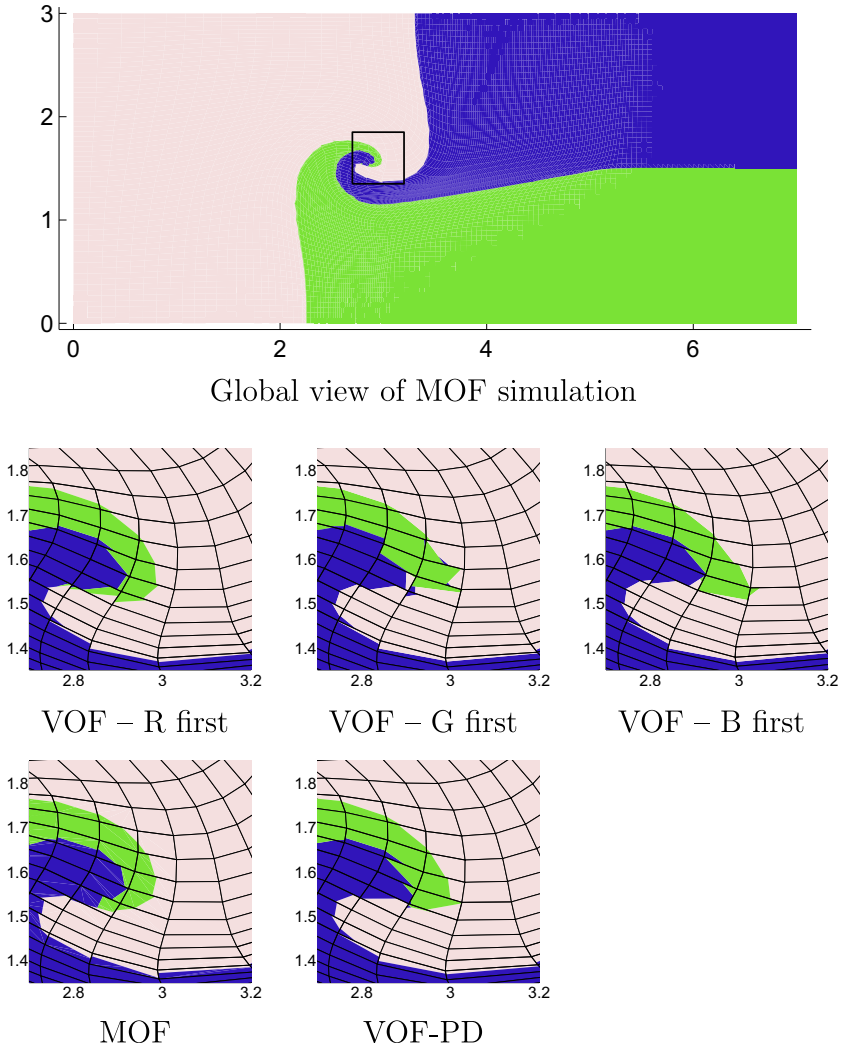


Fig. 9. Materials of triple point problem simulation, time $T = 2.5$. ALE runs (as Lagrangian step and remap to the Winslow smoothed mesh after every 20 Lagrangian steps) using different methods for material reconstruction are shown: global view on the entire computational domain for MOF method, and zooms to the three material junction for Youngs' VOF method (with different material orderings), MOF, and Power Diagram based methods are shown.

the high-energy (and also high-density) material generating the explosion, its values are $\rho = 10, p = 163.88$, and $\gamma = 1.4$. The ring **B** has a very low density $\rho = 0.2$ and a very high ratio of specific heats $\gamma = 50$, which is the simplest approach for approximating a low-compressibility material. Pressure in ring **B** is $p = 10^{-6}$. The fluid values in the high-density ring **C** are $\rho = 5, p = 10^{-6}$, and $\gamma = 5/3$. Finally, in the rest of the domain **D**, the fluid values correspond to the values of a standard Sedov problem described before.

After the simulation starts, the explosion-generated shock wave compresses the low-density material **B**. As the density of the outer ring **C** is high, it has high momentum and is difficult to start moving, which causes even stronger compression of **B**. As the fluid moves outward from the explosion, material **B** is being extended. Due to its high γ, the material gets thinner instead of decreasing of its density. At the end of the simulation, the thickness of ring **B** changes to about 20% of its original thickness.

The simulation results for different material reconstruction methods using an ALE approach performing Winslow mesh smoothing and quantity remapping after every 10 Lagrangian steps are presented in Fig. 12. For the Youngs' VOF method, the **ABCD** ordering is used, which is considered to be correct for this problem (generally, for problems including layered structures, the ordering following the materials from one side to the other one is correct). Even so, the Youngs' VOF methods

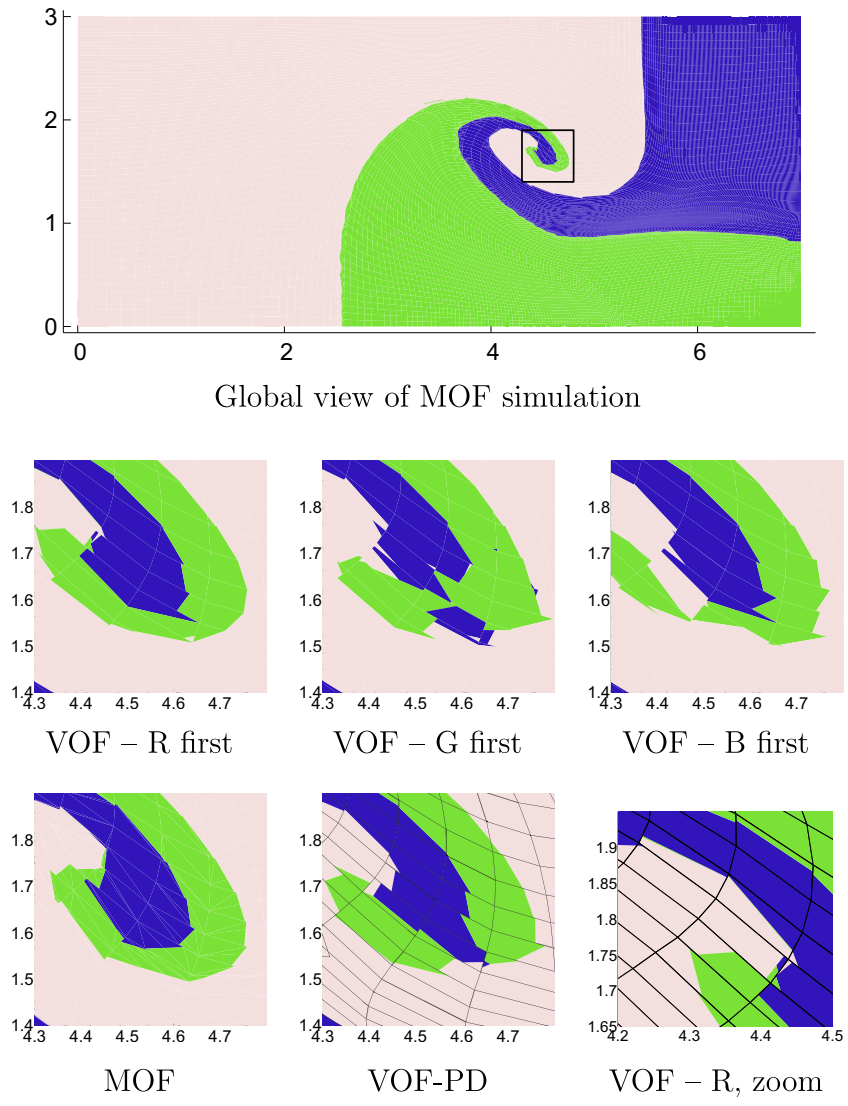


Fig. 10. Materials of triple point problem simulation, time $T = 5.0$. ALE runs (as Lagrangian step and remap to the Winslow smoothed mesh after every 20 Lagrangian steps) using different methods for material reconstruction are shown: global view on the entire computational domain for MOF method, and zooms to the three material junction for Youngs' VOF method (with different material orderings), MOF, and Power Diagram based methods are shown. A filament of green material for Youngs' VOF method with the correct material ordering is zoomed in the last image. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

produces the worst results, and the blue ring **B** is broken at several places. As we can see, the severest material displacement is present at domain boundaries due to the distortion of the volume fraction gradient here. Although, we can also observe fragments of the green material between the red and blue ones along the whole blue ring. For MOF, the blue ring stays compact with smooth interfaces. For this problem, the results obtained by the Power Diagrams based VOF method are comparable to MOF results, with the exception of the inner material interface disturbances close to the domain boundary. These are again caused by the computation of the volume fraction gradient. Let us note, that the circular shock wave position is the same and is not influenced by the particular material reconstruction method.

6.3. Multi-material Saltzman-like problem

The last problem we are going to discuss here is a modification of a standard Saltzman piston problem [32]. We use an orthogonal Cartesian 100×10 computational mesh in the computational domain $\langle -0.5, 0.5\rangle \times \langle -0.05, 0.05\rangle$. The standard Saltzman problem contains a uniform distribution of material density $\rho = 1$ and pressure $p = 2/3 \cdot 10^{-4}$ in the whole domain. The fluid is static and the ratio of specific heats is $\gamma = 5/3$. After the beginning of the simulation, the whole computational domain is compressed by a piston moving the left boundary with the unit velocity. As the simulations goes, a shock wave is formed in front of the piston, which passes the whole domain and reflects from the right boundary. It is possible to perform this simulation until quite a long time, when the shock wave reflects several times from the left and right boundaries. Let us note, that in time $T = 1$, the whole domain would be compressed to the 0 width, so this time is not reachable. This problem is often used for the investigation of the properties of Lagrangian solvers, especially when used in connection with an initially skewed computational mesh.

In our modification, we have placed several rings of different materials to the center of the computational domain, as can be seen in Fig. 13. The radiuses of the material interfaces are set to $r = 0.02, r = 0.027$, and $r = 0.03$. This problem is multi-material only formally, the fluid quantities of all materials are set to the same values as mentioned above. Therefore, the solution should exactly correspond to the 1D symmetric solution of the single-material problem.

In Fig. 14, we can see the comparison of the Youngs' VOF, VOF-PD, and MOF material reconstruction methods applied to the initial data of the described problem. For the Youngs' VOF methods, the **ABCD** material ordering was used, considered to the correct ordering for layered structures. The problem uses 100×10, 200×20, 400×40, and 800×80 mesh resolutions, in the images a zoom of the ring region is shown. For the lowest resolution, the problem is clearly under-resolved. We can

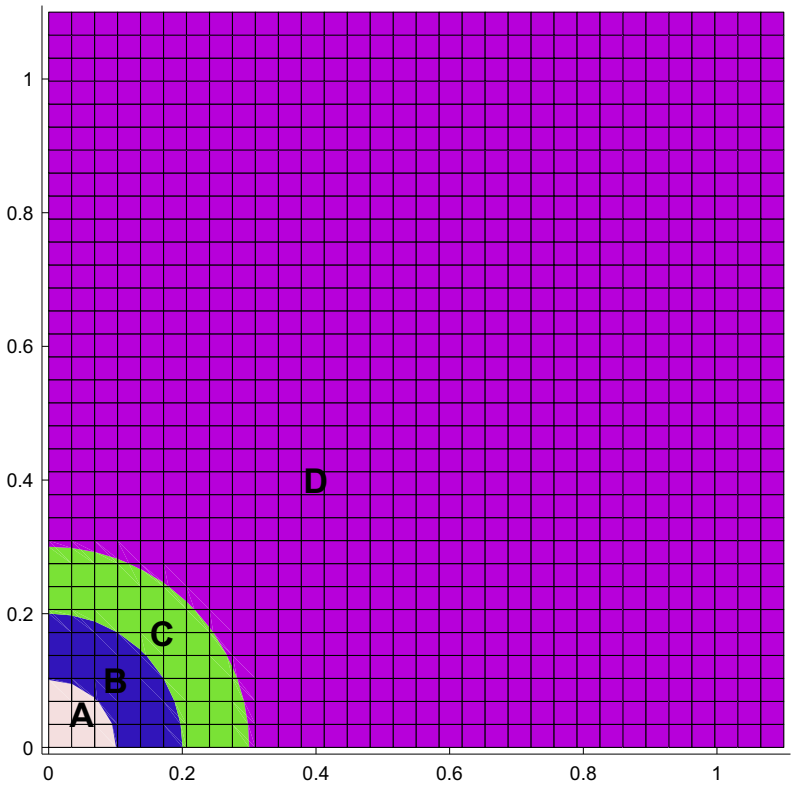


Fig. 11. Initial placement of materials for the multi-material Sedov problem painted onto a Cartesian 32^2 mesh. The material interfaces are placed at radiuses $r = 0.1, r = 0.2$, and $r = 0.3$.

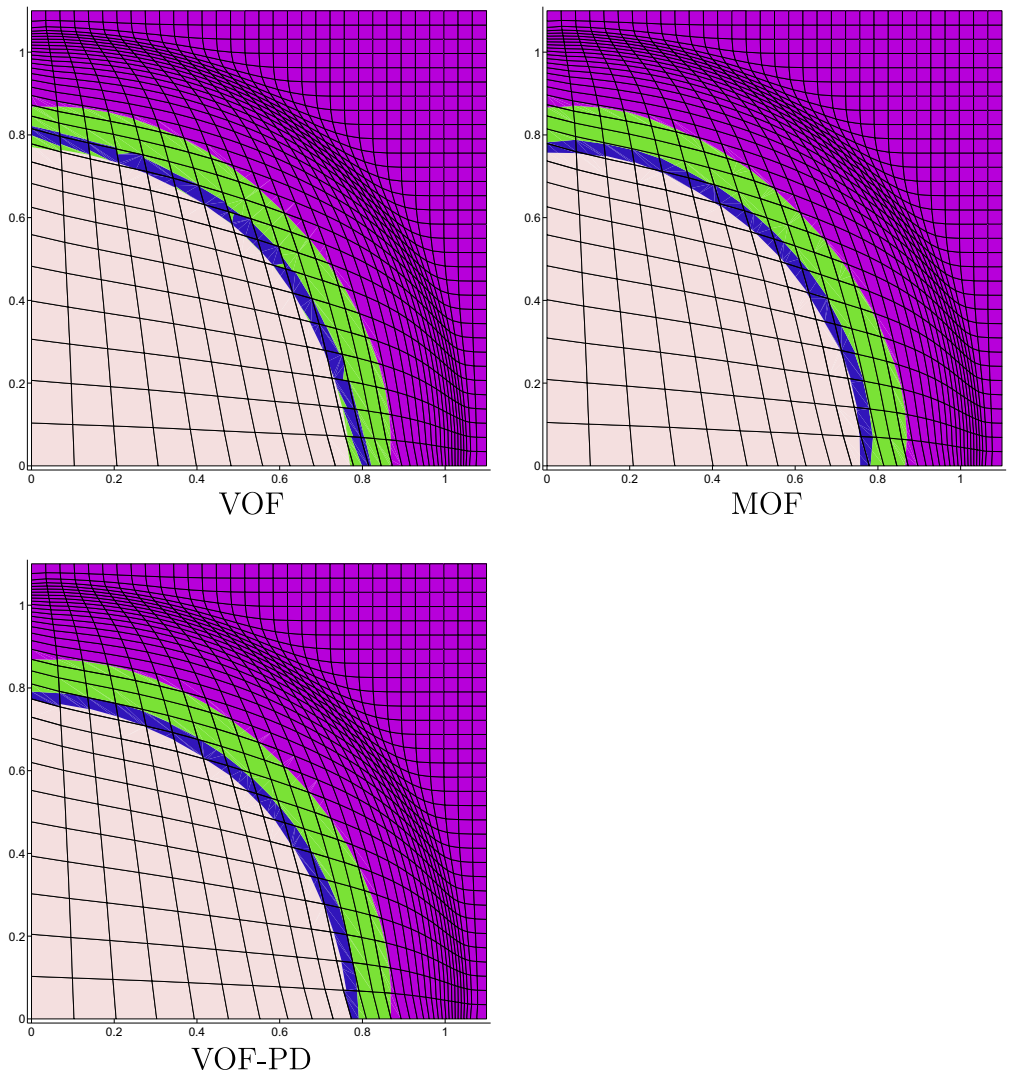


Fig. 12. Materials of multi-material Sedov problem, time $T = 1$, 32^2 mesh. ALE runs (as Lagrangian step and remap to the Winslow smoothed mesh after every 10 Lagrangian steps) using different methods for material reconstruction are shown: material interfaces for Youngs' VOF (in ABCD order), MOF, and Power Diagram based methods are shown.

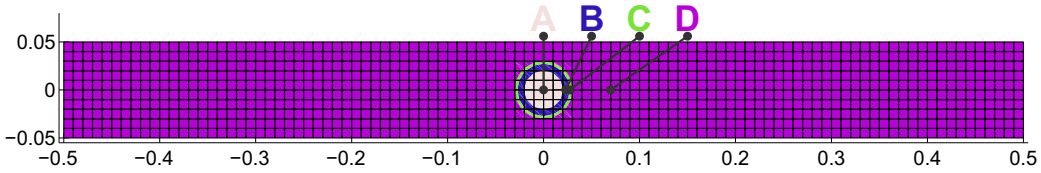


Fig. 13. Initial placement of materials for the multi-material Saltzman problem painted onto a Cartesian 100×10 mesh. The material interfaces are placed to the radiuses $r = 0.02, r = 0.027$, and $r = 0.03$.

observe severe distortions of the green ring due to inaccurate gradient computation. The gradient is computed by the Green–Gauss approach, and the stencil of the surrounding cells including the green material is not big enough for the filament structure to resolve the gradient accurately. We can even see pieces of the green material between the light-red and blue materials. The VOF-PD and MOF methods keep the material topology correctly, but the MOF method produces much smoother interfaces. The reason for non-smooth VOF-PD interfaces are the same as for the Youngs' VOF method – inaccurate

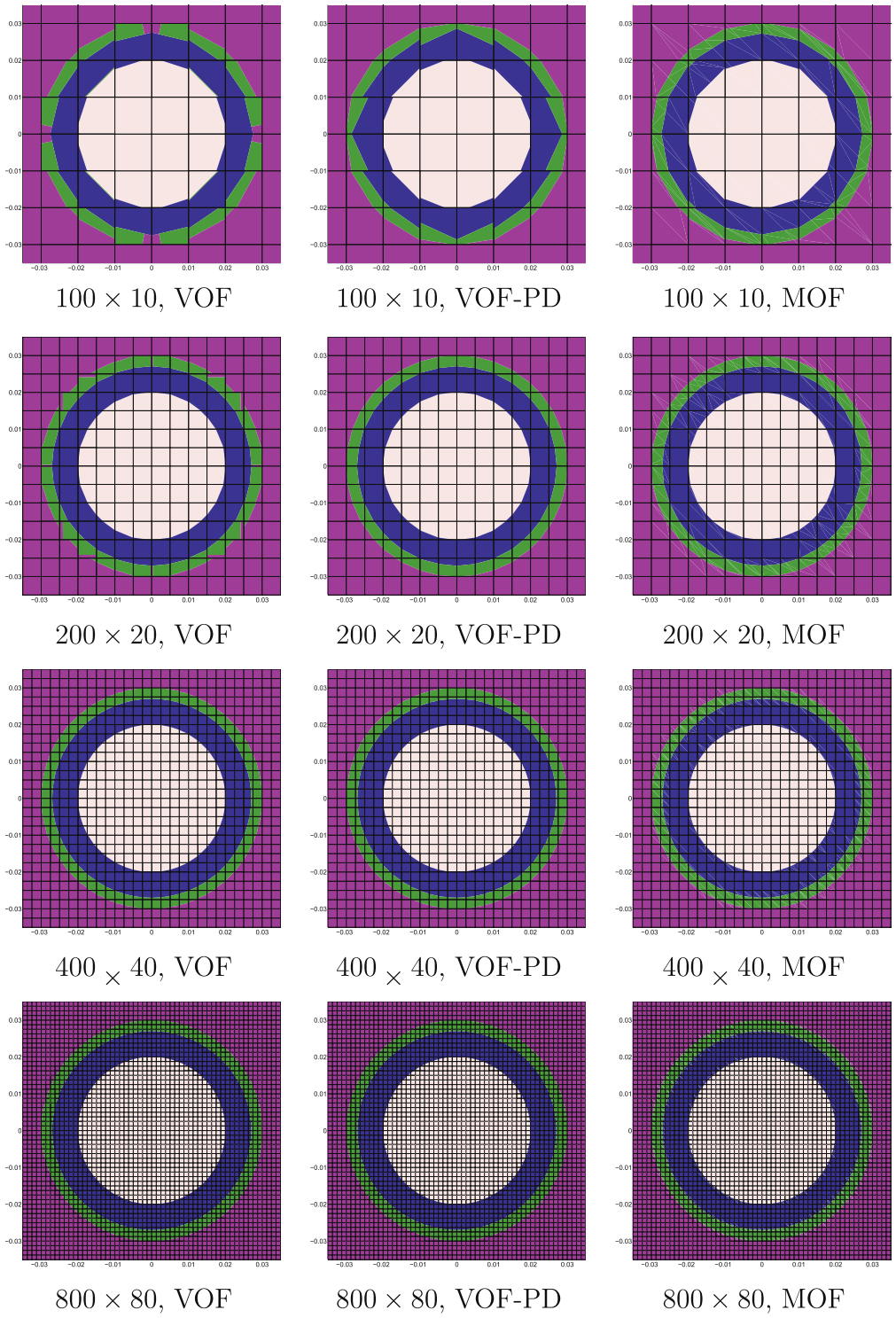


Fig. 14. Multi-material Saltzman problem in time $T = 0$ on a Cartesian meshes with different resolutions. The computational mesh and material polygons reconstructed by the different reconstruction methods in the ring regions are shown.

gradient computation. For the 200 × 20 mesh, the situation is similar. We can still observe perturbances of the green ring material due to the inaccurate gradient computation. The MOF and VOF-PD results are now comparable, all interfaces are smooth. For the 400 × 40 mesh, few tiny magenta pieces can still be found between the blue and green rings in the case

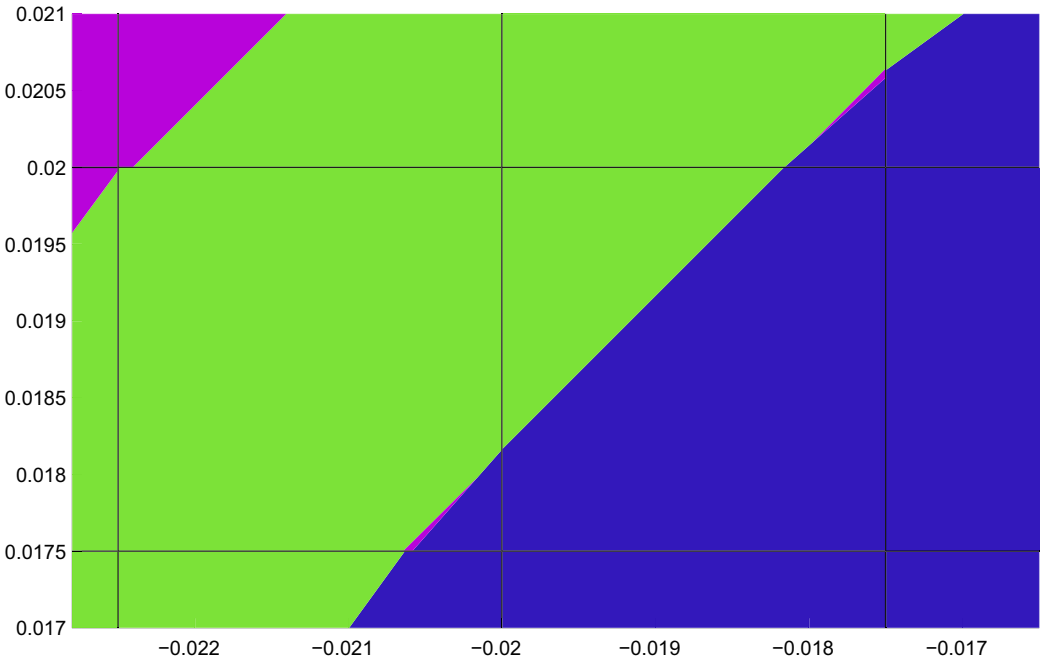


Fig. 15. Multi-material Saltzman problem in time $T = 0$ on a Cartesian 400×40 mesh. The computational mesh and material polygons reconstructed by the VOF reconstruction method are shown. Zoom shows fragments of the magenta material.

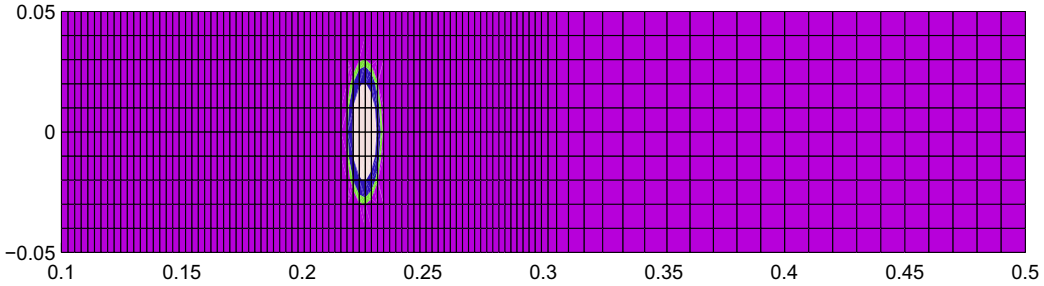


Fig. 16. Multi-material Saltzman problem in time $T = 0.6$ on a Cartesian 100×10 mesh obtained by the ALE1 simulation (Winslow mesh smoothing followed by the quantity remapping is performed after every single Lagrangian step). The computational mesh and material polygons reconstructed by the MOF reconstruction method are shown.

of VOF method. This can be seen in the zoom shown in Fig. 15. Finally, for the highest resolution mesh 800×80, all material features are at least 3 cells wide and all methods provide comparable results with smooth interfaces and correct material topology.

Let us also note, that the problems with the Youngs' VOF method could be improved in this particular problem by changing the material ordering such that the thin green filament would be treated as last. As we have already said, generally for layered structures, the ordering respecting the ordering of the layers is considered as correct and used in most simulations. Moreover, this fix is not applicable if there would be two or more filaments next to each other.

We perform the simulation of the multi-material Saltzman-like problem till the (quite an early) final time $T = 0.6$, in which the shock wave passes the rings for the first time, as can be seen for the case of MOF method on the 100×10 mesh in Fig. 16. As we can see, in this time the piston has reached the position 0.1, the shock position is about 0.3. The left part of the domain (the part behind the shock wave, including the rings) is compressed and the computational cells have the aspect ratio of about 1/4. A zoom of the ring region is shown in Fig. 17.

For comparison of different material reconstruction methods, see Fig. 18. The images show zooms to the ring regions, the aspect ratio is not preserved here. As in Fig. 14, the **ABCD** material ordering was used for the Youngs' VOF method. For the lowest 100×10 mesh resolution, both VOF and VOF-PD produce very bad results. VOF-PD surpasses the Youngs' VOF method slightly – the blue and green rings are mixed together, but stays relatively well separated from the background magenta

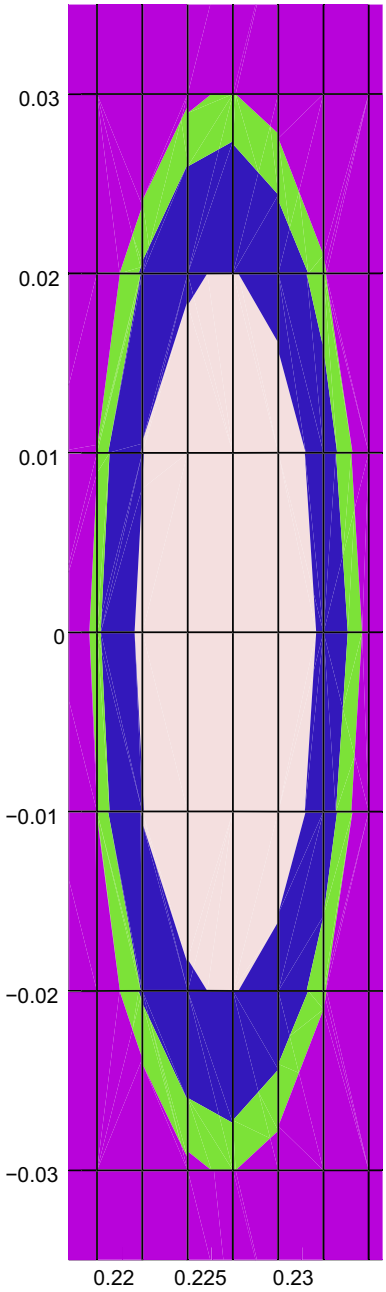


Fig. 17. Zoom-in of the multi-material Saltzman problem in time $T = 0.6$ on a Cartesian 100×10 mesh obtained by the ALE1 simulation (Winslow mesh smoothing followed by the quantity remapping is performed after every single Lagrangian step). The computational mesh and material polygons reconstructed by the MOF reconstruction method are shown.

material, contrary to the Youngs' VOF method for which all these three materials are mixed together. The problems of Youngs' VOF start in the very early stages of the simulations, as we saw in Fig. 14. We can observe severe distortions of the thin green ring due to the Green–Gauss computation of the material volume fraction gradient in the coarse mesh. Despite the coarseness of the mesh the MOF result is superior. For the finer 200×20 computational mesh, the Youngs' VOF method (with the correct material ordering) is the worst – the green ring is completely distorted, and we can see small pieces of the green and magenta materials on the blue-light red interface. The VOF-PD result is significantly better, the green ring is much more compact, but still small pieces of the magenta and blue materials can be found inside the left and right parts of the green circle. For MOF, all materials stay compact and no problems with the tiny material pieces appear. In the case of 400×40 mesh resolution, the VOF method provides better results than in the lower resolutions, but there are still fragments

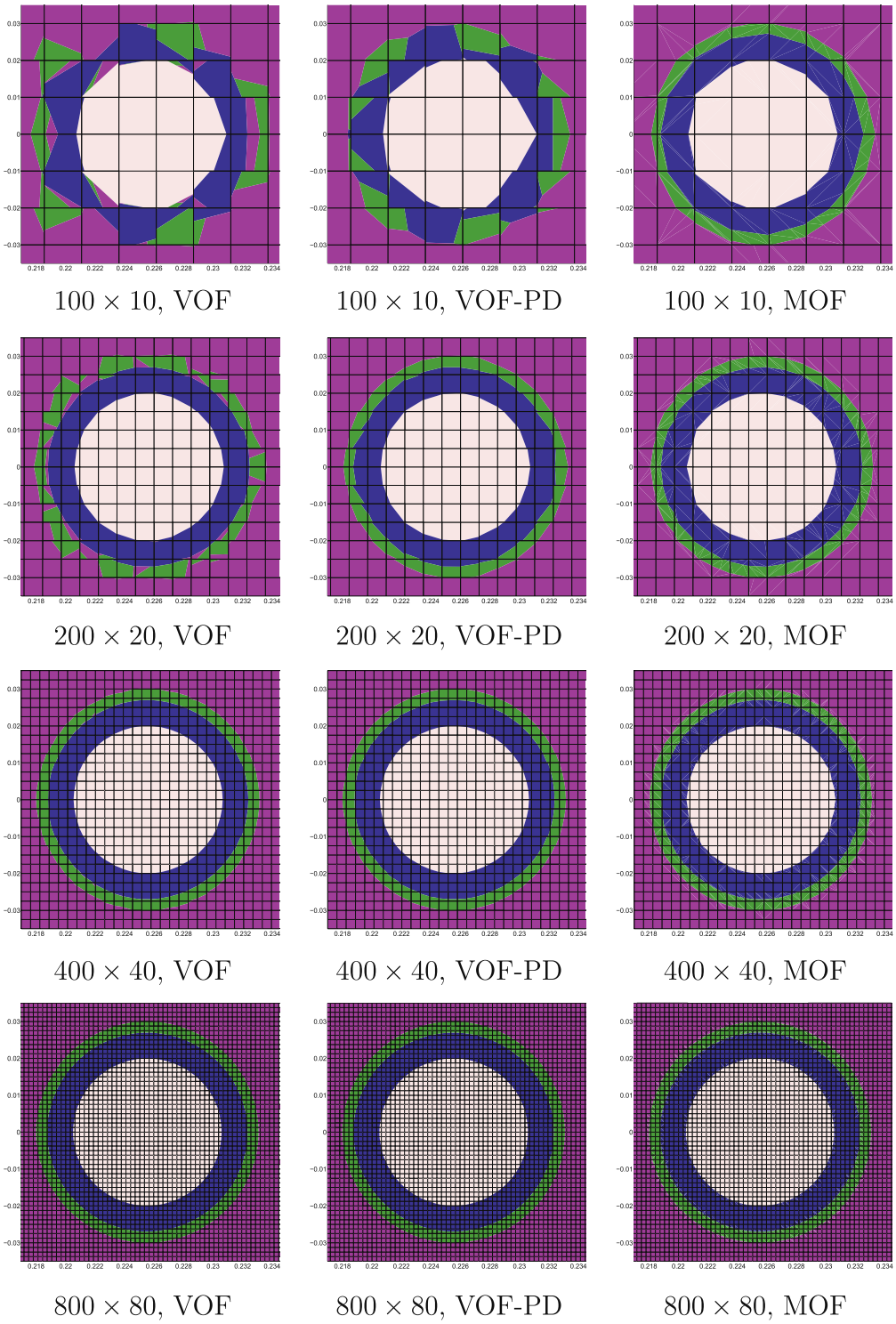


Fig. 18. Multi-material Saltzman problem in time $T = 0.6$ on a Cartesian meshes with different resolutions, obtained by the ALE1 simulation (Winslow mesh smoothing followed by the quantity remapping is performed after every single Lagrangian step.) The computational mesh and material polygons reconstructed by the different reconstruction methods in the ring regions are shown.

of the magenta material between the green and blue rings. We do not observe any material topology problems with VOF-PD or MOF method. Finally, the same set of simulations was performed on the finest 800×80 mesh. As we can see, even the thinnest green ring is now about 3 cells wide. This allow an exact computation of the gradient of the volume fractions

functions, and so the final materials are smooth, compact, and comparable to the results of the MOF method. Similarly, VOF-PD results are comparable to MOF results, materials are smooth, and no problems with material topology are visible. As we can see, all methods converge towards the same material distribution as the mesh refines.

7. Conclusions

We have presented a comparison of a material-order-dependent VOF method, a material-order-independent VOF method and a material-order-independent MOF method for a complex compressible flow involving more than two materials.

From the simulations that we have run, we conclude that:

- MOF performs the most accurate reconstructions, generally capturing filaments accurately and getting the material topology correct. Since MOF is quite recent it generally does not exist in many codes. Therefore, this method is the best choice when developing new flow codes or when revamping the interface tracking machinery. It is not advisable to introduce MOF reconstruction into a flow code without ensuring that the advection (remapping) of centroids is done accurately through overlays (exact, intersection-based remapping method).
- VOF with the correct material order performs remarkably well although the resolution of filaments and other small features is poorer than MOF. Since VOF commonly exists in flow codes that perform this type of interface tracking, it is a natural choice when the flow is simple and the material order can be predicted quite easily. It is also a good choice when the flow has only two materials and no filamentary or other structures smaller than 3–4 times the grid resolution are expected.
- Compared to VOF, the MOF method is less efficient. As it was already mentioned, all possible material combinations are tested to find the optimal material placement. If the number of materials in one cell would happen to be high (more than 5) in many cells, the MOF method could represent a considerable computational cost of the simulation. Fortunately, in the usual simulations, this situation is very rare – typically, there are many 2-material cells, some 3-material cells, and just a few 4 or more-material cells.
- VOF with power diagrams performs more poorly than MOF or VOF with the right material order but usually gets the interface topology right. This method is a good choice when the advection machinery cannot be revamped to perform overlays but the interface reconstruction can be rewritten simply to partition cells using the power diagram.
- VOF with the wrong order performs poorly even for simple flows and is not advised. If the ordering cannot be predicted or enforced strictly, it is better to use VOF with the power diagram reconstruction.

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 No. DE-AC52-06NA25396 and supported by the DOE Advanced Simulation and Computing (ASC) program. The authors acknowledge the partial support of the DOE Office of Science ASCR Program. The first author was supported in part by the Czech Ministry of Education Grant MSM 6840770022, and the Czech Grant Agency project P201/10/P086.

Appendix A. *A.1. Lagrangian update of material centroids*

The Lagrangian step may be viewed as the implicit creation of a family of maps, $\phi^n(\mathbf{x}) : \mathbb{R}^d \mapsto \mathbb{R}^d$, such that $\mathbf{x}^{n+1} = \phi^{n+1}(\mathbf{x}^n)$. Any material region, $\Omega^t \subset \mathbb{R}^d$, evolves over a time step as

$$\Omega^{n+1} = \phi^{n+1}(\Omega^n). \tag{A.1}$$

The map ϕ^{n+1} is illustrated in Fig. A.1.

If the map is an affine transformation, that is

$$\phi^{n+1}(\mathbf{x}) = A\mathbf{x} + \mathbf{b}, \tag{A.2}$$

where $A \in \mathbb{R}^{d\times d}$ is invertible and $\mathbf{b} \in \mathbb{R}^d$, then if $\mathbf{x}_c(\Omega^n)$ is the centroid of the region and $\Omega^{n+1} = \phi^{n+1}(\Omega^n)$, then $\mathbf{x}_c(\Omega^{n+1}) = A\mathbf{x}_c(\Omega^n) + b$. That is, the transformed centroid is the centroid of the transformed region.

To demonstrate this,

$$\|\Omega^{n+1}\|\mathbf{x}_c(\Omega^{n+1}) = \int_{\Omega^{n+1}} \mathbf{x}\,dx = \int_{\Omega^n} (A\mathbf{y} + b)\det A\,dy = (\det A)\|\Omega^n\|A\mathbf{x}_c(\Omega^n) + \mathbf{b}(\det A)\|\Omega^n\|.$$

Noting that

$$\|\Omega^{n+1}\| = \int_{\Omega^{n+1}} dx = \int_{\Omega^n} \det A\,dx = \|\Omega^n\|\det A,$$

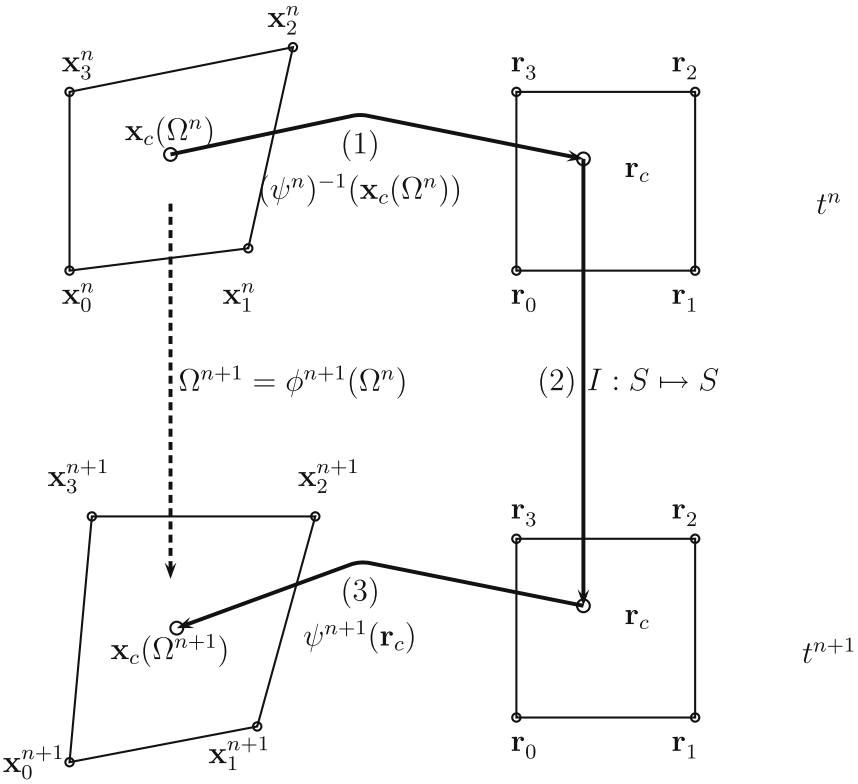


Fig. A.1. Steps in the constant parametric coordinate method. (1) The logical coordinates of the centroid at time t^n are calculated. (2) It is assumed the centroid has the same logical coordinates at time t^{n+1}. (3) The logical coordinates are mapped to physical coordinates to give the location. This gives a second order accurate approximation to the centroid of the evolved region $\Omega^{n+1} = \phi^{n+1}(\Omega^n)$.

we obtain,

$$\mathbf{x}_c(\Omega^{n+1}) = A\mathbf{x}_c(\Omega^n) + \mathbf{b}.$$

The actual Lagrangian evolution of the region is given by the pointwise equation

$$\frac{d\mathbf{x}}{dt} = \mathbf{u}(\mathbf{x}, t) \quad \forall \mathbf{x} \in \Omega^t \tag{A.3}$$

assuming the velocity field is known. The transformation, ϕ^{n+1}, is then the solution to Eq. (A.3) over the time interval $[t^n, t^{n+1}]$.

With sufficient regularity, the velocity field can be expanded as

$$u_j(\mathbf{x}, t) = u_j(\mathbf{x}_0, t^n) + (t - t^n)\frac{\partial u_j(\mathbf{x}_0, t^n)}{\partial t} + (x_i - x_i^0)\frac{\partial u_j(\mathbf{x}_0, t^n)}{\partial x_i} + \mathcal{O}(\Delta x^2) + \mathcal{O}(\Delta t^2) + \mathcal{O}(\Delta x \Delta t). \tag{A.4}$$

Substituting this into Eq. (A.3) and integrating, we find that

$$\phi^{n+1}(\mathbf{x}) = \mathbf{x} + \mathbf{u}(\mathbf{x}_0, t^n)\Delta t + \mathcal{O}(\Delta t^2) + \mathcal{O}(\Delta t \Delta x). \tag{A.5}$$

Assuming $\Delta t \approx \Delta x$, then the transformation defining the Lagrangian evolution over a time step may be approximated as an affine transformation with second order accuracy.

A.2. Constant parametric coordinate method

A method for updating material centroids during a Lagrangian step can exploit this implicit evolution operator described above. The method described [2,19] is based on the existence of a mapping of the computational cell to and from a logical space. It is assumed that the centroid of the material region has the same logical coordinates, before and after the Lagrangian motion of the cell. To obtain the centroid after the Lagrangian motion, the logical coordinates of the centroid at the previous step are given to the logical to physical mapping corresponding to the cell after the motion. This process is illustrated in Fig. A.1. It is important to note that the logical to physical space mapping is different for each time step and the cells evolve in time.

The accuracy of the method relies on the properties of the logical to physical coordinate transformations used. Assume each cell has local coordinates, $\mathbf{r} \in S$, with an invertible map into physical coordinates, $\psi^n : S \mapsto \Omega^n$.

We define a family of local parameterizations, $\{\psi^n\}$ to be **linearity preserving**, if points from the parametric space, S, are mapped such that if

$$\mathbf{x}^{n+1} = A\mathbf{x}^n + b, \tag{A.6}$$

then if $\mathbf{x}^n = \psi^n(\mathbf{r})$,

$$\mathbf{x}^{n+1} = \psi^{n+1}(\mathbf{r}) = A\psi^n(\mathbf{r}) + b = A\mathbf{x}^n + \mathbf{b}. \tag{A.7}$$

Equivalently,

$$\psi^{n+1} = A\psi^n + \mathbf{b}. \tag{A.8}$$

The bilinear parameterization of quads satisfies this property: the two orthogonal coordinates, $(r,s) \in [0,1]^2$ linearly interpolate the vertices (see Fig. A.1 for node numbering)

$$\psi^n(r,s) = (1-r)\left[(1-s)\mathbf{x}_0^n + s\mathbf{x}_3^n\right] + r\left[(1-s)\mathbf{x}_1^n + s\mathbf{x}_2^n\right]. \tag{A.9}$$

Clearly, $\psi^{n+1} = A\psi^n + \mathbf{b}$ as $\mathbf{x}_j^{n+1} = A\mathbf{x}_j^n + \mathbf{b}$ for $j = 0,\ldots,3$. The generalized barycentric coordinates of polygon with vertices $\{\mathbf{v}_i\}$ also satisfies the linearity preserving property. To demonstrate this, barycentric coordinates satisfy the properties [33],

$$\mathbf{x} = \sum_i \lambda_i \mathbf{v}_i = \psi^n(\lambda), \tag{A.10}$$

$$\sum_i \lambda_i = 1, \tag{A.11}$$

$$\lambda_i \geqslant 0. \tag{A.12}$$

If $\mathbf{x}$ has barycentric coordinates $\boldsymbol{\lambda}$, then if $\mathbf{x}^n = \psi^n(\lambda)$,

$$A\mathbf{x} = \sum_i \lambda_i A\mathbf{v}_i, \tag{A.13}$$

$$A\psi^n(\lambda) + \mathbf{b} = \sum_i \lambda_i A\mathbf{v}_i + \mathbf{b}\sum_i \lambda_i, \tag{A.14}$$

$$A\psi^n(\lambda) + \mathbf{b} = \sum_i \lambda_i (A\mathbf{v}_i + \mathbf{b}) = \psi^{n+1}(\boldsymbol{\lambda}), \tag{A.15}$$

where $\sum_i \lambda_i = 1$ was utilized in the second step.

If the family of transformations satisfy the linearity preserving property, then we may analyze the accuracy of the constant parametric coordinate method. If the parameterization family, $\{\psi^n\}$, is linearity preserving, then updating the location of a material centroid by assuming its parametric coordinates are unchanged is exact for linear motions, since for an arbitrary subdomain mapped with an affine transformation,

$$\mathbf{x}_c(\Omega^{n+1}) = A\mathbf{x}_c(\Omega^n) + \mathbf{b}. \tag{A.16}$$

If the transformation is linearity preserving, then

$$\mathbf{x}_c(\Omega^{n+1}) = \varphi^{n+1}(\mathbf{r}) = A\varphi^n(\mathbf{r}) + \mathbf{b} = A\mathbf{x}_c(\Omega^n) + \mathbf{b}. \tag{A.17}$$

In general, the Lagrangian motion will not be linear. However, as was shown in the previous section, for sufficient regularity in time an affine approximation to the Lagrangian motion is second order accurate.

References

[1] D.J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Computer Methods in Applied Mechanics and Engineering 99 (1992) 235–394.
[2] D.J. Benson, Volume of fluid interface reconstruction methods for multi-material problems, Applied Mechanics Review 55 (2) (2002) 151–165.
[3] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, Journal of Computational Physics 141 (1998) 112–152.
[4] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annual Review of Fluid Mechanics 31 (1999) 567–603.
[5] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, Journal of Computational Physics 39 (1981) 201–225.
[6] V. Dyadechko, M.J. Shashkov, Reconstruction of multi-material interfaces from moment data, Journal of Computational Physics 11 (2008) 5361–5384.
[7] W.F. Noh, P. Woodward, SLIC (simple line interface calculation), in: A.I. van der Vooren, P.J. Zandbergen (Eds.), Fifth International Conference on Numerical Methods in Fluid Dynamics, Springer-Verlag, 1976, pp. 330–340.
[8] D.L. Youngs, Time dependent multi-material flow with large fluid distortion, in: K.W. Morton, M.J. Baines (Eds.), Numerical Methods for Fluid Dynamics, Academic Press, 1982, pp. 273–285.
[9] D.L. Youngs, An interface tracking method for a 3D Eulerian hydrodynamics code, Technical Report AWE/44/92/35, AWRE Design and Math Division, 1984.
[10] J.E. Pilliod Jr., E.G. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, Journal of Computational Physics 199 (2004) 465–502.
[11] B.K. Swartz, The second-order sharpening of blurred smooth borders, Mathematics of Computation 52 (186) (1989) 675–714.
[12] H. Steinhaus, A note on the ham sandwich theorem, Mathesis Polska 9 (1938) 26–28.

[13] A.H. Stone, J.W. Tukey, Generalized "sandwich" theorems, Duke Mathematical Journal 9 (1942) 356–359.
[14] S. Mosso, B.K. Swartz, D.B. Kothe, A parallel, volume-tracking algorithm for unstructured meshes. In: P. Shiano, A. Ecer, J. Periaux, N. Satofuka (Eds.), Parallel Computational Fluid Mechanics, Elsevier, 1997, pp. 368–375.
[15] R.V. Garimella, V. Dyadechko, B.K. Swartz, M.J. Shaskov, Interface reconstruction in multi-fluid, multi-phase flow simulations, in: Proceedings of the 14th International Meshing Roundtable, San Diego, CA, September 2005, Springer, pp. 19–32.
[16] S.P. Schofield, R.V. Garimella, M.M. Francois, R. Loubere, Material order-independent interface reconstruction using power diagrams, International Journal for Numerical Methods in Fluids 56 (6) (2007) 643–659.
[17] H.T. Ahn, M. Shashkov, Multi-material interface reconstruction on generalized polyhedral meshes, Journal of Computational Physics 226 (2007) 2096–2132.
[18] S.P. Schofield, R.V. Garimella, M.M. Francois, R. Loubere, A second-order accurate material-order-independent interface reconstruction technique for multi-material flow simulations, Journal of Computational Physics 228 (3) (2009) 731–745.
[19] D.J. Benson, Eulerian finite element methods for the micromechanics of heterogeneous materials: Dynamic prioritization of material interfaces, Computer Methods in Applied Mechanics and Engineering 151 (1998) 343–360.
[20] F. Aurenhammer, Power diagrams: properties, algorithms and applications, SIAM Journal on Computing 16 (1) (1987) 78–96.
[21] H. Imai, M. Iri, K. Murota, Voronoi diagram in the Laguerre geometry and its applications, SIAM J on Computing 14 (1) (1985) 93–105.
[22] E.J. Caramana, D.E. Burton, M.J. Shashkov, P.P. Whalen, The construction of compatible hydrodynamics algorithms utilizing conservation of total energy, Journal of Computational Physics 146 (1998) 227–262.
[23] E.J. Caramana, M.J. Shashkov, P.P. Whalen, Formulations of artificial viscosity for muti-dimensional shock wave computations, Journal of Computational Physics 144 (1998) 70–97.
[24] G. Scovazzi, E. Love, M. Shashkov, Multi-scale Lagrangian shock hydrodynamics on Q1/P0 finite elements: Theoretical framework and two-dimensional computations, Computer Methods in Applied Mechanics and Engineering. 197 (2008) 1056–1079.
[25] M. Shashkov, Closure models for multidimensional cells in arbitrary Lagrangian–Eulerian hydrocodes, International Journal for Numerical Methods in Fluids 56 (2008) 1497–1504.
[26] A.M. Winslow, Equipotential zoning of two-dimensional meshes, Technical Report UCRL-7312, Lawrence Livermore National Laboratory, 1963.
[27] L.G. Margolin, M. Shashkov, Second-order sign-preserving conservative interpolation (remapping) on general grids, Journal of Computational Physics 184 (2003) 266–298.
[28] 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, Technical Report UCRL-JC-139820, Lawrence Livermore National Laboratory, 2000.
[29] M. Kucharik, M. Shashkov, Conservative multi-material remap for staggered discretization. in preparation.
[30] P-H. Maire, Personal Communication, Los Alamos National Laboratory, 2007.
[31] L.I. Sedov, Similarity and Dimensional Methods in Mechanics, Academic Press, 1959.
[32] J.K. Dukowicz, B.J.A. Meltz, Vorticity errors in multidimensional Lagrangian codes, Journal of Computational Physics 99 (1) (1992) 115–134.
[33] J. Warren, S. Schaefer, A.N. Hirani, M. Desbrun, Barycentric coordinates for convex sets, Advances in Computational Mathematics 27 (3) (2007) 319–338.

Computers & Fluids 46 (2011) 293–297

Contents lists available at ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

Hybrid remap for multi-material ALE ☆

M. Kucharik [a,*], J. Breil [b], S. Galera [b], P.-H. Maire [b], M. Berndt [c], M. Shashkov [d]

[a] Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Brehova 7, Praha 1, 115 19, Czech Republic
[b] UMR CELIA CEA–CNRS–Université Bordeaux I, 33 405 Talence Cedex, France
[c] CCS-2 Group, MS D413, Los Alamos National Laboratory, P.O. Box 1663, Los Alamos, NM 87545, USA
[d] XCP-4 Group, MS B284, Los Alamos National Laboratory, P.O. Box 1663, Los Alamos, NM 87545, USA

ARTICLE INFO

Article history:
Received 27 April 2010
Received in revised form 15 July 2010
Accepted 5 August 2010
Available online 10 August 2010

Keywords:
Multi-material ALE
Conservative interpolations
Hybrid remap

ABSTRACT

Remapping is one of the essential parts of most arbitrary Lagrangian–Eulerian (ALE) methods. In this short paper we focus on multi-material fluid flows. We present a hybrid remapping method combining the swept remapping algorithm in pure regions with the intersection-based remapping algorithm close to material interfaces. We describe the hybrid remapping method in two formulations, as a one-step and a two-step procedure and compare behaviour of both approaches with the standard intersection-based algorithm using several numerical examples.

1. Introduction

There exist two basic approaches for hydrodynamic simulations – Eulerian and Lagrangian methods. Eulerian methods utilize a static computational mesh and the fluid flows in the form of mass flux through it. In the Lagrangian framework, the computational mesh moves with the fluid and naturally follows the changing computational domain. Therefore, it is more suitable for certain types of applications including severe compressions and expansions. Due to the mesh motion, its tangling or degeneration can occur causing the failure of the simulation. In the pioneering paper [4], the Arbitrary Lagrangian–Eulerian (ALE) method was introduced, combining best properties of both approaches. The computational mesh follows the physical domain due to the embodied Lagrangian solver, while the Eulerian part (consisting of mesh rezoning followed by remapping, interpolating conservatively all fluid quantities from the Lagrangian to the rezoned mesh) keeps it smooth. The ALE approach became very popular, and many authors contributed to this topic [1,10,5,12]. For real life applications, multi-material ALE must be used, allowing more materials to share the same computational cell.

We focus on the remapping stage of the multi-material ALE algorithm. There exist generally two approaches based on (1) approximate (swept) fluxes and (2) intersections. In simulations with only one material, the simple (and fast) swept approximation can be used, integrating their piecewise-linearly reconstructed densities in the swept region, defined by the motion of the particular mesh edge in the rezoning stage, see for example [9,11,8]. In the case when several materials are present, the rezoned mesh will include mixed cells, and the classical (intersection-based) approach must be used, in which pure material polygons of the Lagrangian mesh are intersected with the new mesh cells and summed up to the total value of each fluid quantity in the new cells. As intersections are used, this approach can become rather expensive in 2D and almost impossible to use in 3D. The intersection-based remapper can be also formulated in a flux form [11].

In this paper, we describe a new method that we call hybrid remapping, combining both approaches. It incorporates the cheap swept approach in pure regions and employs intersections only in the regions where more materials are present. Hence, intersections are avoided in pure regions that often cover most of the domain. This hybrid remapping approach may decrease the cost of the multi-material remap.

The rest of the paper is organized as follows. In Section 2, the two-step hybrid remapper is discussed. It is based on a clever trick – in the first step, only pure regions are rezoned and remapped by the swept approach, while in the second step, mixed regions are treated by the intersection-based approach. In Section 3, the one-step hybrid remapping approach is discussed, in which the mass (and other) fluxes are constructed at once, by the combination of the swept and intersection-based fluxes. In Section 4, the numerical errors and time costs of both approaches are compared with the standard methods.

☆ LA-UR-10-01861

* Corresponding author.
E-mail address: kucharik@newton.fjfi.cvut.cz (M. Kucharik).

0045-7930/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2010.08.004

2. Two-step hybrid remap

In the two-step hybrid remapping method, treatment of pure and mixed cells is separated into two distinct phases. At the beginning of the rezone/remap stage, every mesh node must be marked as pure or mixed. If all cells attached to a node are pure and contain the same material, the node is pure, otherwise it is marked as mixed. For an example of the marking process, see image (a) of Fig. 1, where two materials separated by a straight interface are remapped from a randomly perturbed to the equidistant orthogonal mesh.

In the first (pure) stage, only pure nodes are moved in the rezoning process and the swept remapping is used, see image (b) of Fig. 1. As all pure nodes are surrounded by pure cells only, no mixed cell is affected by the mesh motion and therefore no multi-material cells participate in the swept remap. In the second (mixed) stage, mixed nodes are rezoned and the intersection-based remapper follows, capable of remapping fluid quantities in the presence of multiple materials, see image (c) of Fig. 1. Hence, the existing swept and intersection-based remappers are naturally combined without doing anything special between the pure and mixed cells. On the other hand, the buffer cells (layer of pure cells attached to the mixed ones) are treated in both steps, increasing the overhead cost of the two-step hybrid method. For more details on the two-step hybrid remapper and its cost analysis, see [2].

3. One-step hybrid remap

The one-step hybrid remap [7] computes all fluxes at once for quadrilateral meshes. It combines the swept volume fluxes

$$V_{\tilde{c}} = V_c + \sum_{e\in\partial c} \Omega_e \delta V_e = V_c + \sum_{c'\in C(c)} F^{V,\text{swept}}_{c,c'} \tag{1}$$

with the intersection-based material volume fluxes [11]

$$V_{\tilde{c},k} = V_{c,k} + \sum_{c'\in C(c)} \left(V_{\tilde{c}\cap c',k} - V_{c\cap\tilde{c}',k}\right) = V_{c,k} + \sum_{c'\in C(c)} F^{V,\text{exact}}_{c,c',k}. \tag{2}$$

Here, c represents a cell in the Lagrangian mesh, $\tilde{c}$ is the corresponding rezoned cell, k is the material index, e is an edge of cell c and $C(c)$ is a set of all neighbors of cell c. δV_e is the swept volume corresponding to edge e, defined by the motion of its vertices to the rezoned positions, and Ω_e is equal to either +1 or −1, depending on the orientation of edge e in cell c. The swept fluxes are equal to zero for corner neighbors, while the intersection-based corner fluxes are present. Let us note that no additional corner coupling error is created, there is no corner flux present in the classical swept approach, while in the exact integration method it is present. The same formulas as in (1) and (2) can be used for the computation of fluxes of other simple integrals, $\int x$ and $\int y$, instead of the flux volume $V = \int 1$. All these integrals can be precomputed at the beginning of the remapping step and reused for the computation of fluxes of all quantities.

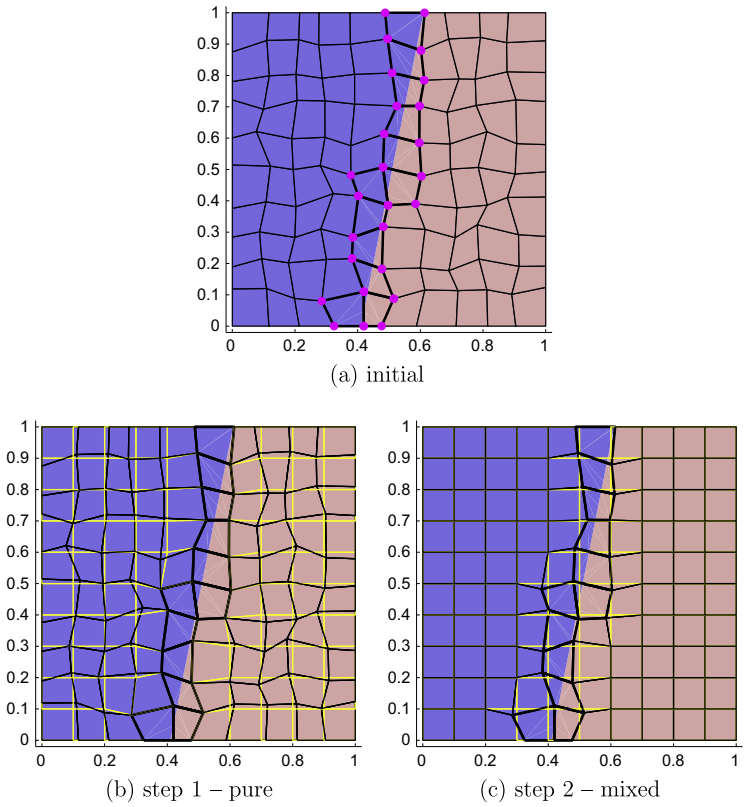


Fig. 1. Straight interface dividing two materials on a randomly perturbed quadrilateral mesh. (a) Initial situation, mixed cells marked by thick edges, mixed nodes by magenta circles. (b) Pure step – pure nodes moved, doing swept remap, new mesh in yellow edges. (c) Mixed step – mixed nodes moved, doing intersection-based remap. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The pure/mixed nodes are marked as in the two-step hybrid remapper. Next, edge fluxes are marked. An edge flux is pure if both involved vertices are pure and mixed otherwise, see image (a) of Fig. 2. A corner flux is mixed if the involved vertex is mixed and it is marked as hybrid otherwise. The remapping process sweeps through the cells and through all fluxes in each cell (including the corner ones). For pure/mixed edges, the fluxes are computed by swept/intersection-based remap, as required. For mixed corner fluxes, they are computed by intersections. The remaining hybrid corner fluxes are treated in the following way. If both cell edges attached to this node (+ and − in image (b) of Fig. 2) are swept, the corner flux is ignored as there is no discrepancy to be fixed. In the opposite case (c), there is a discrepancy between the swept and exact edge fluxes shown as the yellow triangles in image (d) of Fig. 2. In this case, we mark the opposite mesh edges by symbols = and ‡. In the case shown in image (d) of Fig. 2, we compute the intersection of the new edge $\bar{e}_{\ddagger}$ with the old edge e_- (marked in (d) by magenta crosses), construct the triangle surrounded by the intersection and the old and new position of node n, integrate over this triangle and add this hybrid flux into the mixed flux through e_-. The described process helps in most situations with just one exception, when the sideward swept region is non-convex and passes through the node, as shown in Fig. 3. This situation is easily detected – only one of four possible mixed intersections $e_+ \cap \bar{e}_=$, $\bar{e}_+ \cap e_=$, $e_- \cap \bar{e}_{\ddagger}$ and $\bar{e}_- \cap e_{\ddagger}$ exists instead of 2 in the standard situation from Fig. 2. As before, this triangle must be added to the involved (possibly existent sideward mixed) flux as before and two cases can occur. If the hybrid node moves outward of the mixed flux (images (a) and (b) of Fig. 3), the vertically and horizontally dashed yellow triangles contributed to the mixed flux when treating it from both attached cells, c and c'. These triangles overlap (double dashed yellow triangle in image (b) of Fig. 3), so this triangle must be recovered by connecting the old and new nodal position with the extra same-sign intersection $e_+ \cap \bar{e}_{\ddagger}, \bar{e}_+ \cap e_{\ddagger}, e_- \cap \bar{e}_=$ or $\bar{e}_- \cap e_=$, whichever exists. This contribution must be removed from the mixed edge flux. In the opposite case shown in images (c) and (d) of Fig. 3, the hybrid node moves inward, no triangles were added to the mixed flux (they belong to the possible sideward edge flux) and a part of the swept and intersection fluxes overlap. The overlapping triangle is bounded by the nodal coordinates and the existing same-sign intersection and this contribution must be removed from the mixed flux. This process ensures smooth transition of the intersection-based fluxes to the swept fluxes in the buffer region.

4. Numerical examples

Here, we present several numerical examples performed using our multi-material remapping research code. All tests start on an

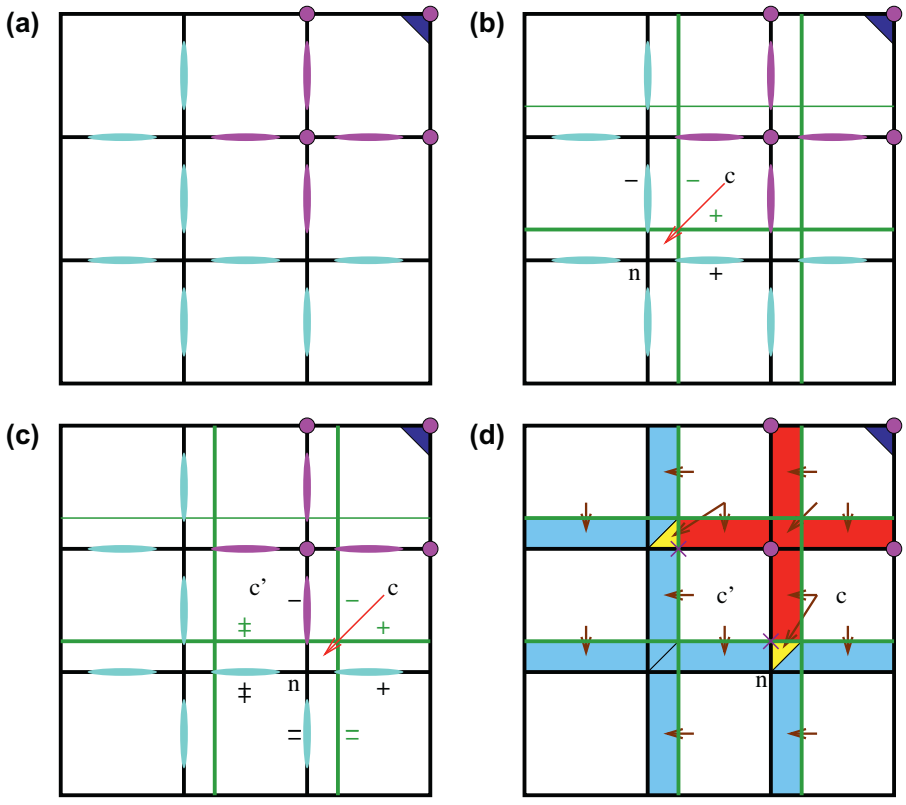


Fig. 2. One mixed cell in a multi-material mesh (a) – material segment shown as dark blue triangle, mixed nodes highlighted by magenta circles and mixed and pure edge fluxes highlighted by magenta and cyan ellipses. Pure node n viewed from cell c (b), as shown by the red arrow, treated by a purely swept approach. Edges of old (black) new (green) meshes annotated by + and − signs with respect to cell c. Hybrid node n viewed from cell c (c), cell c' is neighbor over the hybrid-modified edge. Fluxes (d) computed either by swept movement (light blue) or intersections (red), yellow triangles represent the hybrid correction of the intersection-based fluxes. Brown arrows show, from which cell is the reconstruction taken to construct the flux. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

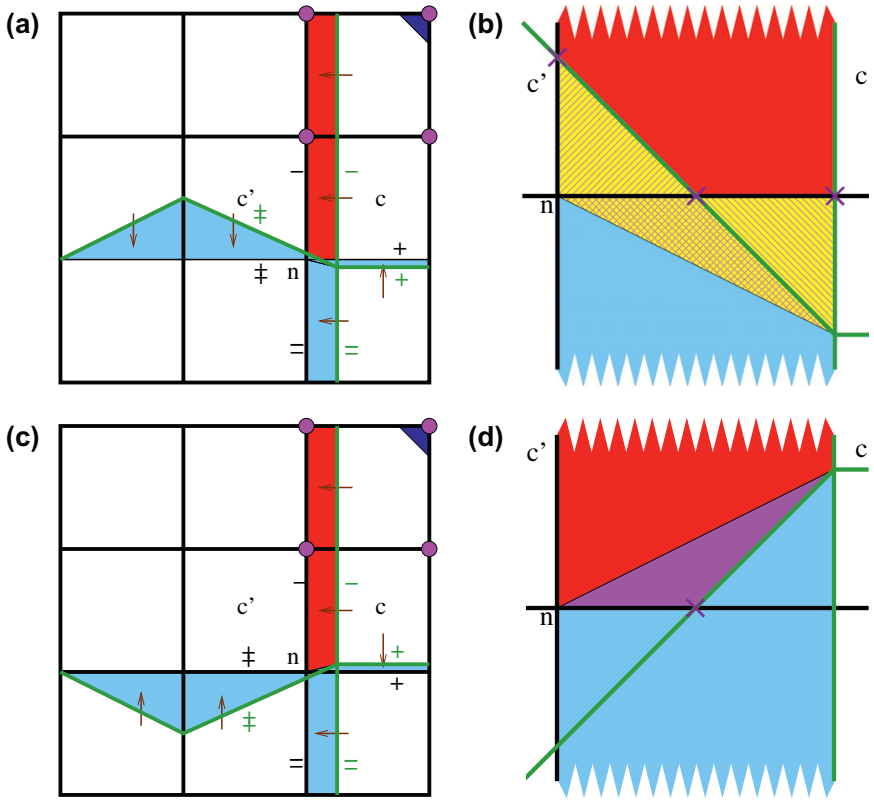


Fig. 3. Similar situation as in Fig. 2, exceptional situations originated from self-overlapping swept region passing through the hybrid node. Situation (a) – hybrid node moving outward from the intersection treated edge. Sketch (b) (showing only fluxes corresponding to vertical edges) with old and new edge intersections highlighted by magenta crosses and the hybrid corrections from the yellow triangles (two additional left and right triangles, which overlap and form the central triangle). Situation (c) – hybrid node moving toward the intersection treated edge. Sketch (d) shows the purple triangle where the swept and intersection fluxes overlap. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

equidistant orthogonal mesh in $\langle 0,1\rangle^2$, 100 random mesh movements followed by the remapping process are performed and in the last step, the data are remapped back to the initial mesh. The random mesh motion is used as it introduces all possible combinations of edge movements through the material interface and therefore tests the consistency of the remapper. Only material density, volume fractions and material centroids are remapped in the described flux form. For more details on centroid remap, see [2,6]. The MOF method [3] was used for material reconstruction in all tests. All tests were performed on a 2.7 GHz AMD Opteron computer.

In the first test, we have remapped a single-material global linear function on meshes with resolutions from 16×16 to 512×512 cells. All tested methods (intersections, one-step, two-step hybrid and even the swept method in this single-material test) prove exact preservation of the linear profile up to the machine accuracy. The time costs (in logarithmic scale) are shown in the upper image of Fig. 4. We see that the cost of the hybrid methods is very close to the swept approach, while the one-step hybrid cost almost coincides with the swept cost. The exact intersection-based remapper is significantly more expensive.

In the second test, we have remapped two distinct linear functions in two materials separated by a straight interface, as shown in Fig. 1. We omit the swept method in multi-material tests. All methods show exact preservation of a multi-material linear function, even in mixed cells.

In the last test, we define a non-linear density function $10+\left(x-\frac{1}{2}\right)^2+\left(y-\frac{1}{2}\right)^2$ inside a circle of radius $\frac{1}{4}$ in the center of the domain and $10+e^{2xy}$ otherwise. The central circle includes a different material than the rest of the domain, the material interface is therefore curved. The computational cost is shown in the lower image of Fig. 4. We can see the benefit of the hybrid approaches when compared with the costly intersections. In Table 1, the numerical errors of material density $\rho_{c,k}$ with respect to the analytic density $\left(L_1^{\mathrm{mat}}=\sum_{\forall c}\sum_{\forall k\in c}|\rho_{c,k}-\rho(x_{c,k},y_{c,k})|\,V_{c,k}/\sum_{\forall c}\sum_{\forall k\in c}\rho(x_{c,k},y_{c,k})V_{c,k}\right)$ are shown. Errors originate from two sources – non-linear density and the MOF error due to the curved interface. We observe second order accuracy for all methods and their errors are comparable.

5. Conclusions

We have described two approaches for combining the swept and intersection-based fluxes in the presence of multiple materials, decreasing the computational cost of the remapper due to costly intersections. To add the two-step hybrid method into an existing code, one does not need to do any work on the side of the remapper, changes in the main routine logic are, however, required. The opposite is the case for the one-step hybrid remapper. Both methods show significant cost improvement when compared with the intersection-based approach and exhibit numerical errors that are comparable with existing methods for both linear and non-linear densities.

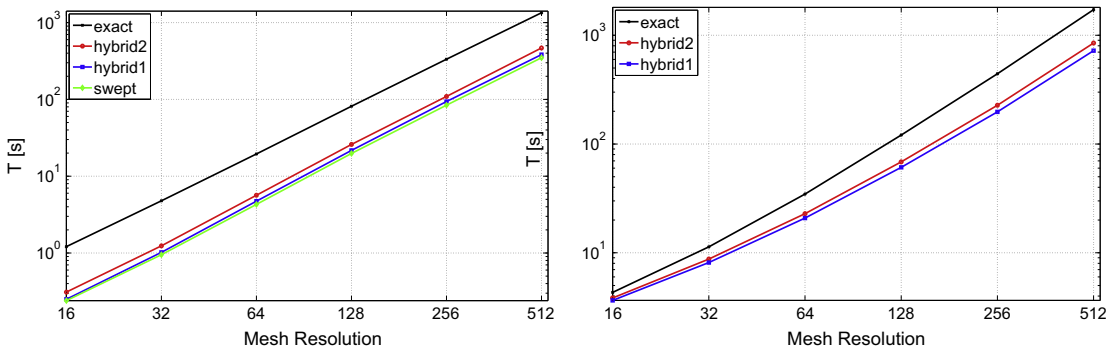


Fig. 4. Time of simulation (in s) of 100 remapping steps between randomly changing meshes of different resolutions, performed by different remapping methods. Graphs are shown in logarithmic scale for single-material linear function (top) and two different non-linear functions separated by circle interface (bottom).

Table 1
Material L_1^{mat} error after 100 remapping steps between random meshes of different resolutions, performed by different remapping methods. Data for two different non-linear functions separated by circle interface shown.

Method	16×16	32×32	64×64	128×128	256×256	512×512
Intersections	1.03×10^{-2}	1.60×10^{-3}	2.68×10^{-4}	5.41×10^{-5}	1.29×10^{-5}	3.45×10^{-6}
Two-step hybrid	9.32×10^{-3}	1.48×10^{-3}	2.49×10^{-4}	5.32×10^{-5}	1.32×10^{-5}	3.56×10^{-6}
One-step hybrid	9.18×10^{-3}	1.46×10^{-3}	2.47×10^{-4}	5.29×10^{-5}	1.32×10^{-5}	3.57×10^{-6}

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 No. DE-AC52-06NA25396. The authors gratefully acknowledge the partial support of the US Department of Energy Office of Science Advanced Scientific Computing Research (ASCR) Program in Applied Mathematics Research and the partial support of the US Department of Energy National Nuclear Security Administration Advanced Simulation and Computing (ASC) Program. The first author was supported by the Czech Ministry of Education Grants MSM 6840770022, MSM 6840770010 and LC528 and the Czech Science Foundation project P201/10/P086. Partial support of the BARRANDE MEB021020 program and the European Science Foundation OPTPDE Network on Optimization with PDE Constraints is also gratefully acknowledged.

References

[1] Benson DJ. Computational methods in Lagrangian and Eulerian hydrocodes. Comput Method Appl Mech Eng 1992;99(2–3):235–394.
[2] Berndt M, Breil J, Galera S, Kucharik M, Maire P-H, Shashkov M. Two step hybrid remapping (conservative interpolation) for multimaterial arbitrary Lagrangian-Eulerian methods. J Comput Phys 2010, submitted for publication. LA-UR-2010-05438.
[3] Dyadechko V, Shashkov M. Reconstruction of multi-material interfaces from moment data. J Comput Phys 2008;227(11):5361–84.
[4] Hirt CW, Amsden AA, Cook JL. An arbitrary Lagrangian–Eulerian computing method for all flow speeds. J Comput Phys 1974;14(3):227–53.
[5] Kjellgren P, Hyvarinen J. An arbitrary Lagrangian–Eulerian finite element method. Comput Mech 1998;21(1):81–90.
[6] Kucharik M, Garimella RV, Schofield SP, Shashkov MJ. A comparative study of interface reconstruction methods for multi-material ALE simulations. J Comput Phys 2010;229(7):2432–52.
[7] Kucharik M, Shashkov M. One-step hybrid remapping algorithm for multi-material arbitrary Lagrangian–Eulerian methods, in preparation.
[8] Kucharik M, Shashkov M, Wendroff B. An efficient linearity-and-bound-preserving remapping method. J Comput Phys 2003;188(2):462–71.
[9] Loubère R, Shashkov M. A subcell remapping method on staggered polygonal grids for arbitrary Lagrangian–Eulerian methods. J Comput Phys 2005;209(1):105–38.
[10] Margolin LG. Introduction to An arbitrary Lagrangian–Eulerian computing method for all flow speeds. J Comput Phys 1997;135(2):198–202.
[11] Margolin LG, Shashkov M. Second-order sign-preserving conservative interpolation (remapping) on general grids. J Comput Phys 2003;184(1):266–98.
[12] Peery JS, Carroll DE. Multi-material ALE methods in unstructured grids. Comput Method Appl Mech Eng 2000;187(3–4):591–619.

Journal of Computational Physics 230 (2011) 6664–6687

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Two-step hybrid conservative remapping for multimaterial arbitrary Lagrangian–Eulerian methods

Markus Berndt [a], Jérôme Breil [b], Stéphane Galera [c], Milan Kucharik [d], Pierre-Henri Maire [e], Mikhail Shashkov [f,*]

[a] Los Alamos National Laboratory, CCS-2, Los Alamos, NM 87545, USA
[b] UMR CELIA, Université Bordeaux I 351, Cours de la Libération, 33 405 Talence, France
[c] INRIA, Team Bacchus, 351 Cours de la Libération, 33405 Talence Cedex, France
[d] Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Brehova 7, Praha 1, 115 19, Czech Republic
[e] CEA-CESTA BP 2, 33 114 Le Barp, France
[f] Los Alamos National Laboratory, XCP-4, Los Alamos, NM 87545, USA

ARTICLE INFO

Article history:
Received 18 August 2010
Received in revised form 24 February 2011
Accepted 2 May 2011
Available online 10 May 2011

Keywords:
Multimaterial arbitrary Lagrangian–Eulerian methods
Hybrid remapping
Conservative interpolation

ABSTRACT

We present a new hybrid conservative remapping algorithm for multimaterial Arbitrary Lagrangian–Eulerian (ALE) methods. The hybrid remapping is performed in two steps. In the first step, only nodes of the grid that lie inside subdomains occupied by single materials are moved. At this stage, computationally cheap swept-region remapping is used. In the second step, nodes that are vertices of mixed cells (cells containing several materials) and vertices of some cells in a buffer zone around mixed cells are moved. At this stage, intersection-based remapping is used. The hybrid algorithm results in computational expense that lies between swept-region and intersection-based remapping We demonstrate the performance of our new method for both structured and unstructured polygonal grids in two dimensions, as well as for cell-centered and staggered discretizations.

1. Introduction

In numerical simulations of fluid flow, the choice of the computational grid is crucial. Traditionally, there have been two viewpoints, either utilizing the Lagrangian or the Eulerian framework, each with its own advantages and disadvantages. In a pioneering paper [20], Hirt et al. developed the formalism for a grid whose motion could be determined as an independent degree of freedom, and showed that this general framework could be used to combine the best properties of the Lagrangian and Eulerian methods. This class of methods has been termed Arbitrary Lagrangian–Eulerian or ALE. Many authors have described ALE strategies to optimize accuracy, robustness, or computational efficiency, see for example [6,7,38,24,25,41,37].

For multimaterial flows, it is common to separate the ALE scheme into three distinct stages. These are: (1) a Lagrangian stage in which the solution and the grid are updated (this includes updating parameters of each material); (2) a rezoning stage in which the nodes of the computational grid are moved to a more optimal position; and (3) a remapping stage, in which one needs to conservatively transfer the Lagrangian solution for each material onto the rezoned grid, in modern methods it usually requires remapping of material interfaces (that is, reconstruction of interfaces on the rezoned grid from their representation on the Lagrangian mesh).

* Corresponding author.
E-mail addresses: berndt@lanl.gov (M. Berndt), breil@celia.u-bordeaux1.fr (J. Breil), Stephane.Galera@inria.fr (S. Galera), kucharik@newton.fjfi.cvut.cz (M. Kucharik), maire@celia.u-bordeaux1.fr (P.-H. Maire), shashkov@lanl.gov (M. Shashkov).

0021-9991/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2011.05.003

For multimaterial flows, the initial grid is usually aligned with material interfaces, that is, each cell of the grid contains only one material. For simple flows, it is possible to rezone the grid in each material and keep material interfaces aligned with the grid at later times by not moving nodes on interfaces at all or moving them along interfaces during the rezoning stage of ALE. Due to the nature of shock wave propagation in complex high-speed multimaterial flows with strong shear deformations, ALE methods are currently one of the proven technologies for solving such problems.

For complex flows, it is impossible to keep nodes of the grid on interfaces between materials and maintain a valid mesh during the entire calculation (for example, because of possible vorticity formation). Therefore, it is unavoidable that due to non-Lagrangian mesh movement mixed cells containing two or more materials will appear. Mixed cells in ALE methods represent material interfaces that undergo high deformation.

The main problem related to mixed cells in the Lagrangian phase is how to accurately determine the thermodynamic states of the individual material components and the nodal forces that such a zone generates, despite the lack of information about the velocity distribution within multimaterial cells. Usually, a separate set of material properties is maintained for all the materials in each multimaterial cell along with the volume fractions that define the fraction of the cell's volume occupied by each material. A sub-scale model is then required to define how the volume fractions and states of the individual materials evolve during the Lagrangian step. The construction of such a model is beyond the scope of this paper. We refer the interested reader to [48] for more information and appropriate references. For the purpose of this paper, it is important to note that the accuracy of the closure model for a mixed cell depends on the accuracy of the information about each material, its parameters, as well as the material location inside the mixed cell.

In the rezoning stage, the nodes of the computational grid are moved to more optimal positions. The rezoning stage results in the new grid. We assume that the rezoned grid is close to the Lagrangian grid. In particular, we assume that, after rezoning, a node of the grid stays in the union of Lagrangian cells that share this node. The interested reader can find a short review of rezoning methods in [26].

To start a new Lagrangian step, we need to conservatively remap all flow parameters from the Lagrangian grid at the completed time step to the new rezoned grid. The assumption of small movements of nodes during the rezoning stage implies that an exchange of information during the remapping stage occurs only between immediate neighbors.

As we have mentioned before, realistic multimaterial ALE calculations have to deal with mixed cells, which may appear and disappear after the rezoning stage. Therefore, in the remap stage one needs to determine which cells of the rezoned grid are pure and which are mixed, and find parameters for each material in mixed cells. There are several approaches to multimaterial remap – see for example seminal review paper by Benson [7].

There are several possible classifications of remapping methods. For example, one can distinguish methods as flux-based [35,33,34,5,14,47,46] or overlay-intersection-based, [19]. Flux-based methods can only be used if the Lagrangian and rezoned mesh have the same connectivity and are close to each other. Intersection-based methods can be used for any combination of two meshes.

Another possible classification can be based on how a method represents interfaces between materials: does it perform explicit interface reconstruction or not. For a review of methods that perform interface reconstruction the interested reader can refer to a paper by Rider and Kothe [44]. This paper has a special section on the historical perspective, including some classification of notable methods with respect to how interface reconstruction is performed. The authors of [44] distinguish between algebraic and geometric approaches for computing multimaterial fluxes. In the geometric approach, the contribution of materials to the flux between cells is based on the reconstructed interface and its interaction with the swept flux volume. In the algebraic approach, one introduces some ad hoc rules which emulate the effect of the interaction of the reconstructed interface with the flux volume in some simple situations. A review of some algebraic methods is presented in [45].

There are classes of flux-based methods, which do not use any interface reconstruction. Tipton's method that is described in [23] is a typical example. Further developments and improvements of these kind of methods can be found in [32,2].

Readers who interested in historical perspective also can refer to the following books [12,9]. Some recent development of remapping methods in production codes are described in [4,3].

We strongly believe that to perform multimaterial remap one needs to use interface reconstruction on the Lagrangian grid. However, it is beyond the scope of this paper to discuss interface reconstruction methods for the multimaterial case. We refer the interested reader to [1] for a review of such methods. For the purpose of this paper it is sufficient to know that if a mixed cell is a convex polygon in 2D (in this paper we are only concerned with 2D), then we assume each material in the mixed cell is represented by a convex sub-polygon, which contains only this material. If a mixed cell is non-convex, then a material in the mixed cell can be represented by disjoint pieces. In the most general case each material can be represented by set of non-intersecting triangles and convex polygons.

To accurately represent materials on the new grid one needs to intersect cells of the new grid with pure sub-polygons (which maybe triangles) representing materials on the old grid. This intersection can be computationally quite expensive, especially if non-convex cells need to be processed (such non-convex cells can result from the Lagrangian step). However, if a new cell is located inside a single pure material, then simple and cheaper methods that do not require intersections can be used for remapping this cell. One such relatively cheap method, which we use in this paper, is based on the notion of a swept-region, [39].

The goal of this paper is to describe a new efficient hybrid remapping method that can be used in multimaterial ALE simulations. This hybrid method uses a combination of computationally cheap swept-region remapping methods for cells inside

sub-domains containing a single pure material, and intersection-based remapping for sub-domains containing mixed cells plus cells in some buffer region around those mixed cells. Hybrid remapping is performed in two steps. In the first step, only nodes of the grid inside sub-domains occupied by single materials are moved. In this stage, computationally cheap swept-region remapping is used. In the second step, nodes are moved that are vertices of mixed cells (cells containing several materials) and vertices, which are shared by pure cells containing different materials. At this stage, intersection-based remapping is used. In general, our new approach improves the efficiency of remapping for multimaterial ALE codes while retaining the accuracy of intersection-based remapping. The general discussion related to factors affecting relative efficiency of remapping in multiphysics ALE codes is presented in Section 7.

The paper is organized as follows. In Section 2, we describe notations related to grids and representation of the materials. In Section 3, we describe the main ideas of remapping for the single material case. We describe intersection-based remapping as well as swept-region-based remapping and describe their properties. Conventional remapping for multimaterial ALE, which requires intersections, is described in Section 4. The new hybrid remapping method is described in Section 5. In that section, we give a motivation and describe the algorithm. Numerical experiments are presented in Section 6. We present results using two different multimaterial ALE codes. The first code, [27,28] uses structured quadrilateral grids and a staggered discretization. The second code, [18] uses general polygonal grids and a cell-centered discretization. We first present results for stand-alone cyclic remapping, where we know the exact solution. Then we present results for two realistic problems: a Rayleigh–Taylor instability problem and a problem of shock/bubble interaction. For all problems, we present timings using both ALE codes. We conclude with a discussion in Section 7.

2. Notations

2.1. Grids

We consider a two-dimensional computational domain Ω, assumed to be a general polygon. We assume that we are given a grid on Ω that consists of cells C_i, $i = 1,\ldots,i_{max}$ that cover Ω without gaps or overlaps. The cells C_i can be non-convex.

Each cell is defined by a set of vertices (which we will sometimes call points or nodes), denoted by $P(C_i)$, and a set of sides (which are segments of straight lines, we will sometimes refer to them also as faces or edges), denoted by $F(C_i)$. Each side F_k is shared by only two cells, denoted by $C(F_k)$. Each vertex P_m may be shared by an arbitrary number of cells. We denote the set of cells that have a common vertex by $C(P_m)$; similarly, we denote the set of all sides sharing a common vertex P_m by $F(P_m)$. The cells that share a side or vertex with a particular cell are called *neighbors*; the set of all the neighbors of a cell C_i is denoted by $C(C_i)$. To distinguish between neighbors which share a face or a point, we will denote them by $C_f(C_i)$ and $C_p(C_i)$, respectively. The reciprocal relation of the neighborhood defines the *connectivity* of the grid.

In the context of ALE methods, we consider two grids with the same connectivity – i.e., the same number of cells and vertices, and the same neighbor relations. The grid that contains the cells C_i is called the Lagrangian or *old* grid. The second grid, containing the cells $\widetilde{C}_i$, is called the rezoned or *new* grid.

In the ALE method, the rezoned grid results from an algorithm (i.e., a rezoner) that identifies and mitigates inadequacies of the Lagrangian grid. In Figs. 1 and 2, we show examples of pairs of a Lagrangian grid and a rezoned grid. The rezoned grids were generated using the optimization-based reference Jacobian strategy described in [26]. The rezoned grid produced by this algorithm remains "close" to the Lagrangian grid, but has better geometrical quality. Fig. 2 illustrates how complicated the relative locations of the two grids can be, even when displacements of the nodes are small.

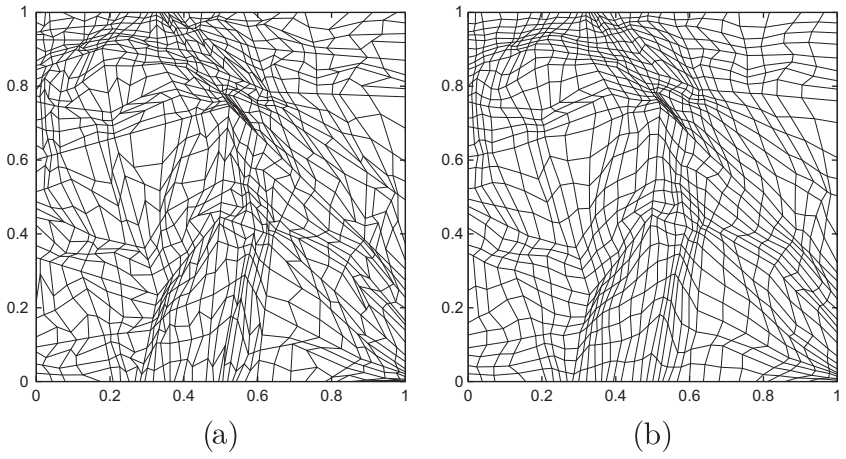


Fig. 1. (a) Old (Lagrangian) grid, (b) New (rezoned) grid.

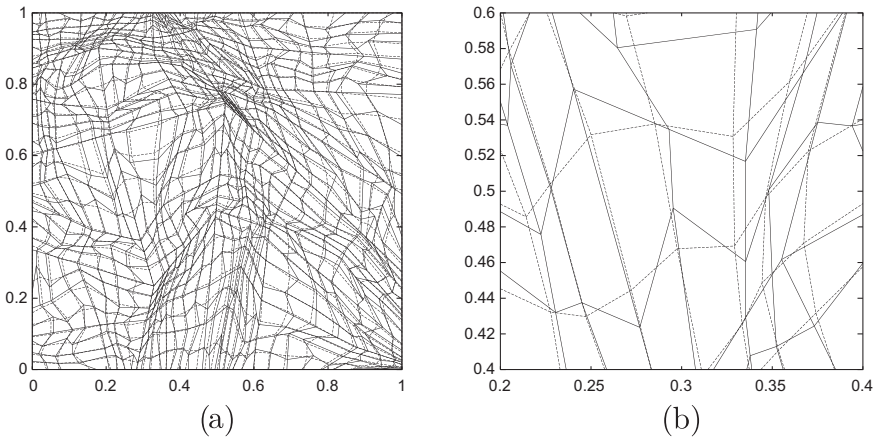


Fig. 2. Lagrangian (solid lines) and rezoned (dashed lines) grid. (a) Entire region, (b) Fragment.

After rezoning, the old grid $\{C_i\}$ is mapped into a new grid $\{\widetilde{C}_i\}$. We define a set $\mathcal{C}(C_i) = \bigcup_k C_k$, such that

$$\widetilde{C}_i \in \mathcal{C}(C_i). \tag{1}$$

For any two grids, such a set exists because $\widetilde{C}_i \in \bigcup_{k=1}^{i_{max}} C_k$. However, we will always consider the reasonably small set for which (1) holds.

In this paper, we will assume that $\mathcal{C}(C_i) = C_i \bigcup C(C_i)$; that is, the new cell $\widetilde{C}_i$ is contained in the union of the old cell C_i and its immediate neighbors, see, for example, Fig. 2(b). We denote set of indices corresponding to cells in $\mathcal{C}(C_i)$ by $\mathcal{I}(i)$.

2.2. Representation of materials

In multimaterial problems each material is assigned some unique number k_m from the global list of materials $\mathcal{K} = \{k_m = 1, 2, \ldots k_{max}\}$.

Each cell in the grid can be pure, containing just one material, or mixed, containing several materials.

In this paper, we assume that materials are represented by pure sub-polygons containing only one material. This assumes that some interface reconstruction has been performed.

Fig. 3(a) is an example of a grid for a two-material problem. Pure cells containing only material 1 are in dark grey, and pure cells containing only material 2 are in white. In mixed cells containing both materials the boundary of the sub-polygons containing material 1 is marked by a red dashed line. Each material, k, in mixed cell C_i occupies part of the cell, which we denote by C_i^k. Each mixed cell C_i has a list $\mathcal{K}_i = \{k_1, k_2, \ldots; k_s \in \mathcal{K}\}$ of materials, where k_i is one of the materials from the global list. In Fig. 3(b), we show one mixed cell which consists of four different materials. For a non-convex cell C_i, C_i^k can consist of several disjoint pieces, as material 2 in (Fig. 3(c)); and the material can be represented by a non-convex polygon, as material 1 in (Fig. 3(c)) (see [1] for details). It is important that the pieces representing all materials do indeed cover the mixed cell without gaps and overlaps. The presence of several disjoint pieces and non-convex polygons can affect the efficiency of the multimaterial remap (see Section 6 for details).

If C_i^k consists of several disjoint pieces, then we will assume that we know the geometry of each such piece. For each material k we also know the total mass of the material, m_i^k. We will denote the volume of region C_i^k by $|C_i^k|$ or V_i^k, and the density of material k by $\bar{\rho}_i^k$.

3. Remapping for the single-material case

3.1. Statement of the remapping problem

We start with a definition of the remapping problem for a single material. Here we assume that there is a positive function $\rho(\vec{\mathbf{r}}) > 0, \mathbf{r} = (x, y)$, which we call density, that is defined throughout the problem domain. The only information that we are given about this function is its mean value in each of the cells of the old grid:

$$\bar{\rho}_i = \frac{\int_{C_i} \rho(\mathbf{r})\, dV}{V(C_i)}, \tag{2}$$

where $V(C_i)$ is the volume of the cell C_i. The numerator of (2) is the cell mass

$$m_i \equiv \int_{C_i} \rho(\mathbf{r})\, dV \tag{3}$$

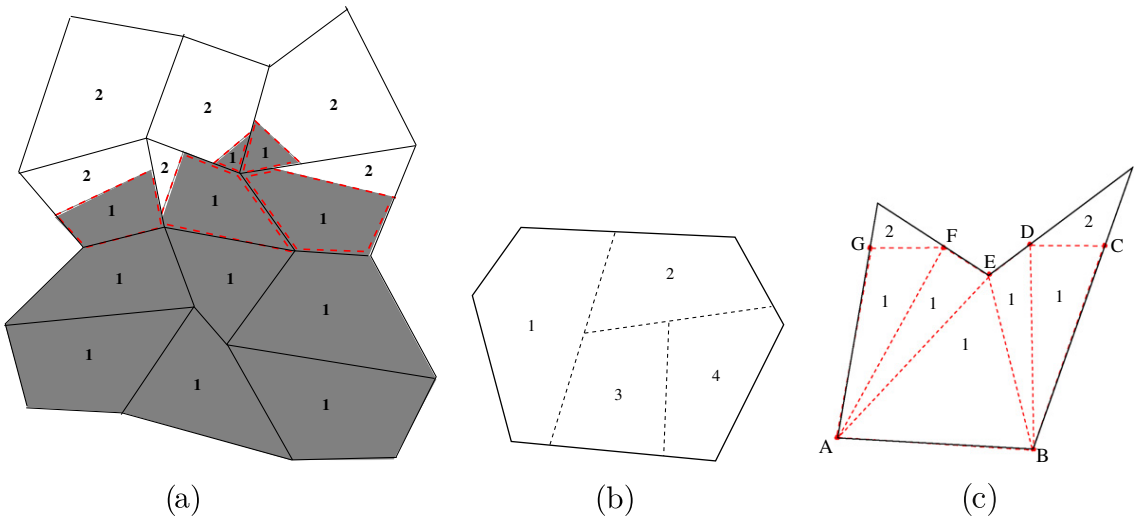


Fig. 3. Representation of materials: (a) Two-material case, in mixed cells boundaries of polygons representing material #1 are marked by a thick dashed red line. Material numbers are inside pure cells and inside polygons representing materials in mixed cells; (b) one mixed cell containing four different materials – material numbers are inside polygons representing materials; (c) one non-convex mixed cell containing two materials – material #1 is represented by one non-convex polygon *ABCDEFG*. At the interface remapping stage this non-convex polygon is subdivided into triangles; material #2 is represented by two disjoint triangles. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

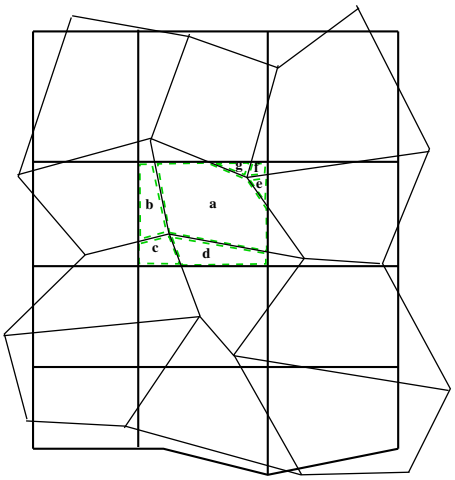


Fig. 4. Aggregated-intersection-based remapping. The old grid is shown by thin solid lines, new grid is thick solid lines. Boundaries of intersection polygons, (a)–(g), corresponding to $\widetilde{C}_i \bigcap C_{i'}$ are marked by dashed lines.

and so the mean density is

$$\bar{\rho}_i = \frac{m_i}{V(C_i)}. \tag{4}$$

The total problem mass is

$$M \equiv \int_\Omega \rho(\mathbf{r})\, dV = \sum_{i=1}^{i_{max}} \int_{C_i} \rho(\mathbf{r})\, dV = \sum_{i=1}^{i_{max}} m_i = \sum_{i=1}^{i_{max}} \bar{\rho}_i V(C_i). \tag{5}$$

The problem statement is to find accurate approximations $\tilde{m}_i$ for the exact masses, m_i^{ex}, of the new cells

$$\tilde{m}_i \approx m_i^{ex} = \int_{\widetilde{C}_i} \rho(\mathbf{r})\, dV. \tag{6}$$

The issue is to define what is meant by "accurate", since the underlying density field is not known in detail.

The approximate mean values of density in the new cells are defined by

$$\tilde{\bar{\rho}}_i = \frac{\tilde{m}_i}{V(\widetilde{C}_i)}. \tag{7}$$

The common accuracy requirement for remapping is formulated as *linearity preservation*. That is, if the underlying function $\rho(\mathbf{r})$ is a global linear function, the remap must be exact;

$$\tilde{m}_i = m_i^{ex}. \tag{8}$$

Another important property of remapping is *bound preservation*. If we assume that a new cell is contained in the immediate neighborhood of the old cell, then it is natural to require

$$\min_{i' \in \mathcal{I}(i)} \bar{\rho}_{i'} \leqslant \tilde{\bar{\rho}}_i \leqslant \max_{i' \in \mathcal{I}(i)} \bar{\rho}_{i'}. \tag{9}$$

Finally, a statement of *global conservation* is formulated as

$$\sum_{i=1}^{i_{max}} \tilde{m}_i = M. \tag{10}$$

We will refer to the problem of finding accurate, bounded approximations for the masses and the corresponding mean densities on the new grid, such that total mass is conserved, as *bound-preserving conservative remapping (interpolation)*.

3.2. Intersection-based remapping

3.2.1. Aggregated intersection-based remapping

Each cell of the new grid $\widetilde{C}_i$ is formed from pieces of the cells of the old grid C_i

$$\widetilde{C}_i = \bigcup_{i'=1}^{i_{max}} \left(\widetilde{C}_i \bigcap C_{i'}\right) = \bigcup_{i' \in \mathcal{I}(i)} \left(\widetilde{C}_i \bigcap C_{i'}\right). \tag{11}$$

The most natural approach to remapping can be based on this representation of the new cell:

$$m_i^{ex} = \int_{\widetilde{C}_i} \rho(\mathbf{r})\, dV = \sum_{i' \in \mathcal{I}(i)} \int_{\widetilde{C}_i \bigcap C_{i'}} \rho(\mathbf{r})\, dV. \tag{12}$$

The remapping based on this formula would be exact if we knew the density function everywhere on the old grid. However, as pointed out earlier, we only know the average value of $\rho(\mathbf{r})$ within a cell. Thus, it is necessary *to reconstruct* the density function in each cell of the old grid. Usually, this is a piece-wise linear reconstruction over cells of the old grid. This reconstruction involves an estimate of the gradient of the function on the cell and some limiting procedure for the gradient to guarantee bound preservation, (9) . Details of the reconstruction are not important for this paper and can be found elsewhere (see, for example, [31]). For the purpose of this paper, it is important to know that there is some cost associated with the reconstruction.

We denote the reconstruction over cell C_i as $\rho_i(\mathbf{r})$. For aggregated intersection-based remapping, this must be a *conservative* reconstruction

$$\int_{C_i} \rho_i(\mathbf{r})\, dV = \bar{\rho}_i. \tag{13}$$

Then, we can use the following approximation

$$\int_{\widetilde{C}_i \bigcap C_{i'}} \rho(\mathbf{r})\, dV \approx \int_{\widetilde{C}_i \bigcap C_{i'}} \rho_{i'}(\mathbf{r})\, dV. \tag{14}$$

This leads to the following formula for the remapped masses on the new grid

$$\tilde{m}_i = \sum_{i' \in \mathcal{I}(i)} \int_{\widetilde{C}_i \bigcap C_{i'}} \rho_{i'}(\mathbf{r})\, dV. \tag{15}$$

It is important to note that any polynomial function can be integrated *exactly* over a polygonal cell.

We refer to the remapping method of (15) as aggregated intersection-based (AIB) method (Fig. 4). Here, *aggregated* refers to the fact that the mass of a new cell is obtained by collecting pieces of masses from old cells.

The AIB method is conservative because

$$\sum_{i=1}^{i_{max}} \tilde{m}_i = \sum_{i=1}^{i_{max}} \left(\sum_{i' \in \mathcal{I}(i)} \int_{\widetilde{C}_i \bigcap C_{i'}} \rho_k(\mathbf{r})\, dV \right) = \sum_{i=1}^{i_{max}} \left(\sum_{i'=1}^{i_{max}} \int_{\widetilde{C}_i \bigcap C_{i'}} \rho_{i'}(\mathbf{r})\, dV \right) = \sum_{i'=1}^{i_{max}} \left(\sum_{i=1}^{i_{max}} \int_{\widetilde{C}_i \bigcap C_{i'}} \rho_{i'}(\mathbf{r})\, dV \right) = \sum_{i'=1}^{i_{max}} \left(\int_{C_{i'}} \rho_{i'}(\mathbf{r})\, dV \right)$$
$$= \sum_{i'=1}^{i_{max}} m_{i'} = M.$$

AIB remapping is also bound and linearity preserving if the reconstructed function $\rho_k(\mathbf{r})$ satisfies these properties, [39].

We note that AIB methods are very general and in principle can be applied when the old and new grids are not related to each other – they may even have a different number of cells of arbitrary shapes and a different connectivity.

Even in 2D, if cells of the old or new grids, or both, are non-convex, then it may be more efficient to subdivide cells into triangles and to intersect triangles representing old and new cells. We believe that, in 3D, subdividing cells into tetrahedra is the only way of doing intersections, because cells in 3D almost always have non-flat faces.

3.2.2. Flux-intersection-based remapping

When the old and new grids have the same connectivity (as we assume in this paper), a new cell can be represented as follows

$$\widetilde{C}_i = C_i \bigcup \left(\bigcup_{i' \in \mathcal{I}'(i)} \widetilde{C}_i \bigcap C_{i'} \right) \setminus \left(\bigcup_{i' \in \mathcal{I}'(i)} C_i \bigcap \widetilde{C}_{i'} \right), \tag{16}$$

where

$$\mathcal{C}'(C_i) = \mathcal{C}(C_i) \setminus C_i; \quad \mathcal{I}'(i) \text{ is corresponding set of indices}, \tag{17}$$

and where $\setminus$ is the difference operation on sets. In words, the new cell is the old cell plus pieces of neighboring cells that are added minus pieces of the old cell lost to other new cells. The corresponding representation for the exact mass of the new cell is

$$m_i^{ex} = \int_{\widetilde{C}_i} \rho(\mathbf{r})\, dV = \int_{C_i} \rho(\mathbf{r})\, dV + \sum_{i' \in \mathcal{I}'(i)} \int_{\widetilde{C}_i \bigcap C_{i'}} \rho(\mathbf{r})\, dV - \sum_{i' \in \mathcal{I}'(i)} \int_{C_i \bigcap \widetilde{C}_{i'}} \rho(\mathbf{r})\, dV = m_i + \sum_{i' \in \mathcal{I}'(i)} \mathcal{F}_{i,i'}^{ex}, \tag{18}$$

where

$$\mathcal{F}_{i,i'}^{ex} = \int_{\widetilde{C}_i \bigcap C_{i'}} \rho(\mathbf{r})\, dV - \int_{C_i \bigcap \widetilde{C}_{i'}} \rho(\mathbf{r})\, dV; \quad i' \neq i \tag{19}$$

are *generalized mass fluxes*. We note that the second cell indicated by the index i' may be any cell in the neighborhood and is not restricted to those with whom the cell C_i has a common side. Eq. (18) is exact, and illustrates that the mass of the new cell can be written as the mass of the corresponding old cell plus the exchange of masses with neighboring cells.

The remapping based on representation (19) has the important theoretical advantage that it is conservative without requiring any properties in the reconstruction nor an exact integration of the reconstructed function. In fact, any formula of the form

$$\tilde{m}_i = m_i + \sum_{i' \in \mathcal{I}'(i)} \mathcal{F}_{i,i'}, \tag{20}$$

where $\mathcal{F}_{i,i'} = -\mathcal{F}_{i',i}$ is some approximation of the flux, will be conservative because of detailed balance.

We will refer to methods based on Eq. (20) and some approximation of (19) as flux-intersection-based (FIB) methods.

In Fig. 5, we schematically present positive and negative pieces of the generalized fluxes. We note that, if AIB and FIB methods are using the same reconstruction of ρ on the old grid and generalized fluxes in the FIB method are computed as

$$\mathcal{F}_{i,i'} = \int_{\widetilde{C}_i \bigcap C_{i'}} \rho_{i'}(\mathbf{r})\, dV - \int_{C_i \bigcap \widetilde{C}_{i'}} \rho_i(\mathbf{r})\, dV; \quad i' \neq i, \tag{21}$$

then the two methods are algebraically equivalent.[1] However, their implementation may be quite different.

3.3. Swept-region-based methods

Flux form (20) allows us to construct approximate fluxes without calculation of the intersections between the cells of the new and old grids.

The approximations that we use are based on two ideas, [39]. First, up to fourth-order accuracy, the exact masses of new cells can be represented as line integrals of polynomial functions over the boundary of a new cell. Second, the line integral

[1] In the rest of this paper we always will be using definition (21) for generalized fluxes in FIB methods.

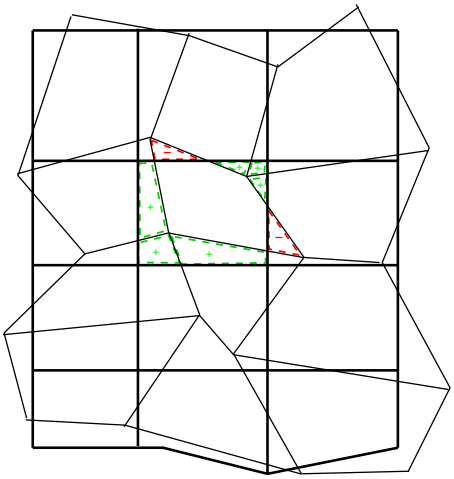


Fig. 5. Flux-intersection-based remapping. The old grid is shown by thin solid lines and the new grid is shown by thick solid lines. Boundaries of intersection polygons, are marked by dashed lines. Signs + and − correspond to positive and negative pieces of corresponding generalized fluxes. For example, the generalized flux corresponding to the edge shared with right cell has two pieces; and the flux corresponding to the left edge has only one piece.

over the boundary of the new cell is the line integral over the boundary of the old cell (which is the old mass) plus the line integrals over the regions swept by the movement of the faces (i.e., sides) of the cell C_i. The face which shares cells C_i and $C_{i'}$ can be denoted by two indices, i, i'. The corresponding swept region is denoted by $\delta F_{i,i'}$.

We define swept-region-based remapping (SRB) as follows

$$\tilde{m}_i = m_i + \sum_{i' \in I_f(i)} \mathcal{F}_{i,i'}, \tag{22}$$

where

$$\mathcal{F}_{i,i'} = -\mathcal{F}_{i',i},$$

and $\mathcal{I}_f(i)$ is set of indices of cells, which have common face with cell i. In (22), fluxes correspond only to neighbors which share faces. The approximate "fluxes" $\mathcal{F}_{i,i'}$ are

$$\mathcal{F}_{i,i'} = \int_{\delta F_{i,i'}} \rho_{i,i'}(\mathbf{r}), \tag{23}$$

where the density function associated with the faces is defined depending on the sign of the volume of the swept region[2]

$$\rho_{i,i'} = \begin{cases} \rho_{i'}(\mathbf{r}), & V(\delta F_{i,i'}) \geqslant 0, \\ \rho_i(\mathbf{r}), & V(\delta F_{i,i'}) < 0, \end{cases} \tag{24}$$

where $V(\delta F_{i,i'})$ is the signed volume of the swept region.

The main advantage of the SRB method (23), (24) is that it does not require finding intersections of the old and new grids, which makes it computationally much cheaper when compared to intersection-based methods.

The detailed derivation and analysis of swept-region-based methods is presented in [39,31].

3.4. Properties of remapping methods for a single material

Both intersection-based and swept-region based methods are linearity-preserving, if the reconstruction procedure for $\rho_i(\mathbf{r})$ is linearity-preserving. The formal order of accuracy (the accuracy of remapping for smooth functions) is the same for both methods. In particular, both methods exactly compute volumes of cells of the new grid, which corresponds to setting $\rho_i(\mathbf{r}) = 1$. Hence, they satisfy the geometric conservation law (GCL), [17].

Clearly, intersection-based methods are the most accurate methods for a given reconstruction of the function on the old grid. The remapped value in the new cell depends on all values in the neighboring cells. In particular, it involves values in neighboring cells which are sharing only a vertex with the cell under consideration. This can be significant, if the underlying

[2] The signed volume is defined using a line integral representation of the volume where orientation of the boundary of the swept region is taken in such way that the signed volume is positive if the swept region is added to the cell and negative other-wise, [39].

function is discontinuous and values of the function in neighbor cells that are only vertex connected are orders of magnitude different from the value in the cell under consideration. Swept-region-based methods described in Section 3.3 are using information only from face connected neighbor cells and can be less accurate in such situations. We note that in principle one can construct more complicated swept-region methods, where the face flux includes information about vertex neighbors (see for example, CTU – Corner Transport Upwind method by Colella, [14], and references in the introduction therein).

However, it is interesting to note that results presented in [31], as well as results presented in Section 6.2 in this paper show that for single material cyclic remapping examples (that is, repeated remapping of a given function on a sequence of grids), the accuracy of swept-region-based and intersection-based methods described in this paper is comparable even for discontinuous functions.

It is important to consider that, if a hydro code requires remapping of many physical quantities and not just density, then reconstruction must be done for all quantities but intersections can be done once and used for all quantities. Similarly, the most costly part of integration, which is the integration of $1,x,y,xy,\ldots$ can be done once for all quantities. This observation suggests that the relative efficiency of remapping methods must be analyzed for each specific situation.

An alternative remapping approach is presented in [21,22], It can be considered an intermediate between intersection-based and swept-region-based methods. In this approach, if the swept region is self-overlapping, then it is represented as two triangles. To find these two triangles one needs to intersect the old and new edge (which is much cheaper than the intersection of polygons). Each of these two triangles now is treated as a swept region by itself and the reconstructed function in the triangle is taken according to the sign of its signed volume.

4. Remapping for the multimaterial case

In this section, we describe conventional remapping methods for multimaterial ALE methods.

4.1. Statement of remapping

In multimaterial remap, pure and mixed cells in the new grid must be identified. For each pure cell, the mass of the corresponding material must be computed. This is similar to the case of single material remap. For mixed cells, the mass of each material must be determined, as well as its volume, because after remap we may need to reconstruct interfaces on the new grid. We also may need some additional information. For example, the advanced interface reconstruction moment-of-fluid (MOF) method requires remapping of material centroids, [1,16].

In the case of several materials the total volume $\mathcal{V}_k$ and total mass M_k of each material k must be conserved

$$\mathcal{V}_k = \sum_i V_i^k = \sum_i \widetilde{V}_i^k = \widetilde{\mathcal{V}}_k, \quad M_k = \sum_i m_i^k = \sum_i \tilde{m}_i^k = \widetilde{M}_k, \tag{25}$$

where the sum is formally over all cells, but in practice can be taken only over pure cells containing material k and mixed cells containing this material.

Accuracy and monotonicity issues of multimaterial remap are beyond of the scope of this paper and will be addressed in the future.

In general, for multimaterial ALE one needs to use intersection-based remapping, as is done, for example, in [53,27,28,18].

4.2. Multimaterial aggregated intersection-based remapping

In the multimaterial case, pieces of material k are collected in the new cell $\widetilde{C}_i$ according to the following formula

$$\widetilde{C}_i^k = \bigcup_{i' \in \mathcal{I}(i)} \left(\widetilde{C}_i \bigcap C_{i'}^k \right). \tag{26}$$

Eq. (26) is similar to Eq. (11), except that new cell $\widetilde{C}_i$ intersects with pure sub-polygons C_i^k. This process is illustrated in Fig. 6. We denote the method based on formula (26) by MAIB - multimaterial aggregated-intersection based.

The total volume of material k in new cell C_i^k is computed as the sum of volumes of corresponding intersections

$$|\widetilde{C}_i^k| = \widetilde{V}_i^k = \sum_{i' \in \mathcal{I}(i)} \int_{\widetilde{C}_i \bigcap C_{i'}^k} dV. \tag{27}$$

The total mass of material k in new cell C_i^k is computed similarly

$$\tilde{m}_i^k = \sum_{i' \in \mathcal{I}(i)} \int_{\widetilde{C}_i \bigcap C_{i'}^k} \rho_{i'}^k(\mathbf{r})\, dV, \tag{28}$$

where $\rho_{i'}^k(\mathbf{r})$ is some reconstruction of the density of material k in $C_{i'}^k \in C_{i'}$.

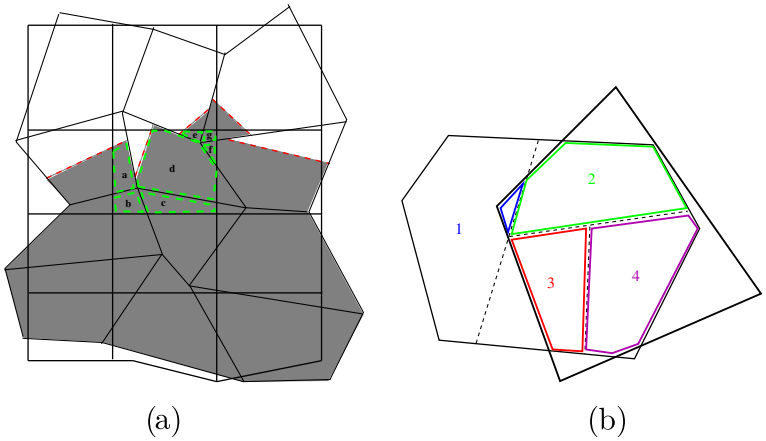


Fig. 6. MAIB – Multimaterial remap AIB method. Old grid – thin solid lines, and new grid – thick solid lines: (a) Two-material case – old and new grid. Pieces of material #1 from numerous old cells which contribute to central new cell are marked by *a*, *b*, *c*, *d*, *e*, *f*, *g* in mixed cells in mixed cells. Boundaries of polygons representing material #1 are marked by thick dashed red line. Material ids are inside pure cells and polygons representing material in mixed cells; (b) one mixed cell containing four different materials – (material ids are inside polygons representing material) intersecting with new cell. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

For advanced interface reconstruction, we may also need to know the centroids of each material in a mixed cell. The centroid is defined as the ratio of first and zeroth moment of the domain occupied by the material. The zeroth moment is simply the volume of the domain occupied by the material and it is defined by Eq. (27). The first moment, $\tilde{\mu}_i^k$, is defined as

$$\tilde{\mu}_i^k = \int_{\widetilde{C}_i^k} \mathbf{r}\, dV = \sum_{i' \in \mathcal{I}(i)} \int_{\widetilde{C}_i \bigcap C_{i'}^k} \mathbf{r}\, dV. \tag{29}$$

In the process of remapping, we must determine if a new cell is pure or mixed. This requires additional logical operations, and building a list of materials for each new mixed cell, as well as identifying the situation when a mixed cell becomes pure. As a result of intersections, for each new mixed cell, we know volume, mass, and centroid of each material. In fact, we know exactly all pieces of pure polygons representing materials on the old grid, which are now in the new cell. In Fig. 6(a) these pieces are marked by *a*, *b*, *c*, *d*, *e*, *f*, *g*. The goal of interface reconstruction is to simplify this representation by replacing a collection of pieces by one or possibly several (in case of non-convex cell $\widetilde{C}_i$) polygons, such that the volume of this polygon exactly equals the sum of volumes of the pieces.

We note that, again, as in the case of a single material all integrals in the equations in this section can be computed exactly, because they are integrals of polynomial functions over polygonal domains.

It is clear that, in comparison with the single material case, MAIB remapping requires more intersections – a new cell has to be intersected with all material polygons in its neighboring old cells.

However, the main stages of MAIB are similar to AIB. In conventional multimaterial ALE codes, MAIB remapping is used for the entire grid. Clearly, the cost of intersection based MAIB remapping is dominated by the cost of intersections, and it is much higher than the cost of SRB remapping for the single material case for the same grid.

4.3. Multimaterial flux-intersection-based remapping

Formulas for multimaterial flux-intersection-based (MFIB) remapping are

$$\widetilde{C}_i^k = C_i^k \bigcup \left(\bigcup_{i' \in \mathcal{I}'(i)} \widetilde{C}_i \bigcap C_{i'}^k \right) \setminus \left(\bigcup_{i' \in \mathcal{I}'(i)} C_i^k \bigcap \widetilde{C}_{i'} \right), \tag{30}$$

$$\tilde{m}_i^k = m_i^k + \sum_{i' \in \mathcal{I}'(i)} \left(\int_{\widetilde{C}_i \bigcap C_{i'}^k} \rho_{i'}^k(\mathbf{r})\, dV - \int_{C_i^k \bigcap \widetilde{C}_{i'}} \rho_i^k(\mathbf{r})\, dV \right), \tag{31}$$

$$\widetilde{V}_i^k = V_i^k + \sum_{i' \in \mathcal{I}'(i)} \left(\int_{\widetilde{C}_i \bigcap C_{i'}^k} dV - \int_{C_i^k \bigcap \widetilde{C}_{i'}} dV \right), \tag{32}$$

$$\tilde{\mu}_i^k = \mu_i^k + \sum_{i' \in \mathcal{I}'(i)} \left(\int_{\widetilde{C}_i \bigcap C_{i'}^k} \mathbf{r}\, dV - \int_{C_i^k \bigcap \widetilde{C}_{i'}} \mathbf{r}\, dV \right). \tag{33}$$

In comparing the MAIB and MFIB methods the same comments as at the end of Section 4.2 apply.

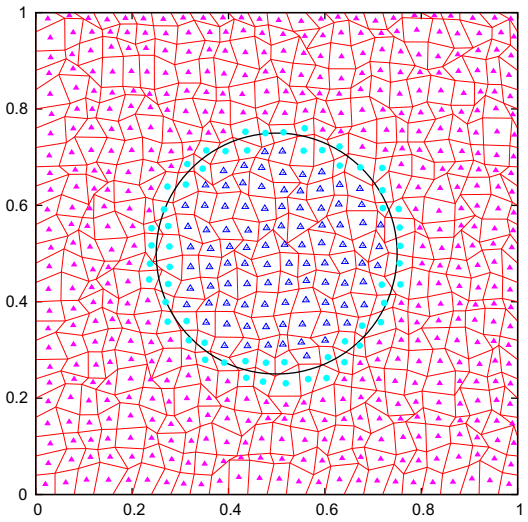


Fig. 7. Two-material case. Lagrangian (old grid) – red solid lines; interface (circle) – solid black line; pure cells containing only material 1 are marked by blue triangles placed inside the cells; pure cells containing only material 2 are marked by red solid triangles placed inside the cells; mixed cells containing both material are marked by cyan solid circles. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

5. Hybrid multimaterial remapping

5.1. Motivation

As motivation we consider the example with two materials depicted in Fig. 7. The computational domain is the unit square. Materials are separated by the circular interface with radius 0.25 and center (0.5, 0.5). Material 1 is inside the circle and material 2 is outside the circle. The computational grid is a distorted logically rectangular grid of 625 = 25 × 25 cells. The number of pure cells containing material 1 is 95 or 15.2% of the total number of cells. The number of pure cells containing material 2 is 474 or 75.84% of the total number of cells. The total number of pure cells is 569 or 91.04%, and the total number of mixed cells is 56 or 8.96%. Even for this coarse grid, the number of mixed cells is significantly smaller than the number of pure cells. In general, for the non-degenerate case, if the total number of cells is $\mathcal{N}$ then the number of mixed cells is proportional to $\sqrt{\mathcal{N}}$ and therefore the percentage of mixed cells approaches zero as the grid is refined. For example, if we refine the grid and make the total number of cells equal to $\mathcal{N} = 160000 = 400 \times 400$ then the percentage of mixed cells will drop to only 0.5%. For this particular example, the number of mixed cells is approximately equal to $2\sqrt{\mathcal{N}}$.

A straightforward approach for remapping in this multimaterial case is to use the MAIB or MFIB intersection-based methods described in the previous sections.

We now recall that intersection-based methods are quite expensive in comparison with swept-region-based methods because they require the intersection of the old and new grid. This fact and the observation that the percentage of mixed cells is small motivate us to construct a *hybrid* method, where "inside" pure materials we use swept-region-based remapping and for mixed cells we use an intersection-based method. The cost of such a hybrid method can be expected to be between the cost of a swept-region-based method (which it approaches, when the grid is refined and the percentage of mixed cells goes to zero) and the cost of intersection-based remapping.

Recall that we assume vertices of the new grid to be located inside the union of the old cells sharing a corresponding vertex. It is due to this assumption that we can be sure that a pure cell is surrounded by pure cells of the same material. As a consequence, this cell will remain pure in the new grid and therefore we can use swept-region remapping to update the density in this cell.

For cells which are mixed or may become mixed we need to use intersection-based methods.

The main problem in constructing hybrid method is to decide what to do with fluxes corresponding to the edges, which are shared by pure and mixed cells.

Intersection-based and swept-region based remapping methods cannot easily be combined, even in the single material case.[3] To explain the problem, we consider a hypothetical hybrid method which combines SRB, the flux-swept-region-based method, and FIB, the flux-intersection-based method. The first idea that comes to mind is to use the flux from the SRB or FIB method depending on the type of edge: Use the flux from the SRB method for edges that are shared by pure cells and use the

[3] For some special cases like structured quadrilateral grids this can be done, [27].

the flux from the FIB method for edges that are shared by pure and mixed cells. Unfortunately, such an approach does not work for several reasons. First, the number of fluxes for the SRB and FIB methods are different; SRB has only edge related fluxes and FIB additionally has vertex based fluxes. Second, even edge based fluxes are of a different nature in either method. In the SRB method, an edge flux controls the exchange not only between cells sharing this edge, but also between cells sharing end points of this edge and the cell under consideration. In contrast, for the FIB method, the edge flux controls the exchange only with the cell sharing this edge. Thus, SRB and FIB edge fluxes refer to different geometric objects. One can be easily convinced that such a simple approach will not work by observing that replacing the SRB flux with the FIB flux for one of the edges will not be exact even for remapping of the constant density, $\rho = 1$, hence, it does not satisfy the geometric conservation law.

Now we will describe one possible approach to constructing a hybrid remapping method.

5.2. Pure and mixed points

We start with a classification of points (vertices) of an old grid. We will call a point *mixed* in two cases: if the point is the vertex of a mixed cell or if the point is the vertex that is shared by pure cells and not all of these cells contain the same material; the rest of points are pure. Fig. 8 illustrates this definition.

The main reason for such a classification of points is that the movement of *pure* points *cannot* create new mixed cells, Fig. 9. Therefore, if only pure points have been moved, remapping between the old grid and an *intermediate* grid obtained from the old grid by movement of only pure points can be performed by the SRB method.

In contrast, it is obvious that movement of *mixed* points *can* create new mixed cells. This observation suggests the following two-step rezoning/remapping algorithm.

5.3. Algorithm

5.3.1. Swept-region rezoning/remapping step

In the first step, only pure points are moved. This creates an intermediate mesh, where pure points have been moved and mixed points remain at their original positions. The question is how to move pure points.

Moving pure points can be done by using any rezoning algorithm, (cf., [50,26]), while keeping mixed points fixed. We assume that the rezoning algorithm creates a valid grid. This is the approach we use in the real hydro calculations presented in this paper. Examples of an old and an intermediate grid obtained by this approach are presented in Fig. 9(a) and (b), respectively. In this case, the intermediate grid is obtained by one Laplace smoothing step, which is just the simple average of coordinates of vertices of surrounding cells.

There is another approach, that we use in stand-alone cyclic remapping described in Section 6.2. In this case final positions of all pure and mixed points are predetermined. Therefore, in the first step of the hybrid algorithm, pure points are moved to their final positions and mixed points remain at their original positions. Even if the final mesh is valid it is not guaranteed that the intermediate mesh will also be valid. In Section 6.2 validity of the mesh is checked. In general we do not recommend this approach, because it is not easy to check validity of the intermediate mesh.

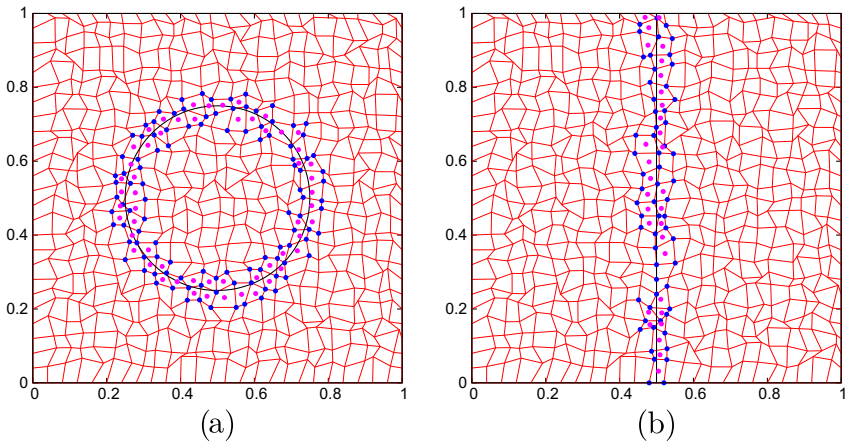


Fig. 8. Two-material case. Lagrangian (old grid) – red solid lines. Left panel – (a): interface (circle) – solid black line; mixed cells containing both material are marked by magenta solid circles. Right panel – (b) interface (straight line) – solid black line; mixed cells containing both material are marked by magenta solid circles; mixed points are marked by dark blue circles. In right panel there are several mixed points (in region close to (0.5, 0.3)), that lie on the interface and are shared by pure cells of different materials. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

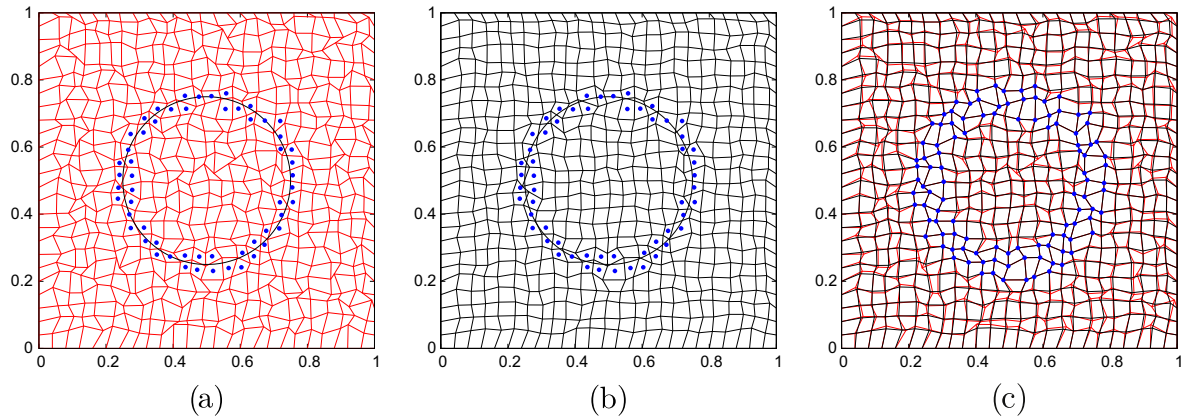


Fig. 9. Left panel: old grid – solid red lines, interface and mixed cells. Central panel: only pure points have been moved, grid after movement of pure cells – solid black lines; no new mixed cells have been created. Right panel: overlap of old and new grid obtained by movement of only pure points; mixed points are marked by solid blue circles; mixed cells are the same for old and new grid. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

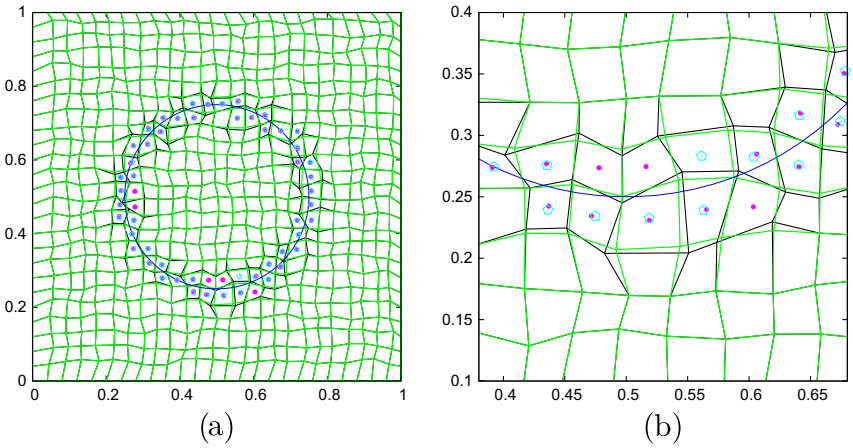


Fig. 10. Overlap of intermediate grid (solid black line) and final new grid (solid green line). Left panel entire grid: mixed cells on old grid are marked by magenta solid circles; mixed cells on new grid are marked by cyan pentagons. Right panel – fragment, interface is also shown. Some cells which were mixed on old grid now are pure and vice versa. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Remapping from the old to the intermediate grid is performed using the SRB method. In fact, this method can be applied to the entire grid, because edges of mixed cells and edges between pure cells containing different materials do not move (hence, corresponding fluxes are zero). Alternatively, we can mark such edges and not compute the corresponding fluxes which we know to be zero. As a result of the first step we have updated values of mass and density in all pure cells. Densities in mixed cells did not change. At this stage, interface reconstruction in mixed cells is not needed because mixed cells are not directly affected. Therefore, interfaces in mixed cells are the same as they were on the old grid.[4]

5.3.2. Intersection-based rezoning/remapping step

In the second step, we start with the intermediate grid where we know all data and move only mixed points.

Again, we can use any rezoning algorithm keeping pure points fixed in the position that they had been moved to in the first step.

Alternatively, we can move mixed points to some predetermined positions as we will do in stand-alone cyclic remapping tests in Section 6.2.

[4] Strictly speaking, this is not exactly true, because densities and geometry of surrounding pure cells has been changed. If one decides to perform interface reconstruction using the updated grid and geometry using the VOF method, the interface in mixed cells can change slightly. If one uses MOF interface reconstruction instead, then the interface will not change because MOF does not use information from neighboring cells. In this work we use MOF for interface reconstruction and therefore we do not need this additional step.

The second step creates the final grid, Fig. 10 (this is the mesh which in real ALE calculations will be used at the beginning of next Lagrangian step). Remapping from the intermediate grid to the new grid is performed using an intersection based method. We now explain the details of this step.

Clearly, in the second step, a mixed cell can exchange data only with other mixed cells or with pure cells which share an edge or a vertex with this mixed cell. We note that there are degenerate situations when a mixed point is shared by pure cells that do not all contain the same material. In this situation, pure cells of different materials exchange mass and, thus, all of them become mixed. This observation leads to the notion of *buffer* cells. Such buffer cells are cells that may be involved in data exchange in the second step. Buffer cells are cells which have at least one mixed point as a vertex. In Fig. 11 we depict mixed and buffer cells in the old grid.

If the MFIB method is used in the second step of hybrid remapping, only fluxes need to be computed that are related to edges for which at least one end point is a mixed point, and for fluxes related to mixed points. We denote such edges as *mixed* edges in contrast to *pure* edges, for which both end points are pure points. If the MAIB method is used in the second step of hybrid remapping, it only needs to be performed for mixed and buffer cells.

Before performing the second step, we need to reconstruct the density function in buffer cells using data obtained from buffer cells and its pure neighbors that were the result of the first step SRB remapping. We denote this new two-step hybrid multimaterial algorithm by MHYB.

Note that it is critical that pure cells are processed first. If mixed cells were processed first, a new mixed cell might be created in the intermediate mesh.

6. Numerical experiments

In this section we present results obtained using the algorithms implemented in two multimaterial ALE codes.

6.1. Brief description of two multimaterial codes

The *first code*, [27,29], uses a staggered discretization on a logically rectangular grid consisting of general quadrilateral cells. In a staggered discretization, density, internal energy and pressure are cell-centered, and the velocity vector is defined at grid points. Staggered discretizations require special algorithms for remapping nodal quantities, (see for example, [8,36]). We will not describe such algorithms here and refer the interested reader to cited papers and the review paper [7]. For the purpose of this paper, it is only important that the remapping of nodal quantities may require some additional work which can increase the cost of overall remapping. In this code, we were originally using MFIB remapping for multimaterial problems and SRB remapping for single material problems. In the new version of the code, we have implemented the new hybrid remapping described in this paper, where in the second multimaterial step we use MFIB remapping. In the rest of the paper we will call this code ST – for staggered. In the ST code, the old grid may have non-convex cells as a result of the Lagrangian step.

The ST code employs MOF, [1,16], as the interface reconstruction method. For each mixed cell, the MOF package returns a set of triangles which represent a particular material. Even if a material can be represented by one polygon inside a mixed cell, the MOF package subdivides it into triangles. This is done to allow unified processing of the materials even in the case when a material is represented by several disjoint pieces.

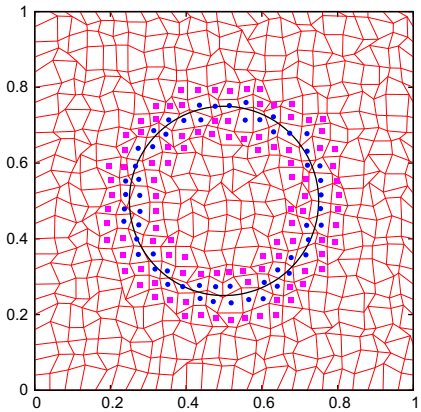


Fig. 11. Mixed and buffer cells. Mixed cells are marked by blue solid circles, and buffer cells are marked by magenta solid squares. Mixed cells are not affected in the first step of hybrid remapping. Data in buffer cells is updated in the first step of hybrid remapping. The second step of remapping requires the reconstruction of functions in buffer cells, because these cells will exchange information with mixed cells and/or between each other. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

When a new cell is intersected with a pure old cell (which may be non-convex) we use an algorithm for the intersection of two quads. The algorithm is based on half-plane intersections (half-planes are defined by the edges of the new cell) with the old cell; it is an extremely robust algorithm that for close to parallel line segments is bisection-based, [30]. In case of an intersection of a new cell with a multimaterial cell we intersect the new cell with each triangle representing the material using the same algorithm as quad-quad intersection. We note that in any case, the new cell is always represented as a quad and it is not subdivided into triangles.

The *second code* uses a cell-centered discretization on a general polygonal grid, [18]. In a cell-centered discretization, all quantities are cell-centered and a special algorithm for movement of the nodes is used. Therefore, all quantities are remapped in a unified way using only cells. In this code we were originally using MAIB remapping for multimaterial problems and SRB remapping for single material problems. In the new version of the code we have implemented the new hybrid remapping described in this paper, where in the second multimaterial step we use MAIB remapping. In the rest of the paper we will refer to this code as CC – for cell-centered. In the CC code, we also use the MOF method for interface reconstruction. The CC code differs from the ST code in the way intersections are performed. In this code not only pure sub-polygons representing materials in mixed cells are subdivided into triangles, but all pure cells of the old grid are subdivided into triangles. Cells of the new grid are also subdivided into triangles and all intersection-based remapping processes are reduced to triangle–triangle intersections, where the first triangle represents a cell of the new grid and the second triangle represents a pure or mixed cell of the old grid. Details of this approach can be found in [18]. For the purpose of this paper, it is important to note that this process is more expensive than the one used in the ST code, since it requires many more intersections. Note also that, in 3D, cells may become non-convex as their faces become non-planar. Thus, it will be necessary to perform subdivisions of the cells into tetrahedra to perform intersections. Therefore, the timings for the CC code can give an indication about the relative cost of the MAIB remapping method in the entire domain and about hybrid remapping in 3D.

The choice of these two codes for performing numerical experiments allows us to demonstrate different scenarios for our new multimaterial hybrid remapping strategy in multimaterial ALE codes as well as give an idea about the relative cost of the different methods.

All the remapping methods that we discussed have similar computational complexities whose order is the number of cells in the mesh. We omit a detailed theoretical cost analysis, since it would have to rely solely on the number floating point operations as a cost measure. Such an analysis of algorithms of comparable complexity can be very misleading, since factors other than the number of floating point operations can significantly impact the computational performance of an algorithm. One of such factors that impacts the performance of computational codes on modern computers is the layout of data in memory (see, for example, [11]). Our first code uses a structured mesh, and the second uses an unstructured mesh. Neither of them has been written with a view toward optimizing their data layout for the specific remapping algorithms that they employ. Instead of theoretical estimates of the computational cost, we provide times for the remapping step in the numerical examples in the following sections to give an indication of the relative performance of the individual remapping algorithms as they were implemented in our two codes.

6.2. Stand-alone cyclic remapping

In this section we compare different remapping methods using a stand-alone cyclic remapping problem. To that end, we define masses of the materials on the initial grid according to a specified density function, then do several remapping steps changing the grid and using results of the previous remapping step as initial data for the next remapping step. After several remapping steps we return to the original grid and compare the remapped to the original masses. This approach for testing remapping was introduced in [39].

The computational domain is a unit square. In this domain we have a circle of radius 0.25 with its center at (0.5,0.5). For a single-material problem, this circle is used to define a discontinuous density function. For a multimaterial problem with two materials, this circle contains a material which is different from the material in the rest of computational domain. In Fig. 12(a) we depict this circle and a uniform grid.

We define the initial distribution of density as follows

$$f(x,y) = \begin{cases} 1 + e^{10\sqrt{(x-1/2)^2+(y-1/2)^2}} & \text{for } \sqrt{(x-1/2)^2+(y-1/2)^2} \leqslant 1/4, \\ 1 + e^{6\left(\sqrt{(x-1/2)^2+(y-1/2)^2}-1/4\right)} & \text{in the rest of computational domain.} \end{cases} \tag{34}$$

This density distribution is used both for the single-material and for the two-material problem.

For the single-material problem, results of hybrid remapping are identical to SRB remapping. We use the hybrid method for the single-material case to investigate the computational cost of sweeping through the mesh to identify pure and mixed cells.

For the multimaterial problem the interface between materials inside a mixed cell is represented by a segment of a straight line. This line segment is determined using a PLIC (piece-wise linear interface construction) method. In this paper, we use the MOF method, which requires knowledge of volume fraction and centroid of each material. Interface reconstruction results in two pure material sub-polygons in each mixed cell. Mass and mean density is then computed in each pure

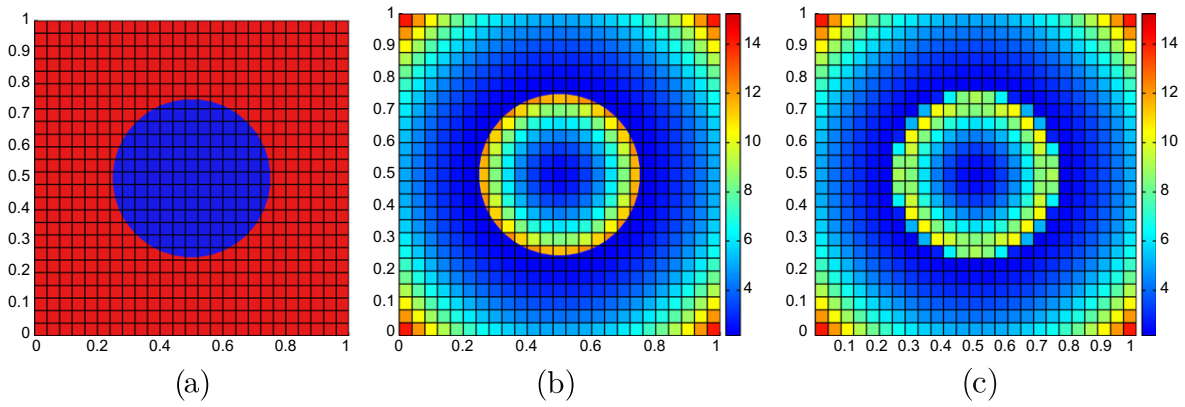


Fig. 12. Cyclic remapping problem: (a) Initial square grid. (b) Initial density distribution for multimaterial problem; (c) Initial density distribution for single-material problem.

material sub-polygon using the corresponding density function (34). The color map for mean values of density for the multimaterial problem is presented in Fig. 12(b).

For the single-material problem, the entire computational domain is occupied by one material and the circle is only used to define a discontinuous density. In this case, formally, there are no mixed cells. However, in each cell which is intersected by the circle we define one mass by summing masses of "materials" as described for the multimaterial case. The mean density for such cells equals mass divided by total volume. The color map for mean values of density for the single-material problem is presented in Fig. 12(c).

In this section we use the sequence of the meshes, $\{(x_i^n, y_i^n)\}; n = 0, \ldots n_{max}$ defined as follows

$$x_i^n = x_i^0 + \alpha(t^n)\sin(2\pi x_i^0)\sin(2\pi y_i^0), \quad y_i^n = y_i^0 + \alpha(t^n)\sin(2\pi x_i^0)\sin(2\pi y_i^0), \tag{35}$$
$$\alpha(t) = \begin{cases} t/5 & \text{if } t \leqslant 0.5, \\ (1-t)/5 & \text{if } t > 0.5, \end{cases} \quad t^n = n/n_{max}$$

where (x_i^n, y_i^n) is the actual position of node i at time t^n (that is, in the nth mesh in the sequence) and x_i^0, y_i^0 is its initial position.

We recall that in these tests in the hybrid algorithm, the intermediate mesh is formed using t^n positions of mixed points and t^{n+1} positions of pure points. We have checked the validity of these intermediate meshes explicitly.

The error of remapping is measured using the following relative norms. For single material problems

$$L_{1m} = \frac{\sum_{\forall c}|m_c^0 - m_c|}{\sum_{\forall c} m_c^0}, \tag{36}$$

where m_c^0 is the initial mass of cell c and m_c is the mass of the same cell on the initial grid after two remapping steps (initial to perturbed grid and back to initial grid).

For the multimaterial problem, we combine errors for each material into one error as follows

$$L_{1m}^{\text{mat}} = \frac{\sum_{\forall c}\sum_{\forall k}|m_{c,k}^0 - m_{c,k}|}{\sum_{\forall c}\sum_{\forall k} m_{c,k}^0}, \tag{37}$$

where k is the material index and $m_{c,k}^0, m_{c,k}$ are the initial and final mass of material k in cell c.

To investigate convergence and also compare CPU (Central Processing Unit) time required by different methods we perform cyclic remapping as described before on a series of refined grids.

In the first experiment, we use initial uniform grids of $N \times N$ cells, $N = 25, 50, 100, 200, 400$; and use grid movement defined by Eq. (35); we use a number of remapping steps equals the spatial resolution, that is, $n_{max} = N$.

First, we present the result of cyclic remapping using cell-centered remapping using the ST code. For the single-material case, we present the density color map on the initial grid, on the grid at $t = 0.5$ and on the initial grid after the second remapping, Fig. 13.

In Table 1, we present errors and timings for the single-material cyclic remapping test. Time measurements are performed on the same particular computer and, thus, it makes sense to compare relative CPU time needed for different methods.

First, Table 1 shows that all methods demonstrate approximately first-order convergence, which is what is expected when remapping a discontinuous function. Second, the accuracy of the hybrid method (which is the same as for the SRB method) almost equals the accuracy of the FIB methods. One might expect that intersection based methods should be more accurate than swept-region based methods. However, in this example this is not the case.

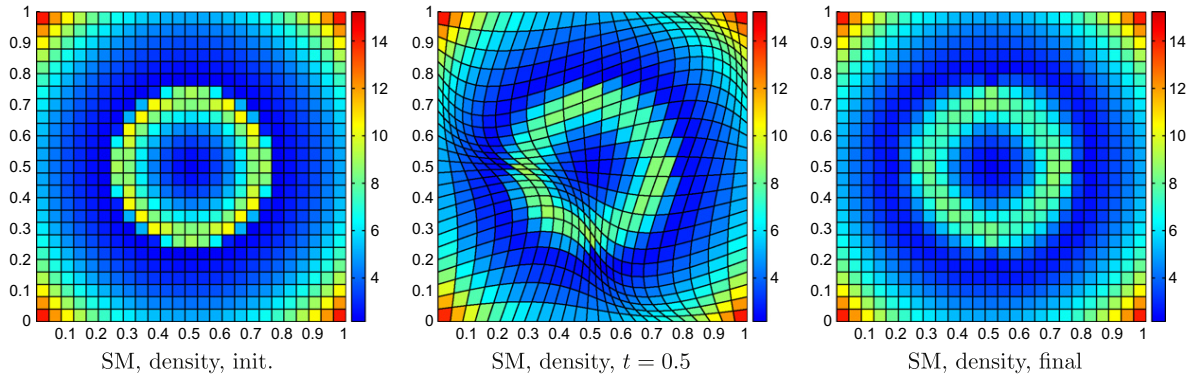


Fig. 13. Density distribution for initial, $t = 0.5$ grid (maximum distortion), and final grid of SM static remap test for double-exponential function using algorithms from ST code.

Table 1
L_1 mass errors and simulation times, T [s] (in seconds), for single-material case using algorithms from ST code.

Resolution	Swept	Intersection	Hybrid
25×25, L_{1m}	0.555×10^{-1}	0.577×10^{-1}	0.555×10^{-1}
25×25, T[s]	0.081	0.230	0.101
50×50, L_{1m}	0.431×10^{-1}	0.434×10^{-1}	0.431×10^{-1}
50×50, T[s]	0.739	1.956	0.885
100×100, L_{1m}	0.290×10^{-1}	0.291×10^{-1}	0.290×10^{-1}
100×100, T[s]	6.286	16.42	7.426
200×200, L_{1m}	0.183×10^{-1}	0.184×10^{-1}	0.183×10^{-1}
200×200, T[s]	50.00	133.1	61.45
400×400, L_{1m}	0.113×10^{-1}	0.114×10^{-1}	0.113×10^{-1}
400×400, T[s]	390.4	1048	470.1

Now we consider CPU time. For all methods CPU time increases approximately eight times with increasing resolution. Because with increasing of resolution we increase the number of cells by a factor of four and double number of remapping operations, this indicates that CPU time per cell is approximately a constant.

Now consider the finest resolution 400×400 grid, which can be considered close to asymptotic regime. For this example and this resolution SRB remapping saves about 63% of CPU time in comparison with FIB remapping, which gives an indication as to the possible improvement in CPU time. In the case of a single material the new hybrid remapping method produces results, which coincide with SRB remapping because there are no mixed cells and swept-region remapping is performed for the entire domain. However, the hybrid method requires preprocessing to mark nodes and edges which takes additional time. The new hybrid remapping method saves only about 55% of CPU time in comparison with FIB remapping. This indicates that some gain of the hybrid method over intersection-based remapping is lost due to preprocessing. In this particular case, the CPU time due to preprocessing overhead is about 8% with respect to FIB remapping, and 20% with respect to SRB remapping.

We now consider a multimaterial example, which uses the same density function as in the single-material case. However, we now have two materials – one is inside and another outside the circle. In the left panel in Fig. 14, we present materials (result of interface reconstruction) and color map for density for the initial grid; in the central panel the same information is depicted for the perturbed grid; and the right panel again depicts the same information for the final grid.

The number of mixed cells is changing with mesh movement. In Table 2 we present some statistics related to the number of mixed cells.

From Table 2 we can conclude that the percentage of mixed cells with respect to the total number of cells decreases and behaves as was described in Section 5.1. Additionally we note that the number of mixed cells does not change significantly for this example.

In Table 3, we present errors and timings for the multimaterial cyclic remapping example. This table demonstrates that for this example, the accuracy of MFIB and hybrid methods is almost the same and the convergence for both methods is between first- and second-order, but closer to second-order. This can be explained as follows. As a result of interface reconstruction, we effectively construct a finer grid which increases resolution near the interface and in our case near the line where the function has a discontinuity. Also the function reconstruction for the multimaterial case uses a special procedure to reconstruct the slope in pure sub-polygons in mixed cells, [27,29], which in contrast to many other codes leads to a non-zero slope because it uses information from pure sub-polygons containing the same material from surrounding cells (mixed and pure).

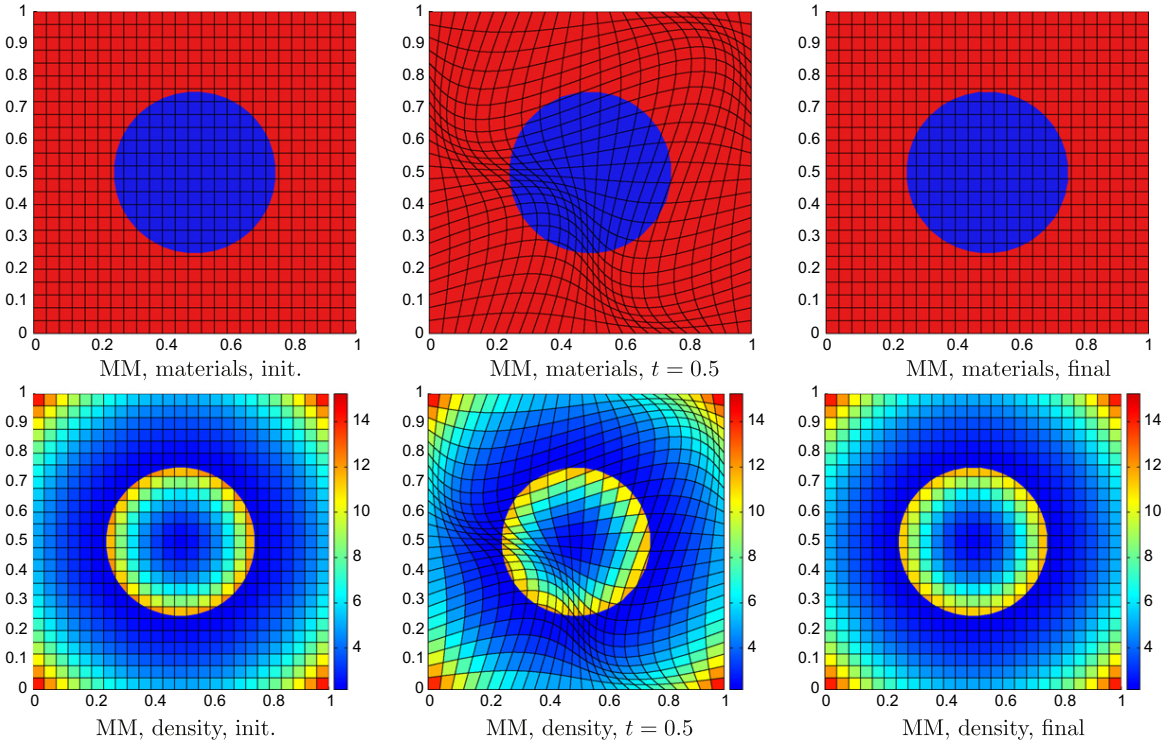


Fig. 14. Material and density distribution for initial, $t = 0.5$, and final grid using algorithms for ST code.

Table 2
Number of mixed cells in percent of the total number of cells for different resolutions. We present initial, final, average and maximum values. Maximum value is achieved at $t = 0.5$, when the mesh is most distorted.

Resolution	Initial	Final	Average	Max
25×25	7.68	7.68	8.64	9.92
50×50	4.00	4.00	4.32	4.72
100×100	1.80	2.00	2.15	2.34
200×200	0.95	0.99	1.07	1.16
400×400	0.49	0.50	0.53	0.58

Table 3
L_1 mass errors and simulation times, T[s] (in seconds), for the multimaterial case using algorithms from the ST code.

Resolution	Intersection	Hybrid
$25 \times 25, L_{1m}^{\text{mat}}$	0.684×10^{-2}	0.696×10^{-2}
25×25, T[s]	0.834	0.756
$50 \times 50, L_{1m}^{\text{mat}}$	0.184×10^{-2}	0.192×10^{-2}
50×50, T[s]	4.536	3.714
$100 \times 100, L_{1m}^{\text{mat}}$	0.444×10^{-3}	0.466×10^{-3}
100×100, T[s]	27.98	20.65
$200 \times 200, L_{1m}^{\text{mat}}$	0.100×10^{-3}	0.105×10^{-3}
200×200, T[s]	190	123.1
$400 \times 400, L_{1m}^{\text{mat}}$	0.241×10^{-4}	0.248×10^{-4}
400×400, T[s]	1392	818.8

We again consider the finest resolution, that is, 400×400. In this case, the hybrid method provides a savings of about 41% of CPU time over the MFIB method.

Now we present results for the same problem using the algorithms employed in the CC code. We remind the reader that the main differences to the ST code are that the CC code uses the MAIB method instead of MFIB and that intersections are done by subdividing all polygons into triangles.

Here, we present only tables. For the single material case, we present results in Table 4, which is analogous to Table 1. However, we do not present errors because they are in fact the same as for the ST code, since both the ST and CC codes use different implementations of the same formulas.

We note that the discrepancy in timings between the CC and ST codes for the SRB method is due to the fact that these two codes are different and that different computers and different compilers were used to generate these results. Comparing these results with Table 1, we conclude that the intersection-based algorithm implemented in the CC code is 7 times more expensive (for 400×400 resolution) than the intersection-based algorithm implemented in the ST code. This can be explained by the fact that each quad cell of the old and new grid is subdivided into triangles which results in the intersection of each triangle representing the new cell with triangles representing cells of the old grid in the neighborhood of the old cell. This alone can account for a factor of four difference in CPU time. Additionally, the AIB method is more expensive then the FIB method. There are also differences to the ST code's triangle intersection algorithm. Another very important difference is that the CC code is unstructured and memory access to geometric data from a neighboring cell may take much longer than in a structured code. Our goal here is not to analyze why the difference between two implementations of intersection algorithms is so dramatic, but to demonstrate what can happen in real relatively simple ALE codes.

It is most interesting to compare the relative cost of SRB, MAIB and hybrid methods using the CC code. We again consider the finest resolution – 400×400. For this example and this resolution SRB remapping saves about 96% of CPU time in comparison with MAIB remapping, which gives an indication of the imum possible gain in CPU time. The relative efficiency of SRB with respect to MAIB is much higher for the CC method because of the characteristics of the CC code, which we have mentioned above. The new hybrid remapping method saves about 93% of CPU time in comparison with MAIB remapping.

We now consider the multimaterial case. The results are presented in Table 5.

In the multimaterial case for the finest grid, the hybrid method saves about 89% of CPU time in comparison with the intersection-based method. Recall that when using the ST code, the hybrid method was saving only 41% with respect to the intersection-based method.

The CC code is an unstructured code, which can handle arbitrary polygons. We will demonstrate its performance on the cyclic remapping example described at the beginning of this section, but using an initial polygonal grid.

The initial polygonal grid is presented in Fig. 15(a). This grid is a centroidal Voronoi grid, [15], where there are four generators at the corners of the computational domain and the same number of generators as number of squares in the previous quad grid examples. The grid at $t = 0.5$ is presented in Fig. 15(b).

In Table 6, we present timing results for the single-material case (errors are similar to the quads test case and are not presented). In general, for a polygonal grid all the methods are more expensive because cells have more edges, and must be subdivided into more triangles than for a quad grid. The qualitative conclusions are similar to the quad grid case and we do not repeat them here.

Timings for the multimaterial case are presented in Table 7. Again, qualitative conclusions are the same as for the quad grid.

6.3. ALE calculation of a Rayleigh–Taylor instability

In this section, we are considering a Rayleigh–Taylor instability (RTI) problem for two immiscible ideal gases, one heavy and one light, with gravitational field directed vertically downward and with magnitude $g = 0.1$ The computational domain is

Table 4
Simulation timing, T[s] (in seconds), for single-material case using algorithms from the CC code.

Resolution	Swept	Intersect	Hybrid
25×25, T[s]	4.0×10^{-2}	2.2×10^{0}	1.4×10^{-1}
50×50, T[s]	3.1×10^{-1}	1.29×10^{1}	7.3×10^{-1}
100×100, T[s]	3.27×10^{0}	1.04×10^{2}	7.8×10^{0}
200×200, T[s]	2.83×10^{1}	8.35×10^{2}	7.4×10^{1}
400×400, T[s]	2.31×10^{2}	7.15×10^{3}	5.0×10^{2}

Table 5
Timing, T[s] (in seconds), for multimaterial case using algorithms from CC code.

Resolution	Intersect	Hybrid
25×25, T[s]	2.15×10^{0}	8.1×10^{-1}
50×50, T[s]	1.57×10^{1}	3.9×10^{0}
100×100, T[s]	1.19×10^{2}	2.04×10^{1}
200×200, T[s]	9.22×10^{2}	1.27×10^{2}
400×400, T[s]	7.31×10^{3}	8.1×10^{2}

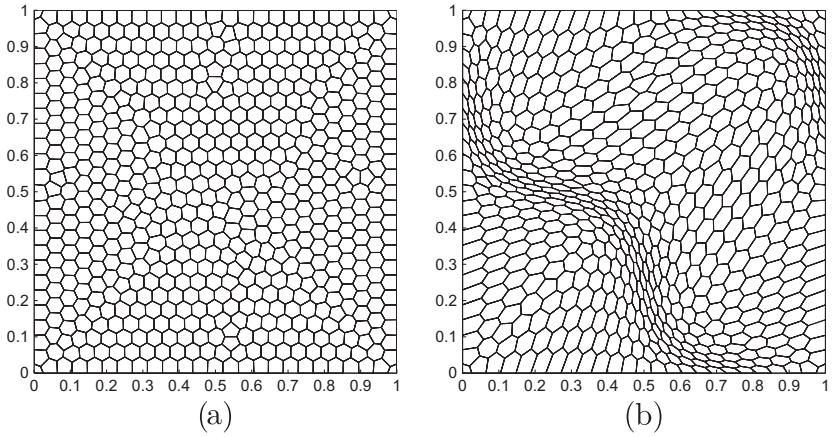


Fig. 15. CC code, polygonal mesh: (a) Initial mesh; (b) Mesh at $t = 0.5$.

Table 6
Polygonal grid: simulation times for single-material case – CC code.

Resolution	Swept	Intersect	Hybrid
25×25, T[s]	$6. \times 10^{-2}$	4.3×10^{0}	1.0×10^{-1}
50×50, T[s]	$5. \times 10^{-1}$	3.7×10^{1}	1.0×10^{0}
100×100, T[s]	$5. \times 10^{0}$	2.9×10^{2}	9.4×10^{0}
200×200, T[s]	$4. \times 10^{1}$	2.3×10^{3}	8.1×10^{1}

Table 7
Polygonal grid: simulation times for multimaterial case – CC code.

Resolution	Intersect	Hybrid
25×25, T[s]	9.3×10^{0}	1.84×10^{0}
50×50, T[s]	7.6×10^{1}	7.1×10^{0}
100×100, T[s]	5.85×10^{2}	3.4×10^{1}
200×200, T[s]	4.56×10^{3}	2.0×10^{2}

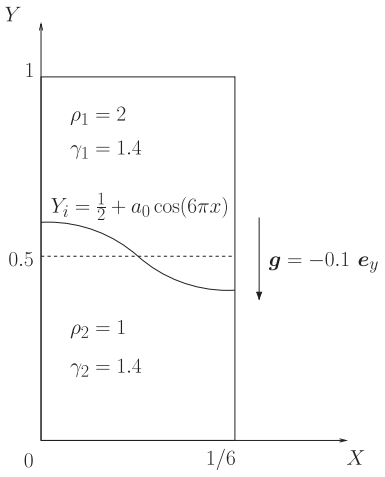


Fig. 16. Statement of Rayleigh–Taylor instability problem.

defined by $(x,y) \in [0,1/6] \times [0,1]$. The initial density for the heavier gas is $\rho_1 = 2$, and $\rho_2 = 1$ for the lighter gas. Both gases have the same adiabatic constant $\gamma = 1.4$. The interface has been deliberately perturbed according to the formula $Y_i(x) = \frac{1}{2} + 10^{-2} \cos(6\pi x)$, see also Fig. 16. The initial pressure distribution is approximately hydrostatic and is defined as follows:

$$P_1(x,y) = 1 + \rho_1\|\boldsymbol{g}\|(1-y), \quad \text{if } y > Y_i(x),$$
$$P_2(x,y) = 1 + \rho_1\|\boldsymbol{g}\|(1-Y_i(x)) + \rho_2\|\boldsymbol{g}\|(Y_i(x)-y), \quad \text{if } y < Y_i(x).$$

It is well known that this configuration is unstable and as time progresses, the heavier gas will sink and the lighter gas will rise through the formation of bubbles and spikes. This problem does not involve any shock wave, but the vorticity is so high that pure Lagrangian schemes eventually fail, therefore we use ALE methods. The final time for this problem is $t = 10$ s.

We model this problem using both the ST ALE and CC ALE codes. Because this problem involves two gases with the same equation of state we can run it both in single material mode and multimaterial mode similar to the stand-alone remapping test presented in the previous section. This allows us to do a similar analysis of the numerical results.

6.3.1. Single material calculations of RTI problem

For the single material RTI problem, we can compare the swept-region method for the entire domain, the intersection-based method for the entire domain, and the hybrid method.

We start with results obtained from the ST code. We have chosen to use this code to run the problem in Eulerian = Lagrange + Remap mode, that is, after each Lagrangian step we return back to the original grid. The initial grid is almost a rectangular grid aligned with the interface consisting of $17 \times 100 = 1700$ cells. Details of the calculation and in particular the color maps for density at the final time for the different methods are presented in [10].

We emphasize that we have demonstrated the accuracy of different methods on test examples in the previous Section. For problems, like this RTI problem, where we do not know the exact solution, we will not attempt to rigorously compare the accuracy of the methods. The main goal is to compare the efficiency of different methods for a reasonable resolution. However, the color maps of density for the different methods presented in [10] look very similar.

We now consider the timings for this problem, which are presented in Table 8. Our first observation is that the total time of calculation increases about ten times with increasing resolution, which indicates that there is no simple dependence of the total cost of ALE on the grid resolution. Our second observation is that for this example using the ST ALE code the cost of the hybrid method is almost the same as the cost of the swept-region-based method. So in the overall ALE calculation the relative cost of overhead related to the hybrid method is negligible. Finally, for the finest resolution, intersection-based remapping is about 27% more expensive than swept-region-based remapping. This may suggest that in this situation one may choose to use intersection-based remapping because in principle it should give more accurate results.

Now we present some results for the CC code. We chose to run this code using the initial polygonal grid. Again details are given in [10]. We run this problem in real ALE mode, where the grid is smoothed in each time step.

For the CC ALE code, at a resolution comparable with 17×100 resolution on a rectangular grid, it takes 3.6×10^3 s to run this test using swept-region-based remapping, 3.83×10^4 s to run it using intersection based remapping and 4.54×10^3 when the hybrid method is used. So the intersection-based method is about eight times more expensive than the hybrid method and ten times more expensive than the swept-region-based method. This is a dramatic difference in comparison with the ST code, where the intersection-based method is only 1.37 times more expensive than the hybrid method and 1.4 times more expensive than the swept-region-based method. However, overall, the CC code is still much more expensive than the ST code.

6.3.2. Multimaterial calculations of for the RTI problem

Similar to the single material RTI problem, we first present results obtained by ST ALE in Eulerian = Lagrange + Remap mode.

In Table 9, we present timings for different resolutions. We observe in Table 9 that for this problem, intersection-based remapping in the ST code is about 31% more expensive than hybrid remapping. We remind the reader that the timings in Table 9 are for the entire calculation.

Now, consider results obtained with the *CC multimaterial ALE code*, which we use in real ALE mode. The CC ALE code takes 9.65×10^3 s to run using hybrid remapping and 3.51×10^4 s to run using intersection-based remapping for a resolution comparable with 17×100 resolution on a rectangular grid, so the intersection-based method is about 3.63 times more expensive than the hybrid method. For the ST ALE code and similar spatial resolution the intersection-based method was approximately 1.2 times more expensive than the hybrid method. The details of this calculation can be found in [10].

We note that the number of mixed and buffer cells is changing during calculation because "length" and complexity of the interface between materials are growing. In particular for the 17×100 resolution calculation performed by the ST code this

Table 8
Single material RTI problem – ST code timing results.

Resolution	Swept	Intersection	Hybrid
17×100	1.42×10^3	2.08×10^3	1.51×10^3
33×200	1.15×10^4	1.65×10^4	1.19×10^4
66×400	9.80×10^4	1.41×10^5	1.01×10^5
133×800	1.02×10^6	1.40×10^6	1.04×10^6

Table 9
Multimaterial RTI problem – ST code timing results.

Resolution	Intersection	Hybrid
17×100	3.47×10^3	2.95×10^3
33×200	2.68×10^4	2.15×10^4
66×400	2.37×10^5	1.78×10^5
133×800	2.21×10^6	1.68×10^6

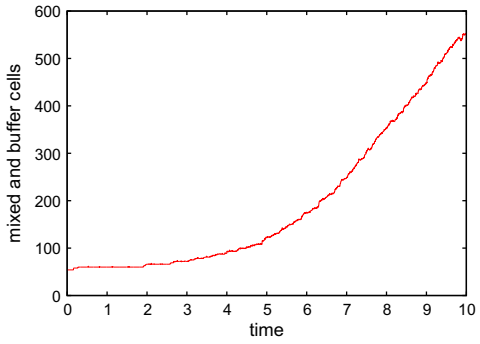


Fig. 17. Multimaterial Rayleigh–Taylor instability: number of mixed and buffer cells as function of time.

number of mixed and buffer cells is changing from 54 to 554. A time history of this number is presented in Fig. 17. This is one of the reasons why we do not make an attempt to estimate CPU time per cell for a realistic ALE calculation.

6.4. Shock-bubble interaction

This test case is taken from the paper [43] (see also, [18]). The setup consists of a Helium bubble surrounded by air. The initial domain is the rectangular box $[0, 0.650] \times [-0.089, 0.089]$, Fig. 18. The bubble is a circle defined by the coordinates of its center $(x_c, y_c) = (0.320, 0)$ and its radius $R_b = 0.025$. We prescribe wall boundary conditions on all boundaries with the exception of the right boundary, where we impose a piston-like boundary condition defined by the inward velocity $\boldsymbol{V}^{\star} = (u^{\star}, 0)$. The incident shock wave produced by motion of the piston is defined by its Mach number, $M_s = 1.22$. The bubble and the air are initially at rest. The initial data for Helium are $(\rho_1, P_1) = (0.182, 10^5)$, its molar mass is $\mathcal{M}_1 = 5.269 \cdot 10^{-3}$ and its adiabatic constant is $\gamma_1 = 1.648$. The initial data for air are $(\rho_2, P_2) = (1, 10^5)$, its molar mass is $\mathcal{M}_2 = 28.963 \cdot 10^{-3}$ and its adiabatic constant is $\gamma_2 = 1.4$. Using the Rankine–Hugoniot relations, we find that the x-velocity of the piston is given by $u^{\star} = -124.824$. The x-component of the incident shock velocity is $D_c = -456.482$. The incident shock wave hits the bubble at time $t_i = 668.153 \times 10^{-6}$. The stopping time for our computation is $t_{end} = t_i + 674 \times 10^{-6} = 1342.153 \times 10^{-6}$. It corresponds to the time for which an experimental shadow-graph is displayed in [43].

This problem is a two-material problem and to model it, we use a multimaterial ALE code. As for the previous examples, we will present results both for the ST multimaterial ALE code as well as for the CC multimaterial ALE code. Because this is a true multimaterial problem, we compare only intersection-based and hybrid remapping.

We now present results obtained by the ST code. We chose to use an initial rectangular grid with $134 \times 36 = 4824$ cells, and to run this problem in real ALE mode smoothing the grid and remapping after every Lagrangian step. Details of the calculation are presented in [10].

It takes about 2078 s to run this calculation using the intersection based method and about 1529 s to run it using hybrid remapping. Thus, the intersection-based method is about 35% more expensive than hybrid method, which is a similar result to the multimaterial Rayleigh–Taylor instability problem.

Now, we present results obtained by using the CC multimaterial ALE code. We chose to run this code in real ALE mode using grid smoothing and remapping after each Lagrangian step. The initial grid is polygonal consisting of 4847 polygons (see details in [10]).

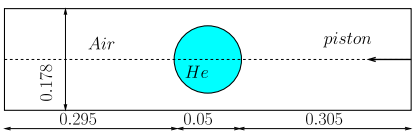


Fig. 18. Computational domain for shock-bubble interaction problem.

The CC multimaterial ALE code takes 3.51×10^4 s to run this problem using intersection-based remap and 9.65×10^3 s using hybrid remapping. So, intersection-based remapping is about 3.64 times more expensive than hybrid remapping. This result is similar to the multimaterial Rayleigh–Taylor instability problem.

7. Conclusions and discussion

In this paper, we have presented a new hybrid rezoning/remapping algorithm for multimaterial ALE methods. Hybrid remapping is performed in two steps. In the first step only nodes of the grid are moved that are inside subdomains occupied by single materials. At this stage, computationally cheap swept-region remapping is used. In the second step, nodes are moved that are vertices of mixed cells or vertices of some cells in the buffer zone around mixed cells. At this stage, intersection-based remapping is used. We have demonstrated the performance of this new method for both structured and unstructured polygonal grids in two dimensions as well as for ALE codes using staggered and cell-centered discretizations.

We have shown that the accuracy of intersection-based remapping and hybrid remapping is almost the same. This is demonstrated on a cyclic stand-alone remapping example, where the exact solution is known.

We showed that CPU time strongly depends on implementation of the intersection algorithm and other implementation issues.

We now discuss some considerations which must be taken into account when considering the efficiency of remapping in multimaterial ALE codes.

First, it is necessary to estimate the cost of the hydro part relative to other physics that is implemented in the code. In the case of multiphysics codes, which include hydrodynamics, strength of materials, radiation hydrodynamics, astrophysics, combustion and so on, [42,49,40,13,53], the relative computational cost of the hydro part may be small. In this situation, accuracy is the most important issue and, thus, intersection-based remap should be used everywhere. In multiphysics codes many quantities must be remapped. In this situation the relative cost of intersection based remap is lower because intersections are done only once and used for many quantities – the cost of remap is dominated by reconstruction of all variables and by computing fluxes, which has to be done for swept-region remapping, too. LASNEX, [53], is an example of a multiphysics code where intersection-based remapping is used for the entire grid.

Another consideration is the cost of remapping in ALE itself relative to the cost of the Lagrangian and rezoning stages. This depends on the type of grid the code is using: structured, block structured, or unstructured, and on the type of data structures that are used to represent it. One also needs to take into account the type of discretization the code is using: staggered or cell-centered. Another consideration is how many materials are used in the problem, which affects the fraction of the computational domain that is occupied by mixed cells. All these considerations may affect performance of the algorithm and in particular cost of memory access.

For 2D codes it is important to know if the Lagrangian step allows grid movement that creates valid non-convex cells or not. If so, then for intersection based remap more complicated and expensive intersection algorithms must be used, for example, based on subdivision of the cell into triangles, [18]. In 3D, subdivision of cells into tetrahedra appears to be unavoidable. Therefore, even for codes with relatively simple physics, the choice of remapping algorithm can be different in 2D and 3D.

Finally, the computer architecture on which algorithm will be implemented and available parallelization and acceleration strategies must be considered. (See [52,51] for examples of efficient implementations of compressible gas dynamics on LANL's Roadrunner supercomputer.)

Considerations presented in this section, and timing results presented in Section 6 for two different multimaterial ALE codes suggest that in multiphysics codes all options should be available, such that different remapping methods can be used for different problems and/or on different computer platforms.

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 No. DE-AC52-06NA25396 and supported by the DOE Advanced Simulation and Computing (ASC) program. The authors acknowledge the partial support of the DOE Office of Science ASCR Program. Milan Kucharik was supported by the Czech Ministry of Education Grants MSM 6840770022, MSM 6840770010 and LC528, and the Czech Science Foundation project P201/10/P086. Partial support of the BARRANDE MEB021020 program is also gratefully acknowledged. The authors thank A. Barlow and R. Loubère for stimulating discussions over many years.

References

[1] H.T. Ahn, M. Shashkov, Multi-material interface reconstruction on generalized polyhedral meshes, J. Comput. Phys. 226 (2007) 2096–2132.
[2] P. Anninos, Split and Unsplit Volume of Fluid Methods for Interface Advection, Technical Report UCRL-JC-139169, Lawrence Livermore National Laboratory, 2000.
[3] P. Anninos, Kull ALE: I, Unstructured Mesh Advection, Interface Capturing, and Multiphase 2T RHD with Material Interface, Technical Report, UCRL-ID-147297-PT-1, Lawrence Livermore National Laboratory, 2002.
[4] P. Anninos, Multiphase Advection and Radiation Diffusion with Material Interfaces on Unstrcutured Meshes, Technical Report UCRL-JC-150129, Lawrence Livermore National Laboratory, 2002.

[5] J.B. Bell, C.N. Dawson, G.R. Shubin, An unsplit, higher order Godunov method for scalar conservation laws in multiple dimensions, J. Comput. Phys. 74 (1988) 1–24.
[6] D.J. Benson, An efficient, accurate, simple ALE method for nonlinear finite element programs, Comput. Methods Appl. Mech. Eng. 72 (1989) 305–350.
[7] D.J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Comput. Methods Appl. Mech. Eng. 99 (1992) 235–394.
[8] D.J. Benson, Momentum advection on a staggered mesh, J. Comput. Phys. 100 (1992) 143–162.
[9] D.J. Benson, M. Souli, Arbitrary Lagrangian Eulerian and Fluid-Structure Interaction: Numerical Simulation, Wiley-ISTE, 2010.
[10] M. Berndt, J. Breil, S. Galera, M. Kucharik, P.H. Maire, M. Shashkov, Two-step Hybrid Remapping (conservative interpolation) for Multimaterial Arbitrary Lagrangian–Eulerian Methods, Technical report, Los Alamos National Laboratory Report LAUR-10-05438, 2010. <http://cnls.lanl.gov/~shashkov>.
[11] K.J. Bowers, B.J. Albright, L. Yin, B. Bergen, T.J.T. Kwan, Ultrahigh performance three-dimensional electromagnetic relativistic kinetic plasma simulation, Phys. Plasmas 15 (5) (2008). 055703–1–1.
[12] R.L. Bowers, J.R. Wilson, Numerical Modeling in Applied Physics and Astrophysics, Jones & Barlett Publishers, 1991.
[13] J.I. Castor, Radiation Hyddrodynamics, Cambridge University Press, 2004.
[14] P. Colella, Multidimensional upwind methods for hyperbolic conservation laws, J. Comput. Phys. 87 (1990) 171–200.
[15] Q. Du, V. Faber, M. Gunzburger, Centroidal Voronoi tesselations: applications and algorithms, SIAM Rev. 41 (1999) 637–676.
[16] V. Dyadechko, M. Shashkov, Reconstruction of multi-material interfaces from moment data, J. Comput. Phys. 227 (2008) 5361–5384.
[17] C. Farhat, P. Geuzaine, C. Grandmont, The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of the flow problems on the moving grids, J. Comput. Phys. 174 (2001) 669–694.
[18] S. Galera, P.-H. Maire, J. Breil, A two-dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction, J. Comput. Phys. 229 (2010) 5755–5787.
[19] G. Grandy, Conservative remapping and region overlays by intersecting arbitrary polyhedra, J. Comput. Phys. 148 (2) (1999) 133–466.
[20] C.W. Hirt, A. Amsden, J.L. Cook, An arbitrary Lagrangian–Eulerian computing method for all flow speeds, J. Comput. Phys. 14 (1974) 227–253.
[21] P. Hoch, An arbitrary Lagrangian–Eulerian strategy to solve compressible fluid flows, HAL : hal-00366858, version 1, 2009. <http://hal.archives-ouvertes.fr/docs/00/36/68/58/PDF/ale2d.pdf>.
[22] P. Hoch, Mesh quality and conservative projection in Lagrangian compressible hydrodynamic, in: Conference on Numerical Methods for Multi-material Fluid Flows; Czech Technical University in Prague on September 10–14, 2007; <http://www-troja.fjfi.cvut.cz/~multimat07/presentations/tuesday/Rebourcet_Hoch.pdf>.
[23] P. Hoch, Mesh quality and conservative projection in Lagrangian compressible hydrodynamic, in: Conference on Numerical Methods for Multi-material fluid flows; Czech Technical University in Prague on September 10–14, 2007; <http://www-troja.fjfi.cvut.cz/~multimat07/presentations/tuesday/Rebourcet_Hoch.pdf>.
[24] D.S. Kershaw, M.K. Prasad, M.J. Shaw, J.L. Milovich, 3D unstructured mesh ALE hydrodynamics with the upwind discontinuous finite element method, Comput. Methods Appl. Mech. Eng. 158 (1998) 81–116.
[25] P. Kjellgren, J. Hyvarien, An arbitrary Lagrangian–Eulerian finite element method, Comput. Mech. 21 (1998) 81–90.
[26] P. Knupp, L.G. Margolin, M. Shashkov, Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods, J. Comput. Phys. 176 (2002) 93–128.
[27] M. Kucharik, J. Breil, S. Galera, P.-H. Maire, M. Berndt, M. Shashkov, Hybrid remap for multi-material ALE, Computers & Fluids, 2010, doi:10.1016/j.compfluid.2010.08.004, in press.
[28] M. Kucharik, R. Garimella, S. Schofield, M. Shashkov, A comparative study of interface reconstruction methods for multi-material ALE simulations, J. Comput. Phys. 229 (2010) 2432–2452.
[29] M. Kucharik, M. Shashkov, Multi-material remap for staggered ALE in 2D, in: Conference on Numerical methods for multi-material fluids and Structures; Pavia, Italy, September 21–25, 2009; <http://www.eucentre.it/index.php/content/view/1287/415/lang,it/kucharik.pdf>.
[30] M. Kucharik, M. Shashkov, Conservative multi-material remap for staggered discretization, 2010, in preparation.
[31] M. Kucharik, M. Shashkov, B. Wendroff, An efficient linearity-and-bound-preserving remapping methods, J. Comput. Phys. 188 (2003) 462–471.
[32] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, G. Zanetti, Modeling merging and fragmentation in multiphase flows with SURFER, J. Comput. Phys. 113 (1994) 134–147.
[33] R.J. Leveque, High-resolution conservative algorithms for advection in incompressible flows, SIAM J. Numer. Anal. 33 (1996) 627–665.
[34] R.J. Leveque, Finite Volume Methods for Hyperbolic Problems, Cambridge, 2002.
[35] R.J. Leveque, O. Langseth, A wave propagation method for three-dimensional hyperbolic conservation laws, J. Comput. Phys. 165 (2000) 126–166.
[36] R. Loubère, M. Shashkov, A subcell remapping method on staggered polygonal grids for arbitrary-Lagrangian–Eulerian methods, J. Comput. Phys. 204 (23) (2004) 155–160.
[37] H.U. Mair, Review: Hydrocodes for structural response to underwater explosions, Shock Vibr. 6 (1999) 81–96.
[38] L.G. Margolin, Introduction to "an Arbitrary Lagrangian–Eulerian computing method for all flow speeds, J. Comput. Phys. 135 (1997) 198–202.
[39] L.G. Margolin, M. Shashkov, Second-order sign-preserving conservative interpolation (remapping) on general grids, J. Comput. Phys. 184 (1) (2003) 266–298.
[40] J.W. Murphy, A. Burrows, BETHE-HYDRO: an arbitrary Lagrangian–Eulerian multidimensional hydrodynamics code for astrophysical simulations, Astrophys. J. Suppl. Ser. 179 (2008) 209–241.
[41] J.S. Peery, D.E. Carroll, Multi-material ALE methods in unstructured grids, Comput. Methods Appl. Mech. Eng. 187 (2000) 591–619.
[42] D. Post, Codes Written by the National and International Computational Physics Community, Technical Report LA-UR-02-6284, Los Alamos National Laboratory, 2002. <http://www.highproductivity.org/026284coverCEEGodes.pdf>.
[43] J. Quirk, S. Karni, On the dynamics of a shock-bubble interaction, J. Fluid Mech. 318 (1996) 129–163.
[44] W.J. Rider, D.B. Kothe, Reconstruction volume tracking, J. Comput. Phys. 141 (1998) 112–152.
[45] M. Rudman, Volume tracking methods for interfacial flow calculations, Int. J. Numer. Methods Fluid 24 (1997) 671–691.
[46] J. Saltzman, Monotonic Difference Schemes for the Linear Advection Equation in Two and Three Dimensions, Technical report, Los Alamos National Laboratory Report LAUR-87-2479, 1987.
[47] J. Saltzman, An unsplit 3D upwind method for hyperbolic conservation laws, J. Comput. Phys. 115 (1994) 153–168.
[48] M. Shashkov, Closure models for multimaterial cells in arbitrary Lagrangian–Eulerian hydrocodes, Int. J. Numer. Methods Fluid 56 (2007) 1497–1504.
[49] D.J. Torres, M.F. Trujillo, KIVA-4: an unstructured ale code for compressible gas flow with sprays, J. Comput. Phys. 219 (2006) 943–975.
[50] A. Winslow, Numerical solution of the quasilinear Poisson equations in a nonuniform triangle mesh, J. Comput. Phys. 1 (1966) 149–172.
[51] J.G Wohlbier, R.B. Lowrie, B. Bergen, Acceleration of a multi-material hydrodynamics algorithm on roadrunner, in: Conference on Numerical Methods for Multi-material Fluids and Structures; Pavia, Italy, September 21–25, 2009, <http://www.eucentre.it/index.php/content/view/1287/415/lang,it/wohlbier.pdf>.
[52] P.R. Woodward, J. Jayaraj, P.-H. Lin, W. Dai, First experience of compressible gas dynamics simulation on the Los Alamos Roadrunner machine, Concurency Comput.: Practice Exp. 21 (2009) 2160–2175.
[53] G. Zimmerman, D. Kershaw, D. Bailey, J. Harte, LASNEX code for inertial confinement fusion, J. Opt. Soc. Am. 68 (1978) 549.

Journal of Computational Physics 231 (2012) 2851–2864

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

One-step hybrid remapping algorithm for multi-material arbitrary Lagrangian–Eulerian methods

Milan Kucharik [a,*], Mikhail Shashkov [b]

[a] Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Brehova 7, Praha 1, 115 19, Czech Republic
[b] XCP-4 Group, MS-F644, Los Alamos National Laboratory, P.O. Box 1663, Los Alamos, NM 87545, USA

ARTICLE INFO

Article history:
Received 25 August 2011
Received in revised form 20 December 2011
Accepted 24 December 2011
Available online 4 January 2012

Keywords:
Swept regions
Flux-based hybrid remap
Conservative interpolations
Multi-material ALE

ABSTRACT

In this paper, a new flux-based one-step hybrid remapping method for multi-material arbitrary Lagrangian–Eulerian (ALE) approach is introduced. In the vicinity of material interfaces, the swept region is intersected with pure material polygons in the Lagrangian mesh to construct the material fluxes. Far from interfaces, the fluxes are constructed in a standard swept-region manner without intersections. This method is conservative, second-order accurate and linearity-preserving (in case of straight material interfaces), and faster than method based on intersections, as shown on selected numerical examples.

1. Introduction

In the arbitrary Lagrangian–Eulerian (ALE) methods, a fast and accurate remapping method for data transfer of all fluid quantities from the Lagrangian to the rezoned mesh is necessary. Typically, methods based on swept regions are used in practical calculations [25,21], as these methods are significantly faster than the natural methods based on intersections. Unfortunately in the multi-material ALE, one needs to distribute the fluxes to particular materials, which we believe cannot be done consistently without intersections. The compatibility condition then leads to the requirement of doing the intersections in the entire computational domain. As the intersection-based methods are typically significantly more expensive, this leads to slower numerical computations.

To improve efficiency of the multi-material remapping, the concept of hybrid remap was introduced [5,16]. It is based on the main idea of performing the expensive intersections only in the vicinity of material interfaces, while in pure material regions the cheap swept region method can be used.

In [5], the basic concepts of the hybrid remapping methods were described and the two step hybrid remapping (TSHR) method was introduced. This method employs the rezoning and remapping phases separately for pure and mixed regions, in two distinct steps. In the first step, only nodes in pure regions are moved and the swept region method is used to remap all fluid quantities from the Lagrangian to this inter-mediate mesh. In the second step, the remaining nodes in mixed regions are moved and the multi-material fluid quantities are remapped using the intersection-based method. This method is applicable for any meshes, however, due to different treatment of pure and mixed nodes, symmetry of the problem can be violated, which can be very important in certain types of problems [24], for example for ICF applications [12,6]. Moreover, this method requires modifications in the main program routine, which may not be acceptable in the particular production code.

* Corresponding author.
E-mail addresses: kucharik@newton.fjfi.cvut.cz (M. Kucharik), shashkov@lanl.gov (M. Shashkov).

0021-9991/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2011.12.033

In [16], the combined one-step hybrid remapping (COSHR) method is introduced. This method explicitly combines the swept fluxes in pure material regions with the intersection-based fluxes close to material interfaces. At the buffer regions where both types of fluxes meet, a special treatment is required due to the existence of the corner fluxes in the intersection-based method, while they are missing in the swept region approach. This method keeps the problem symmetry, however, it is only applicable for logically rectangular meshes, its generalization to general polygons is not straightforward.

Both hybrid approaches reduce the computational cost of the remapping step, however, both approaches suffer from particular deficiencies. Here, a novel flux-based swept-region-with-intersections one-step hybrid remapping (SIOSHR) method is introduced, which does not suffer from these disadvantages. In this paper, we only focus on remap of the material masses and material information (volumes and centroids), no additional fluid quantities are considered. In this method, the swept regions are constructed even for multi-material regions of the mesh, and the corresponding multi-material fluxes are constructed by intersecting the swept regions with the pure material polygons of the involved cells. In case of self-overlapping swept region, the swept region is split into triangles which are then intersected with the pure material polygons separately. Therefore, the fluxes in the pure-material regions are consistent with the fluxes in the mixed regions – that is, for each cell there is the same number of fluxes computed by integration over the same swept regions, just the internal structure of the particular swept region can be different. Due to this construction, this method is consistent, conservative, linearity-preserving (even for multi-material cells in case of straight material interface), and more efficient than standard intersection-based method. It does not require modifications of the main routine and preserves symmetry.

This paper is organized as follows. In Section 2, the flux-based remapping framework incorporating geometric exchange integrals is described. In Section 3, the standard single-material remapping methods are reviewed: the remapping method based on cell intersections in Section 3.1, and the swept region remapping method in Section 3.2. In Section 4, the intersections are performed between the original cells and the standard swept regions (which may eventually be decomposed into triangles), and we demonstrate that this approach is equivalent to the standard intersection-based method from Section 3.1. Finally, the new swept-region-with-intersections one-step hybrid remapping method is introduced in Section 5, which combines the standard swept region fluxes from Section 3.2 in pure material regions with the swept region fluxes computed by intersections from Section 4 close to material interfaces. In Section 6, two numerical tests are used to demonstrate properties of the new method and compare it with the standard intersection-based and swept region approaches. This paper is concluded in Section 7.

2. General considerations

We assume that a two-dimensional domain is fully covered by polygonal cells (a particular cell is denoted by the c symbol in this paper), each cell has its vertices (mesh nodes, a particular mesh node is denoted by the n symbol here). In the context of ALE methods, we consider two different meshes, the original (old, Lagrangian) mesh with cells c, and a new (rezoned) mesh with cells $\tilde{c}$. We assume that both meshes have the same topology and are similar – we assume that every new node remains in the region covered by the cells adjacent to this node in the original mesh.

Each computational cell can contain one material (pure cell) or more materials (mixed cell). In this paper, we assume that the materials in a cell are represented by pure material polygons, which we denote as (c,k) – polygon of material k in cell c. We have the following information about the materials: their volume fractions $\alpha_{c,k}$, and their centroids $[x_{c,k},y_{c,k}]$. These quantities must be remapped also to provide the material information for the next step, in particular for the material reconstruction on the new mesh. In order to recover the pure material polygons from the material data, a material reconstruction method must be used. We employ the moment of fluid (MOF) material reconstruction method [10,1], which preserves the specified volume fractions exactly and tries to match the material centroids as close as possible. For comparison of the MOF method with different material reconstruction approaches, see [17].

Material mass in the new cells (as well as the other material quantities) can be composed from simple integrals evaluated in the new cell – we call them the geometric exchange integrals here. Let us define the cell and material integrals (which coincide in case of single-material cell):

$$I_c^\lambda = \int_c \lambda \, dx \, dy \quad \text{and} \quad I_{c,k}^\lambda = \int_{(c,k)} \lambda \, dx \, dy, \tag{1}$$

where λ is a simple polynomial function of x and y, typically $\lambda = 1$, $\lambda = x$, $\lambda = y$, or some higher order term if required. Using the Green theorem, these integrals can be reduced to boundary integrals and evaluated analytically for any polygon (even self-overlapping). The material volumes and volume fractions can be written then in the form:

$$V_{c,k} = I_{c,k}^1 \quad \text{and} \quad \alpha_{c,k} = \frac{I_{c,k}^1}{I_c^1}, \tag{2}$$

and the material centroid in the form:

$$x_{c,k} = \frac{I_{c,k}^x}{I_{c,k}^1}, \quad y_{c,k} = \frac{I_{c,k}^y}{I_{c,k}^1}. \tag{3}$$

This approach provides us all the data required for the MOF material reconstruction in the form of simple integrals.

Similarly, the material mass can be written as:

$$m_{c,k} = \int_{(c,k)} \rho_{c,k}(x,y)\ dx\ dy, \tag{4}$$

where we consider a piece-wise linear reconstruction of the density function in material k of cell c in the standard form:

$$\rho_{c,k}(x,y) = \rho_{c,k} + \left(\frac{\partial\rho}{\partial x}\right)_{c,k}(x - x_{c,k}) + \left(\frac{\partial\rho}{\partial y}\right)_{c,k}(y - y_{c,k}), \tag{5}$$

where $\rho_{c,k}$ is the density mean value in (c,k), and slopes $(\partial\rho/\partial x)_{c,k}$ and $(\partial\rho/\partial\ y)_{c,k}$ are computed using a standard function reconstruction process on a material by material basis. In our computations, we use the least-squares-based method [13], and the resulting slopes are limited by the Barth-Jespersen limiter [4].

After constructing the density slopes in each material of each computational cell, we can rewrite the mass formula (4) using the pre-computed integrals (1) as:

$$m_{c,k} = \rho_{c,k}\ I^1_{c,k} + \left(\frac{\partial\rho}{\partial x}\right)_{c,k}\left(I^x_{c,k} - x_{c,k}\ I^1_{c,k}\right) + \left(\frac{\partial\rho}{\partial y}\right)_{c,k}\left(I^y_{c,k} - y_{c,k}\ I^1_{c,k}\right). \tag{6}$$

Similar formula can be constructed for most fluid quantities, however we focus here on material mass and material information only. For more information about remap of a complete set of fluid quantities, see for example [8,9,23,19,20].

3. Review of standard single-material remapping methods

In this Section, the standard single-material remapping methods are reviewed. The method based on intersections is described and its flux form (including corner fluxes) is presented. Next, we review the remapping approach based on swept regions, where the particular flux is approximated by integration of the density function from one of its edge neighbors over the whole swept region.

3.1. Intersection-based remap

The intersection-based algorithm reviewed in [25] is based on the following formula:

$$\tilde{c} = \bigcup_{\forall c'} (\tilde{c} \cap c'), \tag{7}$$

combining the new cell $\tilde{c}$ from its intersections with the cells of the original (Lagrangian) mesh. This formulation can provide a global remapper between arbitrary meshes covering the same domain. In case of new mesh obtained from the original mesh by small nodal displacements (any mesh smoothing algorithm, typically), we can rewrite formula (7) as:

$$\tilde{c} = \bigcup_{c' \in C(c)} (\tilde{c} \cap c'), \tag{8}$$

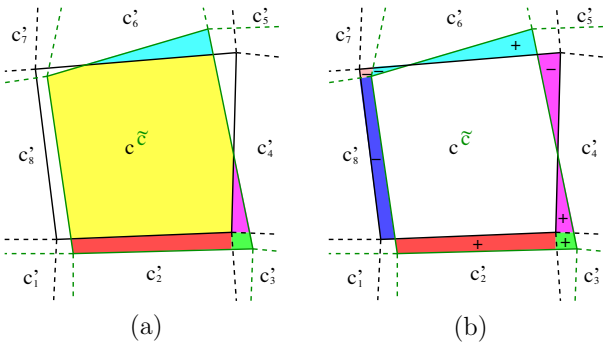


Fig. 1. Old Lagrangian cell c (shown by solid black boundary), and new rezoned cell $\tilde{c}$ (shown by solid green boundary) obtained from c by small nodal displacements. Remap treated by (a) standard intersections (8) and (b) by intersections in flux form (10). Intersections with various cells involved in both methods shown by different colors, pieces added and subtracted in the flux-based method denoted by the + and − signs. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

performing intersection only with the patch of cells including the original cell c and its neighborhood $C'(c)$, $C(c) = C'(c) \cup c$. This decomposition is shown in Fig. 1(a). See also [11,26] for more details. Formula (8) can be used to remap the cell volume:

$$V_{\tilde{c}} = I^1_{\tilde{c}} = \int_{\tilde{c}} 1 \; dx \; dy = \int_{\bigcup_{c' \in C(c)} \tilde{c} \cap c'} 1 \; dx \; dy = \sum_{c' \in C(c)} \int_{\tilde{c} \cap c'} 1 \; dx \; dy, \tag{9}$$

and similarly for other exchange integrals, which can be used for remap of most fluid quantities, as mentioned before.

Flux form of this approach (for the single-material case) is derived in [8,25]:

$$\tilde{c} = c \cup \bigcup_{c' \in C'(c)} ((\tilde{c} \cap c') \setminus (c \cap \tilde{c}')). \tag{10}$$

The terms in parentheses represent the positive and negative part of each flux into the neighboring cells (including the corner neighbors). This decomposition is demonstrated in Fig. 1(b). We can see, that the fluxes between c and c'_2 and c'_3 have only the positive component (they are added to c), fluxes between c and c'_7 and c'_8 have only the negative component (they are subtracted from c), while fluxes between c and c'_4 and c'_6 contain both negative and positive part of the flux. There is no flux between c and c'_1 and c'_5. As we can see in Fig. 1(b), each flux can generally have both components.

Formula (10) can be used for the remap of cell mass:

$$m_{\tilde{c}} = m_c + \sum_{c' \in C'(c)} F^m_{c,c'}, \tag{11}$$

where the mass fluxes are composed from their negative and positive parts:

$$F^m_{c,c'} = F^m_{\tilde{c} \cap c'} - F^m_{c \cap \tilde{c}'}. \tag{12}$$

These partial mass fluxes can be composed from the pre-computed exchange integrals similarly as we did for cell mass (6):

$$F^m_{\tilde{c} \cap c'} = \rho_{c'} \; I^1_{\tilde{c} \cap c'} + \left(\frac{\partial \rho}{\partial x}\right)_{c'} \left(I^x_{\tilde{c} \cap c'} - x_{c'} \; I^1_{\tilde{c} \cap c'}\right) + \left(\frac{\partial \rho}{\partial y}\right)_{c'} \left(I^y_{\tilde{c} \cap c'} - y_{c'} \; I^1_{\tilde{c} \cap c'}\right), \tag{13}$$

and similarly for $F^m_{c \cap \tilde{c}'}$. Similar formula can be used for other quantities, for examples of nodal momenta or cell internal energy, see [19,20].

3.2. Swept region remap

The notion of swept regions was described and used for example in [7,8,25,21]. A swept region Ω_e is defined by the motion of a particular edge e into its new position $\tilde{e}$ during the rezoning phase of the ALE algorithm. When viewed from one of its adjacent cells c, it is represented as a polygon with vertices $[n_1, \tilde{n}_1, \tilde{n}_2, n_2]$, where n_1 and n_2 are vertices of e such that n_1 is before n_2 with respect to cell c in the counter-clockwise direction. A swept region has always four vertices and it is enclosed by the original edge e, the same edge in the new mesh $\tilde{e}$, and two lines connecting the old edge vertices with their counterparts in the new mesh. There is always one and only one swept flux corresponding for each cell edge, no explicit corner fluxes are involved in this method.

The remap of cell mass can be written as:

$$m_{\tilde{c}} = m_c + \sum_{e \in E(c)} F^m_e, \tag{14}$$

where $E(c)$ stands for a set of all edges of cell c, and where the mass fluxes are computed as the integral of the reconstructed density over the swept region Ω_e:

$$F^m_e = \int_{\Omega_e} \rho_{c^*}(x,y) \; dx \; dy. \tag{15}$$

The reconstruction is taken from the cell c^*, which is either the original cell c, or its neighbor c' over the edge e, depending on the sign of algebraic area of the swept region:

$$c^* = \begin{cases} c & \text{if} \quad \int_{\Omega_e} 1 \; dx \; dy < 0 \\ c' & \text{if} \quad \int_{\Omega_e} 1 \; dx \; dy > 0 \end{cases}. \tag{16}$$

The integration can be again performed by composing the swept mass from the pre-computed integrals $I^1_{\Omega_e}$, $I^x_{\Omega_e}$, and $I^y_{\Omega_e}$, which can again be computed by their transformation to the line integrals along the boundary of the swept region using the Green theorem, and can be positive or negative.

As we can see, there is no intersection involved in the swept remap approach. Although the swept region can in reality overlap with several cells of the Lagrangian mesh, the swept mass is approximated by integrating the reconstructed density function from a single cell c^*. Fortunately, this additional discrepancy does not affect the formal order of accuracy of the method. Swept region remapper is second order accurate and the relative errors are comparable to the errors of the

intersection-based methods [21,5,22]. As no intersections are used in the remapper, the swept region approach is significantly faster than the intersection-based approaches. Therefore, the swept region methods are more popular in real ALE codes, allowing to get the problem solution in a shorter time. The swept region approach can be naturally extended to 3D [13], and even for certain special cases of meshes with changing connectivity [18].

4. Remap by intersections in swept regions

In this Section, we describe the remapping approach based on intersections of the swept region with the adjacent cells of the original mesh. Similar approach based on classification of all possible swept region shapes is described in [8] for the particular case when one of the meshes is a regular square mesh. In case of self-overlapping swept regions, its decomposition into triangles [15,14] is used. This method basically combines the swept region approach from Section 3.2 with the intersection-based approach from Section 3.1. We demonstrate here, that this approach is equivalent to the flux-form of the intersection-based approach (11), (12) based on decomposition (10).

For possible configurations of a swept region, see Fig. 2. Its shape can be convex (a), non-convex (b), and even self-intersecting (c).

First, let us explain the algorithm for the case of a compact (not self-overlapping) swept region. The algorithm is based on the following decomposition of the new cell:

$$\tilde{c} = c \cup \bigcup_{e \in E(c)} \left(\left(\bigcup_{\substack{c' \in C(e) \\ V(\Omega_e) \geqslant 0}} c' \cap \Omega_e \right) \setminus \left(\bigcup_{\substack{c' \in C(e) \\ V(\Omega_e) < 0}} c' \cap \Omega_e \right) \right), \tag{17}$$

where $C(e)$ is a set of all cells neighboring with edge e, which share at least one vertex with e (6 cells in case of logically rectangular meshes), and $V(\Omega_e)$ stands for the signed volume of the swept region Ω_e in the counter-clockwise sense. The swept region for edge $e = e_4$ (viewed from cell c) is again defined as a polygon with vertices $[n_1, \tilde{n}_1, \tilde{n}_2, n_2]$, where n_1 and n_2 are vertices of e such that n_1 is before n_2, when watching it from c in the counter-clockwise direction. Let us compare decomposition (17) with the decomposition used in the flux form of the intersection-based approach (10).

The intersection-based formula (10) evaluated for the convex case shown in Fig. 2(a) can be written as:

$$\tilde{c} = c \cup (c_4 \cap \tilde{c}) \setminus (c \cap \tilde{c}_2) \setminus (c \cap \tilde{c}_6). \tag{18}$$

The first intersection is shown as the magenta, vertically-lined rectangle, the next two intersections represent the lower and upper horizontally-lined white triangles. Now, we evaluate the intersections in the swept region given by formula (17) and obtain:

$$\tilde{c} = c \cup (c_4 \cap \Omega_{e_4}) \setminus (c \cap \Omega_{e_2}) \setminus (c_4 \cap \Omega_{e_2}) \setminus (c \cap \Omega_{e_6}) \setminus (c_4 \cap \Omega_{e_6}). \tag{19}$$

After composing the swept regions according to Fig. 2(a), we can rewrite it as:

$$\tilde{c} = c \cup ((c_4 \cap \tilde{c}) \cup T_1 \cup T_2) \setminus (c \cap \tilde{c}_2) \setminus T_1 \setminus (c \cap \tilde{c}_6) \setminus T_2 = c \cup (c_4 \cap \tilde{c}) \setminus (c \cap \tilde{c}_2) \setminus (c \cap \tilde{c}_6). \tag{20}$$

Here, T_1 and T_2 represent the triangles with the square pattern attached to nodes n_1 and n_2. As we can see, both triangles cancel and the formula reduces exactly to the same form as the intersection-based method.

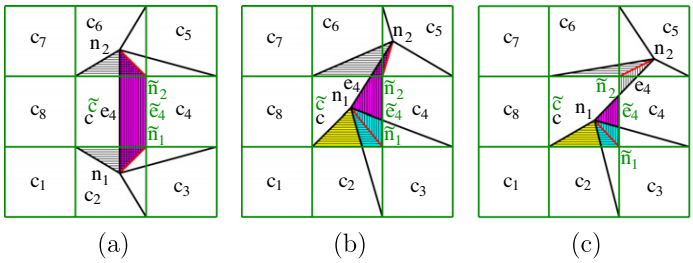


Fig. 2. Old disturbed cell c (shown by solid black boundary) and new regular cell $\tilde{c}$ (shown by solid green boundary) with its neighbors c_1–c_8. Patches coincide with the exception of edge e_4 (between c and c_4), which moved to its new position $\tilde{e}_4$, and created three swept regions of c – with cells c_2, c_4, and c_6. Three principally different typical shapes of swept region shown: (a) convex, (b) non-convex, and (c) self-intersecting. Parts of swept regions belonging to different original cells are shown in different colors. Swept region with c_4 marked by vertical pattern, swept regions with c_2 and c_6 by horizontal pattern, and overlapping parts marked by both patterns. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Let us try the same analysis for the non-convex case shown in Fig. 2(b). The intersection-based formula (10) can be evaluated as:

$$\tilde{c} = c \cup (c_2 \cap \tilde{c}) \cup (c_3 \cap \tilde{c}) \cup (c_4 \cap \tilde{c}) \setminus (c \cap \tilde{c}_5) \setminus (c \cap \tilde{c}_6), \tag{21}$$

where the first term in parentheses is shown as the yellow triangle, the second term is shown as the blue quadrilateral, and the third term is shown by the vertically-lined magenta quadrilateral. The last two terms together represent the horizontally-lined white triangle. Now, we can evaluate the intersections in the swept region given by formula (17):

$$\tilde{c} = c \cup (c_2 \cap \Omega_{e_2}) \cup (c_3 \cap \Omega_{e_2}) \cup (c_3 \cap \Omega_{e_4}) \cup (c_4 \cap \Omega_{e_4}) \setminus (c_4 \cap \Omega_{e_6}) \setminus (c \cap \Omega_{e_6}). \tag{22}$$

We can again compose the swept regions from pieces belonging to different new cells:

$$\tilde{c} = c \cup (c_2 \cap \tilde{c}) \cup (c_3 \cap T_{1,l}) \cup (c_3 \cap T_{1,r}) \cup (c_4 \cap (\tilde{c} \cup T_2)) \setminus (c_4 \cap T_2) \setminus (c \cap (\tilde{c}_5 \cup \tilde{c}_6)), \tag{23}$$

where $T_{1,l}$ and $T_{1,r}$ represent the horizontally- and vertically-lined blue triangles attached to node n_1, and T_2 represents the magenta triangle with square pattern attached to node n_2 in Fig. 2(b). The T_2 term cancels, and after substituting for $T_{1,l} \cup T_{1,r} = c_3 \cup \tilde{c}$ (which is obvious from Fig. 2(b)), we can rewrite it exactly in the same form, as (21). Therefore, for the non-convex case, the method of intersections in the swept region is equivalent to the intersection-based approach also.

Finally, let us analyze the self-overlapping configuration shown in Fig. 2(c). In this case, performing the intersections is not only technically difficult, but in fact we need to add one piece and remove the other piece. Therefore, we split the swept region naturally to two triangles – $\Omega_e^+ = [n_1, \tilde{n}_1, e \cap \tilde{e}]$ and $\Omega_e^- = [e \cap \tilde{e}, \tilde{n}_2, n_2]$, and the whole process is done in the same manner for each triangle separately. In this case, we need to use generalization of formula (17):

$$\tilde{c} = c \cup \bigcup_{\substack{e \in E(c) \\ \Omega_e \text{ compact}}} \left(\left(\bigcup_{\substack{c' \in C(e) \\ V(\Omega_e) \geqslant 0}} c' \cap \Omega_e \right) \setminus \left(\bigcup_{\substack{c' \in C(e) \\ V(\Omega_e) < 0}} c' \cap \Omega_e \right) \right) \cup \bigcup_{\substack{e \in E(c) \\ \Omega_e \text{self-intersecting}}} \times \bigcup_{\pm} \left(\left(\bigcup_{\substack{c' \in C(e) \\ V(\Omega_e^\pm) \geqslant 0}} c' \cap \Omega_e^\pm \right) \setminus \left(\bigcup_{\substack{c' \in C(e) \\ V(\Omega_e^\pm) < 0}} c' \cap \Omega_e^\pm \right) \right). \tag{24}$$

Here, by $\Omega_e^\pm$ we mean either Ω_e^+ or Ω_e^-, and $\cup_\pm$ stands for the joint of both options. Let us note that for the self-intersecting swept region, one of the triangles has a positive volume and its intersections with the original cells must be added, while the second triangle has a negative volume and all pieces must be removed.

For the case shown in Fig. 2(c), the intersection-based formula (10) can be evaluated as:

$$\tilde{c} = c \cup (c_2 \cap \tilde{c}) \cup (c_3 \cap \tilde{c}) \cup (c_4 \cap \tilde{c}) \setminus (c \cap \tilde{c}_4) \setminus (c \cap \tilde{c}_5) \setminus (c \cap \tilde{c}_6). \tag{25}$$

Now, we can evaluate the intersections in the swept region given by formula (17):

$$\tilde{c} = c \cup (c_2 \cap \Omega_{e_2}) \cup (c_3 \cap \Omega_{e_2}) \cup \left(c_3 \cap \Omega_{e_4}^+\right) \cup \left(c_4 \cap \Omega_{e_4}^+\right) \setminus \left(c \cap \Omega_{e_4}^-\right) \setminus (c \cap \Omega_{e_6}). \tag{26}$$

As before, we can compose the swept regions from their pieces belonging to different new cells, and rewrite the formula as:

$$\tilde{c} = c \cup (c_2 \cap \tilde{c}) \cup (c_3 \cap T_{1,l}) \cup (c_3 \cap T_{1,r}) \cup (c_4 \cap \tilde{c}) \setminus (c \cap T_{2,l}) \setminus (c \cap T_{2,r}) \setminus (c \cap \tilde{c}_6). \tag{27}$$

Here, $T_{2,l}$ and $T_{2,r}$ triangles represent the vertically- and horizontally-lined triangles at node n_2, and as we can see, $T_{2,l} \cup T_{2,r} = c \cap \tilde{c}_5$. After using this and similar expression for $T_{1,l}$ and $T_{1,r}$, we can rewrite the formula exactly in the form of (25). Again, both approaches are equivalent even for self-intersecting swept regions.

5. Swept-region-with-intersections one-step hybrid remap (SIOSHR)

Unfortunately, the swept region method in its pure form is not very suitable for multi-material simulations. In the case of the presence of material interfaces, it is not obvious how to approximate various material fluxes. We are aware of several approximations implemented in various codes [2,3]. For example, it is possible to distribute the total swept mass to the materials according to the cell volume fractions, or approximate the swept region by a single rectangle of the same volume. All these methods suffer from various problems: material flux can be bigger than the actual material mass in the cell, exchanging a particular material with a cell where it is not present, or just violating the material linearity preservation condition. In our opinion, in the case of multi-material cells with represented material interfaces, intersections are unavoidable if one wants to create a non-mixing method, which works in general situation. On the other hand, performing the intersections in the entire mesh makes the remapper inefficient. This leads us to the idea of the hybrid remapper – use the expensive

intersection-based methods only in the neighborhood of material interfaces, and switch to much cheaper swept region methods in pure material regions.

In this Section, a new hybrid remapping approach is presented – the swept-region-with-intersections one-step hybrid remapping (SIOSHR) method. In this approach, all fluxes are treated in the swept-like manner, no explicit corner fluxes are present. The swept regions in the vicinity of material interfaces are intersected with the pure material polygons to construct the material fluxes, and in case of self-overlapping swept region, they are decomposed into triangles which are then intersected separately. This is a direct multi-material generalization of the method described in Section 4. Far from material interfaces, the standard single-material swept fluxes are constructed. Let us note that the fluxes in the pure-material regions are compatible with the fluxes in the mixed regions, they are both computed using the swept regions, just the internal structure of the swept regions is different.

We need to emphasize here, that the SIOSHR method does not necessarily need to be used for multi-material calculations only. It can be practical even in single-material case to switch between standard swept region approach and the intersections in swept region approach (equivalent to the standard intersection-based method, as shown in Section 4). For example, in case of big density or pressure gradients, the standard swept region approach can produce high numerical error as the approximation of the density function in the swept region from a single cell can become erroneous. It can be practical in this situation to treat the swept regions crossing the high gradient by intersections, while in the smooth density regions, the cheap standard swept region approach can be used. However, we focus here on use of the SIOSHR method in case of multiple materials.

Let us now describe the algorithm of the SIOSHR method. At first, all mesh nodes must be marked as pure or mixed. If all cells attached to a particular node are pure and contain the same material, the node is marked as pure, otherwise, it is mixed, [5]. Similarly, mesh edges must be marked – if an edge connects two pure nodes, it is marked as pure, otherwise as mixed. For an example, see Fig. 3. In case of pure edge e, the exchange integrals over the swept region:

$$I^{\lambda}_{\Omega_e} = \int_{\Omega_e} \lambda \, dx \, dy, \quad \lambda = \{1, x, y\}, \tag{28}$$

are computed and used for construction of fluxes of all quantities, as we did in the standard swept region approach, while mixed edges are treated by intersections. For mixed cell c, formula similar to (24) can be directly used for a particular material k as:

$$(\tilde{c}, k) = (c, k) \cup \bigcup_{\substack{e \in E(c) \\ \Omega_e \text{ compact}}} \left(\left(\bigcup_{\substack{c' \in C(e) \\ V(\Omega_e) \geqslant 0}} (c', k) \cap \Omega_e \right) \setminus \left(\bigcup_{\substack{c' \in C(e) \\ V(\Omega_e) < 0}} (c', k) \cap \Omega_e \right) \right) \cup \bigcup_{\substack{e \in E(c) \\ \Omega_e \text{ self-intersecting}}} \times \bigcup_{\pm} \left(\left(\bigcup_{\substack{c' \in C(e) \\ V(\Omega_e^{\pm}) \geqslant 0}} (c', k) \cap \Omega_e^{\pm} \right) \setminus \left(\bigcup_{\substack{c' \in C(e) \\ V(\Omega_e^{\pm}) < 0}} (c', k) \cap \Omega_e^{\pm} \right) \right), \tag{29}$$

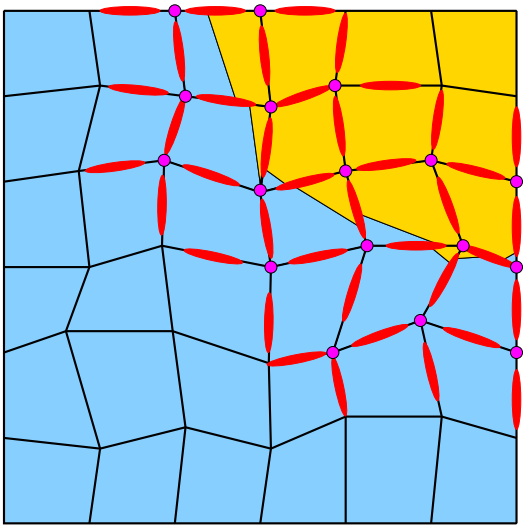


Fig. 3. Typical situation in 6×6 mesh with two materials (blue and yellow). Mixed nodes marked by magenta circles, mixed edges by red ellipses. Pure nodes and edges are not marked. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

where the swept regions are now intersected with each polygon of material k instead of the whole original cells. This directly leads to mass remap in the form:

$$m_{\tilde{c},k} = m_{c,k} + \sum_{\substack{e\in E(c)\\ \Omega_e \text{compact}}} \left(\left(\sum_{\substack{c'\in C(e)\\ V(\Omega_e)\geqslant 0}} F^m_{(c',k)\cap\Omega_e} \right) - \left(\sum_{\substack{c'\in C(e)\\ V(\Omega_e)<0}} F^m_{(c',k)\cap\Omega_e} \right) \right) + \sum_{\substack{e\in E(c)\\ \Omega_e \text{ self-intersecting}}} \times \sum_{\pm} \left(\left(\sum_{\substack{c'\in C(e)\\ V(\Omega_e^{\pm})\geqslant 0}} F^m_{(c',k)\cap\Omega_e^{\pm}} \right) - \left(\sum_{\substack{c'\in C(e)\\ V(\Omega_e^{\pm})<0}} F^m_{(c',k)\cap\Omega_e^{\pm}} \right) \right), \tag{30}$$

where the mass fluxes are obtained by integration of the reconstructed density function from material k of cell c' over the particular intersection, and can be directly composed from the pre-computed geometric exchange integrals in a similar way, as we did in the single-material case (13):

$$F^m_{(c',k)\cap\Omega_e} = \rho_{c',k}\, I^1_{(c',k)\cap\Omega_e} + \left(\frac{\partial\rho}{\partial x}\right)_{c',k} \left(I^x_{(c',k)\cap\Omega_e} - x_{c',k}\, I^1_{(c',k)\cap\Omega_e} \right) + \left(\frac{\partial\rho}{\partial y}\right)_{c',k} \left(I^y_{(c',k)\cap\Omega_e} - y_{c',k}\, I^1_{(c',k)\cap\Omega_e} \right) \tag{31}$$

and similarly for remaining F^ms.

As we can see in (29), for mixed edge e, we must distinguish, whether the corresponding swept region is compact or self-overlapping, similarly as we did in the single-material case (24). For an example, see Fig. 4(a), where a new and old meshes are shown containing three materials, the mixed nodes and edges are shown in Fig. 4(b). Let us note that this algorithm will work for arbitrary number of materials. If the particular swept region is compact, as shown in Fig. 4(c and d), the flux through this edge is simply obtained by intersecting the swept region with all pure material polygons in all adjacent cells. In case that the swept region is self-overlapping, as shown in Fig. 4(e and f), we break it in two triangles and intersect them separately.

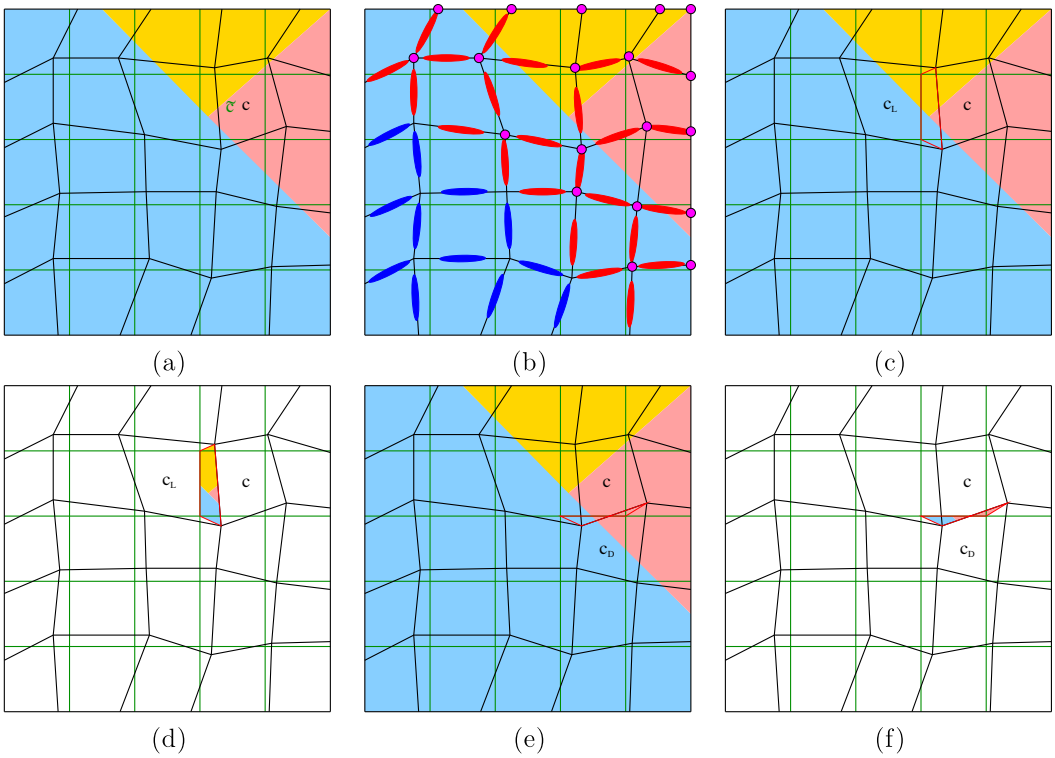


Fig. 4. Situation in 5×5 mesh with three materials. (a) Original cell c (shown by solid black line) and rezoned cell $\tilde{c}$ (solid green line) containing blue, yellow, and pink materials. (b) Mixed nodes marked by purple circles. Mixed internal edges marked by red ellipses, pure edges by blue ellipses. (c and d) Convex swept region between c and left neighbor c_L, contains all three materials. (e and f) Self-intersecting swept region between c and down neighbor c_D. Swept region is split into two triangles, left triangle contains fluxes of both blue and pink materials, where flux of the blue material is obtained by intersecting the triangle with the blue polygons in two different cells: the original cell c and its left neighbor c_L. Right triangle contains only pink material coming from c_D. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

We demonstrate here, that a flux can generally have both positive and negative components, which can generally contain more materials.

Let us note, that this approach is very elegant when compared with the previous hybrid approaches [5,16], as all fluxes are of the same nature now, all fluxes (pure and mixed) are treated in a consistent manner, no corner fluxes are required. Particularly, when compared with the COSHR hybrid method, the SIOSHR method is simple to implement and generalizable for general polygonal meshes. On the other hand, we need to store more exchange integrals for each flux, not only its negative and positive part as in all previous methods. This is caused by the fact, that the (for example mass) flux through a particular edge is constructed by intersection of several different reconstructed (for example density) functions, so we need to store not only the total exchange integrals over the entire swept region, but also all the partial integrals over its intersections with the involved cells. For an example, see Fig. 4(f), where the highlighted flux involves parts not only from cells c and c_D, but also a small triangle belonging to c_L. Integrals of this small triangle must be kept separately as they are used for integration of the reconstructed function from c_L, to construct the total quantity flux properly. Therefore, in case of logically rectangular meshes, one needs to store six times the number of integrals for each edge (instead of just one set of integrals), causing higher memory requirements of the method. This can cause lower performance of the SIOSHR method as the memory access can become significant.

One can think, that the flux can contain a piece of material, which is actually not present in one of the adjacent cells. For an explanation, see the situation shown in Fig. 5. Both cells c_{LL} and c_{UL} are pure and include only the red material. Cell c_{UR} is also pure, but includes only blue material, and cell c_{LR} is mixed. As we can see in Fig. 5(b), when the upper edge of c_{LL} is treated, the flux between c_{LL} and c_{UL}(which both contain only red material) includes part of the blue material (piece 4), which is removed from the lower cell. However, this problem is fixed when the right edge of c_{LL} is treated as can be seen in Fig. 5(c) – the same piece 4 of blue material is added to c_{LL}, so these contributions cancel and no blue material plays any role in the final quantities of c_{LL}. This consistency is the main advantage of the intersection-based SIOSHR approach when compared with various approximations of the multi-material flux avoiding intersections, the contributions cancel exactly while in the approximate approaches usually some portion of the non-present material flux is considered and introduces disturbances in various fluid quantities.

This brings us to the notion of the indirect corner flux, which is embraced in the method. As we have mentioned before, all material exchange is comprised by the edge fluxes only, no explicit corner fluxes exist. However, the flux is implicitly present through the edge fluxes. This is demonstrated in Fig. 5(d) and (e). Cells c_{UL} and c_{LR} do not communicate with each other directly as no corner flux exists. The corner region includes pieces of materials denoted by numbers 3, 4, 6, and 7. As we can see, pieces 3 and 4 are moved from cell c_{LR} to cell c_{LL} by flux shown in Fig. 5(c), and the same pieces are moved from cell c_{LL} to cell c_{UL} by flux shown in Fig. 5(b). Similarly, pieces 6 and 7 are moved from cell c_{LR} to cell c_{UR} through the flux shown in Fig. 5(e), and the same pieces are moved from cell c_{UR} to cell c_{UL} by flux shown in Fig. 5(d). In the end, all these pieces are moved from the original cell c_{LR} to the new cell $\tilde{c}_{UL}$, however, the interaction is not direct, it is performed through the intermediate cells c_{LL} and c_{UR}.

Let us also note that the triangular decomposition of the swept region which we did for the self-intersecting case can be done for the case of non-convex swept region also, though it is not necessary. However, it can eliminate some technical problems related to intersections of non-convex polygons. In this case, the edge intersection defining the new point shared by both triangles must be computed by intersecting both original and new edges in the sense of infinite lines, as can be seen in Fig. 2(b). The volumes of both triangles has then the same sign and their contributions are both either added to or subtracted from the cell.

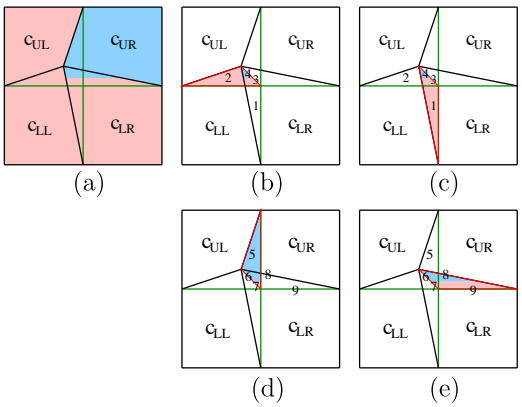


Fig. 5. Situation around pure cell c_{LL}, different materials in its neighbors. (a) Original mesh shown by solid black lines, new mesh shown by solid green lines, materials by red and blue colors. (b–e) All swept regions contain several materials. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

6. Numerical examples

In this Section, we demonstrate the properties of the SIOSHR method on two selected stand-alone cyclic remapping numerical examples from [16,5]. We define material volume fractions and material centroids in each cell of the initial computational mesh, and using the MOF material interface method, pure material polygons are defined in each cell. In each material polygon, material density is set by an analytic density function, which is then used for initialization of material masses. At the beginning of each step, the new (rezoned) computational mesh is generated from the old (Lagrangian) one. Next, all material quantities (material volume fractions, centroids, and masses) are remapped to the new mesh using the methods described in this paper. This process is repeated several times until the desired number of remapping steps is reached. This approach was designed in [25] to emphasize properties of the particular remapping method.

6.1. Double-linear function, straight interface

The first numerical test from [16] is a sanity check demonstrating consistency of the remapping method. The computational domain is a unit square, which is initially covered by an equidistant orthogonal computational mesh. We use a series of initial meshes containing $25^2, 50^2, \ldots, 400^2$ cells, number of remapping steps is proportional to the mesh size, for example, for 25^2 mesh, we do 25 remapping steps. The computational mesh is split by a straight line going through points [0.41,0] and [0.61,1] into two parts containing different materials. Density is defined by two different linear functions:

$$\rho(x,y) = 10 + 1\left(x - \frac{1}{2}\right) + 2\left(y - \frac{1}{2}\right) \tag{32}$$

right from the interface and

$$\rho(x,y) = 20 + 3\left(x - \frac{1}{2}\right) + 0.5\left(y - \frac{1}{2}\right) \tag{33}$$

in the left material. We also consider a single-material version of this problem, where just one material occupies the whole domain and density is defined by (32) everywhere. The sequence of meshes is defined by the smooth sine formula introduced in [25]:

$$x_i^n = x_i^0 + \alpha(t^n)\ \sin(2\ \pi\ x_i^0)\ \sin(2\ \pi\ y_i^0) \tag{34}$$

$$y_i^n = y_i^0 + \alpha(t^n)\ \sin(2\ \pi\ x_i^0)\ \sin(2\ \pi\ y_i^0), \tag{35}$$

where the time depending coefficient:

$$\alpha(t) = \begin{cases} t/5 & \text{for } t \leqslant 1/2, \\ (1-t)/5 & \text{for } t > 1/2, \end{cases} \tag{36}$$

and where $t^n = n/n_{\max}$. Here, $n_{\max}$ is the number of desired remapping steps, $[x_i^n, y_i^n]$ is the actual position of node i in time t^n, and $[x_i^0, y_i^0]$ is the initial position of the same node. This motion guarantees that the final computational mesh is identical with the initial one.

After the last remapping step, we measure the relative L_1 discrepancy of the numerical values from the initial values:

$$L_1 = \sum_{\forall c} \sum_{\forall k \subset c} \frac{\left| m_{c,k}^{t=0} - m_{c,k} \right|}{m_{c,k}^{t=0}}. \tag{37}$$

For the double-linear function with the straight interface, the SIOSHR method preserves linear functions up to machine accuracy. Let us note, that several cells change their status during the computation, they become mixed and pure again, materials are being removed from them, and still the SIOSHR method is exact and no disturbance is introduced.

In Table 1, we can see the comparison of computational times in different mesh resolutions. For the single-material case, the new SIOSHR method is compared with the method based on intersections[1] and with the fast method based on swept regions. For the multi-material case, the method based on swept regions is not available. Let us note, that these numbers correspond to one particular implementation in a particular staggered research code, all runs were performed on a regular AMD 2.4 GHz Opteron machine. The results can differ for a different implementation especially due to the overhead of the methods in different codes, see [5] for comparison of two different implementations of similar method in two significantly different codes.

As we can see, for the highest resolution in the single-material case, the swept region remap costs about 43% of the intersection-based remap, the SIOSHR hybrid method is about 7% more expensive. As this problem is single-material and

[1] For polygon intersections, we use a simple algorithm intersecting the old (in the ALE context generally non-convex Lagrangian) material polygon with the half-planes defined by edges of the new (in the ALE context convex rezoned) cell. This algorithm has n^2 complexity (faster algorithms typically have $n \log n$ complexity), however, it is simple, robust, and even fast as only small polygons (with typically $n \lessapprox 5$ vertices) are intersected.

Table 1
Comparison of computational cost (in seconds) of SIOSHR method with the standard approaches for single- and multi-material version of the double-linear function with straight interface problem in different mesh resolutions.

Method	25^2	50^2	100^2	200^2	400^2	$\frac{T^{400^2}}{T^{400^2}_{\text{int}}}$
SM, swept	0.1	0.8	8.0	69.6	591.7	0.43
SM, int	0.3	2.3	20.1	167.3	1383.1	1.00
SM, SIOSHR	0.1	1.0	9.6	80.0	688.3	0.50
MM, int	0.7	4.1	29.9	227.1	1853.3	1.00
MM, SIOSHR	0.5	3.0	19.3	142.9	1229.9	0.66

no intersections are done in the SIOSHR method, this overhead is caused by the nodal and edge marking processes and especially by the higher memory requirements for flux storage.

For the two-material case, the swept region approach is not available. The intersection-based remap is about 25% more expensive than it was in the single material case, simply because the intersections are performed in the entire mesh as it was done in the single-material case, just several more intersections are done in the mixed cells, and cost of the MOF material reconstruction method must be also added. The hybrid method on the other side did not perform any intersections in the single material case, but needs to do it in the multi-material case, so the increase of the computational cost is more significant. Still, the SIOSHR method reduces the computational cost of the simulation significantly, by about 33%.

6.2. Double-non-linear function, curved interface

This test was introduced in [5] and is similar to the previous example, however, the interface is not straight and the density functions are not linear. The computational domain and sequence of meshes is the same as before, however, the material interface is defined by a circle centered at the $[1/2, 1/2]$ position with the radius of $1/4$. The density function is exponential and is defined as:

$$\rho(x,y) = 1 + e^{10\sqrt{(x-1/2)^2+(y-1/2)^2}} \tag{38}$$

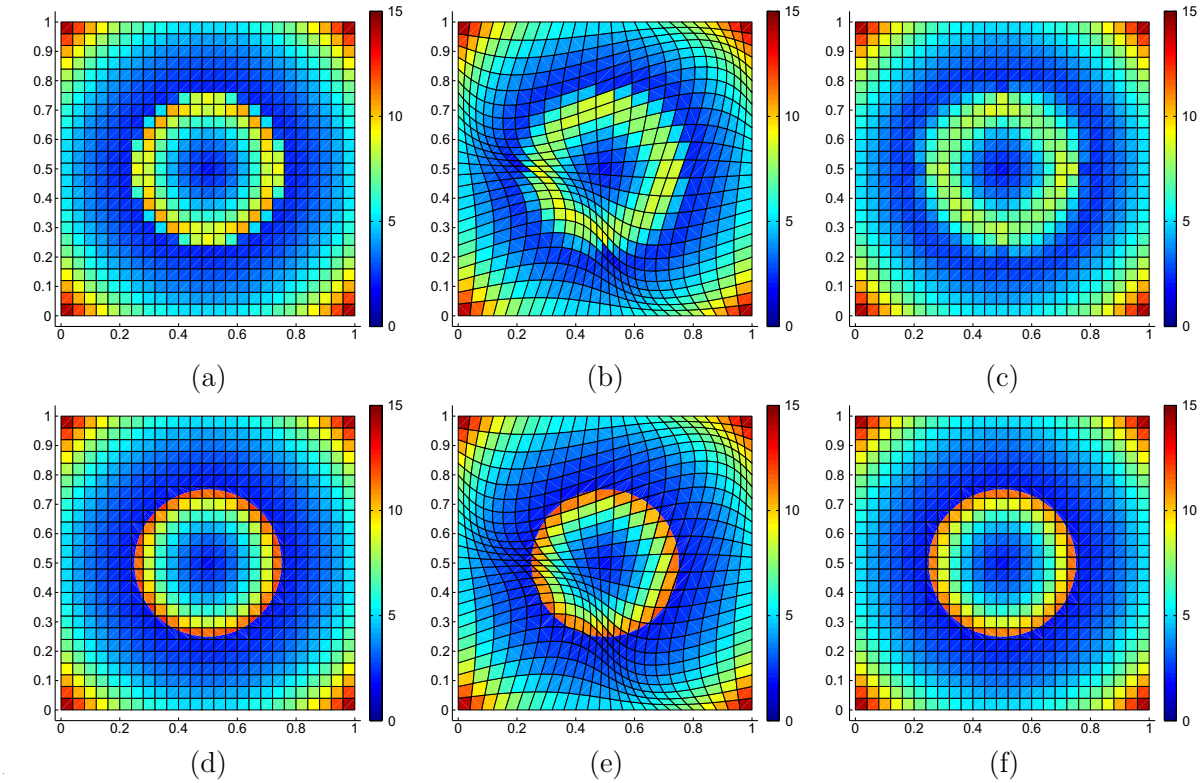


Fig. 6. Density profile for single-material (a–c) and multi-material (d–f) double-non-linear problem using the SIOSHR method. Initial (a and d), middle (b and e), and final (c and f) situation shown.

Table 2
Comparison of computational cost (in seconds) of SIOSHR method with the standard approaches for single- and multi-material version of the double-non-linear function with curved interface problem in different mesh resolutions.

Method	25^2	50^2	100^2	200^2	400^2	$\frac{T^{400^2}}{T^{400^2}_{\text{int}}}$
SM, swept	0.1	0.9	8.5	71.3	602.5	0.43
SM, int	0.3	2.4	20.8	170.9	1390.6	1.00
SM, SIOSHR	0.1	1.0	9.6	80.5	687.1	0.49
MM, int	0.9	5.3	34.3	245.7	1906.0	1.00
MM, SIOSHR	0.8	4.2	24.7	161.1	1248.2	0.65

Table 3
Comparison of L_1 error and order of accuracy of SIOSHR method with the standard approaches for single- and multi-material version of the double-non-linear function with curved interface problem in different mesh resolutions.

Method	25^2	50^2	100^2	200^2	400^2	$\log_2\left(\frac{L_1^{200}}{L_1^{400}}\right)$
SM, swept	0.555e−1	0.431e−1	0.290e−1	0.183e−1	0.113e−1	0.70
SM, int	0.577e−1	0.434e−1	0.291e−1	0.184e−1	0.114e−1	0.69
SM, SIOSHR	0.555e−1	0.431e−1	0.290e−1	0.183e−1	0.113e−1	0.70
MM, int	0.684e−2	0.184e−2	0.444e−3	0.100e−3	0.241e−4	2.05
MM, SIOSHR	0.677e−2	0.183e−2	0.440e−3	0.992e−4	0.237e−4	2.07

inside the circle, and

$$\rho(x,y)1 + e^{6\ (\sqrt{(x-1/2)^2+(y-1/2)^2}-1/4)} \tag{39}$$

outside.

In Fig. 6, we can see the density in three stages of the SIOSHR computation for the 25^2 mesh – after initialization, after remap number 12 (close to the maximal mesh distortion), and in the final moment after remap number 25. We can see both single- and multi-material setup of the simulation. For the single-material case, the cell density is constructed by reduction to average density values from the multi-material initialization. As we can see, the solution in the single-material case is severally diffusive, while in the multi-material case the solution stays resolved close to the material interface without any disturbance. Both single- and multi-material profiles are comparable with the results shown in [5].

Let us note that the single-material problem can be run by the SIOSHR method in the hybrid mode also, just a different marking is required. Instead of nodal marking based on the presence of material interfaces, one can mark the nodes according to a different criteria, for example in case of high density gradients.

In Table 2, we can see comparison of computational cost of the SIOSHR method with the intersection-based method, and in case of single-material problem also with the swept region method. As we can see, the results are similar to those presented for the double-linear test. In the single-material case, the computational times are basically the same as before. In the multi-material case, the absolute times are slightly higher, which is given by a higher number of mixed cells in the problem, however, the relative cost (proportional to the intersection-based method) remains the same. The hybrid approach saves about 33% of the computational cost of the intersections-based method.

In Table 3, we can see the comparison of the numerical error given by (37) for the non-linear test in both single- and multi-material cases. Due to the curved interface and non-linear functions, the numerical error is not zero any more. In the single-material case, all methods are close to first order accurate, which is expected due to remapping of strongly non-linear and discontinuous data. We can see that the SIOSHR method and the swept region method perform slightly better than intersections, see [22] for possible explanation. In the multi-material case, both intersection-based and SIOSHR methods are second order accurate. Again, the numerical error of the SIOSHR method is slightly smaller than the error of intersections.

7. Conclusions

In this paper, we have introduced a new swept-region-with-intersections one-step hybrid remapping (SIOSHR) method. We have presented several numerical tests to compare the efficiency and accuracy of the new method with the standard intersection-based method, and the single-material approach based on swept regions. From the numerical tests, we can make several conclusions.

The new method is linearity preserving and second order accurate in both single- and multi-material tests. The numerical results are comparable with the standard methods, we do not observe any significant differences in their accuracy. When compared with the intersection-based method for multi-material simulations, the SIOSHR method reduces the

computational time significantly, by about one third in our implementation. When compared with the previously described hybrid methods TSHR [5] and COSHR [16], the new SIOSHR method has several advantages:

Simple: All fluxes are treated in the same swept-like manner, no corner fluxes are needed.
Easy to implement: There is no need for a special treatment of buffer fluxes as in the COSHR method nor need to change main program routine as in the TSHR method.
General mesh: The SIOSHR method is directly applicable to general polygonal meshes without any further modification, which may be complicated for the COSHR method.
Symmetry preserving: The SIOSHR method is a one-step method, all nodes are moved during the rezoning process in the same manner, so no symmetry violations appear, which can happen in case of the TSHR method due to different treatment of pure and mixed mesh nodes during the rezoning phase.

The SIOSHR method can be used for any material configurations, and is applicable for remapping in any flux-based ALE code.

Acknowledgements

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 No. DE-AC52-06NA25396. The authors gratefully acknowledge the partial support of the US Department of Energy Office of Science Advanced Scientific Computing Research (ASCR) Program in Applied Mathematics Research and the partial support of the US Department of Energy National Nuclear Security Administration Advanced Simulation and Computing (ASC) Program. The first author was supported by the Czech Ministry of Education Grants MSM 6840770022, MSM 6840770010 and LC528, and the Czech Science Foundation Project P201/10/P086. The authors thank Markus Berndt, Jérôme Breil, Stéphane Galera, and Pierre-Henri Maire for fruitful discussions and constructive comments related to the topic of hybrid remap.

References

[1] H.T. Ahn, M. Shashkov, Multi-material interface reconstruction on generalized polyhedral meshes, Journal of Computational Physics 226 (2) (2007) 2096–2132.
[2] P. Anninos, Kull ALE: I. Unstructured mesh advection, interface capturing, and multiphase 2T RHD with material interfaces, Technical report UCRL-ID-147297-PT-1, Lawrence Livermore National Laboratory, 2002.
[3] P. Anninos, Multiphase advection and radiation diffusion with material interfaces on unstructured meshes, Technical report UCRL-JC-150129, Lawrence Livermore National Laboratory, 2002.
[4] T.J. Barth, Numerical methods for gasdynamic systems on unstructured meshes, in: Proceedings of the International School on Theory and Numerics for Conservation Laws. An introduction to Recent Developments in Theory and Numerics for Conservation Laws, 1997, in: C. Rohde, D. Kroner, M. Ohlberger (Eds.), Lecture Notes in Computational Science and Engineering, Springer, Berlin, 1997, ISBN 3-540-65081-4.
[5] M. Berndt, J. Breil, S. Galera, M. Kucharik, P.-H. Maire, M. Shashkov, Two step hybrid remapping (conservative interpolation) for multimaterial arbitrary Lagrangian–Eulerian methods, Journal of Computational Physics 230 (17) (2010) 6664–6687.
[6] J. Breil, S. Galera, P.-H. Maire, Multi-material ALE computation in inertial confinement fusion code CHIC, Computers & Fluids 46 (1) (2011) 161–167.
[7] Phillip Colella, Multidimensional upwind methods for hyperbolic conservation laws, Journal of Computational Physics 87 (1) (1990) 171–200.
[8] J.K. Dukowicz, J.R. Baumgardner, Incremental remapping as a transport/advection algorithm, Journal of Computational Physics 160 (1) (2000) 318–335.
[9] J.K. Dukowicz, J.W. Kodis, Accurate conservative remapping (rezoning) for arbitrary Lagrangian–Eulerian computations, SIAM Journal of Scientific and Statistical Computing 8 (3) (1987) 305–321.
[10] V. Dyadechko, M. Shashkov, Reconstruction of multi-material interfaces from moment data, Journal of Computational Physics 227 (11) (2008) 5361–5384.
[11] P.E. Farrell, M.D. Piggott, C.C. Pain, G.J. Gorman, C.R. Wilson, Conservative interpolation between unstructured meshes via supermesh construction, Computer Methods in Applied Mechanics and Engineering 198 (33-36) (2009) 2632–2642.
[12] S. Galera, P.-H. Maire, J. Breil, A two-dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction, Journal of Computational Physics 229 (16) (2010) 5755–5787.
[13] R. Garimella, M. Kucharik, M. Shashkov, An efficient linearity and bound preserving conservative interpolation (remapping) on polyhedral meshes, Computers & Fluids 36 (2) (2007) 224–237.
[14] Ph. Hoch, Mesh quality and conservative projection in Lagrangian compressible hydrodynamic, in: Numerical Methods for Multi-material Fluid Flows, Prague. <http://www-troja.fjfi.cvut.cz/multimat07/presentations/tuesday/Hoch.pdf>, 2007.
[15] Ph. Hoch, An arbitrary Lagrangian–Eulerian strategy to solve compressible fluid flows, Technical report, CEA. HAL: hal-00366858. Available at: <http://hal.archives-ouvertes.fr/docs/00/36/68/58/PDF/ale2d.pdf>, 2009.
[16] M. Kucharik, J. Breil, S. Galera, P.-H. Maire, M. Berndt, M. Shashkov, Hybrid remap for multi-material ALE, Computers & Fluids 46 (1) (2011) 293–297.
[17] M. Kucharik, R.V. Garimella, S.P. Schofield, M.J. Shashkov, A comparative study of interface reconstruction methods for multi-material ALE simulations, Journal of Computational Physics 229 (7) (2010) 2432–2452.
[18] M. Kucharik, M. Shashkov, Extension of efficient, swept-integration based conservative remapping method for meshes with changing connectivity, International Journal for Numerical Methods in Fluids 56 (8) (2008) 1359–1365.
[19] M. Kucharik, M. Shashkov. Conservative multi-material remap for staggered discretization, in preparation.
[20] M. Kucharik, M. Shashkov, Flux-based approach for conservative remap of multi-material quantities in 2D arbitrary Lagrangian–Eulerian simulations, in: Jaroslav Fořt, Jiří Fürst, Jan Halama, Raphaèl Herbin, Florence Hubert (Eds.), Finite Volumes for Complex Applications VI Problems and Perspectives, Springer Proceedings in Mathematics, vol. 1, Springer, 2011, pp. 623–631.
[21] M. Kucharik, M. Shashkov, B. Wendroff, An efficient linearity-and-bound-preserving remapping method, Journal of Computational Physics 188 (2) (2003) 462–471.
[22] P.H. Lauritzen, Ch. Erath, R. Mittal, On simplifying 'incremental remap'-based transport schemes, Journal of Computational Physics (2011), doi:10.1016/j.jcp.2011.06.030.
[23] R. Loubere, M. Shashkov, A subcell remapping method on staggered polygonal grids for arbitrary-Lagrangian–Eulerian methods, Journal of Computational Physics 209 (1) (2005) 105–138.

[24] L. Margolin, M. Shashkov, Using a curvilinear grid to construct symmetry-preserving discretizations for Lagrangian gas dynamics, Journal of Computational Physics 149 (2) (1999) 389–417.
[25] L.G. Margolin, M. Shashkov, Second-order sign-preserving conservative interpolation (remapping) on general grids, Journal of Computational Physics 184 (1) (2003) 266–298.
[26] S. Menon, D.P. Schmidt, Conservative interpolation on unstructured polyhedral meshes: an extension of the supermesh approach to cell-centered finite-volume variables, Computer Methods in Applied Mechanics and Engineering 200 (41–44) (2011) 2797–2804.

Journal of Computational Physics 258 (2014) 268–304

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Conservative multi-material remap for staggered multi-material Arbitrary Lagrangian–Eulerian methods

Milan Kucharik [a,*], Mikhail Shashkov [b]

[a] Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Brehova 7, Praha 1, 115 19, Czech Republic
[b] XCP-4 Group, MS-F644, Los Alamos National Laboratory, Los Alamos, NM, 87545, USA

ARTICLE INFO

Article history:
Received 28 March 2013
Received in revised form 17 August 2013
Accepted 26 October 2013
Available online 4 November 2013

Keywords:
Conservative interpolations
Multi-material ALE
Flux-based remap
Intersection-based remap

ABSTRACT

Remapping is one of the essential parts of most multi-material Arbitrary Lagrangian–Eulerian (ALE) methods. In this paper, we present a new remapping approach in the framework of 2D staggered multi-material ALE on logically rectangular meshes. It is based on the computation of the second-order material mass fluxes (using intersections/overlays) to all neighboring cells, including the corner neighbors. Fluid mass is then remapped in a flux form as well as all other fluid quantities (internal energy, pressure). We pay a special attention to the remap of nodal quantities, performed also in a flux form. An optimization-based approach is used for the construction of the nodal mass fluxes. The flux-corrected remap (FCR) approach for flux limiting is employed for the nodal velocity remap, which enforces bound preservation of the remapped constructed velocity field. Several examples of numerical calculations are presented, which demonstrate properties of our remapping method in the context of a full ALE algorithm.

1. Motivation, background and preliminaries

In the numerical simulations of fluid flows, the choice of the computational mesh is crucial. Traditionally, there have been two viewpoints utilizing the Lagrangian or the Eulerian framework, each with its own advantages and disadvantages. In a pioneering paper [25], Hirt et al. developed the formalism for a mesh whose motion can be determined as an independent degree of freedom, and showed that this general framework can be used to combine the best properties of the Lagrangian and Eulerian methods. This class of methods has been termed Arbitrary Lagrangian–Eulerian or ALE. Many authors have described the ALE strategies to optimize accuracy, robustness, or computational efficiency, see for example [10,44,28,53,2,36].

It is common to separate the ALE scheme into three distinct stages. These are: (1) a Lagrangian stage, in which the solution and the computational mesh are updated; (2) a rezoning stage, in which the nodes of the computational mesh are moved to more optimal positions; and (3) a remapping stage, in which the Lagrangian solution is conservatively transferred to the rezoned mesh. The main topic of this paper is a new remapping algorithm for multi-material ALE method in 2D Cartesian geometry.

To better understand what is required from the remapping stage of the multi-material ALE method we will briefly describe the conceptual steps in the Lagrangian stage first.

The Lagrangian stage is based on a staggered discretization [13]. We assume that all materials have the same velocity, but each material has its own density, internal energy and pressure. In the staggered discretization, velocity is defined in the nodes of the mesh. In the single material case, mass, density, internal energy and pressure are defined in the cells of

* Corresponding author.
E-mail addresses: kucharik@newton.fjfi.cvut.cz (M. Kucharik), shashkov@lanl.gov (M. Shashkov).

0021-9991/$ – see front matter
http://dx.doi.org/10.1016/j.jcp.2013.10.050

the mesh. In case of multi-material flows, we will have pure cells which contain only one material, and multi-material cells containing several materials. In either case, every cell has to produce forces to its nodes. In case of a pure cell, this force is computed from pressure of the material in this cell. In case of a multi-material cell, the force is computed using information about all materials in this multi-material cell, using the *common* pressure, which is the result of so-called closure model, which is also responsible for time evolution of material properties in the multi-material cell, [8,12,16,17,57,46,47,56].

The Lagrangian stage starts with updating of the nodal velocities as a result of applying the zonal forces computed from material pressure in pure cells or common pressure in multi-material cells. The updated velocities allow to update volumes of all computational cells. For the pure cells, it allows to update density of the material (because during the Lagrangian stage, mass of the material does not change) and internal energy of the material (using some form of $p\,dV$ work equation). For multi-material cells, the situation is more complicated. First, one needs to decide how to distribute change in the total volume of the multi-material cell between materials (which will allow to update densities for each material). Second, one needs to decide how to update internal energy for each material. And finally, how to define the common pressure, which will be used in the computation of the forces participating in the momentum equation on the next time step. All these questions have to be answered by an appropriate closure model. For the purpose of this paper, it is not important which closure model is used. It is important that all closure models require information about the material properties at the previous time step and therefore, to start a new Lagrangian stage in the framework of multi-material ALE method, we need to produce this information as a result of multi-material remapping stage.

The discretization at the Lagrangian stage (which includes a closure model for multi-material cells) conserves total momentum and total energy as well as mass of each material. Therefore, during the remapping stage we have to conserve these quantities as well. In addition, we also need to conserve volume of each material because remapping does not represent any physical model, it is just a data transfer from one computational mesh to another.

The modern multi-material ALE methods usually use interface reconstruction methods [63,3,1,20,22], which allow to subdivide the multi-material cells into a set of convex polygons containing only a single material – pure material polygons. Let us note that in the case when a multi-material Lagrangian cell is non-convex, it may happen that a particular material can be represented by several pure polygons. Such representation of materials inside the multi-material cell is required for advanced closure models [23,24,7] as well as it is one of the most important prerequisites for accurate remapping of material interfaces.

The most popular interface reconstruction method is the volume-of-fluid (VOF) method [63], which is used in many multi-physics production codes. However, its use for more than 2 materials is not apparent because it requires an explicit specification of material ordering and there is no obvious way of comparing different orderings. The VOF method recovers the pure material polygons from the volume fraction (ratio of material and total cell volume) data in the neighboring cells.

In this paper, we employ the modern moment-of-fluid (MOF) method [20,1] utilizing not only material volume (zeroth moment) information, but also the material centroids (ratio of the first and zeroth moment). There are several main advantages of the MOF method when compared with the VOF approach: it does not require specification of material ordering as the VOF method; it uses only information from the single cell under reconstruction; it is able to resolve interfacial details of the order of one cell size (while VOF methods resolve details of the order of 2–3 cells). For comparison of the VOF and MOF methods in the context of a multi-material ALE code, see [33].

We will assume that (as a result of the Lagrangian step) we know the geometry of the new mesh, nodal velocity, and all material parameters in pure cells as well as in multi-material cells. More specifically, for each multi-material cell we know mass, volume and internal energy of each material (material pressure can be computed from the material equation of state). We also know the common pressure to be used in the momentum equation. If one uses an advanced interface reconstruction method like the Moment-of-Fluid (MOF) method [20], which requires an explicit knowledge of the reference centroid for each material, then the Lagrangian step has to be complemented by some algorithm for advancing the material centroids in time, see for example, [33], where the Lagrangian update of the reference centroids is based on a bilinear map inversion. Let us note that the centroid update is a part of the closure models like the interface-aware sub-scale dynamics model described in [24,7]. Therefore, in the framework of a multi-material ALE method, to start a new Lagrangian step after remapping we need to have exactly the same information but on the rezoned mesh.

If the closure model requires the knowledge of the material topology in the multi-material cell at the Lagrangian step, then the interface reconstruction is required even for pure Lagrangian method. If the closure model at the Lagrangian step does not require interface reconstruction, then the remapping stage will start with it because we will need it for the accurate remapping of the interface from the Lagrangian mesh to the rezoned mesh. That is, we assume that at the beginning of the remapping stage we know, which cells of the Lagrangian mesh are pure and which are multi-material and we know the pure material polygons representing materials in the multi-material cells.

There are several properties required from a remapping method to be practically usable in the context of multi-material ALE algorithm:

- continuity – if the computational mesh does not change, no quantity is supposed to change;
- conservation of multi-material quantities – material mass and volume have to be conserved per material;
- conservation of nodal quantities – nodal mass, momenta, and total energy (which are not aware of particular materials) have to be conserved;
- efficiency – all geometrical calculations are done only once, at the beginning of the remapping process.

Conceptually, the remapping stage can be subdivided in two different parts. The first part is the remapping of zonal material quantities, such as volumes, masses, internal energies and pressures. The second part is remapping of nodal velocities.

There are several different approaches for remapping of zonal quantities, for review, see for example, the seminal paper by Benson [10]. The method of concentrations is one example of a method which does not require an explicit knowledge of the material location in the computational cells [22].

There exist several approaches for approximating the swept-type fluxes in the multi-material case, which try to avoid intersections of Lagrangian and rezoned meshes and improve the efficiency of the remapper. Unfortunately, none of the approaches which we are aware of and which are not using intersections is perfectly consistent. Typically, due to the approximation of the material fluxes from the total approximate swept fluxes, the following problematic situations can occur:

- A particular material flux is too big, bigger than the amount of the material in the cell. This can lead to negative mass of the material after remap, so additional flux corrections are required.
- A particular material flux is too small. Even if a particular material should disappear from a cell (as its location in the cell is fully eliminated from the cell by the motion of one of its edges), it does not disappear and leads to inaccuracies and material fragmentation.
- A particular material is included in the flux even though its location in the cell should not produce any flux. This is the consequence of the material flux approximation.

The approaches suffering from the described problems include, for example, the material flux projection to 1D (splitting in the logical directions) or the approximation of the particular (possibly non-convex) swept region by a convex rectangular or trapezoidal region and its intersection with pure material sub-polygons.

In a series of reports by Anninos [3,6,4,5], a remapping method incorporating the VOF material reconstruction method with an automatic material ordering and intersections of the pure material polygons with the transfer volumes in the face-normal direction is described. All these methods imitate the intersections and only work well in certain situations – typically for layered material structures, while they behave poorly for general configurations (such as T-junctions).

We strongly believe that for an accurate remapping of zonal material quantities, one needs to use intersection (overlay) based remap, and this is what we do in this paper. We consider a rezoning strategy in which the mesh does not change its connectivity and the rezoned mesh is close to the Lagrangian mesh such that the rezoned cell is completely contained in the union of its Lagrangian prototype and its immediate neighbors through an edge or a vertex. In this paper we use the standard Winslow mesh smoothing algorithm described in [62,31]. The details of the rezoning algorithm are not important for the purpose of this paper. We are also able to run our simulations in the Eulerian mode, in which the rezoned mesh is equal to the initial mesh. The interested reader can find a brief reviews of the rezoning algorithms in [30,18]. For this type of rezone strategy when the mesh does not change its connectivity, the intersection based remapping can be formulated in an equivalent flux form. The fluxes in this formulation are *generalized* fluxes, which are computed by intersection of the cells of the rezoned mesh with the pure cells and the polygons representing pure materials in the multi-material Lagrangian cells. A flux-based approach which is similar in spirit is presented in [38].

As a result of intersections, we know which materials are in each rezoned cell (that is, we can identify which rezoned cells are multi-material) and volume of each material. This is a purely geometric calculation. The geometry of sub-polygons, which are result of the intersections of the rezoned cell with the pure polygons representing materials in the Lagrangian cell, is the basis for the remapping of all other zonal quantities.

As a result of the Lagrangian step, we have the integral averages of the material densities in each pure polygon representing a material in a multi-material cell. To accurately remap material densities, one needs to do some kind of high-order reconstruction of the density function over these polygons. We use a piece-wise linear reconstruction. The slope of the linear function is first estimated using the data from the neighboring cells containing the same material, and then limited to keep the values of the reconstructed function in physically justified bounds.

To conserve mass of each material, we remap mass in a flux form using the described density reconstruction. The computation of the material fluxes involves integration of the reconstructed linear functions over intersections of pure material polygons (representing materials in the Lagrangian cells) with the neighboring cells in the new mesh. These material fluxes are then used in one of possible methods for the remapping of internal energy and for the remapping of common pressure. These fluxes are also used in the constrained-optimization algorithm to determine the inter-nodal mass fluxes needed for remapping of nodal velocity.

The velocities and momenta are defined at the nodes of the mesh, which requires the definition of nodal mass. The nodal mass *does not see* the material, that is, it is just some mass which is assigned to the node from the surrounding cells. At the initial time moment, nodal mass can be defined using the notion of sub-zonal masses [13]. The sub-zonal masses also do not see materials and are obtained from the total mass of all materials in the cell by its distribution to the nodes of this cell. The nodal mass is a Lagrangian object and does not change during the Lagrangian phase. However, it has to change at the remapping stage. To make its change consistent with the change of the zonal mass, we use the distribution approach described in [50,51,49]. In this approach, the total mass in a particular cell of the rezoned mesh after remapping is distributed to the nodes (vertices) of this cell. That is, to define the rezoned nodal mass we do not need to compute any inter-nodal fluxes. However, to conservatively remap the nodal momentum and compute the remapped

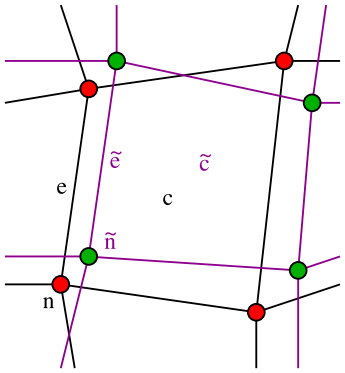


Fig. 1. Computational cell c, node n and edge e in the old mesh $\{c\}$, and the same entities in the new mesh $\{\tilde{c}\}$.

nodal velocity (as a ratio of the remapped nodal momentum and the remapped nodal mass), we need to compute the inter-nodal mass fluxes, which are consistent with the definition of the remapped sub-zonal masses. That is, the usual logic is reversed here and we are using a local constrained-optimization-based approach introduced in [50,51,49] for finding the sub-zonal mass fluxes in each zone, which correspond to the predetermined remapped sub-zonal masses. We extend the original approach by including the *diagonal* mass fluxes, which correspond to the connection through the vertices of dual cells surrounding the node. The inter-nodal momentum fluxes are computed as a product of the inter-nodal mass fluxes and some velocity. The monotonicity of the remapped velocity field is enforced by using the flux corrected remapping (FCR) algorithm, which combines the low-order and high-order momentum fluxes. The FCR algorithm is based on the ideas of flux corrected transport [35]. In the FCR algorithm, the low-order momentum fluxes are computed using the piece-wise constant reconstruction of velocity on the dual mesh and the high-order fluxes are computed using a bilinear reconstruction of velocity in the cell.

To the best of our knowledge, this is the first paper where a complete remapping algorithm for multi-material ALE in the staggered discretization is presented in all details.

This paper is organized as follows. In Section 2, we introduce notation used in this paper for geometric mesh quantities, material polygons, fluid hydrodynamic quantities and their fluxes. In Section 3, the remap of the complete set of cell- and material-centered geometric and fluid quantities is described. The employment of the Moment-of-Fluid (MOF) material reconstruction method [20,1] in the remapping algorithm is described in Section 3.1. In Section 3.2, we explain the geometric background of the method and describe the construction of the multi-material exchange integrals. The remap of the material quantities (volume fractions and centroids) is also discussed. The piece-wise-linear reconstruction of a multi-material quantity is described in Section 3.3. The construction of multi-material mass fluxes and the remap of mass is presented in Section 3.4. The remap of the material internal energy is discussed in Section 3.5. In Section 3.6, we focus on the remap of the common cell pressure. The remap of nodal quantities is described in Section 4. In Section 4.1, we discuss the new nodal remapping method based on the distribution of the inter-cell mass fluxes to the inter-nodal fluxes, which are used for the remap of nodal mass. The construction of the inter-nodal momentum fluxes and the remap of nodal velocity in a flux form is described in Section 4.2. The low- and high-order velocity fluxes are discussed in Sections 4.3 and 4.4, the combination of both fluxes via the FCR methodology is described in Section 4.5. The energy fix is described in Section 5, including the remap of nodal kinetic energy, which guarantees total energy conservation. In Section 6, the entire flux-based multi-material remapper is summarized. Finally, in Section 7 the behavior of the remapping method is demonstrated on a suite of selected numerical examples. The whole paper is concluded in Section 8.

2. Notations

In this section, we summarize the notation used in this paper. It is split in two parts related to the geometric quantities and the quantities related to multi-material hydrodynamics.

2.1. Geometry

In this paper, we assume two computational meshes – the Lagrangian (original or old) mesh $\{c\}$ consisting of cells c, and the rezoned (new or smoothed) mesh $\{\tilde{c}\}$ consisting of cells $\tilde{c}$. For each cell c, its volume is denoted by V_c and centroid by $[x_c, y_c]$. In general, the tilde accent denotes the entities related to the new mesh, as shown in Fig. 1. Besides cells, a mesh contains also nodes n (or $\tilde{n}$) and edges e (or $\tilde{e}$).

In this paper, we assume that the original and rezoned meshes have the same connectivity (we mostly talk about logically-rectangular meshes, but all methods are applicable for other mesh topologies also). A set of cells surrounding a particular cell c is denoted by the symbol $C'(c)$. If this set contains c, we denote it by $C(c)$, so $C(c) = C'(c) \cup c$. The identical

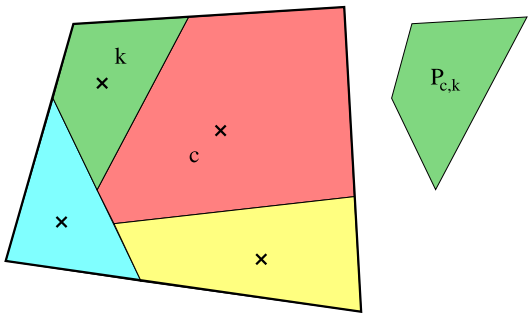


Fig. 2. Material reconstruction in cell c (thick black line), different materials shown by different colors, material reference centroids shown as black crosses. Polygon of green material k is denoted by the symbol $P_{c,k}$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

connectivity implies that the neighborhood around a particular cell is identical, $C(\tilde{c}) = \widetilde{C(c)}$. In a similar logic, the set of nodes around a cell is denoted by $N(c)$, the set of edges connected to node n is denoted by $E(n)$, etc.

In logically-rectangular meshes, we often use explicit integer indices for nodes, $n = [i, j]$, and half-integers for cells, $c = [i + 1/2, j + 1/2]$. For quantities located on the cell edges (inter-cell fluxes, for example), we use indices with one integer and one half-integer components, for example $e = [i + 1/2, j]$. When suitable, we use arrows above the particular quantity showing the direction of the flux in the logical orientation, see for example Section 3.2.

This paper discusses the remapping algorithm for a multi-material computational mesh, in which each cell can contain one or more different materials. We focus here on the situation where materials in a multi-material cell can be unambiguously located in the cell, no mixing of materials is assumed. Each material in cell c is denoted by symbol k, its amount is defined by its material volume $V_{c,k}$ and its centroid by $[x_{c,k}, y_{c,k}]$.

The MOF material reconstruction method is used for the positioning of each material in the cell. The relative amount of each material is given by its volume fraction $\alpha_{c,k}$, its approximate position is given by its centroid $[x_{c,k}, y_{c,k}]$. After the material reconstruction is performed, the individual materials are represented by the pure material polygons, as shown in Fig. 2. The union of all polygons in a particular cell covers the entire cell and the polygons do not overlap. For a particular material k in cell c, we denote the corresponding polygon as $P_{c,k}$. Let us note that in the case of non-convex c, $P_{c,k}$ can be composed of several distinct pieces, such as described in [1].

2.2. Hydrodynamic quantities

In the staggered discretization, the kinematic quantities are defined on the mesh nodes. The nodal quantities are not aware of the material distribution in the neighborhood of the node as we want to avoid material reconstruction on the (generally non-convex) dual cells (nodal control volumes). The nodal mass is denoted by the symbol m_n, the nodal momentum in the x, y direction by $\mu_n^{x,y}$, the nodal velocity components by u_n and v_n, and finally the total nodal kinetic energy by K_n.

The fluid thermodynamic quantities are located in the computational cells, and typically, we define these values separately for each material in the mixed cell as well as the common (total, average) value for the entire cell. For example, a sum of all material masses $m_{c,k}$ in cell c is equal to the total cell mass m_c. Similarly, density of material k in cell c is denoted by $\rho_{c,k}$, the average cell density (ratio of m_c and V_c) by ρ_c. In the same manner, we denote material, resp. cell specific internal energy by $\varepsilon_{c,k}$, resp. ε_c. The material pressure in the cell, which is computed from the equation of state is denoted by $p_{c,k}$. The common pressure in the cell, computation of which is a part of the closure model, is denoted by p_c.

3. Remap of zonal quantities for pure and multi-material zones

In this section, we present the algorithm for remapping of cell- and material-centered quantities. First, we briefly summarize the MOF material reconstruction method in the context of multi-material ALE. We describe the procedure for remapping of material volumes and material centroids in a flux form. We overview the procedure for the piece-wise linear reconstruction of multi-material quantities, including limiting in the multi-material cells. Next, remapping of material mass in a flux form is presented. Remap of the material internal energy then utilizes the same flux form. Finally, remap of cell-centered (common) pressure is performed in a similar form.

3.1. Interface reconstruction

In this section, we briefly summarize the use of the MOF [20,1] interface reconstruction technique in the multi-material remapping method.

Let us assume that we have a Lagrangian computational mesh produced by the Lagrangian step. In each computational cell, we expect to know the reference volume fraction (relative amount) of each material contained in the cell. This informa-

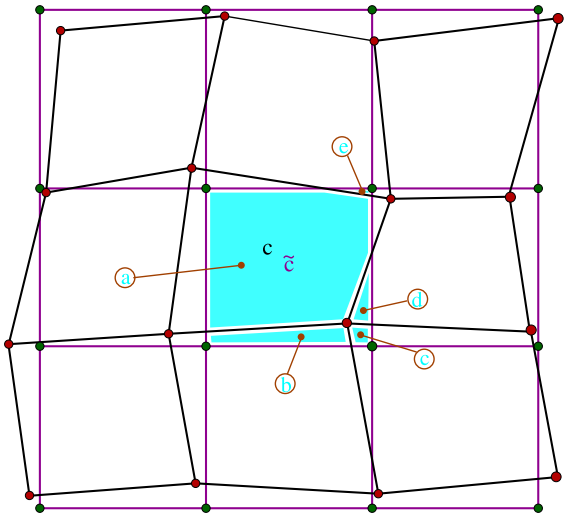


Fig. 3. Intersections (cyan regions) of new cell $\tilde{c}$ with all cells from the patch $C(c)$ around the old cell c. Lagrangian (black) and rezoned (magenta) meshes shown. Required intersections are marked by different labels (a–e). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

tion results from a particular multi-material closure model [26,56]. For the MOF material interface reconstruction method, additional information is needed – the reference centroid of each material. This information can be also a product of the closure model [23,24,7], or it can result from the assumption of a constant centroid position in the parametric space [33]. The MOF method is used to perform the material reconstruction by matching the reference volume fractions exactly and trying to match the reference material centroids as close as possible. The result of this process is the construction of the pure-material polygons $P_{c,k}$ in each multi-material cell, which have their own volumes and actual centroids computed from their geometry.

3.2. Remapping of material volumes and centroids

Let us start with repeating the well known results for pure material cells. A new cell can be composed from the parts of the original cells in the local neighborhood as

$$\tilde{c} = \bigcup_{c' \in C(c)} \tilde{c} \cap c'. \tag{1}$$

This situation is shown in Fig. 3. In the case of the same connectivity meshes, this expression can be rewritten as

$$\tilde{c} = c \cup \left(\bigcup_{c' \in C'(c)} \tilde{c} \cap c' \right) \setminus \left(\bigcup_{c' \in C'(c)} c \cap \tilde{c}' \right). \tag{2}$$

Using this representation, expression for the volume of the rezoned cell can be rewritten in the flux form

$$V_{\tilde{c}} = V_c + \sum_{c' \in C'(c)} F^V_{c,c'}, \tag{3}$$

where the volume fluxes are defined in the form

$$F^V_{c,c'} = \int_{\tilde{c} \cap c'} 1 \, dx dy - \int_{\tilde{c}' \cap c} 1 \, dx dy. \tag{4}$$

In this formula, the first integral represents the outward part, and the second integrals represents the inward part of the volume flux. This situation is shown in Fig. 4. Let us note that the corner fluxes are also considered (for example, flux (g) in Fig. 4). As we can see, only one part of the flux can exist (for example, for left edge flux (h)), both components can exist (for example, right edge fluxes (c) and (e)), or both components of the flux are zero (for example, lower-left corner flux).

Because we are interested in a multi-material remapper, we can write the remapping formula for the volume of the material k in the form

$$V_{\tilde{c},k} = V_{c,k} + \sum_{c' \in C'(c)} F^V_{c,c',k}, \tag{5}$$

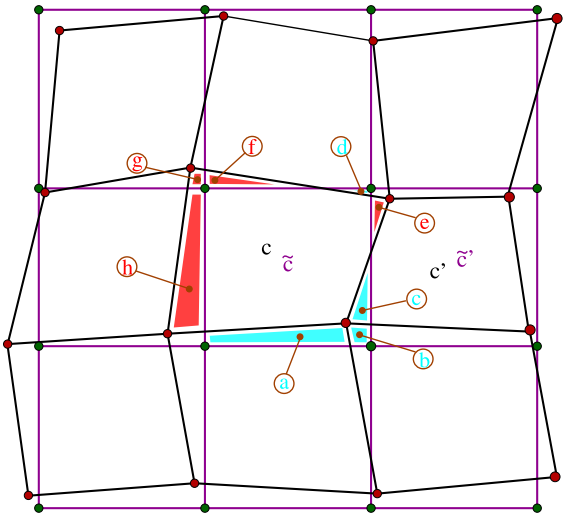


Fig. 4. Outward (cyan) and inward (red) fluxes around old cell c and new cell $\tilde{c}$, including corner fluxes. Lagrangian (black) and rezoned (magenta) meshes shown. Required intersections are marked by different labels (a–h). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

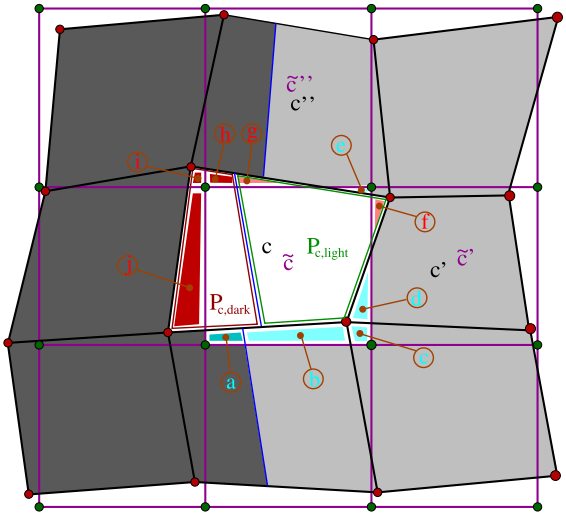


Fig. 5. Outward (cyan) and inward (red) fluxes of dark (left) and light (right) material around old cell c and new cell $\tilde{c}$, including corner fluxes. Materials are separated by material interfaces (blue lines) in each cell. Lagrangian (black) and rezoned (magenta) meshes shown. The particular pure material polygons $P_{c,k}$ are shown by the green and red lines. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

where $V_{c,k}$ represents the volume of material k in cell c. The material volume fluxes are obtained by intersecting the rezoned cells with the pure material sub-polygons instead of the entire Lagrangian cells,

$$F^V_{c,c',k} = \int\limits_{\tilde{c}\cap P_{c',k}} 1\,dx\,dy - \int\limits_{\tilde{c}'\cap P_{c,k}} 1\,dx\,dy. \tag{6}$$

The material volume fluxes are shown in Fig. 5. The patch of 3×3 cells includes two materials, separated by an interface represented by the blue lines. The dark material occupies the left column of cells and the left part of the central column, the light material occupies the right column of cells and the right part of the central cells. We can also see one particular pure material polygon $P_{c,k}$, in this case for k corresponding to the light material, which is obtained by the MOF material reconstruction method. The outward fluxes of cell c are shown as the cyan regions, the inward fluxes are shown as the red regions. As we can see, there can be only single material fluxes, which is the case for example for the neighboring cell c'. Fluxes between c and c'' include both materials. A similar situation of multi-material fluxes can appear also for the corner fluxes.

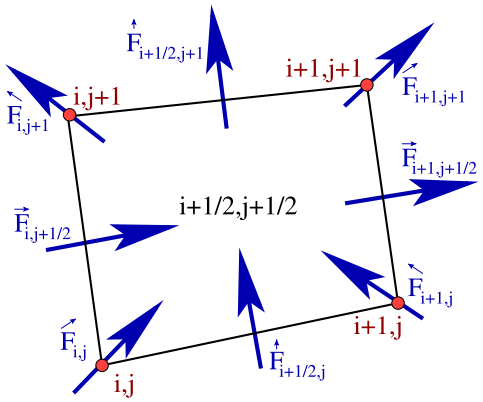


Fig. 6. Enumeration of inter-cell fluxes in cell $[i+1/2, j+1/2]$.

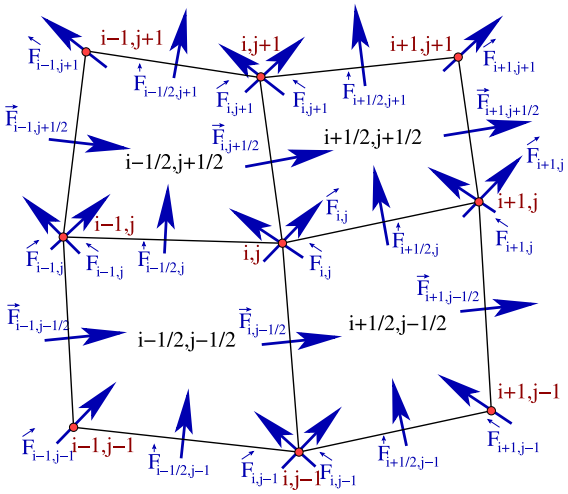


Fig. 7. Enumeration of inter-cell fluxes in a patch of four cells around node $[i, j]$.

For a logically rectangular mesh, Eq. (5) reads as follows

$$V_{[i+1/2,\widetilde{j+1/2}],k} = V_{[i+1/2,j+1/2],k} + \vec{F}^V_{i+1,j+1/2,k} - \vec{F}^V_{i,j+1/2,k} + \nearrow F^V_{i+1,j+1,k} - \nearrow F^V_{i,j,k} + \uparrow F^V_{i+1/2,j,k} - \uparrow F^V_{i+1/2,j,k} + \nwarrow F^V_{i,j+1,k} - \nwarrow F^V_{i+1,j,k}, \tag{7}$$

where, for example,

$$\vec{F}^V_{i,j+1/2,k} = I^1_{[i-1/2,\widetilde{j+1/2}]\cap P_{[i+1/2,j+1/2],k}} - I^1_{P_{[i-1/2,j+1/2],k}\cap[i+1/2,\widetilde{j+1/2}]}. \tag{8}$$

The *exchange integral* $I^1_{[i-1/2,\widetilde{j+1/2}]\cap P_{[i+1/2,j+1/2],k}}$ is defined in Appendix A and it represents the integral of 1 over $[i-1/2,\widetilde{j+1/2}]\cap P_{[i+1/2,j+1/2],k}$; another exchange integrals are defined similarly.

The fluxes are enumerated according to the relative location of the flux in the mesh, as shown in Fig. 6. The arrow above the flux shows its (logical) direction because there are several fluxes located in one mesh vertex, as shown in Fig. 7. For example, there are two fluxes located in node $[i, j]$ – $\nearrow F_{i,j}$ between the cells $[i-1/2, j-1/2]$ and $[i+1/2, j+1/2]$, and $\nwarrow F_{i,j}$ between the cells $[i-1/2, j+1/2]$ and $[i+1/2, j-1/2]$. In the generic notation used before (cell-to-cell fluxes), we skip the arrow indicating the flux direction, because it is uniquely defined by the cell pair, for example $\nearrow F_{i,j} = F_{[i-1/2,j-1/2],[i+1/2,j+1/2]}$.

From the remapped new material volumes $V_{\tilde{c},k}$ and the geometry-updated cell volumes $V_{\tilde{c}}$, the new material volume fractions can be computed in the standard way,

$$\alpha_{\tilde{c},k} = V_{\tilde{c},k}/V_{\tilde{c}}. \tag{9}$$

From the new volume fractions, the presence of the particular material in the new cell and its status can be determined. Let us also note that this is the only part of a typical multi-material ALE code, where the material status in the cells can change. Typically, some tolerance ϵ is used for checking whether a material is present in the cell or not. If $\alpha_{\tilde{c},k} > \epsilon$, the material is present, otherwise it is eliminated from the cell and new volume fraction is set to zero. For practical calculations, the value of $\epsilon = 10^{-8}$ was used. This cleaning process is important because it prevents the numerical artifacts of the order of the machine accuracy from being understood as the real material pieces. The eliminated material is simply erased in our code and we did not observe any problems, however, this can in principal cause a complete material deletion from the entire domain in small amounts. It is therefore often recommended to redistribute the eliminated material volume, mass, and internal energy into the same-material polygons of the neighboring cells.

Similarly to the material volumes, we can define the reference centroids needed for the MOF interface reconstruction. They are defined as the ratio of the first and zeroth moment,

$$x_{c,k} = \frac{M^x_{c,k}}{M^1_{c,k}}, \qquad y_{c,k} = \frac{M^y_{c,k}}{M^1_{c,k}}. \tag{10}$$

These moments can be expressed in the integral form,

$$M^1_{c,k} = I^1_{P_{c,k}}, \qquad M^x_{c,k} = I^x_{P_{c,k}}, \qquad M^y_{c,k} = I^y_{P_{c,k}}, \tag{11}$$

where I^x and I^y are defined in Appendix A and represent the integrals of x and y over the corresponding polygons, and therefore remapped in the same flux form. For example, the x component can be remapped as

$$M^x_{\widetilde{[i+1/2,j+1/2],k}} = M^x_{[i+1/2,j+1/2],k} + \vec{F}^{M^x}_{i+1,j+1/2,k} - \vec{F}^{M^x}_{i,j+1/2,k} + \vec{F}^{M^x}_{i+1,j+1,k} - \vec{F}^{M^x}_{i,j,k} + \vec{F}^{M^x}_{i+1/2,j+1,k} - \vec{F}^{M^x}_{i+1/2,j,k} + \vec{F}^{M^x}_{i,j+1,k} - \vec{F}^{M^x}_{i+1,j,k}, \tag{12}$$

where the fluxes are constructed similarly, for example

$$\vec{F}^{M^x}_{i,j+1/2,k} = I^x_{\widetilde{[i-1/2,j+1/2]}\cap P_{[i+1/2,j+1/2],k}} - I^x_{P_{[i-1/2,j+1/2],k}\cap\widetilde{[i+1/2,j+1/2]}}. \tag{13}$$

The y coordinates of the material centroids are treated in the same manner. To summarize, we have used the pre-computed geometric exchange integrals for constructing the fluxes of volume and centroid, and remapped these quantities in a flux form to the new computational mesh.

The employment of a material reconstruction for the remap of material quantities is demonstrated in Fig. 8. The MOF method was performed on the Lagrangian computational mesh and the obtained pure material polygons are shown in images (a) and (b). During the remapping process, the reference material volume fractions and the positions of material centroids in each multi-material cell of the rezoned mesh have been produced using (7) and (12), this process is illustrated in images (c) and (d). The result of the interface reconstruction using the reference volume fractions and centroids on the rezoned mesh is shown in images (e) and (f). As we can see, the overall shape of the material interface is well preserved, no artificial interface deformations appear.

3.3. Multi-material quantity reconstruction

In this section, we describe how the piece-wise linear reconstruction of a discrete density function is performed in each material. This reconstruction is used for the construction of the material mass fluxes. Analogous approach is used to construct the fluxes of the remaining quantities.

The piece-wise linear reconstruction is performed in the standard form

$$\rho_{c,k}(x,y) = \rho_{c,k} + S^x_{c,k}(x - x_{c,k}) + S^y_{c,k}(y - y_{c,k}), \tag{14}$$

where the material density mean value $\rho_{c,k}$ and the material centroid $[x_{c,k}, y_{c,k}]$ are known, and slopes $S^{\{x,y\}}_{c,k}$ are to be computed. We use the approach described in [45], which we perform on a material-by-material basis.

At first, let us construct the following functional

$$\mathcal{F}\left(S^x_{c,k}, S^y_{c,k}\right) = \sum_{c'\in C'(c)} \left(\rho_{c',k} - \frac{\int_{P_{c',k}} \rho_{c,k}(x,y)\,dx\,dy}{V_{c',k}}\right)^2 \tag{15}$$

measuring the difference (in the least square sense) of the reconstructed function for a particular material from the mean values of the same-material polygons in the neighboring cells. In the case that the neighboring cell c' does not contain material k, c' is not included in the functional. Therefore, there is no problem with division by zero material volume in (15).

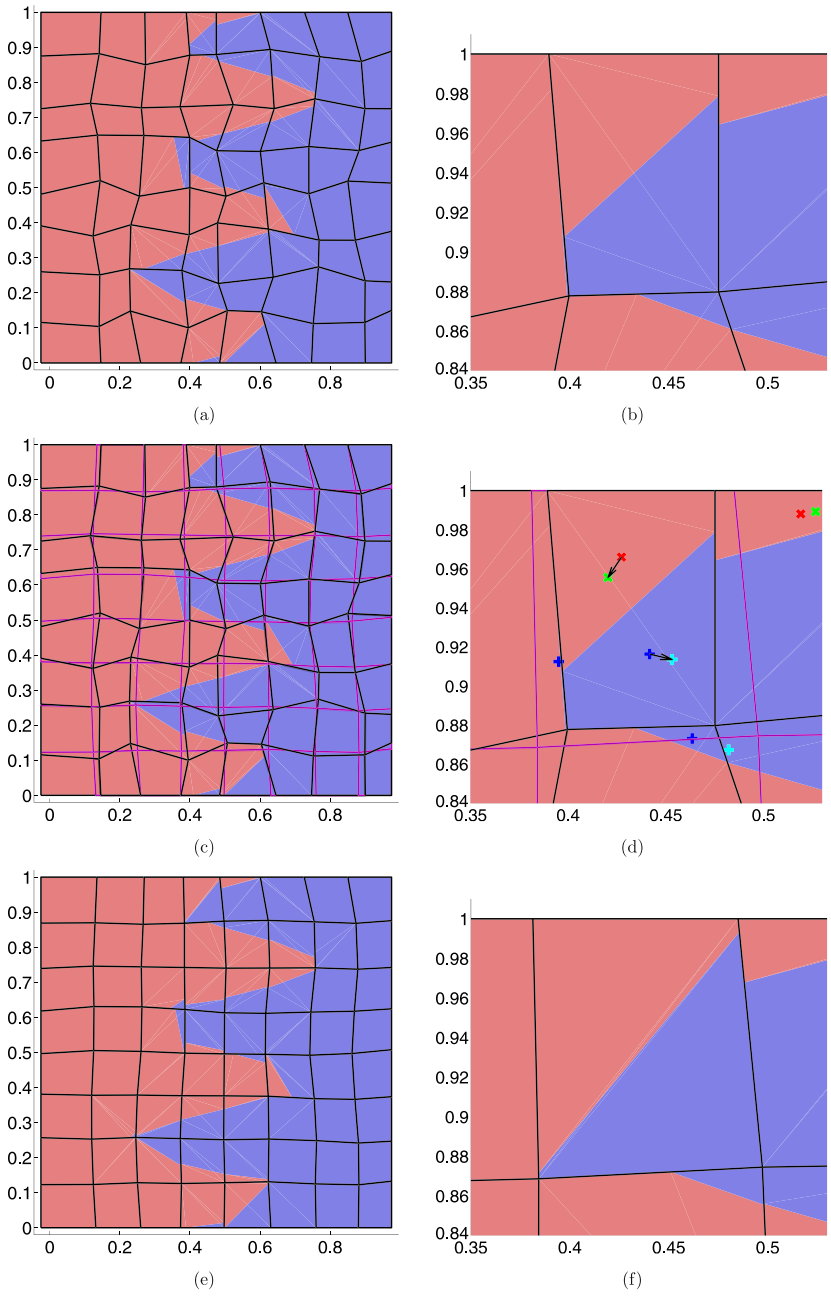


Fig. 8. Material distribution during one remapping step in the entire mesh (left column) and one particular cell (right column). Images (a) and (b) show the material distribution at the Lagrangian computational mesh. In images (c) and (d), both Lagrangian (black) and rezoned (magenta) computational meshes are shown. The material centroids (crosses) move during the remapping step as shown by the black arrows. The result of interface remapping in the rezoned cells is shown in images (e) and (f). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

The conditions for minima are obtained analytically by differentiating the functional with respect to $S^x_{c,k}$ and $S^y_{c,k}$, and requiring these derivatives to be equal to zero. We obtain a simple linear system

$$A \begin{pmatrix} S^x_{c,k} \\ S^y_{c,k} \end{pmatrix} = \begin{pmatrix} a_{xx} & a_{xy} \\ a_{xy} & a_{yy} \end{pmatrix} \begin{pmatrix} S^x_{c,k} \\ S^y_{c,k} \end{pmatrix} = \begin{pmatrix} b_x \\ b_y \end{pmatrix} \tag{16}$$

for unknown slopes, where

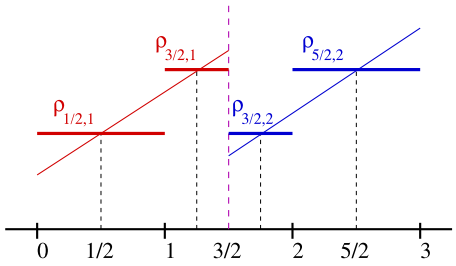


Fig. 9. 1D reconstruction of multi-material density function in a multi-material cell. Nodes (integer indices) separate the 1D domain into 3 (half-integer) cells, the central cell 3/2 is separated by the dashed magenta interface into two materials – 1 (red) and 2 (blue). Analytic linear material densities are shown by thin red and blue solid lines, material mean values by the thick solid lines. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

$$a_{\alpha\beta} = 2 \sum_{c' \in C'(c)} (\alpha_{c,k} - \alpha_{c',k})(\beta_{c,k} - \beta_{c',k}), \tag{17}$$

$$b_{\alpha} = 2 \sum_{c' \in C'(c)} (\alpha_{c,k} - \alpha_{c',k})(\rho_{c,k} - \rho_{c',k}), \tag{18}$$

where α, β indices stand for the x and y coordinates. This system can be simply solved by matrix inversion.

In the second step, to avoid problems with oscillations and overshoots of the reconstructed function, we perform limiting of these slopes using the Barth–Jespersen limiter [9] in the standard form

$$\Phi_{c,k} = \min_{v \in N(P_{c,k})} \Phi_{v,k}, \tag{19}$$

where the values in the vertices v of the pure sub-polygons $P_{c,k}$ are computed as

$$\Phi_{v,k} = \begin{cases} \min(1, \frac{\rho^{\max}_{c,k} - \rho_{c,k}}{\rho^{\mathrm{UL}}_{c,k}(x_v, y_v) - \rho_{c,k}}) & \text{for } \rho^{\mathrm{UL}}_{c,k}(x_v, y_v) - \rho_{c,k} > 0, \\ \min(1, \frac{\rho^{\min}_{c,k} - \rho_{c,k}}{\rho^{\mathrm{UL}}_{c,k}(x_v, y_v) - \rho_{c,k}}) & \text{for } \rho^{\mathrm{UL}}_{c,k}(x_v, y_v) - \rho_{c,k} < 0, \\ 1 & \text{for } \rho^{\mathrm{UL}}_{c,k}(x_v, y_v) - \rho_{c,k} = 0, \end{cases} \tag{20}$$

and where $\rho^{\mathrm{UL}}_{c,k}(x_v, y_v)$ is the unlimited density reconstruction

$$\rho^{\mathrm{UL}}_{c,k}(x, y) = \rho_{c,k} + S^{x\,\mathrm{UL}}_{c,k}(x - x_{c,k}) + S^{y\,\mathrm{UL}}_{c,k}(y - y_{c,k}) \tag{21}$$

with slopes $S^{x\,\mathrm{UL}}_{c,k}$, $S^{y\,\mathrm{UL}}_{c,k}$ defined by formula (15) evaluated in the particular vertex v. The values of $\rho^{\min}_{c,k}$ and $\rho^{\max}_{c,k}$ are local density extrema over pieces of material k in the neighboring cells and the cell c itself,

$$\rho^{\min}_{c,k} = \min_{c' \in C(c): \alpha_{c',k} > 0} \rho_{c',k}, \tag{22}$$

$$\rho^{\max}_{c,k} = \max_{c' \in C(c): \alpha_{c',k} > 0} \rho_{c',k}. \tag{23}$$

Alternatively, one can use a nodal-based density extrema computed over the materials of cells attached to this node,

$$\rho^{\min}_{v,k} = \min_{c' \in C(v): \alpha_{c',k} > 0} \rho_{c',k}, \tag{24}$$

$$\rho^{\max}_{v,k} = \max_{c' \in C(v): \alpha_{c',k} > 0} \rho_{c',k}, \tag{25}$$

which can be used instead of $\rho^{\min}_{c,k}$ and $\rho^{\max}_{c,k}$ in (20). This approach is supposed to be less diffusive due to the smaller extrema patch (over 2×2 instead of 3×3 cells).

Unfortunately, the limiting process leads to the violation of the linearity preservation condition in the multi-material cell. Let us look at the situation shown in Fig. 9. The unlimited slopes computed by solving (16) will recover the analytic linear functions exactly, but the Barth–Jespersen limiting with extrema defined as in (22) and (23) will lead to zero slopes in the multi-material cell and degeneracy to first order due to linearity violation (the same problems appears when (24) and (25) formulas are used instead). This problem is equivalent to the single-material reconstruction at the domain boundary. In this case, we recommend to set $\rho^{\min}_{c,k} = 0$ and $\rho^{\max}_{c,k} = +\infty$ in the multi-material cells, which prevents the density to become negative, otherwise no limitation will be used. This approach can cause density oscillations to appear at the interfaces (which do not grow significantly, as indicated by the numerical results), but maintains the second order of accuracy.

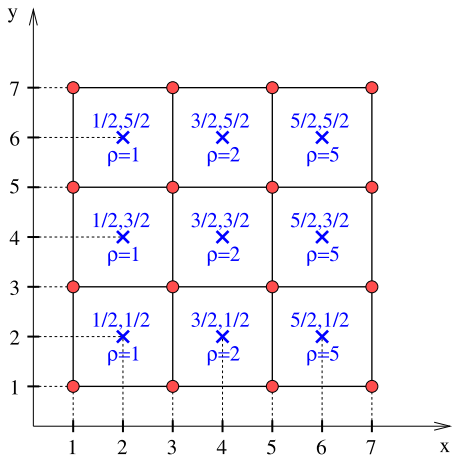


Fig. 10. Density values in an orthogonal, uniform patch of 3×3 cells for 1D symmetry reconstruction test.

The described approach based on the least square minimization of the error functional works well in general, however, it can introduce non-symmetries in 2D simulations of 1D multi-material (and even single-material) problems. Typically, the slopes are reconstructed incorrectly in the layer of boundary and interface cells due to the missing corner-neighboring cells with the same material. If the simulation is expected to preserve the 1D symmetry, we suggest to eliminate the corner neighbors from formula (15) and consequently from (17) and (18), and keep only the edge neighbors. To demonstrate this problem, let us consider the reconstruction process on the data shown in Fig. 10. This problem is clearly one-dimensional, the density function is non-linear, the S^x slopes in cells must be the same in cells $[3/2, 1/2]$, $[3/2, 3/2]$, and $[3/2, 5/2]$, and all S^y slopes must be equal to 0 to keep the 1D symmetry after the remap. Let us apply the reconstruction approach described above to the central cell $[3/2, 3/2]$, the system (16) can then be expressed as

$$\begin{pmatrix} 48 & 0 \\ 0 & 48 \end{pmatrix} \begin{pmatrix} S^{x\,\mathrm{UL}}_{[3/2,3/2]} \\ S^{y\,\mathrm{UL}}_{[3/2,3/2]} \end{pmatrix} = \begin{pmatrix} 48 \\ 0 \end{pmatrix}. \tag{26}$$

Solution of this system represents the correct values of the unlimited slopes $S^{x\,\mathrm{UL}}_{[3/2,3/2]} = 1$, $S^{y\,\mathrm{UL}}_{[3/2,3/2]} = 0$. On the other hand, when the same approach is used in the boundary cell $[3/2, 1/2]$, we obtain the system

$$\begin{pmatrix} 48 & 0 \\ 0 & 120 \end{pmatrix} \begin{pmatrix} S^{x\,\mathrm{UL}}_{[3/2,1/2]} \\ S^{y\,\mathrm{UL}}_{[3/2,1/2]} \end{pmatrix} = \begin{pmatrix} 48 \\ 24 \end{pmatrix} \tag{27}$$

resulting in the values $S^{x\,\mathrm{UL}}_{[3/2,1/2]} = 1$, $S^{y\,\mathrm{UL}}_{[3/2,1/2]} = 1/5$. Thus, the presented reconstruction approach leads to introducing 2D features in a 1D problem along the domain boundary, violating its 1D nature. When only edge neighbors are used in formulas (17) and (18), we obtain the systems in the form

$$\begin{pmatrix} 16 & 0 \\ 0 & 16 \end{pmatrix} \begin{pmatrix} S^{x\,\mathrm{UL}}_{[3/2,3/2]} \\ S^{y\,\mathrm{UL}}_{[3/2,3/2]} \end{pmatrix} = \begin{pmatrix} 16 \\ 0 \end{pmatrix}, \tag{28}$$

$$\begin{pmatrix} 16 & 0 \\ 0 & 40 \end{pmatrix} \begin{pmatrix} S^{x\,\mathrm{UL}}_{[3/2,1/2]} \\ S^{y\,\mathrm{UL}}_{[3/2,1/2]} \end{pmatrix} = \begin{pmatrix} 16 \\ 0 \end{pmatrix}, \tag{29}$$

leading to the correct reconstructions $S^{x\,\mathrm{UL}}_{[3/2,1/2]} = S^{x\,\mathrm{UL}}_{[3/2,3/2]} = 1$ and $S^{y\,\mathrm{UL}}_{[3/2,1/2]} = S^{y\,\mathrm{UL}}_{[3/2,3/2]} = 0$.

3.4. Remap of mass

Logic of the remapping of material masses is similar to the remapping of material volumes:

$$\begin{aligned} m_{\widetilde{[i+1/2,\,j+1/2],k}} = m_{[i+1/2,j+1/2],k} &+ \vec{F}^m_{i+1,j+1/2,k} - \vec{F}^m_{i,j+1/2,k} + \nearrow\!\!\!\!\!F^m_{i+1,j+1,k} - \nearrow\!\!\!\!\!F^m_{i,j,k} \\ &+ \uparrow\!\!\!\!F^m_{i+1/2,j+1,k} - \uparrow\!\!\!\!F^m_{i+1/2,j,k} + \nwarrow\!\!\!\!\!F^m_{i,j+1,k} - \nwarrow\!\!\!\!\!F^m_{i+1,j,k}, \end{aligned} \tag{30}$$

where the fluxes are composed from their inward and outward parts again, for example

$$\vec{F}^m_{i,j+1/2,k} = F^m_{[i-\widetilde{1/2,j+1/2}]\cap P_{[i+1/2,j+1/2],k}} - F^m_{P_{[i-1/2,j+1/2],k}\cap[i+\widetilde{1/2,j+1/2}]}. \tag{31}$$

The parts of the mass fluxes are obtained by integration of the reconstructed density function in the original material polygons, in our example, we have

$$F^m_{[i-\widetilde{1/2,j+1/2}]\cap P_{[i+1/2,j+1/2],k}} = \int_{[i-\widetilde{1/2,j+1/2}]\cap P_{[i+1/2,j+1/2],k}} \rho_{[i+1/2,j+1/2],k}(x,y)\,dx\,dy, \tag{32}$$

$$F^m_{P_{[i-1/2,j+1/2],k}\cap[i+\widetilde{1/2,j+1/2}]} = \int_{P_{[i-1/2,j+1/2],k}\cap[i+\widetilde{1/2,j-1/2}]} \rho_{[i-1/2,j+1/2],k}(x,y)\,dx\,dy. \tag{33}$$

The mass fluxes consist of their inward and outward part again, computed as the integral of the reconstructed density function over the appropriate intersection polygon. Let us evaluate this integral for one of the fluxes,

$$\begin{aligned} F^m_{P_{c,k}\cap\tilde{c}'} &= \int_{P_{c,k}\cap\tilde{c}'} \rho_{c,k}(x,y)\,dx\,dy = \int_{P_{c,k}\cap\tilde{c}'} \left(\rho_{c,k} + S^x_{c,k}(x-x_{c,k}) + S^y_{c,k}(y-y_{c,k})\right) \\ &= \left(\rho_{c,k} - S^x_{c,k}x_{c,k} - S^y_{c,k}y_{c,k}\right) I^1_{P_{c,k}\cap\tilde{c}'} + S^x_{c,k} I^x_{P_{c,k}\cap\tilde{c}'} + S^y_{c,k} I^y_{P_{c,k}\cap\tilde{c}'}. \end{aligned} \tag{34}$$

This way, we have composed the inward part of the material mass flux from the pre-computed geometric exchange integrals. Analogous formula can be derived for the outward part of the flux and for the other fluxes, just by replacing the lower indices of the density mean values, slopes, and integrals to the appropriate polygons. Let us also note that this process is used also for the computation of the material mass fluxes in the corner direction, where only one part of the flux (inward of outward) can be non-zero, in contrast to the side fluxes where both parts can be non-zero due to self-intersecting swept regions. These mass fluxes are finally used for the remap of the material masses (30).

3.5. Remap of internal energy

In this section, we describe the remapping of specific internal energy, also performed in a flux form. We remap the internal energy in the flux form

$$E_{\tilde{c},k} = m_{\tilde{c},k}\varepsilon_{\tilde{c},k} = m_{c,k}\varepsilon_{c,k} + \sum_{c'\in C'(c)} F^{\varepsilon}_{c,c',k}, \tag{35}$$

analogous with formula (5) for the remap of material volume (and material mass). The most natural approach for the internal energy remap is based on the simple construction of the internal energy fluxes by multiplication of the reconstructed specific internal energy values by the mass fluxes. However, this approach does not guarantee monotonicity preservation of the remapped specific internal energy field. Therefore, we adopt the approach introduced in a different context by Dukowicz [19].

This approach is based on the evaluation of the material internal energy fluxes obtained by the integration of the density reconstruction multiplied by the specific internal energy reconstruction function. We remap the internal energy in the flux form (35), where

$$F^{\varepsilon}_{c,c',k} = F^{\varepsilon}_{\tilde{c}\cap P_{c',k}} - F^{\varepsilon}_{P_{c,k}\cap\tilde{c}'}, \tag{36}$$

and where

$$F^{\varepsilon}_{\tilde{c}\cap P_{c',k}} = \int_{\tilde{c}\cap P_{c',k}} \rho_{c',k}(x,y)\varepsilon_{c',k}(x,y)\,dx\,dy, \tag{37}$$

$$F^{\varepsilon}_{P_{c,k}\cap\tilde{c}'} = \int_{P_{c,k}\cap\tilde{c}'} \rho_{c,k}(x,y)\varepsilon_{c,k}(x,y)\,dx\,dy. \tag{38}$$

The conservation of the total material internal energy requires that for the reconstructed density and internal energy, we have

$$\int_{P_{c,k}} \rho_{c,k}(x,y)\varepsilon_{c,k}(x,y)\,dx\,dy = m_{c,k}\varepsilon_{c,k}. \tag{39}$$

The density reconstructions $\rho_{c,k}(x,y)$ are expressed in the standard form (14). To satisfy the conservation requirement (39), the piece-wise linear reconstruction of the specific internal energy has to be centered at the center of mass, that is,

$$\varepsilon_{c,k}(x,y) = \varepsilon_{c,k} + S^{x,\varepsilon}_{c,k}\left(x - x^m_{c,k}\right) + S^{y,\varepsilon}_{c,k}\left(y - y^m_{c,k}\right), \tag{40}$$

where $x^m_{c,k}$ and $y^m_{c,k}$ represent the center of mass of material k in cell c, computed as

$$x^m_{c,k} = \frac{\int_{P_{c,k}} \rho_{c,k}(x,y)x\,dx\,dy}{\int_{P_{c,k}} \rho_{c,k}(x,y)\,dx\,dy}, \qquad y^m_{c,k} = \frac{\int_{P_{c,k}} \rho_{c,k}(x,y)y\,dx\,dy}{\int_{P_{c,k}} \rho_{c,k}(x,y)\,dx\,dy}. \tag{41}$$

The remap based on this reconstruction is bound-preserving, see [19].

After substituting (14) for the density reconstruction, we can rewrite the formulas as

$$x^m_{c,k} = \frac{(\rho_{c,k} - S^x_{c,k}x_{c,k} - S^y_{c,k}y_{c,k})I^x_{P_{c,k}} + S^x_{c,k}I^{x^2}_{P_{c,k}} + S^y_{c,k}I^{xy}_{P_{c,k}}}{m_{c,k}}, \tag{42}$$

$$y^m_{c,k} = \frac{(\rho_{c,k} - S^x_{c,k}x_{c,k} - S^y_{c,k}y_{c,k})I^y_{P_{c,k}} + S^x_{c,k}I^{x\,y}_{P_{c,k}} + S^y_{c,k}I^{y^2}_{P_{c,k}}}{m_{c,k}}, \tag{43}$$

so they are constructed from the pre-computed material integrals again. The integrals of the second-order polynomials are computed in the same way as we did for first-order polynomials (see Eqs. (A.1), (A.2), (A.3) in Appendix A), resulting to formulas

$$I^{x^2}_P = \sum_{e\in\partial P} \frac{1}{12}(x_1+x_2)\left(x_1^2+x_2^2\right)(y_2-y_1), \tag{44}$$

$$I^{xy}_P = \sum_{e\in\partial P} \frac{1}{24}\left(x_1^2(3y_1+y_2) + x_2^2(3y_2+y_1) + 2x_1x_2(y_1+y_2)\right)(y_2-y_1), \tag{45}$$

$$I^{y^2}_P = -\sum_{e\in\partial P} \frac{1}{12}(y_1+y_2)\left(y_1^2+y_2^2\right)(x_2-x_1). \tag{46}$$

The coordinates of the center of mass are then used in the internal energy reconstruction formula (40). They are also used during the reconstruction process described in Section 3.3 instead of the material centroids for the construction of the internal energy slopes $S^{x,\varepsilon}_{c,k}$, $S^{y,\varepsilon}_{c,k}$.

Let us now construct the internal energy flux (37), the flux (38) is constructed in the same manner. It can be written in the following form

$$F^{\varepsilon}_{P_{c,k}\cap\tilde{c}'} = C_1 I^1_{P_{c,k}\cap\tilde{c}'} + C_x I^x_{P_{c,k}\cap\tilde{c}'} + C_y I^y_{P_{c,k}\cap\tilde{c}'} + C_{x^2} I^{x^2}_{P_{c,k}\cap\tilde{c}'} + C_{xy} I^{xy}_{P_{c,k}\cap\tilde{c}'} + C_y I^{y^2}_{P_{c,k}\cap\tilde{c}'}, \tag{47}$$

where the expressions in front of each integral can be expressed as

$$C_1 = \left(\rho_{c,k} - S^x_{c,k}x_{c,k} - S^y_{c,k}y_{c,k}\right)\left(\varepsilon_{c,k} - S^{x,\varepsilon}_{c,k}x^m_{c,k} - S^{y,\varepsilon}_{c,k}y^m_{c,k}\right), \tag{48}$$

$$C_x = S^x_{c,k}\left(\varepsilon_{c,k} - S^{x,\varepsilon}_{c,k}x_{c,k} - S^{y,\varepsilon}_{c,k}y^m_{c,k}\right) + S^{x,\varepsilon}_{c,k}\left(\rho_{c,k} - S^y_{c,k}y_{c,k} - S^x_{c,k}x^m_{c,k}\right), \tag{49}$$

$$C_y = S^y_{c,k}\left(\varepsilon_{c,k} - S^{y,\varepsilon}_{c,k}y_{c,k} - S^{x,\varepsilon}_{c,k}x^m_{c,k}\right) + S^{y,\varepsilon}_{c,k}\left(\rho_{c,k} - S^x_{c,k}x_{c,k} - S^y_{c,k}y^m_{c,k}\right), \tag{50}$$

$$C_{x^2} = S^x_{c,k}S^{x,\varepsilon}_{c,k}, \tag{51}$$

$$C_{xy} = S^x_{c,k}S^{y,\varepsilon}_{c,k} + S^y_{c,k}S^{x,\varepsilon}_{c,k}, \tag{52}$$

$$C_{y^2} = S^y_{c,k}S^{y,\varepsilon}_{c,k}. \tag{53}$$

All quantities on the right-hand sides are known, so the constants can be evaluated explicitly. The geometric exchange integrals included in the internal energy flux (47) are also pre-computed, as described in Appendix A, so the internal energy flux can be directly evaluated also.

In Appendix C it is shown, that this approach is self-consistent in the sense of a complete removal of the material energy in the case when the material has been removed from a particular cell. In other words, if a particular material is removed from a cell during remap, its energy vanishes also.

3.6. Remap of pressure

Let us briefly comment on the update of the common cell pressure, needed for the computation of the nodal velocities in the next coming Lagrangian step. Whereas the material pressures can be computed from the material equation of state, it is not clear what to do with the pressure of their mixture. There exist two different approaches – re-initialization and remap of the common pressure.

The re-initialization approaches approximate the common pressure value from the inter-cell closure model. However, this contradicts our requirement that the remapper should not change the value if the computational mesh does not move. This is the main reason why we switched from re-initialization to the pressure remap.

The common pressure has to be remapped in such a way that it does not change if the mesh is not changing. We remap the common cell pressure in a similar form as we did for density,

$$p_{\tilde{c}} = \frac{1}{V_{\tilde{c}}}\left(p_c V_c + \sum_{c' \in C'(c)} F^{pV}_{c,c'}\right), \tag{54}$$

where the fluxes are computed in the form

$$F^{pV}_{c,c'} = I^1_{\tilde{c}\cap c'} p_{c'}(x_{\tilde{c}\cap c'}, y_{\tilde{c}\cap c'}) - I^1_{c\cap\tilde{c}'} p_c(x_{c\cap\tilde{c}'}, y_{c\cap\tilde{c}'}). \tag{55}$$

Let us remind that the pressure is not a conservative quantity, formula (54) is only a way of interpolation of the new common pressure. The volume fluxes have been pre-computed at the beginning of the remapping step, and the pressure reconstruction can be obtained by the piece-wise linear process described in Section 3.3, modified for the pressure. The locations of the reconstruction $[x_{\tilde{c}\cap c'}, y_{\tilde{c}\cap c'}]$ and $[x_{\tilde{c}'\cap c}, y_{\tilde{c}'\cap c}]$ are computed in a similar form, as we did for the material centroids (10). Let us note that for the limited reconstruction, this approach guarantees local bound preservation, and no new pressure oscillations are developed.

4. Remap of nodal quantities

In this section, we describe the method for the remapping of the nodal mass, momenta, and nodal kinetic energy. Due to single nodal velocities (no material velocities are present in our discretization), we deal only with the common nodal quantities. Our approach is based on the construction of the inter-nodal mass fluxes, and remapping all nodal quantities carried by these inter-nodal mass fluxes. The nodal quantities "do not see" the material mass fluxes, they only deal with the final mass fluxes (total for all materials).

4.1. Nodal mass in a flux form

In our approach, the remapped nodal mass $m_{\tilde{n}}$ is defined by first the distribution of the remapped cell masses $m_{\tilde{c}}$ to define sub-zonal masses $m_{\tilde{c},\tilde{n}}$, and then by summing of the sub-zonal masses around a particular nodes. Therefore, the total nodal mass in the entire domain is the same as the total cell mass,

$$\sum_n m_n = \sum_c m_c = \sum_{\tilde{c}} m_{\tilde{c}} = \sum_{\tilde{n}} m_{\tilde{n}}. \tag{56}$$

The middle equality holds due to the conservative remap of the cell mass, the first and the third equality hold due to the definition of the sub-zonal masses, for example for the Lagrangian mesh we have

$$\sum_n m_n = \sum_n \sum_{c\in C(n)} m_{c,n} = \sum_c \sum_{n\in N(c)} m_{c,n} = \sum_c m_c. \tag{57}$$

Initially, the sub-zonal masses are defined during the initialization, and are used for the computation of the nodal masses. All masses (zonal, sub-zonal, and nodal) stay constant during the Lagrangian step, and they are updated only during the remapping step of the ALE algorithm. In previous sections we have already described remapping of zonal mass. The nodal mass update is essentially based on the distribution of these cell masses, written in a flux form.

The nodal mass can be remapped in a flux form

$$m_{\tilde{n}} = m_n + \sum_{n'\in N'(n)} F^m_{n,n'}, \tag{58}$$

or particularly in our quadrilateral mesh

$$\begin{aligned} m_{\widetilde{[i,j]}} = m_{[i,j]} &+ \vec{F}^m_{i+1/2,j-1/2} - \vec{F}^m_{i-1/2,j-1/2} + \vec{F}^m_{i+1/2,j+1/2} - \vec{F}^m_{i-1/2,j+1/2} \\ &+ \nearrow F^m_{i+1/2,j+1/2} - \nearrow F^m_{i-1/2,j-1/2} + \uparrow F^m_{i+1/4,j+1/2} - \uparrow F^m_{i+1/4,j-1/2} \\ &+ \uparrow F^m_{i-1/4,j+1/2} - \uparrow F^m_{i-1/4,j-1/2} + \nwarrow F^m_{i-1/2,j+1/2} - \nwarrow F^m_{i+1/2,j-1/2}. \end{aligned} \tag{59}$$

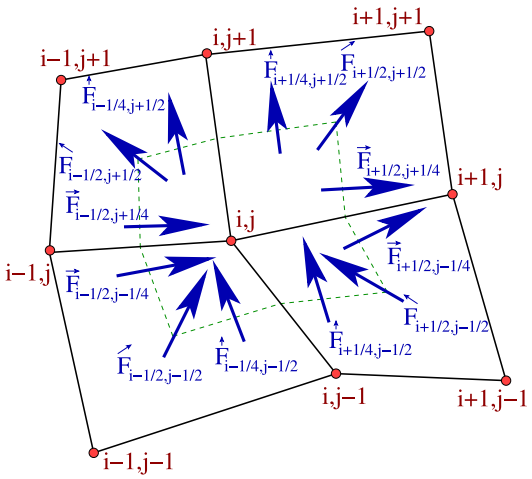


Fig. 11. Notation for marking the inter-nodal fluxes around node $[i, j]$. Only fluxes affecting mass of $[i, j]$ are shown as blue arrows. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

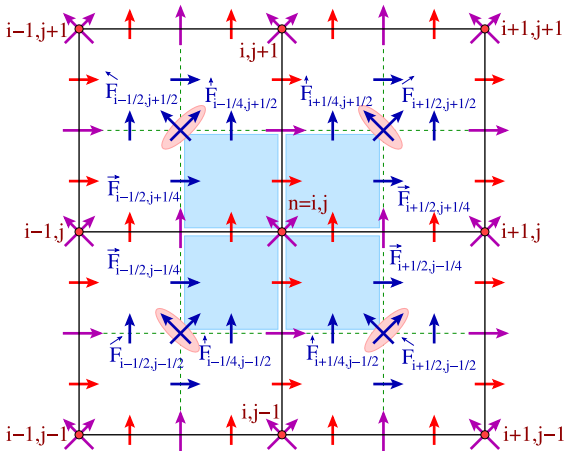


Fig. 12. A complete set of fluxes for a patch around node $n = [i, j]$. Light blue regions represent sub-zones attached to the node n, magenta arrows represent inter-cell fluxes used in (30), red arrows represent inter-cell fluxes distributed equally to half-edges (i.e. inter-corner fluxes), and blue arrows represent inter-nodal fluxes used in (59). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

The situation around one particular node $[i, j]$ is shown in Fig. 11. Only fluxes affecting the nodal mass of $[i, j]$ are shown. The inter-nodal fluxes are enumerated similarly as the inter-cell ones, their index addresses the relative location of the flux in the nodal control volume and the arrow above the flux represents its relative direction. Alternatively, the "generic" notation $F^m_{n,n'}$ can be used representing the total mass flux between nodes n and n'. In this notation, we skip the arrow indicating flux direction, because it is uniquely defined by the node pair. We define the nodal mass as

$$m_n = \sum_{c \in C(n)} m_{c,n}, \tag{60}$$

where $m_{c,n}$ represents the sub-zonal mass of cell c corresponding to node n. We will use the notion of the sub-zonal mass to define mass fluxes participating in Eq. (59). Let us look at the new corner and the new nodal masses, and which fluxes affect them.

The complete set of fluxes, including the fluxes affecting the sub-zonal masses around one particular node, is shown in Fig. 12. For the remapping of the nodal mass we can neglect all red arrows (halves of the inter-cell fluxes), as they only exchange mass between the corners belonging to the same nodal mass. The same reason applies for the diagonal inter-cell fluxes through the central node. We can also neglect all inter-nodal fluxes with pink background, which do not participate in the mass exchange with the highlighted corners. Thus, the new nodal mass only depends on the remaining inter-nodal fluxes, 3 per attached cell shown in Fig. 11.

A possible approach of constructing the fluxes $F^m_{n,n'}$ is based on repeating the process which we described for cells in Appendix B, i.e. intersect the nodal control volumes, compute the inter-nodal geometric exchange integrals, and construct the fluxes of each nodal quantity from them. To do this, we would need to intersect the nodal control volumes (for an example of such control volume, see the dashed region in Fig. 11), which are both generally non-convex (even for convex cells). The intersection of two non-convex polygons requires a significantly more expensive algorithm than our approach for the intersection of a non-convex polygon with a convex polygon described in Appendix B. Another option is to break the non-convex polygon to convex sub-polygons (in this case, sub-zones of the adjacent cells are the most natural choice), and intersect these. This is possible, but in the case of our quadrilateral mesh, we need to intersect $4 \times 4 = 16$ sub-polygons to construct a single inter-nodal flux, so this approach leads to a significantly expensive remapper. For a completely sub-zonal remapper, see [37].

Here, we show two alternative approaches avoiding the expensive intersections of the nodal (or sub-cell) dual volumes. One of the possible options is a cheap interpolation of the inter-nodal mass fluxes from the inter-cell fluxes described in [54]. Unfortunately, by introducing the corner fluxes, positivity of the nodal mass cannot be guaranteed. Therefore, we use a generalization of the approach based on solving a constrained optimization problem introduced in [50,51,49] and adapted it for our remapping framework, as described in the following subsections.

Let us emphasize again that both these approaches couple the cell and nodal masses and the corresponding inter-cell and inter-nodal mass fluxes through the notion of sub-zonal (corner) masses. The way of computing the inter-nodal mass fluxes only defines the inter-corner fluxes inside a particular cell. Therefore, the cell and nodal masses stay consistent and the corresponding cell and nodal densities cannot diverge from each other significantly. Let us also note that in principle the nodal mass remap by formula (59) is not needed as the nodal mass is pre-determined as a sum of the corresponding pre-determined sub-zonal masses (see Section 4.1.2). The construction of the inter-nodal mass fluxes have only one purpose – they participate in the definition of the inter-nodal momentum fluxes described later, and therefore define the remap of nodal velocity. Formula (59) is also needed to prove the DeBar condition [15] (constant velocity must be preserved for an arbitrary density field, which is usually understood as the consistency condition for the velocity remap) and it can be demonstrated that (59) is equivalent to the definition of the new nodal mass as a sum of sub-zonal masses.

4.1.1. Interpolated inter-nodal mass fluxes

A simple option for constructing the inter-nodal mass fluxes is based on the generalization of the method described in [54], which constructs them by averaging of the appropriate total (all material) inter-cell fluxes in the same logical direction. For example, the three fluxes in the upper right cell in Fig. 11 can be computed as

$$\vec{F}^m_{i+1/2,j+1/4} = \frac{1}{4}\left(\vec{F}^m_{i,j+1/2} + \vec{F}^m_{i+1,j+1/2}\right), \tag{61}$$

$$\nearrow F^m_{i+1/2,j+1/2} = \frac{1}{2}\left(\nearrow F^m_{i,j} + \nearrow F^m_{i+1,j+1}\right), \tag{62}$$

$$\uparrow F^m_{i+1/4,j+1/2} = \frac{1}{4}\left(\uparrow F^m_{i+1/2,j} + \uparrow F^m_{i+1/2,j+1}\right). \tag{63}$$

The additional $1/2$ in the horizontal and vertical fluxes comes from the splitting of the inter-cell fluxes into two inter-nodal fluxes, for example, the fluxes $\vec{F}^m_{i,j+1/2}$ and $\vec{F}^m_{i+1,j+1/2}$ are split between the inter-nodal fluxes $\vec{F}^m_{i+1/2,j+1/4}$ and $\vec{F}^m_{i+1/2,j+3/4}$. In all formulas, the inter-cell mass fluxes are the total mass fluxes, for example $\vec{F}^m_{i,j+1/2} = \sum_{\forall k} \vec{F}^m_{i,j+1/2,k}$ is the total contribution of all materials to the final flux.

Let us note that for the original approach without the corner fluxes [54], it can be shown that the remap of the nodal mass will keep the average relation between the nodal and cell masses. For example, let us assume that a particular nodal mass in the previous steps was defined as an average of the adjacent cell masses (which is a very usual technique in many simulation codes), the remap will keep this relation. Because the remapped cell masses are positive and non-zero (due to the properties of the mass remap described in Section 3.4), their average (nodal mass) is also positive and non-zero, which is a very reasonable property of a suitable remapper. However, we cannot show such relation in case when the corner fluxes are present in the remapping formula, so positivity of the new nodal mass cannot be guaranteed. To avoid this problem, an alternative way of defining the inter-nodal mass fluxes is described in the next subsection, which is an extension of [50,51,49] for the presence of the corner fluxes. Still, this simple approach is important as it is used for the constructing the reference mass fluxes in the advanced approach in the next subsection.

4.1.2. Inter-nodal mass fluxes by optimization

The approach which we use is an extension of the method described in [50,51,49]. In these papers, the authors describe the situation when the corner fluxes are not present. This approach is based on several main ingredients:

(1) construction of positive pre-defined target sub-zonal masses, obtained by the distribution of the remapped zonal mass;
(2) construction of the reference internal fluxes; and
(3) minimization of the difference functional measuring the discrepancy of the internal fluxes from the reference ones, under constraints that the flux form of the remapped sub-zonal masses produces the target values.

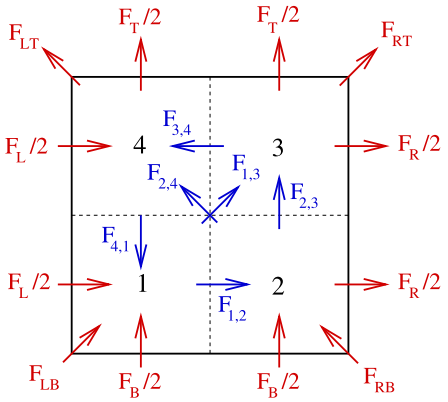


Fig. 13. Situation in one cell. Corners (sub-zones) marked by black numbers, external (inter-cell) fluxes by red arrows and internal (inter-nodal) fluxes by blue arrows. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

This approach is described in all details in [52] and generalized for the $r-z$ geometry in [27]. Let us also note that a technique similar in spirit was presented in [41].

We are looking for the new (target) sub-cell (corner) masses $\widetilde{m}_i$, which must be positive $\widetilde{m}_i > 0$ and sum to the cell mass $\sum_i \widetilde{m}_i = m_{\tilde{c}}$. Here, we used the generic i index for each particular sub-zone of cell c. There exist several options for this definition [50,51,49], we are using the approach which keeps the ratio of the corner mass constant before and after remap, i.e.

$$\widetilde{m}_i = \frac{m_i}{m_c} m_{\tilde{c}}, \tag{64}$$

where the mass of the old cell m_c and the corner m_i are known, and the total mass of the new cell $m_{\tilde{c}}$ was computed during the cell mass remap, so it is known also. This formula is continuous in the sense that the value of the corner mass cannot change dramatically and also it will not change if the mesh has not changed.

This method aims to define the inter-nodal mass fluxes in such a way that the defined target masses expressed in the flux form will be recovered. The situation in a single cell is shown in Fig. 13. It shows all internal (inter-corner) fluxes as blue arrows and all external (inter-cell) fluxes as red arrows, including the corner fluxes. Note that different orientation of the internal fluxes is used to be consistent with [50,51,49]. The external fluxes are known from the remap of the cell mass, the only unknown quantity are the values of the internal fluxes. We require the corner masses to satisfy the following set of constrains (which allows to connect the definition of the sub-zonal and nodal masses in a flux form),

$$\widetilde{m}_1 \equiv m_1 + (F_{1,2} - F_L/2) + (-F_{4,1} - F_B/2) + (F_{1,3} - F_{LB}), \tag{65}$$

$$\widetilde{m}_2 \equiv m_2 + (F_{2,3} - F_B/2) + (-F_{1,2} + F_R/2) + (F_{2,4} - F_{RB}), \tag{66}$$

$$\widetilde{m}_3 \equiv m_3 + (F_{3,4} + F_R/2) + (-F_{2,3} - F_T/2) + (-F_{1,3} + F_{RT}), \tag{67}$$

$$\widetilde{m}_4 \equiv m_4 + (F_{4,1} + F_T/2) + (-F_{3,4} - F_L/2) + (-F_{2,4} + F_{LT}) \tag{68}$$

with respect to notation from Fig. 13. There are no unique internal fluxes satisfying these conditions. For this reason we choose to define these fluxes using a constrained optimization framework and the method of reference fluxes, using the approach of Lagrangian multipliers.

Let us construct a functional

$$\begin{aligned}
&\Phi(F_{1,2}, F_{2,3}, F_{3,4}, F_{4,1}, F_{1,3}, F_{2,4}, \lambda_1, \lambda_2, \lambda_3, \lambda_4) \\
&= \left(F_{1,2} - F^{\text{ref}}_{1,2}\right)^2 + \left(F_{2,3} - F^{\text{ref}}_{2,3}\right)^2 + \left(F_{3,4} - F^{\text{ref}}_{3,4}\right)^2 + \left(F_{4,1} - F^{\text{ref}}_{4,1}\right)^2 + \left(F_{1,3} - F^{\text{ref}}_{1,3}\right)^2 + \left(F_{2,4} - F^{\text{ref}}_{2,4}\right)^2 \\
&\quad + \lambda_1\left(m_1 + (F_{1,2} - F_L/2) + (-F_{4,1} - F_B/2) + (F_{1,3} - F_{LB}) - \widetilde{m}_1\right) \\
&\quad + \lambda_2\left(m_2 + (F_{2,3} - F_B/2) + (-F_{1,2} + F_R/2) + (F_{2,4} - F_{RB}) - \widetilde{m}_2\right) \\
&\quad + \lambda_3\left(m_3 + (F_{3,4} + F_R/2) + (-F_{2,3} - F_T/2) + (-F_{1,3} + F_{RT}) - \widetilde{m}_3\right) \\
&\quad + \lambda_4\left(m_4 + (F_{4,1} + F_T/2) + (-F_{3,4} - F_L/2) + (-F_{2,4} + F_{LT}) - \widetilde{m}_4\right),
\end{aligned} \tag{69}$$

measuring the difference of the internal fluxes from the given reference fluxes, while satisfying the constraints (65)–(68). As the reference fluxes, we can use fluxes (61)–(63) – interpolation from the given inter-cell (external) fluxes described in Section 4.1.1. For the fluxes from Fig. 13, we construct the reference fluxes as

$$F^{\text{ref}}_{1,2} = (F_L/2 + F_R/2)/2 = \frac{1}{4}(F_L + F_R), \tag{70}$$

$$F^{\text{ref}}_{2,3} = (F_B/2 + F_T/2)/2 = \frac{1}{4}(F_B + F_T), \tag{71}$$

$$F^{\text{ref}}_{3,4} = (-F_L/2 - F_R/2)/2 = -\frac{1}{4}(F_L + F_R), \tag{72}$$

$$F^{\text{ref}}_{4,1} = (-F_B/2 - F_T/2)/2 = -\frac{1}{4}(F_B + F_T), \tag{73}$$

$$F^{\text{ref}}_{1,3} = (F_{LB} + F_{RT})/2 = \frac{1}{2}(F_{LB} + F_{RT}), \tag{74}$$

$$F^{\text{ref}}_{2,4} = (F_{RB} + F_{LT})/2 = \frac{1}{2}(F_{RB} + F_{LT}). \tag{75}$$

To minimize the functional (69), we compute its derivatives with respect to all unknowns and require these derivatives to be zero. We obtain two sets of equations. The first set (for the derivatives with respect to the internal fluxes) has the form

$$F^{\text{ref}}_{1,2} - F_{1,2} - \frac{1}{2}\lambda_1 + \frac{1}{2}\lambda_2 = 0 \quad \Longrightarrow \quad F_{1,2} = F^{\text{ref}}_{1,2} + \frac{1}{2}(\lambda_2 - \lambda_1) \tag{76}$$

and similarly for the other fluxes. By differentiating with respect to the Lagrangian multipliers $\boldsymbol{\lambda}$, we recover the second set of equations – the exact equations for constraints (65)–(68). From the first set of Eqs. (76), the internal fluxes can be expressed and substituted into the equations from the second set, which eliminates the internal fluxes from the equations completely. This leads to the final system for the Lagrangian multipliers

$$\mathbb{A}\boldsymbol{\lambda} = \boldsymbol{b}, \tag{77}$$

where the matrix

$$\mathbb{A} = \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & 3 \end{pmatrix}, \tag{78}$$

the vector of unknowns contains the Lagrangian multipliers

$$\boldsymbol{\lambda} = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{pmatrix}, \tag{79}$$

and the right-hand side can be written as

$$\boldsymbol{b} = \begin{pmatrix} 2(m_1 - \tilde{m}_1) + 2F^{\text{ref}}_{1,2} - F_L - 2F^{\text{ref}}_{4,1} - F_B + 2F^{\text{ref}}_{1,3} - 2F_{LB} \\ 2(m_2 - \tilde{m}_2) + 2F^{\text{ref}}_{2,3} - F_B - 2F^{\text{ref}}_{1,2} + F_R + 2F^{\text{ref}}_{2,4} - 2F_{RB} \\ 2(m_3 - \tilde{m}_3) + 2F^{\text{ref}}_{3,4} + F_R - 2F^{\text{ref}}_{2,3} + F_T - 2F^{\text{ref}}_{1,3} + 2F_{RT} \\ 2(m_4 - \tilde{m}_4) + 2F^{\text{ref}}_{4,1} + F_T - 2F^{\text{ref}}_{3,4} - F_L - 2F^{\text{ref}}_{2,4} + 2F_{LT} \end{pmatrix}. \tag{80}$$

As we can see, the right-hand side (80) only contains known quantities – the external fluxes, the reference fluxes (defined in (70)–(75)), the old corner masses, and the new corner masses defined by (64).

It is easy to check, that the system (77) is under-determined, the equations are linearly dependent. However, it can still be solved. One approach is based on eliminating one of the equations, constructing the inverse 3×3 matrix, and direct solving of the reduced system for three lambdas. The fourth lambda is then supplemented from the eliminated equation. As we can see, there is no iterative process involved in the functional minimization and all constraints (65)–(68) are satisfied up to the machine precision.

Let us note, that the solution for $\boldsymbol{\lambda}$ is not unique and a different method (elimination of a different lambda, for example) generally finds a different solution. However, when computing the final internal fluxes using Eq. (76) and similarly for other fluxes, the same values for the internal fluxes are obtained as the null space of the problem has dimension 1, so lambdas obtained by different methods are the same up to a constant. As in the flux formulas (76) the lambdas are subtracted, this constant is eliminated and the fluxes are the same.

Let us also note that for a special case of the initial nodal masses defined by averaging, defining the reference fluxes as (76), and all corner fluxes equal to 0, the simple approach described in Section 4.1.1 is recovered.

4.2. Remapping of momentum in flux form

The conservative remap of momentum is performed in the following flux form

$$\begin{aligned}\mu_{\tilde{n}} = m_n u_n &+ \vec{F}^m_{i+1/2,j-1/4} u^*_{i+1/2,j-1/4} - \vec{F}^m_{i-1/2,j-1/4} u^*_{i-1/2,j-1/4} \\ &+ \vec{F}^m_{i+1/2,j+1/4} u^*_{i+1/2,j+1/4} - \vec{F}^m_{i-1/2,j+1/4} u^*_{i-1/2,j+1/4} \\ &+ \nearrow F^m_{i+1/2,j+1/2} u^*_{i+1/2,j+1/2} - \nearrow F^m_{i-1/2,j-1/2} u^*_{i-1/2,j-1/2} \\ &+ \uparrow F^m_{i+1/4,j+1/2} u^*_{i+1/4,j+1/2} - \uparrow F^m_{i+1/4,j-1/2} u^*_{i+1/4,j-1/2} \\ &+ \uparrow F^m_{i-1/4,j+1/2} u^*_{i-1/4,j+1/2} - \uparrow F^m_{i-1/4,j-1/2} u^*_{i-1/4,j-1/2} \\ &+ \nwarrow F^m_{i-1/2,j+1/2} u^*_{i-1/2,j+1/2} - \nwarrow F^m_{i+1/2,j-1/2} u^*_{i+1/2,j-1/2},\end{aligned} \tag{81}$$

so the momentum flux is obtained by multiplication of the inter-nodal mass fluxes by the particular velocities reconstructed by one of the methods described later. The remapped velocity is then computed as $u_{\tilde{n}} = \mu_{\tilde{n}}/m_{\tilde{n}}$. A similar approach is used for the remap of momentum and velocity in y direction. Let us note here, that treating velocity remap by components can result in violation of radial symmetry of the solution. There exist several approaches for dealing with the problem of monotonicity of vectors, see for example [39,42,43,61]. This issue is discussed in details in [60].

It is obvious, that this approach satisfies the DeBar condition [15,10], 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 a random density field. After an arbitrary mesh movement, the remapping process must reproduce the constant velocity field. Any velocity reconstruction method suggested later will give us $u^* = \bar{u}$ for all flux velocities, so $\bar{u}$ can be factored from the whole right-hand side of (81). The rest of the right-hand side corresponds exactly to the remapped nodal mass definition (59), which cancels with the denominator in the definition of the remapped velocity. Therefore, the described remapping algorithm preserves a constant velocity field and is DeBar-consistent (under the condition that the reconstructed velocity preserves constant function).

The only remaining question is how to reconstruct the velocity field and define the flux velocities u^*. Here, we describe a low-order approach based on a piece-wise constant reconstruction and a high-order method based on a bilinear velocity interpolation. Finally, we describe the inter-nodal flux corrected remap combining the chosen high-order velocities with the low order ones in order to eliminate possible velocity oscillations. All these methods preserve constant velocity fields, and thus do not violate the DeBar consistency.

4.3. Low-order velocity reconstruction

The fist approach for determining the nodal flux velocity is based on a simple piece-wise constant velocity reconstruction. We define the velocity field as a constant equal to the nodal velocity inside the control volume of a particular node. For the flux velocity, we choose the nodal velocity depending on the sign of the corresponding mass flux, for example, for a particular case of fluxes belonging to the upper right cell in Fig. 11,

$$u^*_{i+1/2,j+1/4} = \begin{cases} u_{[i+1,j]} & \text{if } \vec{F}^m_{i+1/2,j+1/4} > 0, \\ u_{[i,j]} & \text{if } \vec{F}^m_{i+1/2,j+1/4} \leqslant 0, \end{cases} \tag{82}$$

$$u^*_{i+1/2,j+1/2} = \begin{cases} u_{[i+1,j+1]} & \text{if } \nearrow F^m_{i+1/2,j+1/2} > 0, \\ u_{[i,j]} & \text{if } \nearrow F^m_{i+1/2,j+1/2} \leqslant 0, \end{cases} \tag{83}$$

$$u^*_{i+1/4,j+1/2} = \begin{cases} u_{[i,j+1]} & \text{if } \uparrow F^m_{i+1/4,j+1/2} > 0, \\ u_{[i,j]} & \text{if } \uparrow F^m_{i+1/4,j+1/2} \leqslant 0. \end{cases} \tag{84}$$

If the particular mass flux is positive, then the edge between the nodal control volumes moves (imaginary, we do not explicitly track its motion) in the direction of the flux arrow into the control volume of the neighboring node. In the case of negative mass flux, it moves toward the original node and stays in its control volume.

This interpolation approach is first-order accurate. To be applicable as the low-order flux for the flux-corrected transport, it needs to satisfy the bound-preservation condition for velocity (i.e., the remapped nodal velocity stays between the local minimum and maximum of nodal velocity in the neighborhood of the particular node),

$$u_n^{\min} \leqslant u_{\tilde{n}} \leqslant u_n^{\max}, \tag{85}$$

where the local extrema are defined as

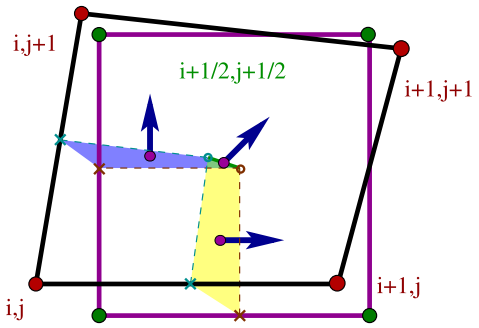


Fig. 14. "Inter-nodal swept regions" in cell $i+1/2$, $j+1/2$ attached to node i, j. Lagrangian (black) and rezoned (magenta) cells are shown, as well as bounds of the Lagrangian (cyan) and rezoned (brown) control volumes. Horizontal and vertical "inter-nodal swept regions" (blue and yellow regions) are bounded by the appropriate edge centers (cyan and brown crosses) and cell centers (cyan and brown circles), corner "inter-nodal swept region" is represented by the green line connecting the cell centers. Centers of the swept regions are represented by the magenta circles, inter-nodal fluxes are represented by the blue arrows. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

$$u_n^{\min} = \min_{n' \in N(n)} u_{n'}, \tag{86}$$

$$u_n^{\max} = \max_{n' \in N(n)} u_{n'}. \tag{87}$$

In [60] it is proved that these conditions hold, the method is bound-preserving, and can be used in the FCR framework.

4.4. High-order flux velocity

The second approach of the nodal flux velocity computation is based on a bilinear interpolation of the velocity field. In the first step, we construct the "inter-nodal swept regions", as demonstrated in Fig. 14. For horizontal and vertical fluxes, we define the swept region as the polygon between the Lagrangian and new control volume edges. For corner fluxes, it is defined at the line segment connecting the Lagrangian and rezoned cell centers. For each swept region, we construct its center $[x_s, y_s]$ by averaging its vertices, as shown in Fig. 14. These centers define the positions in which the velocity value will be taken.

For each cell in the mesh, we have four velocities defined in the cell vertices, and several positions (centers of the "inter-nodal swept regions" for each inter-nodal flux inside this cell) where the velocity should be reconstructed. The most natural approach is the bilinear interpolation. The center of each "inter-nodal swept region" $[x_s, y_s]$ is transformed into the logical ξ, η coordinates by inverting the isoparametric maps [29],

$$x_s = (1-\xi_s)(1-\eta_s)x_{[i,j]} + \xi_s(1-\eta_s)x_{[i+1,j]} + \xi_s\eta_s x_{[i+1,j+1]} + (1-\xi_s)\eta_s x_{[i,j+1]}, \tag{88}$$

$$y_s = (1-\xi_s)(1-\eta_s)y_{[i,j]} + \xi_s(1-\eta_s)y_{[i+1,j]} + \xi_s\eta_s y_{[i+1,j+1]} + (1-\xi_s)\eta_s y_{[i,j+1]}. \tag{89}$$

These equations can be solved analytically for ξ_s and η_s by converting it to a quadratic equation for one of the unknowns. The velocity reconstruction can then be interpolated in a bilinear form as

$$u_s = (1-\xi_s)(1-\eta_s)u_{[i,j]} + \xi_s(1-\eta_s)u_{[i+1,j]} + \xi_s\eta_s u_{[i+1,j+1]} + (1-\xi_s)\eta_s u_{[i,j+1]}, \tag{90}$$

$$v_s = (1-\xi_s)(1-\eta_s)v_{[i,j]} + \xi_s(1-\eta_s)v_{[i+1,j]} + \xi_s\eta_s v_{[i+1,j+1]} + (1-\xi_s)\eta_s v_{[i,j+1]}. \tag{91}$$

This interpolation is second-order accurate. However, when used in the velocity remap, it can violate the local bounds for the components of the velocity field.

4.5. Flux-corrected remap

A high-order velocity interpolation (for example, the bilinear interpolation from Section 4.4) can introduce oscillations in the remapped velocity field. In this section, we describe the standard Flux-Corrected Remap (FCR) mechanism based on the flux-corrected transport (FCT) [64] adapted for the nodal velocity, combining the low- and high-order momenta fluxes to achieve a non-oscillatory velocity field.

Let us assume a combination of outward low and high-order momentum fluxes in the form

$$F^{\mu}_{n,n'} = F^{\mu,L}_{n,n'} + C_{n,n'}\delta F^{\mu}_{n,n'}, \tag{92}$$

where the low- and high-order fluxes are computed as

$$F^{\mu,L}_{n,n'} = F^{m}_{n,n'}u^{*,L}_{n,n'}, \qquad F^{\mu,H}_{n,n'} = F^{m}_{n,n'}u^{*,H}_{n,n'}, \tag{93}$$

and the flux difference is defined as

$$\delta F^{\mu}_{n,n'} = F^{\mu,H}_{n,n'} - F^{\mu,L}_{n,n'}. \tag{94}$$

In the previous formulas, $F^{m}_{n,n'}$ are the inter-nodal mass fluxes used for the nodal mass remapping, $u^{*,L}_{n,n'}$ are the flux velocities obtained by the low-order piece-wise constant reconstruction, and $u^{*,H}_{n,n'}$ are the high-order flux velocities obtained by the bilinear interpolation described in the previous section. Our goal is to find the set of constants $C_{n,n'}$ such that the following constrains are satisfied

$$u^{\min}_{n} \leqslant u_{\tilde{n}} \leqslant u^{\max}_{n}, \tag{95}$$

where the value of $u_{\tilde{n}}$ is obtained by the momentum remap using the momentum fluxes (92), and the local extrema are computed by (86), (87). Let us multiply (86), (87) by the remapped mass, and we obtain

$$\mu^{\min}_{n} \equiv u^{\min}_{n} m_{\tilde{n}} \leqslant \mu_{\tilde{n}} \equiv u_{\tilde{n}} m_{\tilde{n}} \leqslant u^{\max}_{n} m_{\tilde{n}} \equiv \mu^{\max}_{n}. \tag{96}$$

The central term (new momentum) can be written in the flux form

$$\mu^{\min}_{n} \leqslant \mu_{n} + \sum_{n' \in N'(n)} F^{\mu}_{n,n'} \leqslant \mu^{\max}_{n}, \tag{97}$$

and after using (92), we can write

$$\mu^{\min}_{n} \leqslant \mu^{L}_{\tilde{n}} + \sum_{n' \in N'(n)} C_{n,n'} \delta F^{\mu}_{n,n'} \leqslant \mu^{\max}_{n}, \tag{98}$$

where $\mu^{L}_{\tilde{n}}$ is the nodal momentum remapped by the low-order momentum fluxes $F^{\mu,L}_{n,n'}$. We know that the constraints are satisfied if we choose all $C_{n,n'} = 0$. Therefore, the feasible set is not empty. Now, we can use the usual logic of the FCT approach to find the optimal values of $C_{n,n'}$ which still satisfy the constraints but are as close to unity, as possible (it means, that the final flux will be as close to the high-order flux, as possible). After substituting (93) to (92), the final flux velocities are

$$u^{*}_{n,n'} = C_{n,n'} u^{*,H}_{n,n'} + (1 - C_{n,n'}) u^{*,L}_{n,n'}. \tag{99}$$

As both $u^{*,L}_{n,n'}$ and $u^{*,H}_{n,n'}$ reconstructions respect the constant velocity field, the DeBar condition is fulfilled automatically.

The same process can be used for the computation of the flux velocities in the y direction $v^{*}_{n,n'}$. However, treating the velocity components separately leads to the change of the direction of the velocity vector, see [35], which can introduce artificial phenomena in the flow and break the problem symmetries. Therefore, it is recommended to compute a common value $\min(C_{n,n'}, D_{n,n'})$, which is used for the correction of the momentum fluxes in both directions, where $C_{n,n'}$ represents the constant related to the u velocity components as described above, and $D_{n,n'}$ is the corresponding constant for the v velocity component. The corrected flux velocities $u^{*}_{n,n'}$ and $v^{*}_{n,n'}$ are then used for the remap of the nodal momenta. An alternative approach avoiding a possible over-dissipation is described in [40].

The constructed flux velocities can be also used for the construction of the inter-nodal kinetic energy fluxes, which are used in the kinetic energy remap. However, to prevent oscillations and positivity-violation in kinetic energy (affecting the internal energy smoothness through the energy conservation fix described in Section 5), they can be used for the construction of the high-order inter-nodal kinetic energy fluxes, which are combined with the low-order fluxes by an additional FCR. The same FCR process can be used with

$$F^{K,L}_{n,n'} = F^{m}_{n,n'} k^{L}_{n,n'}, \qquad F^{K,H}_{n,n'} = F^{m}_{n,n'} k^{H}_{n,n'}, \tag{100}$$

instead of (93), where the specific kinetic energies are constructed as

$$k^{L}_{n,n'} = \frac{1}{2} \left(\boldsymbol{u}^{*,L}_{n,n'} \right)^2, \qquad k^{H}_{n,n'} = \frac{1}{2} \left(\boldsymbol{u}^{*,H}_{n,n'} \right)^2. \tag{101}$$

The local kinetic energy extrema are computed from the specific kinetic energy

$$k^{\min}_{n} = \min_{n' \in N(n)} k_{n'}, \tag{102}$$

$$k^{\max}_{n} = \max_{n' \in N(n)} k_{n'}, \tag{103}$$

$$k_{n'} = \frac{1}{2} (u_{n'})^2, \tag{104}$$

and the rest of the process is analogous. This technique guarantees bound-preservation for the kinetic energy and thus slightly reduces the amount of kinetic energy transferred to internal by the fix.

5. Total energy conservation

Two different kinetic energies can be defined at the end of the remapping step. First, the actual one, obtained in a standard way from the remapped velocities

$$K_{\tilde{n}} = \frac{1}{2} m_{\tilde{n}} \left(u_{\tilde{n}}^2 + v_{\tilde{n}}^2 \right), \tag{105}$$

and the second one, obtained by the remapping of the kinetic energy in a flux form

$$\bar{K}_{\tilde{n}} = \frac{1}{2} m_n \left(u_n^2 + v_n^2 \right) + \sum_{n' \in N'(n)} F_{n,n'}^K, \tag{106}$$

where the kinetic energy fluxes

$$F_{n,n'}^K = \frac{1}{2} F_{n,n'}^m k_{n,n'}^* \tag{107}$$

are constructed by the FCR approach similar to the one described at the end of Section 4.5. The kinetic energy $K_{\tilde{n}}$ defined from the remapped velocity and mass is the actual kinetic energy participating in the definition of total energy in the next Lagrangian step. This kinetic energy is not conservative during the remapping. The second kinetic energy $\bar{K}_{\tilde{n}}$, on the other hand, is the remapped (and thus conservative) kinetic energy. So, for each node, we can compute the kinetic energy discrepancy

$$\delta K_n = \bar{K}_{\tilde{n}} - K_{\tilde{n}}, \tag{108}$$

expressing the difference of the new kinetic energy from the conservative value. We enforce the energy conservation (which is important to obtain a correct shock speed) by the energy fix (see, for example, in [10]), distributing the kinetic energy discrepancy into the internal energy of the adjacent cells, generalized for the multi-material case. In our approach, the discrepancy is redistributed to the adjacent cells $c \in C(n)$ proportionally to their masses, and to the materials k in the cell proportionally to their volume fractions,

$$\delta K_{n,c,k} = \alpha_{c,k} \frac{m_c}{\sum_{c' \in C(n)} m_{c'}} \delta K_n. \tag{109}$$

The way of distributing the discrepancy into the internal energy is not unique, one can for example use the distribution proportional to the material masses

$$\delta K_{n,c,k} = \frac{m_{c,k}}{\sum_{c' \in C(n)} m_{c'}} \delta K_n, \tag{110}$$

or proportional to the original internal energies

$$\delta K_{n,c,k} = \frac{m_{c,k} \varepsilon_{c,k}}{\sum_{c' \in C(n)} m_{c'} \varepsilon_c} \delta K_n. \tag{111}$$

The internal energy is then modified as

$$E_{\tilde{c},k} = E_{\tilde{c},k} + \sum_{n \in N(c)} \delta K_{n,c,k}, \tag{112}$$

and the final specific internal energy is computed as

$$\varepsilon_{\tilde{c},k} = \frac{E_{\tilde{c},k}}{m_{\tilde{c},k}}. \tag{113}$$

This process distributes all the kinetic energy discrepancy into the internal energies, and thus the total energy conservation is guaranteed.

Let us also note that the described energy discrepancy fix is controversial. On one hand, it enforces the energy conservation and correct shock speeds. On the other hand, it can create disturbances of the originally-smooth internal energy field, and consequently in the pressure field. Especially in the case of a fast (high kinetic energy) and cold (low internal energy) fluid flow, the discrepancy correction can dominate and create severe internal energy disturbances.

For completeness, let us note that there exist different approaches for dealing with the energy fix. One option is constructing the cell-based kinetic energy as

$$K_c = \sum_{n \in N(c)} \frac{1}{2} m_{c,n} \left(u_n^2 + v_n^2 \right), \tag{114}$$

which can be remapped in the same manner as the internal energy. A cell based kinetic energy discrepancy is obtained, which is redistributed into the materials of the particular cell. This way, one ambiguity is avoided (the nodal discrepancy does not have to be redistributed to the surrounding cells).

Another option is giving up the exact total energy conservation and performing the energy fix only in regions where shocks are detected to obtain a correct shock speed. In smooth regions no fix is done to avoid the disturbance of the internal energy field. Finally, one can combine the described strategies in such a way which best suits the needs of the particular simulation. This topic requires more research in the future to understand the advantages of each approach and relations among them.

6. Summary of multi-material remapping algorithm

Here, we present a brief overview of the complete remapper at one place to show its main logic. The remapping stage for multi-material ALE can be summarized as follows.

- Perform the material interface reconstruction on the Lagrangian mesh to obtain the polygons of pure materials $P_{c,k}$ in each cell c (Section 3.1).
- Intersect the pure material polygons in multi-material cells of the Lagrangian mesh with the neighboring cells in the rezoned mesh $\tilde{c}'$ (including corner neighbors), $P_{c,\tilde{c}',k}$. Integrate simple polynomials ($f = 1, x, y$) over these intersections to construct the geometric exchange integrals $I^f_{c,c',k} = \int_{P_{c,\tilde{c}',k}} f\,dV$ (Section 3.2 and Appendix A).
- Remap volumes $V_{\tilde{c},k}$ and reference centroids $[x_{\tilde{c},k}, y_{\tilde{c},k}]$ of material k in the new cell $\tilde{c}$ (see Fig. 8) in a flux form $V_{\tilde{c},k} = V_{c,k} + \sum_{c' \in C'(c)} I^1_{c,c',k}$ and similarly for $x_{\tilde{c},k}$ and $y_{\tilde{c},k}$. From the cell geometry, update the total new cell volumes $V_{\tilde{c}}$ and centroids $[x_{\tilde{c}}, y_{\tilde{c}}]$ (Section 3.2).
- From the new $V_{\tilde{c}}$ and $V_{\tilde{c},k}$, compute the new volume fractions $\alpha_{\tilde{c},k} = V_{\tilde{c},k}/V_{\tilde{c}}$. From the new volume fractions, the status of each new cell can be determined – is it a single- or multi-material cell, which materials are present, etc.
- Reconstruct the material density (Section 3.3), construct the material mass fluxes by composing the integrals of the reconstructed density from the exchange integrals. Remap the material masses $m_{\tilde{c},k}$ using (30) by materials, compute the new total cell mass $m_{\tilde{c}} = \sum_k m_{\tilde{c},k}$. Update the new cell and material densities, $\rho_{\tilde{c}} = m_{\tilde{c}}/V_{\tilde{c}}$, $\rho_{\tilde{c},k} = m_{\tilde{c},k}/V_{\tilde{c},k}$ (Section 3.4).
- Reconstruct the material specific internal energy (Section 3.3), construct the material energy fluxes (either by multiplying the mass fluxes by the reconstructed energy, or by the advanced approach from [19]). Remap the material total internal energy $E_{\tilde{c},k}$, compute the material specific internal energy $\varepsilon_{\tilde{c},k} = E_{\tilde{c},k}/m_{\tilde{c},k}$. Compute the new cell internal energy $E_{\tilde{c}} = \sum_k E_{\tilde{c},k}$ and the specific internal energy $\varepsilon_{\tilde{c}} = E_{\tilde{c}}/m_{\tilde{c}}$ (Section 3.5).
- Remap the cell common pressure $p_{\tilde{c}}$ in a flux form, fluxes obtained by multiplication of the reconstructed pressure by the exchange integrals $I^1_{c,c'}$ (Section 3.6).
- Define the sub-cell masses by distribution of cell masses (Section 4.1). Construct total nodal mass $m_{\tilde{n}}$.
- Construct the inter-nodal mass fluxes by constrained optimization [50,51,49] (Section 4.1.2).
- Remap the nodal momenta $u_{\tilde{n}}$, $v_{\tilde{n}}$ in a flux form to recover the new nodal velocities, fluxes computed by multiplication of the inter-nodal mass fluxes by the reconstructed velocity components (Section 4.2).
- Compute the kinetic energy discrepancy (remap the nodal kinetic energy and find its difference from the kinetic energy obtained from the remapped velocities) and redistribute it to the material internal energies to obtain the final specific material internal energies (Section 5).
- Update the material pressures $p_{\tilde{c},k}$ from the equation of state applied to the remapped energy.

All these steps together form our multi-material remapping approach and are described in details in the previous sections.

7. Numerical examples

In this section, we demonstrate the properties of the described remapping algorithm on a set of selected numerical examples. We have decided to demonstrate the behavior on fully hydrodynamic tests where the influence of remapping of all fluid quantities affects the final solution.

For purely static remapping tests, one can see for example [11,32,34] where the same remapping scheme as described here was employed in the context of hybrid remapping.

7.1. 1D multi-material sod problem

First, let us demonstrate the properties for the well known Sod shock tube problem to demonstrate that the remapper does not introduce any artificial features to the solution and keeps its characteristics untouched. The standard planar single-material version of this problem is a standard hydrodynamic test for numerical methods, see for example [59]. Here, we investigate a multi-material modification of this problem [56], where the computational domain is split between two materials, as shown in Fig. 15. The materials have different ratio of specific heats γ, so they have really different physical

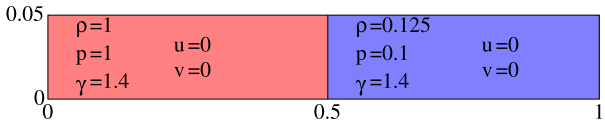


Fig. 15. Initial data for the multi-material Sod problem, different materials shown by red and blue colors. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

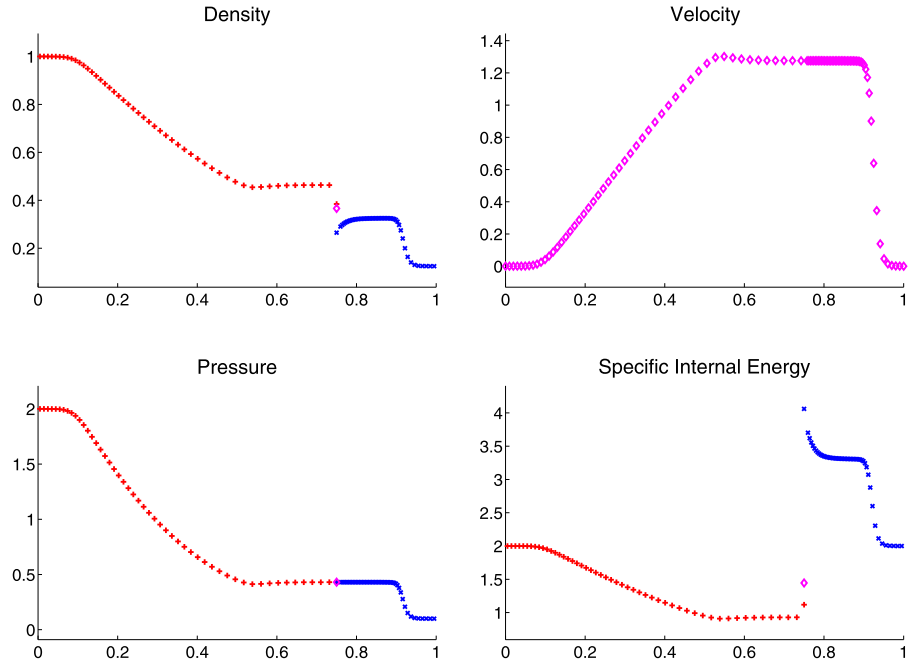


Fig. 16. Purely Lagrangian simulation of the two-material Sod problem (with one layer of two-material cells) in time $t = 0.2$, using the Tipton closure model. Cut through the domain shows the profiles of fluid density, velocity, pressure, and the specific internal energy. Red and blue colors denotes different materials, magenta color shows the average cell value. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

properties. The $\langle 0, 1\rangle \times \langle 0, 0.05\rangle$ domain is paved with 101×5 equidistant computational cells. The odd number in the x direction causes that the material interface at $x = 0.5$ is located in the middle of the cells and the simulation starts with already multi-material data. The simulation stops in the final time $t = 0.2$, just before the waves hit the domain boundary.

In Fig. 16, we can see the results of this simulation in the Lagrangian regime. The pressure-relaxation Tipton closure model [58,56] is used to represent the material interaction in the layer of multi-material cells at the center of the domain. For more information about the multi-material closure models, see [26,56]. The Lagrangian model preserves 1D symmetry of the problem. We can see relatively well resolved contact of both materials without excessive oscillations in any quantity. Small decrease of red density and small overshoot in specific internal energy on both sides of the interface is visible.

In Fig. 17, we can see the same simulation performed with the full ALE machinery, including the described remapping method. The ALE10 strategy was used (a typical strategy in the hydrodynamic simulations), performing one mesh rezoning/remapping step after every 10 Lagrangian steps. The 1D symmetry of the problem was preserved as all exchange integrals (and therefore fluxes of all quantities) in the y direction are zero as well as all corner fluxes. For the mesh rezoning, one step of a simple Laplacian averaging was used, so the boundaries of the cells are different from the purely Lagrangian simulation and the values are slightly different. We can see the most significant difference at the material interface where small and large computational cells belonging to different materials meet, so the rezoner did the biggest changes in this region. Still, all properties of the solution are preserved, including the material energy overshoots and the density decrease. No significant disturbance in any quantity due to the remapping is visible.

7.2. Multi-material Sedov-like problem

In this section, we demonstrate the performance of the remapper on a modification of the Sedov problem, this time containing four materials with different properties [33]. The $\langle 0, 1.2\rangle^2$ domain is covered by an equidistant orthogonal 100^2 computational mesh, the initial material distribution is shown in Fig. 18 (a), (b). Initially, all materials are at rest and have zero pressure (in practice, we use $p = 10^{-6}$), except the innermost (red) material with the radius of $r = 0.1$, corresponding to

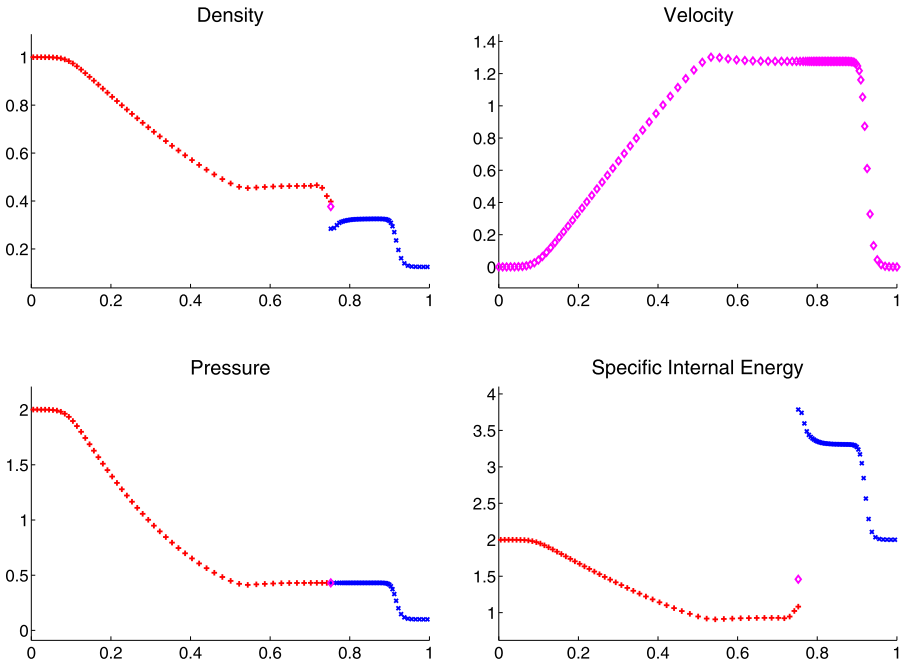


Fig. 17. ALE10 simulation of the two-material Sod problem in time $t = 0.2$, using the Tipton closure model. Cut through the domain shows the profiles of fluid density, velocity, pressure, and the specific internal energy. Red and blue colors denotes different materials, magenta color shows the average cell value. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

the explosive, which has $p = 16.388$, $\rho = 10$, and $\gamma = 1.4$. The second material (blue) has very low density $\rho = 0.2$ and very high gas constant $\gamma = 50$, and forms a ring or the outer radius $r = 0.2$ around the explosive. The next ring with the outer radius $r = 0.3$ is formed by the green material of $\rho = 5$ and $\gamma = 5/3$. Finally, the rest of the domain is filled by the cyan material with $\rho = 1$ and $\gamma = 1.4$, corresponding to the fluid in the standard Sedov problem setup. This problem is run (in the ALE10 regime, i.e. the mesh rezoning and remap is done after every tenth Lagrangian step) till the final time $t = 1$ and demonstrates mainly preservation of solution compactness and handling of multiple interfaces with significantly different material properties. The Tipton closure model is used in the Lagrangian step for the material interaction representation.

Similarly as in the standard Sedov problem, the "point" explosion (high pressure material at the origin) generates a circular shock wave, which passes through different material interfaces. Due to the different properties of each material, various waves pass through the interfaces and another waves are reflected back. The material interfaces are shifted in the outward direction and, in principle, the symmetric nature of the problem should be preserved (only approximately as the Cartesian mesh is used here). The material distribution in the final time is shown in Fig. 18 (c), (d). We can see that the layer of blue material becomes very thin and stretched to about one cell size, however, it does not break apart and stays compact and smooth. We can also see nicely circular shape of all material interfaces.

In Fig. 19, we can see the profiles of all fluid quantities as functions of the radius. For the Cartesian mesh, the solution cannot be fully symmetric. However, we can see that the symmetry is very well preserved, we can only observe small widening of the profiles in the post-shock part of the cyan material. Next to it, we can see the cell average values in the multi-material cells spread over the entire interface. This is not a problem in fact, this situation is caused by the different positions of the material interfaces in each cell. In general, we do not observe any significant oscillations of any quantity at the interfaces, even for such significantly different materials.

7.3. Rayleigh–Taylor instability

In this section, we demonstrate the properties of the complete remapping method on the standard Rayleigh–Taylor instability problem. The initial setup is taken from [21]. The $\langle -3, 3\rangle \times \langle 0, 1\rangle$ computational domain is covered by 100×600 equidistant rectangular computational mesh. The heavy fluid of $\rho = 2$ is located on the top of the light fluid of $\rho = 1$, their interface is curved according to $0.06\cos(2\pi x)$, so the materials covered by the pure material cells are separated by a layer of mixed cells. For both materials $\gamma = 1.4$, so the materials are different only formally. Initially, the fluid is at rest and the hydrostatic pressure is used in the entire domain. We run this problem in the Eulerian regime till the final time $t = 20$, gravity is $g = 0.1$.

The initial material density distribution and its evolution in time are shown in Fig. 20. We can see the development of the instability from the initial growth, over the well known mushroom shape, to the final complicated form containing secondary instabilities and thin red material filaments. We can see starting fragmentation of the filaments due to the

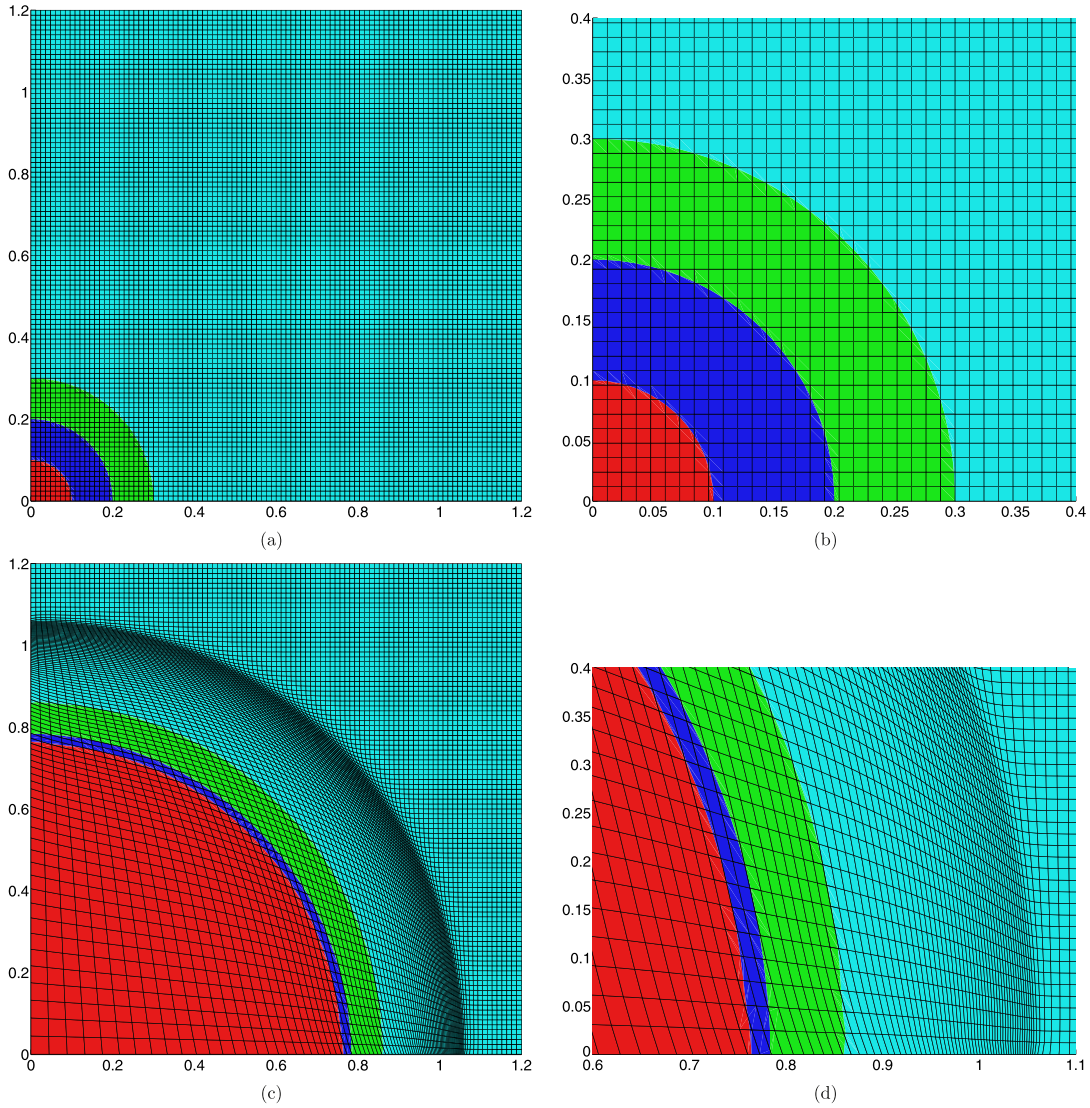


Fig. 18. Material distribution for the multi-material Sedov-like problem computed on the initially equidistant rectangular mesh in the ALE10 regime. Initial (a, b) and final (c, d) situation in time $t = 1$ is shown. The entire computational domain is shown in images (a, c), zoom of an interesting region containing all materials is shown in images (b, d). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

interference of the filaments with each other. This is basically a problem of the material reconstruction machinery which will show up for any remapping method as the filaments become thin and close to each other.

In Fig. 21, we can see the growth of the instability in time during the beginning of the simulation, which corresponds to the linear phase of the instability development. In this phase, the linear theory provides an estimate of the growth (black line in Fig. 21), see [21] for more details. We can see the growth for the simulations run in the Lagrangian and Eulerian regimes (the initial stage can be run in the Lagrangian regime before any significant mesh distortions appear). As we can see, the numerical instability growth is slightly below the analytic value, which corresponds well to the results presented for similar methods in [21]. The growth rate of the Eulerian simulation coincides with the growth rate of the Lagrangian simulation for most of the time range shown in the figure. We can see a slight difference for the later times – the growth rate of the Lagrangian simulation is actually smaller as the fixed computational mesh topology decelerates the instability development.

7.4. Multi-material shock–bubble interaction

In this section, we present the results for the well known problem of shock–bubble interaction. The experimental results of this problem have been presented in [55] and numerical simulations of this problem were shown in many papers, see for

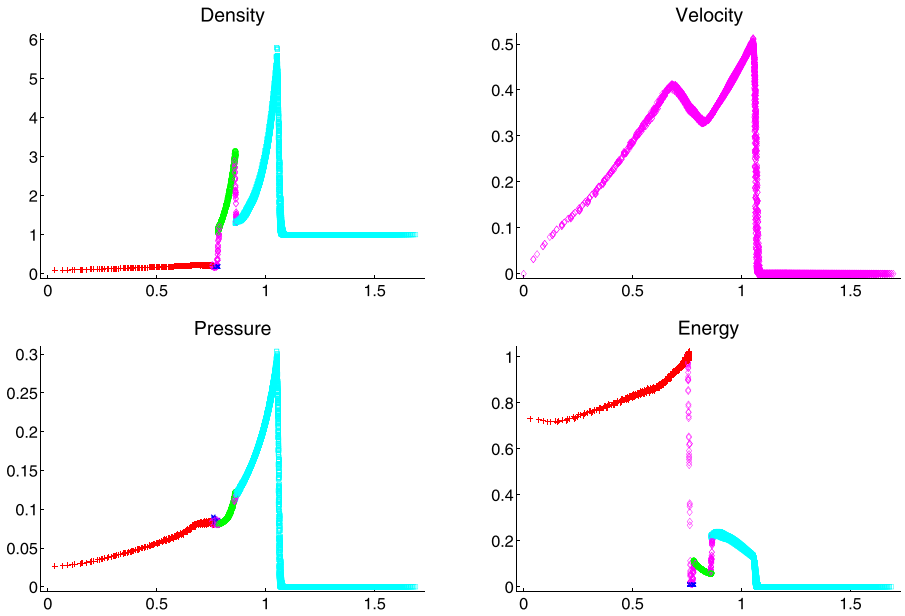


Fig. 19. Values of all fluid quantities (as functions of the radius) in time $t = 1$ for the multi-material Sedov problem on the Cartesian mesh run in the ALE10 regime. Different colors distinguish different materials, magenta diamonds represent the cell average values. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

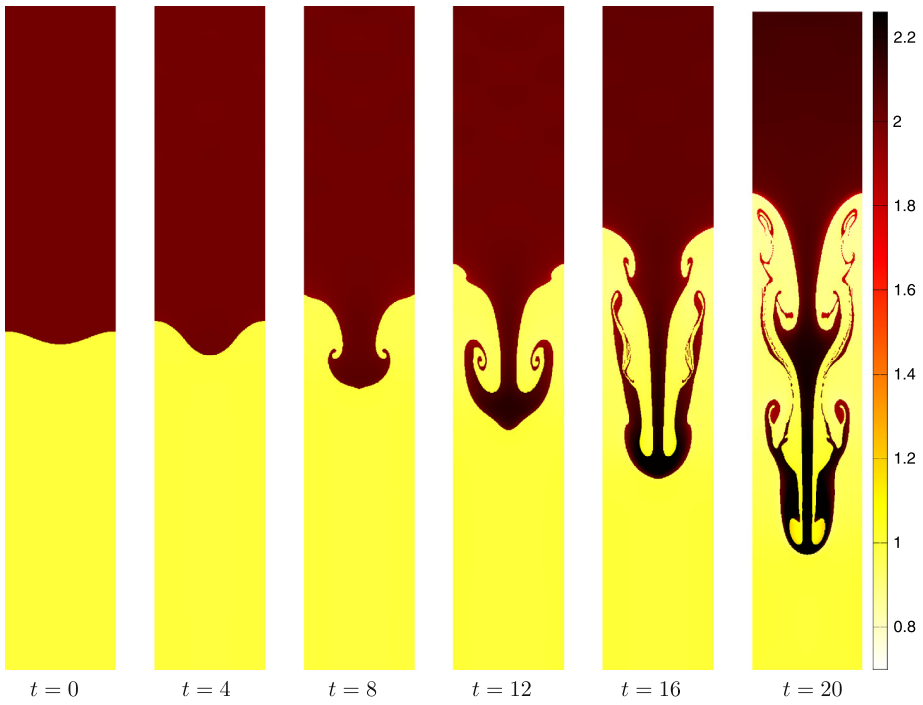


Fig. 20. The Rayleigh–Taylor instability problem in different times of the simulation. Different materials (formally, both materials have the same properties) are shown by different colors. The Eulerian regime on a rectangular equidistant 100×600 mesh was used. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

example [22] for a comparison with a cell-centered multi-material ALE scheme. The initially equidistant rectangular mesh of 520×144 cells covers the $\langle 0, 0.65\rangle \times \langle -0.089, 0.089\rangle$ domain. Initially, the fluid is at rest everywhere except the right boundary nodes, where $\boldsymbol{u} = [-124.824, 0]$ represents a piston generating a shock wave propagating through the domain. A Helium bubble is centered at $[0.32, 0]$ and has the radius $r = 0.025$, its parameters are $\rho = 0.181875$, $p = 10^5$, and

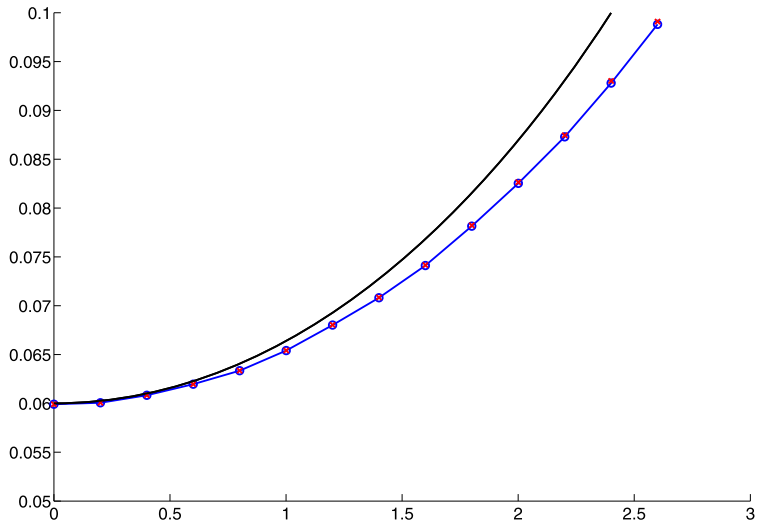


Fig. 21. Growth (computed as the difference of the y coordinates of the lowest fragment of the red material and the highest fragment of the yellow material) at the initial stage of the Rayleigh–Taylor instability problem computed in the Lagrangian (blue solid line) and Eulerian (red crosses) regime on the 100×600 mesh compared to the theoretical growth rate (black solid line). Both numerical growth rates almost coincide. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

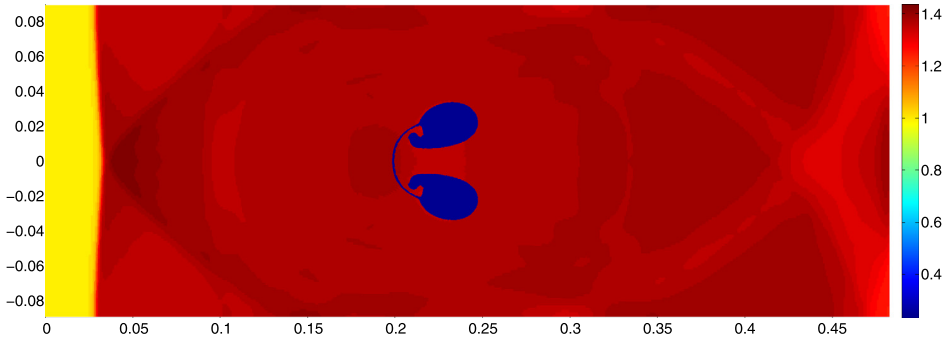


Fig. 22. Density distribution for the multi-material shock–bubble interaction problem in time $t = 1342.153 \cdot 10^{-6}$ run in the ALE10 regime.

$\gamma = 1.648$. The rest of the domain contains the air with $\rho = 1$, $p = 10^5$, and $\gamma = 1.4$. The simulation runs in the ALE10 regime and stops in the final time $t = 1342.153 \cdot 10^{-6}$, just before the shock wave hits the left domain boundary.

After the shock wave hits the bubble, two vortexes are generated. The bubble is split in two parts connected by a thin filament. The density in the entire domain can be seen in Fig. 22. In Fig. 23, we can see the material distribution in the vicinity of the bubble. The shape and location of the bubble are comparable with the results in [22]. We can see that the filament remains smooth and compact, it does not break apart even if the mesh is not aligned with the materials and its cells have higher aspect ratio.

7.5. Multi-material triple-point problem

The triple-point problem comes from [22], for its simulations in the context of the material reconstruction methods, see [33]. A rectangular $\langle 0, 7\rangle \times \langle 0, 3\rangle$ domain is split among three materials, which are initially at rest. Material boundaries are initially aligned with the equidistant mesh containing 490×210 square cells. Left from $x = 1$ line, a high-pressure material is located ($\gamma = 1.5$, $\rho = 1$, $p = 1$), which generates a shock wave propagating to the right. The rest of the domain is split by the $y = 1.5$ line into two parts. Above this line, a light material ($\gamma = 1.5$, $\rho = 0.125$, and $p = 0.1$) is located. Below this line, the third material ($\gamma = 1.4$, $\rho = 1$, and $p = 0.1$) is located. As there is initially the same pressure across the horizontal line, initially no waves are generated there. However, different fluid properties cause different speeds of the shock wave generated by the pressure gradient over the vertical line. Therefore, a vortex is generated, rolling all three materials around the triple point.

We run this simulation in the Eulerian regime until the final time $t = 5$ is reached. The material distribution in the final time is shown in Fig. 24. Zoom of the region around the triple point is shown in Fig. 25. As we can see, the materials

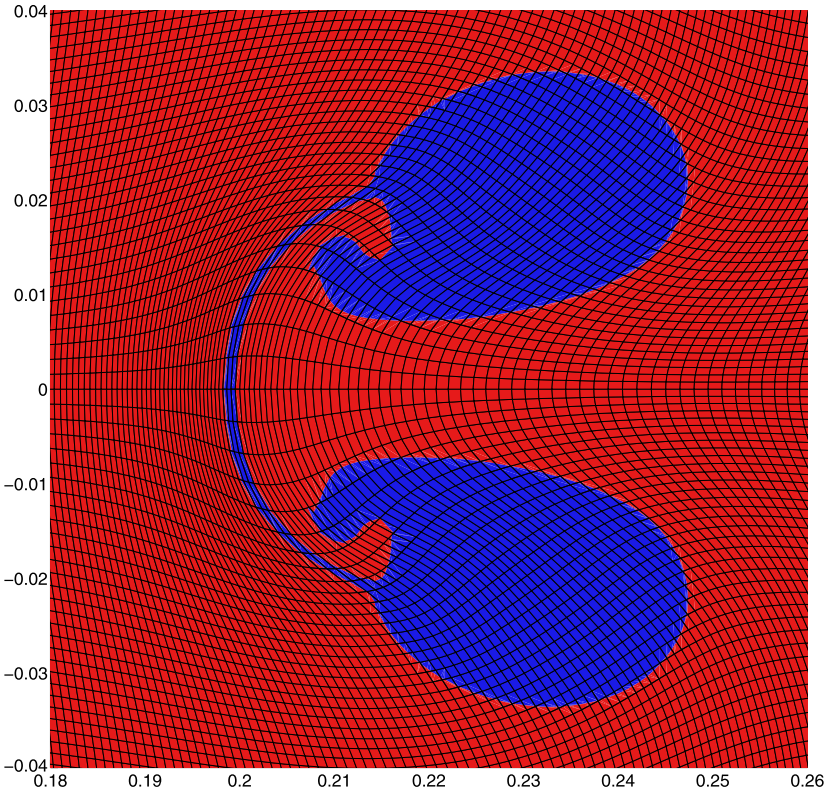


Fig. 23. Material distribution in the vicinity of the bubble for the multi-material shock–bubble interaction problem in time $t = 1342.153 \cdot 10^{-6}$ run in the ALE10 regime.

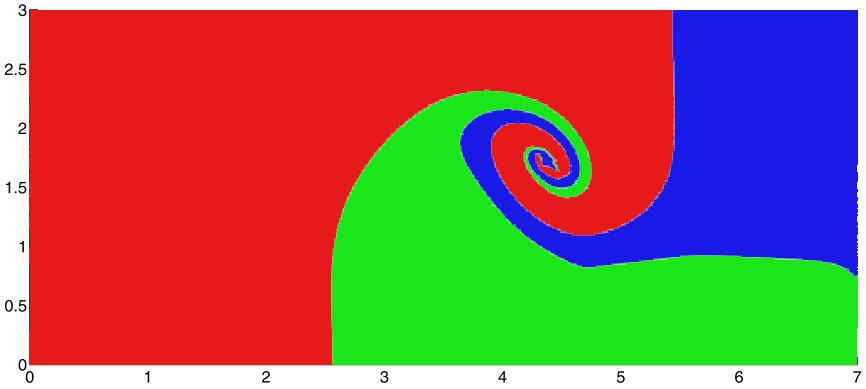


Fig. 24. Material distribution (result of the MOF interface reconstruction) for the triple-point problem in time $t = 5$, the simulation was done in the Eulerian regime.

remain reasonably compact and the interfaces are smooth. The only exception is the very tip of the green material, which is separated from the rest as its thickness decreases to the size of one mesh cell.

In Fig. 26, we can see the distribution of all fluid quantities. As we can see, there is no hot-spot nor oscillations in any quantity as a side-product of the remapper. All quantities remain smooth and the simulation can eventually continue for a longer time.

8. Conclusion

In this paper, we have presented a complete description of a new method for the remapping of all fluid quantities in the framework of the staggered multi-material Arbitrary Lagrangian–Eulerian methods. In the proposed method, all fluid

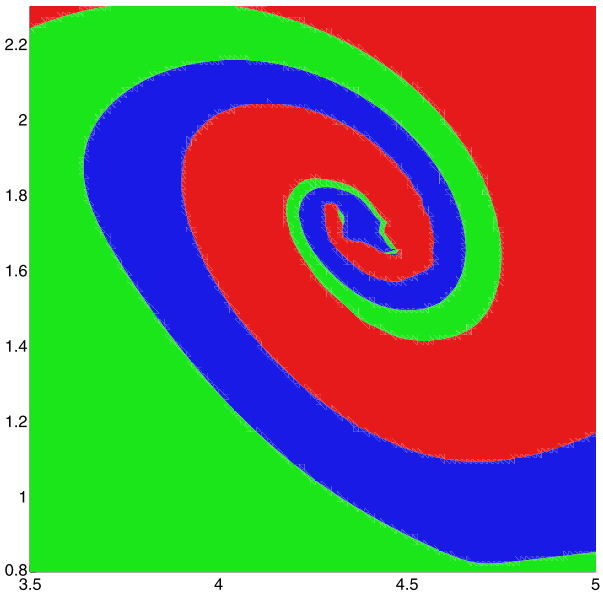


Fig. 25. Material distribution in the vicinity of the triple point for the triple-point problem in time $t = 5$, the simulation was done in the Eulerian regime.

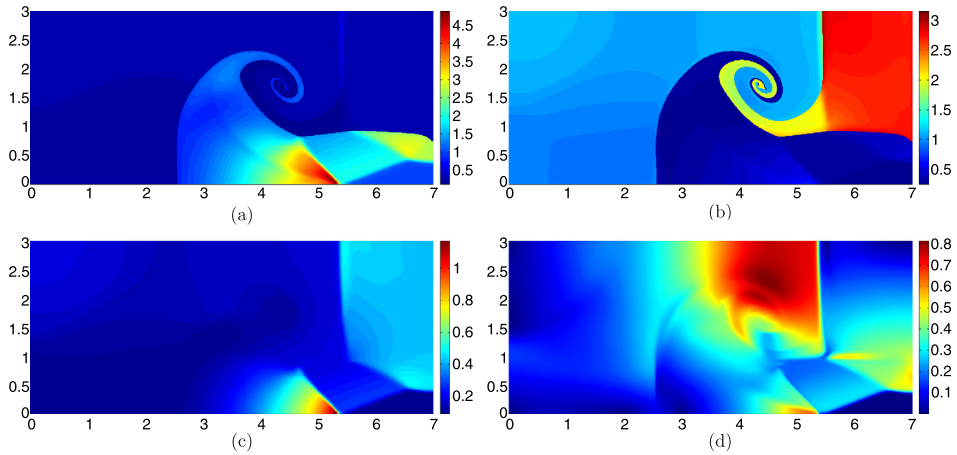


Fig. 26. Distribution of all fluid quantities for the triple-point problem in time $t = 5$, the simulation was done in the Eulerian regime. Material density (a), material specific internal energy (b), material pressure (c), and the nodal velocity magnitude (d) are shown in the entire computational domain.

quantities are remapped in a flux form, where the fluxes of all quantities are constructed from the geometric exchange integrals (computed by intersections). These integrals are precomputed at the beginning of the remapping step and reused for all quantities. Remapping of the material quantities (volume fractions and centroids), material densities, and internal energies is fully described in this paper. Next to it, we present the remapping of the common cell pressure, nodal mass, and nodal velocity values. For each quantity we have described the complete process of remapping, allowing the reader to implement the entire algorithm in his/her own flux-based hydrodynamic code. Properties of the remapper were shown on a suite of realistic numerical tests, in which the algorithm preserves all features of the solution and does not introduce any artificial numerical artifacts.

Acknowledgements

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 No. DE-AC52-06NA25396 and supported by the DOE Advanced Simulation and Computing (ASC) program. The authors acknowledge the partial support of the DOE Office of Science ASCR Program. Milan Kucharik was supported by the Czech Ministry of Education grants MSM 6840770022, MSM 6840770010,

and LC528, the Czech Science Foundation project P201/10/P086, and RVO 68407700. The authors thank H. Ahn, D. Bailey, A. Barlow, K. Lipnikov, P.-H. Maire, and M. Owen for fruitful and stimulating discussions over many years.

Appendix A. Geometric exchange integrals

In this section, we describe one individual part of the remapping algorithm – the integration of the low-degree polynomials over a polygon. These integrals are required for several purposes in the remapping step – they are used (1) for the computation of the geometric exchange integrals between the cells of the computational mesh, (2) for the computation of the material volumes and centroids, and (3) for the integration of the reconstructed density (and internal energy) function over the intersections. This approach is based on a direct use of the Green formula. Let us note that the polygon can be non-convex, degenerate (as described in Appendix B), or even self-intersecting,[1] the Green formula gives correct results for all these cases.

Let us compute the integrals of the polynomials 1, x, and y over an arbitrary polygon P. Using the Green theorem, we can rewrite them in the form

$$I_P^1 = \iint_P 1\,dx\,dy = \oint_{\partial P} x\,dy = \sum_{e\in\partial P} \int_{y_1}^{y_2} x\,dy, \tag{A.1}$$

$$I_P^x = \iint_P x\,dx\,dy = \oint_{\partial P} \frac{1}{2}x^2\,dy = \sum_{e\in\partial P} \int_{y_1}^{y_2} \frac{1}{2}x^2\,dy, \tag{A.2}$$

$$I_P^y = \iint_P y\,dx\,dy = -\oint_{\partial P} \frac{1}{2}y^2\,dx = -\sum_{e\in\partial P} \int_{x_1}^{x_2} \frac{1}{2}y^2\,dx, \tag{A.3}$$

where e represents an edge of polygon P, with ending points $[x_1, y_1]$ and $[x_2, y_2]$, and the following notation

$$I_P^{f(x,y)} = \iint_P f(x,y)\,dx\,dy \tag{A.4}$$

is used for the polygon integrals. Now, let us substitute for x in (A.1) and (A.2), and for y in (A.3) from the equation of line

$$x = x_1 + \frac{x_2 - x_1}{y_2 - y_1}(y - y_1), \tag{A.5}$$

$$y = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1), \tag{A.6}$$

and obtain the formulas in the form

$$I_P^1 = \sum_{e\in\partial P} \frac{1}{2}(x_1 + x_2)(y_2 - y_1), \tag{A.7}$$

$$I_P^x = \sum_{e\in\partial P} \frac{1}{6}\left(x_1^2 + x_1x_2 + x_2^2\right)(y_2 - y_1), \tag{A.8}$$

$$I_P^y = -\sum_{e\in\partial P} \frac{1}{6}\left(y_1^2 + y_1y_2 + y_2^2\right)(x_2 - x_1). \tag{A.9}$$

Let us note that all fractions included in the line equations (A.5), (A.6) cancel and there is no problem with the division by zero.

This integration process is used for computation of the geometric exchange integrals at the beginning of the remapping step. The integrals are pre-computed for all inter-cell intersections, and used in the construction of fluxes of all fluid quantities being remapped. They are also used in the computation of material volumes and material centroids in the Lagrangian mesh,

[1] Generally, neither the pure material sub-polygons nor their intersections with neighboring cells can be self-intersecting. However, the same formulas can be used for the computation of the integrals over the cell swept regions required during the hybrid remap [11,32,34]. The swept regions can clearly be self-overlapping.

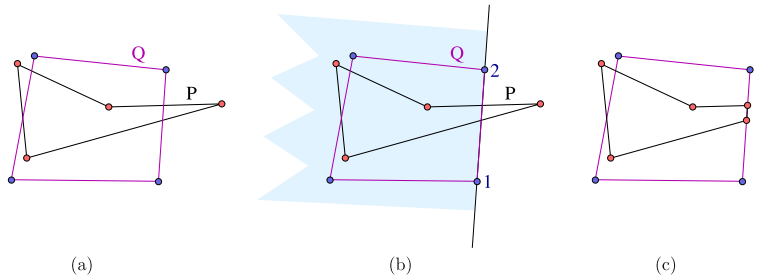


Fig. 27. (a) Non-convex Lagrangian P (black) and convex rezoned Q (magenta) polygons. (b) Left-oriented half-plane defined by 1–2 vertices of polygon Q. (c) New polygon obtained by adding two intersection vertices and removing the right vertex outside of the half-plane. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

$$V_{c,k} = I^1_{P_{c,k}}, \tag{A.10}$$

$$x_{c,k} = \frac{I^x_{P_{c,k}}}{I^1_{P_{c,k}}}, \tag{A.11}$$

$$y_{c,k} = \frac{I^y_{P_{c,k}}}{I^1_{P_{c,k}}}. \tag{A.12}$$

Appendix B. Robust intersection algorithm

For the computation of the intersections of the Lagrangian cells (or pure material sub-polygons in the case of multi-material cells) with the cells of the rezoned mesh, a robust and fast algorithm must be used. There exist several methods for convex polygon intersections, the best known approaches are probably the chasing algorithm described in [48] and the half-plane algorithms based on plain intersections of half-planes defined by the edges of both polygons [14]. Unfortunately, these algorithm are not able to deal with the non-convex polygons, which is our requirement in order to allow the algorithm to remap from non-convex Lagrangian meshes. Moreover, the usual algorithms have problems with the intersections of polygons with almost identical edges, which is a very common situation in continuous remapping. Therefore, we have developed a robust algorithm based on the half-plane intersections, incorporating an extremely robust, bisection based algorithm for intersecting close-to-parallel line segments. Let us note that due to its half-plane intersection nature, the algorithm has poor $n \times m$ complexity, where n and m are the numbers of vertices of the corresponding polygons; however, its performance is comparable with the alternative approaches as only small polygons (typically containing 3 to 5 vertices) are intersected in our application.

Suppose, that we cannot guarantee the convexity of the mesh resulting from the Lagrangian phase. This is a very usual situation, the Lagrangian solver can typically run even in the presence of non-convex (but still valid) cells. On the other hand, the meshes produced by the rezoner can be guaranteed being convex. Therefore, an intersection algorithm does not have to be able to intersect two general non-convex meshes, but must be able to handle one non-convex and one convex cell.

The algorithm which we suggest goes in the following way. Suppose that we have a generally non-convex polygon P (Lagrangian cell or sub-polygon of pure material in the non-convex cell) with n_P vertices, and a convex polygon Q (rezoned cell) with $n_Q = 4$ vertices, as shown in Fig. 27. The algorithm sweeps through all edges e_Q of the convex polygon Q in the counter-clock wise direction and constructs the equation for a straight line in the following form

$$l_{e_Q}(x, y) = (x_2 - x_1) * (y - y_1) - (y_2 - y_1) * (x - x_1), \tag{B.1}$$

where $[x_1, y_1]$ and $[x_2, y_2]$ stand for the ending points of the edge e_Q. This function represents a signed (unnormalized) distance of the point $[x, y]$ from the line defined by the edge e_Q. Clearly, if the point $[x, y]$ lies on this edge, then $l_{e_Q}(x, y) = 0$. If it lies in the left half-plane defined by this oriented edge, which means that it is located in the correct (internal) half-plane, then $l_{e_Q}(x, y) > 0$, otherwise the point is in the wrong (external) half-plane and $l_{e_Q}(x, y) < 0$.

The algorithm evaluates and remembers $L_i^{e_Q} = l_{e_Q}(x_i^P, y_i^P)$ for all vertices $[x_i, y_i]$ of P. If the values of $L_i^{e_Q}$ and $L_{i+1}^{e_Q}$ have different signs, the edge e_P connecting these two vertices is intersected by the edge e_Q. This intersection $[x_I, y_I] = e_P \cap e_Q$ is located and added to polygon P, with the value of l_{e_Q} equal to zero, $L_I^{e_Q} = l_{e_Q}(x_I, y_I) = 0$. The actual segment–segment intersection process is described later. Finally, the algorithm sweeps through the updated polygon P once again, and removes all vertices $[x_i, y_i]$ with $L_i^{e_Q} < 0$.

The last part of the polygon–polygon intersection algorithm is a robust algorithm for line segment–segment intersection. Let us note that in the context of the remapping process, we are very often dealing with the intersections of almost-parallel and almost-identical lines, which appear in the situation when the mesh rezoning step performs only a minor modification of the nodal positions. Our approach is based on the combination of the analytical line intersection formula with bisection.

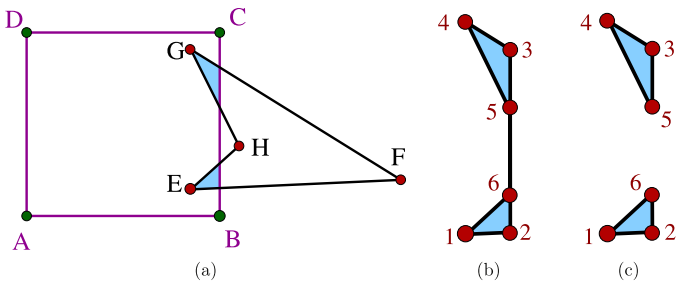


Fig. 28. Intersection of a convex polygon [*ABCD*] with a non-convex polygon [*EFGH*] (a) producing a degenerate polygon [123456] (b) instead of two separate polygons [1′2′3′] and [1″2″3″] (c).

Suppose, that we want to intersect a line segment of P defined by its end points $[x_1^P, y_1^P]$ and $[x_2^P, y_2^P]$, with a line segment of Q with end points $[x_1^Q, y_1^Q]$ and $[x_2^Q, y_2^Q]$. At first, we evaluate $l_{e_P}(x_1^Q, y_1^Q)$ and $l_{e_P}(x_2^Q, y_2^Q)$, where l_{e_P} means that the $[x_1, y_1]$ and $[x_2, y_2]$ points in the l definition (B.1) are set to the edge vertices of polygon P, $[x_1^P, y_1^P]$ and $[x_2^P, y_2^P]$. Similarly, we evaluate $l_{e_Q}(x_1^P, y_1^P)$ and $l_{e_Q}(x_2^P, y_2^P)$. In the situation that any of the following two conditions $\text{sign}(l_{e_P}(x_1^Q, y_1^Q)) \neq \text{sign}(l_{e_P}(x_2^Q, y_2^Q))$ and $\text{sign}(l_{e_Q}(x_1^P, y_1^P)) \neq \text{sign}(l_{e_Q}(x_2^P, y_2^P))$ is not satisfied, then the intersection does not exist, for example because it is located outside of one of the segments. In the opposite case, the intersection exists and can be located. As the first step, we compute the cross product of both lines,

$$c = \left(x_2^P - x_1^P\right)\left(y_2^Q - y_1^Q\right) - \left(y_2^P - y_1^P\right)\left(x_2^Q - x_1^Q\right), \tag{B.2}$$

and if its absolute value is big enough, $c > \epsilon$, the intersection is located analytically using the formulas

$$t = \frac{1}{c}\left(x_1^P\left(y_1^Q - y_2^Q\right) + x_1^Q\left(y_2^Q - y_1^P\right) + x_2^Q\left(y_1^P - y_1^Q\right)\right), \tag{B.3}$$

$$x_I = x_1^P + t\left(x_2^P - x_1^P\right), \tag{B.4}$$

$$y_I = y_1^P + t\left(y_2^P - y_1^P\right). \tag{B.5}$$

In the opposite case ($c < \epsilon$), the edge lines are close to parallel, and the analytical formula does not work due to the division by a small number. In this case, we switch to bisection. Because we know, that $l_{e_P}(x_1^Q, y_1^Q)$ and $l_{e_P}(x_2^Q, y_2^Q)$ have different signs, we can locate the place where $l_{e_P}(x, y) = 0$ using bisection in an arbitrary precision (practically, we use machine accuracy). With this approach we did not observe any problems with intersections of similar polygons, experienced with any other intersection algorithm available. Let us note that there is no hidden dependence on the value of chosen ϵ because the analytical formula produces the same intersection point as bisection for the cross product big enough, and we could use bisection anyways. In practical numerical simulations, we have used the value of $\epsilon = 10^{-12}$. We have decided to combine bisection with the analytical formula because it is significantly faster to evaluate than the bisection iterative process.

Let us also note that this algorithm can produce degenerate polygons as shown in Fig. 28, because one of the polygons can be non-convex and we never decide, whether the resulting intersection is a single polygon or whether it splits into multiple objects. For our purposes, the degenerate polygons do not introduce any problems, as we only use them for the computation of the geometric exchange integrals using the Green theorem, which works fine for any such polygons correctly. Let us note that the identification of distinct polygons would introduce a dependence of the algorithm on a chosen barrier, and significantly decrease robustness of the whole intersection process.

We have considered here the intersection algorithm for the case of convex and non-convex cells. For the intersection of the pure material sub-polygons with the new cell, the situation is similar. We did not consider the case of intersection of a pure material region broken into several pieces (which may be produced by the material reconstruction algorithm in the case of non-convex cells). This situation is avoided by breaking the material sub-polygons to (always convex) triangles and treating them separately. Our implementation of the MOF material reconstruction method always breaks the pure material polygons into triangles. Therefore, the polygon P is either a triangle (part of the pure material polygon in a multi-material cell) or a quadrilateral (pure cell). Therefore, no problems related to more pieces of a single material have to be faced, as they are treated one by one in the form of triangles.

Appendix C. Internal consistency of energy remap

Let us check the internal consistency of the internal energy remap by checking the internal energy of a piece of material eliminated from the cell during the rezoning/remapping process. Let us assume, that there is a piece of material k in cell c, which is fully transferred to the neighboring cell $\tilde{c}'$ and nothing remains in $\tilde{c}$, as shown in Fig. 29. From the following equation, it is obvious that the material mass is eliminated completely,

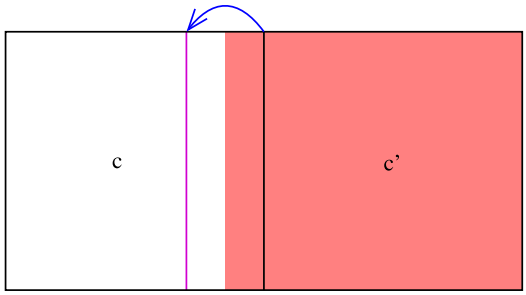


Fig. 29. Elimination of the red material from cell c. The old (black) and new (magenta) meshes are shown. Only one edge moved generating the mass fluxes of both red and white materials. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

$$\begin{aligned} m_{\tilde{c},k} &= m_{c,k} - F^m_{P_{c,k}\cap\tilde{c}'} = m_{c,k} - \int\limits_{P_{c,k}\cap\tilde{c}'} \rho_{c,k}(x,y)\,dx\,dy = m_{c,k} - \int\limits_{P_{c,k}} \rho_{c,k}(x,y)\,dx\,dy \\ &= m_{c,k} - \rho_{c,k} I^1_{P_{c,k}} - S^x_{c,k}\left(I^x_{P_{c,k}} - x_{c,k} I^1_{P_{c,k}}\right) + S^y_{c,k}\left(I^y_{P_{c,k}} - y_{c,k} I^1_{P_{c,k}}\right) \\ &= m_{c,k} - m_{c,k} - S^x_{c,k}\left(I^x_{P_{c,k}} - \frac{I^x_{P_{c,k}}}{I^1_{P_{c,k}}} I^1_{P_{c,k}}\right) + S^y_{c,k}\left(I^y_{P_{c,k}} - \frac{I^y_{P_{c,k}}}{I^1_{P_{c,k}}} I^1_{P_{c,k}}\right) \\ &= 0. \end{aligned} \tag{C.1}$$

To pass from the second to the third line, we used the information that the whole $P_{c,k}$ is fully included in the flux region $c\cap\tilde{c}'$, $P_{c,k} = P_{c,k}\cap\tilde{c}'$.

For the internal energy remap in the flux form (35), we can write the internal energy update in a similar form

$$E_{\tilde{c},k} = E_{c,k} - F^{\varepsilon}_{P_{c,k}\cap\tilde{c}'}, \tag{C.2}$$

and after substitution from (38), we can rewrite it as

$$\begin{aligned} E_{\tilde{c},k} &= m_{c,k}\varepsilon_{c,k} - \int\limits_{P_{c,k}\cap\tilde{c}'} \rho_{c,k}(x,y)\,\varepsilon_{c,k}(x,y)\,dx\,dy \\ &= m_{c,k}\varepsilon_{c,k} - \int\limits_{P_{c,k}} \rho_{c,k}(x,y)\varepsilon_{c,k}(x,y)\,dx\,dy, \end{aligned} \tag{C.3}$$

because the pure material polygon $P_{c,k}$ is fully covered by the flux region. After substituting the internal energy reconstruction from (40), we can rewrite it as

$$\begin{aligned} E_{\tilde{c},k} = m_{c,k}\varepsilon_{c,k} &- \int\limits_{P_{c,k}} \rho_{c,k}(x,y)\varepsilon_{c,k}\,dx\,dy - \int\limits_{P_{c,k}} \rho_{c,k}(x,y) S^{x,\varepsilon}_{c,k}\left(x - x^m_{c,k}\right)dx\,dy \\ &- \int\limits_{P_{c,k}} \rho_{c,k}(x,y) S^{y,\varepsilon}_{c,k}\left(y - y^m_{c,k}\right)dx\,dy. \end{aligned} \tag{C.4}$$

The first integral can be simplified as follows,

$$\begin{aligned} \int\limits_{P_{c,k}} \rho_{c,k}(x,y)\varepsilon_{c,k}\,dx\,dy &= \varepsilon_{c,k}\left(\int\limits_{P_{c,k}} \rho_{c,k}\,dx\,dy + \int\limits_{P_{c,k}} S^x_{c,k}(x - x_{c,k})\,dx\,dy + \int\limits_{P_{c,k}} S^y_{c,k}(y - y_{c,k})\,dx\,dy\right) \\ &= \varepsilon_{c,k}\left(m_{c,k} + S^x_{c,k}\left(I^x_{P_{c,k}} - \frac{I^x_{P_{c,k}}}{I^1_{P_{c,k}}} I^1_{P_{c,k}}\right) + S^y_{c,k}\left(I^y_{P_{c,k}} - \frac{I^y_{P_{c,k}}}{I^1_{P_{c,k}}} I^1_{P_{c,k}}\right)\right) \\ &= m_{c,k}\varepsilon_{c,k}. \end{aligned} \tag{C.5}$$

The second integral can be fully eliminated by the following process,

$$
\begin{aligned}
&\int_{P_{c,k}} \rho_{c,k}(x,y) S^{x,\varepsilon}_{c,k}\left(x - x^m_{c,k}\right) dx\,dy \\
&= S^{x,\varepsilon}_{c,k}\left(\rho_{c,k}\left(I^x_{P_{c,k}} - x^m_{c,k} I^1_{P_{c,k}}\right) + S^x_{c,k} \int_{P_{c,k}} \left(x - x^m_{c,k}\right)\left(x - x_{c,k}\right) dx\,dy + S^y_{c,k} \int_{P_{c,k}} \left(x - x^m_{c,k}\right)\left(y - y_{c,k}\right) dx\,dy\right) \\
&= S^{x,\varepsilon}_{c,k}\left(-S^x_{c,k}\left(I^{x^2}_{c,k} - \frac{I^x_{c,k}}{I^1_{c,k}} I^x_{c,k}\right) - S^y_{c,k}\left(I^{xy}_{c,k} - \frac{I^x_{c,k}}{I^1_{c,k}} I^y_{c,k}\right)\right) + S^{x,\varepsilon}_{c,k} S^x_{c,k}\left(I^{x^2}_{c,k} - x^m_{c,k} I^x_{c,k} - \frac{I^x_{c,k}}{I^1_{c,k}} I^x_{c,k} + \frac{I^x_{c,k}}{I^1_{c,k}} I^1_{c,k} x^m_{c,k}\right) \\
&\quad + S^{x,\varepsilon}_{c,k} S^y_{c,k}\left(I^{xy}_{c,k} - y^m_{c,k} I^x_{c,k} - \frac{I^x_{c,k}}{I^1_{c,k}} I^y_{c,k} + \frac{I^x_{c,k}}{I^1_{c,k}} I^1_{c,k} y^m_{c,k}\right) \\
&= 0.
\end{aligned} \tag{C.6}
$$

Similarly, we can show, that the third integral is equal to zero also. After substituting the formulas for the integrals, we can rewrite formula (C.4) as

$$E_{\tilde{c},k} = m_{c,k}\varepsilon_{c,k} - m_{c,k}\varepsilon_{c,k} - 0 - 0 = 0. \tag{C.7}$$

Thus, the internal energy is completely eliminated also, and the energy remapping approach is self-consistent in this sense also.

References

[1] H.T. Ahn, M. Shashkov, Multi-material interface reconstruction on generalized polyhedral meshes, J. Comput. Phys. 226 (2) (2007) 2096–2132.
[2] R.W. Anderson, N.S. Elliott, R.B. Pember, An arbitrary Lagrangian–Eulerian method with adaptive mesh refinement for the solution of the Euler equations, J. Comput. Phys. 199 (2) (2004) 598–617.
[3] P. Anninos, New VOF interface capturing and reconstruction algorithms, Technical Report UCRL-ID-135084, Lawrence Livermore National Laboratory, 1999.
[4] P. Anninos, Kull ALE: I. Unstructured mesh advection, interface capturing, and multiphase 2T RHD with material interfaces, Technical Report UCRL-ID-147297-PT-1, Lawrence Livermore National Laboratory, 2002.
[5] P. Anninos, Kull ALE: II. Grid motion on unstructured arbitrary polyhedral meshes, Technical Report UCRL-ID-147297-PT-2, Lawrence Livermore National Laboratory, 2002.
[6] P. Anninos, Multiphase advection and radiation diffusion with material interfaces on unstructured meshes, Technical Report UCRL-JC-150129, Lawrence Livermore National Laboratory, 2002.
[7] A. Barlow, R.N. Hill, M.J. Shashkov, Constrained optimization framework for interface-aware sub-scale dynamics closure models for multimaterial cells in Lagrangian and arbitrary Lagrangian–Eulerian hydrodynamics, Technical Report LA-UR-13-26180, Los Alamos National Laboratory, 2013.
[8] A.J. Barlow, A new Lagrangian scheme for multi-material cells, in: Proceeding of European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS Computational Fluid Dynamics Conference, Swansea, Wales, UK, 4–7 September 2001, 2001.
[9] T.J. Barth, Numerical methods for gasdynamic systems on unstructured meshes, in: C. Rohde, D. Kroner, M. Ohlberger (Eds.), An Introduction to Recent Developments in Theory and Numerics for Conservation Laws, Proceedings of the International School on Theory and Numerics for Conservation Laws, in: Lecture Notes in Computational Science and Engineering, Springer, Berlin, ISBN 3-540-65081-4, 1997.
[10] D.J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Comput. Methods Appl. Mech. Eng. 99 (2–3) (1992) 235–394.
[11] M. Berndt, J. Breil, S. Galera, M. Kucharik, P.-H. Maire, M. Shashkov, Two step hybrid remapping (conservative interpolation) for multimaterial arbitrary Lagrangian–Eulerian methods, J. Comput. Phys. 230 (17) (2011) 6664–6687.
[12] Yu. Bondarenko, Yu. Yanilkin, Computation of the thermodynamic parameters in the mixed cells in gas dynamics, Math. Model. 14 (6) (2002) 63–81.
[13] E.J. Caramana, D.E. Burton, M.J. Shashkov, P.P. Whalen, The construction of compatible hydrodynamics algorithms utilizing conservation of total energy, J. Comput. Phys. 146 (1998) 227–262.
[14] M. de Berg, M. van Krefeld, M. Overmars, O. Schwarzkopf, Computational Geometry: Algorithms and Applications, Springer, 1998.
[15] R.B. DeBar, Fundamentals of the KRAKEN code, Technical Report UCIR-760, Lawrence Livermore Laboratory, 1974.
[16] V.I. Delov, V.V. Sadchikov, Comparison of several models for computation of thermodynamical parameters for heterogeneous Lagrangian cells, VANT, Math. Model. Phys. Process. 1 (2005) 57–70.
[17] B. Despres, F. Lagoutiere, Numerical resolution of a two-component compressible fluid model with interfaces, Prog. Comput. Fluid Dyn. 7 (6) (2007) 295–310.
[18] J. Donea, A. Huerta, J.-Ph Ponthot, A. Rodriguez-Ferran, Arbitrary Lagrangian–Eulerian methods, in: E. Stein, R. de Borst, T. Hughes (Eds.), The Encyclopedia of Computational Mechanics, Wiley, 2004, pp. 413–437, Chapter 14.
[19] J.K. Dukowicz, J.R. Baumgardner, Incremental remapping as a transport/advection algorithm, J. Comput. Phys. 160 (1) (2000) 318–335.
[20] V. Dyadechko, M. Shashkov, Reconstruction of multi-material interfaces from moment data, J. Comput. Phys. 227 (11) (2008) 5361–5384.
[21] J. Fung, M. Francois, E. Dendy, M. Kenamond, R. Lowrie, Calculations of the Rayleigh–Taylor instability: RAGE and FLAG hydrocode comparisons, in: Proceedings of NECDC06, 2006.
[22] S. Galera, P.-H. Maire, J. Breil, A two-dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction, J. Comput. Phys. 229 (16) (2010) 5755–5787.
[23] A.K. Harrison, M.J. Shashkov, J. Fung, J.R. Kamm, T.R. Canfield, Development of a sub-scale dynamics model for pressure relaxation of multi-material cells in Lagrangian hydrodynamics, in: L. Soulard (Ed.), New Models and Hydrocodes for Shock Wave Processes in Condensed Matter, Paris, France, May 24–28, 2010, in: EPJ Web of Conferences, vol. 10, 2010, p. 000039.
[24] R.N. Hill, A. Barlow, M.J. Shashkov, Interface-aware sub-scale dynamics closure model, in: International Conference on Numerical Methods in Multiphase Flows, State College, Pennsylvania, United States, June 12–14, 2012. Report of Los Alamos National Laboratory. See also http://www.mmm.ucar.edu/eulag/Workshop2012-agenda.html.
[25] C.W. Hirt, A.A. Amsden, J.L. Cook, An arbitrary Lagrangian–Eulerian computing method for all flow speeds, J. Comput. Phys. 14 (3) (1974) 227–253.

[26] J.R. Kamm, M.J. Shashkov, J. Fung, A.K. Harrison, T.R. Canfield, A comparative study of various pressure relaxation closure models for one-dimensional two-material Lagrangian hydrodynamics, Int. J. Numer. Methods Fluids 65 (11–12) (2011) 1311–1324.
[27] M. Kenamond, M. Bement, M. Shashkov, Compatible, total energy conserving and symmetry preserving arbitrary Lagrangian–Eulerian hydrodynamics in 2D rz-cylindrical coordinates, J. Comput. Phys. (2013), LA-UR 12-26781. Submitted for publication.
[28] P. Kjellgren, J. Hyvarinen, An arbitrary Lagrangian–Eulerian finite element method, Comput. Mech. 21 (1) (1998) 81–90.
[29] P. Knupp, S. Steinberg, Fundamentals of Grid Generation, CRC Press, Boca Raton, 1993.
[30] P.M. Knupp, L.G. Margolin, M.J. Shashkov, Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods, J. Comput. Phys. 176 (1) (2002) 93–128.
[31] P.M. Knupp, Winslow smoothing on two-dimensional unstructured meshes, Eng. Comput. 15 (1999) 263–268.
[32] M. Kucharik, J. Breil, S. Galera, P.-H. Maire, M. Berndt, M. Shashkov, Hybrid remap for multi-material ALE, Comput. Fluids 46 (1) (2011) 293–297.
[33] M. Kucharik, R.V. Garimella, S.P. Schofield, M.J. Shashkov, A comparative study of interface reconstruction methods for multi-material ALE simulations, J. Comput. Phys. 229 (7) (2010) 2432–2452.
[34] M. Kucharik, M. Shashkov, One-step hybrid remapping algorithm for multi-material arbitrary Lagrangian–Eulerian methods, J. Comput. Phys. 231 (7) (2012) 2851–2864.
[35] D. Kuzmin, R. Lohner, S. Turek (Eds.), Flux-Corrected Transport: Principles, Algorithms, and Applications, Springer, 2012.
[36] R. Loubere, P.-H. Maire, M. Shashkov, J. Breil, S. Galera, ReALE: A reconnection-based arbitrary-Lagrangian–Eulerian method, J. Comput. Phys. 229 (12) (2010) 4724–4761.
[37] R. Loubere, M. Shashkov, A subcell remapping method on staggered polygonal grids for arbitrary-Lagrangian–Eulerian methods, J. Comput. Phys. 209 (1) (2005) 105–138.
[38] G. Luttwak, Second order discrete rezoning, in: S.C. Schmidt, et al. (Eds.), High Pressure Science and Technology, AIP Conf. Proc. 309 (1994) 1777–1780.
[39] G. Luttwak, J. Falcovitz, Slope limiting for vectors: A novel vector limiting algorithm, Int. J. Numer. Methods Fluids 65 (11–12) (2011) 1365–1375.
[40] G. Luttwak, J. Falcovitz, VIP (Vector Image Polygon) multi-dimensional slope limiters for scalar variables, Comput. Fluids 83 (2013) 90–97.
[41] G. Luttwak, S. Hillel, J. Falcovitz, Lump-consistent remapping for ALE hydrodynamics, in: Presented at Advances Computational Mechanics, San Diego, USA, February 24–27, 2013.
[42] P.-H. Maire, A high-order one-step sub-cell force-based discretization for cell-centered Lagrangian hydrodynamics on polygonal grids, Comput. Fluids 46 (1) (2011) 341–347.
[43] P.-H. Maire, R. Loubere, P. Vachal, Staggered Lagrangian discretization based on cell-centered Riemann solver and associated hydrodynamics scheme, Commun. Comput. Phys. 10 (4) (2011) 940–978.
[44] L.G. Margolin, Introduction to "An arbitrary Lagrangian–Eulerian computing method for all flow speeds", J. Comput. Phys. 135 (2) (1997) 198–202.
[45] D.J. Mavriplis, Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes, in: 16th AIAA Computational Fluid Dynamics Conference, Orlando, Florida, June 23–26, 2003, AIAA 2003-3986.
[46] D.S. Miller, G.B. Zimmerman, An algorithm for time evolving volume fractions in mixed zones in Lagrangian hydrodynamics calculations, Technical Report UCRL-PRES-223908, Lawrence Livermore National Laboratory, 2006.
[47] A. Murrone, H. Guillard, A five equation reduced model for compressible two phase flow problems, J. Comput. Phys. 202 (2005) 664–698.
[48] J. O'Rourke, Computational Geometry in C, Cambridge University Press, Cambridge, UK, ISBN 0-521-64010-5, 1996.
[49] J.M. Owen, Subzonal corner mass treatment during ALE remap, Technical Report, Lawrence Livermore National Laboratory, 2011, LLNL-PRES-413307.
[50] J.M. Owen, M. Shashkov, ALE remap treatment consistent with corner based compatible total energy conserving lagrangian methods, in: Nuclear Explosive Code Developers Conference, Los Alamos, NM, 2010, LLNL-PRES-457355.
[51] J.M. Owen, M. Shashkov, A corner and dual mesh ALE remapping algorithm for use with the compatible energy Lagrangian discretization, in: SIAM Conference on Computation Science and Engineering, Reno, NV, 2011, LLNL-PRES-457355.
[52] M.J. Owen, M.J. Shashkov, Arbitrary Lagrangian Eulerian remap treatments consistent with staggered compatible total energy conserving Lagrangian methods, J. Comput. Phys. (2013), LA-UR 13-24423. Submitted for publication.
[53] J.S. Peery, D.E. Carroll, Multi-material ALE methods in unstructured grids, Comput. Methods Appl. Mech. Eng. 187 (3–4) (2000) 591–619.
[54] 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, Technical Report, LLNL, 2000, UCRL-JC-139820.
[55] J.J. Quirk, S. Karni, On the dynamics of a shock–bubble interaction, J. Fluid Mech. 318 (1996) 129–163.
[56] M. Shashkov, Closure models for multimaterial cells in arbitrary Lagrangian–Eulerian hydrocodes, Int. J. Numer. Methods Fluids 56 (8) (2008) 1497–1504.
[57] H. Stewart, B. Wendroff, Two-phase flow: Models and methods, J. Comput. Phys. 56 (1984) 363–409.
[58] R.E. Tipton, CALE mixed zone pressure relaxation model, Private communication, 1989.
[59] E.F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, Springer-Verlag, Berlin, Heidelberg, ISBN 3-540-61676-4, 1997.
[60] J. Velechovsky, M. Kucharik, R. Liska, M. Shashkov, P. Vachal, Symmetry- and essentially-bound-preserving flux-corrected remapping of momentum in staggered ALE hydrodynamics, J. Comput. Phys. 255 (2013) 590–611.
[61] J. Velechovsky, M. Kucharik, R. Liska, M. Shashkov, P. Vachal, Symmetry-preserving momentum remap for ALE hydrodynamics, J. Phys. Conf. Ser. 454 (1) (2013) 012003.
[62] A.M. Winslow, Equipotential zoning of two-dimensional meshes, Technical Report UCRL-7312, Lawrence Livermore National Laboratory, 1963.
[63] D.L. Youngs, Time dependent multi-material flow with large fluid distortion, in: K.W. Morton, M.J. Baines (Eds.), Numerical Methods for Fluid Dynamics, Academic Press, 1982, pp. 273–285.
[64] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 31 (3) (1979) 335–362.

Journal of Computational Physics 255 (2013) 590–611

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Symmetry- and essentially-bound-preserving flux-corrected remapping of momentum in staggered ALE hydrodynamics

J. Velechovský [a,*], M. Kuchařík [a], R. Liska [a], M. Shashkov [b], P. Váchal [a]

[a] Czech Technical University in Prague, FNSPE, Břehová 7, Praha 1, 115 19, Czech Republic
[b] Los Alamos National Laboratory, XCP-4, MS-F644, Los Alamos, NM 87545, USA

ARTICLE INFO

Article history:
Received 20 March 2013
Received in revised form 5 August 2013
Accepted 18 August 2013
Available online 30 August 2013

Keywords:
Staggered hydrodynamics
ALE
Remap of vectors
Symmetric reconstruction

ABSTRACT

We present a new flux-corrected approach for remapping of velocity in the framework of staggered arbitrary Lagrangian–Eulerian methods. The main focus of the paper is the definition and preservation of coordinate invariant local bounds for velocity vector and development of momentum remapping method such that the radial symmetry of the radially symmetric flows is preserved when remapping from one equiangular polar mesh to another. The properties of this new method are demonstrated on a set of selected numerical cyclic remapping tests and a full hydrodynamic example.

1. Introduction

In the simulations of fluid flows, various numerical methods can be used. These methods have been traditionally referred to either as Lagrangian or Eulerian, depending on the treatment of the computational mesh in time, each with its own advantages and disadvantages. In a pioneering paper [10], Hirt et al. developed the formalism for a mesh, motion of which can be determined as an independent degree of freedom, and showed that this general framework could be used to combine the best properties of the Lagrangian and Eulerian methods. This class of methods has been termed Arbitrary Lagrangian–Eulerian or ALE. This approach is currently very popular and many researchers have contributed to this topic [2,27,31,1,29,19,9].

The ALE scheme is typically subdivided into three distinct phases: (1) a Lagrangian phase, in which the solution and the computational mesh are advanced in time; (2) a rezoning phase, in which the nodes of the computational mesh are moved to improve the mesh quality; and (3) a remapping stage, in which the Lagrangian solution is conservatively interpolated (remapped) onto the rezoned mesh. Here we focus on the last stage of the ALE algorithm – remapping. In this paper we use the staggered compatible discretization on logically rectangular meshes, where discrete values of density, internal energy and pressure are assigned to the computational cells (cell-centered quantities), while velocity $\boldsymbol{w}$ belongs to the nodes [5]. Remapping of cell-centered conservative quantities (mass, for example) in this context is described in detail elsewhere, [8,28,15]. Usually, the remapping of mass is done in such way that the physically justified bounds for density are satisfied – for example, remapped density in a particular cell is supposed to be in the bounds defined by maximum and minimum values of the density in the neighboring cells before remap. When the remapped density is not in bounds, one can put it into bounds by the repair technique [35,20]. In this paper, we are specifically interested in the remap of nodal velocity. An overview of traditional methods can be found in the seminal review paper by Benson [2].

* Corresponding author.
E-mail address: velecjan@fjfi.cvut.cz (J. Velechovský).

0021-9991/$ – see front matter
http://dx.doi.org/10.1016/j.jcp.2013.08.037

There are several requirements for the remapping of velocity. First of all, it is the conservation of total momentum. It implies that nodal momentum is remapped and then nodal velocity is recovered as a ratio of remapped nodal momentum and remapped nodal mass, that is, the remapped velocity is a derived quantity. This also brings the question how to remap nodal mass in a way consistent with remapping of cell mass [14,12]. Another requirement, which is often called DeBar condition [7], is the requirement that if velocity is constant, then it is supposed to be remapped exactly for arbitrary density distribution. The DeBar condition is usually satisfied by constructing the momentum flux as a product of corresponding mass flux by some reconstructed velocity. Therefore, for given inter-nodal mass fluxes, the momentum (and velocity) remap is completely defined by the velocity in the momentum flux.

Another important question is how to define bound preservation for velocity. The traditional way is to define bounds for Cartesian components of the velocity vector. However, it was recognized that such definition is not coordinate invariant and, in particular, gives non-symmetric bounds (for example, different bounds for radial velocity vector on an equiangular polar mesh for nodes with the same radius but different angles).

A very interesting new idea, which was suggested in the context of vector field reconstruction, defines the bounds by using the Vector Image Polygon (VIP) [21] constructed as the convex hull of velocity vectors in the plane, where for radial flow on the axis one has the Cartesian components of the velocities and velocities in the neighboring vertices are represented by points in this plane. Such convex polygon can be quite complicated in shape. This approach employs a constrained-optimization procedure which allows to satisfy this type of constraints in the context of reconstruction. We are exploring this approach in the symmetric remapping context in [37]. The VIP approach has been already applied to remapping in [23,22].

In the framework of cell-centered Lagrangian discretization and vector field reconstruction inside the cell it was suggested to use bounds related to the projection of the velocity vector to principal axes of deformation tensor direction [24–26], that is, to use directions related to the flow.

In our paper we a use similar idea to define coordinate invariant bounds at mesh points. We project all velocity vectors in neighboring points into the direction of velocity at the point under consideration and the direction orthogonal to it – this is a linear transformation, achieved by applying rotation matrix to the velocity vectors; then we define bounds using these projections. In particular, it allows us to obtain the same bounds for radial velocity vector on an equiangular polar mesh for nodes with the same radius but different angles. This is a necessary condition for preservation of the spatial symmetry when remapping from one equiangular polar mesh to another. The preservation of this type of spatial symmetry is crucial for certain types of applications, such as those related to inertial confinement fusion (ICF). Most of these symmetry requiring applications are defined in the axisymmetric cylindrical geometry. In this paper we however stay in Cartesian geometry, making the first step concerning symmetry preservation. The generalization of our approach to cylindrical geometry will be the next step.

To enforce the defined bound preservation we use a non-trivial modification of the Flux-Corrected Remapping (FCR) (see, for example, [17]), which combines low- and high-order momentum fluxes obtained from the corresponding low- and high-order reconstruction of the velocity vector.

The remaining text is organized in the following way. The problem treated in this paper is defined in Section 2. The bounds for the velocity vector are defined in Section 3. The velocity reconstruction methods used are reviewed in Section 4. The main contribution of this paper, the symmetry preserving flux-corrected remapping algorithm, is presented in Section 5. Numerical examples are shown in Section 6. Three appendices contain the proof of symmetry of the piecewise linear velocity reconstruction, the proof of velocity bounds preservation for low-order velocity remap and the proof of symmetry of flux-corrected velocity remap.

2. Problem statement

Assume that we have two computational meshes – old (Lagrangian) mesh $\{c\}$ and new (rezoned) mesh $\{\tilde{c}\}$. Further also assume that both meshes have the same connectivity and are close to each other (meaning that each new cell $\tilde{c}$ stays inside the region defined by old cell c and its immediate neighbors, including corner neighbors, in the old mesh). The remap of cell mass can be written in the following flux form

$$\widetilde{m}_c = m_c + \sum_{c' \in C'(c)} F^m_{c,c'}, \tag{1}$$

where $C'(c)$ represents the set of all cells neighboring to c (including corners) and $F^m_{c,c'}$ stands for the (outwards oriented) mass flux from c to c', so $F^m_{c,c'} = -F^m_{c',c}$ as shown in Fig. 1(a). Depending on the particular remapping method, the $F^m_{c,c'}$ fluxes can be constructed in various ways. For example, they can be zero for corner neighbors, while for the edge neighbors the swept-region approach [15] is used. Another possibility is the integration of the density function over all intersections of the original cell c with all its neighbors in the new mesh and vice versa, see [8,28] for more details. This approach is more computationally expensive, however, unlike the swept region approach, it can be directly generalized for multi-material flows. Eventually, one can combine both approaches in one of the hybrid ways described in [3,11,13], requiring to construct the expensive intersections only in the vicinity of material interfaces, while computationally cheap swept fluxes are computed in pure material regions covering typically most of the computational domain. For the rest of the paper it does not

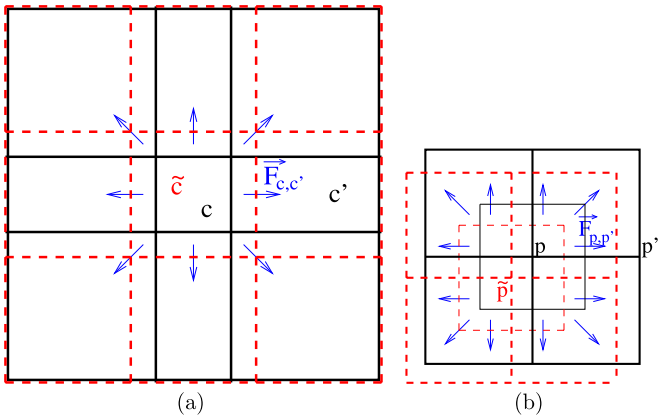


Fig. 1. Location of inter-cell (a) and inter-nodal (b) fluxes used for staggered remap. Lagrangian mesh is shown by thick black solid line, rezoned mesh by thick red dashed line, fluxes by blue arrows, dual (nodal) cells by corresponding thin lines.

matter which particular method is utilized, it is only important that a flux-based method is used and $F^m_{c,c'}$ represents the total mass flux between the cells.

Symmetry preservation of inter-cell mass fluxes is necessary to achieve the symmetry of all remaining fluid quantities. Let us assume that both Lagrangian and rezoned meshes are equiangular polar meshes, and assume a symmetric distribution of cell density. It implies that masses are symmetric, i.e. masses m_c, corresponding to the same radius, are the same (independent on the angle). As both meshes are equiangular, no fluxes in the angular and diagonal directions exist and all fluxes in the radial direction are symmetric too, because volumes of all intersections defining the fluxes are same. This the implies symmetry of the remapped cell masses $\widetilde{m}_c$. As we have mentioned in the introduction, to remap the nodal momentum and velocity we need to know how to remap nodal masses. In this paper, we use nodal mass remap in a flux form analogous to (1), however, we split the horizontal and vertical inter-nodal fluxes into two parts corresponding to the particular attached cells, as shown in Fig. 1(b),

$$\widetilde{m}_p = m_p + \sum_{c\in\mathcal{C}(p)} \sum_{\substack{q\in\mathcal{P}(c)\\ q\neq p}} F^m_{p,q,c}. \tag{2}$$

Here, $\mathcal{C}(p)$ stands for the set of all cells adjacent to node p, $\mathcal{P}(c)$ stands for the set of all nodes (points) of cell c, and $F^m_{p,q,c}$ is the inter-nodal mass flux from point p to point q through cell c, so $F^m_{p,q,c} = -F^m_{q,p,c}$. To avoid the double sums in the formula, we can introduce a simplified formula

$$\widetilde{m}_p = m_p + \sum_{s\in\mathcal{S}(p)} F^m_{p,s}, \tag{3}$$

where $\mathcal{S}(p) = \{(q,c)\colon\ c\in\mathcal{C}(p),\ q\in\mathcal{P}(c),\ q\neq p\}$ is the set of pairs (neighboring node, adjacent cell) giving all needed flux locations.

The most natural way of constructing the inter-nodal fluxes is by intersecting the dual cells corresponding to old and new mesh nodes similarly as we did for computational cells in the case of remap of cell mass. However, the dual cells can be generally non-convex, which significantly increases the computational cost and decreases the robustness of available intersection algorithms. Moreover, in case of multi-material flows one would need to deal with the material reconstruction on a subzonal level in order to construct the intersections, which again complicates the algorithm and increases its computational cost. Therefore, a simpler and cheaper approach is used. One possibility is the interpolation of the inter-cell mass fluxes $F^m_{c,c'}$ to the inter-nodal ones $F^m_{p,s}$ in a way similar to the one described in [32]. Another approach [30] uses the constrained optimization to find the inter-nodal mass fluxes as close to the target fluxes (such as those from [32]) as possible, while reaching the desired subcell masses. In this way it is guaranteed, that if the old nodal mass was defined as the average mass of adjacent cells, the remapped nodal mass is again an average of the new cell masses. The particular method is not important for this paper, it is crucial that the nodal mass is remapped in the flux form (3), where $F^m_{p,s}$ (with $s=(q,c)$) is the total inter-nodal mass flux (sum of fluxes of all materials) from point p to point q through cell c and the nodal mass remapping preserves symmetry of nodal mass under the same assumptions as cell mass.

For the remap of nodal momentum vector, formula analogous to (3) is used,

$$\widetilde{\boldsymbol{\mu}}_p = \boldsymbol{\mu}_p + \sum_{s\in\mathcal{S}(p)} \boldsymbol{F}^{\mu}_{p,s}, \tag{4}$$

where the vectors of momentum are defined as $\widetilde{\boldsymbol{\mu}}_p = \widetilde{m}_p\widetilde{\boldsymbol{w}}_p$ and $\boldsymbol{\mu}_p = m_p\boldsymbol{w}_p$ (where $\boldsymbol{w}_p$ is the velocity vector at point p), so it can be rewritten for nodal velocity as

$$\widetilde{\boldsymbol{w}}_p = \frac{1}{\widetilde{m}_p}\left(m_p \boldsymbol{w}_p + \sum_{s\in\mathcal{S}(p)} \boldsymbol{F}^{\mu}_{p,s}\right). \tag{5}$$

The new nodal mass $\widetilde{m}_p$ is already known as it has been remapped above using (3), and the momentum fluxes $\boldsymbol{F}^{\mu}_{p,s}$ can be obtained in several ways. Although alternative constructions of $\boldsymbol{F}^{\mu}_{p,s}$ are possible, we prefer reusing the computed mass fluxes $F^{m}_{p,s}$ and constructing the momentum fluxes as their multiplication by the reconstructed velocity vector,

$$\boldsymbol{F}^{\mu}_{p,s} = F^{m}_{p,s}\boldsymbol{w}^{*}(\boldsymbol{x}_{p,s}), \tag{6}$$

which is a computationally inexpensive approach. Although the position of reconstruction $\boldsymbol{x}_{p,s}$ can be chosen in various ways, we use simple averages of points' coordinates corresponding to given flux (average of the old and new cell center positions for corner fluxes, and averages of edge midpoints and cell centers in the old and new cell for half-edge fluxes, see the starting locations of blue arrows in Fig. 1(b)).

To describe the entire remapping process, let us also mention the remapping of energy. The internal energy is remapped in a similar flux form used for mass,

$$\widetilde{\varepsilon}_c = \frac{1}{\widetilde{m}_c}\left(\varepsilon_c m_c + \sum_{c'\in C'(c)} F^{\varepsilon}_{c,c'}\right), \tag{7}$$

where the internal energy fluxes are constructed as the integral of the reconstructed density multiplied by the reconstructed internal energy, as proposed in [8], which guarantees bound preservation of the remapped internal energy. The nodal-based kinetic energy is remapped in a similar flux form as (5),

$$\bar{k}_p = \frac{1}{\widetilde{m}_p}\left(m_p k_p + \sum_{s\in\mathcal{S}(p)} \boldsymbol{F}^{K}_{p,s}\right), \tag{8}$$

where the specific kinetic energy is $k_p = 1/2\,|\boldsymbol{w}_p|^2$ and the fluxes are constructed as $\boldsymbol{F}^{K}_{p,s} = 1/2 F^{m}_{p,s}|\boldsymbol{w}^{*}(\boldsymbol{x}_{p,s})|^2$, so the reconstructed velocity vector is reused here. The kinetic energy is then $\bar{K}_p = \widetilde{m}_p\bar{k}_p$. On the other hand, the actual kinetic energy is obtained from the remapped velocity,

$$\widetilde{K}_p = 1/2\widetilde{m}_p|\widetilde{\boldsymbol{w}}_p|^2. \tag{9}$$

Thus, we have a kinetic energy discrepancy in each node $\delta K_p = \bar{K}_p - \widetilde{K}_p$, measuring the difference between the real and generally not conservative kinetic energy $\widetilde{K}_p$ used in the energy conservation check and the conservative remapped kinetic energy $\bar{K}_p$. To guarantee total energy conservation, this discrepancy is distributed to the remapped internal energy using the standard energy conservation fix [2]. This updated internal energy is finally used in the equation of state for the computation of the new cell pressure. For the full and detailed description of the entire remapping algorithm, see [14].

Although there are several possibilities how to remap mass, momentum and energy, we choose and use in the numerical examples in Section 6 the following approach. The cell mass is remapped by (1) with $C'(c)$ being all 9 neighbors (i.e. both edge and corner neighbors). The cell mass fluxes $F^{m}_{c,c'}$ are computed by cells' intersections. The internodal mass fluxes $F^{m}_{p,s}$ in (3) are obtained by interpolation from the cell mass fluxes $F^{m}_{c,c'}$ at 12 positions shown in Fig. 1(b) (4 at vertices of dual cells and 8 at half-edges of dual cells). The momentum remap is performed by (4) with s running again over these 12 positions and with momentum fluxes given by (6). Such approach allows to be used also for multi-material cells. The cell internal energy is remapped in a flux form (7), the nodal kinetic energy as (8). The actual kinetic energy is computed from the remapped velocity (9), and the resulting energy discrepancy is distributed to the remapped internal energy by the standard conservation fix. This approach guarantees the conservation of the total energy. The new density and the modified internal energy are finally used in the equation of state for the computation of the new cell pressure.

Another reason for using the momentum remap in the form (5) with fluxes defined by (6) is their consistency in the DeBar sense [7], which is usually understood as the condition for internal self-consistency of a particular method: for constant velocity field and arbitrary mass fluxes $F^{m}_{p,s}$, the remap has to recover the constant velocity. If we assume that all reconstructed velocities $\boldsymbol{w}^{*}$ are equal to the constant velocity field (which is a natural property of any reasonable reconstruction method), then all $\boldsymbol{w}^{*}$ and $\boldsymbol{w}_p$ can be factored out in front of the formula. The rest reduces to the nodal mass remapping formula (3), which cancels with the denominator and (5) recovers the constant velocity field exactly. Therefore, formula (5) with fluxes (6) is DeBar consistent by construction.

Our first goal in this paper is to define the symmetry preservation of velocity for symmetric flows for remapping from one polar mesh to another. The second goal is to define coordinate invariant bounds for the velocity vector and describe the approach to enforce those bounds. In the next section we define bounds for the remapped velocity vector. In Section 4 we describe low- and high-order approaches to constructing the velocity $\boldsymbol{w}^{*}(\boldsymbol{x}_{p,s})$ and discuss their properties with respect to bound and symmetry preservation on a polar grid. Then, in Section 5, following the FCR approach, the final momentum flux is constructed as a combination of corresponding low- and high-order momentum fluxes. Section 6 demonstrates how the developed remapping method works for selected numerical examples.

3. Definition of bounds

To project all velocity vectors into the direction of the velocity vector at point p, we introduce the following rotation matrix

$$\mathsf{A}_p = \frac{1}{|\boldsymbol{w}_p|}\begin{pmatrix} w_p^x & w_p^y \\ -w_p^y & w_p^x \end{pmatrix}. \tag{10}$$

This matrix transforms Cartesian components of the velocity vector, $\boldsymbol{w}_p^{xy} = (w_p^x, w_p^y)^T$ to the components $\boldsymbol{w}_p^{\xi,\eta} = (A_p \boldsymbol{w}_p^{xy})^{\xi,\eta}$, where the ξ direction coincides with the direction of vector $\boldsymbol{w}_p$ and the η direction is perpendicular to it. If the nodal velocity is close to zero, then we set A_p to identity.

As in [24–26], the constraints on the velocity in the directions (ξ, η) for point p are

$$w_p^{\xi,\min} \leqslant \widetilde{w}_p^{\xi} \leqslant w_p^{\xi,\max}, \qquad w_p^{\eta,\min} \leqslant \widetilde{w}_p^{\eta} \leqslant w_p^{\eta,\max}, \tag{11}$$

where

$$\boldsymbol{w}_p^{\xi\eta} = \mathsf{A}_p \boldsymbol{w}_p^{xy}, \qquad \boldsymbol{w}_p^{\xi\eta} = \begin{pmatrix} w_p^{\xi} \\ w_p^{\eta} \end{pmatrix}, \qquad \boldsymbol{w}_p^{xy} = \begin{pmatrix} w_p^{x} \\ w_p^{y} \end{pmatrix}$$

with

$$w_p^{\xi,\min} = \min_{q\in\{p,\mathcal{P}(p)\}} \left(\mathsf{A}_p \boldsymbol{w}_q^{xy}\right)^{\xi}, \qquad w_p^{\xi,\max} = \max_{q\in\{p,\mathcal{P}(p)\}} \left(\mathsf{A}_p \boldsymbol{w}_q^{xy}\right)^{\xi}$$

and similarly for $w_p^{\eta,\min}$ and $w_p^{\eta,\max}$.

Multiplying the velocity constraints (11) by the new mass provides equivalent momentum constraints:

$$\widetilde{m}_p w_p^{\xi,\min} \leqslant \widetilde{m}_p \widetilde{w}_p^{\xi} \leqslant \widetilde{m}_p w_p^{\xi,\max}, \tag{12a}$$

$$\widetilde{m}_p w_p^{\eta,\min} \leqslant \widetilde{m}_p \widetilde{w}_p^{\eta} \leqslant \widetilde{m}_p w_p^{\eta,\max}. \tag{12b}$$

Let us assume that the velocity field is radial and the velocities in all neighboring points q have the same magnitude. The radial directions at point p and neighboring points q with different angle coordinates are different, and therefore $|(\mathsf{A}_p \boldsymbol{w}_p)^{\xi}| > |(\mathsf{A}_p \boldsymbol{w}_q)^{\xi}|$ and $(\mathsf{A}_p \boldsymbol{w}_q)^{\eta} \neq 0$. Therefore the preservation of the local bounds in the way we have defined them does not imply the preservation of bounds for the radial component or the magnitude of the velocity for a radial flow. In other words, bounds defined in such way are not equivalent to the standard definition of bounds for velocity in 1D flow.

4. Overview of velocity reconstruction methods

In this section we describe the low- and high-order reconstruction of the velocity field. As the high-order reconstruction we use below the piecewise linear reconstruction, however higher-order reconstruction, such as the piecewise parabolic, can be used instead. We have developed remapping with piecewise parabolic reconstruction in 1D in [38] and we are working on its extension to 2D. We focus on the reconstruction of the nodal velocity field with respect to its symmetry. We define symmetry of the reconstruction only with respect to the polar mesh with radial movement of its nodes. To be symmetric, computational grid and physical quantities have to be angle independent. We assume that the polar mesh is equiangular. On this grid, even if the underlying velocity field is radial, the velocity reconstruction is piecewise linear in Cartesian coordinate system and thus not radial. We require the reconstruction (i) to be symmetric w.r.t. the line connecting the actual node to the coordinate origin and (ii) to be rotationally invariant, i.e. not to depend on the angle between this line and the coordinate axis. Note that the standard Barth–Jespersen (BJ) limited velocity reconstruction applied component by component is not symmetric in this sense. After remapping, the new nodal velocity has to be radial.

4.1. Piecewise constant reconstruction

The low-order piecewise constant velocity reconstruction uses only known nodal values of velocity. Particular node is chosen with respect to the sign of inter-nodal mass flux, which is determined by the grid movement. Suppose that we have mass flux $F_{p,q,c}^m$ from node p to q given at point $\boldsymbol{x}_{p,q,c}$ and nodal velocities $\boldsymbol{w}$. Then the low-order (L) reconstructed velocity $\boldsymbol{w}^*(\boldsymbol{x}_{p,q,c})$ has the form

$$\boldsymbol{w}^L(\boldsymbol{x}_{p,q,c}) = \begin{cases} \boldsymbol{w}_p & \text{for } F_{p,q,c}^m > 0 \\ \boldsymbol{w}_q & \text{for } F_{p,q,c}^m < 0. \end{cases} \tag{13}$$

In our particular implementation, this reconstruction is closely related to the piecewise constant one on a dual mesh. If $F_{p,q,c}^m = 0$, then the resulting momentum flux is zero and we do not need to define the velocity reconstruction in this case.

For a polar mesh with symmetric density and radial mesh motion, the only nonzero inter-nodal mass fluxes are those in the radial direction. For a given pair of nodes, velocity reconstruction is constant and therefore obviously the same for both mass fluxes in the pair, resulting in symmetric remapping.

4.2. Piecewise linear reconstruction

There are many ways how to perform a high-order velocity reconstruction, which are more or less equivalent in regions with sufficiently smooth velocity field. Our high-order (H) reconstructed velocity $\boldsymbol{w}^*(\boldsymbol{x}_{p,q,c})$ has the form

$$\boldsymbol{w}^H(\boldsymbol{x}_{p,q,c}) = \begin{cases} \boldsymbol{w}_p + (\nabla \boldsymbol{w})_p(\boldsymbol{x}_{p,q,c} - \boldsymbol{x}_p) & \text{for } F^m_{p,q,c} > 0 \\ \boldsymbol{w}_q + (\nabla \boldsymbol{w})_q(\boldsymbol{x}_{p,q,c} - \boldsymbol{x}_q) & \text{for } F^m_{p,q,c} < 0. \end{cases} \tag{14}$$

Slopes in the velocity gradient $(\nabla \boldsymbol{w})_p$ are computed as the mean values of the slopes in the neighborhood of the point, and evaluated by reducing the volume integral to the boundary integral along the eight neighboring nodes using the Green formula. For example, the slope of the first velocity component in x direction is computed as

$$\left(\frac{\delta w^x}{\delta x}\right)_p = \frac{\int_{V_p^{\text{neigh}}} (\frac{\delta w^x}{\delta x})_p \,\mathrm{d}x\,\mathrm{d}y}{\int_{V_p^{\text{neigh}}} 1 \,\mathrm{d}x\,\mathrm{d}y} = \frac{\oint_{\partial V_p^{\text{neigh}}} w^x \,\mathrm{d}y}{V_p^{\text{neigh}}}, \tag{15}$$

where

$$\oint_{\partial V_p^{\text{neigh}}} w^x \,\mathrm{d}y = \frac{1}{2} \sum_{q \in \mathcal{P}(p)} \left(w^x_q + w^x_{q-1}\right)(y_q - y_{q-1}) \tag{16}$$

and V_p^{neigh} is the volume of the integration area consisting of all cells from $\mathcal{C}(p)$, therefore $V_p^{\text{neigh}} = \sum_{c \in \mathcal{C}(p)} V_c$, with V_c being the cell volume. Here $\mathcal{P}(p)$ is a set of all points neighboring to p (both edge and corner neighbors), and we assume (without the loss of generality) an ordering which allows us to select the previous $(q-1)$ and the next $(q+1)$ point in this set in the counter-clockwise sense. The computation of these velocity slopes at the boundary nodes uses the same formulas with ghost nodes, where the velocity values at the ghost nodes are obtained from particular boundary conditions. This unlimited reconstruction as well as the resulting remapping is also symmetric (see Appendix A). However, it can produce new extrema, especially in regions near shocks.

5. Flux-corrected remap

The flux-corrected transport (FCT) approach, originally proposed by Boris and Book [4] and later enhanced by Zalesak [39], is typically used to solve advection problems and hyperbolic systems of partial differential equations, however its framework can be used also for other purposes. A comprehensive summary of FCT, its varieties and application areas, can be found in [16]. In particular, we are adapting the concept for remapping of conservative variables of fluid dynamics, which was suggested in [36,17,18], and following these papers we also adopt the term flux-corrected remapping (FCR).

The basic idea of FCT/FCR is to avoid overshoots, undershoots and oscillations (which are typical for the high-order methods) in the solution by combining some higher-order fluxes F^{H} with low-order fluxes F^{L}, so that the local bounds are preserved. In particular, the FCR flux across the interface of two cells (in our case of two dual cells) has the form

$$F^{\text{FCR}} = F^{\text{L}} + C \underbrace{\left(F^{\text{H}} - F^{\text{L}}\right)}_{dF} = C F^{\text{H}} + (1 - C) F^{\text{L}},$$

where dF is usually referred to as the antidiffusive flux and the correction factor $0 \leqslant C \leqslant 1$ controls the amount of the high-order portion of the flux used. The FCR method finds the highest C (i.e. closest to the high-order fluxes) for which the FCR fluxes preserve local bounds. To decouple the global (mesh-wide) optimization problem into a set of local problems (one per dual cell), C is computed using the worst case scenario.

5.1. Principle of symmetry preserving FCR

The momentum remap (4) cannot be performed directly in the local directions of the flow (ξ, η) defined in Section 3 because these directions are different in each dual cell. To guarantee momentum conservation, one global coordinate system must be used for all nodal momenta and momentum fluxes, such as the Cartesian system.

The remapped momentum in Cartesian coordinates is

$$\widetilde{m}_p \widetilde{\boldsymbol{w}}^{xy}_p = m_p \boldsymbol{w}^{xy}_p + \sum_{s \in \mathcal{S}(p)} \boldsymbol{F}^{\mu,\text{FCR},xy}_{p,s},$$

where the flux is a combination of the low-order and high-order fluxes:

$$\boldsymbol{F}_{p,s}^{\mu,\mathrm{FCR},xy} = \boldsymbol{F}_{p,s}^{\mu,\mathrm{L},xy} + C_{p,s}\,\boldsymbol{dF}_{p,s}^{\mu,xy}, \qquad \boldsymbol{dF}_{p,s}^{\mu,xy} = \boldsymbol{F}_{p,s}^{\mu,\mathrm{H},xy} - \boldsymbol{F}_{p,s}^{\mu,\mathrm{L},xy}.$$

If we express the momentum flux as a product of (scalar) mass flux $F_{p,s}^{m}$ and "flux velocity" $\boldsymbol{w}_{p,s}$, we have

$$\boldsymbol{F}_{p,s}^{\mu,\mathrm{L},xy} = F_{p,s}^{m}\boldsymbol{w}_{p,s}^{\mathrm{L},xy}, \qquad \boldsymbol{F}_{p,s}^{\mu,\mathrm{H},xy} = F_{p,s}^{m}\boldsymbol{w}_{p,s}^{\mathrm{H},xy}, \qquad \boldsymbol{dF}_{p,s}^{\mu,xy} = F_{p,s}^{m}\big(\boldsymbol{w}_{p,s}^{\mathrm{H},xy} - \boldsymbol{w}_{p,s}^{\mathrm{L},xy}\big),$$

which yields the FCR flux in the form

$$\boldsymbol{F}_{p,s}^{\mu,\mathrm{FCR},xy} = F_{p,s}^{m}\left[\boldsymbol{w}_{p,s}^{\mathrm{L},xy} + C_{p,s}\big(\boldsymbol{w}_{p,s}^{\mathrm{H},xy} - \boldsymbol{w}_{p,s}^{\mathrm{L},xy}\big)\right]. \tag{17}$$

Because of the choice of $C_{p,s}$ as a scalar (common for both velocity components), the remapped momentum is transferred to the (ξ,η) coordinates as

$$\mathsf{A}_p\big(\widetilde{m}_p\,\widetilde{\boldsymbol{w}}_p^{xy}\big) = \widetilde{m}_p\,\overbrace{\mathsf{A}_p\widetilde{\boldsymbol{w}}_p^{xy}}^{\widetilde{\boldsymbol{w}}_p^{\xi\eta}} = m_p\,\overbrace{\mathsf{A}_p\boldsymbol{w}_p^{xy}}^{\boldsymbol{w}_p^{\xi\eta}} + \sum_{s\in\mathcal{S}(p)} F_{p,s}^{m}\left[\mathsf{A}_p\boldsymbol{w}_{p,s}^{\mathrm{L},xy} + \mathsf{A}_p C_{p,s}\big(\boldsymbol{w}_{p,s}^{\mathrm{H},xy} - \boldsymbol{w}_{p,s}^{\mathrm{L},xy}\big)\right].$$

Therefore, the complete set of constraints (12) for dual cell p is

$$\begin{aligned}
\widetilde{m}_p\,\overbrace{\min_{q\in\{p,\mathcal{P}(p)\}}\big(\mathsf{A}_p\boldsymbol{w}_q^{xy}\big)^{\xi}}^{w_p^{\xi,\min}} &\leqslant m_p\,\overbrace{\big(\mathsf{A}_p\boldsymbol{w}_p^{xy}\big)^{\xi}}^{w_p^{\xi}} + \sum_{s\in\mathcal{S}(p)} F_{p,s}^{m}\left[\mathsf{A}_p\boldsymbol{w}_{p,s}^{\mathrm{L},xy} + C_{p,s}\mathsf{A}_p\big(\boldsymbol{w}_{p,s}^{\mathrm{H},xy} - \boldsymbol{w}_{p,s}^{\mathrm{L},xy}\big)\right]^{\xi} \\
&\leqslant \widetilde{m}_p\,\underbrace{\max_{q\in\{p,\mathcal{P}(p)\}}\big(\mathsf{A}_p\boldsymbol{w}_q^{xy}\big)^{\xi}}_{w_p^{\xi,\max}}, \qquad (18a)\\
\widetilde{m}_p\,\underbrace{\min_{q\in\{p,\mathcal{P}(p)\}}\big(\mathsf{A}_p\boldsymbol{w}_q^{xy}\big)^{\eta}}_{w_p^{\eta,\min}} &\leqslant m_p\,\underbrace{\big(\mathsf{A}_p\boldsymbol{w}_p^{xy}\big)^{\eta}}_{w_p^{\eta}} + \sum_{s\in\mathcal{S}(p)} F_{p,s}^{m}\left[\mathsf{A}_p\boldsymbol{w}_{p,s}^{\mathrm{L},xy} + C_{p,s}\mathsf{A}_p\big(\boldsymbol{w}_{p,s}^{\mathrm{H},xy} - \boldsymbol{w}_{p,s}^{\mathrm{L},xy}\big)\right]^{\eta} \\
&\leqslant \widetilde{m}_p\,\underbrace{\max_{q\in\{p,\mathcal{P}(p)\}}\big(\mathsf{A}_p\boldsymbol{w}_q^{xy}\big)^{\eta}}_{w_p^{\eta,\max}}. \qquad (18b)
\end{aligned}$$

Next task is to find the scalar flux correction factor $C_{p,s}$ for each flux $s=(q,c)$ around node p. That is, to find the maximum possible $C_{p,q,c}$ which satisfies constraints (12) and (18) for dual cell p and analogous constraints for dual cell $q\in\mathcal{P}(p)$, all with respect to cell c.

5.2. FCR correction factors

Let us consider the constraints for ξ-component of momentum in dual cell p (18a). We reorder and define available space for antidiffusive correction of momentum Q at lower and upper bound as

$$\begin{aligned}
Q_p^{\xi,\min} &= \widetilde{m}_p w_p^{\xi,\min} - \widetilde{m}_p\widetilde{w}_p^{\xi,L} = \widetilde{m}_p w_p^{\xi,\min} - m_p w_p^{\xi} - \sum_{s\in\mathcal{S}(p)} F_{p,s}^{m}\big(\mathsf{A}_p\,\boldsymbol{w}_{p,s}^{\mathrm{L},xy}\big)^{\xi} \leqslant 0,\\
Q_p^{\xi,\max} &= \widetilde{m}_p w_p^{\xi,\max} - \widetilde{m}_p\widetilde{w}_p^{\xi,L} = \widetilde{m}_p w_p^{\xi,\max} - m_p w_p^{\xi} - \sum_{s\in\mathcal{S}(p)} F_{p,s}^{m}\big(\mathsf{A}_p\boldsymbol{w}_{p,s}^{\mathrm{L},xy}\big)^{\xi} \geqslant 0,
\end{aligned}$$

where the inequalities (sign of Q) follow from the fact that the low-order momentum flux preserves local bounds by construction, which is proved in Appendix B. Now the constraint for the ξ-component becomes

$$Q_p^{\xi,\min} \leqslant \sum_{s\in\mathcal{S}(p)} C_{p,s}F_{p,s}^{m}\left[\mathsf{A}_p\big(\boldsymbol{w}_{p,s}^{\mathrm{H},xy} - \boldsymbol{w}_{p,s}^{\mathrm{L},xy}\big)\right]^{\xi} \leqslant Q_p^{\xi,\max} \tag{19}$$

and similarly for the η-component

$$Q_p^{\eta,\min} \leqslant \sum_{s\in\mathcal{S}(p)} C_{p,s}F_{p,s}^{m}\left[\mathsf{A}_p\big(\boldsymbol{w}_{p,s}^{\mathrm{H},xy} - \boldsymbol{w}_{p,s}^{\mathrm{L},xy}\big)\right]^{\eta} \leqslant Q_p^{\eta,\max}. \tag{20}$$

Let us denote

$$d\mu_{p,s}^{\xi} = F_{p,s}^{m}\left[\mathsf{A}_p\left(\boldsymbol{w}_{p,s}^{\mathrm{H},xy} - \boldsymbol{w}_{p,s}^{\mathrm{L},xy}\right)\right]^{\xi}, \tag{21}$$

$$d\mu_{p,s}^{\eta} = F_{p,s}^{m}\left[\mathsf{A}_p\left(\boldsymbol{w}_{p,s}^{\mathrm{H},xy} - \boldsymbol{w}_{p,s}^{\mathrm{L},xy}\right)\right]^{\eta} \tag{22}$$

and split the sums according to the sign of the unlimited terms:

$$Q_p^{\xi,\min} \leqslant \sum_{s;d\mu_{p,s}^{\xi}>0} C_{p,s}\, d\mu_{p,s}^{\xi} + \sum_{s;d\mu_{p,s}^{\xi}<0} C_{p,s}\, d\mu_{p,s}^{\xi} \leqslant Q_p^{\xi,\max},$$

$$Q_p^{\eta,\min} \leqslant \sum_{s;d\mu_{p,s}^{\eta}>0} C_{p,s}\, d\mu_{p,s}^{\eta} + \sum_{s;d\mu_{p,s}^{\eta}<0} C_{p,s}\, d\mu_{p,s}^{\eta} \leqslant Q_p^{\eta,\max}$$

(we omitted the formal zero contributions). Now the sufficient conditions, based on the worst case scenarios, are

$$Q_p^{\xi,\min} \leqslant \sum_{s;d\mu_{p,s}^{\xi}<0} C_{p,s}\, d\mu_{p,s}^{\xi}, \tag{23a}$$

$$Q_p^{\xi,\max} \geqslant \sum_{s;d\mu_{p,s}^{\xi}>0} C_{p,s}\, d\mu_{p,s}^{\xi}, \tag{23b}$$

$$Q_p^{\eta,\min} \leqslant \sum_{s;d\mu_{p,s}^{\eta}>0} C_{p,s}\, d\mu_{p,s}^{\eta}, \tag{23c}$$

$$Q_p^{\eta,\max} \geqslant \sum_{s;d\mu_{p,s}^{\eta}<0} C_{p,s}\, d\mu_{p,s}^{\eta}, \tag{23d}$$

where the correction factor $C_{p,s}$ will be kept between 0 (low-order flux) and 1 (high-order flux), so it will not change the sign of the sums.

Now we switch from edge-related correction factors $C_{p,s}$ to cell-related bounds D_p. Let us define $D_p^{\xi,\max}$ as

$$D_p^{\xi,\max} = Q_p^{\xi,\max} \Big/ \sum_{s;d\mu_{p,s}^{\xi}>0} d\mu_{p,s}^{\xi}.$$

Note that if $\sum_{s;d\mu_{p,s}^{\xi}>0} d\mu_{p,s}^{\xi} = 0$, then we do not need to define $D_p^{\xi,\max}$, since in this case (23b) is satisfied for any $C_{p,s} \geqslant 0$ because $Q_p^{\xi,\max} \geqslant 0$.

Clearly, if the correction factors $C_{p,s}$ are chosen so that

$$0 \leqslant C_{p,s} \leqslant D_p^{\xi,\max} \quad \text{for all } s \text{ such that } d\mu_{p,s}^{\xi} > 0,$$

then (23b) holds, since

$$Q_p^{\xi,\max} = D_p^{\xi,\max} \sum_{s;d\mu_{p,s}^{\xi}>0} d\mu_{p,s}^{\xi} = \sum_{s;d\mu_{p,s}^{\xi}>0} D_p^{\xi,\max}\, d\mu_{p,s}^{\xi} \geqslant \sum_{s;d\mu_{p,s}^{\xi}>0} C_{p,s}\, d\mu_{p,s}^{\xi}.$$

Carrying out the same considerations for the other constraints, we have four factors

$$D_p^{\xi,\min} = Q_p^{\xi,\min} \Big/ \sum_{s;d\mu_{p,s}^{\xi}<0} d\mu_{p,s}^{\xi}, \tag{24a}$$

$$D_p^{\xi,\max} = Q_p^{\xi,\max} \Big/ \sum_{s;d\mu_{p,s}^{\xi}>0} d\mu_{p,s}^{\xi}, \tag{24b}$$

$$D_p^{\eta,\min} = Q_p^{\eta,\min} \Big/ \sum_{s;d\mu_{p,s}^{\eta}<0} d\mu_{p,s}^{\eta}, \tag{24c}$$

$$D_p^{\eta,\max} = Q_p^{\eta,\max} \Big/ \sum_{s;d\mu_{p,s}^{\eta}>0} d\mu_{p,s}^{\eta} \tag{24d}$$

with the following sufficient conditions for (23a)–(23d):

$$C_{p,s} \leqslant D_p^{\xi,\min} \quad \text{for all } s \text{ such that } d\mu_{p,s}^{\xi} < 0,$$
$$C_{p,s} \leqslant D_p^{\xi,\max} \quad \text{for all } s \text{ such that } d\mu_{p,s}^{\xi} > 0,$$
$$C_{p,s} \leqslant D_p^{\eta,\min} \quad \text{for all } s \text{ such that } d\mu_{p,s}^{\eta} < 0,$$
$$C_{p,s} \leqslant D_p^{\eta,\max} \quad \text{for all } s \text{ such that } d\mu_{p,s}^{\eta} > 0.$$

Note that this approach is equivalent to the application of standard FCR on projected velocities and fluxes. Also let us stress that while evaluating these constraints, one has to keep in mind that generally $d\mu_{p,q,c}^{\xi} \neq -d\mu_{q,p,c}^{\xi}$ and $d\mu_{p,q,c}^{\eta} \neq -d\mu_{q,p,c}^{\eta}$.

We are now looking at the interface p, q, c and knowing all necessary values of A, $d\mu$, Q and D. The process of computing the correction factor $C_{p,q,c}$ goes like this:

Algorithm 1.

- Initialize $C_{p,q,c} = 1$.
- Look at the constraints from dual cell p:
  - If $d\mu_{p,q,c}^{\xi} < 0$, then set $C_{p,q,c} = \min(C_{p,q,c}, D_p^{\xi,\min})$
  - If $d\mu_{p,q,c}^{\xi} > 0$, then set $C_{p,q,c} = \min(C_{p,q,c}, D_p^{\xi,\max})$
  - If $d\mu_{p,q,c}^{\eta} < 0$, then set $C_{p,q,c} = \min(C_{p,q,c}, D_p^{\eta,\min})$
  - If $d\mu_{p,q,c}^{\eta} > 0$, then set $C_{p,q,c} = \min(C_{p,q,c}, D_p^{\eta,\max})$.
- Look at the constraints from dual cell q:
  - If $d\mu_{q,p,c}^{\xi} < 0$, then set $C_{p,q,c} = \min(C_{p,q,c}, D_q^{\xi,\min})$
  - If $d\mu_{q,p,c}^{\xi} > 0$, then set $C_{p,q,c} = \min(C_{p,q,c}, D_q^{\xi,\max})$
  - If $d\mu_{q,p,c}^{\eta} < 0$, then set $C_{p,q,c} = \min(C_{p,q,c}, D_q^{\eta,\min})$
  - If $d\mu_{q,p,c}^{\eta} > 0$, then set $C_{p,q,c} = \min(C_{p,q,c}, D_q^{\eta,\max})$.

The described approach is symmetric for radially symmetric problems on polar grids (proof is given in Appendix C) and preserves local bounds in the direction of the flow.

We want to emphasize again, that the preservation of local velocity bounds in directions ξ and η does not imply the preservation of bounds in the radial velocity component. If we have a radial velocity field with constant radial velocity component $W^{\xi} > 0$, the local bounds give us a lower bound on $w^{\xi,\min} < W^{\xi}$ as the projection of radial velocity from six neighboring nodes (located off axis ξ) to direction ξ which is strictly less than W^{ξ}. This means that the method allows small undershoots in the radial velocity component which are proportional to the angular resolution of the mesh. Because of the symmetry, the local and the radial component of the remapped velocity has the same direction, all the projections are smaller and therefore preservation of the maximum of the velocity magnitude is enforced by the method. This is why we call our method *essentially* bound preserving.

6. Numerical examples

In the first part, we use the cyclic remapping technique [28], to verify the remapping methods described above. For each test, a sequence of grids is generated, where the nodal positions gradually change from one grid to another following some prescribed motion. Given velocity and density functions are discretized on the first grid and their discrete values remapped onto the next grid, from it onto the next one, and so on until the last grid is reached. By comparing the final solution with the analytic profile of the function we can estimate the error accumulated during the entire sequence of remappings. Repeating this process with refined grids and increased number of remaps (grids) and comparing accuracy with coarse mesh results provides the experimental convergence rate of the method.

Selected tests demonstrate the symmetry of the method on a polar grid and its order of convergence. Following properties are shown in tables for the tests: L1 error of the remapped velocity

$$\mathrm{L1}^{\mathrm{error}} = \frac{\sum_p |\widetilde{\boldsymbol{w}}_p - \boldsymbol{w}_p^e| V_p}{\sum_p |\boldsymbol{w}_p^e| V_p} \tag{25}$$

(where $\widetilde{\boldsymbol{w}}_p$ is the velocity after remap at node p, $\boldsymbol{w}_p^e$ is the exact velocity point value, i.e. $\boldsymbol{w}_p^e = w_r(r_p)\binom{\cos\varphi}{\sin\varphi}$ with the velocity magnitude w_r and the polar angle φ, V_p is the volume of the dual cell around node p with radius r_p and the summation goes over all nodes of the mesh), ratio of the errors for the finer grid, next the remapped velocity violation of the global minimum/maximum initial bounds followed by the violation of symmetry. First of them is the deviation of remapped velocity magnitude with respect to the mean values along the angular direction $\mathrm{NS}_{\mathrm{magnitude}}$. The L1 norm of angular remapped velocity component $\mathrm{NS}_{\mathrm{direction}}$ is presented in the last column. For a polar mesh with radial velocity, both

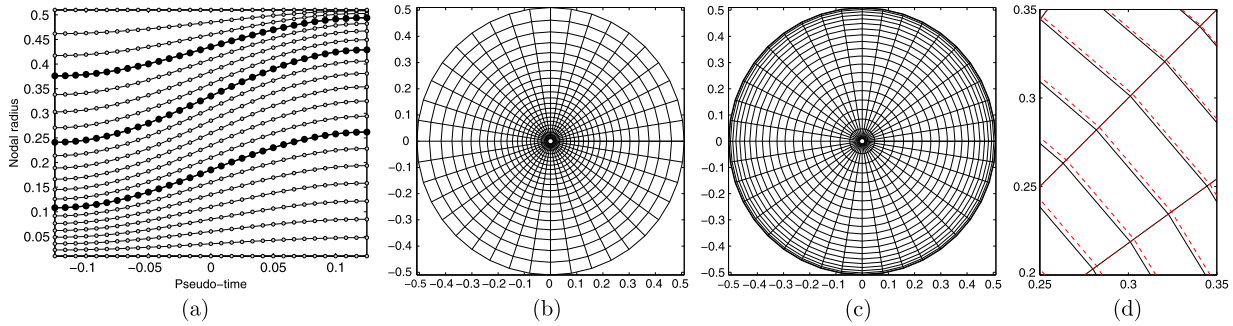


Fig. 2. Example of nodal motion in radial direction (a) and initial (b) and final (c) polar grids for cyclic remapping. Panel (d) shows a zoom of two selected consecutive grids – black for old and dashed red for new grid.

of them should be zero to preserve symmetry. Further, Cartesian grids were used to assess the robustness of the method on non-polar grids. Finally, a full hydrodynamic test is presented to demonstrate the applicability of the developed method in the framework of a staggered ALE code.

In all our numerical examples, inter-cell mass fluxes in (1) were obtained by exact integration of Barth–Jespersen (BJ) limited piecewise linear density reconstruction and inter-nodal mass fluxes were simply interpolated from the inter-cell fluxes by averaging [32].

Let us start with a polar grid. To preserve symmetry, no motion in the angular direction and smooth motion in the radial direction was used. The radial motion of a particular grid is given by

$$r(\chi,t)=\left[\left(1-\alpha(t)\right)\chi+\alpha(t)\chi^3\right](r_{\max}-r_{\min})+r_{\min},\quad \alpha(t)=\frac{\sin(4\pi t)}{2},$$
$$0\leqslant\chi\leqslant 1,\ -1/8=t^0\leqslant t\leqslant t^K=1/8. \tag{26}$$

The symbol χ represents the initial relative nodal position in the radial direction, so for the i-th node in the radial direction, it corresponds to $\chi_i=i/I$, where I is the total number of nodes in the radial direction and $i\in\langle 0,I\rangle$. The mesh is equiangular and has J cells in angular direction, $j\in\langle 0,J\rangle$, so that the polar coordinates of node (i,j) are $(r(\chi_i,t),2\pi j/J)$. Let us stress, that t is pseudo-time, i.e. each of its discrete values t^k represents a particular mesh in the sequence – there is no real "motion" of the mesh in time. The k-th mesh is constructed for the pseudo-time $t^k=k(t^k-t^0)/K+t^0$, where K represents the number of new computational meshes (and therefore the number of remapping steps) and k goes from 0 (initial mesh) to K (final mesh). Compared to [28], we use different values of t^0 and t^K to avoid superconvergence due to error cancellation caused by the symmetric change of the computational mesh to and from the point of maximum stretch. We use the interval in radial direction $(r_{\min},r_{\max})=(0.01,0.51)$, 20 resp. 40 cells in radial and 40 resp. 80 cells in angular direction corresponding to 60 resp. 120 remaps. Such sequence of grids is illustrated in Fig. 2. Plot (a) shows the evolution of the radial position of the nodes in pseudo-time. This evolution is highlighted for three selected nodes. The initial and final coarse grids for the cyclic remap are shown in Fig. 2(b), (c). One layer of rigid ghost cells inside and outside the ring is added to have simple boundary conditions for remapping.

6.1. Constant density and piecewise constant radial velocity on a polar grid

As the functions being remapped we use here the constant density $\rho=1$ and piece-wise constant radial velocity $w_r=3$ for $r\leqslant 0.25$, $w_r=1$ for $r>0.25$. Results in Table 2 and Fig. 3 reveal non-symmetry of the classical extension of scalar FCR (FCRxy; called component FCR, which corresponds to the presented method with A_p equal to the identity matrix and independent $C_{p,s}$ in (18a) and (18b) for the x- and y-velocity components) and compare the presented symmetric FCR (FCRs) to the low-order remap, resp. high-order one with piecewise linear reconstruction. Note that the undershoot in velocity magnitude for FCRs in Table 2 and Fig. 3 is due to our definition of bounds (velocity projection in neighboring node in the angular direction is smaller than its magnitude). For symmetric FCR the undershoot is reduced with increasing resolution much faster than the undershoot for component FCR, which is very close to the unlimited. The ratio of the undershoot decrease (between coarse and twice finer mesh) is about 1.6 for the symmetric FCR (so close to the convergence ratio of errors) and 1.2 for the component FCR and the unlimited. The overshoots remain approximately the same for FCRxy and the unlimited methods. Computational cost comparison of the cyclic remap for selected methods is presented in Table 1. Performance of the component- and the symmetric FCR is almost the same and both methods are not much more expensive than the BJ-limited method, which is about 1/6 slower than unlimited one.

6.2. Constant density and smooth radial velocity on a polar grid

Table 3 presents the results for constant density $\rho=1$ and smooth radial velocity $w_r(r)=1-\mathrm{e}^{-(5r)^2}$. We can see the second order of convergence for the unlimited and both FCR high-order methods.

Table 1
Ratio of the processor time for several grid resolutions and the processor time of unlimited method on the finest mesh for the constant density and piecewise constant radial velocity test.

Method	40×20	80×40	160×80	320×160
Low-order	0.19%	1.45%	11.56%	92.66%
Unlimited	0.20%	1.59%	12.55%	100.00%
BJ-limited	0.25%	1.88%	14.75%	116.71%
Component FCR	0.25%	1.93%	15.17%	120.95%
Symmetric FCR	0.25%	1.96%	15.44%	122.68%

Table 2
Convergence, errors, bounds and symmetry violations in velocity for the cyclic remap on a polar grid for a function with a jump in radial velocity.

Method	Grid	$\frac{L1^{error}}{L1^{error}_{finer}}$	$L1^{error}$	Undershoot	Overshoot	$NS_{magnitude}$	$NS_{direction}$
Low-order	40×20		1.24e−01	1.67e−15	0.00e+00	5.68e−16	1.85e−16
	80×40	1.46	8.52e−02	2.55e−15	0.00e+00	1.14e−15	2.62e−16
	160×80	1.39	6.13e−02	3.77e−15	9.77e−15	2.08e−15	3.00e−16
	320×160	1.42	4.31e−02	6.00e−15	1.51e−14	3.49e−15	3.13e−16
Unlimited	40×20		7.04e−02	9.93e−02	1.25e−01	5.36e−16	3.63e−16
	80×40	1.72	4.08e−02	7.93e−02	1.25e−01	1.06e−15	6.18e−16
	160×80	1.69	2.41e−02	7.00e−02	1.40e−01	2.18e−15	9.08e−16
	320×160	1.57	1.53e−02	5.75e−02	1.64e−01	4.09e−15	1.41e−15
Component FCR	40×20		6.90e−02	1.03e−01	1.25e−01	4.99e−03	4.70e−04
	80×40	1.77	3.90e−02	8.27e−02	1.25e−01	3.17e−03	3.09e−04
	160×80	1.68	2.32e−02	7.05e−02	1.40e−01	2.16e−03	2.09e−04
	320×160	1.60	1.45e−02	5.75e−02	1.64e−01	1.44e−03	1.50e−04
Symmetric FCR	40×20		6.30e−02	8.15e−02	2.04e−14	1.60e−15	3.54e−16
	80×40	1.89	3.34e−02	4.94e−02	5.28e−14	3.88e−15	3.83e−16
	160×80	1.65	2.02e−02	3.42e−02	1.68e−12	5.64e−14	4.36e−16
	320×160	1.68	1.20e−02	2.17e−02	4.71e−12	1.08e−13	4.70e−16

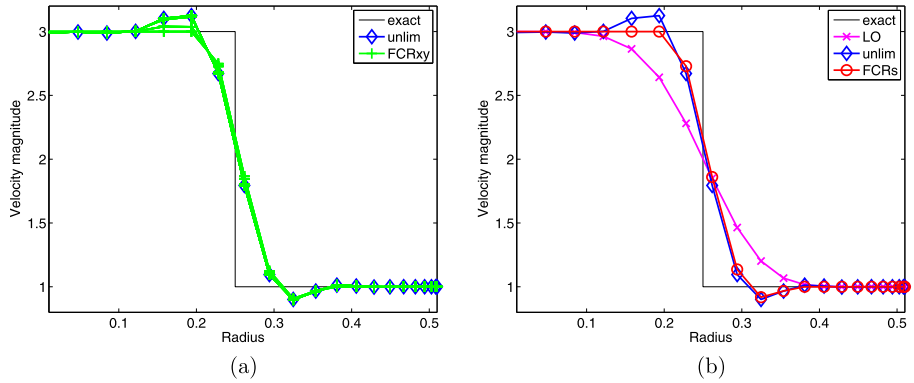


Fig. 3. Piecewise constant radial velocity problem on the coarse polar 40×20 mesh. Plots of velocity magnitude versus radius over all lines with constant polar angle containing mesh nodes (see Fig. 2(a,b)) for unlimited piecewise linear (blue diamonds) and component FCR (green plus signs) method (a), resp. for the low-order (magenta crosses), unlimited (blue diamonds), and symmetric FCR (red circles) method (b).

Table 3
Convergence, errors, bounds and symmetry violations in velocity for the cyclic remap on a polar grid for a function with smooth radial velocity. The bounds and symmetry violations (in the last four columns) are presented for the finer 80×40 mesh (for the coarser mesh these are approximately the same).

Method	$L1^{error}_{40\times20}/L1^{error}_{80\times40}$	$L1^{error}_{80\times40}$	Undershoot	Overshoot	$NS_{magnitude}$	$NS_{direction}$
Low-order	1.96	5.29e−03	0.00e+00	0.00e+00	1.62e−15	2.90e−16
Unlimited	4.10	1.21e−04	0.00e+00	0.00e+00	1.51e−15	8.90e−16
Component FCR	4.10	1.21e−04	0.00e+00	0.00e+00	1.59e−10	2.80e−13
Symmetric FCR	4.10	1.21e−04	0.00e+00	0.00e+00	5.44e−15	9.23e−16

Table 4
Convergence, errors, bounds and symmetry violations in velocity for the cyclic remap on a polar grid for the Sedov-like problem. $L1^{+}$, resp. $L1^{-}$ corresponds to (25), where summation in the numerator goes only over the nodes where $|\tilde{\boldsymbol{w}}_p| > |\boldsymbol{w}^e_p|$, resp. $|\tilde{\boldsymbol{w}}_p| < |\boldsymbol{w}^e_p|$. The bounds and symmetry violations (in the last three columns) are presented for the finer 80×40 mesh.

Method	$\frac{L1^{error}_{40\times 20}}{L1^{error}_{80\times 40}}$	$L1^{error}_{80\times 40}$	$L1^{+}_{80\times 40}$	$L1^{-}_{80\times 40}$	Overshoot	$NS_{magnitude}$	$NS_{direction}$
Low-order	1.37	3.57e−01	2.26e−01	1.31e−01	0.00e+00	4.94e−15	4.44e−16
Unlimited	1.87	1.43e−01	8.04e−02	6.28e−02	1.04e−03	6.16e−15	1.74e−15
Component FCR	1.92	1.20e−01	5.29e−02	6.71e−02	0.00e+00	1.04e−02	1.35e−03
Symmetric FCR	2.02	1.04e−01	4.38e−02	6.03e−02	0.00e+00	6.49e−15	1.56e−15

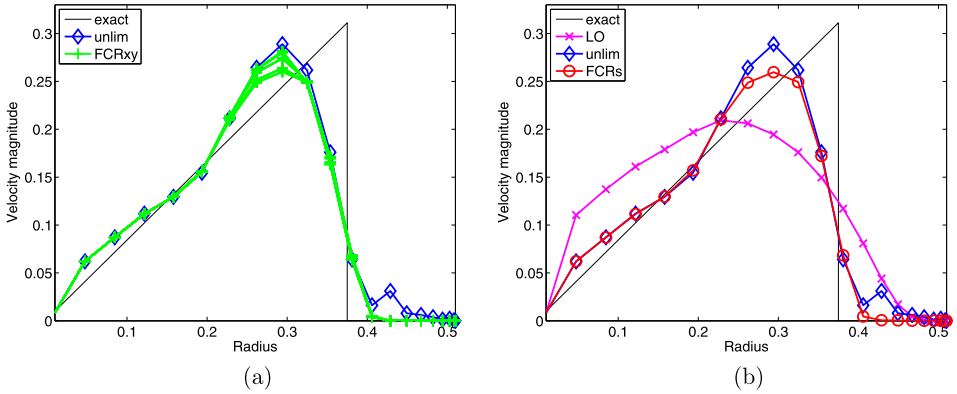


Fig. 4. Sedov-like problem on the coarse 20×40 polar grid. Plots of velocity magnitude versus radius on all mesh lines with constant polar angle for unlimited piecewise linear (blue diamonds) and component FCR (green plus signs) method (a), resp. for the low-order (magenta crosses), unlimited (blue diamonds), and symmetric FCR (red circles) method (b).

6.3. Sedov-like problem on a polar grid

This remapping test profiles, which are close to the solution of Sedov problem [17] at time $t = 1$, are

$$\rho(r) = \begin{cases} 6(\frac{r}{0.375})^8 & \text{for } r < 0.375, \\ 1 & \text{for } r \geqslant 0.375, \end{cases} \qquad w_r(r) = \begin{cases} 0.83r & \text{for } r < 0.375, \\ 0 & \text{for } r \geqslant 0.375, \end{cases} \tag{27}$$

where $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$ and $x_0 = y_0 = 0$. Results for the methods are presented in Table 4 and Fig. 4. Both FCR methods remove the overshoot produced by the unlimited approach, however standard component based FCR in x and y violates symmetry in both the magnitude and direction of the remapped velocity, see also Fig. 4(a). In the fourth and the fifth column of Table 4, the error of the methods is split to the positive and negative deviation from the exact solution, so $L1^{error}_{80\times 40} = L1^{+}_{80\times 40} + L1^{-}_{80\times 40}$.

6.4. Smooth velocity and density on a Cartesian grid

In the following tests, the mesh moves similarly to the cyclic remapping example from [17]. However, we present results in half of the final time (compared to the original paper). This pseudo-time corresponds to the maximal deformation of the grid. Our selection was motivated by the fact that errors may cancel during backward mesh movement [38]. Cartesian grid with 64×64 resp. 128×128 initially equally spaced cells with 160 resp. 320 remaps is used. Grid movement is given by

$$\begin{aligned} x(\xi, \eta, t) &= \xi + t \sin(2\pi\xi) \sin(2\pi\eta), \\ y(\xi, \eta, t) &= \eta + t \sin(2\pi\xi) \sin(2\pi\eta), \end{aligned} \qquad 0 \leqslant \xi \leqslant 1,\ 0 \leqslant \eta \leqslant 1,\ 0 \leqslant t \leqslant 0.5. \tag{28}$$

Examples of the initial and final grid are in Fig. 5.

Smooth density and the radial component of velocity profiles $\rho(r) = 0.1 + e^{-(5r)^2}$ and $w_r(r) = 1 - e^{-(5r)^2}$ are used here. Results are shown in Table 5 and Fig. 6. The low-order method is first order accurate and all other methods are of second order and give exactly the same results. So both component and symmetric FCR results are the same as the unlimited result.

6.5. Sedov-like problem on a Cartesian grid

This test is taken from [17], but as explained above, results at half time compared to the original paper are shown. The density and velocity profiles are the same as in (27) for polar grid, now with $x_0 = 0.5$ and $y_0 = 0.5$. The results are

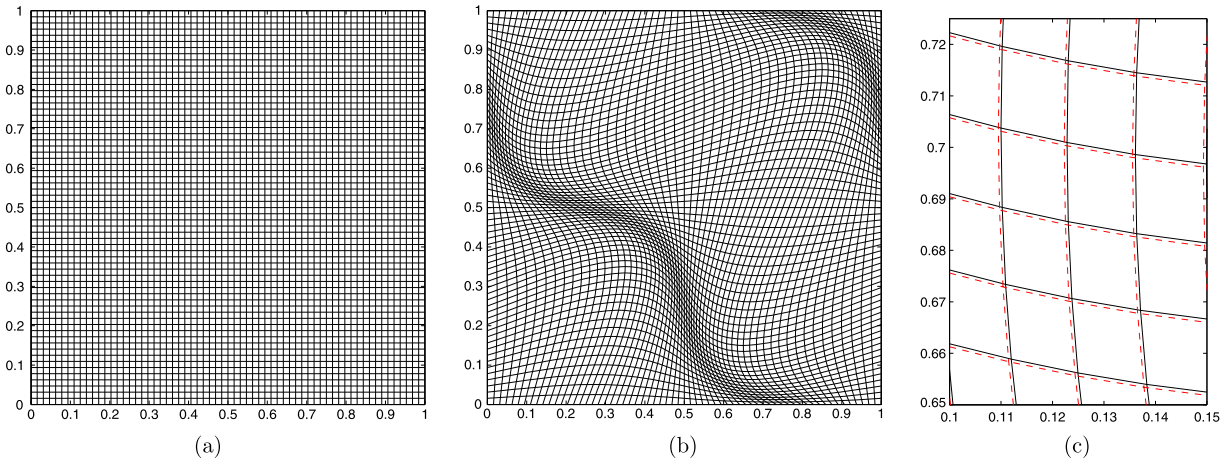


Fig. 5. Example of initial (a) and final (b) Cartesian grids for cyclic remapping. Panel (c) shows a zoom of two selected consecutive grids – black solid line for the old grid and dashed red line for the new grid.

Table 5
Convergence, errors, bounds and symmetry violations in velocity for the cyclic remap on a Cartesian grid for a function with smooth radial velocity. The bounds and symmetry violations (in the last four columns) are presented for the finer 128×128 mesh.

Method	$L1^{error}_{64\times64}/L1^{error}_{128\times128}$	$L1^{error}_{128\times128}$	Undershoot	Overshoot	$NS_{direction}$
Low-order	2.05	5.63e−03	0.00e+00	0.00e+00	3.33e−03
Unlimited	4.04	2.70e−04	0.00e+00	0.00e+00	2.19e−04
Component FCR	4.04	2.70e−04	0.00e+00	0.00e+00	2.19e−04
Symmetric FCR	4.04	2.70e−04	0.00e+00	0.00e+00	2.19e−04

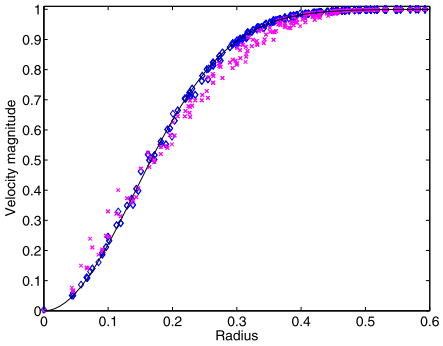


Fig. 6. Scatter plot of remapped velocity magnitude dependence on radius for the cyclic remap of smooth velocity on Cartesian a grid. Low-order (magenta cross) and unlimited piecewise linear (blue diamonds) methods are shown. FCRxy and FCRs results are the same as the unlimited result. Exact solution is shown by the black curve.

presented in Table 6 and Fig. 7. The component FCR produces an overshoot, which is comparable with the overshoot of the unlimited approach. On the contrary, the symmetric FCR keeps remapped velocity in bounds. The symmetry (measured by $NS_{direction}$ in Table 6) for symmetric FCR is slightly worse than for component FCR due to stronger limiting (tighter bounds) of symmetric FCR.

The contours of remapped velocity magnitude for selected methods are shown in Fig. 8. In this case the low-order method is much less symmetric than all the high-order ones.

6.6. Full hydrodynamic test – polar Sedov problem

In previous sections, static tests of the velocity remapping methods were performed without any Lagrangian step to emphasize their features. Here, we demonstrate the properties of the new method in a full hydrodynamic example – Sedov point explosion test [34].

The computational domain $(r, \varphi) \in (0.01, 1.2) \times (0, 2\pi)$ is covered by an equidistant and equiangular 100×40 polar mesh. At the initial time $t = 0$, the fluid (ideal gas with $\gamma = 1.4$) is static everywhere ($\boldsymbol{w} = \boldsymbol{0}$) and has uniform density $\rho = 1$. The specific internal energy is $\varepsilon = 10^{-8}$ everywhere except the innermost layer of cells at the center of the domain,

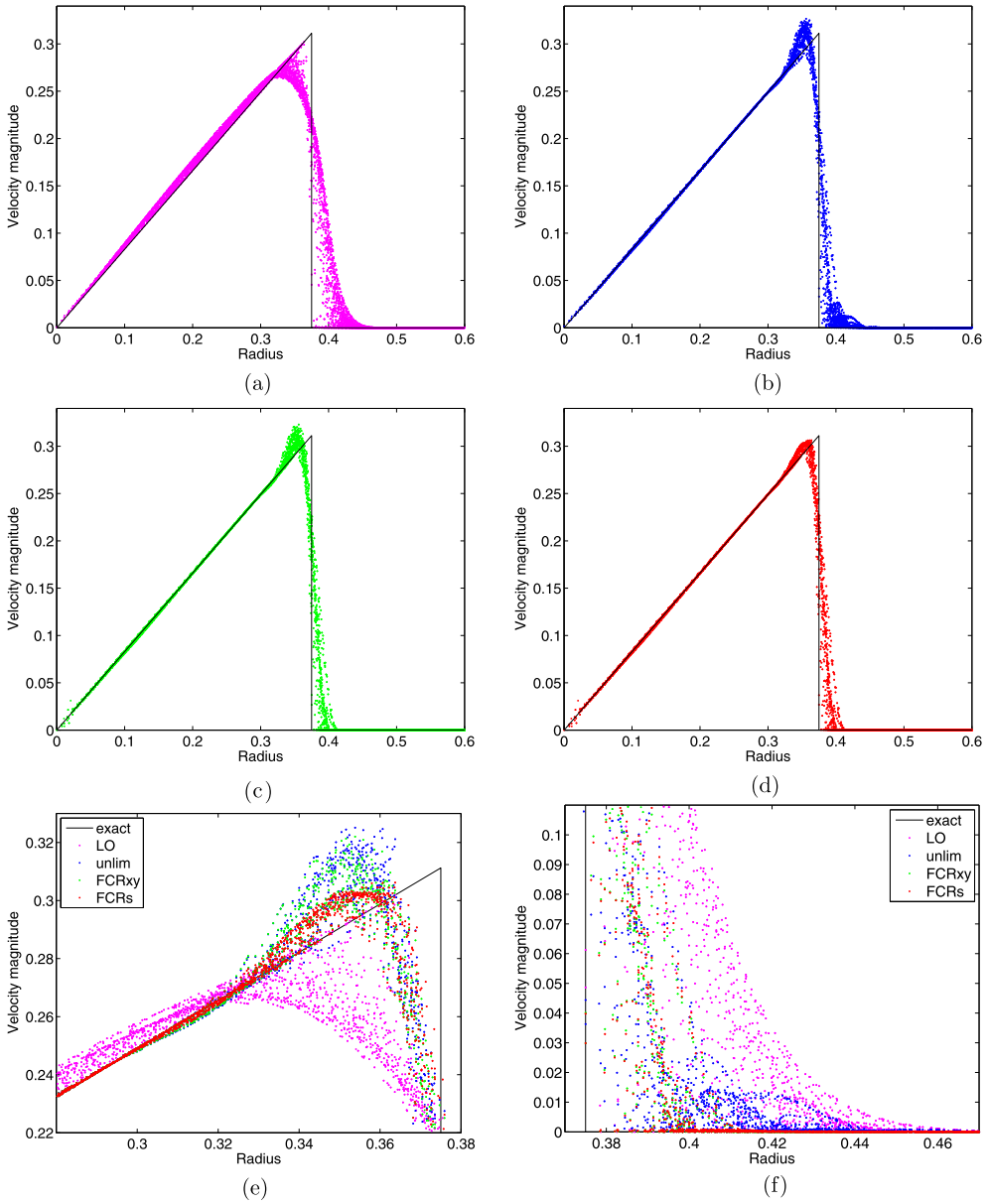


Fig. 7. Scatter plot of velocity magnitude versus radius for the Sedov-like problem on the coarser 128×128 Cartesian grid. Low-order (a), unlimited (b), component FCR (c) and symmetric FCR (d). The radius axis is limited to interval $r \in (0, 0.6)$ as nothing interesting can be seen for radius greater than 0.6. Zoom on the top (e) and bottom (f) of the discontinuity.

where $\varepsilon = 821.105$, which corresponds to the amount of energy in the definition of the Sedov problem presented in [33], adapted for given polar mesh. The high-energy cells represent point explosion generating a circular shock wave propagating from the center. At the final time $t = 1$, the shock wave is located at the radius $r = 1$.

This simulation has been run in the framework of a staggered ALE code employing the mimetic discretization of the Euler equations as presented in [5], and the edge artificial viscosity model with the viscosity limiter from [6]. For remapping, the flux-based method described in [14,12] is used – all quantities are remapped in flux form, the remap of fluid momentum is done by methods described above. To enhance the effects of remapping, this simulation has been run in the Eulerian regime, in which all fluid quantities are remapped back to the initial computational mesh after each Lagrangian step. We present the results of four different runs using the same momentum remapping methods as in the previous tests – low-order, unlimited, FCR in (x, y) directions, and the new symmetric FCR approach.

The velocity magnitude distributions at the final time moment are shown in Fig. 9 for each remapping method. Fig. 10 shows the final velocity distribution in the entire domain as a function of radius. For this particular problem, we can see only

Table 6
Convergence, errors and bounds violations in velocity for the cyclic remap on a Cartesian grid for the Sedov-like problem.

Method	Grid	$\frac{L1^{error}}{L1^{error}_{finer}}$	$L1^{error}$	Overshoot	$NS_{direction}$
Low-order	16×16		5.57e−01	0.00e+00	7.88e−02
	32×32	1.54	3.63e−01	0.00e+00	3.07e−02
	64×64	1.41	2.58e−01	0.00e+00	1.47e−02
	128×128	1.44	1.79e−01	0.00e+00	7.45e−03
Unlimited	16×16		5.04e−01	0.00e+00	5.22e−02
	32×32	2.02	2.49e−01	0.00e+00	1.47e−02
	64×64	1.71	1.46e−01	6.73e−03	4.57e−03
	128×128	1.70	8.58e−02	1.39e−02	1.50e−03
Component FCR	16×16		4.77e−01	0.00e+00	5.05e−02
	32×32	2.04	2.34e−01	0.00e+00	1.40e−02
	64×64	1.73	1.36e−01	0.00e+00	4.42e−03
	128×128	1.72	7.90e−02	1.08e−02	1.54e−03
Symmetric FCR	16×16		4.76e−01	0.00e+00	5.26e−02
	32×32	2.05	2.33e−01	0.00e+00	1.45e−02
	64×64	1.74	1.34e−01	0.00e+00	4.55e−03
	128×128	1.73	7.71e−02	0.00e+00	1.55e−03

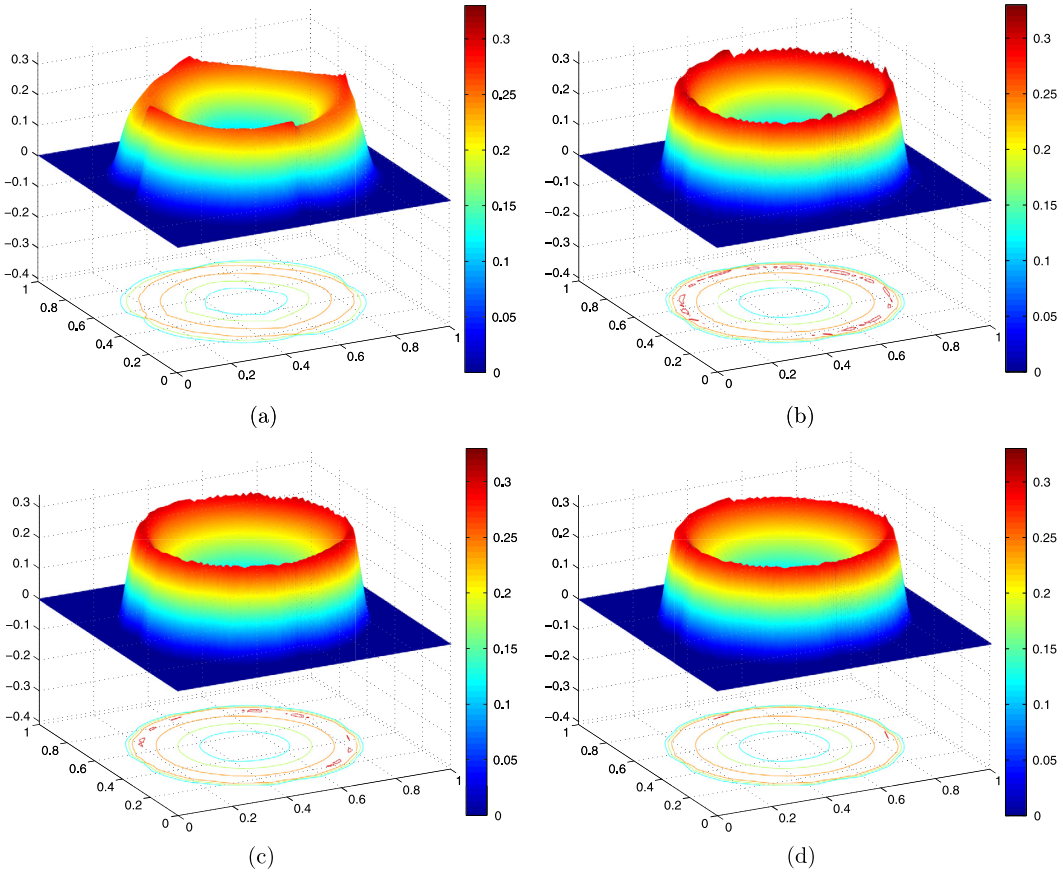


Fig. 8. Velocity magnitude for the Sedov-like problem on a Cartesian grid. Low-order (a), unlimited (b), component FCR (c) and symmetric FCR (d) on the coarser 64×64 Cartesian mesh.

a small difference between the low- and high-order approaches at the maximum velocity. The FCR by components causes a severe symmetry violation at the domain center, rendering the mesh useless for realistic simulations. After switching to the new symmetric FCR approach, the solution is perfectly symmetric and the velocity profile is very close to the high-order approach.

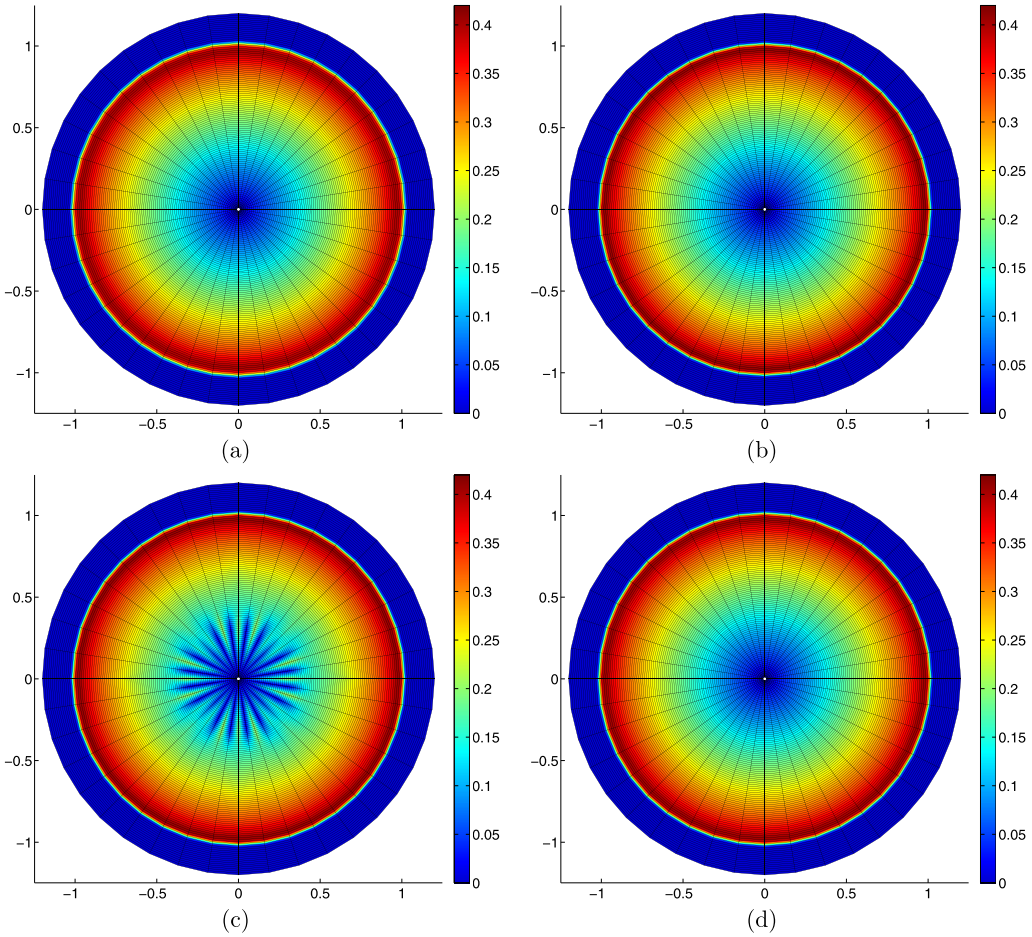


Fig. 9. Velocity magnitude distribution for the Sedov full hydrodynamic test at time $t = 1$ using the low-order (a), unlimited (b), component FCR (c), and symmetric FCR (d) method for velocity reconstruction.

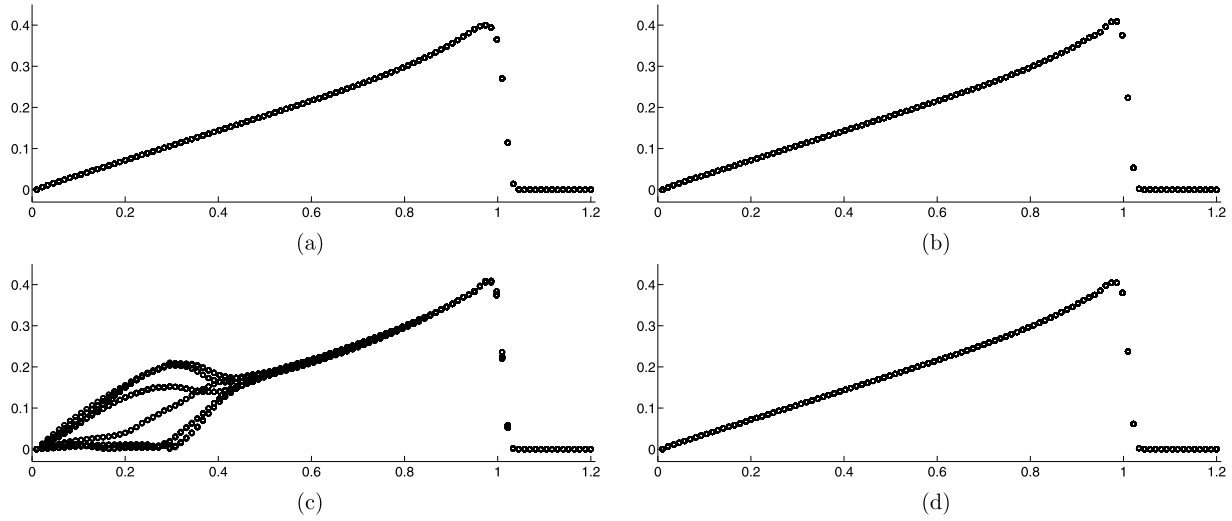


Fig. 10. Scatter plot of velocity magnitude for the Sedov full hydrodynamic test at time $t = 1$ using the low-order (a), unlimited (b), component FCR (c), and symmetric FCR (d) method for velocity reconstruction.

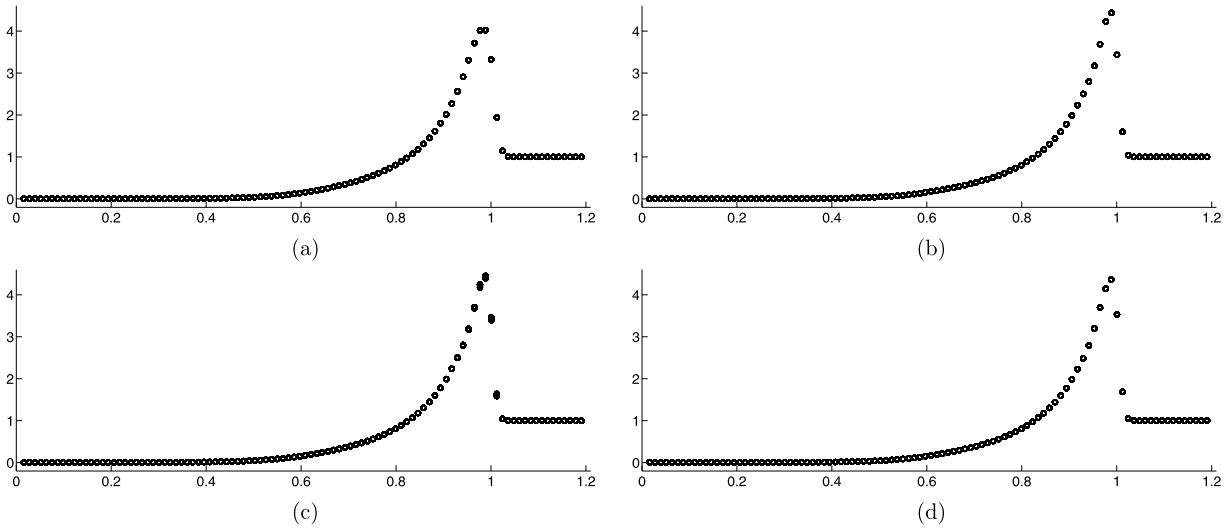


Fig. 11. Scatter plot of fluid density for the Sedov full hydrodynamic test at time $t = 1$ using the low-order (a), unlimited (b), component FCR (c), and symmetric FCR (d) method for velocity reconstruction.

In Fig. 11, we can see the final density distribution in the entire domain for the same runs. The differences between low- and high-order velocity reconstruction is more apparent here. We can observe blurring of the density profile for the component FCR approach due to velocity non-symmetry, which affects all fluid quantities. The new method produces a perfectly symmetric solution.

7. Conclusions

A new flux-corrected momentum remapping method has been developed. The method preserves local bounds in velocity components in directions parallel and orthogonal to the local velocity. It however allows a small undershoot in the velocity magnitude. The method preserves polar symmetry of the velocity field for remap from an equiangular polar mesh to another polar mesh. Performance of the method is demonstrated on selected cyclic remapping tests and a full hydrodynamic example.

Acknowledgements

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 No. DE-AC52-06NA25396. The authors gratefully acknowledge the partial support of the US Department of Energy Office of Science Advanced Scientific Computing Research (ASCR) Program in Applied Mathematics Research and the partial support of the US Department of Energy National Nuclear Security Administration Advanced Simulation and Computing (ASC) Program. This research was supported in parts by the Czech Technical University grant SGS13/220/OHK4/3T/14, the Czech Science Foundation project P201/12/P554 and RVO: 68407700. We thank both anonymous referees for their useful comments.

Appendix A. Proof of symmetry for piecewise linear reconstruction

For a radial velocity and equiangular polar grid, derivative approximations can be expressed as follows. Let us denote $\Delta w_r = w_r(r_3) - w_r(r_1)$, $\Delta r = r_3 - r_1$ and $\bar{r} = \frac{r_3+r_1}{2}$, where r_1, r_2 and r_3 correspond to nodal positions according to Fig. A.12 and w_r stands for the radial component of nodal velocity. The approximate derivatives given by (15) are

$$\frac{\delta w^x}{\delta x} = \frac{1}{2}\left(\frac{\Delta w_r}{\Delta r}\frac{r_1}{\bar{r}} + \frac{w_r(r_3)}{\bar{r}}\right) + \left(\cos^2\varphi - \frac{1}{2}\right)\left(\frac{\Delta w_r}{\Delta r}\frac{r_2}{\bar{r}} - \frac{w_r(r_2)}{\bar{r}}\right)\cos\Delta\varphi, \tag{A.1}$$

$$\frac{\delta w^y}{\delta y} = \frac{1}{2}\left(\frac{\Delta w_r}{\Delta r}\frac{r_1}{\bar{r}} + \frac{w_r(r_3)}{\bar{r}}\right) - \left(\cos^2\varphi - \frac{1}{2}\right)\left(\frac{\Delta w_r}{\Delta r}\frac{r_2}{\bar{r}} - \frac{w_r(r_2)}{\bar{r}}\right)\cos\Delta\varphi, \tag{A.2}$$

$$\frac{\delta w^x}{\delta y} = \frac{\delta w^y}{\delta x} = \left(\frac{\Delta w_r}{\Delta r}\frac{r_2}{\bar{r}} - \frac{w_r(r_2)}{\bar{r}}\right)\cos\Delta\varphi\sin\varphi\cos\varphi. \tag{A.3}$$

For $r_1 = r_2 = r_3$ and $\Delta\varphi = 0$ this corresponds to the following analytical formulas. Knowing $\frac{\partial}{\partial r} = \frac{\partial x}{\partial r}\frac{\partial}{\partial x} + \frac{\partial y}{\partial r}\frac{\partial}{\partial y}$ and $\frac{\partial}{\partial \varphi} = \frac{\partial x}{\partial \varphi}\frac{\partial}{\partial x} + \frac{\partial y}{\partial \varphi}\frac{\partial}{\partial y}$, we have

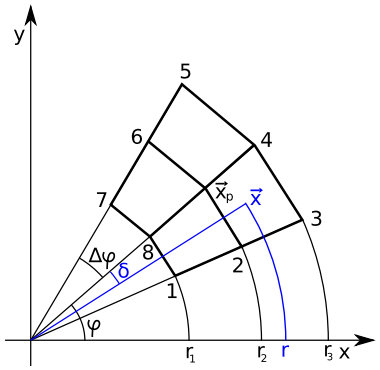


Fig. A.12. Notation of nodes and angles. Node p is located in the center of the grid and it is surrounded by 4 computational cells (thick black line).

$$\frac{\partial}{\partial x} = \cos\varphi \frac{\partial}{\partial r} - \frac{\sin\varphi}{r}\frac{\partial}{\partial \varphi} \quad \Rightarrow \quad \begin{aligned} \frac{\partial w^x}{\partial x} &= \frac{\partial w_r}{\partial r}\cos^2\varphi + \frac{w_r}{r}\sin^2\varphi \\ \frac{\partial w^y}{\partial x} &= \left(\frac{\partial w_r}{\partial r} - \frac{w_r}{r}\right)\sin\varphi\cos\varphi, \end{aligned}$$

$$\frac{\partial}{\partial y} = \sin\varphi \frac{\partial}{\partial r} + \frac{\cos\varphi}{r}\frac{\partial}{\partial \varphi} \quad \Rightarrow \quad \begin{aligned} \frac{\partial w^y}{\partial y} &= \frac{\partial w_r}{\partial r}\sin^2\varphi + \frac{w_r}{r}\cos^2\varphi \\ \frac{\partial w^x}{\partial y} &= \left(\frac{\partial w_r}{\partial r} - \frac{w_r}{r}\right)\sin\varphi\cos\varphi. \end{aligned}$$

To prove symmetry of the linear part of reconstruction (14), we simply rotate the high-order (H) term $(\nabla \boldsymbol{w})_p(\boldsymbol{x} - \boldsymbol{x}_p)$ clockwise by angle φ, using the rotation matrix $\mathsf{R}^+ = \left(\begin{smallmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{smallmatrix}\right)$. After some algebra, we get

$$\mathsf{R}^+(\nabla \boldsymbol{w})_p(\boldsymbol{x} - \boldsymbol{x}_p) = \begin{pmatrix} \left[\frac{1}{2}\left(\frac{\Delta w_r}{\Delta r}\frac{r_2}{\tilde{r}} - \frac{w_r(r_2)}{\tilde{r}}\right)\cos\Delta\varphi + \frac{1}{2}\left(\frac{\Delta w_r}{\Delta r}\frac{r_1}{\tilde{r}} + \frac{w_r(r_3)}{\tilde{r}}\right)\right](r\cos\delta - r_2) \\ \left[-\frac{1}{2}\left(\frac{\Delta w_r}{\Delta r}\frac{r_2}{\tilde{r}} - \frac{w_r(r_2)}{\tilde{r}}\right)\cos\Delta\varphi + \frac{1}{2}\left(\frac{\Delta w_r}{\Delta r}\frac{r_1}{\tilde{r}} + \frac{w_r(r_3)}{\tilde{r}}\right)\right] r\sin\delta \end{pmatrix}. \tag{A.4}$$

To make this expression more clear, we can introduce local cell coordinates $\boldsymbol{\xi} = \left(\begin{smallmatrix} \xi \\ \eta \end{smallmatrix}\right)$. After the rotation, $\varphi = 0$ and therefore we can rewrite

$$\mathsf{R}^+(\nabla \boldsymbol{w})_p(\boldsymbol{x} - \boldsymbol{x}_p) = (\nabla \boldsymbol{w})_p^{\boldsymbol{\xi}}(\boldsymbol{\xi} - \boldsymbol{\xi}_p) = \begin{pmatrix} \alpha\cos\Delta\varphi + \beta & 0 \\ 0 & -\alpha\cos\Delta\varphi + \beta \end{pmatrix}_p \begin{pmatrix} \xi - \xi_p \\ \eta \end{pmatrix} \tag{A.5}$$

with

$$\alpha = \frac{1}{2}\left(\frac{\Delta w_r}{\Delta r}\frac{r_2}{\tilde{r}} - \frac{w_r(r_2)}{\tilde{r}}\right), \qquad \beta = \frac{1}{2}\left(\frac{\Delta w_r}{\Delta r}\frac{r_1}{\tilde{r}} + \frac{w_r(r_3)}{\tilde{r}}\right),$$

where the first component, $\xi = r\cos\delta$, represents the radial direction and the second, $\eta = r\sin\delta$, the angular direction. For the special case of a radial flow on a polar mesh, the directions ξ and η correspond to the local flow directions introduced in (10) and indicated by superscripts $^{\xi}$ and $^{\eta}$ throughout Section 5.

Formula (A.5) proves that the unlimited high-order term is symmetric. We started with reconstruction for a general angle φ and obtained a result which is independent of it. Symmetry with respect to the cell axis is also guaranteed. Next point is that for this special cell orientation aligned with axes, the resulting velocity gradient has to have only diagonal components for any radial velocity field. This is in agreement with the following simple consideration. To preserve symmetry, angular velocity component reconstruction plane is fixed to cell axis. On the contrary, reconstruction plane for the radial velocity component has to be constant in direction perpendicular to the axis.

Appendix B. Proof of velocity bounds preservation for low-order remap

Let us denote

$$\boldsymbol{w}_p^{\xi\eta,\min} = \begin{pmatrix} w_p^{\xi,\min} \\ w_p^{\eta,\min} \end{pmatrix}, \qquad w_p^{\xi,\min} = \min_{q\in\{p,\mathcal{P}(p)\}} \left(\mathsf{A}_p\, \boldsymbol{w}_q^{xy}\right)^{\xi}, \qquad w_p^{\eta,\min} = \min_{q\in\{p,\mathcal{P}(p)\}} \left(\mathsf{A}_p \boldsymbol{w}_q^{xy}\right)^{\eta}.$$

Because A_p is a constant matrix for a particular dual cell, and $F_{p,s}^m$ is a scalar, we can write

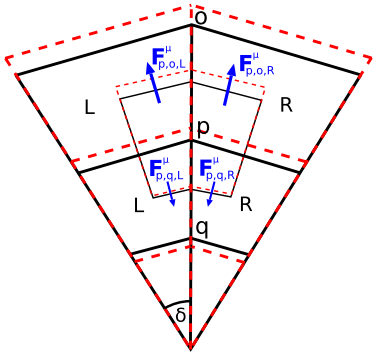


Fig. C.13. Numerical momentum fluxes for a polar grid with radial velocity. Lagrangian mesh is shown by thick black solid line, rezoned mesh by thick red dashed line, fluxes by blue arrows, dual (nodal) cell by corresponding thin lines. Note that the momentum fluxes $\boldsymbol{F}^{\mu}$ are not generally radial, however, they are symmetric with respect to the central line.

$$
\begin{aligned}
\widetilde{m}_p\,\widetilde{\boldsymbol{w}}_p^{\xi\eta} - \widetilde{m}_p\,\boldsymbol{w}_p^{\xi\eta,\min} &= m_p \mathsf{A}_p \boldsymbol{w}_p^{xy} + \sum_{s\in\mathcal{S}(p)} F_{p,s}^m \mathsf{A}_p \boldsymbol{w}_{p,s}^{\mathrm{L},xy} - \left(m_p + \sum_{s\in\mathcal{S}(p)} F_{p,s}^m\right)\boldsymbol{w}_p^{\xi\eta,\min} \\
&= \left[m_p + \sum_{s\in\mathcal{S}(p)} \min\left(F_{p,s}^m, 0\right)\right]\mathsf{A}_p \boldsymbol{w}_p^{xy} + \sum_{s\in\mathcal{S}(p)} \max\left(F_{p,s}^m, 0\right)\mathsf{A}_p \boldsymbol{w}_q^{xy} \\
&\quad - \left[m_p + \sum_{s\in\mathcal{S}(p)} \min\left(F_{p,s}^m, 0\right) + \sum_{s\in\mathcal{S}(p)} \max\left(F_{p,s}^m, 0\right)\right]\boldsymbol{w}_p^{\xi\eta,\min} \\
&= \left[m_p + \sum_{s\in\mathcal{S}(p)} \min\left(F_{p,s}^m, 0\right)\right]\underbrace{\left(\mathsf{A}_p \boldsymbol{w}_p^{xy} - \boldsymbol{w}_p^{\xi\eta,\min}\right)}_{\geqslant \mathbf{0}} \\
&\quad + \sum_{s\in\mathcal{S}(p)} \underbrace{\max\left(F_{p,s}^m, 0\right)}_{\geqslant \mathbf{0}}\underbrace{\left(\mathsf{A}_p \boldsymbol{w}_q^{xy} - \boldsymbol{w}_p^{\xi\eta,\min}\right)}_{\geqslant \mathbf{0}},
\end{aligned}
$$

where the inequalities below the braces are to be understood component by component. The line above implies

$$
\left\{\begin{array}{l} \widetilde{m}_p\,\widetilde{w}_p^{\xi} - \widetilde{m}_p\,w_p^{\xi,\min} \geqslant 0 \\ \widetilde{m}_p\,\widetilde{w}_p^{\eta} - \widetilde{m}_p\,w_p^{\eta,\min} \geqslant 0 \end{array}\right\} \quad \text{if } m_p + \sum_{s\in\mathcal{S}(p)} \min\left(F_{p,s}^m, 0\right) \geqslant 0,
$$

which is obviously a requirement that the sum of outgoing inter-nodal mass fluxes must not exceed the original nodal mass. This condition is related to the CFL-like condition for grid movement during the rezoning phase. It could be also rewritten as

$$
m_p + \sum_{s\in\mathcal{S}(p)} \min\left(F_{p,s}^m, 0\right) = \rho_p\left[V_p + \sum_{s\in\mathcal{S}(p)} \min\left(\frac{F_{p,s}^m}{\rho_p}, 0\right)\right],
$$

where V_p denotes the volume of dual cell p. Therefore the corresponding sufficient condition is

$$
V_p + \sum_{s\in\mathcal{S}(p)} \min\left(\frac{F_{p,s}^m}{\rho_p}, 0\right) \geqslant 0.
$$

For the maximal bound, a similar process results in the same sufficient condition. This condition cannot be guaranteed *a priori* during the mesh rezoning phase. As it depends on the particular mass fluxes, it must be checked during the remapping step. In case that this condition is violated (which does not seem do happen very often in practice), its satisfaction can be accomplished by any method for treating remapper failures, such as splitting the remapping process to several smaller steps known as subcycling.

Appendix C. Proof of symmetry for the flux-corrected remap

Suppose that we have an equiangular polar mesh and move its nodes as in Fig. C.13 to obtain another polar grid, which is close to the original one. This motion leads to symmetric numerical inter-nodal mass fluxes

$$F^m_{p,q,L} = F^m_{p,q,R}, \qquad F^m_{p,o,L} = F^m_{p,o,R}, \tag{C.1}$$

so the fluxes in the radial direction are angle independent. The remaining mass fluxes (corner and polar ones) are zero, because the corresponding cells do not intersect each other.

For the initial (old) symmetric radial velocity, we will demonstrate that the remapped (new) velocity is also radial and independent of angle. The described FCR method uses a combination of low- and high-order velocity reconstructions. For every node, both of them are symmetric with respect to the axis (line from the origin to the node). Local coordinates defined by the radial (ξ) and the angular (η) component are given by the nodal velocity, so they are aligned with the axis. We assume a symmetric low- and high-order velocity reconstruction with components

$$\left(\boldsymbol{w}^{\mathrm{L}}_{p,q,L}\right)^{\xi} = \left(\boldsymbol{w}^{\mathrm{L}}_{p,q,R}\right)^{\xi}, \qquad \left(\boldsymbol{w}^{\mathrm{H}}_{p,q,L}\right)^{\xi} = \left(\boldsymbol{w}^{\mathrm{H}}_{p,q,R}\right)^{\xi}, \tag{C.2}$$

$$\left(\boldsymbol{w}^{\mathrm{L}}_{p,q,L}\right)^{\eta} = \left(\boldsymbol{w}^{\mathrm{L}}_{p,q,R}\right)^{\eta} = 0, \qquad \left(\boldsymbol{w}^{\mathrm{H}}_{p,q,L}\right)^{\eta} = -\left(\boldsymbol{w}^{\mathrm{H}}_{p,q,R}\right)^{\eta}. \tag{C.3}$$

For a polar mesh, the old velocity as well as the nodal mass are symmetric. Therefore the symmetry of the remapped velocity (5) is given by the sum of numerical momentum fluxes

$$\sum_{s\in\mathcal{S}(p)} \boldsymbol{F}^{\mu,\mathrm{FCR}}_{p,s} = \left(\boldsymbol{F}^{\mu,\mathrm{FCR}}_{p,q,L} + \boldsymbol{F}^{\mu,\mathrm{FCR}}_{p,q,R}\right) + \left(\boldsymbol{F}^{\mu,\mathrm{FCR}}_{p,o,L} + \boldsymbol{F}^{\mu,\mathrm{FCR}}_{p,o,R}\right), \tag{C.4}$$

where

$$\boldsymbol{F}^{\mu,\mathrm{FCR}}_{p,q,R} = F^m_{p,q,R}\left[\boldsymbol{w}^{\mathrm{L}}_{p,q,R} + C_{p,q,R}\left(\boldsymbol{w}^{\mathrm{H}}_{p,q,R} - \boldsymbol{w}^{\mathrm{L}}_{p,q,R}\right)\right]. \tag{C.5}$$

To preserve symmetry, the radial (ξ) components of each pair of the fluxes (C.4) have to be symmetric. According to (C.1)–(C.3),

$$\left(\boldsymbol{F}^{\mu,\mathrm{FCR}}_{p,q,L}\right)^{\xi} = \left(\boldsymbol{F}^{\mu,\mathrm{FCR}}_{p,q,R}\right)^{\xi}, \tag{C.6}$$

$$\begin{aligned} &F^m_{p,q,L}\left[\left(\boldsymbol{w}^{\mathrm{L}}_{p,q,L}\right)^{\xi} + C_{p,q,L}\left(\left(\boldsymbol{w}^{\mathrm{H}}_{p,q,L}\right)^{\xi} - \left(\boldsymbol{w}^{\mathrm{L}}_{p,q,L}\right)^{\xi}\right)\right] \\ &\quad = F^m_{p,q,R}\left[\left(\boldsymbol{w}^{\mathrm{L}}_{p,q,R}\right)^{\xi} + C_{p,q,R}\left(\left(\boldsymbol{w}^{\mathrm{H}}_{p,q,R}\right)^{\xi} - \left(\boldsymbol{w}^{\mathrm{L}}_{p,q,R}\right)^{\xi}\right)\right], \end{aligned} \tag{C.7}$$

$$C_{p,q,L} = C_{p,q,R}. \tag{C.8}$$

Considering (C.3), the symmetry condition for the angular (η) component leads to the same requirement

$$\left(\boldsymbol{F}^{\mu,\mathrm{FCR}}_{p,q,L}\right)^{\eta} = -\left(\boldsymbol{F}^{\mu,\mathrm{FCR}}_{p,q,R}\right)^{\eta}, \tag{C.9}$$

$$\left(\boldsymbol{w}^{\mathrm{R}}_{p,q,L}\right)^{\eta} + C_{p,q,L}\left(\left(\boldsymbol{w}^{\mathrm{H}}_{p,q,L}\right)^{\eta} - \left(\boldsymbol{w}^{\mathrm{L}}_{p,q,L}\right)^{\eta}\right) = -\left(\boldsymbol{w}^{\mathrm{R}}_{p,q,R}\right)^{\eta} + C_{p,q,R}\left(-\left(\boldsymbol{w}^{\mathrm{H}}_{p,q,R}\right)^{\eta} + \left(\boldsymbol{w}^{\mathrm{L}}_{p,q,R}\right)^{\eta}\right), \tag{C.10}$$

$$0 + C_{p,q,L}\left(\left(\boldsymbol{w}^{\mathrm{H}}_{p,q,L}\right)^{\eta} - 0\right) = -0 + C_{p,q,R}\left(-\left(\boldsymbol{w}^{\mathrm{H}}_{p,q,R}\right)^{\eta} + 0\right), \tag{C.11}$$

$$C_{p,q,L} = C_{p,q,R}. \tag{C.12}$$

So if $C_{p,q,L} = C_{p,q,R}$, then the FCR fluxes are symmetric and the remapping is also symmetric. It remains to prove that the correction factors $C_{p,q,L}$, $C_{p,q,R}$ are equal for all pairs of the fluxes. Using the same conditions (C.2), (C.3), we can show that the antidiffusive fluxes (21), (22) also satisfy

$$\left(d\boldsymbol{\mu}_{p,q,L}\right)^{\xi} = \left(d\boldsymbol{\mu}_{p,q,R}\right)^{\xi}, \tag{C.13}$$

$$\left(d\boldsymbol{\mu}_{p,q,L}\right)^{\eta} = -\left(d\boldsymbol{\mu}_{p,q,R}\right)^{\eta}. \tag{C.14}$$

First, we look only at the constraints for the radial component. According to Algorithm 1 in Section 5 and because $\left(d\boldsymbol{\mu}_{p,q,L}\right)^{\xi} = \left(d\boldsymbol{\mu}_{p,q,R}\right)^{\xi}$, $\left(d\boldsymbol{\mu}_{q,p,L}\right)^{\xi} = \left(d\boldsymbol{\mu}_{q,p,R}\right)^{\xi}$, we get $C^{\xi}_{p,q,L} = C^{\xi}_{p,q,R}$.

Second, for the angular component constraints, the signs of the antidiffusive fluxes are opposite. Let us choose e.g. $\left(d\boldsymbol{\mu}_{p,q,L}\right)^{\eta} > 0$. Then the final constraints from node p give $C^{\eta}_{p,q,L} = D^{\eta,\max}_p$ and $C^{\eta}_{p,q,R} = D^{\eta,\min}_p$. According to the lemma below, $D^{\eta,\max}_p = D^{\eta,\min}_p$, which implies that $C^{\eta}_{p,q,L} = C^{\eta}_{p,q,R}$. The situation is analogous for the opposite sign of $\left(d\boldsymbol{\mu}_{p,q,L}\right)^{\eta}$ as well as for node q.

The final correction factors, obtained by $C_{p,q,L} = \min(C^{\xi}_{p,q,L}, C^{\eta}_{p,q,L})$, $C_{p,q,R} = \min(C^{\xi}_{p,q,R}, C^{\eta}_{p,q,R})$, are symmetric, i.e. $C_{p,q,L} = C_{p,q,R}$, as $C^{\xi}_{p,q,L} = C^{\xi}_{p,q,R}$ and $C^{\eta}_{p,q,L} = C^{\eta}_{p,q,R}$. So the final result proving symmetry is obtained. Exactly the same procedure can be used for the other pair of nodes (p, o) appearing in (C.4). The correction factors from the left- and right-hand side of each node are the same.

Lemma 1. *For a polar grid, radial mesh movement and radial velocity, $D^{\eta,\max}_p$ and $D^{\eta,\min}_p$ are equal.*

Proof of Lemma 1. Considering (24a), we obtain

$$\frac{Q_p^{\eta,\max}}{\widetilde{m}_p} = w_p^{\eta,\max} = \max_{s\in\{p,q,o\}}\left\{0, (\boldsymbol{w}_s)^\xi \sin\delta, -(\boldsymbol{w}_s)^\xi \sin\delta\right\}$$

$$= -\min_{s\in\{p,q,o\}}\left\{0, -(\boldsymbol{w}_s)^\xi \sin\delta, (\boldsymbol{w}_s)^\xi \sin\delta\right\} = -\frac{Q_p^{\eta,\min}}{\widetilde{m}_p}. \tag{C.15}$$

From (C.14), we can write

$$0 = \left((d\boldsymbol{\mu}_{p,q,L})^\eta + (d\boldsymbol{\mu}_{p,q,R})^\eta\right) + \cdots = \sum_{s\in\mathcal{S}(p)} (d\boldsymbol{\mu}_{p,s})^\eta = \sum_{s,d\mu>0} (d\boldsymbol{\mu}_{p,s})^\eta + \sum_{s,d\mu<0} (d\boldsymbol{\mu}_{p,s})^\eta, \tag{C.16}$$

$$D_p^{\eta,\max} = \frac{Q_p^{\eta,\max}}{\sum_{s,d\mu>0}(d\boldsymbol{\mu}_{p,s})^\eta} = \frac{-Q_p^{\eta,\min}}{-\sum_{s,d\mu<0}(d\boldsymbol{\mu}_{p,s})^\eta} = D_p^{\eta,\min}. \tag{C.17}$$

References

[1] R.W. Anderson, N.S. Elliott, R.B. Pember, An arbitrary Lagrangian–Eulerian method with adaptive mesh refinement for the solution of the Euler equations, J. Comput. Phys. 199 (2) (2004) 598–617.
[2] D.J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Comput. Methods Appl. Mech. Eng. 99 (2–3) (1992) 235–394.
[3] M. Berndt, J. Breil, S. Galera, M. Kucharik, P.-H. Maire, M. Shashkov, Two step hybrid remapping (conservative interpolation) for multimaterial arbitrary Lagrangian–Eulerian methods, J. Comput. Phys. 230 (17) (2010) 6664–6687.
[4] J. Boris, D. Book, Flux-corrected transport I: SHASTA, a fluid transport algorithm that works, J. Comput. Phys. 11 (1) (1973) 38–69. Reprinted in J. Comput. Phys. 135 (2) (1997) 172–186.
[5] E.J. Caramana, D.E. Burton, M.J. Shashkov, P.P. Whalen, The construction of compatible hydrodynamics algorithms utilizing conservation of total energy, J. Comput. Phys. 146 (1) (1998) 227–262.
[6] E.J. Caramana, M.J. Shashkov, P.P. Whalen, Formulations of artificial viscosity for multi-dimensional shock wave computations, J. Comput. Phys. 144 (2) (1998) 70–97.
[7] R.B. DeBar, Fundamentals of the KRAKEN code, Technical Report UCIR-760, Lawrence Livermore Laboratory, 1974.
[8] J.K. Dukowicz, J.R. Baumgardner, Incremental remapping as a transport/advection algorithm, J. Comput. Phys. 160 (1) (2000) 318–335.
[9] S. Galera, P.-H. Maire, J. Breil, A two-dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction, J. Comput. Phys. 229 (16) (2010) 5755–5787.
[10] C.W. Hirt, A.A. Amsden, J.L. Cook, An arbitrary Lagrangian–Eulerian computing method for all flow speeds, J. Comput. Phys. 14 (3) (1974) 227–253.
[11] M. Kucharik, J. Breil, S. Galera, P.-H. Maire, M. Berndt, M. Shashkov, Hybrid remap for multi-material ALE, Comput. Fluids 46 (1) (2011) 293–297.
[12] M. Kucharik, M. Shashkov, Flux-based approach for conservative remap of multi-material quantities in 2D arbitrary Lagrangian–Eulerian simulations, in: J. Fořt, J. Fürst, J. Halama, R. Herbin, F. Hubert (Eds.), Finite Volumes for Complex Applications VI – Problems & Perspectives, vol. 1, in: Springer Proceedings in Mathematics, vol. 4, Springer, 2011, pp. 623–631.
[13] M. Kucharik, M. Shashkov, One-step hybrid remapping algorithm for multi-material arbitrary Lagrangian–Eulerian methods, J. Comput. Phys. 231 (7) (2012) 2851–2864.
[14] M. Kucharik, M. Shashkov, Conservative multi-material remap for staggered multi-material arbitrary Lagrangian–Eulerian methods, 2013, submitted for publication.
[15] M. Kucharik, M. Shashkov, B. Wendroff, An efficient linearity-and-bound-preserving remapping method, J. Comput. Phys. 188 (2003) 462–471.
[16] D. Kuzmin, R. Löhner, S. Turek (Eds.), Flux-Corrected Transport. Principles, Algorithms and Applications, 2nd edition, Springer Verlag, Berlin, Heidelberg, 2012.
[17] R. Liska, M. Shashkov, P. Váchal, B. Wendroff, Optimization-based synchronized flux-corrected conservative interpolation (remapping) of mass and momentum for arbitrary Lagrangian–Eulerian methods, J. Comput. Phys. 229 (5) (2010) 1467–1497.
[18] R. Liska, M. Shashkov, P. Váchal, B. Wendroff, Synchronized flux corrected remapping for ALE methods, Comput. Fluids 46 (1) (2011) 312–317.
[19] R. Loubere, P.-H. Maire, M. Shashkov, J. Breil, S. Galera, ReALE: A reconnection-based arbitrary-Lagrangian–Eulerian method, J. Comput. Phys. 229 (12) (2010) 4724–4761.
[20] R. Loubère, M. Staley, B. Wendroff, The repair paradigm: New algorithms and applications to compressible flow, J. Comput. Phys. 211 (2) (2006) 385–404.
[21] G. Luttwak, J. Falcovitz, Slope limiting for vectors: A novel vector limiting algorithm, Int. J. Numer. Methods Fluids 65 (11–12) (2011) 1365–1375.
[22] G. Luttwak, J. Falcovitz, Vip (vector image polygon) multi-dimensional slope limiters for scalar variables, Comput. Fluids 83 (2013) 90–97.
[23] G. Luttwak, J. Falcovitz, Vector image polygon (vip) limiters in ale hydrodynamics, EPJ Web Conf. 10 (2010) 00020.
[24] P.-H. Maire, Contribution to the numerical modeling of Inertial Confinement Fusion, Habilitation thesis, Université Bordeaux I, 2011, http://tel.archives-ouvertes.fr/docs/00/58/97/58/PDF/hdr_main.pdf.
[25] P.-H. Maire, A high-order one-step sub-cell force-based discretization for cell-centered Lagrangian hydrodynamics on polygonal grids, Comput. Fluids 46 (1) (2011) 341–347.
[26] P.-H. Maire, R. Loubère, P. Váchal, Staggered Lagrangian discretization based on cell-centered Riemann solver and associated hydrodynamics scheme, Commun. Comput. Phys. 10 (4) (2011) 940–978.
[27] L.G. Margolin, Introduction to "An arbitrary Lagrangian–Eulerian computing method for all flow speeds", J. Comput. Phys. 135 (2) (1997) 198–202.
[28] L.G. Margolin, M. Shashkov, Second-order sign-preserving conservative interpolation (remapping) on general grids, J. Comput. Phys. 184 (1) (2003) 266–298.
[29] J.M. Morrell, P.K. Sweby, A. Barlow, A cell by cell anisotropic adaptive mesh ALE scheme for the numerical solution of the Euler equations, J. Comput. Phys. 226 (1) (2007) 1152–1180.
[30] J.M. Owen, M.J. Shashkov, ALE remap treatment consistent with corner based compatible total energy conserving Lagrangian methods, in: Nuclear Explosive Code Developers Conference, LLNL-PRES-457355, Livermore CA, 2010.
[31] J.S. Peery, D.E. Carroll, Multi-material ALE methods in unstructured grids, Comput. Methods Appl. Mech. Eng. 187 (3–4) (2000) 591–619.
[32] 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, Technical Report, LLNL, 2000, UCRL-JC-139820.

[33] G. Scovazzi, E. Love, M. Shashkov, Multi-scale Lagrangian shock hydrodynamics on Q1/P0 finite elements: Theoretical framework and two-dimensional computations, Comput. Methods Appl. Mech. Eng. 197 (9–12) (2008) 1056–1079.
[34] L.I. Sedov, Similarity and Dimensional Methods in Mechanics, Academic Press, 1959.
[35] M. Shashkov, B. Wendroff, The repair paradigm and application to conservation laws, J. Comput. Phys. 198 (1) (2004) 265–277.
[36] P. Váchal, R. Liska, Sequential flux-corrected remapping for ALE methods, in: A. Bermúdez de Castro, D. Gómez, P. Quintela, P. Salgado (Eds.), Numerical Mathematics and Advanced Applications, ENUMATH 2005, Springer, Berlin, Heidelberg, New York, 2006, pp. 671–679.
[37] J. Velechovsky, M. Kucharik, R. Liska, M. Shashkov, Symmetry-preserving momentum remap for ALE hydrodynamics, J. Phys.: Conf. Ser. 454 (2013) 012003.
[38] J. Velechovsky, R. Liska, M. Shashkov, High-order remapping with piecewise parabolic reconstruction, Comput. Fluids 83 (2013) 164–169.
[39] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 31 (3) (1979) 335–362.

Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2015-0021

Vol. **21**, No. 2, pp. 526-558
February 2017

Local Error Analysis and Comparison of the Swept- and Intersection-Based Remapping Methods

Matej Klima[1], Milan Kucharik[1,*] and Mikhail Shashkov[2]

[1] *Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Brehova 7, Praha 1, 115 19, Czech Republic.*
[2] *XCP-4 Group, MS-F644, Los Alamos National Laboratory, Los Alamos, NM 87545, USA.*

Received 12 December 2015; Accepted (in revised version) 17 July 2016

Abstract. In this paper, the numerical error of two widely used methods for remapping of discrete quantities from one computational mesh to another is investigated. We compare the intuitive, but resource intensive method utilizing intersections of computational cells with the faster and simpler swept-region-based method. Both algorithms are formally second order accurate, however, they are known to produce slightly different quantity profiles in practical applications. The second-order estimate of the error formula is constructed algebraically for both algorithms so that their local accuracy can be evaluated. This general estimate is then used to assess the dependence of the performance of both methods on parameters such as the second derivatives of the remapped distribution, mesh geometry or mesh movement. Due to the complexity of such analysis, it is performed on a set of simplified elementary mesh patterns such as cell corner expansion, rotation or shear. On selected numerical tests it is demonstrated that the swept-based method can distort a symmetric quantity distribution more substantially than the intersection-based approach when the computational mesh moves in an unsuitable direction.

PACS: 47.11.-j, 02.60.Ed

Key words: Conservative interpolation, remapping, numerical error analysis, swept regions, polygon intersections.

1 Introduction

For numerical simulations of fluid dynamics, the computational methods are typically categorized into two classes – the Eulerian and Lagrangian methods. In the pioneering work [11], the authors developed a more general framework combining best properties

*Corresponding author. *Email addresses:* klimamat@fjfi.cvut.cz (M. Klima), kucharik@newton.fjfi.cvut.cz (M. Kucharik), shashkov@lanl.gov (M. Shashkov)

of both Lagrangian and Eulerian approaches. This framework has been termed Arbitrary Lagrangian-Eulerian or ALE and since that, many authors have contributed to the investigation of its robustness, accuracy, or efficiency, see for example [1,3,10,15,24,25,29].

The ALE algorithm is usually separated in three distinct stages: (1) a Lagrangian stage, in which the fluid quantities and the computational mesh are advanced in time; (2) a rezoning stage, in which nodes of the (potentially distorted) computational mesh are moved to more optimal positions with respect to their geometrical quality; and (3) a remapping stage, in which all fluid quantities are conservatively transferred from the Lagrangian mesh to the rezoned one. The ALE algorithm preserves the advantages of the Lagrangian methods (such as low dissipation at the discontinuities or the computational mesh intrinsically following the fluid), while its Eulerian part (rezoning and remapping) prevents the computational mesh from degeneration often appearing in purely Lagrangian simulations. In this paper, we focus on the last part of the ALE algorithm – the remapping stage in single-material simulations.

In case of close computational meshes with the same topology, the remapping process can be formulated in a flux form, using fluxes of the involved quantities between cells which share the face. The fluxes are constructed by integrating the particular fluid quantity over certain transport volumes. Here, we discuss two methods of constructing such volumes [19,21,26,27]. The first method is more intuitive and employs the exact intersections of the computational cells with their neighbors in the new mesh. It is known to be more demanding in terms of computational resources, but it can, under certain circumstances, perform better in terms of solution symmetry (especially in case of discontinuous solutions and corner coupling), such as observed for example in [5,14]. The second widely used method approximates the calculated inter-cell fluxes using regions swept by the cell edges during the transformation from one mesh to another. It is robust and computationally less demanding, however, the approximation used raises concerns about its accuracy. The question is, whether it is possible to determine which method is better suited for a specific application or problem.

As far as we know, there exist two papers addressing the theoretical error analysis of the remapping methods. An error analysis based on the Fourier decomposition of the numerical error was performed in [23], showing that the faster swept-region-based method can under certain circumstances provide better results than using the intersections. Another analysis was presented earlier in [27], confirming the second order of accuracy and other properties of both methods. In the current paper, we theoretically analyze the numerical error of both methods locally, while treating the remapping process as an interpolation method rather than as a fluid flow through the computational cell edges.

However, to be able to determine which method is more suitable for specific data, an analysis of the overall accuracy for the entire computational domain is not sufficient. We need to perform an analysis of the distribution of the remapping error caused by each of the two methods. Either the fluid quantities, their derivatives (including the second derivatives), or the geometrical characteristics (such as cell volume or nodal movement

during the rezoning phase) can serve as parameters for the analysis.

This work extends and complements the preceding research of the hybrid remapping methods [4, 18, 20], where combination of both remapping approaches was employed depending on the presence of the material interfaces. This concept can be extended in the form of a "pseudo-hybrid" method, where the same combination of remapping approaches is used based on the local error estimate, trying to use the more accurate method in each particular computational cell. The local error analysis presented in this paper provides a theoretical justification for such combination, identifying particular terms in the error formula which are responsible for accuracy or symmetry violation of both methods.

The rest of the paper is organized as follows. In Section 2, the intersection-based and swept-region-based remapping algorithms in a flux form are overviewed and the sources of the numerical error for the swept-based algorithm are identified. In Section 3, the main part of the paper is presented. A particular remapping algorithm is analyzed in case of a general smooth function and general mesh motion. In the following subsections, several typical simplified mesh motions are analyzed in detail and the general error formula is adapted, showing which method is more suitable in such cases. In Section 4, the theoretical analysis is supported by selected numerical tests, including both static remapping and a full hydrodynamic simulation. The whole paper is concluded in Section 5.

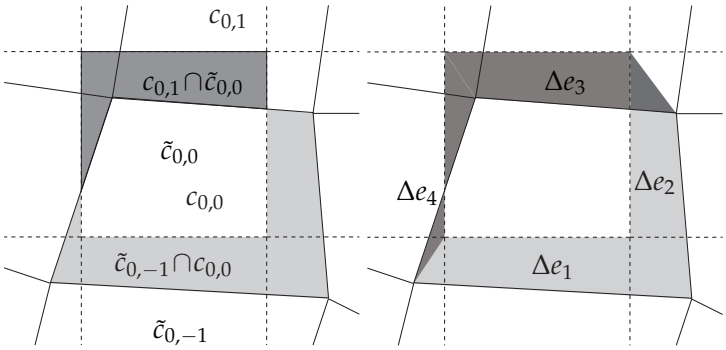
2 Overview of swept- and intersection-based remapping algorithms

The remapping process represents essentially a conservative interpolation of a discrete quantity from one computational mesh to another. We assume that both meshes share the same topology and are close to each other so that no cell intersects any cell of the other mesh that is not in its immediate neighborhood. We will describe this process on a general polygonal mesh (although the numerical examples in this paper are shown on logically rectangular meshes). The discrete scalar quantity (such as density) is defined by the mean value in the computational cell c and will be referred to as f_c.

The remapping algorithm can be formulated in a flux form [9, 19, 21]:

$$\tilde{f}_c \tilde{V}_c = f_c V_c + \sum_{c' \in C(c)} F_{c' \to c}, \tag{2.1}$$

where $\tilde{f}_c$ is the remapped quantity and $\tilde{V}_c$ represents the new cell volume which is calculated from the known new mesh geometry. $C(c)$ is a set of all cells surrounding cell c (sharing at least one vertex with c). $F_{c' \to c}$ represents the quantity flux into cell c from its neighbor c'. The construction of the fluxes depends on the particular remapping method used. Remap defined by this formula is always conservative due to its flux nature, no matter how the fluxes are computed. In case of remapping fluid density ($f_c = \rho_c$), the formula represents remapping of mass and the fluxes $F_{c' \to c}$ represent mass fluxes between the cells.



(a) Exact intersections of cells (b) Regions swept by cell edges

Figure 1: Comparison of flux construction in the remapping methods, dashed line – new mesh, solid line – old mesh, ■ positive flux, ■ negative flux.

The intersection-based method can be used for remapping of fluid quantities between two arbitrary meshes. However, we only consider the case of close meshes with the same topology, allowing to express the remapped quantity in the flux form (2.1). The fluxes are calculated by integrating the reconstructed function over the intersections of the new cell polygon with each its neighbor in the old position and vice versa [19, 21], as seen in Fig. 1(a). The particular flux can be split in two terms, each representing its opposite directions (pointing inside and outside of cell c):

$$F_{c'\to c} = F_{c'\cap\tilde{c}} - F_{c\cap\tilde{c}'}, \tag{2.2}$$

where $\tilde{c}$ represents the region enclosed by the cell c in its new position. Each of these terms can be approximated by integrating a function reconstructed from the discrete values of f_c over the intersecting polygon,

$$F_{c'\cap\tilde{c}} = \int_{c'\cap\tilde{c}} f^{\text{rec}}_{c'}(x,y)\,dV. \tag{2.3}$$

In the second-order method presented here, a piecewise linear reconstruction is computed in the intersecting region:

$$f^{\text{rec}}_{c'}(x,y) = f_{c'} + \left(\frac{\partial f}{\partial x}\right)_{c'}(x-x_{c'}) + \left(\frac{\partial f}{\partial y}\right)_{c'}(y-y_{c'}), \tag{2.4}$$

where $x_{c'}$, $y_{c'}$ are the coordinates of the geometrical centroid of the respective cell,

$$x_{c'} = \frac{\int_{c'} x\,dV}{\int_{c'} 1\,dV}, \quad y_{c'} = \frac{\int_{c'} y\,dV}{\int_{c'} 1\,dV}. \tag{2.5}$$

The function derivatives are estimated numerically using a least squares minimization method with the Barth-Jespersen limiter [2, 17, 28]. Calculation of the fluxes then represents calculation of a first degree polynomial over a polygonal region, that can be evaluated analytically.

The difficult part is to calculate the intersecting polygon coordinates accurately. In general, the design of a numerical polygon intersection algorithm is a complex issue out of the scope of this paper. Most often in case of mesh smoothing, the differences between the original and the rezoned computational meshes are very small. Therefore, the algorithm must be robust enough to handle regions with almost degenerate shapes and close-to-parallel edges [21].

In the swept-based algorithm, the fluxes are not calculated by the exact integration as in (2.3) in each direction separately (2.2), instead they are approximated by only one flux per edge:

$$\tilde{f}_c \tilde{V}_c = f_c V_c + \sum_{e \in E(c)} \int_{\Delta e} f^{\mathrm{rec}}_{c(\Delta e)}(x,y)\, dV, \tag{2.6}$$

where $E(c)$ is a set of all edges of cell c and the integration is performed over the regions swept by each edge Δe (shown in Fig. 1(b)) during the transition from the old to the new computational mesh. The corner fluxes are not explicitly treated in this approach, they are approximated together with the adjacent cell fluxes [27]. $c(\Delta e)$ represents the cell from which the reconstruction is taken – either c or the neighboring cell sharing the edge e, depending on the sign of the oriented volume integral over c, which also determines the direction of the particular flux. The swept region is defined by the old and new positions of the cell vertices so an intersection algorithm is not needed. This makes the method very robust and fast. However, concerns about its accuracy exist. First, this method does not account for the corner fluxes – their contributions are approximated by the reconstruction from the adjacent edge-neighboring cells. This can be seen for example in Fig. 1(b) in the lower right corner of the cell. And second, if the cell edge of the new mesh intersects the original edge, there are two fluxes in the intersection-based approach with opposite directions, each with its own reconstructed function. In this situation, the swept-region method uses only one of the reconstructions which cannot be chosen properly, since it will not be accurate in the other part of the self-intersecting region (the "hourglass" case in Fig. 1(b), as shown at the left edge of the depicted cell). This effect is most significant when the swept region volume approaches zero. Let us note that this effect can be improved by decomposing the swept region to two triangles in this case (requiring only finding an intersections of two straight lines), such as described in [12]. However, the first effect of inaccurate reconstruction in the corner regions remains as full intersection of the swept region with the old mesh [20] is needed for correct reconstruction, which is just a different formulation of the intersection-based approach.

3 Estimating the local remapping error

In most cases, the accuracy of the remapping method utilizing swept regions integration is comparable to integrating the exact cell intersections [4, 20, 22, 27]. The swept-based method is, however, much simpler, robust and computationally more efficient; therefore it is often the first choice for single-material quantity remapping. Albeit in some

cases where symmetry preservation is crucial (eg. when remapping a radially symmetric quantity distribution), the exact method is known to yield better results. To design some criteria for selecting either intersections or swept regions, we need to know how the method error is influenced by the quantity distribution and the computational mesh.

In our analysis, we do not treat the remapping process as an advection scheme, but rather as an interpolation method. That way we can focus on the local remapping error on the level of individual cells. Often, the swept-based method distorts the solution only in a small portion of the cells in the computational mesh, so knowing where the exact intersection method is locally more accurate can be beneficial.

We assume that the discrete quantity f_c represents the mean value of a certain analytical function f in the cell c. This underlying function and its properties are not known in general, but for the purpose of this analysis we presume that this function is continuous and differentiable. Although this approximation may be less accurate for initially discontinuous distributions, it needs to be emphasized that we are making an independent local approximation for each computational cell that is valid only within the cell itself. As such, it will converge with increasing mesh resolution. We use the second-order Taylor expansion in the cell centroid for the purpose of local error estimation:

$$f_c = \frac{1}{V_c}\int_c \left(f(\mathbf{r_c}) + \nabla f(\mathbf{r_c})\cdot(\mathbf{r}-\mathbf{r_c}) + \frac{1}{2}(\mathbf{r}-\mathbf{r_c})^{\mathrm{T}}\mathbf{H}(\mathbf{r}-\mathbf{r_c}) + \mathcal{O}(\mathbf{r}^3) \right) dV, \tag{3.1}$$

where $\mathbf{r}$ is the general position vector, $\mathbf{r_c}$ is the location of the cell centroid and $\mathbf{H}$ is the Hessian matrix of the function f (matrix of its second derivatives). As we assume a second-order approximation, $\mathbf{H}$ is treated as constant in the selected cell and its neighbors – we do not need to evaluate it separately at a certain point. The linear part of this function is reduced to the function value evaluated at the cell centroid after integration:

$$f_c = f(\mathbf{r_c}) + \frac{1}{2V_c}\int_c (\mathbf{r}-\mathbf{r_c})^{\mathrm{T}}\mathbf{H}(\mathbf{r}-\mathbf{r_c}) + \mathcal{O}(\mathbf{r}^3)\, dV. \tag{3.2}$$

The integral in this formula can be analytically evaluated for a polygonal region, yielding a rather complicated result. We will further refer to this integral as:

$$I_c^{\mathbf{H}}(\mathbf{r_c}) = \int_c (\mathbf{r}-\mathbf{r_c})^{\mathrm{T}}\mathbf{H}(\mathbf{r}-\mathbf{r_c})\, dV. \tag{3.3}$$

This integral can be evaluated by expanding the quadratic form:

$$I_c^{\mathbf{H}}(\mathbf{r_c}) = \int_c \mathbf{r}^{\mathrm{T}}\mathbf{H}\mathbf{r} - 2\mathbf{r}^{\mathrm{T}}\mathbf{H}\mathbf{r_c} + \mathbf{r_c}^{\mathrm{T}}\mathbf{H}\mathbf{r_c}\, dV, \tag{3.4}$$

due to the fact that the Hessian is symmetric. The Hessian can be diagonalized in the following way:

$$I_c^{\mathbf{H}}(\mathbf{r_c}) = \int_c \mathbf{r}^{\mathrm{T}}\mathbf{P}\mathbf{\Lambda}\mathbf{P}^{\mathrm{T}}\mathbf{r} - 2\mathbf{r}^{\mathrm{T}}\mathbf{H}\mathbf{r_c} + \mathbf{r_c}^{\mathrm{T}}\mathbf{H}\mathbf{r_c}\, dV, \tag{3.5}$$

where the transformation matrix $\mathbf{P}$ is composed of the principal directions (ie. the eigenvectors of $\mathbf{H}$) as columns. In the first term, we can rotate the coordinates using the transformation $\mathbf{u}=\mathbf{P}^{\mathrm{T}}\mathbf{r}$ to simplify the integration. $\mathbf{P}$ can be considered a rotation matrix without loss of generality. The second integrand is a linear function and the third one a constant. The formula can be expressed as:

$$I_c^{\mathbf{H}}(\mathbf{r_c})=\int_c \mathbf{u}^{\mathrm{T}}\mathbf{\Lambda u}dV+\sum_{i=0}^{n}\begin{vmatrix}x_i & x_{i+1}\\ y_i & y_{i+1}\end{vmatrix}\left(\frac{\mathbf{r_c}}{2}-\frac{\mathbf{r}_i+\mathbf{r}_{i+1}}{3}\right)^{\mathrm{T}}\mathbf{H}\mathbf{r_c}, \tag{3.6}$$

where i is the index of the closed polygon vertices and $\mathbf{r}_i$ is the vertex position ($\mathbf{r}_0=\mathbf{r}_n$). The integral now consists of quadratic terms only and can be evaluated on a polygon in the following way:

$$\int_c \mathbf{u}^{\mathrm{T}}\mathbf{\Lambda u}dV=\frac{1}{12}\sum_{i=0}^{n}\begin{vmatrix}u_{x,i} & u_{x,i+1}\\ u_{y,i} & u_{y,i+1}\end{vmatrix}\begin{pmatrix}u_{x,i}^2+u_{x,i}u_{x,i+1}+u_{x,i+1}^2\\ u_{y,i}^2+u_{y,i}u_{y,i+1}+u_{y,i+1}^2\end{pmatrix}\cdot\begin{pmatrix}k_1\\ k_2\end{pmatrix}, \tag{3.7}$$

where $u_{x,i}$ is the x coordinate of the i-th vertex in the transformed coordinate system. k_1,k_2 are the principal curvatures – the eigenvalues of $\mathbf{H}$. We can now substitute back for $\mathbf{u}$:

$$\int_c \mathbf{r}^{\mathrm{T}}\mathbf{H}\mathbf{r}dV=\frac{1}{12}\sum_{i=0}^{n}\begin{vmatrix}x_i & x_{i+1}\\ y_i & y_{i+1}\end{vmatrix}\left(\mathbf{r}_i^{\mathrm{T}}\mathbf{H}\mathbf{r}_i+\mathbf{r}_i^{\mathrm{T}}\mathbf{H}\mathbf{r}_{i+1}+\mathbf{r}_{i+1}^{\mathrm{T}}\mathbf{H}\mathbf{r}_{i+1}\right) \tag{3.8}$$

and use this result in (3.6). After simplification, we obtain:

$$\begin{aligned}I_c^{\mathbf{H}}(\mathbf{r_c})=\frac{1}{12}\sum_{i=0}^{n}\|\mathbf{r_i}\times\mathbf{r_{i+1}}\|^2\Big((\mathbf{r_i}-\mathbf{r_c})^{\mathrm{T}}\mathbf{H}(\mathbf{r_i}-\mathbf{r_c})\\ +(\mathbf{r_i}-2\mathbf{r_c})^{\mathrm{T}}\mathbf{H}(\mathbf{r_{i+1}}-2\mathbf{r_c})+(\mathbf{r_{i+1}}-\mathbf{r_c})^{\mathrm{T}}\mathbf{H}(\mathbf{r_{i+1}}-\mathbf{r_c})\Big).\end{aligned} \tag{3.9}$$

This formula has been verified numerically for selected functions and mesh shapes and holds exactly.

3.1 Remapping error of an arbitrary single flux

The developed formula (3.9) cannot be analyzed for a general mesh motion and general function. Here, we show how to construct the error formula in the case when only one non-zero flux is present due to a special mesh movement, and in the next sections, we construct the particular error formulas for the swept-based and intersection-based methods in cases of certain elementary motion patterns.

As we can now evaluate the integral of the quadratic form, we can express the remapping error in case of a single flux by comparing the remapped value with the mean value of the analytical function in the new cell:

$$\epsilon_{\mathrm{remap}}=\tilde{f}_c-\frac{1}{\tilde{V}_c}\int_{\tilde{c}}f(\mathbf{r})dV=\frac{1}{\tilde{V}_c}\left(f_cV_c+\int_F f_{c'}^{\mathrm{rec}}(\mathbf{r})dV-\int_{\tilde{c}}f(\mathbf{r})dV\right), \tag{3.10}$$

where F is the region of the flux (in this simplified case, the swept- and intersection-based methods produce the same results), c and $\tilde{c}$ represent the remapped cell in its old and new positions, respectively. c' is the neighboring cell from which the reconstruction is obtained. The integration region of the new cell can be split into two parts, namely $\tilde{c}=c\cup F$, so the cell term cancels out:

$$\epsilon_{\text{remap}}=\frac{1}{\tilde{V}_c}\left(\int_F f_{c'}^{\text{rec}}(\mathbf{r})dV-\int_F f(\mathbf{r})dV\right). \tag{3.11}$$

We can now substitute for $f_{c'}^{\text{rec}}$ from (2.4) with an assumption that the numerical first derivative is exact (the methods used to calculate it may vary and comparing them is out of the scope of this paper). If we replace $f(\mathbf{r})$ with its second-order Taylor expansion in $\mathbf{r}_{c'}$, we obtain:

$$\epsilon_{\text{remap}}=\frac{1}{\tilde{V}_c}\left(\int_F f_{c'}-f(\mathbf{r}_{c'})-\frac{1}{2}(\mathbf{r}-\mathbf{r}_{c'})^{\mathrm{T}}\mathbf{H}(\mathbf{r}-\mathbf{r}_{c'})dV\right). \tag{3.12}$$

Note that the mean value $f_{c'}$ is not equal to the function value in the centroid. For $f_{c'}$, we can substitute from (3.2) and simplify the integrals as:

$$\epsilon_{\text{remap}}=\frac{V_F}{2\tilde{V}_c}\left(\frac{I_{c'}^{\mathbf{H}}(\mathbf{r}_{c'})}{V_{c'}}-\frac{I_F^{\mathbf{H}}(\mathbf{r}_{c'})}{V_F}\right), \tag{3.13}$$

where V_F is the volume of the flux region. Both terms here represent the mean value of the second-order approximation of the remapping error. They differ only in the integration region, the first one being the consequence of the approximate mean value of f in c' used in the reconstruction. The second one represents the error of the reconstruction in the flux region.

We would like to know also the sign of the remapping error. Generally, comparing two mean values of a certain function over a non-specific region is difficult. There are, however, some conditions that are sufficient to determine the sign:

1. $|\mathbf{H}|\geq 0;\ \mathbf{H}_{xx}+\mathbf{H}_{yy}>0;\ \frac{I_{c'}^{\mathbf{H}}(\mathbf{r}_{c'})}{V_{c'}}\leq(\mathbf{r_F}-\mathbf{r_{c'}})^T\mathbf{H}(\mathbf{r_F}-\mathbf{r_{c'}})\ \Rightarrow\ \epsilon_{\text{remap}}\leq 0,$
2. $|\mathbf{H}|\geq 0;\ \mathbf{H}_{xx}+\mathbf{H}_{yy}<0;\ \frac{I_{c'}^{\mathbf{H}}(\mathbf{r}_{c'})}{V_{c'}}\geq(\mathbf{r_F}-\mathbf{r_{c'}})^T\mathbf{H}(\mathbf{r_F}-\mathbf{r_{c'}})\ \Rightarrow\ \epsilon_{\text{remap}}\geq 0,$
3. $|\mathbf{H}|<0\ \cdots$ inconclusive.

Here, $\mathbf{r_F}$ represents the geometrical centroid of the flux region. Its value is known, because its calculation is performed in both remapping methods. The first and second conditions correspond to the case of a positive or negative semi-definite Hessian matrix, respectively. In the first case, we can make a lower estimate of $I_F^{\mathbf{H}}(\mathbf{r}_{c'})/V_F$ over the region of the flux by integrating the tangential plane of the error function (an elliptical paraboloid) instead. As we construct the plane in $\mathbf{r_F}$, this integral is equal to the value of

the error function in the centroid. This can be applied in a similar manner in the case of a negative semi-definite Hessian matrix. Note that the opposite statement is not generally valid, but on the other hand, it can be used as an approximation of the error even for an indefinite Hessian matrix ($|\mathbf{H}|<0$):

$$\epsilon_{\text{remap}} \sim \frac{V_F}{2\tilde{V}_c}\left(\frac{I^{\mathbf{H}}_{c'}(\mathbf{r}_{c'})}{V_{c'}} - (\mathbf{r_F}-\mathbf{r_{c'}})^T\mathbf{H}(\mathbf{r_F}-\mathbf{r_{c'}})\right). \tag{3.14}$$

In this approximation we assume that the dimensions of the flux region are small when compared to the size of the cell c'. Replacing the integral over the flux region with the value in the centroid can be useful in applications where it might be inconvenient to calculate the integrals for every flux.

In the next sections, we focus on several elementary patterns present during the rezoning process. For an overview of the mesh motion patters, see [7]. More complex error terms are derived using the single flux error expression (3.13). Then, we analyze the numerical error of the intersection and swept-based approaches in such simplified cases.

3.2 Remapping error of a corner movement

The first case in which the swept- and intersection-based remapping methods differ is the movement of the corner of one computational cell. In this section we describe a general case where the movement of the upper-right vertex is arbitrary. The other two neighboring vertices can move arbitrarily along the edges so that the intersection-based algorithm finds non-zero intersections only with cells $c_{0,1}$,$c_{1,1}$ and $c_{1,0}$. The swept-based algorithm treats this case as only two swept regions and uses the function reconstruction from cells $c_{0,1}$ and $c_{1,0}$, respectively. Using (3.13) for each flux, we can write the error of the swept-based method as follows:

$$\begin{aligned}\epsilon_{\text{sw}} = \frac{1}{2\tilde{V}_{c_{0,0}}}\Bigg(&\frac{V_{F_{\text{N}}+F_{\text{NE1}}}}{V_{c_{0,1}}} I^{\mathbf{H}}_{c_{0,1}}(\mathbf{r}_{c_{0,1}}) - I^{\mathbf{H}}_{F_{\text{N}}+F_{\text{NE1}}}(\mathbf{r}_{c_{0,1}}) \\ &+ \frac{V_{F_{\text{E}}+F_{\text{NE2}}}}{V_{c_{1,0}}} I^{\mathbf{H}}_{c_{1,0}}(\mathbf{r}_{c_{1,0}}) - I^{\mathbf{H}}_{F_{\text{E}}+F_{\text{NE2}}}(\mathbf{r}_{c_{1,0}})\Bigg),\end{aligned} \tag{3.15}$$

where F_{N}, F_{E}, F_{NE1}, and F_{NE2} represent the flux regions and $c_{i,j}$ are the corresponding cells (see Fig. 2). The error of the intersection-based method can be described in a similar way:

$$\begin{aligned}\epsilon_{\text{int}} = \frac{1}{2\tilde{V}_{c_{0,0}}}\Bigg(&\frac{V_{F_{\text{N}}}}{V_{c_{0,1}}} I^{\mathbf{H}}_{c_{0,1}}(\mathbf{r}_{c_{0,1}}) - I^{\mathbf{H}}_{F_{\text{N}}}(\mathbf{r}_{c_{0,1}}) + \frac{V_{F_{\text{NE1}}+F_{\text{NE2}}}}{V_{c_{1,1}}} I^{\mathbf{H}}_{c_{1,1}}(\mathbf{r}_{c_{1,1}}) \\ &- I^{\mathbf{H}}_{F_{\text{NE1}}+F_{\text{NE2}}}(\mathbf{r}_{c_{1,1}}) + \frac{V_{F_{\text{E}}}}{V_{c_{1,0}}} I^{\mathbf{H}}_{c_{1,0}}(\mathbf{r}_{c_{1,0}}) - I^{\mathbf{H}}_{F_{\text{E}}}(\mathbf{r}_{c_{1,0}})\Bigg).\end{aligned} \tag{3.16}$$

Now, we would like to express the difference between the two errors to analyze, which method is more accurate:

$$\Delta\epsilon = |\epsilon_{\text{int}}| - |\epsilon_{\text{sw}}|, \quad |\Delta\epsilon| \le |\epsilon_{\text{int}} - \epsilon_{\text{sw}}|. \tag{3.17}$$

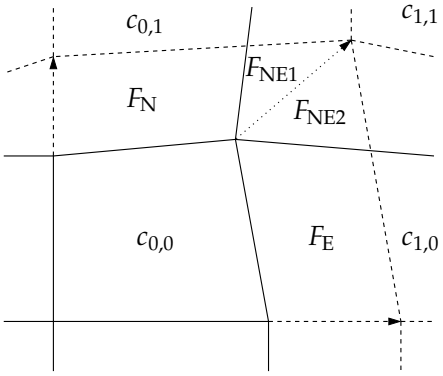


Figure 2: Upper right corner movement, dashed line – new mesh, solid line – old mesh.

Unfortunately, the signs of both errors cannot be determined easily for a general Hessian matrix and cell geometry. We can either directly evaluate both terms or use some approximation of the involved integrals (eg. (3.14)). In this case (distorted initial mesh), the complete analysis is impossible and we only make an upper estimate of the error magnitude. For certain simplified mesh patterns (such as the one presented in the next Section), it is possible to evaluate the error difference exactly and study the influence of the function and movement parameters on the error difference. Then, it is possible to explicitly identify, which method has a lower numerical error for given mesh and function parameters. In our case described in Fig. 2, we can at least make an upper estimate of the error difference magnitude:

$$|\Delta\epsilon| \leq \frac{1}{2\tilde{V}_{c_{0,0}}} \left| \frac{V_{F_{\mathrm{NE1}}+F_{\mathrm{NE2}}}}{V_{c_{1,1}}} I^{\mathbf{H}}_{c_{1,1}}(\mathbf{r}_{c_{1,1}}) - \frac{V_{F_{\mathrm{NE1}}}}{V_{c_{0,1}}} I^{\mathbf{H}}_{c_{0,1}}(\mathbf{r}_{c_{0,1}}) - \frac{V_{F_{\mathrm{NE2}}}}{V_{c_{1,0}}} I^{\mathbf{H}}_{c_{1,0}}(\mathbf{r}_{c_{1,0}}) \right.$$
$$\left. - I^{\mathbf{H}}_{F_{\mathrm{NE1}}+F_{\mathrm{NE2}}}(\mathbf{r}_{c_{1,1}}) + I^{\mathbf{H}}_{F_{\mathrm{NE1}}}(\mathbf{r}_{c_{0,1}}) + I^{\mathbf{H}}_{F_{\mathrm{NE2}}}(\mathbf{r}_{c_{1,0}}) \right|. \tag{3.18}$$

If the cells of the old computational mesh have similar geometry, the integrals over cells cancel out. The remaining terms can be approximated using a similar approach as in (3.14). This way, we obtain a simple formula for estimating the error difference magnitude which can show us whether the remapping method choice is important or not.

It is to be noted that in the opposite case of cell corner compression, ie. all its edges moving inwards, both methods are equivalent because all reconstructions are made using values from the cell itself. Therefore, it is not important whether we use a swept- or intersection-based representation of the fluxes in such case.

In a more general situation of a polygonal mesh, there can be a different number of cells sharing the corner vertex. This will result in similar error difference terms containing $N_C - 2$ corner triangles, where N_C is the number of such cells. In a similar manner, we can describe the situation where there are two swept regions with a triangular overlapping area, only the signs of the fluxes will be different.

3.3 Simplified orthogonality-preserving diagonal corner movement

To see the effects of mesh movement on the remapping error in more details, we reduce the above-presented situation so that the affected horizontal and vertical grid-lines move equally by the distance d in the directions of the axes (see Fig. 3). This simplified mesh movement corresponds to the tensor product mesh motion often used for testing of remapping methods [27]. The old mesh is equidistant and has square cells with the edge length l.

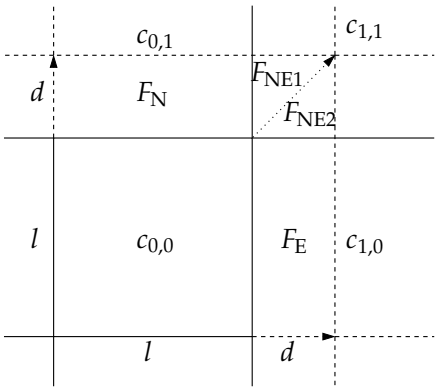


Figure 3: Diagonal movement of orthogonal corner, dashed line – new mesh, solid line – old mesh.

The difference between the methods is dependent on the area of the corner flux regions $V_{F_{\mathrm{NE1}}+F_{\mathrm{NE2}}}$ but also on other factors. However, here we have the possibility to investigate the dependence of the remapping error on a limited set of parameters, namely d, l, and the elements of $\mathbf{H}$. After substituting into (3.15), (3.16) and integrating, we obtain the following remapping error terms:

$$\epsilon_{\mathrm{sw}} = -\frac{d}{12\tilde{V}_{c_{0,0}}}\left[(2d^3+ld^2-2dl^2+l^3)(\rho_{xx}+\rho_{yy})+(3d^3+2ld^2-3dl^2)\rho_{xy}\right], \tag{3.19}$$

$$\epsilon_{\mathrm{int}} = -\frac{d}{12\tilde{V}_{c_{0,0}}}\left[(2d^3-ld^2-2dl^2+l^3)(\rho_{xx}+\rho_{yy})+(3d^3-6ld^2+3dl^2)\rho_{xy}\right], \tag{3.20}$$

where ρ_{xx}, ρ_{xy}, ρ_{yy} are the respective second derivatives, the elements of $\mathbf{H}$. These formulas can be simplified by rotating the coordinate system by $\pi/4$, which corresponds to the axis of symmetry for this problem:

$$x' = \frac{1}{\sqrt{2}}(x+y), \quad y' = \frac{1}{\sqrt{2}}(y-x). \tag{3.21}$$

We can transform the second derivatives similarly by multiplying them by the Jacobian matrix of the transformation,

$$\rho_{x'x'} = \frac{1}{2}(\rho_{xx}+\rho_{yy})+\rho_{xy}, \quad \rho_{x'y'} = \frac{1}{2}(\rho_{xx}-\rho_{yy}), \quad \rho_{y'y'} = \frac{1}{2}(\rho_{xx}+\rho_{yy})-\rho_{xy}. \tag{3.22}$$

After substituting the inverse transformation into (3.19), (3.20), we can see that the mixed derivative term cancels out and the matrix of second derivatives **H** has only two non-zero components. One in the direction of the movement and the other in the perpendicular direction. The relation between these two components plays the key role in determining which method produces larger remapping error. The error terms can now be expressed as

$$\epsilon_{\text{sw}} = -\frac{d}{24\tilde{V}_{c_{0,0}}}\left[(7d^3+4ld^2-7dl^2+2l^3)\rho_{x'x'}+(d^3-dl^2+2l^3)\rho_{y'y'}\right], \tag{3.23}$$

$$\epsilon_{\text{int}} = -\frac{d}{24\tilde{V}_{c_{0,0}}}\left[(7d^3-8ld^2-dl^2+2l^3)\rho_{x'x'}+(d^3+4ld^2-7dl^2+2l^3)\rho_{y'y'}\right]. \tag{3.24}$$

It is to be noted that this rotation of coordinates in the direction of movement is not useful for an arbitrary mesh movement. The mixed derivative is canceled out due to the symmetric initial conditions, in a general case there are many more factors that will affect the remapping error structure.

We can simplify both terms by dividing them by $K_1 = -dl^3\rho_{x'x'}/(24\tilde{V}_{c_{0,0}})$ and substituting $g = \rho_{y'y'}/\rho_{x'x'}$ as the ratio of the second derivative components and $t = d/l$ as the movement relative to the cell size. Due to the remapping stability condition [27] in this case the constraints of t are $0 < t < 0.5$. The derivative ratio g can be seen as a measure of eccentricity of the local second-order error. To determine which method performs better we can evaluate the error difference (3.17):

$$\begin{aligned}\Delta\epsilon = |K_1|\big(&|g(t^3-t+2)+7t^3+4t^2-7t+2| \\ &-|g(t^3+4t^2-7t+2)+7t^3-8t^2-t+2|\big).\end{aligned} \tag{3.25}$$

The sign of $\Delta\epsilon$ indicates which method performs better – this is shown in Fig. 4. We can see that for $|\rho_{y'y'}/\rho_{x'x'}| > 1$, the error difference $\Delta\epsilon < 0$ and thus the intersection method is more accurate than the swept region method in this area.

Concerning the magnitude of the error, we can approximate the error difference using the triangle inequality for the intersection error term and cancel the common terms out:

$$|\Delta\epsilon| \le \frac{d^2 l}{12\tilde{V}_{c_{0,0}}}\left|(3l-6d)\rho_{x'x'}+(2d-3l)\rho_{y'y'}\right|. \tag{3.26}$$

This can be also expressed in terms of total corner flux volume $V_{F_{\text{NE}}} = d^2$ and old cell volume $V_{c_{0,0}} = l^2$. After substituting $2ld = \tilde{V}_{c_{0,0}} - V_{c_{0,0}} - V_{F_{\text{NE}}}$, we obtain

$$|\Delta\epsilon| \le \frac{V_{F_{\text{NE}}}}{12V_{\tilde{C}}}\left|3(V_{F_{\text{NE}}}+2V_{c_{0,0}}-\tilde{V}_{c_{0,0}})\rho_{x'x'}-(\tilde{V}_{c_{0,0}}-4V_{c_{0,0}}-V_{F_{\text{NE}}})\rho_{y'y'}\right|. \tag{3.27}$$

This form of the error approximation by volumes only is useful because a corresponding formula can be applied also to non-rectangular cells, although the validity of this estimate for the general case cannot be easily verified. The dimension is that of mass. To obtain a dimensionless relative error difference estimate, this formula has to be divided by the cell mass.

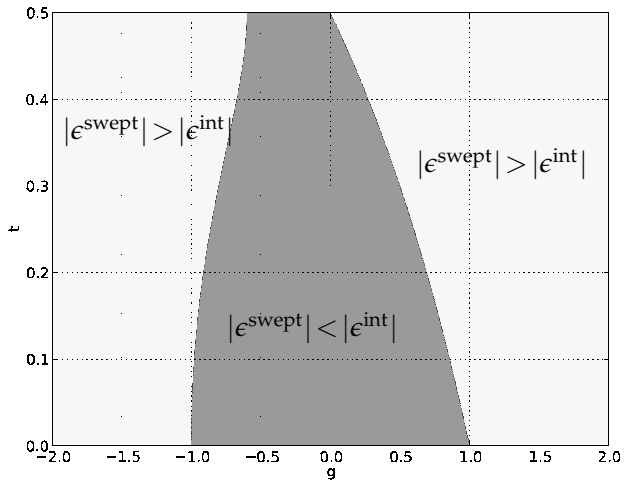


Figure 4: Two grid-lines motion – method preference. Different colors show regions where each method is more accurate.

3.4 Simplified diagonal corner movement - one vertex only

This type of movement concerns of a single node of a computational cell – the upper right vertex of the middle cell is moved in the diagonal direction, as shown in Fig. 5. This corresponds to the correction of the position of a single node, which is often encountered in mesh rezoning algorithms. It is expected to be more difficult to analyze, because not all edges remain orthogonal after the mesh rezoning and the resulting formulas are more complicated. Again, we evaluate the remapping error for the mesh configuration described here. Surprisingly, both errors can be simplified by using the diagonalization

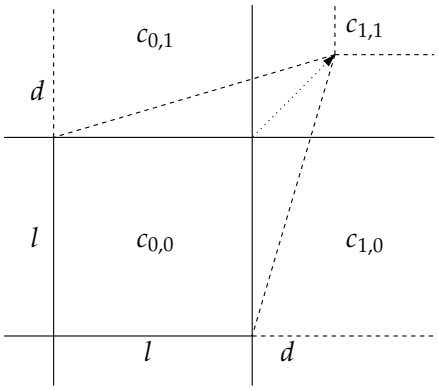


Figure 5: Diagonal one point corner movement, dashed line – new mesh, solid line – old mesh.

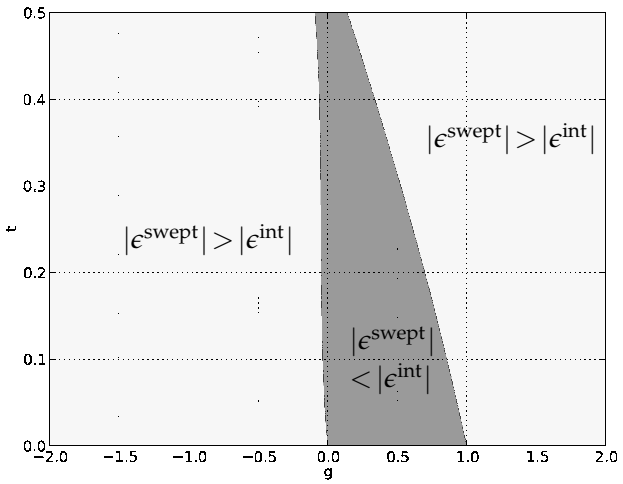


Figure 6: Single vertex movement – method preference. Different colors show regions where each method is more accurate.

process as described in (3.22):

$$\epsilon_{\text{sw}} = -\frac{d}{12(l+d)}\left[(2d^2-ld)\rho_{x'x'}+l^2\rho_{y'y'}\right], \tag{3.28}$$

$$\epsilon_{\text{int}} = -\frac{d}{12(l+d)^3}\left[(2d^4-d^3l-3d^2l^2+2dl^3)\rho_{x'x'}+(l^4-dl^3)\rho_{y'y'}\right]. \tag{3.29}$$

We can simplify them further by dividing by $K_2 = -d\rho_{x'x'}/(12(l+d)^3)$ and substituting t and g as in (3.25). The resulting error difference can be expressed as follows:

$$\Delta\epsilon = |K_2|\left[(1+t)^2|(2t^2-t+g)|-|g(t-1)-2t^4+t^3+3t^2-2t|\right]. \tag{3.30}$$

The sign of this error difference is shown in Fig. 6. As we can see, for this type of the mesh movement the swept-based method is more accurate approximately in the region defined as $0<\rho_{y'y'}/\rho_{x'x'}<1$, so the area is even smaller than in the previous example.

After approximating the absolute value of the error difference by the triangle inequality, we obtain the result in the following form:

$$|\Delta\epsilon| \le \frac{d^2l}{12(l+d)^3}\left|(3l^2-3ld-4d^2)\rho_{x'x'}-(3l^2+ld)\rho_{y'y'}\right|, \tag{3.31}$$

and after substituting for volumes where possible, we can write

$$|\Delta\epsilon| \le \frac{1}{24}\left(\frac{V_{F_{\text{NE}}}}{V_{\tilde{C}}}\right)^{\frac{3}{2}}\left|(9V_{c_{0,0}}-3\tilde{V}_{c_{0,0}}-5V_{F_{\text{NE}}})\rho_{x'x'}-(\tilde{V}_{c_{0,0}}+5V_{c_{0,0}}-V_{F_{\text{NE}}})\rho_{y'y'}\right|. \tag{3.32}$$

3.5 Hourglass edge movement

In the previous sections, we have analyzed movements where the swept region method produces higher numerical error due to the absence of the corner fluxes. Here, we examine the "hourglass" movement – rotating an edge around its center by moving its vertices in the opposite direction (see Fig. 7). This unwanted mesh patters is known to produce higher numerical error also when using the swept fluxes (due to the error resulting from the function reconstruction taken from a "wrong" cell in one part of the flux). Let us note that the analysis is performed for a smooth density function, as we did before. However, the largest error of the swept-based method would be obtained in case of a discontinuous function with the discontinuity oriented along the rotated edge because taking the reconstruction from just one cell in the entire swept region basically means that the discontinuity is ignored during flux construction.

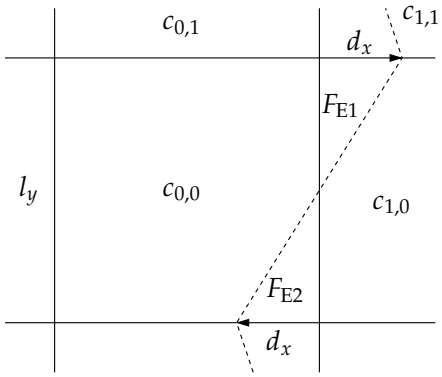


Figure 7: Hourglass movement, dashed line – new mesh, solid line – old mesh.

The situation depicted in the figure is again simplified, in a real-world remapping problem, the hourglass movement will be combined with corner and other movements and also the hourglass region does not have to be perfectly symmetric. The general expression for the remapping error terms is as follows:

$$\epsilon_{\mathrm{sw}} = \frac{1}{2\tilde{V}_{c_{0,0}}} \left(\frac{V_{F_{\mathrm{E1}}-F_{\mathrm{E2}}}}{V_{c_{1,0}}} I^{\mathbf{H}}_{c_{1,0}}(\mathbf{r}_{c_{1,0}}) - I^{\mathbf{H}}_{F_{\mathrm{E1}}-F_{\mathrm{E2}}}(\mathbf{r}_{c_{1,0}}) \right). \tag{3.33}$$

The swept method is constructed so that it perceives the self-intersecting polygon $F_{\mathrm{E1}} - F_{\mathrm{E2}}$ as having volume close to zero. However, based on the sign of its volume it decides which reconstruction will be used for the particular flux. In this case, we presume the region F_{E1} is slightly larger and so values from cell $c_{1,0}$ are used. The intersection method is not prone to such vulnerability because both regions are treated separately:

$$\epsilon_{\mathrm{int}} = \frac{1}{2\tilde{V}_{c_{0,0}}} \left(\frac{V_{F_{\mathrm{E1}}}}{V_{c_{1,0}}} I^{\mathbf{H}}_{c_{1,0}}(\mathbf{r}_{c_{1,0}}) - I^{\mathbf{H}}_{F_{\mathrm{E1}}}(\mathbf{r}_{c_{1,0}}) - \frac{V_{F_{\mathrm{E2}}}}{V_{c_{0,0}}} I^{\mathbf{H}}_{c_{0,0}}(\mathbf{r}_{c_{0,0}}) + I^{\mathbf{H}}_{F_{\mathrm{E2}}}(\mathbf{r}_{c_{0,0}}) \right). \tag{3.34}$$

We can also express the error difference magnitude, similarly as for the corner movement:

$$|\Delta\epsilon| \leq \frac{V_{F_{E2}}}{2\tilde{V}_{c_{0,0}}} \left| \frac{I^{\mathbf{H}}_{c_{1,0}}(\mathbf{r}_{c_{1,0}})}{V_{c_{1,0}}} - \frac{I^{\mathbf{H}}_{F_{E2}}(\mathbf{r}_{c_{1,0}})}{V_{F_{E2}}} - \frac{I^{\mathbf{H}}_{c_{0,0}}(\mathbf{r}_{c_{0,0}})}{V_{c_{0,0}}} + \frac{I^{\mathbf{H}}_{F_{E2}}(\mathbf{r}_{c_{0,0}})}{V_{F_{E2}}} \right|. \tag{3.35}$$

We can see that any dependence on the region F_{E1} was eliminated, the formula only concerns the region for which an incorrect reconstruction is used. If the swept method would originally use the values from cell $c_{0,0}$ instead, the formula would contain only F_{E1}-related terms.

In the idealized situation depicted in Fig. 7, both neighboring cells have the same geometry, thus we can claim that $I^{\mathbf{H}}_{c_{1,0}}(\mathbf{r}_{c_{1,0}}) \approx I^{\mathbf{H}}_{c_{0,0}}(\mathbf{r}_{c_{0,0}})$. Also, both halves of the hourglass polygon are symmetrically positioned relative to the centroid of the cell they are located in. The second-order error function $\mathbf{r}^T\mathbf{H}\mathbf{r}$ is centrally symmetric, therefore $I^{\mathbf{H}}_{F_{E1}}(\mathbf{r}_{c_{1,0}}) = I^{\mathbf{H}}_{F_{E2}}(\mathbf{r}_{c_{0,0}})$. If we apply both identities to (3.34), we can see that the intersection-based method is third-order accurate in this case. On the other hand, the swept-based method produces some error in this case, namely:

$$\epsilon_{\text{sw}} = \frac{d_x}{12}\left(d_x\rho_{xx} + l_y\rho_{xy}\right), \quad \epsilon_{\text{int}} = 0, \tag{3.36}$$

where d_x and l_y are the horizontal node movement and the vertical cell dimension, respectively (see Fig. 7). This formula shows that there is one term linear in mesh movement and dependent on the mixed derivative, and an additional quadratic term. This expression is simple to evaluate and does not require further analysis. In this special case, the swept region remapping is always worse than the intersection-based method.

3.6 Shear movement

This movement is an extension of the hourglass mesh motion described in the previous section. This pattern can be encountered in real hydrodynamic simulations (for example, at the material interfaces), so we present this situation here briefly. The bottom mesh nodes both move to the left by the distance d_x while the upper nodes move in the opposite direction, as shown in Fig. 8.

In this case, the intersection-based method produces no second-order error again. The swept-based method can be also as accurate, however, this depends on the selection of the cell from which the reconstruction is used. If the same side is selected on both moving edges (eg. cell $c_{1,0}$ for the right edge and $c_{0,0}$ for the left edge), this method produces numerical error as follows:

$$\epsilon_{\text{sw}} = \frac{d_x}{6}\left(d_x\rho_{xx} + l_y\rho_{xy}\right). \tag{3.37}$$

However, if the opposite neighbor is used for function reconstruction (e.g. cell $c_{1,0}$ for the right edge and $c_{-1,0}$ for the left edge or $c_{0,0}$ for both), the error terms for both sides cancel out. This is caused by the symmetric positioning of both swept regions relative to the cell from which the reconstruction is obtained.

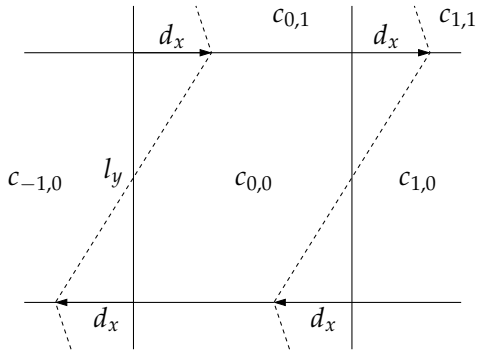


Figure 8: Symmetric cell shear, dashed line – new mesh, solid line – old mesh.

The shear movement consisting of only one edge (upper or lower) moving sideways is not detailed here, because in such case, there are only simple fluxes present and both remapping methods are equivalent.

3.7 Cell rotation

The last simplified movement presented in this paper is the cell rotation around its geometrical centroid shown in Fig. 9. The movement looks complicated to analyze as there are many intersections of the neighboring cells (or self-intersecting regions in the case of the swept-based method). However, the resulting formulas can be simplified due to the symmetric nature of this movement. The total volume of the cell does not change during this movement.

We will describe the results for an equidistant mesh with rectangular cells with dimensions l_x, l_y, as can be seen in Fig. 9. After using (3.13) for every flux and simplifying the results, we obtain the following terms for intersection- and swept-based method error

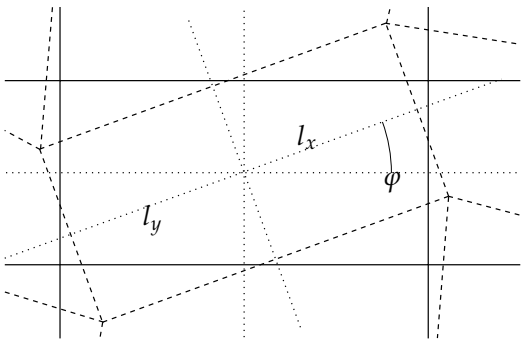


Figure 9: Cell rotation around its centroid by angle φ, dashed line – new mesh, solid line – old mesh.

respectively:

$$\epsilon_{\text{int}} = \frac{1}{24}\left(l_x^2 - l_y^2\right)\sin\varphi\left(\left(\rho_{xx}-\rho_{yy}\right)\sin\varphi - \rho_{xy}\cos\varphi\right), \tag{3.38}$$

$$\epsilon_{\text{sw}} = \frac{1}{12}\left(1-\cos\varphi\right)\left(\left(\rho_{xx}l_x^2 + \rho_{yy}l_y^2\right)\cos\varphi + \rho_{xy}\left(l_x^2 - l_y^2\right)\sin\varphi\right). \tag{3.39}$$

We can see that the error of the intersection-based method is dependent on the difference of the squares of the cell dimensions and therefore it will approach zero when the cell dimensions are close to equal, independently on the values of the remapped distribution. The swept-based method does not have such properties.

To see the effect of the second derivatives on the remapping error, we can express the terms in the second-order gauge coordinates of the Hessian (eg. the coordinates in which **H** is diagonal). We presume that **H** is a symmetric matrix, so this transformation involves rotation of the coordinate system by the angle ψ. This angle is computed by solving the following equation:

$$\left(\rho_{xx}-\rho_{yy}\right)\sin 2\psi = 2\rho_{xy}\cos 2\psi, \quad \psi \in \left(-\frac{\pi}{4}, \frac{\pi}{4}\right\rangle. \tag{3.40}$$

Note that this transformation can change the order of the eigenvalues. The derivatives in the principal directions ρ_{dd}, ρ_{pp} (the eigenvalues of **H**) can be expressed as follows:

$$\rho_{dd} = \rho_{xx}\cos^2\psi + 2\rho_{xy}\sin\psi\cos\psi + \rho_{yy}\sin^2\psi, \tag{3.41}$$

$$\rho_{pp} = \rho_{xx}\sin^2\psi - 2\rho_{xy}\sin\psi\cos\psi + \rho_{yy}\cos^2\psi. \tag{3.42}$$

By substituting these transformations in (3.38), (3.39), we yield the following error terms:

$$\epsilon_{\text{int}} = \frac{1}{24}\left(l_x^2 - l_y^2\right)\left(\rho_{dd}-\rho_{pp}\right)\sin\varphi\sin\left(\varphi - 2\psi\right), \tag{3.43}$$

and

$$\epsilon_{\text{sw}} = \frac{1}{24}\left(1-\cos\varphi\right)\left(\left(\rho_{dd}+\rho_{pp}\right)\left(l_x^2+l_y^2\right)\cos\varphi + \left(\rho_{dd}-\rho_{pp}\right)\left(l_x^2-l_y^2\right)\cos\left(\varphi-2\psi\right)\right). \tag{3.44}$$

Here, we can see another beneficial properties of the intersection-based method. If $\rho_{dd} = \rho_{pp}$, ie. the second derivative is radially symmetric, this method produces no second-order error terms again, regardless of the cell dimensions. It is also accurate when the principal directions are aligned with the axis of symmetry of the movement (half the rotation angle, so $\varphi = 2\psi$).

As we have said before, for a square cell the intersection-based method is third-order accurate. In this case the error term of the swept-based method can be simplified as:

$$\epsilon_{\text{sw},\square} = \frac{1}{12}l^2\left(\rho_{dd}+\rho_{pp}\right)\left(1-\cos\varphi\right)\cos\varphi, \tag{3.45}$$

where $l^2 = l_x^2 = l_y^2$. It is interesting that due to the involved symmetry, even the swept-based method has its error independent on the principal directions orientation. It is proportional to the mean curvature, cell area, and the rotation angle.

4 Numerical examples

In this section, we compare the performance of both remapping methods by applying them on a known distribution of discrete values. Analytical initial functions are selected so that they can be integrated over any cell geometry for the error assessment. Similarly, the accuracy of our second-order error estimation can also be evaluated and verified, if it equals to the error produced by the remapping of a quadratic function. In the first example, we can directly compare the numerical results with the analytic formulas. The error estimate is, however, not expected to be exact for later examples, because the function profiles are distorted after the first remapping step and contain higher-order terms. Moreover, when the entire mesh moves during rezoning, more complicated mesh motions than analyzed are present. In the last example, we apply both methods in the context of a full ALE fluid dynamics simulation and compare their results.

The L_1 error is used to measure the total remapping error over the whole domain and it is calculated as follows:

$$L_1 = \frac{\sum_{c\in C} |f_c - \bar{f}_c|}{\sum_{c\in C} \bar{f}_c}, \quad \bar{f}_c = \frac{\int_c f(\mathbf{r})\, dV}{V_c}, \tag{4.1}$$

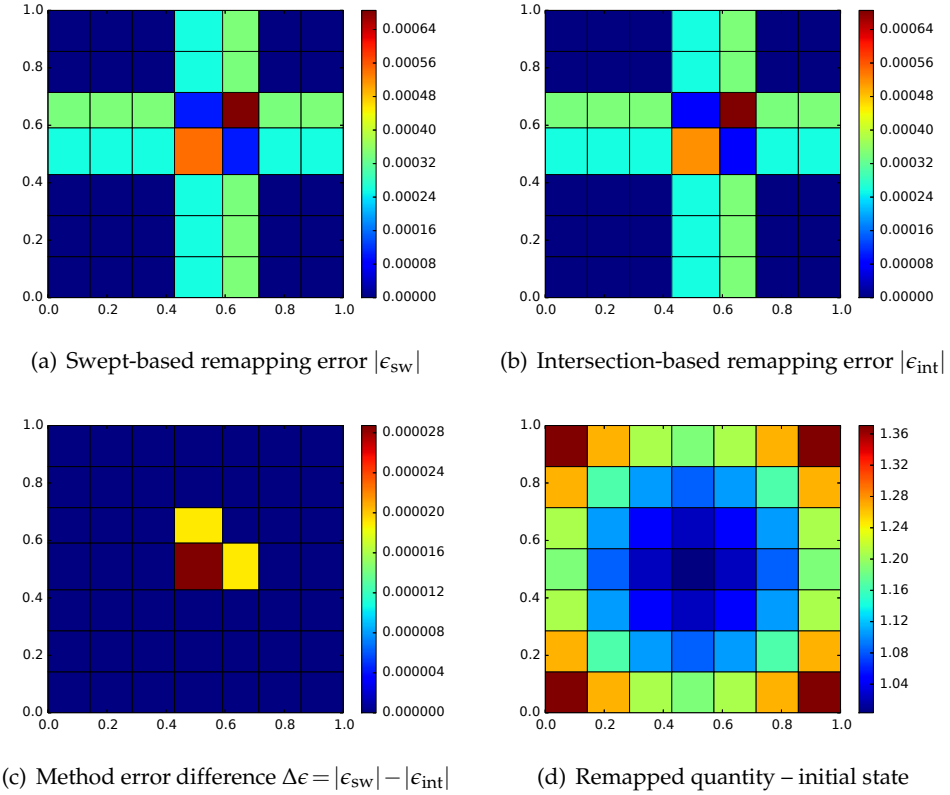
where C is the set of all computational cells, f_c represents the remapped value in the cell and $\bar{f}_c$ is the analytical mean value in the cell.

4.1 One remapping step on a small mesh

The first example involves the smallest computational mesh where there is at least one cell unaffected by any boundary effects during a single remapping step. The least squares minimization of the error functional [17] used for obtaining the function derivative approximation employs the values from the neighboring cells, so for the calculation of the second derivative used here, two such layers are needed. Therefore the smallest mesh with at least one cell unaffected has 7×7 cells.

We have selected $f=(x-1/2)^2+(y-1/2)^2+1$ as the initial distribution function. Analytical integration of this function is performed over the computational cells to get the initial values. Then, the upper right corner of the middle cell moves by $\mathbf{r}=(0.02,0.02)$. Other points on the same index-lines are moved in the horizontal or vertical direction only to keep all edges orthogonal (similar movement as in Fig. 2). The distribution is then remapped on this new computational mesh.

The remapping errors for both methods are shown in Fig. 10. Both methods are equivalent and produce the same error in most of the cells except the four cells in the center



(a) Swept-based remapping error $|\epsilon_{sw}|$ (b) Intersection-based remapping error $|\epsilon_{int}|$

(c) Method error difference $\Delta\epsilon = |\epsilon_{sw}| - |\epsilon_{int}|$ (d) Remapped quantity – initial state

Figure 10: Numerical error of both remapping methods compared. Initial distribution function $(x-1/2)^2+(y-1/2)^2+1$, 7×7 mesh, one step of a diagonal movement.

Table 1: Numerical error of both remapping methods in the central and surrounding cells. Initial distribution function $(x-1/2)^2+(y-1/2)^2+1$, 7×7 mesh, one step of a diagonal movement.

	ϵ_{sw}	ϵ_{int}
$c_{4,4}$	$-5.460\cdot10^{-4}$	$-5.173\cdot10^{-4}$
$c_{5,4}$	$1.033\cdot10^{-4}$	$8.421\cdot10^{-5}$
$c_{4,5}$	$1.033\cdot10^{-4}$	$8.421\cdot10^{-5}$
$c_{5,5}$	$6.857\cdot10^{-4}$	$6.857\cdot10^{-4}$

of the domain. In the error difference plot, we can see that the swept-based remapping method is less accurate due to the missing corner flux. For exact values of the error terms in four central cells, see Table 1. As we can see, the swept-based method produces higher error in each of these cells. In this particular case, there is only second-order error present so the derived error estimate (3.13) equals the numerical remapping error exactly.

4.2 Remapping of 1D symmetric distribution on a diagonally moving mesh

In the next test, we are remapping a distribution that does not depend on y, namely $f=(x+1)^2$. It undergoes a repeated diagonal mesh movement, a so-called tensor product rezoning [26]. Here, both horizontal and vertical grid-lines move in the perpendicular directions at different velocities. This produces a diagonal mesh deformation, although the cells always stay rectangular. The positions of vertices at step n are defined as follows:

$$\begin{aligned} &x_i^n = x_i^0(1-d^n)+\left(x_i^0\right)^3 d^n, \quad y_i^n = y_i^0(1-d^n)+\left(y_i^0\right)^2 d^n, \\ &d^n = \tfrac{1}{2}\sin(2\pi t^n), \quad t^n = n/n_{\max}, \quad n_{\max}=100, \end{aligned} \tag{4.2}$$

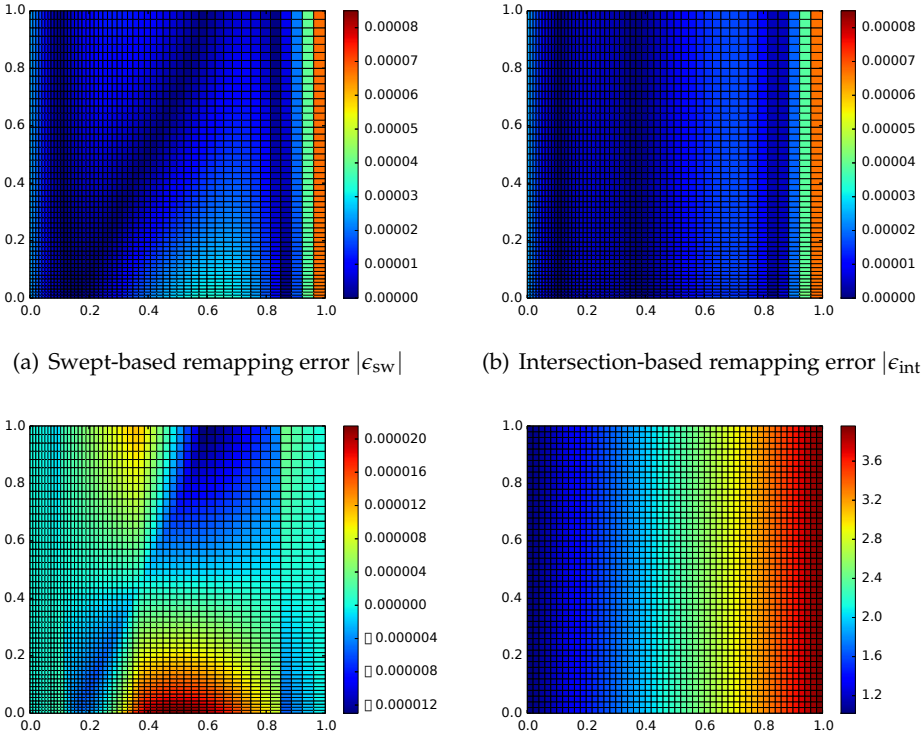
where (x_i^n, y_i^n) is the actual position of node i and (x_i^0, y_i^0) is the initial position. In general, the superscript 0 denotes the initial grid, n the actual grid, and $n_{\max}$ is the total number of the remapping steps required for a completion of the full movement period. The boundary vertices slide along the borderline in the same manner as the inner vertices.

A visible difference between the error distributions for each method after 25 remapping steps (the mesh is most deformed at this moment) is shown in Fig. 11. We can compare the L_1 error of both methods. The numerical error of the swept-based method is $5.786\cdot10^{-6}$, which is slightly worse than the numerical error of the intersection-based method $4.920\cdot10^{-6}$. The swept-based method remapping error varies in the vertical direction due to the asymmetric mesh movement, while the numerical error distribution of the intersection-based method preserves the 1D character of the initial distribution. It is to be noted that in this calculation, the analytical Neumann boundary conditions are imposed on the first derivative:

$$\frac{\partial f}{\partial x}(0)=\frac{\partial f}{\partial x}(1)=2x+2, \quad \frac{\partial f}{\partial y}(0)=\frac{\partial f}{\partial y}(1)=0. \tag{4.3}$$

They are used to reduce the inaccuracy in the least squares approximation of the first derivative which arises at the boundary of the domain.

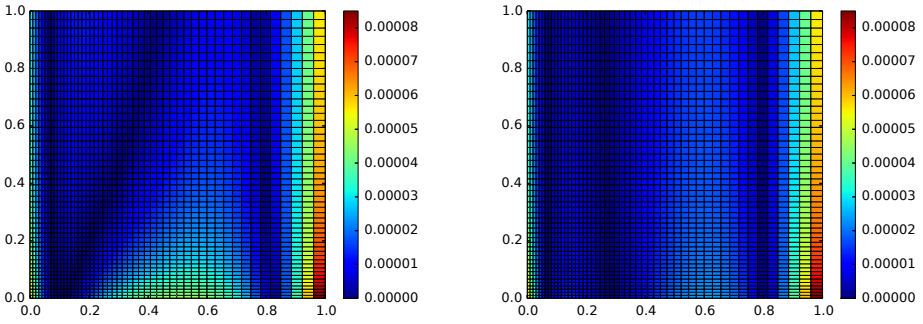
If the remapping steps are repeated it is no longer possible to maintain the exactness of our error estimate. In each successive step, the distribution being remapped is already slightly distorted from the previous one and does not exactly correspond to the second-order polynomial function used in the error analysis. The numerical remapping error presented in Fig. 11 is calculated from the integral mean values of the known analytical function in the computational cells. However, our second-order estimate uses the values from the previous remapping step as a reference and therefore it differs from the exact remapping error. In Fig. 12, a cumulative second-order error estimate (sum of error estimates (3.13) over time steps in each computational cell) for this remapping calculation is presented. The pattern is similar as in the analytical error calculation (compare Fig. 11), although here is a clearly visible shift in the direction of the mesh movement. Also, note the vertical variation of the error estimate values at the right boundary.



(a) Swept-based remapping error $|\epsilon_{\text{sw}}|$ (b) Intersection-based remapping error $|\epsilon_{\text{int}}|$

(c) Method error difference $\Delta\epsilon = |\epsilon_{\text{sw}}| - |\epsilon_{\text{int}}|$ (d) Remapped quantity – initial state

Figure 11: Numerical remapping error of both methods compared. Initial distribution function $(x+1)^2$, 50×50 mesh, step $n=25$ of the tensor product movement.



(a) Swept-based remapping (b) Intersection-based remapping

Figure 12: Estimate of the second-order remapping error. Initial distribution function $(x+1)^2$, 50×50 mesh, step $n=25$ of the tensor product movement.

4.3 Cyclic remapping of a smooth, radially symmetric distribution

In the previous example, we have investigated the effect of the diagonal mesh movement on a 1D distribution and we have seen that the swept-based method tends to distort the symmetry of the resulting error. Now we will demonstrate the effect of a diagonal movement on a radially symmetric distribution:

$$f=10\Big(\mathrm{sinc}(24r-8)+2\Big)=10\left(\frac{\sin(24r-8)}{24r-8}+2\right), \tag{4.4}$$

$$r=\sqrt{(x-0.5)^2+(y-0.5)^2}. \tag{4.5}$$

This function is smooth and has a circular-shaped peak around the central area. The same mesh movement as in the previous example is used, with the only difference that here all 100 steps are performed. This means that the whole period of the movement will be accomplished and the final mesh will be the same as the initial one (hence the term "cyclic remapping").

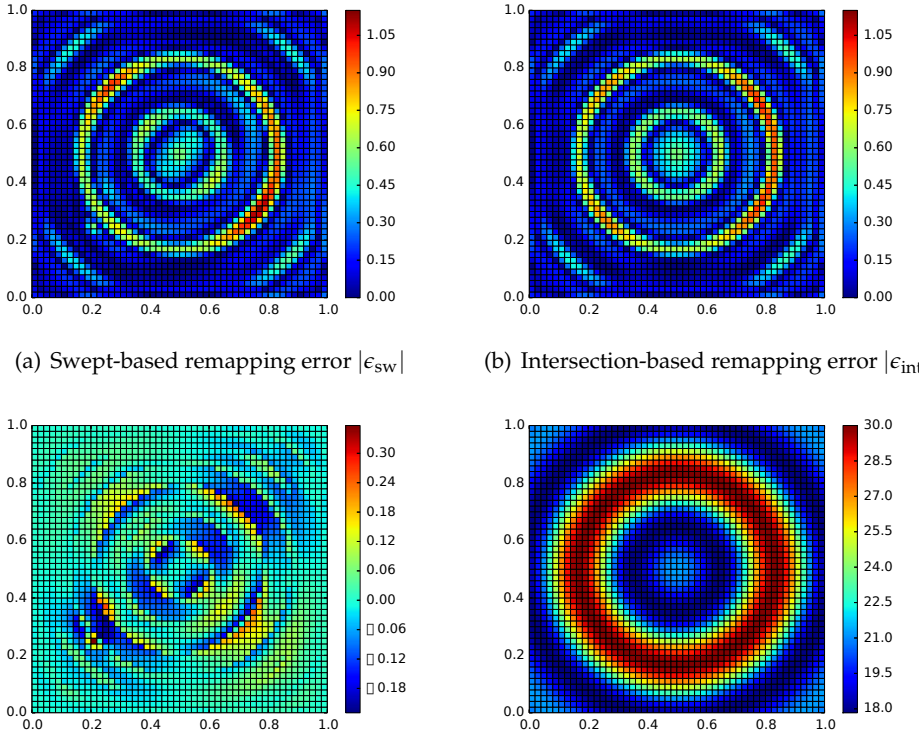
The results for a 50×50 mesh are shown in Fig. 13. We can see that the intersection method produces a more symmetric distribution of the remapping error again, although the overall remapping error for the whole domain is actually slightly higher (the L_1 error in this case is $9.532\cdot10^{-3}$ for the swept-based method and $9.586\cdot10^{-3}$ for the intersection method).

The amount of asymmetry introduced by the remapping of the distribution can be quantified by calculating the L_1 error in the 4 quadrants of the computational mesh separately. This is shown in Table 2. As we can see, the difference between the maximal and minimal quadrant L_1 error is significantly higher for the swept-based method, which confirms the asymmetry seen in Fig. 13.

Table 2: L_1 remapping error of both remapping approaches $m=$sw/int for quadrants 1-4 of the domain (with origin at $(0.5,0.5)$, numbering starts from upper right quadrant and going counter-clockwise). Initial distribution function $10\big(\mathrm{sinc}(24r-8)+2\big)$, 50×50 mesh, step $n=100$ of the tensor product movement.

m	sw	int
$L^1_{1,m}$	$9.351\cdot10^{-3}$	$1.045\cdot10^{-2}$
$L^2_{1,m}$	$9.305\cdot10^{-3}$	$8.676\cdot10^{-3}$
$L^3_{1,m}$	$8.056\cdot10^{-3}$	$8.892\cdot10^{-3}$
$L^4_{1,m}$	$1.142\cdot10^{-2}$	$1.033\cdot10^{-2}$
$\lvert\max_{Q=1..4}(L^Q_{1,m})-\min_{Q=1..4}(L^Q_{1,m})\rvert$	$3.364\cdot10^{-3}$	$1.774\cdot10^{-3}$

In Table 3, we can see the convergence study for both remapping approaches. As we can see, both approaches exhibit second order convergence in the L_1 norm for the smooth function and their total numerical L_1 errors are very close. We can also see the standard



(a) Swept-based remapping error $|\epsilon_{\text{sw}}|$

(b) Intersection-based remapping error $|\epsilon_{\text{int}}|$

(c) Method error difference $\Delta\epsilon = |\epsilon_{\text{sw}}| - |\epsilon_{\text{int}}|$

(d) Remapped quantity – initial state

Figure 13: Remapping error of both methods compared. Initial distribution function $10\left(\text{sinc}\left(24r-8\right)+2\right)$, 50×50 mesh, step $n=100$ of the tensor product movement.

Table 3: L_1 remapping error and its σ_{L_1} standard deviation (in %) for smooth function $10\left(\text{sinc}\left(24r-8\right)+2\right)$ and tensor product movement.

mesh size	$n_{\max}$	$L_{1,\text{sw}}$	$L_{1,\text{int}}$	$\sigma_{L_1}^{\text{sw}}$	$\sigma_{L_1}^{\text{int}}$
50×50	100	$9.532\cdot 10^{-3}$	$9.586\cdot 10^{-3}$	12.7	8.43
100×100	200	$2.337\cdot 10^{-3}$	$2.392\cdot 10^{-3}$	9.2	6.45
200×200	400	$5.571\cdot 10^{-4}$	$5.561\cdot 10^{-4}$	7.4	6.45
400×400	800	$1.266\cdot 10^{-4}$	$1.195\cdot 10^{-4}$	5.8	6.54

relative error deviation over the domain quadrants,

$$\sigma_{L_1}^{m}=\frac{1}{L_{1,m}}\sqrt{\frac{1}{4}\sum_{Q=1}^{4}\left(L_{1,m}^{Q}-L_{1,m}\right)^2}, \tag{4.6}$$

where $L_{1,m}^{Q}$ represents the numerical error of the remapping method m in one particular

Table 4: Time of simulation (in seconds) of both remapping approaches for smooth function $10\left(\text{sinc}(24r-8)+2\right)$ and tensor product movement for different mesh resolutions.

mesh size	$t_{sw}[s]$	$t_{int}[s]$
50×50	0.69	2.11
100×100	5.05	16.54
200×200	41.10	134.81
400×400	322.63	1069.46

quadrant Q. As we can see, the error deviation remains approximately constant for the intersection-based approach, the swept-based approach produces (typically) higher error deviation and is improving with increasing mesh resolution. For the highest mesh resolution, the deviations of both approaches are very close.

In Table 4, we can see the comparison of the computational times for different mesh resolutions. The tests were performed in the context of a simple research remapping code written in C++ on a standard Intel Core 3.4 GHz machine. As we can see, in our particular implementation, the intersection-based method is approximately three times slower than the swept-based approach. The main reason is the computation or more fluxes – we need to compute the 16 fluxes per cell (positive and negative flux for each neighbor, including the corner neighbors), while only 4 fluxes per cell (for edge neighbors only) are needed for the swept-based method. Moreover, polygon intersection has to be done for each flux, which is not needed for the swept remap. The rest of the process (density reconstruction, update of geometrical quantities, etc.) remains the same in both approaches.

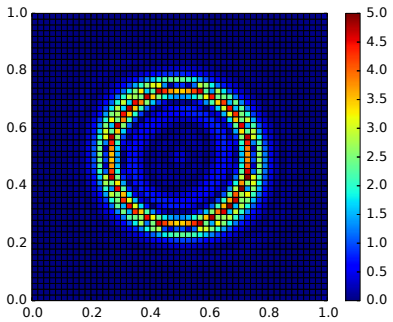
4.4 Cyclic remapping of a discontinuous, radially symmetric distribution

In the previous example, we have seen the convergence test in case of continuous radially-symmetric density function. It has been demonstrated that both methods have second order of convergence and their total numerical errors are comparable. The symmetry of the solution have been studied via the deviation of the error in all quadrants, this value was decreasing with the mesh resolution. In this section, we present similar test, in which the radially-symmetric double-exponential function contains a discontinuity as follows,

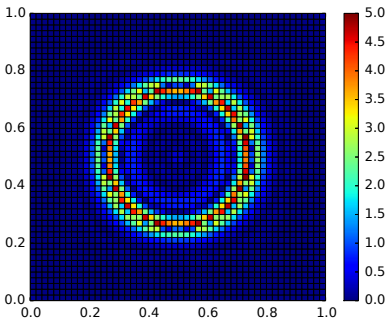
$$f(r)=\begin{cases} 1+e^{10r}, & \text{for } x\leq 1/4, \\ 1+e^{6r-1/4}, & \text{otherwise,} \end{cases} \tag{4.7}$$

where the radius r is defined as in (4.5). This test is taken from [4]. The results are shown in Fig. 14. In the error profiles, we can observe stronger asymmetry for the swept-based approach, especially in the diagonal direction corresponding to the mesh motion. This is even better visible in the plot of the error difference.

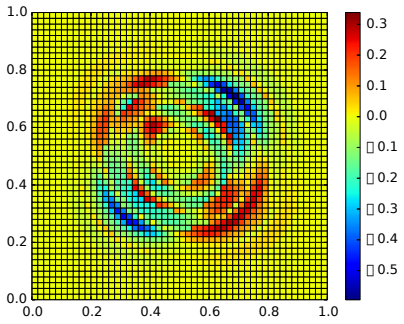
In Table 5, we can see the convergence study for both remapping approaches. As we can see, both approaches converge with just the first order for the discontinuous function,



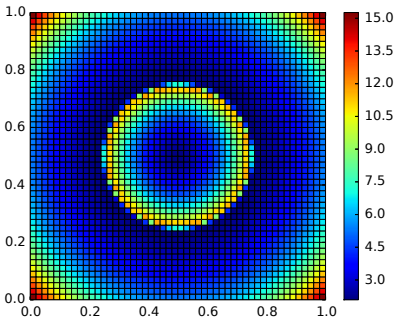
(a) Swept-based remapping error $|\epsilon_{sw}|$



(b) Intersection-based remapping error $|\epsilon_{int}|$



(c) Method error difference $\Delta\epsilon = |\epsilon_{sw}| - |\epsilon_{int}|$



(d) Remapped quantity – initial state

Figure 14: Remapping error of both methods compared. Initial distribution function – double exponential, 50×50 mesh, step $n=100$ of the tensor product movement.

Table 5: L_1 remapping error and its σ_{L_1} standard deviation (in %) for discontinuous double-exponential function and tensor product movement.

mesh size	n_{max}	$L_{1,sw}$	$L_{1,int}$	$\sigma_{L_1}^{sw}$	$\sigma_{L_1}^{int}$
50×50	100	$7.702\cdot10^{-2}$	$7.873\cdot10^{-2}$	9.4	2.7
100×100	200	$4.859\cdot10^{-2}$	$4.940\cdot10^{-2}$	10.7	2.9
200×200	400	$2.930\cdot10^{-2}$	$2.982\cdot10^{-2}$	10.9	2.9
400×400	800	$1.750\cdot10^{-2}$	$1.782\cdot10^{-2}$	10.9	2.8

their total numerical errors are still very close. We can also see that the error deviation remains approximately constant for both methods, the value is significantly higher for the swept-based approach. This confirms our suspicion that the swept-based approach can violate symmetry stronger. This result was not observed so strongly for the previous smooth function, as the piecewise linear density reconstruction approximates the smooth function reasonably well and the error of the swept-based approach resulting from wrong

function approximation in the diagonal direction decreases. However, in the case of discontinuous function, this error is always present (due to erroneous density reconstruction close to the discontinuity) and leads to the symmetry violation.

Regarding the cost of the simulations, the results are almost identical with the results presented in Table 4. The whole remapping process is same as for the smooth function, the only differences are the produced density slopes, which do not affect the simulation time.

4.5 Application in an ALE simulation – Sedov test

In the previous static-remapping tests, we have seen the violation of symmetry significantly higher for the swept-based remapping approach. In this section, we demonstrate that the same behavior can be seen in case of a full hydrodynamic simulation. Due to its symmetric nature, we have chosen the standard Sedov explosion problem [32], which is often used for assessing the performance of hydrocodes. We are aware that the error analysis was performed for smooth functions while the analytic solution of the Sedov problem contains a discontinuity. However, as the numerical solution is discrete we can still see that the behavior of both methods differs in a similar manner as before.

We will compare the remapping methods in the context of a full ALE fluid dynamics simulation consisting of a Lagrangian stage (where the fluid dynamics PDEs are numerically solved) and a mesh smoothing and remapping stage. Note that the produced error is affected mainly by the Lagrangian part of the ALE algorithm. However, some differences resulting from the used remapping method can be also observed.

The underlying equations are the two-dimensional compressible Euler equations:

$$\frac{1}{\rho}\rho_t+\nabla\cdot\mathbf{u}=0, \tag{4.8}$$

$$\rho(\mathbf{u}_x)_t+\nabla p=0, \quad \rho(\mathbf{u}_y)_t+\nabla p=0, \tag{4.9}$$

$$\rho\varepsilon_t+p\nabla\cdot\mathbf{u}=0, \tag{4.10}$$

where ρ represents fluid density, p pressure, $\mathbf{u}$ velocity vector, and ε specific internal energy. The t subscript denotes the partial time derivative, whereas $\mathbf{u}_x$ and $\mathbf{u}_y$ are the velocity vector components in the reference spatial Cartesian coordinates. This system is complemented with the ideal gas equation of state,

$$p=\rho\varepsilon(\gamma-1). \tag{4.11}$$

The Lagrangian scheme employs a staggered predictor-corrector scheme [6] with bulk artificial viscosity [8] to solve these equations. The Winslow rezoning method [16] is employed to keep the computational mesh smooth. For quantity remapping, the flux based approach described in [21] is used. In this approach, the (intersection-based or swept-based) fluxes of mass are constructed by the integration of the limited density reconstruction. Internal energy is remapped in a similar flux form, its fluxes are constructed

by attaching the limited internal energy reconstruction values to the mass fluxes. The nodal mass is remapped, where the inter-nodal mass fluxes are interpolated from the inter-cell ones as in [30]. The nodal velocity (momentum) is remapped in a similar flux form, the momentum fluxes are constructed by attaching the reconstructed velocity to the inter-nodal mass fluxes. To guarantee total energy conservation, the standard energy fix [3] is performed.

The particular initial data are taken from [31]. In this test, the overall energy is released from the single computational cell closest to the origin of the coordinate system while the initial values of other quantities are constant in the whole domain, namely:

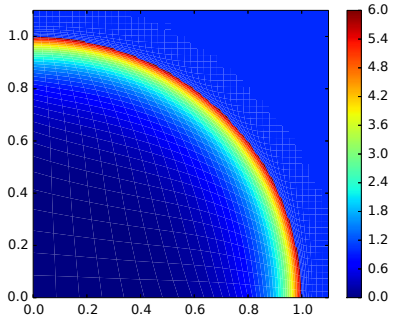
$$\rho = 1.0, \quad p = 0.4 \cdot 10^{-14}, \quad \mathbf{u} = \mathbf{0}, \quad \gamma = 1.4, \quad \varepsilon_{0,0} = 409.7, \quad t = 1.0,$$

where $\epsilon_{0,0}$ is the specific internal energy in the bottom left computational cell, and t simulation time. This corresponds to the total blast energy $E_{\text{blast}} = 0.2448$. The initial computational mesh contains 45^2 uniformly-distributed cells in the $\langle 0,1.1 \rangle^2$ domain. Any numerical problems with negative internal energy are avoided by timestep reduction in the Lagrangian phase, and by limiting the internal energy reconstruction and subcycling in the remapping stage.

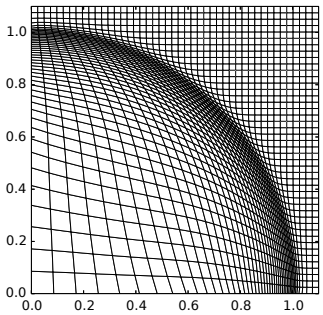
The result is a radially symmetric shock wave shown in Fig. 15. The advantage of this problem is that the analytical solution is known [13] and it is shown sampled on the initial mesh for comparison. The mesh rezoning algorithm used here is the Winslow smoothing [16], applied after every 10 Lagrangian steps (ALE10 regime). The solution is robust, but diffusive. The density shown in Fig. 15 is remapped using the intersection-based algorithm – the difference between both methods is very small and cannot be seen on such plot. Also the L_1 error is similar, $2.061 \cdot 10^{-1}$ for the swept-based method and $2.099 \cdot 10^{-1}$ for the intersection-based method.

The differences between the methods are better visible, when we plot the radial distribution of the density error in the computational cells. This is shown in Fig. 16. We can see that although the overall error average is similar, the swept-based method shows higher variance on the trailing edge of the shock wave. This is caused by the inward rezoning movement counteracting the Lagrangian expansion. For some of the cells this happens in the diagonal direction, which is one of the weak points of the swept-based method.

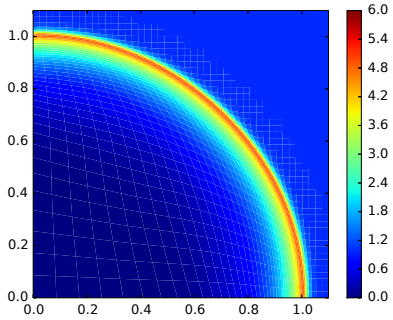
In Fig. 17, we can see the angular distribution of the density error for both methods. To emphasize the differences, only cells between $r = 0.9$ and $r = 0.99$ (distance of the centroid from the origin of the coordinate system) are shown. As we can see, the overall behavior is similar for both methods, the error is slightly lower in the diagonal direction than in the direction of the axes. Let us note that this variance does not result from the boundary conditions, which was confirmed by an additional simulation of the Sedov problem performed on a full $\langle 0, 2\pi \rangle$ domain, but results from the variance of the gradients of fluid quantities and mesh edges directions over the domain. However, it confirms our conclusions from the previous tests – the swept-based approach can produce larger variance of the numerical error while the error of the intersection-based algorithm is distributed more uniformly.



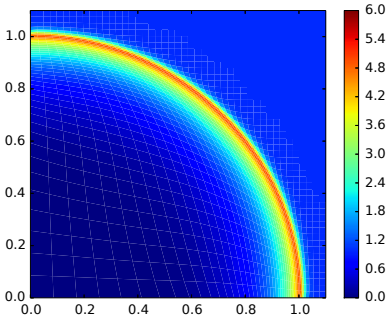
(a) Sampled analytical solution



(b) Computational mesh



(c) Swept, Winslow rezoning



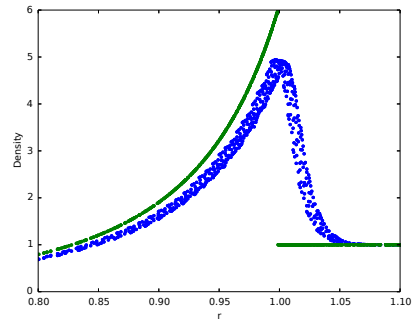
(d) Intersections, Winslow rezoning

Figure 15: Sedov problem, 2D density profile (analytical and numerical), 45×45 mesh, $t=1.0$.

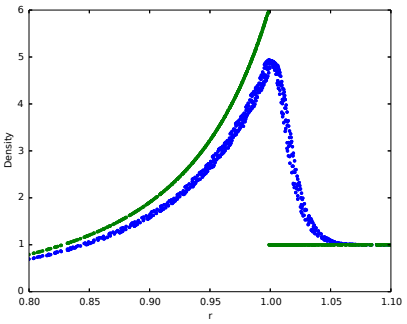
5 Conclusion

In this paper, we have analyzed the local numerical remapping error for the two often used flux-based remapping algorithms. We focused strictly on the situation of same-connectivity meshes and small mesh displacements, for which the swept-based remap is available. A general analytical formula representing the second-order error magnitude was derived. We have investigated several typical mesh motions and the error formula was adapted for such situations. It was shown that in certain special cases, such as corner movement, "hourglass" movement, or cell rotation, the remapping error produced by the swept-based method can have more nonuniform spatial distribution when compared to the error of the intersection-based method. The error of each method often depends, among other terms, on the orientation of the principal curvatures of the remapped distribution relative to the mesh movement.

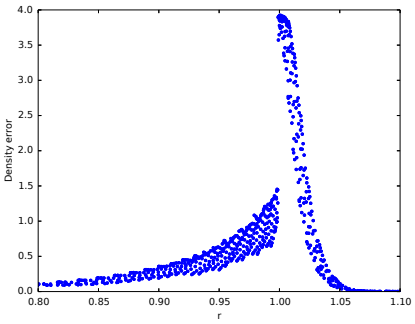
If the initial quantity distribution is symmetric, then although the total remapping error is similar, its variance and spatial distribution is apparently less favorable in the case



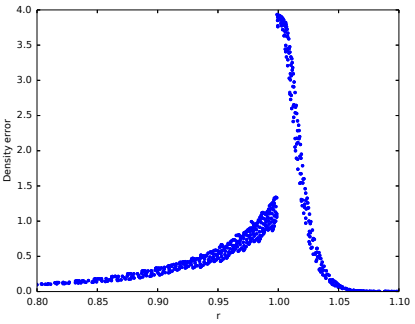
(a) Density, swept-based remapping



(b) Density, intersection-based remapping

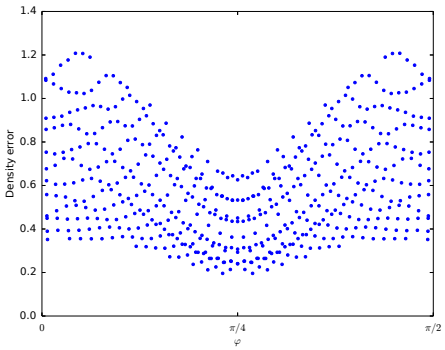


(c) Error, swept-based remapping

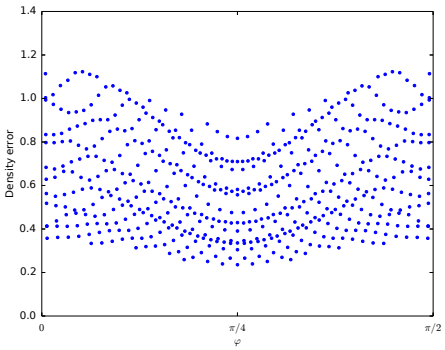


(d) Error, intersection-based remapping

Figure 16: Sedov problem, radial density (numerical and analytic) and density error $\rho_{\text{err}} = |\rho_{i,j} - \rho(\mathbf{r}_{i,j})|$, 45×45 mesh, $t = 1.0$.



(a) Swept-based remapping



(b) Intersection-based remapping

Figure 17: Angular distribution of density error $\rho_{\text{err}} = |\rho_{i,j} - \rho(\mathbf{r}_{i,j})|$ for Sedov test, 45×45 mesh, $t = 1.0$, both remapping approaches are compared. Only values in cells of radius between 0.9 and 0.99 (maximum difference between the methods) are shown.

of the swept-based method. This was validated on several numerical examples where the static remapping procedure was performed on a selected test function and the resulting error was analyzed. Finally, we have demonstrated on the well-known Sedov blast wave problem that in a hydrodynamical application, the difference between the two remapping methods is visible in the form of a small, but still visible error variance in an otherwise symmetric problem.

Future work involves the development of an approach for approximate and fast estimation of the remapping methods error difference. We are developing a pseudo-hybrid remapping method, using this estimate as a switch between the swept- and intersection-based algorithms in different regions of the computational domain in order to minimize the numerical error and the symmetry distortions.

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 No. DE-AC52-06NA25396 and supported by the DOE Advanced Simulation and Computing (ASC) program. The authors acknowledge the partial support of the DOE Office of Science ASCR Program. This work was partially supported by the Czech Technical University grant SGS16/247/OHK4/3T/14, the Czech Science Foundation project 14-21318S, and by the Czech Ministry of Education project RVO 68407700.

References

[1] R. W. Anderson, N. S. Elliott, and R. B. Pember. An arbitrary Lagrangian-Eulerian method with adaptive mesh refinement for the solution of the Euler equations. *Journal of Computational Physics*, 199(2):598–617, 2004.

[2] T. J. Barth. Numerical methods for gasdynamic systems on unstructured meshes. In D. Kroner, M. Ohlberger, and C. Rohde, editors, *An introduction to Recent Developments in Theory and Numerics for Conservation Laws, Proceedings of the International School on Theory and Numerics for Conservation Laws*, Berlin, 1997. Lecture Notes in Computational Science and Engineering, Springer. ISBN 3-540-65081-4.

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

[4] M. Berndt, J. Breil, S. Galera, M. Kucharik, P.-H. Maire, and M. Shashkov. Two step hybrid remapping (conservative interpolation) for multimaterial arbitrary Lagrangian-Eulerian methods. *Journal of Computational Physics*, 230(17):6664–6687, 2011.

[5] D. E. Burton, M. A. Kenamond, N. R. Morgan, T. C. Carney, and M. J. Shashkov. An intersection based ALE scheme (xALE) for cell centered hydrodynamics (CCH). Talk at Multimat 2013, International Conference on Numerical Methods for Multi-Material Fluid Flows, San Francisco, September 2-6, 2013. LA-UR-13-26756.2.

[6] 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.

[7] E. J. Caramana and M. J. Shashkov. Elimination of artificial grid distortion and hourglass-type motions by means of Lagrangian subzonal masses and pressures. *Journal of Computational Physics*, 142(2):521–561, 1998.

[8] E.J. Caramana, M.J. Shashkov, and P.P. Whalen. Formulations of artificial viscosity for multi-dimensional shock wave computations. *Journal of Computational Physics*, 144(1):70–97, 1998.

[9] J. K. Dukowicz and J. R. Baumgardner. Incremental remapping as a transport/advection algorithm. *Journal of Computational Physics*, 160(1):318–335, 2000.

[10] S. Galera, P.-H. Maire, and J. Breil. A two-dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction. *Journal of Computational Physics*, 229(16):5755–5787, 2010.

[11] 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.

[12] Ph. Hoch. An arbitrary Lagrangian-Eulerian strategy to solve compressible fluid flows. Technical report, CEA, 2009. HAL: hal-00366858. Available at `http://hal.archives-ouvertes.fr/docs/00/36/68/58/PDF/ale2d.pdf`.

[13] J.R. Kamm. Evaluation of the Sedov-von Neumann-Taylor blast wave solution. Technical Report LA-UR-00-6055, Los Alamos National Laboratory, 2000.

[14] M. A. Kenamond and D. E. Burton. Exact intersection remapping of multi-material domain-decomposed polygonal meshes. Talk at Multimat 2013, International Conference on Numerical Methods for Multi-Material Fluid Flows, San Francisco, September 2-6, 2013. LA-UR-13-26794.

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

[16] Patrick M. Knupp. Winslow smoothing on two-dimensional unstructured meshes. In *Proceedings of the Seventh International Meshing Roundtable, Park City, UT*, pages 449–457, 1998.

[17] M. Kucharik. *Arbitrary Lagrangian-Eulerian (ALE) Methods in Plasma Physics*. PhD thesis, Czech Technical University in Prague, 2006.

[18] M. Kucharik, J. Breil, S. Galera, P.-H. Maire, M. Berndt, and M. Shashkov. Hybrid remap for multi-material ALE. *Computers & Fluids*, 46(1):293–297, 2011.

[19] M. Kucharik and M. Shashkov. Flux-based approach for conservative remap of multi-material quantities in 2D arbitrary Lagrangian-Eulerian simulations. In Jaroslav Fořt, Jiří Fürst, Jan Halama, Raphaèl Herbin, and Florence Hubert, editors, *Finite Volumes for Complex Applications VI Problems & Perspectives*, volume 1 of *Springer Proceedings in Mathematics*, pages 623–631. Springer, 2011.

[20] M. Kucharik and M. Shashkov. One-step hybrid remapping algorithm for multi-material arbitrary Lagrangian-Eulerian methods. *Journal of Computational Physics*, 231(7):2851–2864, 2012.

[21] M. Kucharik and M. Shashkov. Conservative multi-material remap for staggered multi-material Arbitrary Lagrangian–Eulerian methods. *Journal of Computational Physics*, 258:268–304, 2014.

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

[23] P.H. Lauritzen, Ch. Erath, and R. Mittal. On simplifying 'incremental remap'-based transport schemes. *Journal of Computational Physics*, 230(22):7957–7963, 2011.

[24] R. Loubere, P.-H. Maire, M. Shashkov, J. Breil, and S. Galera. ReALE: A reconnection-based arbitrary-LagrangianEulerian method. *Journal of Computational Physics*, 229(12):4724–4761, 2010.

[25] L. G. Margolin. Introduction to "An arbitrary Lagrangian-Eulerian computing method for all flow speeds". *Journal of Computational Physics*, 135(2):198–202, 1997.
[26] L. G. Margolin and M. Shashkov. Second-order sign-preserving remapping on general grids. Technical Report LA-UR-02-525, Los Alamos National Laboratory, 2002.
[27] 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.
[28] D. J. Mavriplis. Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes. In *AIAA 2003-3986*, 2003. 16th AIAA Computational Fluid Dynamics Conference, June 23-26, Orlando, Florida.
[29] 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.
[30] 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.
[31] G. Scovazzi, E. Love, and M. Shashkov. Multi-scale Lagrangian shock hydrodynamics on Q1/P0 finite elements: Theoretical framework and two-dimensional computations. *Computer Methods in Applied Mechanics and Engineering*, 197(9-12):1056–1079, 2008.
[32] L.I. Sedov. *Similarity and Dimensional Methods in Mechanics, Tenth Edition*. Taylor & Francis, 1993.

Received: 14 January 2016 | Revised: 3 March 2017 | Accepted: 27 March 2017

DOI: 10.1002/fld.4384

RESEARCH PAPER PRESENTED AT MULTIMAT 2015

WILEY

Combined swept region and intersection-based single-material remapping method

Matej Klima[1] | Milan Kucharik[1] | Mikhail Shashkov[2]

[1]Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Brehova 7, Praha 1, 115 19, Czech Republic

[2]XCP-4 Group, MS-F644, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Correspondence
Matej Klima, Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Brehova 7, Praha 1, 115 19, Czech Republic.
Email: klimamat@fjfi.cvut.cz

Funding information
National Nuclear Security Administration of the US Department of Energy at Los Alamos National Laboratory, Grant/Award Number: DE-AC52-06NA25396; DOE Advanced Simulation and Computing (ASC) program; DOE Office of Science ASCR Program; Czech Technical University, Grant/Award Number: SGS16/247/OHK4/3T/14; Czech Science Foundation project, Grant/Award Number: 14-21318S; Czech Ministry of Education project RVO, Grant/Award Number: 68407700

Summary

A typical arbitrary Lagrangian–Eulerian algorithm consists of a Lagrangian step, where the computational mesh moves with the fluid flow; a rezoning step, where the computational mesh is smoothed and repaired in case it gets too distorted; and a remapping step, where all fluid quantities are conservatively interpolated on this new mesh. In single-material simulations, the remapping process can be represented in a flux form, with fluxes approximated by integrating a reconstructed function over intersections of neighboring computational cells on the original and rezoned computational mesh. This algorithm is complex and computationally demanding – Therefore, a simpler approach that utilizes regions swept by the cell edges during rezoning is often used in practice. However, it has been observed that such simplification can lead to distortion of the solution symmetry, especially when the mesh movement is not bound by such symmetry.

For this reason, we propose an algorithm combining both approaches in a similar way that was proposed for multi-material remapping (two-step hybrid remap). Intersections and exact integration are employed only in certain parts of the computational mesh, marked by a switching function – Various different criteria are presented in this paper. The swept-based method is used elsewhere in areas that are not marked. This way, our algorithm can retain the beneficial symmetry-preserving capabilities of intersection-based remapping while keeping the overall computational cost moderate.

KEYWORDS

ALE – arbitrary Lagrangian–Eulerian, compressible flow, error estimation, polygon intersections, remapping, swept regions

1 | INTRODUCTION

Traditionally, two classes of methods are used for hydrodynamic simulations – the Eulerian and Lagrangian approaches. In the pioneering work,[1] a general framework combining both approaches has been developed and termed arbitrary Lagrangian–Eulerian (ALE). Since that, this method has become very popular in the community and many authors have contributed to the topic.

Typically, the ALE algorithm is separated in three distinct steps: (1) a Lagrangian step, in which the set of fluid quantities and the computational mesh are advanced to the next time level; (2) a rezoning step, in which the computational mesh is changed to improve its geometric quality; and (3) a remapping step, in which the complete set of fluid quantities is conservatively transferred

from the Lagrangian to smoothed mesh. In this paper, we focus on the last part of the ALE algorithm, the remapping step. For simplicity, we only focus on remap of a single, cell-centered, single-material quantity (e.g., density).

For remapping, several different approaches can be used; for a review, see the seminal paper.[2] The most intuitive approach is remapping based on intersections.[3] In this approach, the cells of the new mesh are intersected with the original cells and the contributions to the new cell mass are computed as integrals of the particular density reconstruction over the intersections. This approach is straightforward but suffers from high computational cost resulting from the intersection construction and requires robust intersection method able to deal with close-to-parallel edges. On the other hand, the approach based on swept regions[4] is fast and robust by design and does not require any intersection algorithm. However, because of the approximate flux construction, it can have an adverse effect on symmetric distributions of the remapped quantity (observed in[5,6]).

The exact ratio of the computational cost of both methods is implementation dependent. In,[7] the swept-based method was shown to be approximately two times faster than intersections on a staggered mesh. Similar results with even higher ratios were published also in.[8,9] Our implementation can attain a speedup factor of more than three in an optimized remapping test (see results – Section 4). Additionally, there are many approaches for improving the results of both remapping methods, such as a priori limiters,[10] flux-corrected remap,[11] a posteriori repair,[12] or a posteriori order detection using the MOOD method.[13]

In practice, both remapping approaches behave in a very similar way and are second-order accurate for smooth solutions. For the differences to be identified between both methods, a detailed error analysis must be performed. In,[14] the authors analyze both approaches by the Fourier decomposition of the numerical error and show that the swept-based method can, under certain circumstances, provide better results than intersections. In,[4] the authors confirm second order of accuracy of both methods for smooth functions by representing the solution as a Taylor polynomial. In,[15] we have performed a detailed error analysis for several elementary mesh patterns, showing which error terms are different and responsible for a locally higher numerical error and the resulting symmetry violation.

In this paper, we propose a new remapping method – the adaptive hybrid remap, which combines both remapping approaches in different parts of the mesh. A similar idea was presented in,[13] where high-order and low-order reconstructions are combined instead. We follow the logic of the multi-material two-step hybrid remapping introduced in,[7] where the intersection-based remap was employed at the material interfaces and swept remap was used in single-material regions. Our method considers the single-material case only, and instead of detecting the presence of more materials in cell, the information about the remapped quantity and the change of the mesh during rezoning is used to decide which method to use in each cell (we will use the term 'switch' further in this paper). We present here several switches based on the error analysis of the remapping approaches that try to balance the preserving of the solution symmetry and the minimizing of the local error. In this paper, we focus on remapping on logically rectangular grids for the sake of simplicity, although the algorithm is applicable also on general unstructured meshes.

The rest of the paper is organized as follows. In Section 2, we briefly overview the existing remapping approaches formulated in a flux form. In Section 3, the method is described in detail and several switches for choosing the appropriate remapping method are introduced. Two simple switches from Sections 3.1 and 3.2 are based only on the first and second derivative of the remapped function, are simple to compute, and serve only as detector of high-density change or high curvature. The additional two switches are more complex and are based on the error analysis from.[15] The switch from Section 3.3 estimates the curvature of the function in the direction of the mesh motion, trying to use the intersection-based approach in regions where the lack of corner fluxes in the swept remap generates larger error. It also approximates the corner region volume to account for the mesh movement magnitude. The switch from Section 3.4 focuses more on the symmetry of the solution, using instead the discrete second derivative in the direction of every cell diagonal. The corner region volume is considered as well. Finally, in Section 4, the properties of both remapping approaches are compared with the new adaptive hybrid remap on selected static-remapping and fully hydrodynamic examples. The whole paper is concluded in Section 5.

2 | OVERVIEW OF EXISTING REMAPPING APPROACHES

Remapping is a process of transferring a discrete quantity between different computational meshes. In an ALE algorithm, the original mesh is usually determined by the fluid movement in the Lagrangian step. If required, this mesh is smoothed (rezoned) and the result is used as the target mesh. We presume that the rezoning algorithm does not change the mesh connectivity and the rezoned mesh cells are convex.

We require that the remapping process does not violate the conservation laws for the considered quantities, which is guaranteed when using the flux formulation of the remapping process.[16–18] In this paper, we will further focus on the remapping of a single cell-centered quantity: the fluid density, ρ. The flux form for remapped density is then expressed as

$$\rho_{\tilde{c}} V_{\tilde{c}} = \rho_c V_c + \sum_{c' \in S(c)} F^m_{c,c'}, \quad F^m_{c,c'} = -F^m_{c',c}, \tag{1}$$

where c is the cell of the original mesh, $\tilde{c}$ represents the rezoned cell, V_c, $V_{\tilde{c}}$ stand for the original and rezoned cell volume, and $F^m_{c,c'}$ represents the mass flux to cell c from its neighbor c'. $S(c)$ is a set of all cells surrounding c (sharing at least one vertex with c). The new cell volume $V_{\tilde{c}}$ is calculated from the rezoned cell geometry. The construction of the mass fluxes depends on the particular remapping method used. Methods based on cell intersections and swept regions are briefly described further in this section.

2.1 | Remapping by exact intersections

The most intuitive approach for remapping is based on intersecting the new and old computational meshes and approximating the density in each intersecting region. This way, the total mass of the rezoned cell can be written as follows:

$$m_{\tilde{c}} = \sum_{c' \in S(c) \cup \{c\}} m_{\tilde{c} \cap c'}. \tag{2}$$

The method described here involves only overlays with the neighboring cells.[18] The rezoned cell is not directly intersected with the old one ($m_{\tilde{c} \cap c}$); instead, the equivalent flux-oriented approach described in (1) is used:

$$m_{\tilde{c}} = m_c + \sum_{c' \in S(c)} m_{\tilde{c} \cap c'} - \sum_{\tilde{c}' \in S(\tilde{c})} m_{\tilde{c}' \cap c}. \tag{3}$$

Each flux is equal to the mass of the intersected region, as shown in Figure 1. They represent an exchange between edge and corner neighbors. The flux value can be approximated with a piecewise linear reconstruction of the density function:

$$m_{\tilde{c} \cap c'} = \iint_{c' \cap \tilde{c}} \left[\rho_{c'} + \left(\frac{\partial \rho}{\partial x} \right)_{c'} (x - x_{c'}) + \left(\frac{\partial \rho}{\partial y} \right)_{c'} (y - y_{c'}) \right] dx\, dy, \tag{4}$$

where $\rho_{c'}$ is the density mean value in the computational cell and $x_{c'}, y_{c'}$ are the coordinates of the geometrical cell centroid. The derivatives are estimated numerically in each cell (for a brief overview, see Section 2.3).

The method is straightforward; however, the implementation is often difficult. A robust polygon intersection algorithm is required, as the cell movement is usually small compared with the cell edge length, and the intersecting polygon edges can be often close to parallel. In that case, we switch from the analytical formula to iterative calculation, depending on the angle formed by intersecting edges. The implementation of this algorithm is described in detail in,[18] Appendix B.

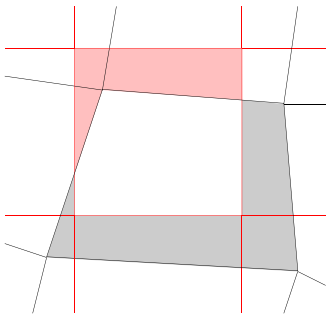


FIGURE 1 Flux construction using intersections of old and new cells. — new mesh, — old mesh, positive flux, negative flux [Colour figure can be viewed at wileyonlinelibrary.com]

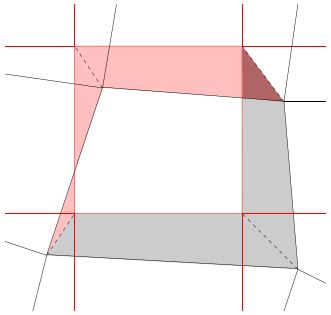


FIGURE 2 Flux construction using regions swept by the cell edges. new mesh, old mesh, positive flux, negative flux, overlapping regions [Colour figure can be viewed at wileyonlinelibrary.com]

2.2 | Remapping by swept regions

An alternative to the exact intersection method approximates the mass fluxes by integrating the reconstructed density over a polygon constructed from the old and new cell edge positions[4,19,20]:

$$m_{\bar{c}} = m_c + \sum_{e \in E(c)} \iint_{\Delta e} \left[\rho_{c'} + \left(\frac{\partial \rho}{\partial x} \right)_{c'} (x - x_{c'}) + \left(\frac{\partial \rho}{\partial y} \right)_{c'} (y - y_{c'}) \right] dx\, dy. \tag{5}$$

Here, the region swept by each edge is labeled as Δe. $E(c)$ is a set of all edges of cell c. c' is either c itself or its neighbor over the edge e, depending on the sign of the oriented volume integral over Δe. As we can see, this method ignores corner fluxes altogether and distributes them in edge fluxes instead (Figure 2). This approach is not equivalent to (3), as the edge fluxes use density reconstruction only in the edge neighbors. It corresponds to (1) with all corner terms set to 0.

2.3 | Numerical derivative approximation

It is important to describe the method used for approximating the first derivative numerically. It is used in the piecewise linear reconstruction of the density function in ((4), (5)) and in the switching functions as well. Later in the paper, we will also need to estimate the second derivatives – These are calculated with the same algorithm used on the numerical gradient values instead of density.

The method used here is based on the least squares minimization of the error functional in the neighboring cells.[18,21] It tries to minimize the difference between the mean density values and the values extrapolated to the neighboring cells using the first derivatives we are interested in.

If used for calculating the first derivative, this method is supplemented by the Barth–Jespersen limiting process[10] to avoid function overshoots and preserve the local extrema. For the second derivative, no such limiter is used as it is a more complex task that would affect the computational cost as well.

3 | REMAPPING BY BOTH METHODS COMBINED

Clearly, the swept region method has many advantages. However, under some circumstances, the intersection method may be more favorable, especially when considering solution symmetry on a mesh that is not aligned with such symmetry. In this paper, we try to balance the computational cost and solution quality by using the more expensive exact intersections only in some parts of the computational domain.

When using the swept-based method, there are no intersections to determine the direction of the flux, the cell from which the reconstruction is taken is selected according to the sign of the oriented integral over the region. This leads to one drawback of this method – When a cell edge rotates so that the resulting swept region is a self-intersecting polygon, the resulting volume integral of such polygon is close to 0 and it is unclear which reconstruction to choose. This type of movement can arise, for example, after the hourglass mesh pattern forms in the Lagrangian step (although most codes usually try to avoid it[22,23]) or in case of strong rotational flows. The solution to this problem during remap has been proposed in[24] – to split the self-intersecting polygon into two separate triangles by calculating only the intersection of the old and new edges. It could be

combined with the method proposed further as well. For this reason, we will focus on the other movement mode for which the swept method is not well suited – movement of the computational cell in the direction of its corner, where dedicated corner fluxes are missing.[15]

The algorithm design is inspired by the principles of multi-material two-step hybrid remapping.[7] Such algorithm marks cells to be remapped by one of the methods in the beginning of the remapping step. Then it calculates fluxes by either one of the methods in the respective cells in two separate steps. Our remapping algorithm has the following structure:

1. Mark all cells if they are to be remapped by swept or by intersections (depending on the particular switching function).
2. Construct an intermediate mesh, where all nodes of the cells marked for intersection remapping are in the original position (Figure 3A). All the other vertices are in their rezoned position.
3. Calculate and save the swept region integrals using the old and intermediate mesh.
4. Calculate the remapped values using swept region fluxes.
5. Construct the fully rezoned mesh (Figure 3B); for example, take the intermediate mesh and move the marked cells.
6. Calculate and save integrals of intersections of cells using the intermediate and rezoned mesh.
7. Calculate the remapped values using intersection fluxes.

The meshes of the two steps of the method are depicted in Figure 3. Note that in the second step, intersection remapping will be performed not only on the marked cells but also on the layer of so-called buffer cells. This happens because some of their edges move in both steps of the algorithm.

As we assume in this paper that all cells contain only one material, our motivation is different from the multi-material method, where the swept-based method as described here is not applicable on mixed cells. Our aim is rather to have a method that is faster than calculating intersections everywhere but has similar beneficial properties related to the solution symmetry.

An important part of such method is the switches that determine which remapping method may be used in which part of the domain. This is performed by comparing a certain function value with a predetermined threshold in each individual computational cell. This threshold is a method parameter that is constant during the whole simulation for all mesh cells. The switches are designed without direct dependence on cell size, so they produce similar triggering patterns for all mesh resolutions. Although the threshold values used further were determined empirically, this way they can be determined at low mesh resolutions and

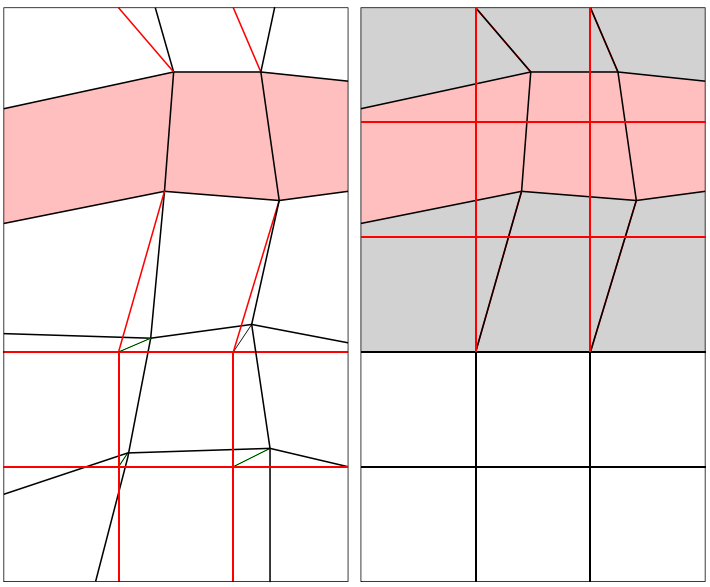


FIGURE 3 Mesh construction in two steps of our adaptive hybrid remapping method. A, Swept remapping step and B, intersection remapping step mesh. ■ cells marked for intersections, ■ buffer cells [Colour figure can be viewed at wileyonlinelibrary.com]

used afterwards in high-resolution computations. The switching functions can take the mesh geometry and movement as well as the discrete quantity and its numerically approximated derivatives as parameters. We further consider four examples of such switching functions in this paper.

The first two functions are selected to evaluate the identified regions with steep slope or high curvature of the density distribution – There we expect the error to differ the most, depending on the mesh orientation. The other two functions are based on our local error analysis of both remapping methods[15] and use the directional second derivatives (DDSs) – one with the task to minimize the error in general and the other measuring curvature in the direction of the cell diagonals.

3.1 | Gradient switch

The first switch is the simplest one – It tries to identify discontinuous areas of the function where its gradient is high:

$$\frac{\sqrt{\left(\frac{\partial \rho}{\partial x}\right)_c^2 + \left(\frac{\partial \rho}{\partial y}\right)_c^2}}{\rho_{\max}} > \alpha_{GS}, \tag{6}$$

where $\rho_{\max}$ is the maximum of discrete density values in the investigated cell and its neighbors. Depending on the nature of the problem, a global maximum can be used here as well. The advantage of this switch is its simplicity and almost no computational overhead, because the first derivatives are needed by both remapping methods for the density reconstruction. The problem is that it can trigger intersections incorrectly in areas with steep, but constant gradient, which requires setting the threshold α_{GS} adequately for each different problem.

If the switch was used on a different quantity that can be negative, maximum of absolute values would be used in the denominator. If it was 0, then it would mean that the quantity is locally constant and swept-based remap could be used directly.

3.2 | Hessian switch

Another switch takes into account not the slope, but rather the local curvature of the remapped function, described by its Hessian matrix (calculated numerically, see Section 2.3):

$$\frac{\sqrt{|\det\ \mathbf{H}(\rho)|}}{\rho_{\max}} > \alpha_{HS}. \tag{7}$$

As we are using a second-order function reconstruction, the error of the remapping method will be proportional to the second derivative, and thus, this switch should be proportional to the possible remapping error. The main disadvantage is that it requires calculation of the second derivative, which adds to the total computational cost (but in general, the increase is lower than that when using intersection remapping). Also, this switch does not take into account the effect of the mesh change during rezoning.

3.3 | Directional second derivative ratio

The local remapping error is dependent not only on the mesh movement magnitude during rezoning but also on the movement direction. This switch investigates each vertex of the cell and from its movement estimates the direction vector. This is then used to calculate DDS values:

$$\rho_{\mathbf{dd}} = \frac{\mathbf{d} \cdot \nabla\,(\mathbf{d} \cdot \nabla \rho)}{\|\mathbf{d}\|^2} = \frac{\mathbf{d}\ \mathbf{H}(\rho)\ \mathbf{d}^{\mathbf{T}}}{\mathbf{d}^2}, \rho_{\mathbf{pp}} = \frac{\mathbf{p}\ \mathbf{H}(\rho)\ \mathbf{p}^{\mathbf{T}}}{\mathbf{p}^2}, \tag{8}$$

where $\mathbf{d}$ is the direction of the movement and $\mathbf{p}$ the perpendicular vector. If the direction of the movement is perpendicular to the direction of the maximal curvature, the swept method is expected to perform worse than intersections, as we have shown in.[15] And also the remapping error is proportional to the corner mesh displacement. These criteria can be formulated as the following switching function:

$$\frac{4\ V_d}{V_c}\left|\frac{\rho_{\mathbf{pp}}}{\rho_{\mathbf{dd}}}\right| > \alpha_{DDS}, \tag{9}$$

where V_d is an estimate of the corner volume flux; we use a rectangle formed by the old and new corner vertex positions. Because of this, the switch value is defined in each corner node and if it triggers in any of the corners, it triggers in the whole cell. V_c is the known cell volume. The complexity of this switch is only slightly higher than that of the HS. Several singular cases can occur here; if $\rho_{\mathbf{pp}} = 0$, we always use swept-based remap, even when $\rho_{\mathbf{dd}} = 0$ as well (constant gradient, no curvature). If $\rho_{\mathbf{pp}} \neq 0$ and $\rho_{\mathbf{dd}} = 0$, intersections are used (curvature only in the perpendicular direction).

3.4 | Diagonal second derivative switch

The previous switch is designed to select the method that produces lower remapping error in each cell. This does not necessarily mean that it will help preserve symmetry. Therefore, we have constructed this switch with symmetry in mind. It calculates the second derivative not in the movement direction but in the direction of cell diagonal, in which the corner flux is maximal:

$$\rho_{\mathbf{gg}} = \frac{\mathbf{g}\ \mathbf{H}(\rho)\ \mathbf{g}^{\mathbf{T}}}{\mathbf{g}^2}, \mathbf{g} = \mathbf{x_i} - \mathbf{x_c}, \tag{10}$$

where $\mathbf{g}$ is the vertex position vector in coordinates centered in the cell centroid. This switch is applied on all cell vertices as well. The second derivative is compared with the maximum density value in the neighboring cells, and the relative corner volume estimate is added. The final formula for this switching function is as follows:

$$\frac{4\ V_d}{V_c}\frac{|\rho_{\mathbf{gg}}|}{\rho_{\max}} > \alpha_{DGS}. \tag{11}$$

Again, this switch triggers if the inequality is true for any of the cell corner nodes.

4 | NUMERICAL RESULTS

The behavior of our remapping method is demonstrated on three numerical examples. The first one involves remapping of one quantity only with no fluid movement. The mesh movement is prescribed, and the initial distribution is a discontinuous one. In the second section, the results for the Sedov problem with mesh smoothing are presented. The last test is the compressible Taylor–Green vortex calculated with rezoning on initial grid.

4.1 | Cyclic remapping of an initially discontinuous distribution

In the first test, we compare the described switching functions on a static test that involves only remapping. The cells are initialized with masses corresponding to a discontinuous radially symmetric density function. This test is taken from[7], and the initial density distribution is shown in Figure 4A. Then we run a given number of successive remapping steps with a predefined rezoning motion. This movement is chosen so that it distorts the mesh and then returns it to its initial state (cyclic remapping), where we can compare the initial cell masses with the remapped ones to obtain the remapping error. This test is run on initially square meshes with increasing resolutions. The initial density function is defined here as follows:

$$f(x,y) = \begin{cases} 1 + e^{10\sqrt{(x-1/2)^2+(y-1/2)^2}} & \text{for } \sqrt{(x-1/2)^2+(y-1/2)^2} \leqslant 1/4, \\ 1 + e^{6\sqrt{(x-1/2)^2+(y-1/2)^2}-1/4} & \text{in other cases.} \end{cases} \tag{12}$$

The mesh movement is chosen so that it does not follow the symmetry of the initial distribution. During this so-called tensor product rezoning, the vertical and horizontal grid lines move at different speeds, but stay perpendicular[4]. The inner mesh vertices thus move in diagonal directions. This movement is defined as follows:

$$\begin{aligned} x_i^n &= x_i^0(1-d^n) + \left(x_i^0\right)^3 d^n,\ y_i^n = y_i^0(1-d^n) + \left(y_i^0\right)^2 d^n, \\ d^n &= \frac{\sin(2\pi t^n)}{2},\ t^n = n/n_{\max}, \end{aligned} \tag{13}$$

where $(x_i^n,\ y_i^n)$ is the actual position of node i and $(x_i^0,\ y_i^0)$ is the initial position. In general, the superscript 0 denotes the initial grid, n the actual grid, and $n_{\max}$ the total number of the remapping steps (which is set to twice the mesh resolution in the horizontal/vertical direction). The boundary vertices slide along the borderline in the same manner as the inner vertices. The mesh with maximal distortion is shown in Figure 4B. We have tried also other cyclic tests presented in,[4] such as the sine-like rezoning. Nevertheless, the selected movement seemed to be the most useful one to demonstrate corner flux-induced remapping error and produced the most visible difference.

Figure 4C and D shows the final state after the cyclic remapping test, with a visible difference in the resulting density distribution for swept-based and intersection-based remapping. The adaptive hybrid methods are visually in-between and were omitted for the sake of brevity. The remapping error and asymmetry of all methods are quantified further in this section.

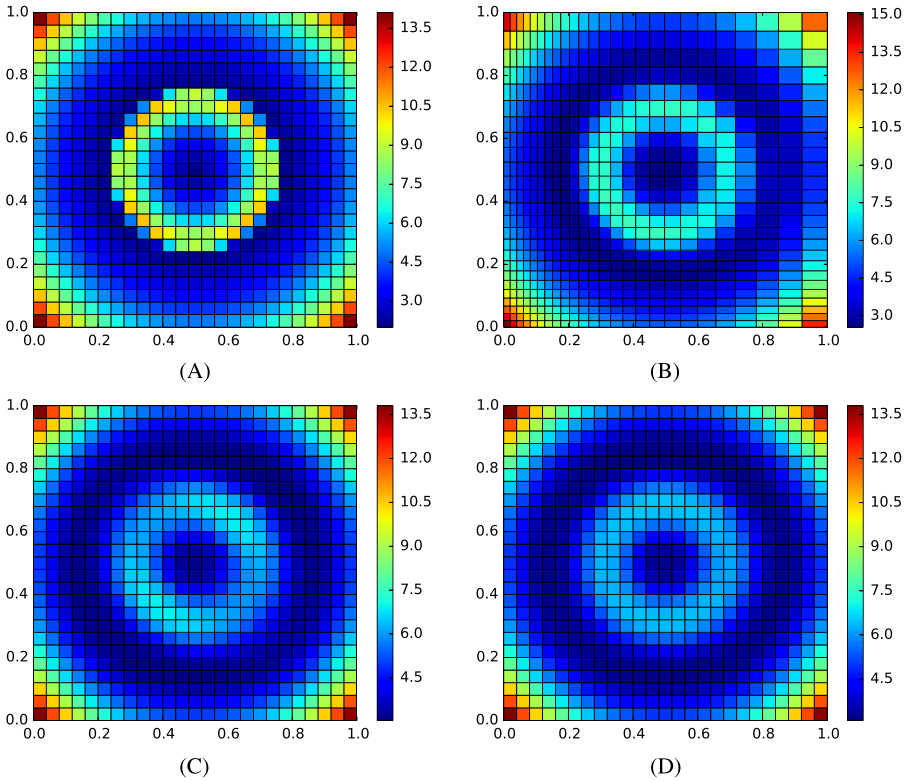


FIGURE 4 Density profile during the cyclic remapping test, 25×25 mesh, discontinuous exponential initial distribution, tensor product rezone, $n_{\max} = 50$. A, Initial state; B, $n = 12$, swept region remapping; C, final state, swept region remapping; D, final state, exact intersection remapping [Colour figure can be viewed at wileyonlinelibrary.com]

To measure the total error produced by the remapping process, we use the L1 norm:

$$L_{1m} = \frac{\sum_c m_c^n - m_c^0}{\sum_c m_c^0}, \tag{14}$$

where m_c^n is the actual (remapped) cell mass, while m_c^0 is the initial cell mass. This simple measurement is possible because the final and initial meshes are identical.

It can be seen that the swept region remapping produces less satisfying results although the total L1 error is similar for all methods as shown in Table 1. The L1 error is shown only for the high-resolution mesh, as the error behaves similarly for lower resolutions. The adaptive hybrid remap with the directional derivative switch is slightly more accurate than both swept-based (∼3.3%) and intersection-based (∼5%) methods. This is due to the switch being able to precisely choose areas where the selected method performs better.

The initial quantity distribution is symmetric, but the mesh movement is not. To obtain some measure of the resulting asymmetry, we calculate the L1 mass errors (14) separately for each quadrant of the computational domain. Then we calculate the relative standard deviation of the L1 error to express its variability over the four quadrants:

$$\sigma_{L_{1m}} = \frac{\sqrt{\frac{1}{4}\sum_{q=1}^{4}(L_{1m}^q - L_{1m})^2}}{L_{1m}}, \tag{15}$$

where L_{1m}^q represents the total mass error for a particular quadrant and L_{1m} for the whole domain.

TABLE 1 Cyclic remapping – total L1 mass error, discontinuous exponential initial distribution, tensor product rezone, 400×400 mesh, step $n = 800$

Method	L_{1m}
INT	$1.782 \cdot 10^{-2}$
SW	$1.749 \cdot 10^{-2}$
GS	$1.783 \cdot 10^{-2}$
HS	$1.782 \cdot 10^{-2}$
DDS	$1.692 \cdot 10^{-2}$
DGS	$1.782 \cdot 10^{-2}$

INT, intersection; SW, swept; GS, gradient switch; HS, Hessian switch; DDS, directional second derivative; DGS, diagonal second derivative.

TABLE 2 Cyclic remapping – variance of L1 mass errors in domain quadrants $\sigma_{L_{1m}}$ [%] for different mesh resolutions, discontinuous exponential initial distribution, tensor product rezone

Method	25^2	50^2	100^2	200^2	400^2
INT	5.46	2.65	2.94	2.91	2.76
SW	10.22	9.44	10.67	10.86	10.92
GS	9.05	4.02	3.85	3.13	2.81
HS	10.00	2.75	2.96	2.91	2.76
DDS	7.30	5.35	6.31	6.58	6.73
DGS	8.19	3.22	3.11	2.93	2.76

INT, intersection; SW, swept; GS, gradient switch; HS, Hessian switch; DDS, directional second derivative; DGS, diagonal second derivative.

TABLE 3 Cyclic remapping – simulation running times [s] for various mesh resolutions, discontinuous exponential initial distribution, tensor product rezone

Method	25^2	50^2	100^2	200^2	400^2
INT	0.71	5.70	45.30	366	2944
SW	0.23	1.60	12.62	97.63	799
GS	0.28	2.69	20.86	165.3	1337
HS	0.34	3.68	27.89	216.5	1647
DDS	0.46	3.08	22.51	166.6	1286
DGS	0.39	3.37	26.23	203.4	1562

INT, intersection; SW, swept; GS, gradient switch; HS, Hessian switch; DDS, directional second derivative; DGS, diagonal second derivative.

Comparison of error variation (relative standard deviation [%]) and the corresponding computational expense for all methods are shown in Tables 2 and 3 (the code is run single-thread using a 2.66-GHz Intel Xeon X5355 processor). The intersection-based remap is more than three times slower than swept remapping, while our adaptive hybridf method lies in-between. The adaptive hybrid methods using gradient switch (GS), Hessian switch (HS), and diagonal second derivative (DGS) switch show behavior much closer to the exact intersection remapping and are significantly faster. The difference between the error variance for swept regions and intersections is approximately constant (four times higher spread for swept regions), and the asymmetry is present also in high-resolution simulations. The DDS switch does not preserve symmetry as well as the simpler ones; this is the downside of minimizing the error at every cost. For this reason, we have not included a combination of symmetry-improving/error-reducing switches, as those demands are mutually incompatible. The relationship between error deviation and resolution for all methods is shown in Figure 5.

The comparison of the triggering pattern of the switching function is shown in Figure 6 at various remapping steps. The first derivative switch targets the discontinuity well, but the asymmetry is also present in other parts of the domain.

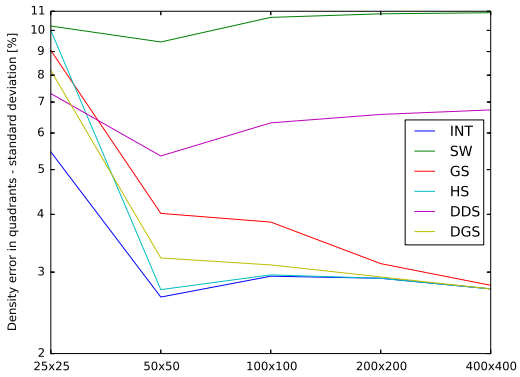


FIGURE 5 Cyclic remapping – variance of L1 mass errors in domain quadrant comparison for various mesh resolutions, discontinuous exponential initial distribution, and tensor product rezone. INT, intersections; SW, swept regions; GS, gradient switch; HS, Hessian switch; DDS, directional second derivative; DGS, diagonal second derivative [Colour figure can be viewed at wileyonlinelibrary.com]

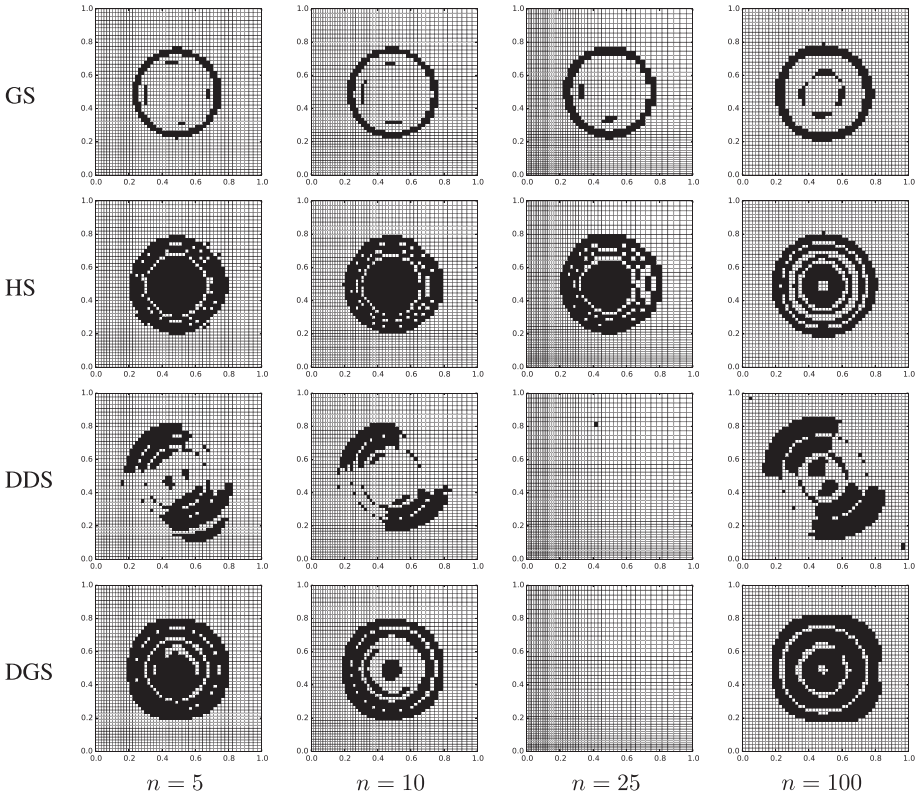


FIGURE 6 Switching functions pattern during cyclic remapping, 50×50 mesh, discontinuous exponential initial distribution, tensor product rezone, and adaptive hybrid remapping. Switch pattern: black, intersections; white, swept regions. GS, gradient switch; HS, Hessian switch; DDS, directional second derivative; DGS, diagonal second derivative

The second derivative switch triggers in larger area at the end, and its symmetry-preserving capabilities can be seen in the error profile. The directional derivative switch has the least radial symmetry in the switch pattern. We can also see that the DDS and DGS switches do not trigger at $n = 25$. This is because there is virtually no mesh rezoning going on (turning point

of the cyclic remapping problem) and there is no need to use any intersections – The switches use approximate corner flux volume and correctly handle this situation. The thresholds were set accordingly: $\alpha_{GS} = 8.0, \alpha_{HS} = 50.0, \alpha_{DDS} = 1.0$, and $\alpha_{DGS} = 50.0$.

Until now, we have measured error distribution in domain quadrants. We can also determine the radial error distribution. This is shown in Figure 7. The cells are grouped according to their distance from the center. In the plot, we can see the standard deviation of the error for each radius level. Because of the square mesh used here, there are always at least four cells in each group. The radial error deviation shows similar behavior as the quadrant distribution – The asymmetry is most severe when using swept region remapping (the spread is approximately twice as big). The best results are achieved when using intersection remapping. The DGS switches perform in a very similar way.

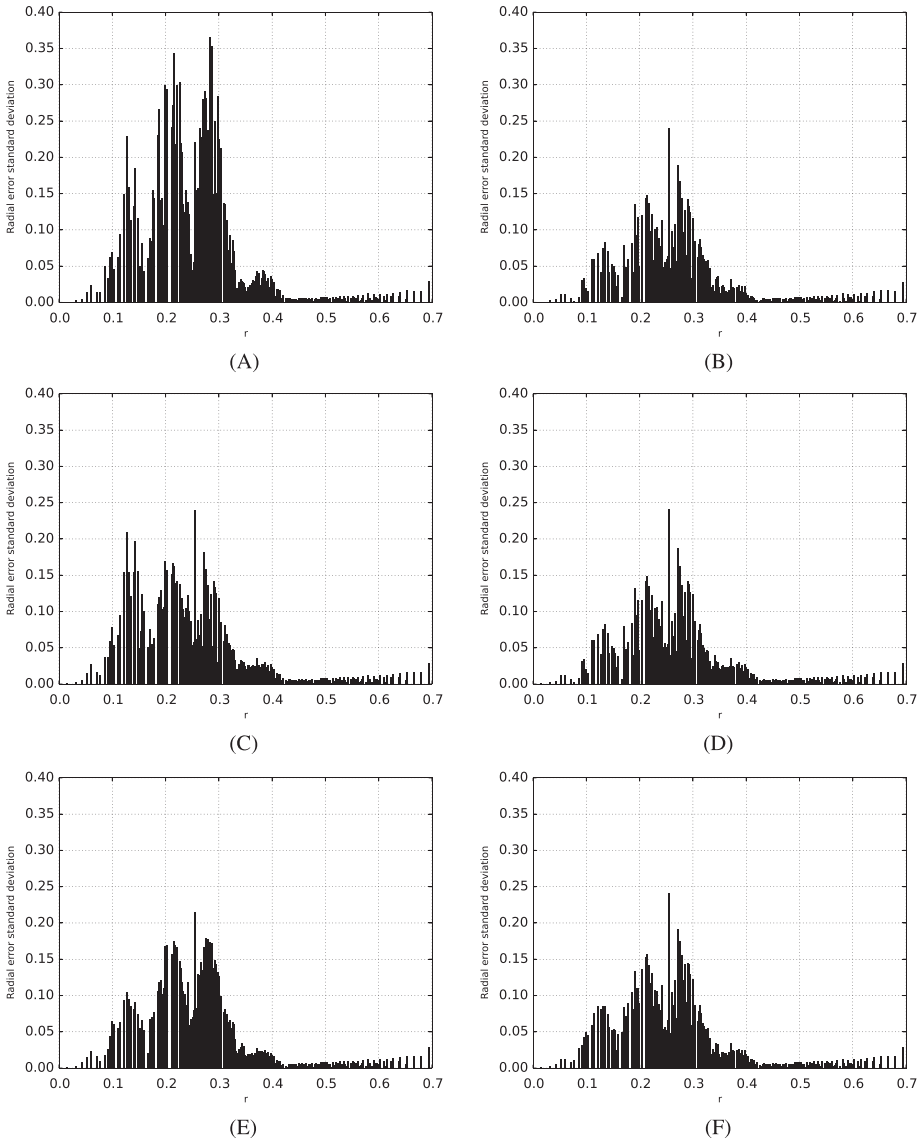


FIGURE 7 Cyclic remapping – L1 mass error radial variance comparison for different remapping methods, 50×50 mesh, discontinuous exponential initial distribution, and tensor product rezone, $n = 100$. A, Swept regions; B, intersections; C, gradient switch; D, Hessian switch; E, directional second derivative; F, diagonal second derivative

4.2 | ALE fluid dynamics – the Sedov problem

This test contains full ALE fluid dynamics simulation,[18] where the 2D compressible Euler equations are solved by the standard Lagrangian solver[25] with predictor–corrector time integration. The remapping stage uses the Winslow mesh smoothing algorithm[26] every 10 Lagrangian steps. The initial conditions of the Sedov test[22] represent an energy release from the origin (here in cell $(0, 0)$), while other quantities are initially constant:

$$\rho = 1.0,\ p = 0.4 \cdot 10^{-14},\ \mathbf{u} = \mathbf{0},\ \gamma = 1.4, \epsilon_{0,0} = 409.7,\ t = 1.0. \tag{16}$$

This creates a shockwave that radially expands through the computational domain. This problem has an analytical solution,[27] which is shown in Figure 8. This solution is shown and sampled on the same mesh as the numerical result. The differences between methods are small and better visible on a radial scatter plot (Figure 10).

For demonstrating our remapping method, the thresholds were set accordingly: $\alpha_{GS} = 12.0, \alpha_{HS} = 100.0, \alpha_{DDS} = 0.25$, and $\alpha_{DGS} = 0.01$. The switch is triggered based on density values only.

The switch pattern time evolution is shown in Figure 9. For the GS, the pattern is radially symmetric and copies the shock front. It is to be noted that the rezoning movement itself is symmetric as well (counteracting the outward movement originating from the Lagrangian stage). The DDS switch shows only relatively few cells triggered for intersections without radial symmetry.

In Figure 10, we compare the resulting radial error spread of swept, intersection, and combined remapping. Again, we can see that although the average error is roughly the same, the swept-based method error spread is larger than that of intersection

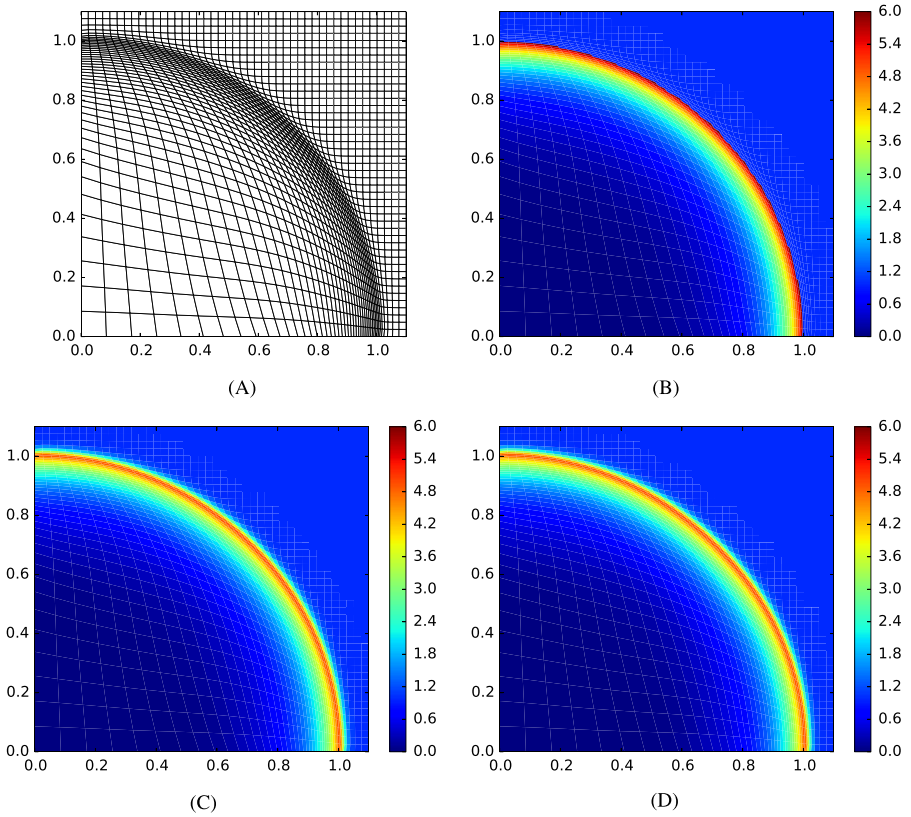


FIGURE 8 Sedov problem – density profile, ALE simulation with Winslow rezoning every 10 time steps, 45×45 cells, $t = 1.0$. A, Computational mesh; B, sampled analytical solution; C, numerical solution (intersection fluxes); D, numerical solution (swept region fluxes) [Colour figure can be viewed at wileyonlinelibrary.com]

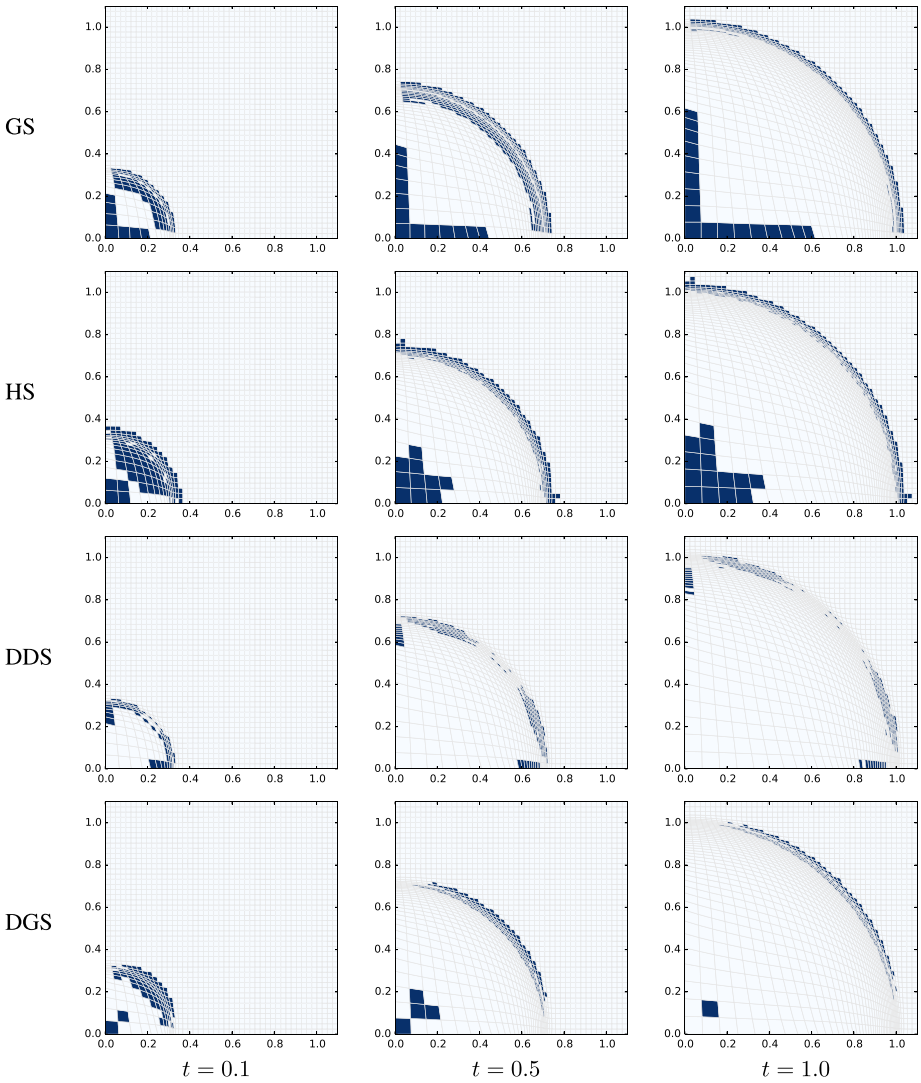


FIGURE 9 Sedov problem – switch pattern time evolution, ALE simulation with Winslow rezoning every 10 time steps, adaptive hybrid remapping, 45×45 cells, $t = 1.0$. Switch pattern: blue, intersections; white, swept regions. GS, gradient switch; HS, Hessian switch; DDS, directional second derivative; DGS, diagonal second derivative [Colour figure can be viewed at wileyonlinelibrary.com]

or combined remapping. Especially, the DDS switch triggers to minimize the local error – In this test, the mesh movement is symmetric and the adaptive hybrid method reduces the spread even compared with the intersection method. Similar behavior can be seen for a refined 90×90 mesh in Figure 11, although at this resolution, the differences between the adaptive hybrid and intersection methods are diminished.

We can demonstrate the influence of the threshold selection on the DGS switch in Figure 12, where the time dependence of the total number of cells remapped by intersections is shown. In this case, halving the threshold parameter corresponds to approximately doubling of the number of triggered cells. The effect on the error distribution is shown in Figure 13. We can see that increasing the number of triggered cells does not bring any significant improvement in the symmetry preservation – The adaptive hybrid method is already effective when used on less than 50 cells. For other switches, the threshold is scaled differently, but the effect on the number of triggered cells and the resulting radial error is similar.

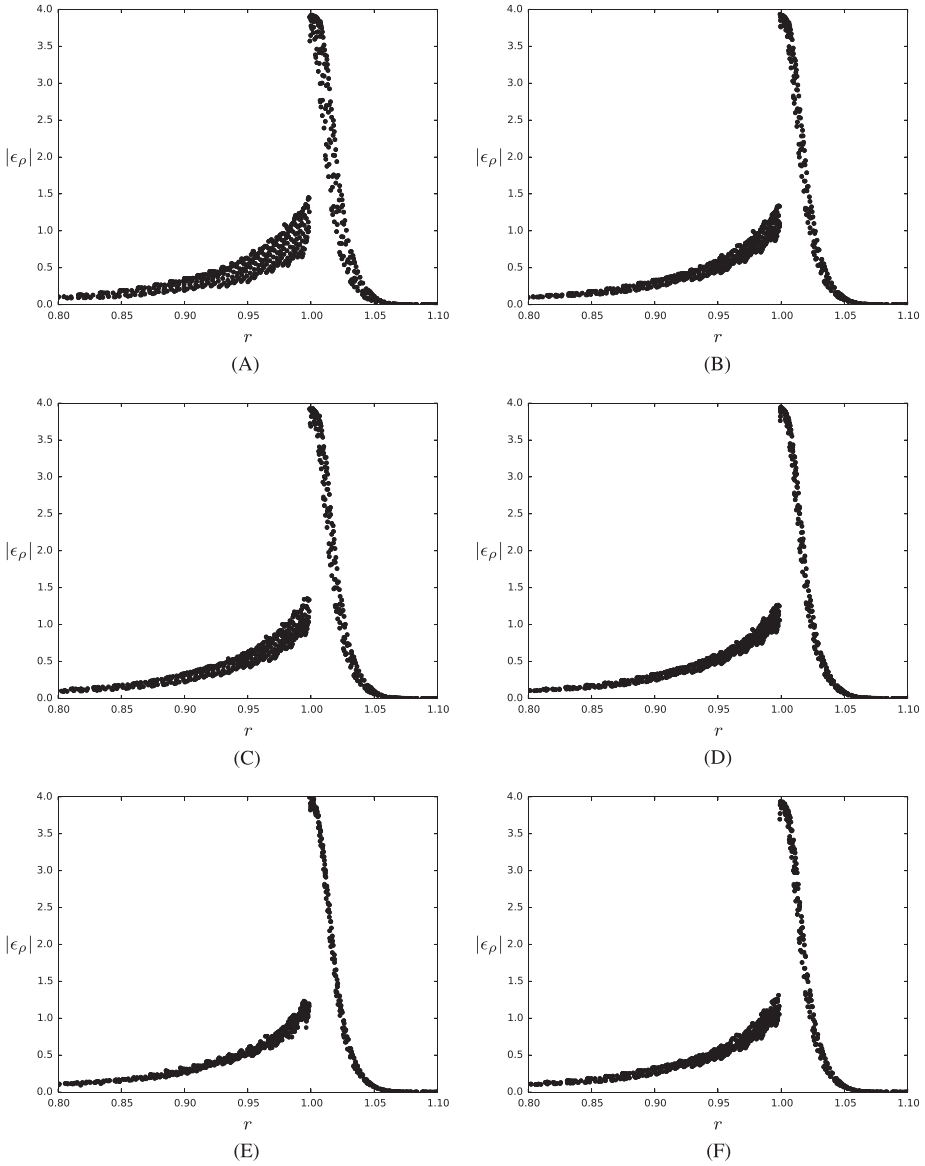


FIGURE 10 Sedov problem – radial density error distribution near the shock front compared by the remapping method, ALE simulation with Winslow rezoning every 10 time steps, 45×45 cells, $t = 1.0$. A, Swept regions; B, intersections; C, gradient switch; D, Hessian switch; E, directional second derivative; F, diagonal second derivative

The breakdown of system time spent in each part of the algorithm is shown in Table 4 for all methods. The resulting computational cost is very implementation dependent, and also, the Lagrangian step consumes most of the resources because rezone/remap is not used in every time step. The most significant difference can be seen when compfaring the total time spent calculating swept/intersection/combined fluxes. The remapper cannot fully benefit from the advantages of a faster flux calculation method as the cost of other parts of the remapping algorithm is approximately constant for all methods. We can also see that the contribution to the computational cost of the second derivative evaluation (HS and DDS and DGS switches) is negligible. Generally, the cost of switch evaluation is much lower than the cost of flux calculation, so using switches for more variables is feasible without significant slowdown. The measurement was carried out using a finer computational mesh

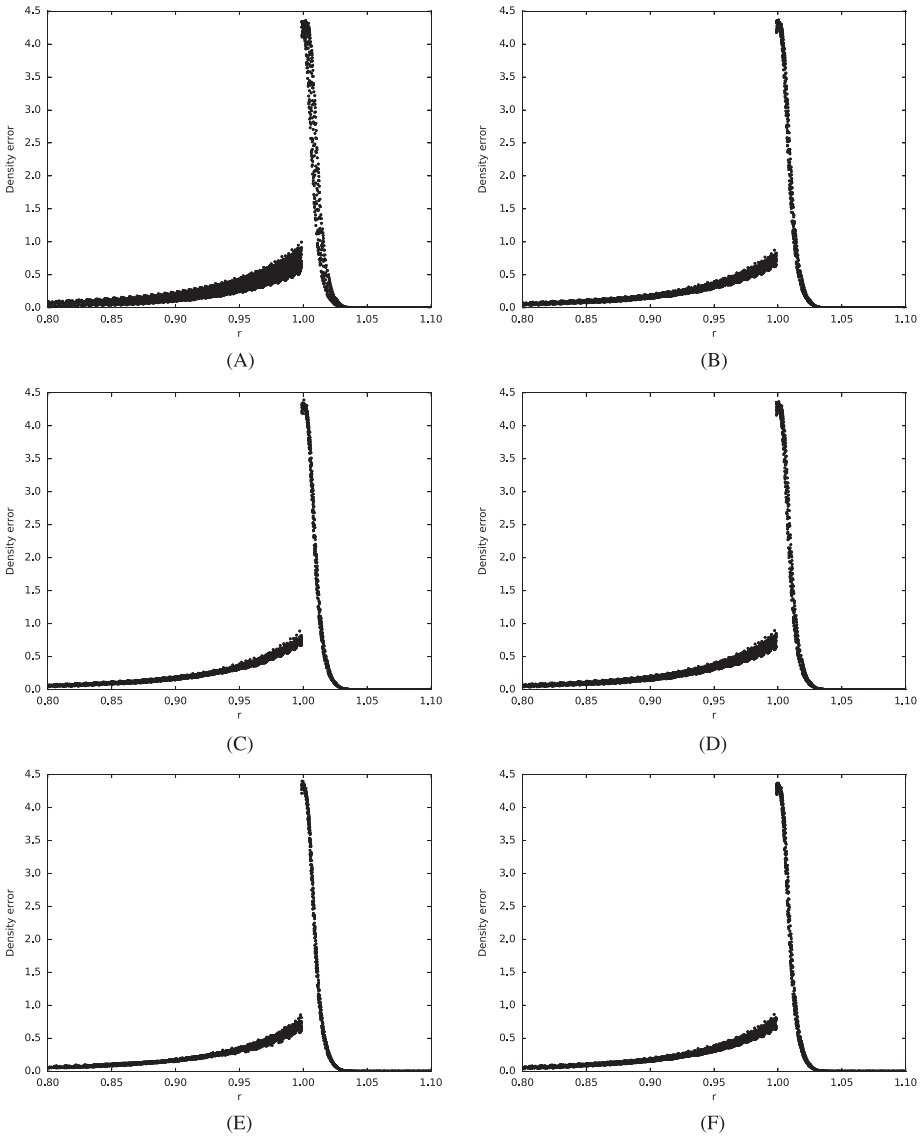


FIGURE 11 Sedov problem – radial density error distribution near the shock front compared by the remapping method, ALE simulation with Winslow rezoning every 10 time steps, 90×90 cells, $t = 1.0$. A, Swept regions; B, intersections; C, gradient switch; D, Hessian switch; E, directional second derivative; F, diagonal second derivative

(90×90 cells) than that in the original test[22] to highlight the differences between the remapping methods. Even so, a slight variation in the time measured in the Lagrangian step is visible, probably because of measurement error and operating system overhead.

4.3 | ALE fluid dynamics – the compressible Taylor–Green vortex

The last numerical test presented in this paper is the 2D Taylor–Green vortex[28] with compressible ideal gas – a standard test that involves rotational flow inside the domain:

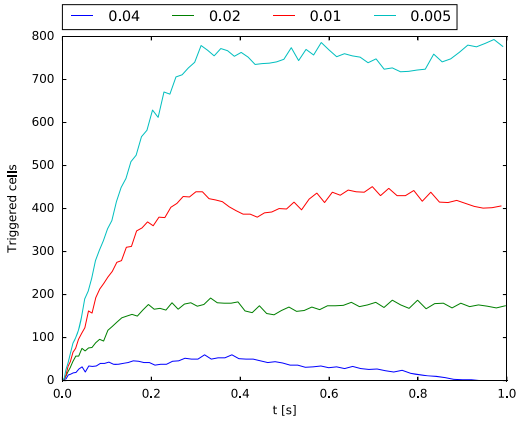


FIGURE 12 Sedov problem – time history of the total number of cells triggered for intersection remapping. Computed using the adaptive hybrid method with the diagonal second derivative switch. Comparison of different threshold settings. ALE simulation with Winslow rezoning every 10 time steps, 45×45 cells [Colour figure can be viewed at wileyonlinelibrary.com]

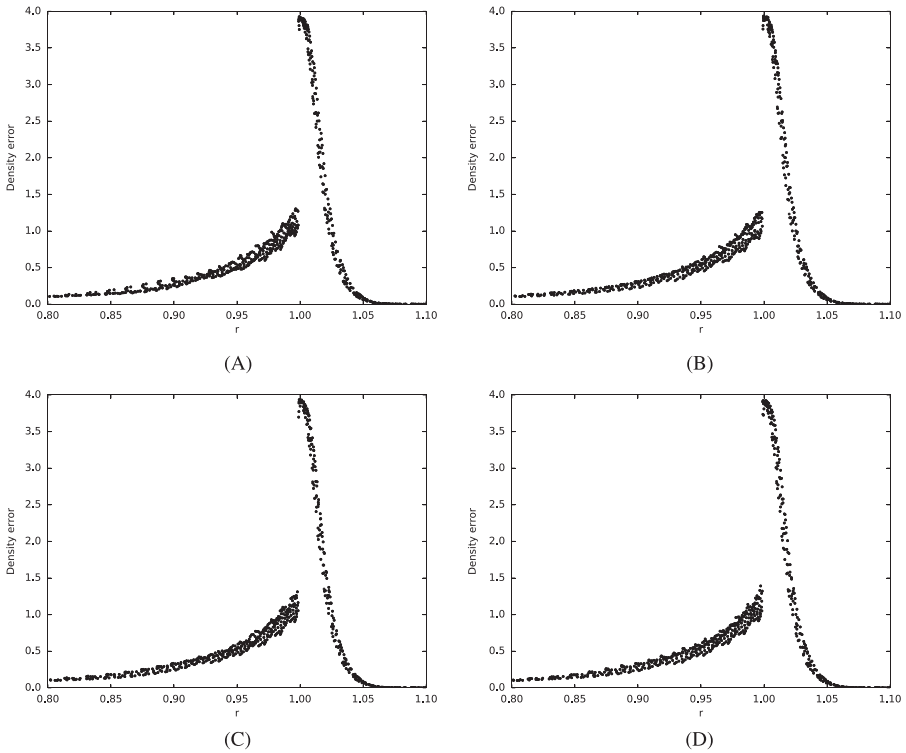


FIGURE 13 Sedov problem – radial density error distribution near the shock front using the adaptive hybrid remapping method with the DGS switch. Comparison of different threshold settings. ALE simulation with Winslow rezoning every 10 time steps, 45×45 cells, $t = 1.0$.
A, $a_{DGS} = 0.04$; B, $a_{DGS} = 0.02$; C, $a_{DGS} = 0.01$; D, $a_{DGS} = 0.0005$

$$\mathbf{u}_x = \sin(\pi x)\cos(\pi y), \quad \mathbf{u}_y = -\cos(\pi x)\sin(\pi y),$$
$$\rho = 1, \quad p = p_0 + \frac{1}{4}(\cos(2\pi x) + \cos(2\pi y)), \quad p_0 = 1, \quad \gamma = 1.4.$$

TABLE 4 Sedov problem – computational time [s] of all remapping methods compared, ALE simulation with Winslow rezoning every 10 time steps, 90×90 mesh, $t = 1.0$

Method	INT	SW	GS	HS	DDS	DGS
Total	146.3	138.7	133.7	136.2	139.1	134.7
Lag. step	95.5	100.1	93.2	94.4	97.6	94.6
Remap	43.4	30.7	33.1	34.1	33.5	32.5
Fluxes	16.9	2.5	4.3	4.2	3.1	3.1
Switch	0	0	0.24	0.65	0.76	0.75

INT, intersection; SW, swept; GS, gradient switch; HS, Hessian switch; DDS, directional second derivative; DGS, diagonal second derivative. Lag., Lagrangian.

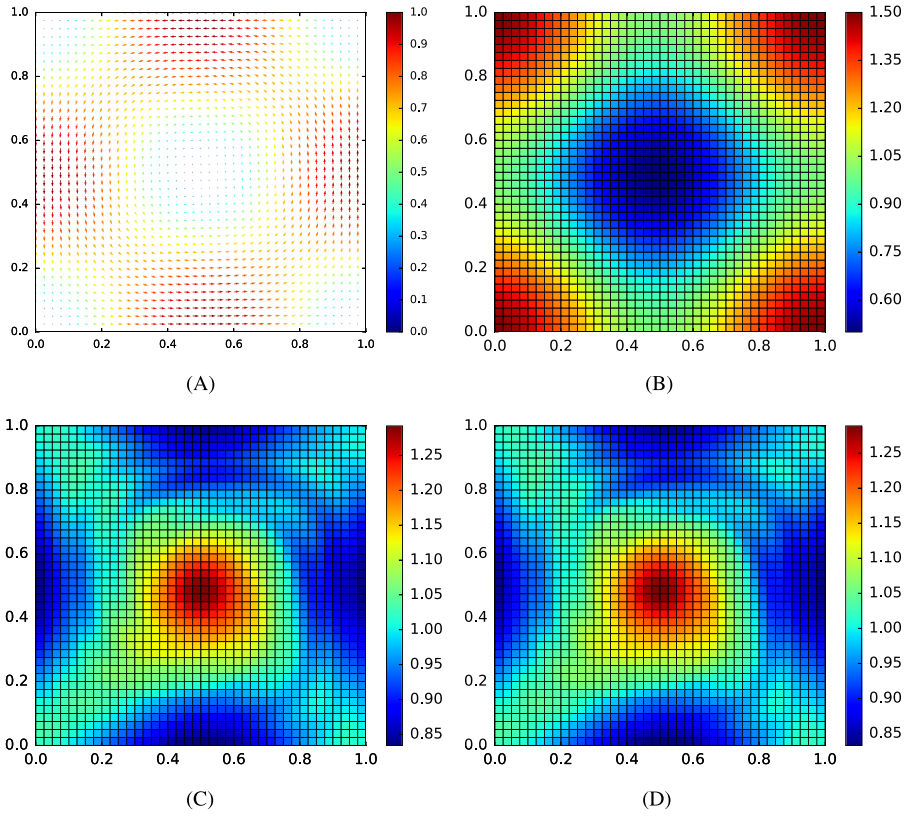


FIGURE 14 Taylor–Green vortex. ALE simulation with initial mesh rezoning on every time step, 40×40 cells. A, Velocity field at $t = 0$; B, pressure at $t = 0$; C, density at $t = 4$, intersection remapping; D, density at $t = 4$, swept-based remapping [Colour figure can be viewed at wileyonlinelibrary.com]

The internal energy is specified using the ideal gas equation of state. In this test, the mesh is smoothed by rezoning to its initial state on every time step. This enables us to run the test with long simulation times and compare initial distributions with the results at any time step. The initial conditions for velocity and pressure are shown in Figure 14A and B.

We can see that the actual difference between density profiles for intersection-based and swept-based remapping is very low in this case (Figure 14C and 14D). The plots are similar for adaptive hybrid remapping as well and were omitted for the sake of brevity. Instead, a comparison of the L1 norm (14) of the density deviation from initial state is shown in Table 5 alongside with its variation in each quadrant (15) to compare the symmetry as well. It is to be noted that these results do not represent the numerical error as the exact density distribution is unknown for the compressible variant of this problem. Nevertheless, the average density in the computational domain is $\bar{\rho} = 1$ at any time because of the conservation of mass.

TABLE 5 Comparison of the L1 norm of $\rho - \rho_{\text{init.}}$, the variation of this L1 norm in domain quadrants, and the corresponding expense of flux computation for various remapping methods

Method	L_{1m}	$\sigma_{L_{1m}}$ [%]	$t_{\text{flux}}[s]$
INT	$6.8367 \cdot 10^{-2}$	11.04	12.26
SW	$6.8323 \cdot 10^{-2}$	10.81	2.95
GS	$6.7443 \cdot 10^{-2}$	11.48	4.02
HS	$6.8365 \cdot 10^{-2}$	10.51	4.52
DDS	$6.8189 \cdot 10^{-2}$	10.84	4.53
DGS	$6.8234 \cdot 10^{-2}$	10.68	4.18

The Taylor–Green compressible vortex test, ALE simulation with rezoning to initial mesh on each step, 40×40 mesh, $t = 4$.INT, intersection; SW, swept; GS, gradient switch; HS, Hessian switch; DDS, directional second derivative; DGS, diagonal second derivative.

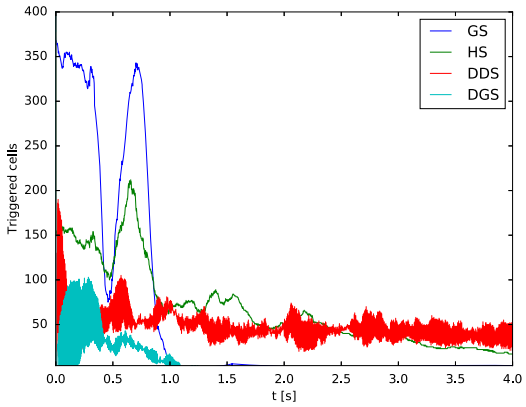


FIGURE 15 Time history of the total number of cells triggered for intersection remapping. Comparison of different switches in the adaptive hybrid remapping method. The Taylor–Green compressible vortex test, ALE simulation with rezoning to initial mesh on each step, 40×40 mesh. GS, gradient switch; HS, Hessian switch; DDS, directional second derivative; DGS, diagonal second derivative [Colour figure can be viewed at wileyonlinelibrary.com]

For this problem, the thresholds were set as $\alpha_{GS} = 2.0, \alpha_{HS} = 10.0, \alpha_{DDS} = 0.005$, and $\alpha_{DGS} = 0.00001$. The results in Table 5 show comparable accuracy and radial symmetry for all remapping methods. The adaptive hybrid remapping methods are all much less computationally demanding than the intersection method, although the swept-based method offers even faster calculation. It is to be noted that (especially for DDS and DGS methods) the thresholds need to be set to much lower values than in case of diverging flow such as the Sedov test in previous section. These switches therefore correctly prefer the swept-based method in this test.

In Figure 15, the time evolution of the number of triggered cells for each adaptive hybrid remapping switch is shown. The DDS and DGS switches show oscillatory behavior, probably because of the very low thresholds and associated high sensitivity of the switch.

5 | CONCLUSION

In this work, we have presented a new combined remapping approach, generalizing hybrid methods to single-material adaptive hybrid remap. Its aim is to optimize the efficiency of existing swept-based and intersection-based remapping methods. Both methods are applied on selected cells in a two-step manner with the intersection fluxes used only where deemed as beneficial. The selection of cells is carried out on the basis of evaluating a switching criterion utilizing quantity values, their derivatives, and mesh geometry and movement data. The sensitivity of this selection is determined by the switching function threshold. Several such switches were proposed and evaluated in this paper.

We have shown that when using such method, the computational expense related to remapping is lower than that when using intersection remapping solely, although the benefit to the overall computational efficiency depends on the implementation, the frequency of remapping, and the number of remapped physical quantities for multiphysics problems. We have also shown that our algorithm retains most of the radial symmetry-preserving capabilities of the intersection remapping when implemented in an ALE code. The adaptive hybrid remapping method is therefore effective in cases where swept-based remapping performs worse – This has been shown for an asymmetric mesh movement over symmetric distribution and also for a radially symmetric diverging flow in the Sedov problem. In the Taylor–Green test case with predominantly rotational flow without discontinuities, the adaptive hybrid remapping does not offer any significant improvement.

Because of the relatively low cost of the switch evaluation, we suggest using more complex functions that can decrease the number of intersections calculated, such as the DGS switch, focusing on the symmetry of the solution. If solution symmetry is not the primary concern, the DDS switch can slightly decrease the overall numerical error compared with both swept-based and intersection-based remapping. Generally, the developed method is designed as a tool allowing to combine two standard remapping approaches and there are many possibilities for other switching conditions to be designed for specific applications.

Future work includes improving the implementation of this remapping method so that the computational expense is significantly reduced even in full ALE codes. A similar approach is interesting especially in 3D remapping, where the potential reduction in CPU time can be even more pronounced. It will be interesting to investigate other different switches for different quantities and applications. Also, designing switches with automatically adjusting thresholds will be useful in simulations where the result is not easily predictable or where the range of movement and quantity values may vary.

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 no. DE-AC52-06NA25396 and supported by the DOE Advanced Simulation and Computing (ASC) program. The authors acknowledge the partial support of the DOE Office of Science ASCR Program. This work was partially supported by the Czech Technical University grant SGS16/247/OHK4/3T/14, the Czech Science Foundation project14-21318S, and the Czech Ministry of Education project RVO68407700.

REFERENCES

1. Hirt CW, Amsden AA, Cook JL. An arbitrary Lagrangian–Eulerian computing method for all flow speeds. *J Comput Phys*. 1974;14(3):227–253.
2. Benson DJ. Computational methods in Lagrangian and Eulerian hydrocodes. *Comput Methods Appl Mech Eng*. 1992;99(2–3):235–394.
3. Dukowicz J. Conservative rezoning (remapping) for general quadrilateral meshes. *J Comput Phys*. 1984;54(3):411–424.
4. Margolin LG, Shashkov M. Second-order sign-preserving conservative interpolation (remapping) on general grids. *J Comput Phys*. 2003;184(1):266–298.
5. Kenamond MA, Burton DE. Exact intersection remapping of multi-material domain-decomposed polygonal meshes. *Talk at Multimat 2013, International Conference on Numerical Methods for Multi-Material Fluid Flows*, San Francisco; 2–6 September 2013. LA-UR-13-26794.
6. Burton DE, Kenamond MA, Morgan NR, Carney TC, Shashkov MJ. An intersection based ALE scheme (xALE) for cell centered hydrodynamics (CCH). *Talk at Multimat 2013, International Conference on Numerical Methods for Multi-Material Fluid Flows*, San Francisco; 2–6 September 2013. LA-UR-13-26756.2.
7. Berndt M, Breil J, Galera S, Kucharik M, Maire PH, Shashkov M. Two-step hybrid conservative remapping for multimaterial arbitrary Lagrangian–Eulerian methods. *J Comput Phys*. 2011;230(17):6664–6687.
8. Kucharik M, Shashkov M. One-step hybrid remapping algorithm for multi-material arbitrary Lagrangian–Eulerian methods. *J Comput Phys*. 2012;231(7):2851–2864.
9. Breil J, Alcin H, Maire PH. A swept intersection-based remapping method for axisymmetric ReALE computation. *Int J Numer Meth Fluids*. 2015;77(11):694–706. fld.3996.
10. Barth TJ. Numerical methods for gasdynamic systems on unstructured meshes. In: Kroner D, Rohde C, Ohlberger M, eds. *An Introduction to Recent Developments in Theory and Numerics for Conservation Laws, Proceedings of the International School on Theory and Numerics for Conservation Laws*, Lecture Notes in Computational Science and Engineering. Springer, Berlin; 1997. ISBN 3-540-65081-4.
11. Liska R, Shashkov M, Vachal P, Wendroff B. Optimization-based synchronized flux-corrected conservative interpolation (remapping) of mass and momentum for arbitrary Lagrangian–Eulerian methods. *J Comput Phys*. 2010;229(5):1467–1497.
12. Kucharik M, Shashkov M, Wendroff B. An efficient linearity-and-bound-preserving remapping method. *J Comput Phys*. 2003;188(2):462–471.
13. Blanchard G, Loubere R. High-order conservative remapping with a posteriori MOOD stabilization on polygonal meshes. 2015. https://hal.archives-ouvertes.fr/hal-01207156, the HAL Open Archive, hal-01207156, Accessed 13.1.2016.
14. Lauritzen P, Erath C, Mittal R. On simplifying 'incremental remap'-based transport schemes. *J Comput Phys*. 2011;230(22):7957–7963.
15. Klima M, Kucharik M, Shashkov M. Local error analysis and comparison of the swept- and intersection-based remapping methods. *Commun Comput Phys*. 2017;21(2):526–558.

16. Dukowicz JK, Baumgardner JR. Incremental remapping as a transport/advection algorithm. *J Comput Phys*. 2000;160(1):318–335.
17. Kucharik M, Shashkov M. Flux-based approach for conservative remap of multi-material quantities in 2D arbitrary Lagrangian–Eulerian simulations. In: Fořt J, Fürst J, Halama J, Herbin R, Hubert F, eds. *Finite Volumes for Complex Applications VI Problems & Perspectives*, Springer Proceedings in Mathematics, vol. 1: Springer, Berlin, Heidelberg; 2011:623–631.
18. Kucharik M, Shashkov M. Conservative multi-material remap for staggered multi-material arbitrary Lagrangian–Eulerian methods. *J Comput Phys*. 2014;258:268–304.
19. Loubere R, Shashkov M. A subcell remapping method on staggered polygonal grids for arbitrary-Lagrangian–Eulerian methods. *J Comput Phys*. 2005;209(1):105–138.
20. Margolin LG, Shashkov M. Second-order sign-preserving remapping on general grids. LA-UR-02-525, Los Alamos National Laboratory, Los Alamos, New Mexico; 2002.
21. Mavriplis DJ. Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes. *16th AIAA Computational Fluid Dynamics Conference*, AIAA 2003-3986, Orlando, Florida; 23–26 June 2003.
22. Scovazzi G, Love E, Shashkov M. Multi-scale Lagrangian shock hydrodynamics on Q1/P0 finite elements: theoretical framework and two-dimensional computations. *Comput Methods Appl Mech Eng*. 2008;197(9–12):1056–1079.
23. Caramana EJ, Shashkov MJ. Elimination of artificial grid distortion and hourglass-type motions by means of Lagrangian subzonal masses and pressures. *J Comput Phys*. 1998;142(2):521–561.
24. Hoch P. An arbitrary Lagrangian–Eulerian strategy to solve compressible fluid flows. CEA-DAM, DIF, Arpajon, Île-de-France; 2009. HAL: hal-00366858. https://hal.archives-ouvertes.fr/docs/00/36/68/58/PDF/ale2d.pdf. Accessed 13.1.2016.
25. Shashkov M. *Conservative Finite-Difference Methods on General Grids*. Boca Raton, Florida: CRC Press; 1996. ISBN 0-8493-7375-1.
26. Knupp PM. Winslow smoothing on two-dimensional unstructured meshes. *Eng Comput*. 1999;15:263–268.
27. Kamm J. Evaluation of the Sedov–von Neumann–Taylor blast wave solution. LA-UR-00-6055, Los Alamos National Laboratory, Los Alamos, New Mexico; 2000.
28. Taylor GI, Green AE. Mechanism of the production of small eddies from large ones. *Proc R Soc Lond A Math Phys Eng Sci*. 1937;158(895):499–521.

How to cite this article: Klima M, Kucharik M, Shashkov M. Combined swept region and intersection-based single-material remapping method. *Int J Numer Meth Fluids*. 2017;0:1–20. https://doi.org/10.1002/fld.4384