

**Czech Technical University in Prague
Faculty of Nuclear Sciences and Physical Engineering**



DISSERTATION

Advanced Tools for Arbitrary Lagrangian-Eulerian Simulations

Praha 2020

Ing. Matěj Klíma

Supervisor:

Doc. Ing. Milan Kuchařík, PhD.

Supervisor-specialist:

Prof. Ing. Richard Liska, CSc.

Field of study:

Physical Engineering

Bibliographic Entry

Author	Ing. Matěj Klíma Czech Technical University in Prague Faculty of Nuclear Sciences and Physical Engineering Department of Physical Electronics
Title of Dissertation	Advanced Tools for Arbitrary Lagrangian-Eulerian Simulations
Degree Programme	(D4CS) Application of Natural Sciences
Field of Study	(3901V012) Physical Engineering
Supervisor	Doc. Ing. Milan Kuchařík, PhD. Czech Technical University in Prague Faculty of Nuclear Sciences and Physical Engineering Department of Physical Electronics
Supervisor-specialist	Prof. Ing. Richard Liska, CSc. Czech Technical University in Prague Faculty of Nuclear Sciences and Physical Engineering Department of Physical Electronics
Academic Year	2019/2020
Number of Pages	223
Keywords	ALE, multi-material problems, adaptive remap, tensor remap, elastic-plastic flows, voids

Bibliografický záznam

Autor	Ing. Matěj Klíma České vysoké učení technické v Praze Fakulta jaderná a fyzikálně inženýrská Katedra fyzikální elektroniky
Název práce	Pokročilé metody pro Lagrangeovsko-Eulerovské simulace
Studijní program	(D4CS) Aplikace přírodních věd
Studijní obor	(3901V012) Fyzikální inženýrství
Školitel	Doc. Ing. Milan Kuchařík, PhD. České vysoké učení technické v Praze Fakulta jaderná a fyzikálně inženýrská Katedra fyzikální elektroniky
Školitel-specialista	Prof. Ing. Richard Liska, CSc. České vysoké učení technické v Praze Fakulta jaderná a fyzikálně inženýrská Katedra fyzikální elektroniky
Akademický rok	2019/2020
Počet stran	223
Klíčová slova	ALE, multi-materiálové problémy, adaptivní remap, remap tenzorů, elasto-plasticita, voids

Acknowledgements

From all the people involved in the research leading to this dissertation and the related publications, first and foremost I would like to thank my supervisor, Milan Kuchařík, for the valuable guidance and expertise I have received during my doctoral study. It was his patient and careful mentoring that made the completion of this work possible. I would like to thank also Mikhail Shashkov for my opportunity to work at LANL. His everyday mentorship during my stay there has really moved the research forward, while making sure all algorithms are working properly. My deepest gratitude and good health wishes go also to Andrew Barlow, who has provided me with assistance and expert knowledge on the topic of closure models.

Special thanks go out to the head of our research group, Richard Liska, and my colleagues Pavel Váchal and David Fridrich for expert advices, additional support and companionship on conference travels. Many thanks go also to Jan Velechovský and his family for all the time spent together in New Mexico, especially the outdoor trips maintaining my mental and physical health. And finally, special thanks go out to Burton Wendroff who made delivering the ideas presented in this dissertation possible by letting me use his bicycle to move around Los Alamos.

This work or its parts were supported by the Czech Technical University grants SGS13/220/OHK4/3T/14, SGS16/247/OHK4/3T/14, SGS19/191/OHK4/3T/14, the Czech Science Foundation projects 14-21318S, 18-20962S, the Czech Ministry of Education project RVO 68407700 and the European Regional Development Fund project CZ.02.1.01/0.0/0.0/16 019/0000778.

Declaration

I hereby declare that I have written this dissertation on my own and that I have cited all used resources. This dissertation is the result of my own work, except where explicit reference is made to the work of others or collective research results are presented – in such case the extent of the contribution of the author is provided. This dissertation has not been submitted for another qualification to this or any other university.

In Prague, August 23, 2020

Prohlašuji, že jsem tuto práci vytvořil samostatně a že jsem uvedl veškerou použitou literaturu. Tato práce prezentuje originální výsledky výzkumu autora, kromě případů, kde se výslovně odkazuje na práci jiných nebo zmiňuje výsledky kolektivní práce – v takovém případě je specifikován podíl autora na společném výsledku. Tato práce nebyla přihlášena k jiné obhajobě na této ani žádné jiné univerzitě.

V Praze, 23.srpna 2020



Abstract

There are various approaches to simulating fluid flows, among them the Arbitrary Lagrangian-Eulerian (ALE) method offers both robustness and accurate representation of high-deformation interactions. Often it is realised as a combination of a Lagrangian stage solving the fluid equations and a rezone/remap step which improves the quality of the computational mesh, if necessary, and conservatively interpolates the fluid quantities onto the improved mesh. We focus on the multi-material ALE approach that allows us to simulate interactions of several distinct materials with very different properties.

In this dissertation we present several extensions to the ALE method. In the Lagrangian stage an improved Interface-Aware Sub-Scale Dynamics multi-material model is derived to allow computations with both low and high speed interactions of fluids, elastic solids or voids (an environment representing vacuum). In the remapping stage we analyze the accuracy of the employed flux calculation methods and propose a mechanism of combining them to preserve symmetry without drastically increasing the computational cost. Also a novel stress tensor remapping algorithm is presented with a special focus on the conservation of total elastic energy in the domain.

Keywords: ALE, multi-material problems, adaptive remap, tensor remap, elastic-plastic flows, voids

Supervisor: Doc. Ing. Milan Kuchařík, PhD.

Abstrakt

Mezi různými přístupy k simulacím dynamiky tekutin vynikají metody ALE (Lagrangeovsko-Eulerovské) robustností a přesností reprezentace dějů s velkými deformacemi. Tyto metody jsou často realizovány pomocí kombinace Lagrangeovské fáze (řešící rovnice popisující chování kontinua) a rezone/remap fáze, kdy je výpočetní síť v případě špatné kvality vyhlazena a fluidní veličiny jsou poté konzervativně interpolovány na tuto síť. Zaměřujeme se na multi-materiálový ALE algoritmus, který umožňuje simulovat interakce více materiálů s velmi rozdílnými vlastnostmi.

V této práci představujeme několik vylepšení této metody. V Lagrangeovské fázi jsme se zaměřili na vylepšení multi-materiálového modelu IASSD (Interface-Aware Sub-Scale Dynamics), který byl upraven tak, aby umožňoval výpočty interakcí tekutin, pevných látek i tzv. voidů (oblastí vyplněných vakuem) v nízkých i velmi vysokých rychlostech. Ve fázi remapu jsme pak analyzovali přesnost metod pro výpočet toků veličin a navrhli mechanismus, který je umožňuje kombinovat tak, aby byla zachována symetrie bez většího nárůstu výpočetní náročnosti. Popsali jsme také nový algoritmus remapování tenzoru napětí splňující požadavek na zachování celkové elastické energie v simulaci.

Klíčová slova: ALE, multi-materiálové problémy, adaptivní remap, remap tenzorů, elasto-plasticita, voids

Překlad názvu: Pokročilé metody pro Lagrangeovsko-Eulerovské simulace

Contents

1	Introduction	1
1.1	New contributions of the Dissertation	4
2	Governing Equations	7
2.1	Constitutive model of solids	8
2.2	Equations of state	9
3	Compatible Lagrangian discrete model	11
3.1	Multi-material spatial discretization	14
3.1.1	Simple closure models	16
3.1.2	Material reconstruction	17
3.2	Predictor-corrector time integration, algorithm overview . . .	18
3.3	Time step considerations	22
3.4	Artificial viscosity	23
3.5	Anti-hourglass mesh stabilization	24
4	IASSD multi-material model for fluids, solids and voids	25
4.1	Bulk stage	26
4.2	Sub-scale dynamics stage	28
4.2.1	Constraints on material volume change	30
4.2.2	Material centroid advection	31
4.3	Void treatment	31
4.3.1	Closure	32
4.3.2	Opening	32
5	Mesh smoothing and remapping of the fluid quantities	35
5.1	Mesh rezoning	35
5.2	Flux-form remapping strategies	36
5.2.1	Piece-wise linear reconstruction	37
5.2.2	Fluxes by integrating over exact intersections	38
5.2.3	Flux regions swept by the cell edges	39
5.2.4	Hybrid remapping techniques	40
5.3	Remapping of all fluid quantities (algorithm overview)	42
5.3.1	Specific internal energy	42
5.3.2	Momentum remap on dual mesh	43
5.3.3	Kinetic energy fix	45
5.3.4	Elastic energy-preserving deviatoric stress remap	46
6	Examples of numerical calculations	49
6.1	Sedov problem – point explosion	49
6.2	Triple point problem	51
6.3	Shock-induced cavity collapse	52
6.4	Elastic vibrating plate	53
6.5	High speed aluminium plate impact	56
6.6	Low speed steel ball impact	58
7	Summary of selected publications	61
7.1	Constrained optimization for IASSD models for voids closure (Appendix A)	61
7.2	IASSD model for solids with void closure and opening at all speeds (Appendix B)	62
7.3	Local error analysis of the swept- and intersection-based remapping (Appendix C)	63
7.4	Combined swept- and intersection based remapping (Appendix D)	63
7.5	J_2 invariant-preserving remap of the 2D dev. stress tensor (Appendix E)	64
8	Conclusion	67
	Bibliography	69
A	Original Article: Constrained optimization for IASSD models for voids closure	83
B	Original Article: IASSD model for solids with void closure and opening at all speeds	115
C	Original Article: Local error analysis of the swept- and intersection-based remapping	137
D	Original Article: Combined swept- and intersection based remapping	173
E	Original Article: J_2 invariant-preserving remap of the 2D dev. stress tensor	195



Section 1

Introduction

The first mathematically accurate general description of inviscid fluid flow was created by Leonhard Euler in 1755 [21]. For almost two centuries to follow, this system of quasi-linear hyperbolic partial differential equations served as a theoretical basis rather than a tool for practical scientific and engineering calculations. This changed at the beginning of the 20th century when numerical analysis has developed sufficiently to be able to approximate solutions of such systems, although with limited success, partially due to the impracticality of using large teams of human “computers” for such tasks [65]. At the end of the Second World War, first digital electronic computers were built to the task of scientific computation during the Manhattan Project. In the following race for the fusion weapon, this development has further accelerated, leading to machines such as the Mathematical and Numerical Integrator and Calculator (MANIAC), constructed in 1951 and based on John Von Neumann’s architecture. These computers, in many ways direct predecessors of modern hardware, have run the first hydrodynamic codes (or “hydrocodes”) [83] that were actually used for simulations involving shock propagation in a wide range of materials, from gases to solids [103, 104]. Further development has led to applications in supersonic fluid flow [34] with important implications in e.g. aircraft and orbital reentry vehicle design. This has marked the beginning of a new inter-disciplinary research field we now know as Computational fluid dynamics – the study of fluid flows using the tools of numerical analysis and scientific computing.

What has prevailed from 18th century continuum mechanics until modern times are the distinct frameworks of fluid description [93]. The Eulerian method assumes that the observer is static and the flow of a fluid through a volume element is analyzed. This is a natural framework for steady flow problems such as encountered in aerodynamics [29], turbo-machinery engineering [64] etc. For unsteady problems with strong deformations, high shear flows or moving boundaries a different description is more suitable. In the Lagrangian method the observed volume element encompasses a constant mass and moves with the fluid. Among its advantages is the inherent fulfillment of the continuity equation, preservation of internal boundaries (which can separate distinct materials) and inherent adjustment of the computational resolution to the simulated problem (if the fluid is strongly compressed at

one point of the domain, so are the observed elements). The downside of this approach is that in methods which employ a computational mesh for discretizing the flow field, mesh cells can deform to non-convex and self-intersecting polygons, a phenomenon known as *mesh tangling* that leads to a failure of the computation when untreated [99].

In 1974, Hirt et al. described in a seminal paper [37] a generalization of both frameworks which became known as the Arbitrary Lagrangian-Eulerian method or ALE [68]. It allows us to construct a moving mesh which can only partially depend on the fluid flow and can be adjusted as needed. This is realized either through an explicit mesh velocity field (direct ALE) [24, 13, 31, 95] or – as used further in this dissertation – by a three step algorithm which will alter the Lagrangian result only when necessary (indirect ALE) [7]:

- **Lagrangian step** – A time step of the Lagrangian fluid dynamics calculation which advances the flow field and moves the computational mesh.
- **Mesh smoothing** – If necessary (a geometric criterion such as cell condition number predicts poor mesh quality) mesh nodes are moved to a more suitable locations.
- **Quantity remapping step** – If the mesh was smoothed, then the fluid quantities must be interpolated onto the new mesh in a conservative and physically consistent way.

The indirect ALE framework preserves most of the advantages of the Lagrangian approach while allowing for corrections if the mesh quality drops, therefore greatly increasing the robustness of the computation over the pure Lagrangian method.

ALE computations have made numerical simulation of extreme events such as interaction of laser-produced plasma with solid targets [61, 50, 1], explosive-driven devices [25, 5] or asteroid impacts [80] possible. One of the current challenges in this field lies in integrating several material models, often with radically different physics. To model a solid continuum, the Euler equations can be extended by replacing the scalar pressure with the Cauchy stress tensor and a constitutive model defining the stress-strain relationship and material strength (such model was employed already in the 60s [104] and further developed in e.g. [66]). Each material may also have a different equation of state – Mie-Gruneisen (metal) [72], ideal gas, QEOS (tabular EOS model useful for high temperature plasma, [74]). The materials may also interact with each other in the form of contact surface physics, sliding [17] or creating gaps that can close or open, forming a void region (which can be itself treated as a special kind of material [82, 5]).

In Lagrangian computations with several different materials present in the simulation domain, mesh edges can be fixed to material boundaries and follow their movements. This is hard to maintain in an ALE computation when the mesh is being smoothed, so a multi-material model is needed to describe

the interaction of different materials inside a single computational cell. The complexities and physical conditions specified above lead us to prefer the sharp interface approach over the diffuse interface approach (such as the method of concentration equations, briefly described in [33], or the Baer-Nunziato multi-phase model [2, 97, 22]). The non-diffuse interface model constitutes of two logical parts – the first is the interface reconstruction algorithm which determines the geometrical location of the material interface within the computational cell. Here it is possible to use global methods (such as level sets [77, 12]) or localized methods working only within the surroundings of a computational cell, which we prefer. We employ the Volume-of-fluid (VOF) method [38] with interface orientation determined by the Youngs method using volume fraction reconstructed gradient normal [105]. Also other options are possible, such as the ordering-independent power diagrams method [87] etc. An entirely local method we have also utilized is the Moment-of-fluid (MOF) method [28, 36] using advected centroids to determine the interface position and orientation.

However, the main subject of our research is the second part of the multi-material algorithm – modeling the physics of the interaction of different materials in the computational cell. The simplest multi-material model (the equal compressibility model [10, 4]) presumes that if a computational cell changes volume, all materials are compressed equally. This is obviously not generally true from a physics point of view and leads to erroneous values produced near the material interface. More complex are pressure relaxation models [94, 23, 39], such as Tipton’s [96, 90, 3] that defines a common pressure for all materials in the cell and the material volumes are adjusted to conform to the pressure equilibrium. This can, however, distort internal energy near the interface [4]. We focus on further improvement of the Interface-Aware Sub-Scale Dynamics model (IASSD) [4, 5, 41] (Appendix A, B) which explicitly determines the interface movement in the computational cell using an approximate Riemann solver. The algorithm consists of two phases: the bulk phase, which deals with the computational cell as a whole and the sub-scale phase, where only internal interactions between materials are processed using an approximate Riemann solver.

The Lagrangian part of the ALE algorithm is realized using a compatible staggered discretization of the Euler equations with predictor-corrector time integration [17, 16]. This scheme achieves inherent second order accuracy without the difficulties of obtaining a conservative reconstructed values of the field variables. At shock-induced discontinuities, artificial viscosity [75] is required to stabilize the solution which would otherwise undergo oscillatory movement. We use a edge-centered viscosity term [19] among other approaches such as the simpler and more diffusive bulk viscosity or tensor viscosity [19, 49] similar to the physical viscosity of a fluid. A mesh stabilization algorithm preventing the formation of the hourglass-shaped cells is employed [18, 54].

The multi-material remapping algorithm was described in [57, 51, 70] – it involves calculating the gradient of flow variables which can be then used for piece-wise linear interpolation of the target quantity. Slope limiters such as

Barth-Jespersen [9] are employed to ensure the preservation of bounds. The interpolation itself is realized as a flux-formulated process (which guarantees conservativity) and there are various methods [10] to compute the fluxes. To our knowledge the most popular ones either use integration over cell intersections [27] or swept-region based approximation [26, 70] – we compare the performance of those methods in [43] (Appendix C) with a special focus on the symmetry properties of the resulting distribution. We develop a method that can combine both approaches [42] (Appendix D) optimizing the computational expense and solution accuracy.

A related area of research is the remapping of the deviatoric part of the Cauchy stress tensor. Remapping the tensor components independently has a negative influence on the overall accuracy and symmetry of the resulting stress distribution. A similar situation arises when remapping vector quantities and several specialized limiter schemes (i.e. [107, 100]) have been proposed to improve the accuracy. Currently, methods for limiting the stress tensor specifically were proposed as well [84, 67, 15] or an invariant bound limiter for general symmetric tensors [62], which can deal with the unwanted overshoots. However, in our case, the problem is also that the total elastic deformation energy, proportional to the J_2 invariant of the tensor, is not conserved. To achieve conservation, we remap the invariant independently with second-order accuracy and a bounds-preserving limiter and then scale the resulting tensor components to match the remapped invariant [44, 45] (Appendix E).

The dissertation is organized in the following parts: In Section 2 we describe the continuum equations and the overall physics model to be used throughout the dissertation. In Section 3, the Lagrangian part of the numerical algorithm is described in detail. Section 4 focuses on the description of the developed multi-material model for voids, fluids and solids. In Section 5, the remapping part of the numerical algorithm is described, detailing flux construction and remapping of each involved fluid quantity. In Section 6 the results of selected numerical simulations are presented to showcase the abilities of the algorithm. In Section 7 each of the papers that were published in relation to the topics of this dissertation is summarized and the author contribution is specified. In the following Appendices A-E the content of the published papers is attached with detailed explanation of the research.

1.1 New contributions of the Dissertation

This section sums up the novel contributions of the author that are described in this dissertation and the related appended papers. The overall research goals were focused on algorithm development in the multi-material ALE numerical framework for simulating fluid and solid dynamics, implementation and integration of the developed methods within a 2D experimental code that is able to run numerical simulations of a wide range of problems such as low- and high-velocity impacts, cavity closures and shock-driven interactions.

The algorithmic developments related to the IASSD multi-material model in the Lagrangian stage are as follows:

- Introducing the void material for representing vacuum environment, enabling void closure for simulating vacuum gaps and cavities. [5] (Appendix A)
- Including solid materials in the IASSD model, equilibrating normal stress instead of pressure. [41] (Appendix B)
- Enabling void opening between two materials in tension. [41] (Appendix B)
- Integrating all of the above in a single multi-material model.
- Modifying the model so that it represents correctly both high-speed and elastic impacts.

The following contributions are relevant to the remapping stage of the algorithm:

- Analyzing the effects of intersection/swept-based flux calculation schemes on the result symmetry and developing a method that can combine the advantages of both. [43, 42] (Appendix C, D)
- Developing a method for remapping the deviatoric part of the Cauchy stress tensor. [45] (Appendix E)
- Integrating all into a single remapping algorithm compatible with the void framework.

Section 2

Governing Equations

We employ a continuum model described by the time-dependent Euler equations [98] extended with the full Cauchy stress tensor. This system consists of the continuity equation (conservation of mass):

$$\rho_t + \nabla \cdot (\rho \mathbf{u}) = 0, \quad (2.1)$$

the momentum conservation law:

$$(\rho \mathbf{u})_t + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} + \sigma) = 0, \quad (2.2)$$

and the total energy conservation law:

$$E_t + \nabla \cdot [(E \cdot I + \sigma) \mathbf{u}] = 0, \quad (2.3)$$

The subscript t represents the partial time derivative. ρ denotes density, \mathbf{u} velocity vector and $E = \varepsilon + \frac{1}{2} \mathbf{u}^2$ specific total energy and ε the specific internal energy. σ is the symmetric Cauchy stress tensor, which can be split as:

$$\sigma = -pI + S, \quad (2.4)$$

the components are the volumetric part (pressure p) and the traceless deviatoric stress tensor S (which is not present in inviscid fluid materials and the equations (2.1) – (2.3) reduce to the hydrodynamic form [98]). This formulation can therefore be used for description of both fluids and solids, where S is responsible for the elastic-plastic behavior [104, 66]. The momentum conservation law (2.2) is a vector equation, where the \otimes operator denotes the tensor product and the tensor field divergence is defined as:

$$(\nabla \cdot T)_i = \sum_j \frac{\partial T_{ij}}{\partial x_j}, \quad (2.5)$$

for any tensor quantity T .

For our intended application we employ the Lagrangian framework, in which the observed volume element moves with the fluid. This leads to a setting which inherently satisfies the continuity equation (2.1). In the remaining equations, we can expand the time derivatives using the product rule and substitute for the density partial derivative from (2.1).

The total time derivative can then be factored out using the following material derivative formula:

$$\frac{d}{dt} = \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla. \quad (2.6)$$

This leads to the following Lagrangian formulation of the momentum and energy conservation laws [104, 66]:

$$\rho \frac{d\mathbf{u}}{dt} - \nabla \cdot \sigma = 0, \quad (2.7)$$

$$\rho \frac{dE}{dt} - \nabla \cdot (\sigma \mathbf{u}) = 0. \quad (2.8)$$

The energy equation can be expressed using the specific internal energy by substituting into the time derivative:

$$\frac{dE}{dt} = \frac{d\varepsilon}{dt} + \mathbf{u} \cdot \frac{d\mathbf{u}}{dt} = \frac{d\varepsilon}{dt} + \frac{1}{\rho} \mathbf{u} \cdot (\nabla \cdot \sigma), \quad (2.9)$$

and then substituting the result into (2.8) while expanding the divergence operator using the product rule:

$$\rho \frac{d\varepsilon}{dt} - (\sigma \nabla) \cdot \mathbf{u} = 0. \quad (2.10)$$

For the closure of the system, an equation of state must be specified as a relation of pressure, density and internal energy:

$$p = p^{EOS}(\rho, \varepsilon), \quad (2.11)$$

where the actual EOS models employed in this dissertation are specified further in Section 2.2.

2.1 Constitutive model of solids

For solid materials, the deviatoric stress S has non-zero components and its evolution becomes another part of the physics model. For the problems relevant to this dissertation (mostly metals undergoing shock compression), a linear elasticity model is sufficient as a basis of the constitutive model. It is to be noted that our main focus is the general numerical algorithm and not the physics model itself. More complex models (e.g. with strain hardening or material fatigue) can be employed for more realistic results but that is out of the scope of our research.

We follow the approach outlined in [104, 66], expressing the Hooke's law using the objective Jaumann deviatoric stress rate and the deviatoric part of the strain rate:

$$\overset{\circ}{S} = \frac{dS}{dt} + SW - WS = 2\mu \left(D - \frac{1}{3} \text{tr}(D) I \right), \quad (2.12)$$

where μ is the shear modulus of the deformed materials. The strain rate (D) and spin (W) tensors are defined as follows:

$$D = \frac{1}{2} \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right), \quad W = \frac{1}{2} \left(\nabla \mathbf{u} - (\nabla \mathbf{u})^T \right), \quad (2.13)$$

where $\nabla \mathbf{u}$ is the velocity gradient (a tensor quantity).

This model is supplemented with the von Mises yield condition [101] describing material plasticity by limiting the magnitude of the deviatoric stress:

$$\|S\| \leq \sqrt{\frac{2}{3}} Y^0, \quad \|S\| = \sqrt{\text{tr}(S^T S)}. \quad (2.14)$$

The constant Y^0 is the yield strength – it determines the point at which the material begins to deform plastically.

2.2 Equations of state

Several equations of state are used in this dissertation. The simplest is the ideal gas equation of state:

$$p = \rho \varepsilon (\gamma - 1), \quad (2.15)$$

where γ is the adiabatic gas constant (ratio of specific heats).

A stiffened gas equation of state is another simple model, useful for example for the description of liquids under shock compression:

$$p = \rho \varepsilon (\gamma - 1) - \gamma p^\infty, \quad (2.16)$$

which approximates the behavior of near-incompressible fluids as an ideal gas already pressurized to p^∞ .

For computations with solids, the Mie-Grüneisen equation of state [81, 72] is used in the following form:

$$p = \rho_0 c_0^2 f(\eta) + \rho_0 \Gamma_0 \varepsilon, \quad \eta = \frac{\rho}{\rho_0}, \quad (2.17)$$

$$f(\eta) = \frac{(\eta - 1) \left(\eta - \frac{1}{2} \Gamma_0 (\eta - 1) \right)}{(\eta - s(\eta - 1))^2}, \quad (2.18)$$

where ρ_0 is the reference density in a resting state, c_0 the bulk speed of sound, s is the linear Hugoniot slope coefficient and Γ_0 is the Grüneisen's constant at the reference state. Using this equation is appropriate for shock-compressed solids, especially metals.

A special case is the *void* material, which is used to represent ideal vacuum in the simulation. It is not described by an equation of state per se – its pressure, internal energy and density is always zero. It requires a special treatment in the multi-material model as its infinite compressibility implies that its volume can reach zero. The processes in which void can disappear (void closure) or emerge (void opening) and are described further (see Section 4.3).

Section 3

Compatible Lagrangian discrete model

We follow the procedure of compatible discretization [17] (extended to unstructured grids in [16]) of the differential operators in the conservation equations (2.7), (2.10) and the constitutive model in Section 2.1. The staggered quantity representation means that discrete scalar ($\rho_c, p_c, \varepsilon_c$) and tensor (S_c) quantities are centered in computational mesh cells while discrete velocity (\mathbf{u}_n) and other auxiliary vector quantities are defined in mesh nodes. The advantage of this formulation is inherent second-order spatial accuracy without the need of computing linear reconstructions of the involved quantities. In this section, the subscript c will represent a cell-centered and n a node-centered quantity. We will further describe the spatial discretization for a polygonal unstructured mesh.

The Lagrangian framework assumes that the mesh moves with the fluid, resulting in a motion equation for the mesh node:

$$\frac{d\mathbf{x}_n}{dt} = \mathbf{u}_n. \quad (3.1)$$

The momentum equation is integrated over the dual mesh cell that is formed by joining the edge mid-points with cell centroids (see Figure 3.1). We will denote this as control volume $CV(n)$ centered in the node n and the spatial integration defines the semi-discrete velocity equation for the mesh node velocity:

$$m_n \frac{d\mathbf{u}_n}{dt} = \int_{CV(n)} \nabla \cdot \sigma \, dV, \quad (3.2)$$

where m_n denotes the nodal mass. It is an approximation of the mass integral over the control volume:

$$m_n = \sum_{c \in C(n)} m_{c,n}, \quad m_{c,n} = \frac{m_c}{N_c}, \quad m_c = \rho_c V_c \quad (3.3)$$

where $C(n)$ is a set containing all neighboring cells of node n , N_c represents the number of nodes which form the polygonal cell c and $m_{c,n}$ is the corner mass corresponding to the particular cell and node combination. This approximation uses the simplified assumption of an uniform density in the computational cell and equal distribution of the time-constant cell mass m_c to the corners. This way the nodal and corner masses are defined as Lagrangian

quantities as well (whereas an integral of the cell density over the exact corner volume can change in time even for isochoric deformation of the whole cell). This approach is consistent with [17] and it defines a simple relation between the cell and node mass without the need to store independent corner densities and masses.

We will further use the divergence theorem to express (3.2) as a closed line integral over the control volume boundary:

$$m_n \frac{d\mathbf{u}_n}{dt} = \oint_{\partial CV(n)} \sigma \mathbf{n} dl = \sum_{c \in C(n)} \int_{e \rightarrow e+1} \sigma_c \mathbf{n} dl = \mathbf{F}_n, \quad (3.4)$$

where the closed integral is split over subzone segments denoted as $e \rightarrow e + 1$. As shown in [17], the integral over such segments is path-independent as long as the path connects the edge mid-points. Therefore we opt to integrate over the cell edges instead of subzone separators as it can be used consistently with internal energy integration (3.6). The example of the geometry of a subzone between edges e and $e + 1$ is shown in Figure 3.1.

The right side of (3.4) effectively represents a force vector acting on the computational node, which we denote as \mathbf{F}_n . The integral segments correspond to a contribution of a single corner subzone to the total nodal force which we denote as a subzonal force $\mathbf{F}_{n,c}$:

$$\mathbf{F}_n = \sum_{c \in C(n)} \mathbf{F}_{n,c} = \sum_{c \in C(n)} l_e \sigma_c \mathbf{n}_e + l_{e+1} \sigma_c \mathbf{n}_{e+1}, \quad (3.5)$$

where l_e, l_{e+1} is equal to the half of the respective cell edge length and $\mathbf{n}_e, \mathbf{n}_{e+1}$ are the unit normal vectors. See Figure 3.1 for details.

The discrete internal energy equation is formed by integrating (2.10) over the computational cell while presuming that the Cauchy stress tensor remains constant over this domain:

$$m_c \frac{d\varepsilon_c}{dt} = \int_c (\sigma \nabla) \cdot \mathbf{u} dV = \oint_{\partial c} (\sigma_c \mathbf{u}) \cdot \mathbf{n} dl, \quad (3.6)$$

where the integral over the cell boundaries (thermodynamic work) can be also expressed using the same integral segments representing subzonal forces as in (3.4):

$$m_c \frac{d\varepsilon_c}{dt} = \sum_{n \in N(c)} \mathbf{u}_n \cdot \int_{e \rightarrow e+1} \sigma_c \mathbf{n} dl = \sum_{n \in N(c)} \mathbf{u}_n \cdot \mathbf{F}_{n,c}, \quad (3.7)$$

where $N(c)$ is the set of all nodes belonging to the computational cell c and m_c is the total cell mass. The conservation of total energy is guaranteed by this compatible selection of the integration path and by using the same force terms in both momentum and internal energy equations.

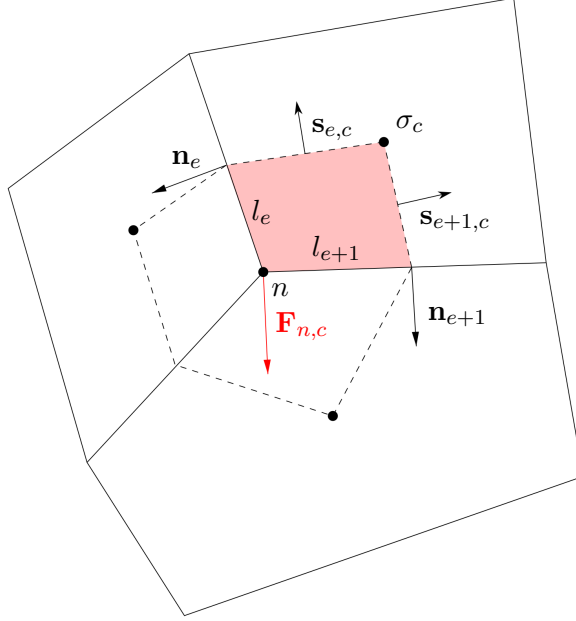


Figure 3.1: Three neighboring mesh cells sharing a node n with the dual mesh polygon displayed (dashed line). Also showing the cell c to node n subzonal elastic force vector $\mathbf{F}_{n,c}$ constructed from half-edge lengths l_e, l_{e+1} , normals $\mathbf{n}_e, \mathbf{n}_{e+1}$ and the cell-centered stress tensor σ_c (subzone area marked with red color). Also showing normal vectors to subzone separators $\mathbf{s}_{e,c}, \mathbf{s}_{e+1,c}$.

It is necessary to obtain the expression for the discrete equivalent of the velocity divergence (it is further required for the algorithm in e.g. time step control). This is done by integration of velocity divergence over the computational cell:

$$(\nabla \cdot \mathbf{u})_c = \frac{1}{V_c} \oint_{\partial c} \mathbf{u} \cdot \mathbf{n} dl = \frac{1}{V_c} \sum_{n \in N(c)} \mathbf{u}_n \cdot (l_e \mathbf{n}_e + l_{e+1} \mathbf{n}_{e+1}). \quad (3.8)$$

In fluid flow the Cauchy stress reduces to a scalar pressure – in such case this divergence term can be substituted in (3.6), showing that the result is consistent with the force-formulated energy equation in (3.7).

For discretizing the evolution equation for the deviatoric stress S we will follow the approach outlined in [104, 66]. We need to derive an expression for the velocity gradient $(\nabla \mathbf{u})_c$ which, as a tensor quantity, is defined in the cell center – we can express it as an integral average over the computational cell:

$$(\nabla \mathbf{u})_c = \frac{1}{V_c} \int_c \nabla \mathbf{u} dV, \quad (3.9)$$

where V_c is the volume of the computational cell.

We can derive the following relation which we will use to convert the volume integral to a boundary integral:

$$\begin{aligned} \int_c (\nabla \mathbf{u})_{ij} dV &= \int_c (\nabla u_i) \otimes \mathbf{e}_j dV = \left(\int_c \nabla \cdot (u_i \mathbf{e}_i) dV \right) \otimes \mathbf{e}_j = \\ &= \left(\oint_{\partial c} (u_i \mathbf{e}_i) \cdot \mathbf{n} dl \right) \otimes \mathbf{e}_j = \oint_{\partial c} u_i \mathbf{n} \otimes \mathbf{e}_j dl. \end{aligned} \quad (3.10)$$

This leads to the following expression for the discrete velocity gradient:

$$(\nabla \mathbf{u})_c = \frac{1}{V_c} \oint_{\partial c} \mathbf{u} \otimes \mathbf{n} dl = \frac{1}{V_c} \sum_{n \in N(c)} \mathbf{u}_n \otimes (\mathbf{n}_e l_e + \mathbf{n}_{e+1} l_{e+1}), \quad (3.11)$$

where \mathbf{n}_e, l_e is the half-edge unit normal vector and length (same geometrical situation as in Figure 3.1). This expression is used to calculate the discrete strain rate and spin tensors (2.13) and then employed in a discrete deviatoric stress evolution equation (2.12) with the plastic yield condition (2.14).

3.1 Multi-material spatial discretization

The sharp interface approach is well suited for simulations of immiscible materials interacting, such as impact and contact problems – we presume that each of the materials occupies a distinct polygonal region within the computational cell (material subpolygon) and each of the materials can have its own constitutive model and state equation. A global representation of the material interface is not used, instead, it is determined locally in each computational cell as an internal edge that is shared by the neighboring material subpolygons (see for example Figure 4.2). The exact geometry of the subpolygons is determined by a *Material reconstruction* algorithm using material quantities (volume, centroids) as an input – the implementation is model-dependent and specified further in Section 3.1.2. Due to the staggered mesh, for cell-centered quantities (ρ, p, ε, S) we distinguish material subpolygon specific values (we will use the c, k subscripts with k representing the material index) and cell-averaged values (denoted with the c subscript only). Node-based fluid velocity is not considered a material-specific value as it is tied to the mesh velocity.

The ratio of volume occupied by one material to the total volume of the computational cell defines the volume fraction $\alpha_{c,k} = V_{c,k}/V_c$. The consistency of total mass and internal energy for material and average values implies the following averaging rules for density (volume fraction-weighted) and specific internal energy (mass fraction-weighted):

$$\rho_c = \sum_{k \in M(c)} \alpha_{c,k} \rho_{c,k}, \quad \varepsilon_c = \sum_{k \in M(c)} \frac{m_{c,k}}{m_c} \varepsilon_{c,k}, \quad (3.12)$$

where $M(c)$ is the set of all materials present in computational cell c . For the remaining non-conservative quantities we will assume a model-dependent weighting factor $\beta_{c,k}$:

$$\begin{aligned} p_c &= \sum_{k \in M(c)} \beta_{c,k} p_{c,k}, & S_c &= \sum_{k \in M(c)} \beta_{c,k} S_{c,k}, \\ \beta_{c,k} &\in [0, 1], & \sum_{k \in M(c)} \beta_{c,k} &= 1 \quad \forall c. \end{aligned} \quad (3.13)$$

As we follow the Lagrangian approach also on the submesh level, the material subpolygon mass is presumed to be constant, which guarantees mass conservation. For momentum, we will need to express the equation (3.4) in terms of the averaged Cauchy stress. Its decomposition in the average of material pressures and deviatoric stresses leads to the following definition of the material-based cell-to-node forces (extending (3.5)):

$$\mathbf{F}_n = \sum_{c' \in C(n)} \sum_{k \in M(c')} \beta_{c',k} \mathbf{F}_{n,c',k}, \quad (3.14)$$

$$\mathbf{F}_{n,c',k} = l_e \sigma_{c',k} \mathbf{n}_e + l_{e+1} \sigma_{c',k} \mathbf{n}_{e+1}, \quad (3.15)$$

which enables us to use the node-based momentum equation (3.4). This formulation also guarantees energy conservation for the cell-averaged internal energy:

$$m_c \frac{d\varepsilon_c}{dt} = \sum_{n \in N(c)} \sum_{k \in M(c)} \beta_{c,k} \mathbf{F}_{n,c,k} \cdot \mathbf{u}_n. \quad (3.16)$$

The corresponding equation is not universally applicable to the material internal energy – it is the responsibility of the multi-material model to define this relation in a compatible way. As the multi-material model is not a full description of the physical state of material subpolygons, thermodynamic approximations in the form of energy fluxes between materials might be required to avoid wall heating-like phenomena and allow correct pressure relaxation within a single cell [3] (see Section 3.1.1 or 4.2).

At last, we need to formulate a multi-material extension of the Wilkins elastic-plastic model. To our knowledge the resources with the description of such model are sparse, mostly incorporating a common strain rate for all materials (theoretically described in [10], practical application in [92]). However, we would like our model to factor in the variations of compressibility in the interacting materials in the computational cell. Therefore we assume not only that the material deviatoric stresses evolve independently but also that there are specific material strain rate tensors (the spin tensors are assumed to be consistent within the cell). As we have only the mesh velocity and not material-based velocity values, we use the following approximations – the cell-averaged strain rate (2.13) is assumed to be distributed to materials proportionally to the weighting parameter $\beta_{c,k}$ and inversely proportional to the volume fraction (because of the division with cell volume in (3.11)):

$$D_{c,k} = \frac{\beta_{c,k} D_c}{\alpha_{c,k}}, \quad W_{c,k} = W_c, \quad (3.17)$$

where the cell-averaged strain rate and spin tensors are computed according to (2.13) and (3.11). The design of the $\beta_{c,k}$ parameter must take into account the volume fraction limit $\alpha_{c,k} \rightarrow 0$ in the denominator (in our case $\beta_{c,k}$ is always dependent on the volume fraction, so this is not of concern). We can easily show the following relations between the total and material deformation (implied by the integral definition of the discrete velocity gradient (3.9)) are valid for both material strain rate and spin by substituting into the following equations:

$$D_c = \sum_{k \in M(c)} \alpha_{c,k} D_{c,k}, \quad W_c = \sum_{k \in M(c)} \alpha_{c,k} W_{c,k}, \quad (3.18)$$

We can now formulate the following evolution equation for the material deviatoric stress by expressing the Hooke's law (2.12) for discrete quantities:

$$\frac{dS_{c,k}}{dt} = 2\mu \left(D_{c,k} - \frac{1}{3} \text{tr}(D_{c,k}) I \right) - (S_{c,k} W_{c,k} - W_{c,k} S_{c,k}). \quad (3.19)$$

We can summarize all the requirements for the closure of the system that the multi-material models must fulfill:

- Determining the volume fraction $\alpha_{c,k}$
- Computing the weight parameters $\beta_{c,k}$
- Performing the material internal energy exchange (if any) and determining material internal energy values $\varepsilon_{c,k}$
- *Determining approximate material subpolygon centroid location – this is required as an input of the MOF material reconstruction algorithm only (Section 3.1.2)*

■ 3.1.1 Simple closure models

The simplest multi-material model is the equal strain (also termed “equal compressibility”) model. It presumes that the volume fractions $\alpha_{c,k}$ are constant in time, meaning that the total volume change of a cell is distributed equally to all materials in the cell. The weighting parameter $\beta_{c,k} = \alpha_{c,k}$ and no internal energy exchange between materials inside the computational cell is allowed. Independent material internal energy equations are formed:

$$m_{c,k} \frac{d\varepsilon_{c,k}}{dt} = \sum_{n \in N(c)} \alpha_{c,k} \mathbf{F}_{n,c,k} \cdot \mathbf{u}_n. \quad (3.20)$$

As the model name suggests, the strain rate is equal for all materials in the computational cell $D_{c,k} = D_c$. Due to the uniform deformation throughout the cell, the approximate material centroids can be calculated by using an isoparametric mapping of each cell from the old to the new time level.

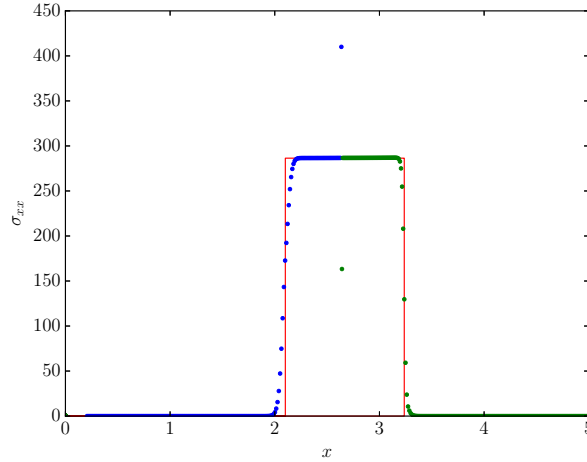


Figure 3.2: 1D Lagrangian simulation of a ● steel plate ($u_x = 2000 \text{ ms}^{-1}$) striking an ● aluminium plate compared with the — exact solution at $t = 1 \mu\text{s}$ after impact. Showing the normal Cauchy stress [kbar] along the impact axis. $N = 401$ computational cells with the equal compressibility model used in the multi-material cell.

This model is very inexpensive in terms of the computational cost, unfortunately for discontinuous states on the material interface (which is often the case) it provides us with physically inconsistent results. In Figure 3.2 a one-dimensional numerical simulation of a high-speed impact problem is shown with incorrect values in the multi-material cell clearly visible (see (6.3), (6.2) for the values of material constants). The equal compressibility assumption leads to stress overshoot in the steel plate and undershoot in the aluminium plate.

As a solution to this issue, pressure relaxation multi-material models have been proposed, such as the Tipton’s model [96, 90, 3]. It is based on the assumption that the material pressures/stresses equilibrate to a common value in the multi-material cell and the volume changes of all materials are then computed accordingly. However, this approach has been shown to produce erroneous internal energy values, especially in cases of shock passing over the material interface [3].

In this dissertation we will further develop the idea of a complex sub-scale dynamics multi-material model, which utilizes the exact interface geometry in the computational cell [6, 35, 4, 5, 41]. The Section 4 and Appendix A, B are devoted to its description.

■ 3.1.2 Material reconstruction

The multi-material model is responsible for determining material volume information (and in some cases centroids), but is not able to define the exact location of each material subpolygon. This is the task of the material or interface reconstruction algorithm, which is a geometrical method of partitioning

the domain based on the information about material locations provided. We need this information locally on the computational cell basis in the form of material subpolygons, global interface reconstruction is unnecessary.

There are two classes of material reconstruction algorithms that are compatible and tested with our multi-material method. The Volume-of-Fluid method [105] uses the volume fraction data from the neighboring cells to define a reconstruction of the volume fraction distribution of each material. The material interface is expected to be perpendicular to the approximated gradient of material volume fraction in the computational cell and is moved to accommodate the material volume given by the multi-material model. The downside of this reconstruction algorithm is that it tends to smooth out the interface features and generally loses accuracy on material segments with few neighbors (filaments etc.). Also it is not strictly local as it requires information from surrounding cells to determine the interface orientation.

The Moment-of-Fluid method [28] is a cell-local method which uses reference material centroid data to determine the exact material partitioning. The material interface geometry in the computational cell is selected so that the material volume is preserved and the cumulative defect of the material centroids is minimized:

$$\Delta M_c = \sqrt{\sum_{k \in M(c)} \|\mathbf{x}_{c,k} - \mathbf{x}_{c,k}^{ref.}\|^2}, \quad (3.21)$$

where $\mathbf{x}_{c,k}^{ref.}$ is the reference centroid (computed by centroid advection in the Lagrangian step or centroid remapping in the remap step). The order in which the material subpolygons are evaluated is determined automatically without requiring any user input. This method is strictly local, however, it requires additional input from the multi-material model in the form of updated approximate material centroid location. For a comparison of the performance of both reconstruction methods see [53].

3.2 Predictor-corrector time integration, algorithm overview

To achieve second-order accuracy in time, a predictor-corrector integration scheme is applied [17]. Time-centered pressure, stress and velocity values are computed in the predictor step and used in the corrector step to advance the fluid state. The old, half-time and new time levels are indicated by the superscript $n, n + \frac{1}{2}, n + 1$, additional index p denotes the predicted values. Subscript describe the spatial and material indices in accordance with Section 3. Δt^n denotes the time step length applicable for the whole time level. An example of one Lagrangian time step as used in this dissertation is given below:

■ **Summary of the primary variables used in the discrete model:**

- **Node-centered** – node location \mathbf{x}_n^n , velocity \mathbf{u}_n^n
- **Cell-based, material-specific** – pressure $p_{c,k}^n$, deviatoric stress tensor $S_{c,k}^n$, specific internal energy $\varepsilon_{c,k}^n$, material volume fraction $\alpha_{c,k}^n$, material weight parameter $\beta_{c,k}^n$, material centroid $\mathbf{x}_{c,k}^n$

Material mass $m_{c,k}$ is constant in the Lagrangian step.

■ **Predictor step:**

1. Compute the predicted cell-to-node stress-based forces using (3.15) along with the artificial viscosity forces (see Section 3.4) and mesh stabilization forces (if applicable, see Section 3.5), the total force is defined as follows:

$$\mathbf{F}_{n,c,k}^n = \mathbf{F}_{n,c,k}^{(\sigma),n} + \mathbf{F}_{n,c,k}^{(Q),n} + \mathbf{F}_{n,c,k}^{(HG),n},$$

the predicted total node force \mathbf{F}_n^n is then established using (3.14).

2. Compute the predicted time-centered nodal velocities by time integration of (3.4):

$$\begin{aligned} \mathbf{u}_n^{n+1,p} &= \mathbf{u}_n^n + \frac{\Delta t^n}{m_n} \mathbf{F}_n^n, \\ \mathbf{u}_n^{n+1/2,p} &= \frac{1}{2} \left(\mathbf{u}_n^n + \mathbf{u}_n^{n+1,p} \right). \end{aligned}$$

3. Move the nodes by using the time-centered velocities by time integration of (3.1):

$$\mathbf{x}_n^{n+1,p} = \mathbf{x}_n^n + \Delta t^n \mathbf{u}_n^{n+1/2,p},$$

and update the geometrical quantities – cell volume $V_c^{n+1,p}$ and cell centroids $\mathbf{x}_c^{n+1,p}$. The new average cell density can now be computed as well $\rho_c^{n+1,p} = \frac{m_c}{V_c^{n+1,p}}$.

4. Execute the material model (predictor phase), obtain $\alpha_{c,k}^{n+1,p}$, $\beta_{c,k}^{n+1,p}$, $\varepsilon_{c,k}^{n+1,p}$, $\mathbf{x}_{c,k}^{(ref.),n+1,p}$. The reference centroid location estimate is computed only if the MOF material reconstruction algorithm is used in the next step.
5. Feed the material reconstruction algorithm with $\alpha_{c,k}^{n+1,p}$, $\mathbf{x}_c^{(ref.),n+1,p}$ to obtain the predicted material subpolygons on time level $n+1, p$. From updated volume fractions compute material volume $V_{c,k}^{n+1,p} = \alpha_{c,k}^{n+1,p} V_c^{n+1,p}$ and material density $\rho_{c,k}^{n+1,p} = \frac{m_{c,k}}{V_{c,k}^{n+1,p}}$.
6. Update the material pressures from the equation of state:

$$p_{c,k}^{n+1,p} = p \left(\varepsilon_{c,k}^{n+1,p}, \rho_{c,k}^{n+1,p} \right).$$

7. Compute the velocity gradient (3.11):

$$(\nabla \mathbf{u})_c^{n+1/2,p} = \frac{1}{V_c^n} \sum_{n \in N(c)} \mathbf{u}_n^{n+1/2,p} \otimes (\mathbf{n}_e l_e + \mathbf{n}_{e+1} l_{e+1}),$$

then, compute the material strain rate and spin tensors (3.17), (2.13):

$$D_{c,k}^{n+1/2,p} = \frac{\beta_{c,k}^{n+1,p}}{2} \left((\nabla \mathbf{u})_c^{n+1/2,p} + (\nabla \mathbf{u})_c^{n+1/2,p,T} \right),$$

$$W_{c,k}^{n+1/2,p} = \frac{1}{2} \left((\nabla \mathbf{u})_c^{n+1/2,p} - (\nabla \mathbf{u})_c^{n+1/2,p,T} \right).$$

Update the material deviatoric stress using the constitutive model (3.19):

$$S_{c,k}^{n+1,p} = S_{c,k}^n + \Delta t \left(2\mu \left(D_{c,k}^{n+1/2,p} - \frac{1}{3} \text{tr} \left(D_{c,k}^{n+1/2,p} \right) I \right) - (S_{c,k}^n W_{c,k}^{n+1/2,p} - W_{c,k}^{n+1/2,p} S_{c,k}^n) \right),$$

and finally enforce the plasticity limit (2.14):

$$\text{if } |S_{c,k}^{n+1,p}| > \sqrt{\frac{2}{3}} Y_k^0 \implies S_{c,k}^{n+1,p} = \frac{S_{c,k}^{n+1,p}}{|S_{c,k}^{n+1,p}|} \sqrt{\frac{2}{3}} Y_k^0.$$

8. Calculate the time-centered predicted material pressures and stresses:

$$p_{c,k}^{n+1/2,p} = \frac{1}{2} (p_{c,k}^n + p_{c,k}^{n+1,p}), \quad S_{c,k}^{n+1/2,p} = \frac{1}{2} (S_{c,k}^n + S_{c,k}^{n+1,p}),$$

followed by the cell-averaged values:

$$p_c^{n+1/2,p} = \sum_{k \in M(c)} \beta_{c,k}^{n+1,p} p_{c,k}^{n+1/2,p},$$

$$S_c^{n+1/2,p} = \sum_{k \in M(c)} \beta_{c,k}^{n+1,p} S_{c,k}^{n+1/2,p}.$$

Corrector:

1. Compute the cell-to-node stress-based forces using (3.15) and the predicted time-centered pressures and stresses $p_{c,k}^{n+1/2,p}$, $S_{c,k}^{n+1/2,p}$. The artificial viscosity and mesh stabilization forces are used with the predictor-based values. The total node force $\mathbf{F}_n^{n+1/2}$ is then established using (3.14).
2. Compute the time-centered nodal velocities by time integration of (3.4):

$$\mathbf{u}_n^{n+1} = \mathbf{u}_n^n + \frac{\Delta t^n}{m_n} \mathbf{F}_n^{n+1/2},$$

$$\mathbf{u}_n^{n+1/2} = \frac{1}{2} (\mathbf{u}_n^n + \mathbf{u}_n^{n+1}).$$

3. Move the nodes by using the time-centered velocities by time integration of (3.1):

$$\mathbf{x}_n^{n+1} = \mathbf{x}_n^n + \Delta t^n \mathbf{u}_n^{n+1/2},$$

and update the geometrical quantities – cell volume V_c^{n+1} and cell centroids \mathbf{x}_c^{n+1} . The new average cell density can now be computed as well $\rho_c^{n+1} = \frac{m_c}{V_c^{n+1}}$.

4. Execute the material model (corrector phase), obtain $\alpha_{c,k}^{n+1}$, $\beta_{c,k}^{n+1}$, $\varepsilon_{c,k}^{n+1}$, $\mathbf{x}_c^{(ref.),n+1}$.
5. Feed the material reconstruction algorithm with $\alpha_{c,k}^{n+1}$, $\mathbf{x}_c^{(ref.),n+1}$ to obtain the final material subpolygons on time level $n + 1$.
6. Update the material pressures from the equation of state:

$$p_{c,k}^{n+1,p} = p\left(\varepsilon_{c,k}^{n+1,p}, \rho_{c,k}^{n+1,p}\right).$$

7. Compute the velocity gradient (3.11):

$$(\nabla \mathbf{u})_c^{n+1/2} = \frac{1}{V_c^n} \sum_{n \in N(c)} \mathbf{u}_n^{n+1/2} \otimes (\mathbf{n}_e l_e + \mathbf{n}_{e+1} l_{e+1}),$$

then, compute the material strain rate and spin tensors (3.17), (2.13):

$$D_{c,k}^{n+1/2} = \frac{\beta_{c,k}^{n+1}}{2} \left((\nabla \mathbf{u})_c^{n+1/2} + (\nabla \mathbf{u})_c^{n+1/2,T} \right),$$

$$W_{c,k}^{n+1/2} = \frac{1}{2} \left((\nabla \mathbf{u})_c^{n+1/2} - (\nabla \mathbf{u})_c^{n+1/2,T} \right).$$

Update the material deviatoric stress using the constitutive model (3.19):

$$\begin{aligned} S_{c,k}^{n+1} = & S_{c,k}^n + \Delta t \left(2\mu \left(D_{c,k}^{n+1/2} - \frac{1}{3} \text{tr} \left(D_{c,k}^{n+1/2} \right) I \right) - \right. \\ & \left. - (S_{c,k}^n W_{c,k}^{n+1/2} - W_{c,k}^{n+1/2} S_{c,k}^n) \right), \end{aligned}$$

and finally enforce the plasticity limit (2.14):

$$\text{if } |S_{c,k}^{n+1}| > \sqrt{\frac{2}{3}} Y_k^0 \implies S_{c,k}^{n+1} = \frac{S_{c,k}^{n+1}}{|S_{c,k}^{n+1}|} \sqrt{\frac{2}{3}} Y_k^0.$$

3.3 Time step considerations

The Courant-Friedrichs-Levy (CFL) condition [20] prescribes the time step upper limit by approximating the time it takes a wave to cross the computational cell:

$$\Delta t^n \leq C_{cfl} \cdot \min_c \frac{l_c^n}{c_c^n}, \quad (3.22)$$

where we set $C_{cfl} \sim 0.25$ to ensure method stability [17]. l_c^n is the characteristic length of cell c , which is rigorously defined as the minimum edge length in the cell. This can often be very restrictive on the time step length and in such case this parameter can be relaxed using the following approximation:

$$l_c^n = \frac{V_c}{\max_{e \in E(c)} l_e}, \quad (3.23)$$

where the cell volume is divided by the maximum edge length in each computational cell. This criterion can in rare cases (very deformed cells) lead to computation failure and the Lagrangian step must be then restarted with a shorter time step. c_c^n is the cell-averaged sound speed. There are several possibilities [4] of determining this value – the actual implications of the choices are not yet well explored. In our experience it is possible to use a relatively simple approximation by mass fraction weighting without a profound influence on the robustness of the computation:

$$c_c^n = \sum_{k \in M(c)} \frac{m_{c,k}}{m_c} c_{c,k}^n, \quad (3.24)$$

where material sound speed $c_{c,k}^n$ is calculated using the formula for longitudinal waves in elastic media [103]. It is composed of the thermodynamic (bulk) and elastic (shear) component:

$$c_{c,k}^n = \sqrt{\frac{K_{c,k}^n + \frac{4}{3}\mu_k}{\rho_{c,k}^n}}, \quad (3.25)$$

where $K_{c,k}^n$ is the material isentropic bulk modulus, which is a state function that can be derived from the EOS using the following formula:

$$K_{c,k}^n = \rho_{c,k}^n \left(\frac{\partial p^{EOS}}{\partial \rho} \right)_s \left(\rho_{c,k}^n, \varepsilon_{c,k}^n \right), \quad (3.26)$$

where p^{EOS} is the pressure state function in the sense of (2.11) and the s subscript denotes constant entropy partial derivative. It is evident that for fluids the shear modulus term disappears and (3.25) simplifies to the known Newton-Laplace equation for speed of sound in fluids $c_{c,k}^n = \sqrt{K_{c,k}^n / \rho_{c,k}^n}$.

Another time step constraint stems from the requirement to control computational cell expansion and compression [17]. This is important to guarantee

energy positivity and also multi-material model consistency [4, 41] (also see Section 4.1). The resulting condition is stated as follows:

$$\Delta t^n \leq C_{DIV} \cdot \min_c \frac{1}{|(\nabla \cdot \mathbf{u})_c^n|}, \quad (3.27)$$

where C_{DIV} is the maximum cell volume change coefficient, a value of $C_{DIV} = 0.8$ is recommended [17]. The discrete velocity divergence is calculated in a consistent way with the internal energy equation (3.8).

The last constraint prevents the time step length from increasing abruptly [17]:

$$\Delta t^n \leq C_{MAX} \Delta t^{n-1}, \quad (3.28)$$

where we set $C_{MAX} = 1.2$. The final time step length is selected to fulfill all the criteria mentioned in this Section.

3.4 Artificial viscosity

The numerical scheme as presented here creates numerical dispersion on discontinuities such as shock waves. To counter these effects an artificial viscosity [75] term is included which reaches non-zero values in areas with high compression. There are several formulations of artificial viscosity which differ in how the viscous stress is approximated – bulk, edge-centered or tensor viscosity approaches [19]. We have selected here the edge-centered formulation which has beneficial properties in symmetry preservation while keeping the implementation and computational costs moderate.

In this formulation the nodal viscosity force consists of edge-centered contributions:

$$\mathbf{F}_n^{(Q)} = \sum_{k \in M(C(n))} \sum_{e \in E(n)} \frac{1}{2} \mathbf{F}_{e,k}^{(Q)}, \quad (3.29)$$

which are summed over all materials present in the surrounding cells and also all edges connecting with the node n . The edge-centered material force is defined as follows:

$$\mathbf{F}_{e,k}^{(Q)} = \sum_{c \in C(e)} q_{e,k} (1 - \Phi_{e,k}) \frac{\min(0, \Delta \mathbf{u}_e \cdot \mathbf{s}_{e,c}) \Delta \mathbf{u}_e}{\|\Delta \mathbf{u}_e\|^2} \quad (3.30)$$

where the $\Delta \mathbf{u}_e = \mathbf{u}_{e_1} - \mathbf{u}_{e_0}$ is the velocity difference at the edge endpoints, $\mathbf{s}_{e,c}$ is the vector perpendicular to the corner separator (as shown in Figure 3.1) and the force is summed over $C(e)$, the set of (two) cells sharing the common edge e . The minimum in the numerator ensures that the viscosity term is used only when the edge is compressed. We use the quadratic formulation of the linear + quadratic viscosity coefficient $q_{e,k}$:

$$q_{e,k} = \rho_e \|\Delta \mathbf{u}_e\| (0.4 c_e + 1.0 \|\Delta \mathbf{u}_e\|), \quad (3.31)$$

where ρ_e, c_e are density and sound speed values reconstructed in edge centers by bilinear interpolation. On quadrilateral meshes it is possible to use a limiter $\Phi_{e,k}$ to diminish the viscosity term in cases of spatially uniform compression, for its description see [19].

3.5 Anti-hourglass mesh stabilization

An unwanted mesh movement mode that can be induced by the Lagrangian algorithm presented here is the hourglass movement – alternating movement of edge endpoints in opposite directions. If untreated, it can lead in certain cases to simulation failure or, if mitigated by mesh rezoning, to distortion of the resulting quantity profile (“checkerboard” patterns).

A possibility to avoid such distortion is to introduce a finer discretization of pressure forces [18] via the corner pressure $p_{c,n,k}$ defined as a variation of the cell-centered pressure:

$$p_{c,n,k} = p_{c,k} + M_{c,k}^F \delta p_{c,n,k}, \quad \delta p_{c,n,k} = (c_{c,k})^2 (\rho_{c,n,k} - \rho_{c,k}), \quad (3.32)$$

where $M_{c,k}^F$ is a coefficient termed the merit factor, which determines the ratio between cell-centered and corner-based pressure approximation. The setting of such constant can be either spatially uniform or adaptive [18], in our case we calculate with $M_{c,k}^F \sim 0.5$ if the simulation requires hourglass suppression. The pressure variation $\delta p_{c,n,k}$ is approximated from the material sound speed equation (3.25) with the assumption of constant entropy and constant sound speed in the computational cell. The corner density $\rho_{c,n,k}$ can be computed similarly to corner mass:

$$\rho_{c,n,k} = \frac{\rho_{c,k} V_{c,k}}{N_c V_{c,n,k}} = \frac{\rho_{c,k} V_c}{N_c V_{c,n}}, \quad (3.33)$$

where a simplified notion that all materials are distributed in all corner volumes evenly is used $V_{c,n,k} = \alpha_{c,k} V_{c,n}$.

Section 4

IASSD multi-material model for fluids, solids and voids

Unlike the equal compressibility or pressure relaxation closure models, the sub-scale approach utilizes exact interface geometry inside the computational cell to determine volume fraction changes, averaging of dynamic quantities and internal energy exchange. This can mimic more closely the actual interface physics and it is also better compatible with the overall Lagrangian scheme as the movement of material interfaces is driven by similar processes that drive the movement of computational mesh edges.

The model has two distinct stages which separate the bulk and internal processes:

1. **bulk stage** – Here, the movement of the whole computational cell is processed, while using a rough compressibility-based approximation of the internal processes. This will allow us to define an approximate material distribution in the new computational cell.
2. **sub-scale stage** – The approximate material interfaces are evolved by treating the interactions of each pair of material sub-cells, while keeping the boundaries of the computational cell static. Acoustic Riemann solvers are used to estimate the interface velocities resulting in corresponding volume and energy exchanges that are limited with physical bounds and pressure equilibration constraints.

The model presented here is a result of continuous development, first presented in [4], where it was applied on fluid dynamics only. In [5] it was extended to problems involving void closure and in [41] it was finally adapted for problems involving both fluids and solids and allowing void closure and void opening. The model presented in this Section corresponds to this latest evolution and for details on its performance and implementation we would like to refer the interested reader to these papers that are included as Appendices A and B.

In the rest of this Section we will omit the cell index c in all variables, because the closure model is defined locally in a single computational cell. Subscript k alone defines a material-dependent quantity, the two subscripts k, l designate an interface-defined quantity (defined on the interface of two neighboring materials in the cell).

4.1 Bulk stage

At this stage we will use a bulk model for the evolution of the multi-material cell as a whole. It is based on the combination of the earlier bulk approximations using the equal compressibility (constant volume fraction) model [4, 5] and a proportional-compressibility based model [6]. The derivation and justification of the model parameters is described in [41] (Appendix B). In the bulk stage, we presume that the weight parameters used for dynamic quantities (3.13) also define the distribution of the bulk volume change:

$$V_k^{bulk,n+1} = V_k^n + \Delta V_k, \quad \Delta V_k = \beta_k \Delta V. \quad (4.1)$$

This assumption means that the parameter β_k is equal to the ratio of bulk stage material volume change and total cell volume change. The bulk model requires that the volume changes of all materials in a cell have an equal sign (or are zero). Therefore this relation is non-ambiguous – if $\beta_k = 1$, then the material k takes all the compression/expansion (depending on the sign of ΔV) of the cell, while if $\beta_k = 0$ the volume of material k does not change at all. It is evident that $\sum_k \beta_k = 1$.

The material and total bulk internal energy change (in the form of $p dV$ relations) must be consistent, which justifies the usage of the β_k parameter to define volume change distribution and weighting of pressure and stress (3.13):

$$m \Delta \varepsilon^{bulk} = \sum_k m_k \Delta \varepsilon_k^{bulk} = \sum_k p_k \Delta V_k = \sum_k p_k \beta_k \Delta V = p \Delta V. \quad (4.2)$$

The volume change distribution is in our view closely linked with the strain rate distribution. In a case of uniform compression or expansion, the material strain rate reduces to an isotropic diagonal tensor with the following magnitude:

$$\|D_k\| = \frac{1}{V_k} \frac{\Delta V_k}{\Delta t} = \frac{\beta_k \Delta V}{\alpha_k V \Delta t} = \frac{\beta_k}{\alpha_k} \|D\|. \quad (4.3)$$

We assume that this bulk relation can be used to determine an approximate strain rate tensor distribution by β_k coefficients as defined in (3.17).

So far we have shown the consistency of the volume change distribution ratio with the stress weighting parameters and strain rate distribution. Now, let's focus on the choices of β_k values. In the equal compressibility (EC) model, the weight parameters for dynamic quantities are equal to the corresponding material volume fractions. This model is reasonably accurate if the material parameters are very similar or if the direction of the deformation is parallel to the material interface. The cell total volume change is distributed equally to all materials, which equals to the following parameter selection:

$$\beta_k^{EC} = \alpha_k. \quad (4.4)$$

On the other hand, the proportional compressibility (PC) model is useful in situations, where the material properties are very different (solid and light gas) and the deformation characteristic direction is perpendicular to the

interface. This model uses the inverse of the bulk modulus to determine the normalized compressibility ratio (such approximation was used already in [73]):

$$\beta_k^{PC} = \frac{\frac{\alpha_k}{K_k}}{\sum_k \frac{\alpha_k}{K_k}}. \quad (4.5)$$

In Appendix B [41] we show that both models preserve the total cell volume and the EC model also preserves the material volume bounds:

$$V^{n+1} \geq V_k^{bulk,n+1} \geq \kappa_{bulk} V_k^n, \quad (4.6)$$

given that the the minimum material volume limit is set lower than the relative cell volume change parameter $\kappa_{bulk} < 1 - C^{DIV}$ defined by the time step criterion (3.27). We can not guarantee minimum volume bound preservation for the PC model.

For this reason we employ a strategy similar to *FCT*-like limiters [59, 106] and we interpolate between the bound-preserving EC model and the PC model which is more accurate in certain situations. This leads to the following definition of the weight parameter:

$$\beta_k = (1 - \theta)\alpha_k + \theta \beta_k^{PC}, \quad 0 \leq \theta \leq 1, \quad (4.7)$$

where the limiter coefficient θ is composed of three components:

$$\theta = \min \left(1, \theta^{lim.}, \theta^{norm}, \theta^{dir.} \right). \quad (4.8)$$

$\theta^{lim.}$ is the actual minimum volume limiter that is defined by the following relation:

$$\theta^{lim.} = \frac{1}{(\beta_k^{PC} - \alpha_k)} \left(\frac{(1 - \kappa_{bulk})V_k^n}{V^n - V^{n+1}} - \alpha_k \right). \quad (4.9)$$

The $\theta^{dir.}$ parameter is an approximation of the typical angle between the deformation characteristic direction and the material interface. We will prefer the EC model in situations where this angle is close to zero or π . A hyperbolic tangent function is applied to make the transition between models steeper:

$$\theta^{dir.} = \frac{1}{2} + \frac{1}{2} \tanh 20 \left(\frac{\sum_{k,l} |\mathbf{n}_{k,l}^T D \mathbf{n}_{k,l}| \mathbf{S}_{k,l}}{\|D\|^2 \sum_{k,l} \mathbf{S}_{k,l}} - \frac{1}{2} \right), \quad (4.10)$$

where $\mathbf{n}_{k,l}$ is the unit normal of the interface between material sub-polygons k, l , $\mathbf{S}_{k,l}$ is the interface length and D is the cell centered strain rate tensor (calculated from the velocity gradient, not material dependent).

As (4.10) loses accuracy if the strain rate tensor norm approaches zero, the third limiting coefficient θ^{norm} will start preferring the EC model in low-strain conditions. This is approximately determined as if the strain rate is lower than one tenth of what is required to reach full plastic loading from unloaded state in one time step:

$$\theta^{norm} = 10 \|D\| / \min_k \frac{Y_{0,k} \alpha_k}{\Delta t \mu_k \sqrt{6}}. \quad (4.11)$$

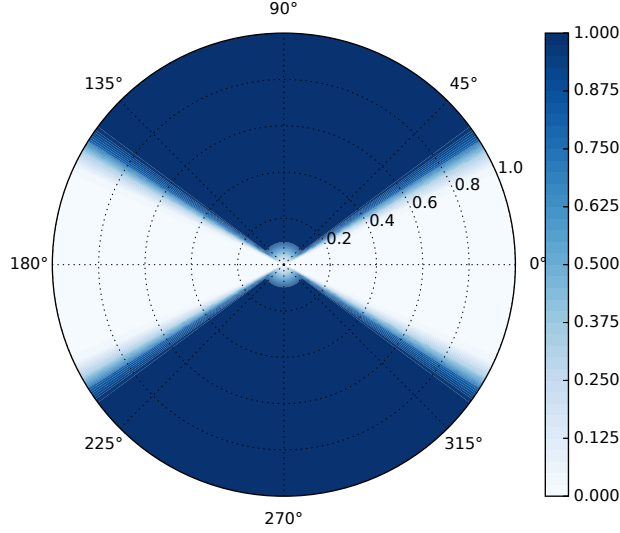


Figure 4.1: A polar color plot showing the dependence of the θ parameter on the angle between the interface and the deformation characteristic direction (the volume limiter is presumed to be inactive). The radial axis represents the normalized strain rate magnitude where 1 corresponds to a strain sufficient to reach the elastic limit in a single time step. Dark color corresponds to full PC model, while light color is full EC.

The influence of the deformation angle and strain rate norm on the overall θ parameter is displayed in Figure 4.1. The parameter range where the EC model is used is denoted with the light color and it clearly shows the main angular dependence with the small spot in the center which corresponds to the usage of EC model in low-strain rate, slow-deforming circumstances.

4.2 Sub-scale dynamics stage

In the sub-scale stage the pair-wise interactions between materials inside the multi-material cell are treated while the cell boundaries are fixed. The principle of the sub-scale approach has been set out in [4], while in Appendix B [41] it was modified to allow stress equilibration and solid materials interaction.

A pair-wise volume exchange between two neighboring materials is defined as follows:

$$\Delta V_k = \Delta V_k^{bulk} + \sum_{l \in M(k)} \delta V_{k,l}, \quad \delta V_{k,l} = \Psi_{k,l} \delta V_{k,l}^{unlim.}, \quad (4.12)$$

where $\delta V_{k,l}^{unlim.}$ is the unlimited volume exchange based on the local approximate Riemann solver [4]. This is done to mimic the Lagrangian computation on the cell boundaries without an exact velocity information. Because the acoustic Riemann solver does not take into account physical bounds of the subcell distribution, interface-based limiter coefficients $0 \leq \Psi_{k,l} \leq 1$ are applied.

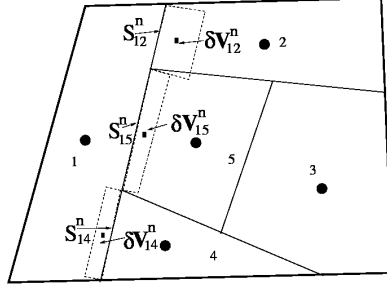


Figure 4.2: Multi-material cell: an example of the sub-cell geometry and relative volume exchange approximation.

The sub-scale volume exchange is computed as follows:

$$\delta V_{k,l}^{unlim.} = u_{k,l}^* \mathbf{S}_{k,l} \Delta t, \quad (4.13)$$

where $\mathbf{S}_{k,l}$ is the interface length (see sub-cell geometry example shown in Figure 4.2), $u_{k,l}^*$ is the interface velocity calculated approximately using the acoustic Riemann solver:

$$u_{k,l}^* = \frac{\tilde{\sigma}_{l,k}^{bulk,n+1} - \tilde{\sigma}_{k,l}^{bulk,n+1}}{\rho_l c_l + \rho_k c_k}, \quad (4.14)$$

where c_l , c_k are the respective material sound speeds including the elastic wave speed as in (3.25). $\tilde{\sigma}_{k,l}^{bulk,n+1}$ is an isentropic approximation of the stress in material k after the bulk stage, normal to the k,l interface (it is a scalar value):

$$\tilde{\sigma}_{k,l}^{bulk,n+1} = p_k^n - \mathbf{n}_{k,l} S_k^n \mathbf{n}_{k,l} - \frac{\rho_k^n (c_k^n)^2}{V_k^n} \Delta V_k^{bulk,n+1}, \quad (4.15)$$

where $\mathbf{n}_{k,l}$ is the unit normal vector on the interface and S_k^n is the material deviatoric stress tensor. Please note that in general $\tilde{\sigma}_{k,l}^{bulk,n+1} \neq \tilde{\sigma}_{l,k}^{bulk,n+1}$ as both terms represent the stress in different materials in the direction of their common interface.

The volume change can also generate an energy flux that is calculated as the thermodynamic work performed by the moving interface:

$$\delta E_{k,l}^{unlim.} = \sigma_{k,l}^* \delta V_{k,l}^{unlim.}, \quad (4.16)$$

where $\sigma_{k,l}^*$ is the interfacial normal stress – unlike previous normal bulk stress terms, this one is an interface-based value defined by the jump condition with an additional viscosity-like term:

$$\sigma_{k,l}^* = \frac{\tilde{\sigma}_{l,k}^{bulk,n+1} \rho_k c_k + \tilde{\sigma}_{k,l}^{bulk,n+1} \rho_l c_l}{\rho_l c_l + \rho_k c_k} + \frac{\rho_l a_l \rho_k a_k V^n (\nabla \cdot \mathbf{u})^n}{\rho_l a_l + \rho_k a_k}, \quad (4.17)$$

where a_k is the thermodynamic/plastic sound speed corresponding to (3.25) without the shear modulus elastic term.

4.2.1 Constraints on material volume change

The $\Psi_{k,l}$ limiter values in (4.12) are chosen so that they are least restrictive while respecting the physical constraints of the multi-material cell system. The limiter values are computed by solving the following quadratic optimization problem in each computational cell:

$$\min_{\Psi_{k,l}} \left\{ \sum_k \left[\sum_{l \in M(k)} (1 - \Psi_{k,l})^2 \right] \right\}, \quad (4.18)$$

where $M(k)$ is the set of all materials in the multi-material cell sharing a common face with material k . If the constraints are specified as linear inequalities, then this can be solved using standard quadratic constrained optimization tools [76]. The ‘‘QL’’ software tool [86] is used to solve this problem in practice.

There are three constraints imposed on the volume change by this limiter:

- **Minimal sub-cell volume** – The first constraint limits the compression of an individual sub-cell in the following way:

$$V_k^{n+1,bulk} + \sum_{l \in M(k)} \Psi_{k,l} \delta V_{k,l}^{unlim.} \geq \kappa_{bot} V_k^{n+1,bulk}, \quad 1 \geq \kappa_{bot} > 0, \quad (4.19)$$

where $\kappa_{bot} \sim 0.1$ is a constant defining the minimal volume decrease during the sub-scale step. Compared to κ_{bulk} , it is much lower on purpose as we expect larger volume exchanges to be allowed during this stage.

- **Positivity of internal energy** – To ensure that the specific internal energy does not reach negative values, the following constraint is applied:

$$m_k \varepsilon_k^{bulk} + \sum_{l \in M(k)} \sigma_{k,l}^* \Psi_{k,l} \delta V_{k,l}^{unlim.} > 0, \quad (4.20)$$

- **Normal stress equilibration** – The interface movement must be controlled in a way that the Cauchy stress in all sub-cells changes towards (or at least not diverge from) a common equilibrium value. To formulate such condition, we approximate the resulting stress value depending on the volume change:

$$\tilde{\sigma}_{k,l}^{n+1} \simeq \tilde{\sigma}_{k,l}^{bulk,n+1} - \frac{\rho_k^n (c_k^n)^2}{V_k^n} \sum_{l \in M(k)} \Psi_{k,l} \delta V_{k,l}^{unlim.}. \quad (4.21)$$

This approximation is based on the assumption of a sufficiently short time step to consider the entropy during interface movement as constant. The target stress value is defined on each interface as a compressibility averaged stress value (from all materials in the cell) projected onto the interface normal:

$$\bar{\sigma}_{k,l} = \sum_{j \in M} \beta_j \left(p_j^n - \mathbf{n}_{k,l} S_j^n \mathbf{n}_{k,l} - \frac{\rho_j^n (c_j^n)^2}{V_j^n} \Delta V_j^{bulk,n+1} \right). \quad (4.22)$$

If we express the difference of the expected (4.21) and target normal stresses we end up with the following constraints, depending on whether the bulk stage normal stress is below or above the target:

If $\tilde{\sigma}_{k,l}^{bulk,n+1} \geq \bar{\sigma}_{k,l}$, then:

$$\frac{V_k^n}{\rho_k^n (c_k^n)^2} (\tilde{\sigma}_{k,l}^{bulk,n+1} - \bar{\sigma}_{k,l}) \geq \sum_{l \in M(k)} \Psi_{k,l} \delta V_{k,l}^{unlim.} \geq 0, \quad (4.23)$$

If $\tilde{\sigma}_{k,l}^{bulk,n+1} \leq \bar{\sigma}_{k,l}$, then:

$$\frac{V_k^n}{\rho_k^n (c_k^n)^2} (\tilde{\sigma}_{k,l}^{bulk,n+1} - \bar{\sigma}_{k,l}) \leq \sum_{l \in M(k)} \Psi_{k,l} \delta V_{k,l}^{unlim.} \leq 0, \quad (4.24)$$

The maximum $\Psi_{k,l}$ values that satisfy all of the constraints specified above guarantee a consistent quantity distribution from both numerical and physical point of view.

4.2.2 Material centroid advection

If the MOF material reconstruction is used later, the multi-material model must determine an estimate of the centroid location of the new material distribution. This reference centroid is computed by updating the bulk stage material centroids with interface-based first moment fluxes:

$$\mathbf{x}_k^{ref.,n+1} = \frac{1}{V_k^{n+1}} \left(V_k^{bulk,n+1} \mathbf{x}_k^{bulk,n+1} + \sum_{l \in M(k)} \delta V_{k,l} \mathbf{x}_{k,l} \right), \quad (4.25)$$

where $\mathbf{x}_{k,l}$ are the centroids of the interface volume change zone, that is approximated as a rectangle formed by the interface line $\mathbf{S}_{k,l}$ with an area equal to $\delta V_{k,l}$ (an example shown in Figure 4.2).

4.3 Void treatment

If the void material is present in the multi-material cell, a special treatment is required as many steps of the model used would not work with zero stress, mass and energy. In this section we will describe all void-specific amendments of the described multi-material model. In the proportional compressibility model in the *bulk stage*, we get to an infinite compressibility limit – if a multi-material cell contains any amount of the void material, the β_k^{PC} parameter is defined as follows:

$$\beta_k^{PC} = \begin{cases} 1 & \text{if } k \text{ is a void material} \\ 0 & \text{if } k \text{ is a non-void material.} \end{cases} \quad (4.26)$$

This is consistent with the fact that the void sub-cell will preferentially change volume at this stage. The equal compressibility part remains unchanged but

we can clearly see that it will not produce accurate results (unless it is a solid-void material combination compressed along the interface).

In the *subscale stage* the interface velocity (4.14) is calculated similarly, using zero stress and density for the void material. However, if one of the interacting materials is void we set $\sigma_{k,l}^*$ to the normal stress value of the non-void material (see Appendix A [5]), which yields a good expansion velocity estimate at the cost of a minor total energy discrepancy (energy flux in void is effectively lost).

Additionally, the internal energy constraint is not applicable in void material as its energy is always zero. If a void material is present in the cell, than all stress equilibration targets (4.22) are set to $\bar{\sigma}_{k,l} = 0$ (we presume that all stress should equilibrate to zero unless the void disappears from the cell).

4.3.1 Closure

Void closure is a process in which void cavities in fluids or solids are compressed to a point that the void effectively disappears from the computational cell. This effect can be used to model flight-impact transitions or cavities collapse [5]. To achieve this, we need to set a numerical void volume fraction threshold (we employ the value $\alpha^{closure} \sim 10^{-6}$) below which the void sub-cell is removed from the cell entirely. Then the volume fractions of the remaining materials must be renormalized to ensure that they always sum to one.

Void closure can happen both in the bulk and sub-scale stages. It requires changes to the minimum volume limiting criteria at both stages:

$$V_{void}^{n+1,bulk} \geq 0, \quad V_{void}^{n+1,bulk} + \sum_{l \in M(oid)} \Psi_{v,l} \delta V_{void,l}^{max} \geq 0. \quad (4.27)$$

These constraints are used for the void sub-cell only, to let the void volume reach zero. Other materials in the multi-material cell are not affected. If the void volume decreases below the closure threshold, it is removed right after new volume fractions are calculated at both stages. Then we renormalize both α_k and β_k and the material reconstruction will generate a consistent sub-cell geometry afterwards.

4.3.2 Opening

While void closure can enable the simulation of closing gaps, the negative stress values in solid materials can also lead to situations where the gaps will open on the interface of two materials (separation, rebound after impact etc.). In this Section we describe an algorithm for void opening in such cases where the contact interface location is previously known.

The material pairs that allow a void to emerge inbetween are designated as “contact pairs”. This might be applied selectively – for example we might want two different solid bodies to separate while solid and gas materials are to remain bonded (neglecting cavitation effects). As it is the tension between materials in contact that triggers void opening, we assume that it is evaluated only in the sub-scale stage.

Unfortunately, due to the limiting conditions, interface tension cannot be used as a sole criterion for void opening. Instead we use a trial-and-error algorithm which will try to open the void at the interface location. If it fails, the sub-scale step is recomputed with the void absent from the cell. The void opening algorithm can be summarized as follows:

1. A *void seed*, a zero volume void sub-cell in place of the contact pair interface, is inserted. The connectivity of the sub-mesh is changed so that the neighboring sub-cells are **material 1 – zero volume void – material 2**.
2. The sub-scale stage is now performed with the void sub-cell in place.
3. If the void sub-cell exceeds the void closure volume fraction threshold (10^{-6}) the void is considered as opened, we perform the material reconstruction and the sub-scale step is completed.
4. If the void did not open, the sub-scale step is restarted with the original sub-mesh with **material 1 – material 2** interface connectivity. Material reconstruction is performed and the sub-scale step is completed with the interface gap remaining closed.

It can happen that there is a contact pair interface where a void can open in a cell that already contains a void. In such situation we want to avoid creating a new void sub-cell on each time step. Instead, we allow opening a second void sub-cell in the sub-scale stage but afterwards we merge both void sub-cells before the material reconstruction is invoked. In all other parts of the numerical computation, only one void sub-cell can be present in a single computational cell. The process of opening a second void and merging is illustrated in Figure 4.3 where void sub-cell 1 is present, another void opens on the interface of materials 2 and 3 and is merged with 1 at the end of the sub-scale step. The resulting reference centroid of the merged void is computed as a combination of the both contributing void subcells $c_{1U4}^{ref.} = (\tilde{V}_1 c_1^{ref.} + \tilde{V}_4 c_4^{ref.}) / (\tilde{V}_1 + \tilde{V}_4)$.

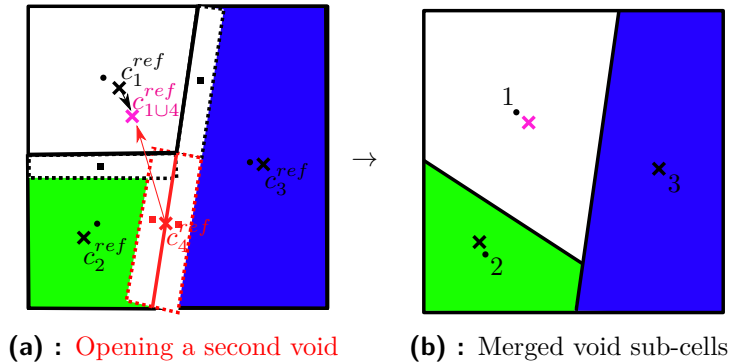


Figure 4.3: Example of the addition of a newly opened void sub-cell to a cell with a pre-existing void. The sub-cells are merged and the final reconstructed sub-mesh is shown.

Section 5

Mesh smoothing and remapping of the fluid quantities

In this Section, the rezone and remap phases of the ALE algorithm are described in detail. First an overview of several mesh smoothing methods is presented, followed by a general overview of the process of remapping a single quantity (reconstruction, limiting, integration and various methods of flux construction). Then, the remap of each of the remaining quantities is discussed separately.

5.1 Mesh rezoning

High-speed flows and shear movements can lead to mesh tangling in Lagrangian simulations by creating non-convex and self-intersecting cells. To prevent this situation, we can move the nodes to get better mesh quality – a process called mesh rezoning. The rezoning methods used here preserve mesh connectivity and cell convexity. As there are many approaches to mesh optimization, the following list is non-exhaustive and represents only the methods that we choose to employ:

- **Initial mesh rezoning** – Possibly the simplest rezoning algorithm, which returns the mesh to the initial state after each time step. The resulting simulation is often termed *Eulerian as Lagrange + Remap* for obvious reasons [79].
- **Prescribed rezoning** (also *cyclic rezone/remap*) – Used mainly for testing the performance of the remap algorithm (see Appendix C, D [43, 42]). There is a prescribed sequence of grids that are used at each time step. Most often the grid sequence is cyclic, as we can compare the results and estimate the remapping error after we reach the same grid setting as on the initial step. This can be also applied in a time-static manner without the Lagrangian step to stress-test the remapping algorithm alone.
- **Simple averaging** – Each node of the computational mesh moves to a location defined by the average of all connected nodes. This approach

is very simple to implement, unfortunately it often results in excessive smoothing of the mesh. [47]

- **Winslow smoothing** – A more sophisticated approach, which is based on a partial solution of the elliptical partial differential equations that leads to mesh as close to equidistant distribution as possible [48].

There are many more mesh smoothing strategies, see for example [14, 46, 102].

5.2 Flux-form remapping strategies

Mesh smoothing constructs a more optimal mesh and we need to perform an interpolation, also termed *remapping*, of all fluid quantities that fulfills the conservation of mass, momentum and total energy (2.1)–(2.3). We will use the original values on the Lagrangian mesh as the basis for the interpolation process and a flux-based remapping formulation [26, 55, 57]. As the material distribution on the rezoned mesh is not yet known, we first need to remap the material volume in the neighborhood of multi-material mesh cells, constructed as a sum of the original value and the inter-cell fluxes:

$$V_{\tilde{c},k} = V_{c,k} + \sum_{c' \in S(c,k)} F_{c,c',k}^V, \quad (5.1)$$

where k is the material index, c is the cell on the original mesh, \tilde{c} the cell on the rezoned mesh, $V_{c,k}$, $V_{\tilde{c},k}$ represent the material volume in the original and rezoned mesh cell, $S(c,k)$ is the neighborhood of cell c (including both edge and corner neighbors) also containing material k and $F_{c,c',k}^V$ is the volume flux of material k from cell c' to c . The material volume flux is always constructed as a volume of the exact intersections of mesh cells (see Figure 5.1 and Section 5.2.2). If we swap the order of the receiving and supplying cell, the flux term changes sign:

$$F_{c,c',k}^V = -F_{c',c,k}^V, \quad (5.2)$$

which guarantees that the geometric conservation law is fulfilled. After this step we have obtained material volume information on the rezoned mesh, but the material subpolygon geometry is not yet known. At this point we can use either use a VOF material reconstruction to determine the geometry or we can proceed to remap of the first moment of the flux region:

$$\mathbf{x}_{\tilde{c},k} V_{\tilde{c},k} = \mathbf{x}_{c,k} V_{c,k} + \sum_{c' \in S(c,k)} F_{c,c',k}^{\mathbf{x}}, \quad (5.3)$$

where the first moment flux $F_{c,c',k}^{\mathbf{x}}$ is constructed by integrating $\int \mathbf{x} dV$ over the intersection regions. The remapped centroid location $\mathbf{x}_{\tilde{c},k}$ can then be used as an input for the MOF material reconstruction algorithm to determine the final material geometry.

After obtaining remapped material volumes, other quantities follow and are remapped in a similar way (here we show density as an example):

$$\rho_{\bar{c},k} V_{\bar{c},k} = \rho_{c,k} V_{c,k} + \sum_{c' \in S(c,k)} F_{c,c',k}^m, \quad (5.4)$$

where $F_{c,c',k}^m$ represents the material k mass flux from cell c' to c . The remapping fluxes $F_{c,c',k}^m$ are in general computed by integrating a reconstructed density function over a flux domain. There are different approaches to construction of the flux domain: The exact method uses intersecting regions of cell polygons. In single-material cases we can approximate these by regions swept by the cell edges and save a non-negligible amount of computational time. Combined (hybrid) methods are also used, which can locally determine which method is more optimal. These will be detailed in further subsections.

5.2.1 Piece-wise linear reconstruction

To construct a second-order method we use a piece-wise linear interpolation to obtain a reconstructed density function inside the original cell c :

$$\rho_{c,k}(\mathbf{x}) = \rho_{c,k} + (\nabla \rho)_{c,k} \cdot (\mathbf{x} - \mathbf{x}_{c,k}), \quad (5.5)$$

where $\mathbf{x}_{c,k}$ is the material centroid location on the original mesh. Each computational cell yields a distinct linear reconstruction of density. The density gradient approximation $(\nabla \rho)_{c,k}$ is obtained using the least squares method [71] (detailed further). This method does not guarantee the preservation of local extrema – to avoid overshoots a limiting coefficient $\varphi_{c,k}$ is used:

$$(\nabla \rho)_{c,k} = \varphi_{c,k} (\nabla \rho)_{c,k}^{unlim.}, \quad 0 \leq \varphi_{c,k} \leq 1. \quad (5.6)$$

The gradient is estimated by minimizing the quadratic error functional:

$$F [(\nabla \rho)_{c,k}] = \sum_{c' \in S(c,k)} \left(\rho_{c',k} - \rho_{c,k}^{unlim.}(\mathbf{x}_{c',k}) \right)^2, \quad (5.7)$$

where $\rho_{c,k}^{unlim.}(\mathbf{x})$ is a density reconstruction (5.5) using the unlimited gradient value. The functional is differentiated by density gradient components [50] yielding the following system of equations:

$$\begin{pmatrix} a_{xx} & a_{xy} \\ a_{xy} & a_{yy} \end{pmatrix} \cdot (\nabla \rho)_{c,k}^{unlim.} = \begin{pmatrix} b_x \\ b_y \end{pmatrix}, \quad (5.8)$$

$$a_{\alpha\beta} = 2 \sum_{c' \in S(c,k)} (\alpha_{c',k} - \alpha_{c,k})(\beta_{c',k} - \beta_{c,k}),$$

$$b_\alpha = 2 \sum_{c' \in S(c,k)} (\alpha_{c',k} - \alpha_{c,k})(\rho_{c',k} - \rho_{c,k}),$$

where α, β can be replaced by x, y to obtain respective material centroid x - and y - coordinate. In most cases this system can be solved by a direct matrix inversion:

$$(\nabla \rho)_{c,k}^{unlim.} = \frac{1}{a_{xx}a_{yy} - a_{xy}^2} \begin{pmatrix} b_x a_{yy} - b_y a_{xy} \\ b_y a_{xx} - b_x a_{xy} \end{pmatrix}. \quad (5.9)$$

The matrix determinant serves as a neighborhood regularity measure. If it approaches zero there are either less than two neighboring cells containing material k or their centroids are linearly dependent. In such cases a direct estimate of the gradient can be made:

$$(\nabla \rho)_{c,k}^{unlim.} = \frac{\sum_{c' \in S(c,k)} (\rho_{c',k} - \rho_{c,k}) (\mathbf{x}_{c',k} - \mathbf{x}_{c,k})}{\sum_{c' \in S(c,k)} \|\mathbf{x}_{c',k} - \mathbf{x}_{c,k}\|^2}. \quad (5.10)$$

To determine $\varphi_{c,k}$ we use the Barth-Jespersen limiter formulation [8] which takes the minimum limiter value from each node of cell c :

$$\varphi_{c,k} = \min_{n \in N(c)} \varphi_{c,k}^n, \quad (5.11)$$

at each node n we adjust the gradient so that the reconstructed density at the node does not exceed the local extrema in the neighborhood of cell c including the cell itself:

$$\varphi_{c,k}^n = \begin{cases} \min \left(1, \frac{\rho_{c,k}^{\max} - \rho_{c,k}}{\rho_{c,k}^{unlim}(\mathbf{x}_n) - \rho_{c,k}} \right) & \text{for } \rho_{c,k}^{unlim}(\mathbf{x}_n) - \rho_{c,k} > 0 \\ \min \left(1, \frac{\rho_{c,k}^{\min} - \rho_{c,k}}{\rho_{c,k}^{unlim}(\mathbf{x}_n) - \rho_{c,k}} \right) & \text{for } \rho_{c,k}^{unlim}(\mathbf{x}_n) - \rho_{c,k} < 0 \\ 1 & \text{for } \rho_{c,k}^{unlim}(\mathbf{x}_n) - \rho_{c,k} = 0 \end{cases},$$

$$\rho_{c,k}^{\min} = \min_{c' \in (S(c,k) \cup c)} \rho_{c',k}, \quad \rho_{c,k}^{\max} = \max_{c' \in (S(c,k) \cup c)} \rho_{c',k}.$$

5.2.2 Fluxes by integrating over exact intersections

The areas where a material subpolygon in the computational cell on the original mesh overlaps with the neighboring cell on the rezoned mesh represent individual flux regions. This is the basic idea behind intersection-based flux construction [55, 57]. This method is well applicable to multi-material cells and does not introduce additional geometrical error in determining the flux regions.

The principle of the method can be shown by analyzing the inter-cell mass flux (5.4) – first we split the term in two distinct cases of possible intersections:

$$F_{c,c',k}^m = F_{(c',k) \cap \tilde{c}}^m - F_{(c,k) \cap \tilde{c}'}^m, \quad (5.12)$$

where the notation $(c', k) \cap \tilde{c}$ represents the intersection of the subpolygon of material k in cell c' with the rezoned mesh cell \tilde{c} . The splitting of the flux in two parts is detailed in Figure 5.1, along with the material partitioning. We can see that there are fluxes present both between corner and edge neighbors.

Note that in the case of edge fluxes, both positive and negative terms can be present at the same time if the old and rezoned edge intersect – as seen on the left edge of the rezoned cell in the figure.

We can then integrate the density reconstruction over each of the intersection polygons. Note that in each part we are using a different reconstruction resulting from the values in cell c or c' correspondingly:

$$F_{(c',k)\cap\tilde{c}}^m = \int_{(c',k)\cap\tilde{c}} \rho_{c',k}(\mathbf{x}) dV, \quad F_{(c,k)\cap\tilde{c}'}^m = \int_{(c,k)\cap\tilde{c}'} \rho_{c,k}(\mathbf{x}) dV. \quad (5.13)$$

We can now substitute from (5.5) and expand the integral of the reconstructed function:

$$F_{(c',k)\cap\tilde{c}}^m = \rho_{c',k} \int_{(c',k)\cap\tilde{c}} dV + (\nabla\rho)_{c',k} \cdot \left(\int_{(c',k)\cap\tilde{c}} \mathbf{x} dV - \mathbf{x}_{c',k} \int_{(c',k)\cap\tilde{c}} dV \right).$$

The remaining integrals are purely geometrical and are in practice pre-computed at the beginning of the algorithm as they are re-used throughout the remapping step for integration of several quantities. Determining the integration region is the task of the intersection algorithm. This has to be done in a robust way as the intersection polygons often have very high aspect ratio and the intersecting edges are close to parallel, requiring combination of analytical and numerical (bisection) line intersection estimates [57].

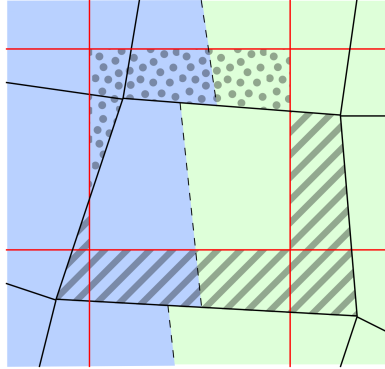


Figure 5.1: Flux regions constructed from exact intersections. – original mesh, – rezoned mesh. Blue and green colors represent material distribution on the original mesh. Striped areas denote intersection flux regions with negative sign, dotted areas are flux regions with positive sign.

5.2.3 Flux regions swept by the cell edges

In single-material situations a simpler algorithm can be used to determine the flux regions by neglecting corner neighbors and approximating edge fluxes only by regions swept by the cell edges [63, 69, 70]:

$$F_{c,c',k}^m = \sum_{e \in E(c)} \int_{\Delta e} \rho_{\eta,k}(\mathbf{x}) dV, \quad \Delta e = [\tilde{e}_0, \tilde{e}_1, e_1, e_0] \quad (5.14)$$

where Δe is the swept polygon defined by the endpoints of the edge on the original mesh (e_0, e_1) and rezoned mesh (\tilde{e}_0, \tilde{e}_1). η defines which cell we use as a basis of the density reconstruction by taking the sign of the oriented volume integral:

$$\eta = \begin{cases} c' & \text{if } \int_{\Delta e} dV > 0 \\ c & \text{if } \int_{\Delta e} dV < 0 \end{cases}.$$

Figure 5.2 shows the construction of the swept regions on all edges of a single computational cell. We can see that the geometric conservation is fulfilled for this configuration as well, although the integrated function (and in some cases also the flux region) differs. On the left edge there are no separate zones as when using intersection remapping, it is computed as a single integral over the self-intersecting polygon and a single reconstruction function is selected. No corner fluxes are present as these regions are incorporated in the edge flux regions. In the upper right corner, we can see that the two swept regions overlap (and the fluxes have opposite signs).

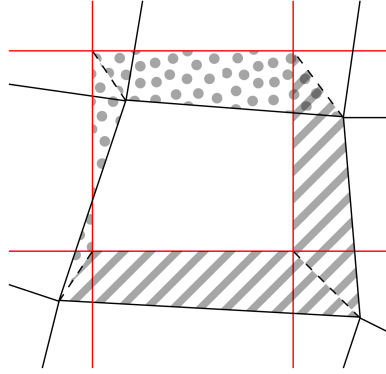


Figure 5.2: Swept flux regions. – original mesh, – rezoned mesh. Striped areas are fluxes with negative sign, dotted areas are fluxes with positive sign.

Due to these approximate flux region constructions, this method saves computational expense but also can generate locally higher error and symmetry distortions. However, it cannot be said that the method is in general less accurate. In several cases [69, 58, 60] it was shown that the method can produce comparable accuracy or even surpass the exact intersection method.

We refer the interested reader to a detailed analysis of the remapping error performed in Appendix C [43], which identifies certain mesh movement modes and quantity distribution profiles that can favor either the swept or intersection remapping method. This was used as a basis for combined remapping methods which can switch between both approaches and are detailed in the next Section.

■ 5.2.4 Hybrid remapping techniques

The concept of hybrid remapping was developed in [11, 52, 56] as a way to lower the computational cost of remapping by using exact intersections

only in multi-material cells and their surroundings, while in the remaining parts of the computational domain the less resource-intensive swept-based remapping is used. Inspired by this approach, we have also developed a combined remapping algorithm for single-material computations where using swept-based remapping leads to symmetry violations (Appendix D [42]). As shown in [43], this can happen in radially symmetrical solutions on non-conforming meshes. This method uses a switching function estimating which method is more suitable for use in the particular computational cell.

As both algorithms share a similar structure, they can be efficiently combined, resulting in the following procedure (inspired by the concept of two-step hybrid remapping [52]):

1. At the beginning of the remapping step, determine for each cell whether it should use intersection- or swept-based remapping. Intersections are used both for cells containing more than 1 material and single-material cells marked by the switching function (Appendix D [42]).
2. An intermediate mesh is formed (see Figure 5.3a), combining original mesh location for nodes of marked cells and rezoned mesh location for all other.
3. Swept region fluxes are calculated between the original and intermediate mesh.
4. Intersection based fluxes are calculated between the intermediate and rezoned mesh.

In Figure 5.3 the process of constructing the intermediate mesh is displayed. There is an additional layer of buffer cells surrounding the marked region where both types of fluxes are computed.

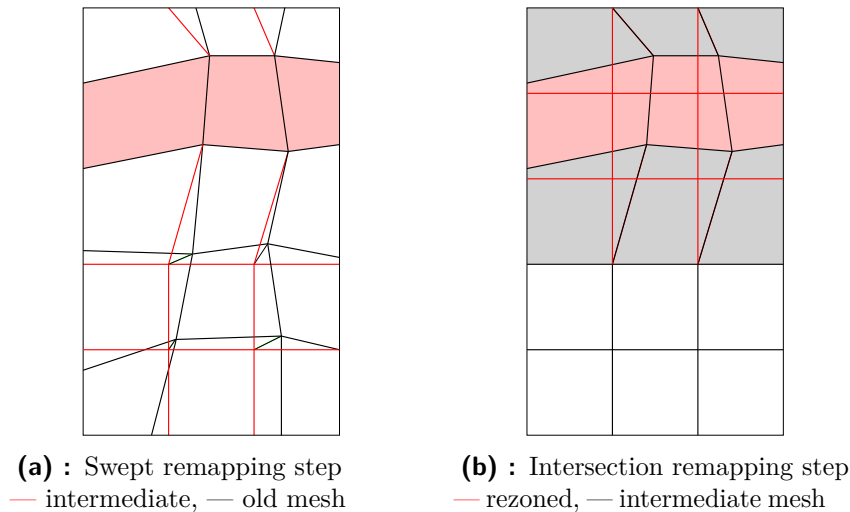


Figure 5.3: Mesh construction in two steps of our adaptive hybrid remapping method, ■ cells marked for intersections, ■ buffer cells.

There are several proposed types of switching functions in Appendix D [42], here we describe the one that has so far produced most convincing results. It requires the calculation of second derivatives of the measured quantity in the principal directions defined by the mesh movement:

$$\rho_{\mathbf{d}\mathbf{d}} = \frac{\mathbf{d} \mathbf{H}(\rho) \mathbf{d}^{\mathbf{T}}}{\mathbf{d}^2}, \rho_{\mathbf{p}\mathbf{p}} = \frac{\mathbf{p} \mathbf{H}(\rho) \mathbf{p}^{\mathbf{T}}}{\mathbf{p}^2}, \quad (5.15)$$

where \mathbf{d} is a unit vector in the direction of the movement of the computational cell, \mathbf{p} the perpendicular direction. It is determined from the centroid movement between the original and rezoned mesh. $\mathbf{H}(\rho)$ the Hessian matrix of second derivatives describing the local curvature of the function (calculated again by the least squares method as in Section 5.2.1). The swept-based method performs worse when the mesh moves in the direction perpendicular to the maximum curvature [43]. The proposed switching function marks a computational cell for intersection-based remapping if it fulfills the following criteria:

$$\exists n \in N(c) \quad \frac{2 \|\mathbf{x}_{\tilde{n}} - \mathbf{x}_n\|^2}{V_c} \left| \frac{\rho_{\mathbf{p}\mathbf{p}}}{\rho_{\mathbf{d}\mathbf{d}}} \right| > \alpha_{DDS}, \quad \alpha_{DDS} \sim 0.25 \quad (5.16)$$

where $\mathbf{x}_{\tilde{n}}, \mathbf{x}_n$ are the original and rezoned node locations. Using this switching function is shown to improve the solution symmetry with little to no effect on the computational cost.

5.3 Remapping of all fluid quantities (algorithm overview)

In Section 5.2 we have described the general principles of the remapping methods as well as the remapping of material volume, centroids and density. In this Section we will continue in the description of the specifics of remapping the remaining fluid quantities, with the only exception of pressure. Material pressure does not need to be remapped – it is calculated from the equation of state and the average cell pressure is computed using the averaging relation (3.13).

5.3.1 Specific internal energy

The approach to internal energy remap [57] is slightly different from density remapping. It is a mass-weighted quantity and the remapping formula is as follows:

$$\epsilon_{\tilde{c},k} m_{\tilde{c},k} = \epsilon_{c,k} m_{c,k} + \sum_{c' \in S(c,k)} F_{c',k}^e. \quad (5.17)$$

Remapped mass $m_{\tilde{c},k}$ is obtained from remapped density (5.4) and volume (5.1). The internal energy flux $F_{c',k}^e$ is also calculated using integration

of a reconstructed distribution over certain flux regions (depending on the method used):

$$F_{c,c',k}^E = \sum_{\text{flux dom.}} \int \rho_{\eta,k}(\mathbf{x}) \epsilon_{\eta,k}(\mathbf{x}) dV, \quad \eta \in [c, c'], \quad (5.18)$$

where the reconstruction of internal energy is approximated in the center of mass $\mathbf{x}_{c',k}^m$ to produce energy values compatible with the remapped mass [26]:

$$\epsilon_{c,k}(\mathbf{x}) = \epsilon_{c,k} + (\nabla \epsilon)_{c,k}(\mathbf{x} - \mathbf{x}_{c,k}^m), \quad \mathbf{x}_{c,k}^m = \frac{1}{m_{c,k}} \int_{(c,k)} \mathbf{x} \rho_{c,k}(\mathbf{x}) dV \quad (5.19)$$

The energy gradient is calculated with the least squares method (Section 5.2.1) and we can then substitute (5.19), (5.5) in the internal energy flux calculation (5.18) and yield integrals which require estimating additional second order terms that have the following form in general:

$$\int_{\text{flux reg.}} \mathbf{x} \otimes \mathbf{x} dV. \quad (5.20)$$

These terms can be precomputed along with the other geometrical integrals calculated over the flux regions.

5.3.2 Momentum remap on dual mesh

For nodal quantities we must use a different approach to remapping. As calculating fluxes by intersecting the dual mesh would be very computationally expensive, we instead opt for a different approach. We present here a modification of the optimization-based algorithm presented in [57, 78], modified to work on polygonal meshes. The motivation for using a more complex algorithm than for example [79] was mainly that such method does not guarantee that nodes surrounded by void cells have zero mass (which can lead to spurious velocities at material-void interfaces).

We start with the nodal mass remap formula [55]:

$$m_{\tilde{n}} = m_n + \sum_{n' \in S(n)} F_{n,n'}^m, \quad (5.21)$$

where $S(n)$ is a set of nodes connected with node n by an edge. This formula is used in an inverse way – we do not need to calculate remapped nodal masses as they can be determined from remapped cell masses using (3.3). Instead we use this relation to obtain inter-nodal mass fluxes $F_{n,n'}^m$ by optimization process. We split the inter-nodal fluxes in finer internal corner flux components:

$$F_{n,n'}^m = F_{n,n',c}^m - F_{n,n',c'}^m, \quad (5.22)$$

where the cells c, c' are cells that share the edge connecting n, n' . A general formula can be then applied on remapped corner mass:

$$m_{\tilde{c},\tilde{n}} = m_{c,n} + \sum_{n' \in S(n)} F_{n,n',c}^m + \sum_{c' \in C(n)} F_{c,n,c'}^m, \quad (5.23)$$

where $F_{c,n,c'}^m$ is the flux from cell c' to corner c, n (see Figure 5.4 for an example of a cell-to-corner flux configuration along with internal corner fluxes). The definition of this flux depends on the geometrical relationship of cells c, c' :

$$F_{c,n,c'}^m = \begin{cases} \sum_{k \in M(c) \cup M(c')} F_{c,c',k}^m & \text{if } c \text{ and } c' \text{ are corner neighbors only,} \\ \sum_{k \in M(c) \cup M(c')} \frac{1}{2} F_{c,c',k}^m & \text{if } c \text{ and } c' \text{ share a common edge.} \end{cases} \quad (5.24)$$

From the point of view of a computational cell c , we allow internal corner fluxes only between corners that share a common edge (as shown in Figure 5.4). Such description is sufficient to represent all possible flux configurations as fluxes between distant corners can be incorporated into several linked fluxes between directly neighboring corners. This formulation guarantees that the number of the unknown internal fluxes $F_{n,n',c}^m$ is the same as the number of corners in each computational cell and hence a linear system of equations (5.23) for all corners in a single cell has a unique solution which can be obtained using direct or iterative methods.

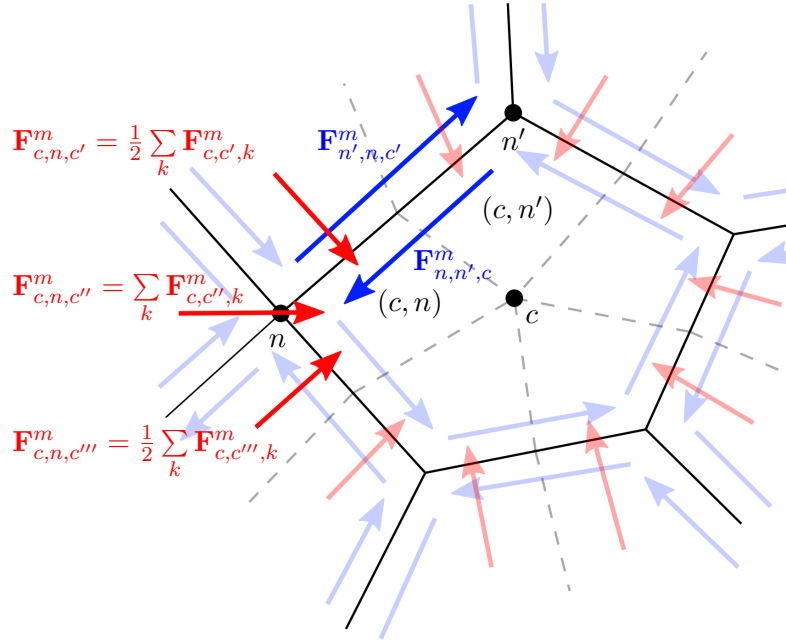


Figure 5.4: Different types of mass fluxes in cell c used to determine nodal mass flux between nodes n, n' . Colors indicate **cell-to-corner mass fluxes**, **internal corner mass fluxes**.

As mentioned at the beginning of this section, the inter-nodal fluxes are not used to remap nodal mass, instead they are used for remapping other related nodal quantities. This gives us the equation for momentum remap [57]:

$$\mathbf{u}_{\tilde{n}} m_{\tilde{n}} = \mathbf{u}_n m_n + \sum_{n' \in S(n)} \sum_{c \in C(n) \cap C(n')} \mathbf{u}_{n,n',c} F_{n,n',c}^m, \quad (5.25)$$

where $\mathbf{u}_{n,n',c}$ is the velocity reconstruction in the center of the internal corner transport volume. We use a first-order “donor-node” approximation here, which determines the reconstruction origin using the flux sign:

$$\mathbf{u}_{n,n',c} = \begin{cases} \mathbf{u}_{n'} & \text{if } F_{n,n',c}^m > 0 \\ \mathbf{u}_n & \text{if } F_{n,n',c}^m < 0 \end{cases} \quad (5.26)$$

This first order method produces non-oscillatory results and is easy to implement. However, in some cases it will hinder instabilities growth rate (such as in the case of Rayleigh-Taylor). Second order accurate methods are useful in these situations, using bilinear interpolation of the velocity field in the computational cell and the flux-corrected remap (FCR) paradigm to avoid violation of bounds [57].

5.3.3 Kinetic energy fix

So far the remapping algorithm guarantees conservation of internal energy and momentum. The conservation of total energy is, however, not enforced by the previous steps, because the kinetic energy is quadratic in velocity. This is the reason for performing the commonly used kinetic energy fix [10]. We begin with remapping the nodal kinetic energy T_n :

$$T_{\tilde{n}} = T_n + \sum_{n' \in S(n)} F_{n,n'}^T, \quad (5.27)$$

where the inter-nodal kinetic energy fluxes are defined as follows:

$$F_{n,n'}^T = \sum_{c \in C(n) \cap C(n')} \frac{1}{2} \mathbf{u}_{n,n',c}^2 F_{n,n',c}^m. \quad (5.28)$$

We then calculate the cell-based discrepancy between the remapped (conserved) kinetic energy and the kinetic energy calculated from remapped velocity:

$$\Delta e_{\tilde{c}} = \sum_{\tilde{n} \in N(\tilde{c})} \frac{m_{\tilde{c},\tilde{n}}}{m_{\tilde{n}}} \left(T_{\tilde{n}} - \frac{m_{\tilde{n}} \mathbf{u}_{\tilde{n}}^2}{2} \right), \quad (5.29)$$

where the kinetic energy is distributed from nodes to cells by the ratio of the remapped corner mass $m_{\tilde{c},\tilde{n}}$ to the remapped nodal mass $m_{\tilde{n}}$. The discrepancy $\Delta e_{\tilde{c}}$ then must be distributed to all materials present in cell c by mass-fraction weighing:

$$m_{\tilde{c},k} \varepsilon_{\tilde{c},k}^{\text{corrected}} = m_{\tilde{c},k} \varepsilon_{\tilde{c},k} + \frac{m_{\tilde{c},k}}{m_{\tilde{c}}} \Delta e_{\tilde{c}}. \quad (5.30)$$

This correction must be done with respect to the physical bounds (void material is not affected and internal energy positivity must be preserved). If some of the discrepancy cannot be distributed we can either choose to allow small energy conservation violation or redistribute this amount further [91].

5.3.4 Elastic energy-preserving deviatoric stress remap

The last quantity which must be remapped onto the rezoned mesh is the deviatoric stress tensor. There are methods for bound-preserving reconstruction of the deviatoric stress [84] or general tensor quantities [62]. However, these do not guarantee the preservation of the total elastic energy (integral of the J_2 invariant). Therefore, we have designed a remapping scheme ourselves (see Appendix E [45]).

The Cauchy stress or its deviatoric part are not conservative quantities, unlike the J_2 invariant which is a quadratic function of the stress components and is related to the elastic energy density:

$$\epsilon_{c,k}^{elast.} = \frac{(J_2)_{c,k}}{2\mu_k}, \quad (J_2)_{c,k} = \frac{1}{2} \|S_{c,k}\|^2 = \frac{1}{2} \text{tr} \left(S_{c,k}^T S_{c,k} \right). \quad (5.31)$$

The remapping of J_2 can be done independently of S [44] in flux form with second order reconstruction and a limiter for bounds and positivity preservation:

$$(J_2)_{\tilde{c},k} V_{\tilde{c},k} = (J_2)_{c,k} V_{c,k} + \sum_{c' \in S(c,k)} F_{c,c',k}^{J_2}. \quad (5.32)$$

The J_2 invariant uniquely determines the stress intensity but not the directional parameters. For this reasons we remap the stress components in the following way:

$$S_{\tilde{c},k} V_{\tilde{c},k} = S_{c,k} V_{c,k} + \sum_{c' \in S(c,k)} F_{c,c',k}^S, \quad (5.33)$$

where the “stress flux” is a first-order approximation as this remap is used only to determine the principal directions of the resulting stress tensor. We then scale $S_{\tilde{c},k}$ to adhere to the remapped J_2 invariant in the following way (Appendix E [45]):

$$S_{\tilde{c},k}^{\text{scaled}} = S_{\tilde{c},k} \sqrt{\xi_{c,k} + 2(1 - \xi_{c,k}) \frac{(J_2)_{\tilde{c},k}}{\|S_{\tilde{c},k}\|^2}}, \quad 0 \leq \xi_{c,k} \leq 1 \quad (5.34)$$

The parameter $\xi_{c,k}$ is used to prevent distortion of the principal directions in cases when the re-scaling is inaccurate, especially around tensor critical points. If in any of the neighboring cells the eigendirection system is close to opposite to the one in cell c then we switch to the component-based remap. The proposed formula for the parameter is as follows:

$$\xi_{c,k} = \begin{cases} \frac{\cos(\frac{\pi(1+\cos\phi)}{2})+1}{2} & \text{for } \cos\phi < 1-w, \\ 0 & \text{otherwise} \end{cases}, \quad (5.35)$$

where the parameter w defines the sensitivity of the transition and in our experience can be set to $w = 1$, therefore the relaxation starts if the angle formed by the principal direction systems $\phi > \pi/2$. This angle is determined from the stress tensor directly:

$$\cos \phi = \min_{c' \in \mathcal{S}(c,k)} \frac{\text{tr} \left(S_{c',k}^T S_{c,k} \right)}{\|S_{c,k}\| \|S_{c',k}\|}. \quad (5.36)$$

The relaxation parameter is used only when necessary (unphysical deformation patterns in symmetrical stress distributions) as it causes a small deviation from the exact conservation of the elastic energy in the remap process.

Section 6

Examples of numerical calculations

We present here several examples of the utilization of the ALE algorithm described in the previous Sections. The selected two-dimensional numerical calculation results are intended to demonstrate the functionality of all parts of the computation and show the possible spectrum of problems where this algorithm can be applied. The first three computations include fluid materials only while the remaining ones are focused on solid dynamics. The main purpose is to demonstrate the diversity of problems that can be simulated with the same algorithm while changing only material and simulation parameters. For rigorous testing and the full justification of the described algorithm components, along with other possible applications of the method, we would like to refer the interested reader to the published results included as the Appendices of this dissertation.

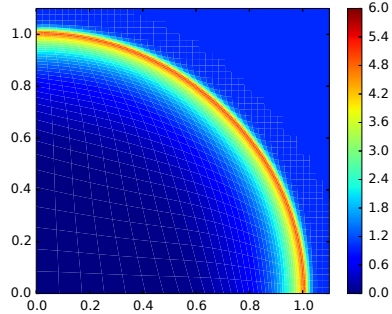
In the first three tests, arbitrary units are used, which is in accordance with the problem specification and the possible convention in the field of numerical algorithm design.

6.1 Sedov problem – point explosion

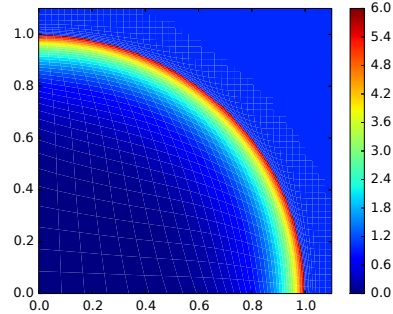
This numerical test is a simulation of a point explosion in a static cold ideal gas environment, initiated by a large internal energy value in the lower left computational cell which drives the shock expansion. The initial conditions [88] are as follows:

$$\varepsilon_{0,0} = 409.7, \quad \varepsilon_{env.} = 10^{-4}, \quad \gamma = 1.4, \quad \rho_{env.} = 1.0.$$

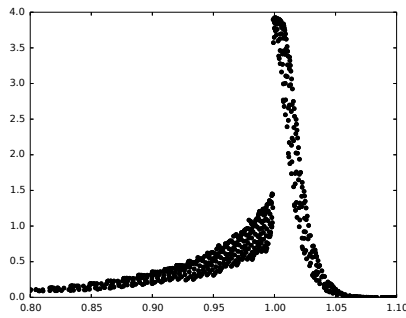
This test involves the usage of the fluid model only (material strength is not involved) without multi-material interactions. Remapping is required to preserve convexity of all computational cells, the mesh is smoothed by the Winslow method every 10 time steps. The used initially square computational mesh with a $(0, 1.1)^2$ domain and 45×45 computational cells does not conform with the expected radial symmetry of the solution. This is selected on purpose to analyze the robustness and accuracy of the 2D computation and identify any violations of symmetry caused by mesh imprinting effects. It was used this way in Appendix C, D [43, 42] to compare the flux calculation methods used in the remap step.



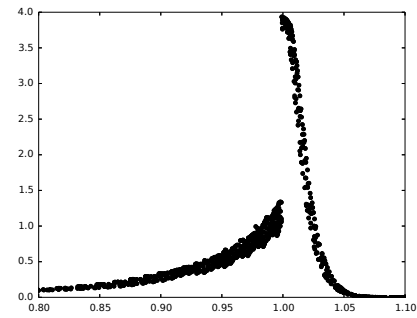
(a) : Density, numerical result with swept-based remapping



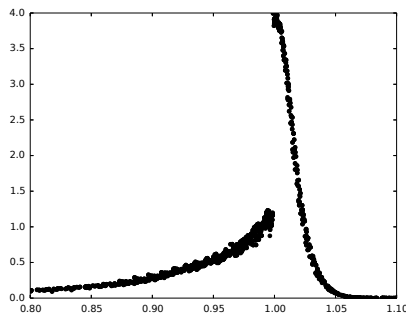
(b) : Density, analytical solution sampled on the numerical mesh



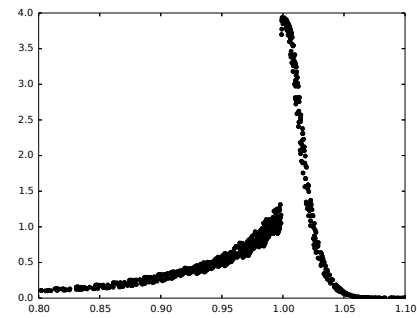
(c) : Radial density error, swept-based fluxes



(d) : Radial density error, intersection fluxes



(e) : Radial density error, combined remapping, switched using curvature in mesh movement direction



(f) : Radial density error, combined remapping, switched using curvature in cell diagonal direction

Figure 6.1: Numerical simulation of the fluid density distribution in the 2D Sedov problem. Initially square 45×45 computational mesh, ALE computation with Winslow rezoning every 10 time steps, $T = 1.0$.

The resulting density distribution at the simulation time $T = 1.0$ is shown in Figure 6.1. The exact solution of this problem is known [89] so we can calculate also the cell-centered absolute density error. The spread of this error (i.e. the radial symmetry of the solution) for swept-based, intersection-based and combined methods for flux region construction in the remap step are shown. The intersection-based and combined remapping methods clearly offer better symmetry properties, the latter being much less computationally intensive. If comparing the same problem on a finer 90×90 mesh, the flux computation takes approximately 16.9s with the intersection-based method, 2.5s with the swept-based method and 3.1s with both combined methods (a single-thread computation on a common server processor [42]). This shows that the combined method provides solution symmetry comparable with the intersection-based method at a much lower computational cost.

6.2 Triple point problem

This problem represents an interaction of three materials in two dimensions. This three-state Riemann problem was used for comparison in e.g. [32]. The initial conditions are defined in the following way – there are three ideal gas materials at rest. On the left in the $(0, 1) \times (0, 3)$ region the following properties are used:

$$\gamma_1 = 1.5, \quad \rho_1 = 1, \quad p_1 = 1.$$

In the $(1, 7) \times (0, 1.5)$ region the parameters are as follows:

$$\gamma_2 = 1.4, \quad \rho_2 = 1, \quad p_2 = 0.1.$$

And at last a lighter gas in the $(1, 7) \times (1.5, 3)$ region is defined with the following properties:

$$\gamma_3 = 1.5, \quad \rho_3 = 0.125, \quad p_3 = 0.1.$$

The initial conditions result in a shock that moves to the right and its propagation speed is higher in the upper right part, resulting in a shear movement and a development of a vortical pattern. We use this test to further demonstrate the viability of our multi-material model in a multi-fluid scenario. Also, the ability of the algorithm to process shear and vortical movements is evaluated.

The results are shown in Figure 6.2. An initially rectangular 70×30 grid was used and the computation was performed in the ALE mode with Winslow rezoning every 10 time steps. The material composition is shown at the initial time level and then material density values are displayed for each time snapshot. The interface of the materials can be seen as a discontinuity in the density values. The position of the contact of the gas on the left with both low and high density gas corresponds well with [32], although the vortex development is less pronounced, owing to the coarser computational mesh used in our example. The utilization of the Moment-of-fluid material reconstruction at the end of the multi-material model algorithmic step is responsible for the sharp interface resolution near the tip of the vortex.

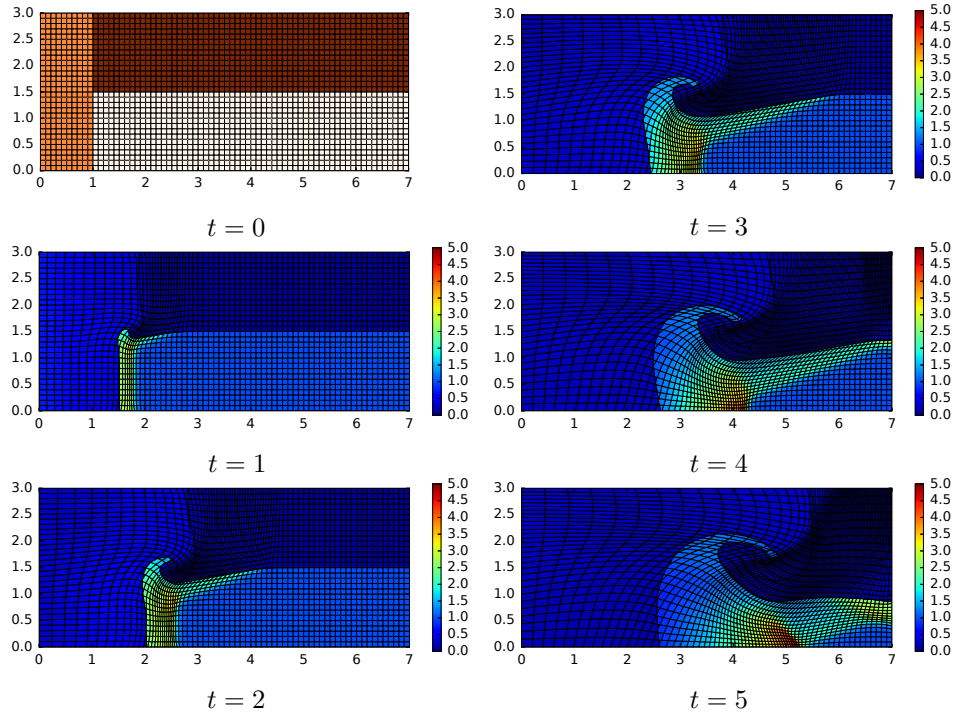


Figure 6.2: The triple point problem, evolution of material density, initial material distribution shown at $t = 0$. ALE computation with Winslow rezoning every 10 time steps.

6.3 Shock-induced cavity collapse

This problem involves a collapse of a void cavity due to a passing shock wave. It is modeled using the fluid model with multiple materials interacting and void closure. It was used in Appendix A [5] to demonstrate the ability to simulate the collapse of a vacuum filled cavity with a complete disappearance of the void material.

The domain is $(-1.5, 1.5) \times (-0.5, 0.5)$, its boundaries are considered as solid walls. There is a moving ideal gas “piston” in the area bounded by $x < -0.5$ with the following initial conditions:

$$\rho_1 = 4, \quad p_1 = 1.3334, \quad \mathbf{u}_1 = (1, 0), \quad \gamma_1 = 1.4,$$

while in the center of the domain a void cavity is initialized with a radius of $r = 0.3$. The rest of the domain is filled with an ideal gas with the following properties:

$$\rho_2 = 1, \quad p_2 = 1 \cdot 10^{-5}, \quad \mathbf{u}_2 = (0, 0), \quad \gamma_2 = 1.4.$$

The simulation is performed with mesh rezoning to the initial mesh with 120×40 square cells on each time step.

This configuration creates a planar shock wave with the initial front location at $x = -0.5$ and moving to the right. Its progression is shown in Figure 6.3

where the passing shock closes the void completely before time $t = 0.5$. We can see that the interface between the moving gas and void moves faster than the shock front. The shape of the shock is substantially deformed by the collapsing bubble. Although the computational mesh is coarse, the interface resolution is preserved well due to its sub-mesh representation.

6.4 Elastic vibrating plate

This testing problem involves a beryllium plate subject to prescribed velocity initiating an oscillatory movement [85, 30]. It is the first problem in this Section that uses a solid material model, using the Mie-Gruneisen equation of state and linear elastic material strength model. The material parameters of beryllium are the following:

$$\begin{aligned} \rho_0 &= 1.845 \text{ g cm}^{-3}, \quad \mu = 1.519 \text{ Mbar}, \quad Y_0 = 1 \text{ Mbar}, \\ c_0 &= 799.8 \text{ m s}^{-1}, \quad \gamma = 1.16, \quad S_\alpha = 1.124, \end{aligned} \quad (6.1)$$

where the yield strength Y_0 of the material is artificially increased, keeping the oscillations in the purely elastic regime. The domain is $(-3, 3) \times (-0.5, 0.5)$ cm and is paved by 100×16 rectangular cells.

The material is unloaded at the beginning and there is an initial vertical velocity field defined by:

$$\begin{aligned} u_y &= A\omega (g_1 (\sinh \Omega (x + 0.03) + \sin \Omega (x + 0.03)) \\ &\quad - g_2 (\cosh \Omega (x + 0.03) + \cos \Omega (x + 0.03))), \end{aligned}$$

where Ω is the first non-zero root of the equation $\cosh(0.06\Omega) \cos(0.06\Omega) = 1$ and the remaining coefficients are defined as follows:

$$\begin{aligned} g_1 &= \cosh(0.06\Omega) - \cos(0.06\Omega), \\ g_2 &= \sinh(0.06\Omega) - \sin(0.06\Omega), \\ A &= \frac{y_{\max}}{2g_2}, \quad y_{\max} = 0.5 \text{ cm}, \\ \omega &= \Omega^2 \Delta y \sqrt{\frac{E}{12\rho(1-\nu^2)}}, \quad \Delta y = 1 \text{ cm} \end{aligned}$$

where E is the elastic Young's modulus and ν is the Poisson's ratio, quantities that are tied to the bulk and shear modulus with the following relations:

$$E = \frac{9K\mu}{3K + \mu}, \quad \nu = \frac{3K - 2\mu}{2(3K + \mu)},$$

where the bulk modulus K is a state function described in (3.26). The problem is simulated using a pure Lagrangian mode (no mesh rezoning is performed). The results are shown in Figure 6.4, where the time evolution of the density field is displayed, showing compression in the convex part

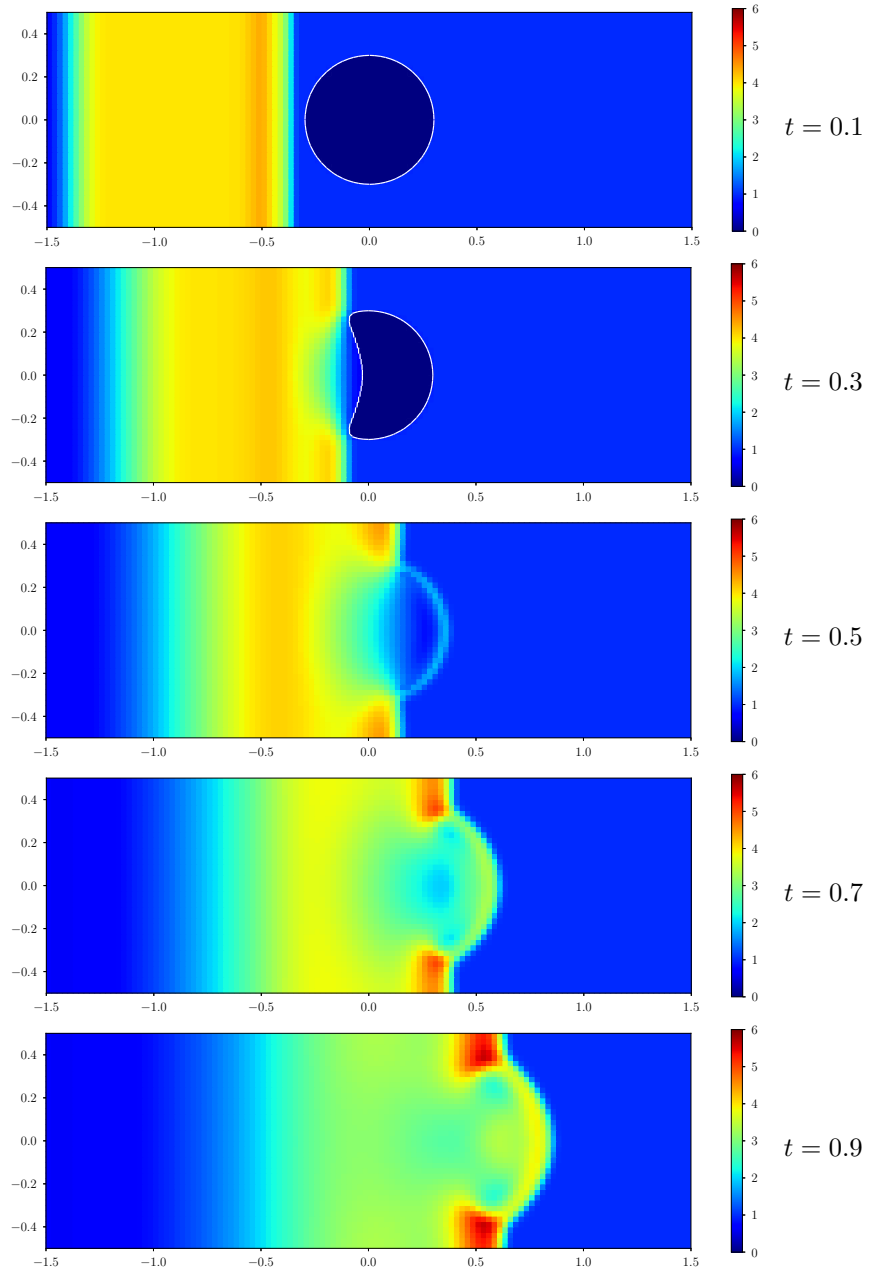


Figure 6.3: Shock wave interacting with a spherical void cavity, density profile, 120×40 . In the top two pictures the boundary of the void bubble is visible, while in the remaining snapshots the bubble is completely closed.

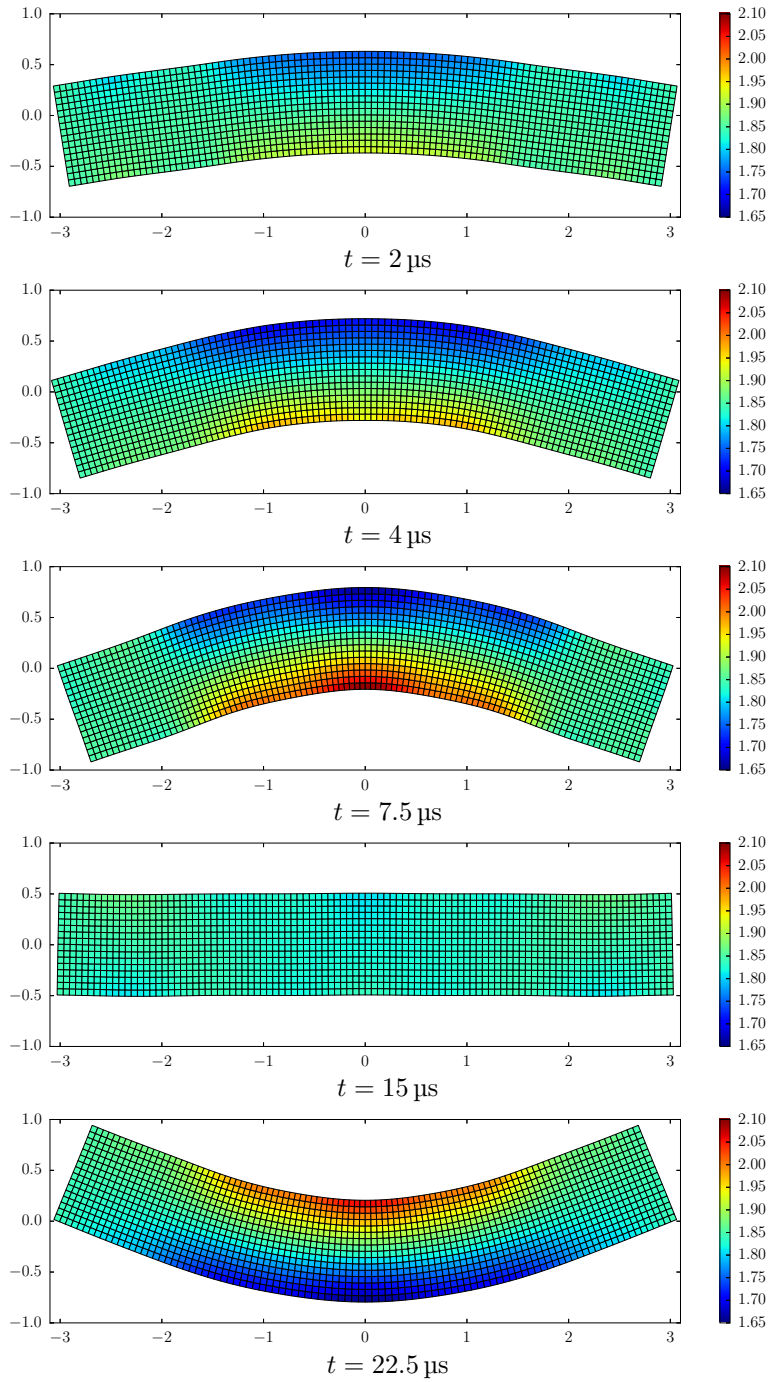
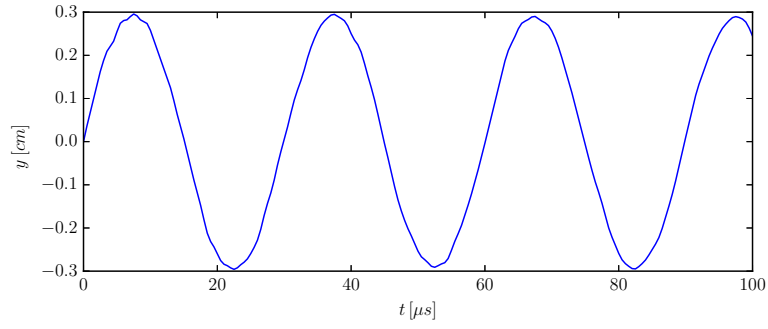
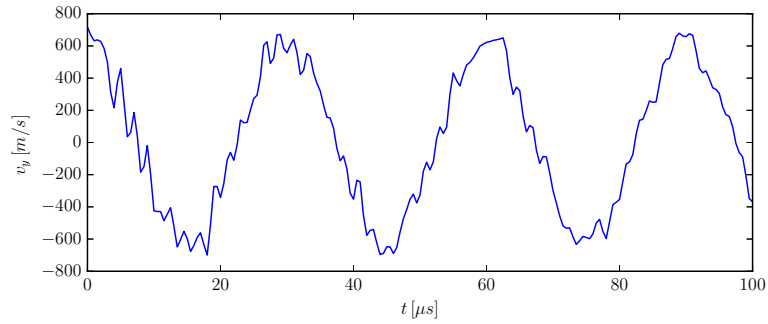


Figure 6.4: Elastic oscillation of a beryllium plate, density distribution. Lagrangian computation with free boundary condition.



(a) : Y coordinate of the plate center



(b) : Vertical velocity of the plate center

Figure 6.5: Elastic oscillation of a beryllium plate, time evolution. Lagrangian computation with free boundary condition.

of the bent plate and expansion in the concave part, corresponding to the expected behavior. In Figure 6.5 we show the evolution of the position and velocity of the plate center in time. The wave profile and the approximate period of $T \sim 30 \mu\text{s}$ corresponds well to the results published in [30], which were obtained using an entirely different (cell-centered) numerical scheme. The higher-frequency oscillatory modes visible in the velocity plot are in accordance with the published results as well [85]. These results demonstrate that our numerical scheme, in combination with the constitutive model for solids, produces a satisfactory representation of elastic deformation.

6.5 High speed aluminium plate impact

In this numerical example we calculate the result of an impact of a fast moving and a static aluminium plate. The Mie-Gruneisen equation of state and the described strength model is used for the modeling of the aluminium material. The plates are treated by the multi-material algorithm as distinct logical materials (to make use of the interface representation and contact mechanics) although they share common material properties:

$$\begin{aligned} \rho_0 &= 2.785 \text{ g cm}^{-3}, \quad \mu = 276 \text{ kbar}, \quad Y_0 = 3 \text{ kbar}, \\ c_0 &= 5328 \text{ m s}^{-1}, \quad \gamma = 2, \quad S_\alpha = 1.338. \end{aligned} \quad (6.2)$$

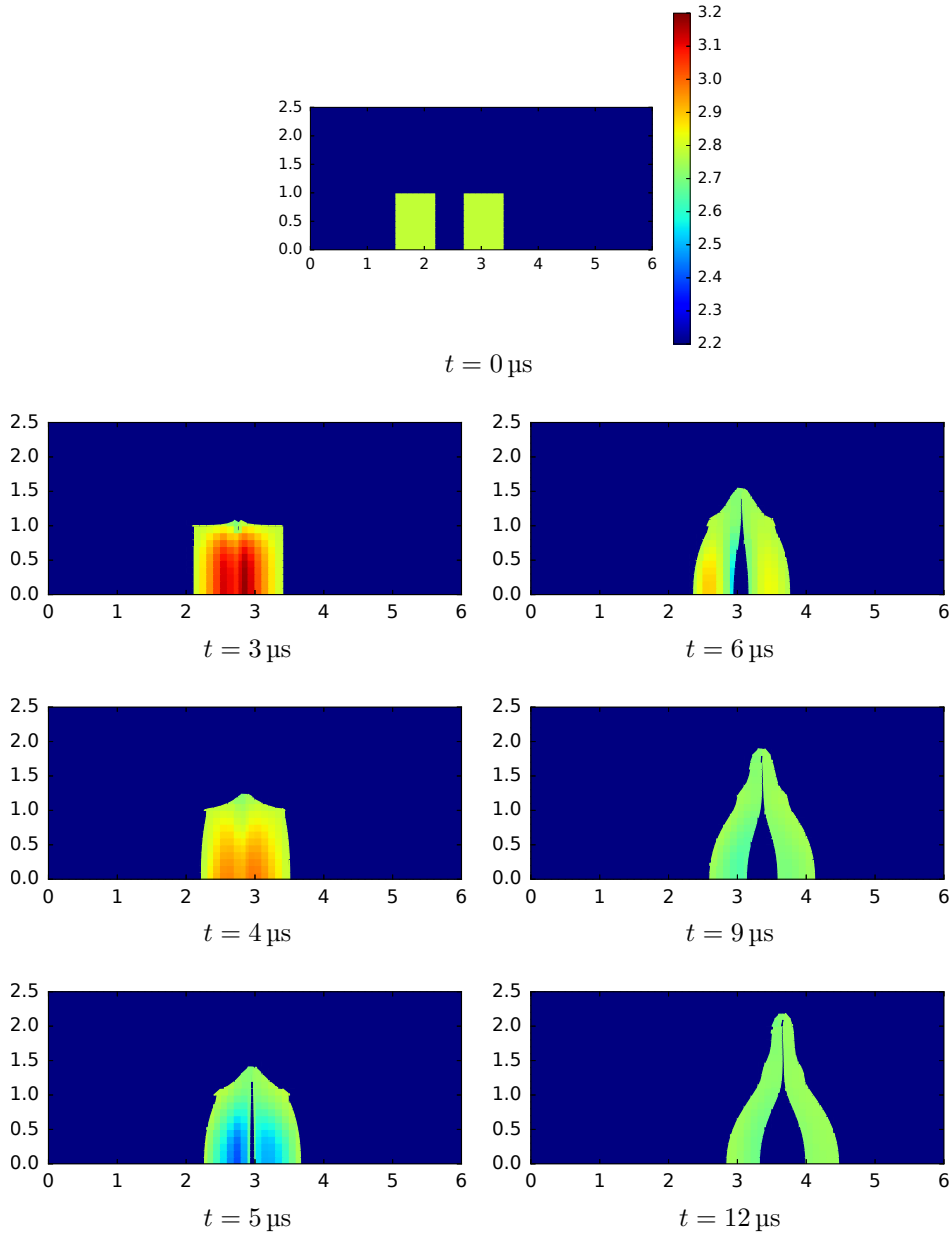


Figure 6.6: A moving aluminium plate collides with a static aluminium plate at $u_{init.} = 2 \text{ km s}^{-1}$. Density distribution at various time moments. ALE computation with initial mesh rezoning.

The simulation is run with rezoning to the initial mesh on each ALE step, so in this test the stress tensor remap is frequently used. The domain is $(0, 6) \times (0, 2.5)$ cm with 60×25 computational cells. The initial location of the left plate is between $(1.5, 2.2) \times (0, 1)$ cm and its initial horizontal velocity is $u_{init.} = 2 \text{ km s}^{-1}$. The second plate is static and is located at $(2.7, 3.4) \times (0, 1)$ cm. The rest of the computational domain is filled with the void material which represents the ambient vacuum environment.

The resulting density evolution is displayed in Figure 6.6 for several time moments. The high speed impact results in a very fast plastic loading of both plates resulting in a significant deformation. First the plates are compressed ($t = 3 \mu\text{s}$) after the void closes on plate contact. Then we can see void opening ($t = 5 \mu\text{s}$) as a result of material rebound and negative stress formation. After the opening a lower density is visible indicating that the material is quickly expanding. In the later time levels, the opening progresses in the center of the plate, while the outward edges of the plates undergo mostly perpendicular deformation. This shows that our model of solid continuum with void closure and opening can be used to simulate two-body impact problems with predominantly plastic deformation.

6.6 Low speed steel ball impact

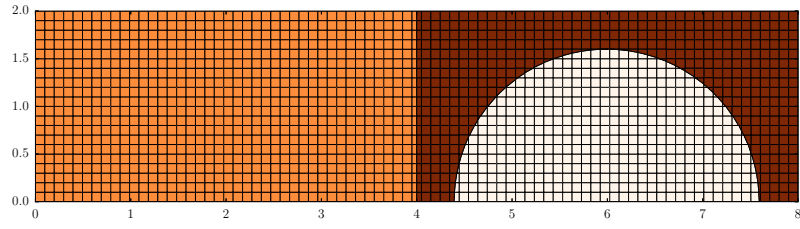
The last presented simulation (from Appendix B [41]) involves an Cartesian approximation of the “bouncing ball” numerical experiment where a steel ball is launched against a wall from the same material at a speed inducing an elastic collision ($u_{init.} = 80 \text{ m s}^{-1}$). The domain is $(0, 8) \times (0, 2)$ cm, the steel wall lies in $x \in (0, 4)$ cm and the steel ball is centered at $(6, 0)$ cm with a diameter of 3.2 cm. The material properties of the used steel alloy are the following:

$$\begin{aligned} \rho_0 &= 7.905 \text{ g cm}^{-3}, \quad \mu = 547 \text{ kbar}, \quad Y_0 = 70 \text{ kbar}, \\ c_0 &= 4570 \text{ m s}^{-1}, \quad \gamma = 2, \quad S_\alpha = 1.49. \end{aligned} \quad (6.3)$$

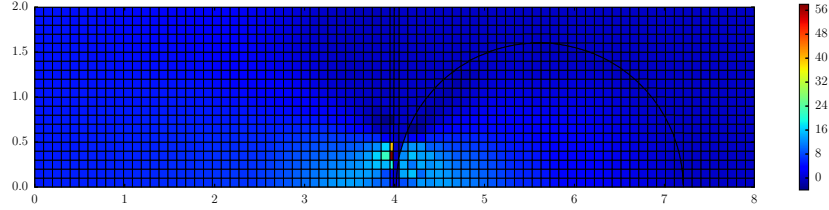
The rest of the domain is filled with void and the boundaries are fixed walls. The computation is performed in the ALE regime with rezoning to the initial square mesh on each time step. A rectangular grid of 81×20 cells is used, the irregular number of cells is to initialize the problem so that the material interface of the steel wall and void does not overlap with the cell edges. This way the multi-material cells are already present from the beginning of the simulation.

The results are shown in Figure 6.7 where we can see the initial material distribution and the normal stress in the horizontal direction at the approximate impact and rebound time moments. We compare the performance of the equal compressibility bulk stage (as in [5]) with the current version of the model as described in this dissertation. At the time of impact, there is already premature interaction (stress) leading to energy dissipation for the equal compressibility model. At later time when the ball should separate

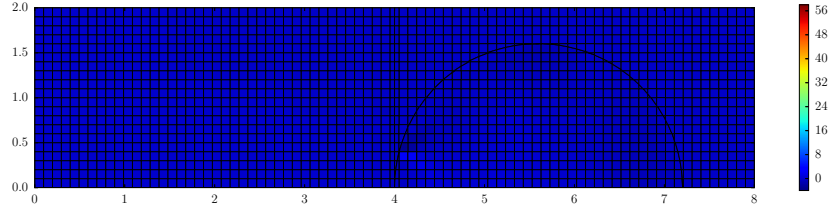
from the target due to rebound, the stress is too high in the contact zone, the separation is delayed and the rebound velocity is extremely low so that the ball effectively stops and all kinetic energy is dissipated. The modified model helps mitigate these shortcomings leading to a much lower kinetic energy dissipation, allowing a more realistic rebound simulation. This difficult test demonstrates the abilities of our model to deal with a situation where the actual deformation is very small and the impact/rebound timing is critical to obtain realistic results.



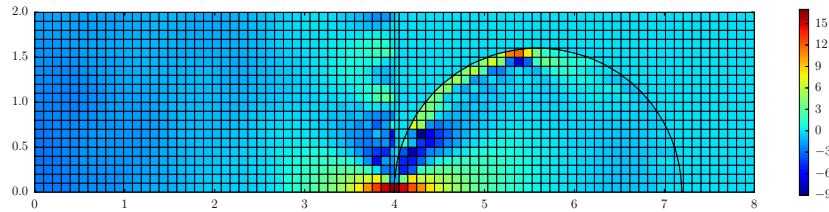
(a) : Material distribution at $t = 0 \mu\text{s}$



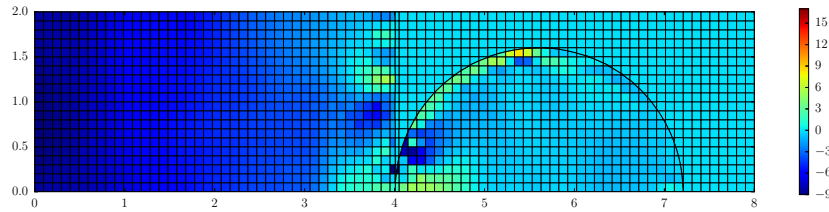
(b) : Normal stress in x-direction σ_x [kbar], $t = 50 \mu\text{s}$ (at the time of impact), IASSD with only equal compressibility bulk stage



(c) : Normal stress in x-direction σ_x [kbar], $t = 50 \mu\text{s}$ (at the time of impact), IASSD, full model



(d) : Normal stress in x-direction σ_x [kbar], $t = 87 \mu\text{s}$ (ideal rebound time), IASSD with only equal compressibility bulk stage



(e) : Normal stress in x-direction σ_x [kbar], $t = 87 \mu\text{s}$ (ideal rebound time), IASSD, full model

Figure 6.7: Deflection of a moving steel ball from a stationary steel wall at low speed ($u_{init.} = 80 \text{ m s}^{-1}$). Comparing the normal stress distributions for the original IASSD model (equal compressibility bulk stage) with the more complex current version.

Section 7

Summary of selected publications

This section provides a commentary on each of the papers published in peer-reviewed journals by the author of this dissertation. The contribution of the author to the research and publication process is described as well as the significance of the papers relative to the overall topic of the dissertation. The papers are attached to this dissertation as appendices A - E.

7.1 Constrained optimization framework for interface-aware sub-scale dynamics models for voids closure in Lagrangian hydrodynamics (Appendix A)

In this paper we have focused mainly on the concept of void closure within the interface-aware sub-scale dynamics multi-material cell model. Voids can represent internal (cavities, gaps) and external (ambient vacuum environment) free boundaries in a Lagrangian simulation. Especially in the second case, an artificial light gas environment has been often used, which can negatively affect the robustness of the simulation as the gas can reach spurious states if trapped between two colliding masses. Also the contact physics cannot be accurately represented in the simulation if the colliding materials are separated by a thin layer of gas. We have proposed an integration of the void material in the already existing sub-scale dynamics model for treating cells with multiple materials.

The novel formulation of the model includes the concept of void closure – void does not resist compression and is bounded only by the volume positivity requirement. If the volume is decreased below certain (relative) threshold it is removed from the computational cell altogether. This allows us to model the transition between movement in vacuum and contact physics. A modification of the original formulation is also needed for other constraints to treat the sub-scale interactions in vacuum expansion and related energy dissipation issues.

On one-dimensional examples we have shown that such formulation is viable for shock propagation, gap closure and expansion in vacuum. Then, several two-dimensional problems are presented – a projectile impact tests the void closure mechanism and a comparison with light gas environment is made.

A closure of a cavity by a shock wave and a point explosion driven shaped charge simulations are among the intended application of this algorithm.

This paper represents the work mostly done in collaboration with Mikhail Shashkov during an internship at the Los Alamos National Laboratory in New Mexico. The author of this dissertation has created the numerical code for this method, participated in the creation of the paper manuscript and has been responsible for most of the visualizations presented in the paper.

7.2 An interface-aware sub-scale dynamics multi-material cell model for solids with void closure and opening at all speeds (Appendix B)

The research of the interface-aware sub-scale dynamics multi-material model has continued further and we have incorporated several new capabilities that are described in this paper. First, the modification of this model to allow solid materials interaction simulation was presented. It required the transition from pressure equilibration to normal stress equilibration along with a more complex constraints for sub-scale interactions. In combination with the void framework, this also brings the need of simulating *void opening*, a process in which the tension in materials can create a new void gap in the location of the interface. This is necessary for simulating physically correct impact problems where rebound can occur.

Another important topic is the adaptation of the model to low-speed impact problems. In high speed interactions, timing issues due to slow movement on the scale of computational cells were not observed. In low-speed interactions (mostly elastic collisions) this can lead to a premature energy dissipation and a stopping of the object that was intended to reflect. To counter this problem we have proposed a modification to the bulk stage of the model which will take into account the compressibility of individual materials and free movement in void, preventing most unwanted interactions before the materials actually come into contact.

The numerical tests have shown that our model is able to work with gases, solids and voids well. The more complex problems were focused mostly on the problem of void opening after impacts at various speed with symmetric initial conditions (planar or cylindrical symmetry on Cartesian meshes) as well as fully two-dimensional tests.

This research was done as a continuation of the collaboration with Andrew Barlow and Mikhail Shashkov, the author of this dissertation has developed the concept of the modified bulk stage in the multi-material model, written the manuscript of the paper, created the numerical code and visualizations presented in this paper. The author of this dissertation is the first and corresponding author of the paper.

7.3 Local error analysis and comparison of the swept- and intersection-based remapping methods (Appendix C)

There are two widely used approaches to construction of the flux integrals in the remapping algorithm. The more straightforward intersection-based remapping calculates fluxes by intersecting original and rezoned mesh cells. The approximation based on regions swept by the cell edge offers improvements in computational efficiency, as it does not require polygon intersection to determine the flux region. Both methods offer second order accuracy in general, but it has been observed [60] that the profiles of the remapped quantities differ slightly for each method. Our goal was to estimate these differences and offer a key to deciding which of the methods is appropriate in certain situations.

In this paper we first perform an analysis of the local distribution of the remapping error for several typical computational mesh movement patterns. The analysis is performed on a general two-dimensional second degree Taylor polynomial as the initial quantity distribution. The flux integrals are evaluated analytically for several movement patterns – namely orthogonal cell corner expansion, corner node movement, cell shear and rotation. In case of the corner movement, it was found that the local error is dependent on the principal curvature of the initial quantity distribution – if the curvature was larger in the direction of movement, the swept-based method could produce lower local error. If the perpendicular component is more prominent, the intersection-based method is locally more accurate. For cell shear and rotation, the intersection-based method was found to be generally more favorable.

The results of the simplified analytical models were compared with results of numerical simulations – from several static remapping tests up to a full ALE computation. It was found that in more complex situations the total error in the whole domain is usually comparable for both methods, but the swept-based remapping can imprint the mesh movement in the error distribution, leading to violations of planar or radial symmetry.

This paper was an extension and publication of the work already conceptualized in the author’s Master’s thesis [40]. The author of this dissertation was responsible for carrying out the analytical model derivations, numerical coding, visualizing the numerical test results and writing of a significant part of the paper itself. The author of this dissertation is the first author of the paper.

7.4 Combined swept region and intersection-based single-material remapping method (Appendix D)

In the previous paper comparing the error distribution of the swept- and intersection-based remapping methods (Appendix C [43]) we have confirmed

that although the overall accuracy is similar, both methods can result in different error profile locally. Knowing this, we have asked the following question: using the analytical local error estimates, would it be possible to optimize the resulting error profile by switching between both remapping methods on a cell-by-cell basis? Such method could result in superior symmetry while at the same time lowering computational cost by using the cheaper swept-based method where possible.

This paper proposes a combined remapping method, which uses a switching function to determine which method will be used for flux calculation. First we have tried simple scalar criteria such as approximate local slope or curvature of the remapped quantity distribution with limited success. From the analysis in [43] we have known that the local error is dependent on directional factors. Therefore we have developed more advanced switching functions that evaluate the second derivative in the direction of the movement of mesh cells or in the direction of cell diagonal. These switches were shown effective in numerical tests utilizing cyclic remapping (where the mesh returns to its initial state after certain number of remapping steps and the resulting and initial quantity distributions can be compared) and ALE computations as well.

The benefits of the method are shown on the results of cyclic remapping tests and the full ALE algorithm is employed in simulating the 2D Sedov problem on Cartesian mesh (Figure 6.1). This indicates that we have succeeded in creating a method with lower symmetry distortion and computational cost at the same time.

The author of this dissertation has created the paper manuscript and was responsible for carrying out the numerical coding and visualizing the numerical results. The author of this dissertation is the first and corresponding author of the paper.

7.5 Second-invariant-preserving remap of the 2D deviatoric stress tensor in ALE methods (Appendix E)

To be able to run simulations with elastic-plastic solids, a framework to remap the stress tensor is needed. In this paper we analyze several possible approaches along with limiter schemes. It is shown that remapping tensor components can produce poor numerical performance at discontinuities in the stress profile (which can occur in elastic shock precursors etc.) – the remapped stress energy profile does not fulfill the local minimum preservation requirement at the discontinuity and falls to zero. We have tried several limiter configurations either inspired by vector field limiting methods or a special tensor-oriented approaches that can be successful in preventing maximum overshoots but not for minima. Also they do not preserve the total deformation energy.

As a solution we have proposed a method in which the second invariant of the stress tensor is remapped independently of the tensor components and the

remapped tensor is then scaled to conform to the remapped invariant. This method preserves the total deformation energy and can guarantee bounds preservation for the stress magnitude. As it can distort the stress principal direction in the vicinity of tensor critical points, it is modified with a relaxation term which switches to low-order component remapping in the surroundings of such points.

This method was tested on a tensor version of the rotating object advection problem (in which there is no solid mechanics, just tensor quantity remapping) that showed our method preserves both the minimum and maximum of the J_2 invariant. An application was then shown with an ALE simulation of a strong shock in an aluminium cylinder, where second invariant remap was demonstrated to avoid some artifacts (local dips in the stress value) that were present when using component-based remapping and the stress magnitude behind the shock was kept almost constant at the plastic threshold.

This paper extended the work presented at the HYP 2016 conference in Aachen, Germany [44] and the author of this dissertation has partly developed the concept of the invariant remapping, written the paper manuscript, created the numerical code and visualized the results. The author of this dissertation is the first and corresponding author of the paper.



Section 8

Conclusion

In this doctoral thesis, we have presented the improvements of the indirect ALE numerical algorithm in fluid and solid dynamics. It was constructed and implemented as a single computer code, including all the novel features presented in this dissertation. It consists of a Lagrangian stage with compatible discretization of fluid quantities on a staggered grid, and a rezone and remap stage for mesh smoothing and conservative interpolation of the involved quantities. The algorithm as a whole is described in this dissertation, while the individual algorithmic improvements are thoroughly described in the papers published in peer-reviewed journals (included as Appendix A–E).

All parts of the algorithm are designed to support multi-material computations with non-diffuse interfaces. The Lagrangian stage includes the Interface-aware sub-scale dynamics multi-material model to process the interaction of distinct materials in a single computational cell. This model was improved to allow interactions of both fluids and elastic-plastic solids. We are also able to include voids in the simulation, representing vacuum-filled regions that do not resist compression and expansion and can emerge or disappear from the computational cell at certain conditions (void closure/void opening). Voids can also act as an ambient environment, allowing for representing impact events as a multi-material problem in a simple domain rather than using multiple meshes and complicated boundary conditions.

In the remapping part an analysis has been made on how the choice of the flux calculation methods (intersection- or swept- based) affects the solution symmetry. A combined method has been designed, which can optimize both solution quality and computational cost. A dedicated method for remapping the deviatoric stress tensor has been incorporated in the remapper, that guarantees the preservation of the second invariant bounds, does not distort the symmetry of deformation directions and preserves the total elastic energy where possible.

We have shown that the ALE method with the presented novel features can handle a wide range of problems including shock propagation in non-homogeneous media, cavity closure, impacts and collisions, deformation and bending. It is possible to perform these computations with a single numerical code where only initial conditions, material models and simulation parameters are changed. Our research has extended the capabilities of the ALE method

8. *Conclusion*

especially in non-trivial solid dynamics problems and has allowed for more complex geometries of problems to be included in the simulations.



Bibliography

- [1] 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*, 35(4):619–630, 2017.
- [2] M. R. Baer and J. W. Nunziato. A two-phase mixture theory for the deflagration-to-detonation transition (ddt) in reactive granular materials. *International Journal of Multiphase Flow*, 12(6):861–889, 1986.
- [3] A. Barlow, R. Hill, and M. Shashkov. Constrained optimization framework for interface-aware sub-scale dynamics closure models for multi-material cells in Lagrangian and arbitrary Lagrangian-Eulerian hydrodynamics. Technical Report LA-UR-13-26180, Los Alamos National Laboratory, 2013.
- [4] 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.
- [5] A. Barlow, M. Klima, and M. Shashkov. Constrained optimization framework for interface-aware sub-scale dynamics models for voids closure in Lagrangian hydrodynamics. *Journal of Computational Physics*, 371:914–944, 2018.
- [6] 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.
- [7] 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.

- [8] T. J. Barth. Numerical methods for gasdynamic systems on unstructured meshes. In C. Rohde D. Kroner, M. Ohlberger, editor, *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.
- [9] 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.
- [10] D. J. Benson. Computational methods in Lagrangian and Eulerian hydrocodes. *Computer Methods in Applied Mechanics and Engineering*, 99(2-3):235–394, 1992.
- [11] 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.
- [12] E. Bezhlebová, V. Dolejší, and M. Feistauer. Discontinuous Galerkin method for the solution of a transport level-set problem. *Computers & Mathematics with Applications*, 72(3):455–480, 2016.
- [13] W. Boscheri. High order direct Arbitrary-Lagrangian–Eulerian (ALE) finite volume schemes for hyperbolic systems on unstructured meshes. *Archives of Computational Methods in Engineering*, 24(4):751–801, 2017.
- [14] J.U Brackbill and J.S Saltzman. Adaptive zoning for singular problems in two dimensions. *Journal of Computational Physics*, 46(3):342–368, 1982.
- [15] D. E. Burton, N. R. Morgan, M. R. J. Charest, M. A. Kenamond, and J. Fung. Compatible, energy conserving, bounds preserving remap of hydrodynamic fields for an extended ALE scheme. *Journal of Computational Physics*, 355:492–533, 2018.
- [16] J. C. Campbell and M. J. Shashkov. A Compatible Lagrangian Hydrodynamics Algorithm for Unstructured Grids. *Selcuk Journal of Applied Mathematics*, 4(2):53–70, 2003.
- [17] 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.
- [18] 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.

- [19] 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.
- [20] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen Differenzgleichungen der mathematischen Physik. *Mathematische annalen*, 100(1):32–74, 1928.
- [21] O. Darrigol. *Worlds of flow: A history of hydrodynamics from the Bernoullis to Prandtl*. Oxford University Press, 2005.
- [22] F. Daude and P. Galon. On the computation of the Baer–Nunziato model using ALE formulation with HLL- and HLLC-type solvers towards fluid–structure interactions. *Journal of Computational Physics*, 304:189–230, 2016.
- [23] B. Després 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.
- [24] 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.
- [25] N. Van Dorsselaer and V. Lapoujade. A contribution to new ALE 2D method validation. In *Proceedings of the 11th International LS-DYNA Users Conference, June*, pages 6–8, 2008.
- [26] J. K. Dukowicz and J. R. Baumgardner. Incremental remapping as a transport/advection algorithm. *Journal of Computational Physics*, 160(1):318–335, 2000.
- [27] J. K. Dukowicz and J. W. Kodis. Accurate conservative remapping (rezoning) for arbitrary Lagrangian-Eulerian computations. *SIAM Journal on Scientific and Statistical Computing*, 8(3):305–321, 1987.
- [28] V. Dyadechko and M. Shashkov. Reconstruction of multi-material interfaces from moment data. *Journal of Computational Physics*, 227(11):5361–5384, 2008.
- [29] J. Fořt, J. Fürst, K. Kozel, A. Jirásek, and M. Kladrubský. Numerical solution of 2D and 3D transonic flows over an airfoil and a wing. In H. Sobieczky, editor, *IUTAM Symposium Transsonicum IV*, pages 211–216, Dordrecht, 2003. Springer Netherlands.
- [30] D. Fridrich, R. Liska, and B. Wendroff. Cell-centered Lagrangian Lax–Wendroff HLL hybrid method for elasto-plastic flows. *Computers & Fluids*, 157:164–174, 2017.

- [31] E. Gaburro, M. Dumbser, and M. J. Castro. Direct Arbitrary-Lagrangian-Eulerian finite volume schemes on moving nonconforming unstructured meshes. *Computers & Fluids*, 159:254–275, 2017.
- [32] 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.
- [33] 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.
- [34] F. H. Harlow. Fluid dynamics in Group T-3 Los Alamos National Laboratory. *Journal of Computational Physics*, 195(2):414–433, 2004.
- [35] A. K. Harrison, M. J. Shashkov, J. Fung, J. R. Kamm, and T. R. Canfield. Development of a sub-scale dynamics model for pressure relaxation of multi-material cells in Lagrangian hydrodynamics. In *New Models and Hydrocodes for Shock Wave Processes in Condensed Matter*, volume 10 of *EPJ Web of Conferences*, page 00039, 2010.
- [36] R. N. Hill and M. Shashkov. The symmetric Moment-of-Fluid interface reconstruction algorithm. *Journal of Computational Physics*, 249:180–184, 2013.
- [37] 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.
- [38] 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.
- [39] 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.
- [40] M. Klima. Pseudo-hybrid remapping. Master’s thesis, Czech Technical University in Prague, 2014.
- [41] M. Klima, A. Barlow, M. Kucharik, and M. Shashkov. An interface-aware sub-scale dynamics multi-material cell model for solids with void closure and opening at all speeds. *Computers & Fluids*, 208:104578, 2020.
- [42] M. Klima, M. Kucharik, and M. Shashkov. Combined swept region and intersection-based single-material remapping method. *International Journal on Numerical Methods in Fluids*, 85(6):363–382, 2017.

- [43] 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.
- [44] 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*, 2018. Proceedings of HYP 2016, XVI International Conference on Hyperbolic Problems - Theory, Numerics, Applications, Aachen, Germany.
- [45] M. Klima, M. Kucharik, J. Velechovsky, and M. Shashkov. Second-invariant-preserving remap of the 2D deviatoric stress tensor in ALE methods. *Computers & Mathematics with Applications*, 78(2):654–669, 2019.
- [46] P. Knupp, L. G. Margolin, and M. Shashkov. Reference Jacobian optimization-based rezone strategies for Arbitrary Lagrangian Eulerian methods. *Journal of Computational Physics*, 176(1):93–128, 2002.
- [47] P. Knupp and S. Steinberg. *Fundamentals of Grid Generation*. CRC Press, Boca Raton, 1993.
- [48] P.M. Knupp. Winslow smoothing on two-dimensional unstructured meshes. *Engineering with Computers*, 15:263–268, 1999.
- [49] Tz. V. Kolev and R. N. Rieben. A tensor artificial viscosity using a finite element approach. *Journal of Computational Physics*, 228(22):8336–8366, 2009.
- [50] M. Kucharik. *Arbitrary Lagrangian-Eulerian (ALE) Methods in Plasma Physics*. PhD thesis, Czech Technical University in Prague, 2006.
- [51] M. Kucharik. Numerical methods for quantity remapping in the context of indirect Arbitrary Lagrangian-Eulerian (ALE) hydrodynamics, 2017. Habilitation, Czech Technical University in Prague.
- [52] 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.
- [53] 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.
- [54] 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*, 354:1–25, 2018.

- [55] 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 Fůrst, Jan Halama, Raphaël 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.
- [56] 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.
- [57] 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.
- [58] M. Kucharik, M. Shashkov, and B. Wendroff. An efficient linearity-and-bound-preserving remapping method. *Journal of Computational Physics*, 188(2):462–471, 2003.
- [59] D. Kuzmin, R. Lohner, and S. Turek, editors. *Flux Corrected Transport: Principles, Algorithms, and Applications*. Scientific Computation Series. Springer-Verlag New York, LLC, 2005.
- [60] P.H. Lauritzen, Ch. Erath, and R. Mittal. On simplifying ‘incremental remap’-based transport schemes. *Journal of Computational Physics*, 230(22):7957–7963, 2011.
- [61] R. Liska, J. Limpouch, M. Kuchařík, and O. Renner. Selected laser plasma simulations by ALE method. In *Journal of Physics: Conference Series*, volume 112, page 022009. IOP Publishing, 2008.
- [62] Ch. Lohmann. Flux-corrected transport algorithms preserving the eigenvalue range of symmetric tensor quantities. *Journal of Computational Physics*, 350:907–926, 2017.
- [63] 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.
- [64] P. Louda and J. Příhoda. On the modelling turbulent transition in turbine cascades with flow separation. *Computers & Fluids*, 181:160–172, 2019.
- [65] P. Lynch. *The emergence of numerical weather prediction: Richardson’s dream*. Cambridge University Press, 2006.
- [66] 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.

- [67] P.-H. Maire, R. Abgrall, J. Breil, and J. Ovardia. A cell-centered Lagrangian scheme for two-dimensional compressible flow problems. *SIAM Journal on Scientific Computing*, 29(4):1781–1824, 2007.
- [68] L. G. Margolin. Introduction to "An arbitrary Lagrangian-Eulerian computing method for all flow speeds". *Journal of Computational Physics*, 135(2):198–202, 1997.
- [69] 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.
- [70] 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.
- [71] 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.
- [72] R. Menikoff. Equations of state and fluid dynamics. Technical Report LA-UR-07-3989, Los Alamos National Laboratory, 2007.
- [73] G. H. Miller and E. G. Puckett. A high-order Godunov method for multiple condensed phases. *Journal of Computational Physics*, 128(1):134–164, 1996.
- [74] R. M. More, K. Warren, D. Young, and G. Zimmerman. A new quotidian equation of state (QEOS) for hot dense matter. *Physics of Fluids*, 31(10):3059–3078, 1988.
- [75] J. Von Neumann and R. D. Richtmyer. A method for the numerical calculation of hydrodynamic shocks. *Journal of Applied Physics*, 21(3):233–237, 1950.
- [76] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999.
- [77] S. Osher and R. P. Fedkiw. Level set methods: an overview and some recent results. *Journal of Computational Physics*, 169(2):463–502, 2001.
- [78] J. M. Owen and M. Shashkov. Arbitrary Lagrangian Eulerian remap treatments consistent with staggered compatible total energy conserving Lagrangian methods. *Journal of Computational Physics*, 273:520–547, 2014.
- [79] 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.

- [80] K. O. Pope, K. H. Baines, A. C. Ocampo, and B. A. Ivanov. Impact winter and the Cretaceous/Tertiary extinctions: Results of a Chicxulub asteroid impact model. *Earth and Planetary Science Letters*, 128(3-4):719–725, 1994.
- [81] M.H. Rice, R.G. McQueen, and J.M. Walsh. Compression of solids by strong shock waves. In F. Seitz and D. Turnbull, editors, *Advances in Research and Applications*, volume 6 of *Solid State Physics*, pages 1–63. Academic Press, 1958.
- [82] W.J. Rider. Void’s unphysical response in hydrocodes. Presentation at Nuclear Explosives Code Development Conference held October 20-24, 2014 in Los Alamos, NM, SAND2014-18762PE. Available at <https://www.osti.gov/scitech/biblio/1241862>.
- [83] W.J. Rider. A very brief history of hydrodynamic codes (i.e. hydrocodes). Presented at Sandia seminar, June 27, 2007, 2007.
- [84] S. K. Sambasivan, M. Shashkov, and D. E. Burton. Exploration of new limiter schemes for stress tensors in Lagrangian and ALE hydrocodes. *Computers & Fluids*, 83:98–114, 2013.
- [85] S. K. Sambasivan, M. J. Shashkov, and D. E. Burton. A finite volume cell-centered Lagrangian hydrodynamics approach for solids in general unstructured grids. *International Journal for Numerical Methods in Fluids*, 72(7):770–810, 2013.
- [86] K. Schittkowski. QL: A Fortran code for convex quadratic programming – user’s guide. Technical report, University of Bayreuth, Dept. of Computer Science, 2011. <http://www.klaus-schittkowski.de/software.htm>, Accessed 21.11.2016.
- [87] S. P. Schofield, R. V. Garimella, M. M. Francois, and R. Loubere. Material order-independent interface reconstruction using power diagrams. *International Journal of Numerical Methods in Fluids*, 56:643–659, 2008.
- [88] 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.
- [89] L. I. Sedov. *Similarity and Dimensional Methods in Mechanics, Tenth Edition*. Taylor & Francis, 1993.
- [90] M. Shashkov. Closure models for multimaterial cells in arbitrary Lagrangian-Eulerian hydrocodes. *International Journal for Numerical Methods in Fluids*, 56(8):1497–1504, 2008.

- [91] M. Shashkov and B. Wendroff. The repair paradigm and application to conservation laws. *Journal of Computational Physics*, 198(1):265–277, 2004.
- [92] C.D. Sijoy and S. Chaturvedi. An Eulerian multi-material scheme for elastic–plastic impact and penetration problems involving large material deformations. *European Journal of Mechanics - B/Fluids*, 53:85–100, 2015.
- [93] A. J. M. Spencer. *Continuum mechanics*. Courier Corporation, 2004.
- [94] H. B. Stewart and B. Wendroff. Two-phase flow: Models and methods. *Journal of Computational Physics*, 56(3):363–409, 1984.
- [95] P. Sváček. Application of finite element method in aeroelasticity. *Journal of Computational and Applied Mathematics*, 215(2):586–594, 2008. Proceedings of the Third International Conference on Advanced Computational Methods in Engineering (ACOMEN 2005).
- [96] R. E. Tipton. CALE mixed zone pressure relaxation model. Private communication, 1989.
- [97] S. A. Tokareva and E. F. Toro. A flux splitting method for the Baer–Nunziato equations of compressible two-phase flow. *Journal of Computational Physics*, 323:45–74, 2016.
- [98] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer Verlag, Berlin, Heidelberg, 1997.
- [99] P. Vachal, R. Garimella, and M. Shashkov. Untangling of 2D meshes in ALE simulations. *Journal of Computational Physics*, 196(2):627–644, 2004.
- [100] 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.
- [101] R. von Mises. Mechanik der festen Körper im plastisch-deformablen Zustand. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 1913:582–592, 1913.
- [102] P. Váchal and P.-H. Maire. Discretizations for weighted condition number smoothing on general unstructured meshes. *Computers & Fluids*, 46(1):479–485, 2011.
- [103] M. L. Wilkins. *Computer Simulation of Dynamic Phenomena*. Scientific Computation. Springer, 1999.
- [104] M.L. Wilkins. Calculation of elastic-plastic flow. Technical Report UCRL-7322, California. Univ., Livermore. Lawrence Radiation Lab., 1963.

- [105] 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.
- [106] S. T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of Computational Physics*, 31(3):335–362, 1979.
- [107] X. Zeng and G. Scovazzi. A frame-invariant vector limiter for flux corrected nodal remap in arbitrary Lagrangian–Eulerian flow computations. *Journal of Computational Physics*, 270:753–783, 2014.

■ List of selected publications by the dissertation author

■ Impacted journals:

- 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.
- M. Klima, M. Kucharik, and M. Shashkov. Combined swept region and intersection-based single-material remapping method. *International Journal on Numerical Methods in Fluids*, 85(6):363–382, 2017.
- A. Barlow, M. Klima, and M. Shashkov. Constrained optimization framework for interface-aware sub-scale dynamics models for voids closure in Lagrangian hydrodynamics. *Journal of Computational Physics*, 371:914–944, 2018.
- M. Klima, M. Kucharik, J. Velechovsky, and M. Shashkov. Second-invariant-preserving remap of the 2D deviatoric stress tensor in ALE methods. *Computers & Mathematics with Applications*, 78(2):654–669, 2019.
- M. Klima, A. Barlow, M. Kucharik, and M. Shashkov. An interface-aware sub-scale dynamics multi-material cell model for solids with void closure and opening at all speeds. *Computers & Fluids*, 208:104578, 2020.

■ Conference proceedings:

- M. Klima, M. Kucharik and M. Shashkov: Combination of Intersection- and Swept-Based Methods for Single-Material Remap, *Proceedings of 6th European Congress on Computational Fluid Dynamics*, pp. 5977-5988, 2014.
- 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*, 2018. *Proceedings of HYP 2016, XVI International Conference on Hyperbolic Problems - Theory, Numerics, Applications*, Aachen, Germany.

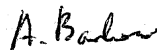
■ Declaration of co-authors

I hereby declare that I agree with including the following research articles that I have co-authored as a part of this dissertation. I confirm that the author of this dissertation has described his affiliation and involvement in the collective research accurately (as stated in Section 7) and that this dissertation contains his original contributions to the research.

 (M. Shashkov)
8/26/2020

Mikhail Shashkov
Los Alamos National Laboratory, New
Mexico, USA

I hereby declare that I agree with including the following research articles that I have co-authored as a part of this dissertation. I confirm that the author of this dissertation has described his affiliation and involvement in the collective research accurately (as stated in Section 7) and that this dissertation contains his original contributions to the research.

 A. Barlow

Andrew Barlow
AWE Aldermaston, Reading, UK

I hereby declare that I agree with including the following research articles that I have co-authored as a part of this dissertation. I confirm that the author of this dissertation has described his affiliation and involvement in the collective research accurately (as stated in Section 7) and that this dissertation contains his original contributions to the research.



Jan Velechovský
Los Alamos National Laboratory, New
Mexico, USA

I hereby declare that I agree with including the following research articles that I have co-authored as a part of this dissertation. I confirm that the author of this dissertation has described his affiliation and involvement in the collective research accurately (as stated in Section 7) and that this dissertation contains his original contributions to the research.



Milan Kuchařík
Czech Technical University in Prague,
Faculty of Nuclear Sciences and Physical
Engineering



Appendix A

Original Research Article: Constrained optimization framework for interface-aware sub-scale dynamics models for voids closure in Lagrangian hydrodynamics

Authors: Andrew Barlow, Matej Klima, Mikhail Shashkov

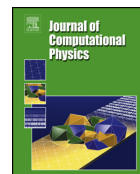
Published in Journal of Computational Physics. 2018; vol. 371, pp. 914-944.



Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp



Constrained optimization framework for interface-aware sub-scale dynamics models for voids closure in Lagrangian hydrodynamics

Andrew Barlow^a, Matej Klima^{b,c}, Mikhail Shashkov^{c,*}^a Computational Physics Group, AWE Aldermaston, Reading, Berkshire, RG7 4PR, UK^b Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Břehova 7, Praha 1, 115 19, Czech Republic^c X-Computational Physics, XCP-4, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

ARTICLE INFO

Article history:

Received 28 September 2017

Received in revised form 26 January 2018

Accepted 21 March 2018

Available online 2 April 2018

Keywords:

Lagrangian hydrodynamics

Multimaterial cells

Interface-aware sub-scale closure models

Void closure

ABSTRACT

In hydrocodes, voids are used to represent vacuum and model free boundaries between vacuum and real materials. We give a systematic description of a new treatment of void closure in the framework of the multimaterial arbitrary Lagrangian–Eulerian (ALE) methods. This includes a new formulation of the interface-aware sub-scale-dynamics (IA-SSD) closure model for multimaterial cells with voids, which is used in the Lagrangian stage of our indirect ALE scheme.

The results of the comprehensive testing of the new model are presented for one- and two-dimensional multimaterial calculations in the presence of voids. We also present a sneak peek of a realistic shaped charge calculation in the presence of voids and solids.

© 2018 Elsevier Inc. All rights reserved.

1. Introduction and rationale

In this paper we consider the class of so-called indirect Arbitrary Lagrangian–Eulerian (ALE) numerical methods [11,4] in which the solution process is separated into three distinct stages. These are:

1. **Lagrangian stage**, in which the solution and the computational mesh is updated,
2. **Rezoning stage**, in which the nodes of the computational mesh are moved to more optimal positions,
3. **Remapping stage**, in which the Lagrangian solution is conservatively transferred to the rezoned mesh.

In particular, any Eulerian method on an arbitrary mesh can be considered an ALE method if the mesh is always returned to its initial configuration at each rezone stage. This type of Eulerian methods is usually referred to as “Eulerian = Lagrange + Remap”.

Multimaterial cells, which may contain several materials, can appear even in pure Lagrangian calculations. This can happen if the initial configuration consists of multiple material regions that have complicated geometric shapes, such that it is hard, or impossible, for a given mesh resolution to create an initial mesh which conforms to the material interfaces.

In many multimaterial problems, the initial mesh can be aligned with a material interface so that each cell of the mesh contains only one material. However, complex high-speed multimaterial flows with strong shear deformations often

* Corresponding author.

E-mail addresses: Andy.Barlow@awe.co.uk (A. Barlow), klimamat@jfci.cvut.cz (M. Klima), shashkov@lanl.gov (M. Shashkov).

make it impossible to keep material interfaces aligned with the mesh. ALE methods are therefore currently the only proven technology to solve such problems. In ALE methods, the mesh does not move with the fluid and so it is unavoidable that multimaterial cells containing several materials will appear.

In many real applications one might need to deal not only with real materials, but also with voids. Voids can be considered as a way of representing both internal free boundaries (e.g. cavities, pores) and external free boundaries (expansion in to vacuum). Both Lagrangian and Eulerian hydrocodes encounter difficulties with modeling problems related to expansion in to vacuum [15,24] and more recent papers [17,12].

The artificial gas approach

In some codes, voids are modeled as a light and compressible *artificial gas*, often with air-like material properties. Here we will give some arguments why, in our opinion, such an approach may lead to physically wrong results and affect the robustness of the algorithm. We can start by saying that clearly there are no universal gas properties that can be used to represent voids for all problems. And even in a single problem where different materials are in contact with the vacuum, there is no clear recommendations as to how light and how compressible the gas should be.

Let us now consider an example – projectile impact problem, it is critical for such a problem to have an accurate void closure and void opening capability. If one models such an impact problem with both the impactor and the target surrounded by air, some of the air can often be trapped between the two materials as they come together. The air may then end up in a spurious state, which may in turn cause robustness problems. This can also lead to the wrong contact states as the pressure in the air will not match the normal stress between the plates at impact when there is no air present between the plates. For an enclosed cavity the air may stop the materials coming into close contact (e.g. 1D impact of plates separated by gas). The remaining air may reach non-physical states in equation of state (EOS) space, with high sound speeds – e.g. projectile impact problems and collapsing pores filled with air.

Consider a situation where the air remains after an impact. Gases cannot support tension, so how can void opening be detected? And if the materials do open, what will happen to the state of the gas as it is stretched to lower and lower density? Will the gas that was not trapped flow back into the opening to produce the correct gas thermodynamic states or lead to spurious states and sound speeds?

Alternatively, one may think of modeling void opening in isolation either within a material or at an initially closed material interface. Without introducing voids, from where is the artificial gas going to come from? Adding an arbitrary amount of gas will create mass.

The robustness issue is also important for problems which involve very high deformation. In such situations the artificial gas often has no impact on the physics of the problem, but must be meshed to allow multimaterial ALE to be used to handle the high deformation in the solid materials. If air is used, the air equation of state can get in to spurious high sound speed states very easily, which often cause calculations to fail for many reasons, e.g. mesh tangling, over advection etc.

In addition, the artificial gas representing the void requires EOS calls in the code. Moreover, in ALE methods the artificial gas is treated just like any other real material during the rezoning and remapping stages. These factors both contribute to the overall computational cost.

Treating the space around solid components as voids removes all these robustness issues without degrading the fidelity of the simulation and allows impacts and material opening to be handled accurately and robustly. We are convinced of the importance of developing a void treatment within the framework of Lagrangian and ALE methods, hence, the need for this paper. In this paper we will consider void closure only.

A review of our algorithm

The staggered compatible discretization of Lagrangian hydrodynamics is used in this paper [7,4]. In this framework the velocity is defined at the nodes of the mesh – it is assumed that all materials have the same velocity (single velocity model) within multimaterial cells. Each material has its own volume, density, specific internal energy and pressure. Void has non-zero volume (that is the volume of the space it occupies), but the rest of the quantities are zero. In the single material case, mass, density, internal energy and pressure are defined in the cells of the mesh. In the case of multimaterial flows, there may be pure cells containing only one material (a real material or void) and multimaterial cells which may contain several real materials and void. In a multimaterial cell, each material (including void) is represented by its pure material polygon (sub-cell). These are obtained by using an interface reconstruction method. In our paper we use the moment-of-fluid (MOF) interface reconstruction method [10,1] which was recently extended [13] such that it returns not just material polygons but also their connectivity (*mini mesh*), which is needed for our interface-aware closure model.

Multimaterial cells in our ALE method can represent interfaces between real materials as well as free boundary (interfaces of the real materials with vacuum) that can undergo high deformation. The main research concerns are:

- Accurately updating the thermodynamic states of the real individual material components in the multimaterial cell,
- deciding when the void closes,
- determining the nodal forces that the multimaterial zone generates despite the lack of information about the velocity distribution within multimaterial cells.

A *closure model* is then required to close the governing equations, which are otherwise under-determined. Its role is to define how the volume fractions of all the materials (including void) will evolve in time and how the states of the real individual materials evolve during a single Lagrangian step.

The approach described in this paper is based on the IA-SSD concept [3] originally designed for multimaterial flows in which computational cells may contain several real materials. In this paper we introduce special procedures for void closure and extend the IA-SSD approach so that the multimaterial cells can contain voids as well. The IA-SSD closure model for the Lagrangian stage of ALE consists of two stages:

1. **Bulk stage** uses the equal compressibility model [6] (Section 3.11), [3,4] or the constant volume fraction model in the case of fluids. In this model, the total change in the volume of the multimaterial cell is distributed between all materials according to the volume fraction that is constant in time. This applies for both real materials and voids, therefore, void cannot close at this stage. This model does not take into account any sub-scale interactions between materials inside the multimaterial cell.
2. **Sub-scale stage** takes into account the interactions of the materials inside the multimaterial cell. At this stage, information about the topology of the sub-cells inside the multimaterial cell is utilized, allowing the orientation of internal interfaces to be included in the model. Each material, including void, interacts in a pair-wise fashion with all materials that it shares a common boundary with.

The pair-wise interactions of the real materials and real material and void are based on the solution of the acoustic Riemann problem and are limited using physically justified constraints. Those constraints are:

- Positivity of volume for real material sub-cells,
- non-negativity of volume of void sub-cells,
- positivity of internal energy,
- a controlled rate of pressure relaxation.

The volume of voids can be zero as a result of the sub-scale stage – this indicates that the void has been closed. To determine the values of the limiter coefficients, a constrained-optimization framework is employed using a quadratic objective function with linear constraints.

There is nothing special with the rezone stage for ALE in the presence of voids – all points are moved according to some rezone strategy [4]. Clearly, the choice of the rezone strategy will strongly affect the results of calculations. To make our results reproducible, in all our examples we use the Eulerian = Lagrange + Remap approach in which all points move back to their original positions.

In this paper remapping is performed using the flux form of the intersection-based remap as described in [4,14]. To make our results reproducible, we use a first-order remap. There is nothing special about the remapping of cells with voids, void just acts as a special “material” with zero density (mass) and zero internal energy. In the first order case, there is nothing special about nodal remap. However, in Section 5 we briefly describe the so-called *kinetic energy fix* which is needed in the general case to enforce the conservation of total energy. We do not describe any other details of the remapping in this paper and instead we refer the interested reader to [4,14].

The remainder of the paper is organized as follows. In Section 2 we introduce the notation related to the representation of the materials. The generic description of the Lagrangian stage of ALE is given in Section 3. The sub-scale stage of the closure model is presented in Section 4. Algorithmic choices for the closure model are described and justified in Section 5. Numerical examples are presented in Section 6. Conclusions and a description of future research is provided in Section 7.

2. Notation and the representation of materials

To be consistent with [3] we will use the same notation. The cell (zone) of the mesh is denoted by z and all zonal quantities are identified by the subscript z . For example, V_z represents the volume of the zone. The superscript n is used for identifying quantities at the time level t^n . For example, the volume of the zone z at time t^n is denoted by V_z^n .

The multimaterial zones are identified by the presence of more than one material, real or void. The index i is used to identify specific materials (including void). The volume of the material i in zone z at time t^n is denoted by $V_{z,i}^n$. Each real material has its own mass, $m_{z,i}$; density $\rho_{z,i}^n$; pressure, $p_{z,i}^n$ and internal energy, $\varepsilon_{z,i}^n$. It is useful to introduce the volume fraction for each material, $0 < \alpha_{z,i}^n < 1$.

If a multimaterial cell contains void the index v will be used to distinguish it from the real materials *only* if void is treated in a different way. Void is described by its volume $V_{z,v}^n$ and volume fraction $\alpha_{z,v}^n$. The void density (mass), pressure, speed of sound and internal energy are always zero.

There are also two special types of cell vertices/points. The *void* points are points surrounded by pure void cells only. Points are designated as *real-material* if at least one of the surrounding cells contains at least one real material sub-cell (but voids can be present as well).

3. Generic description of the Lagrangian stage for multimaterial flows

In this section we will give a generic description of the Lagrangian stage in the presence of multimaterial cells with voids. While voids can close, a real material present in the multimaterial cell at time t^n will remain in that cell upon the completion of the Lagrangian step. We will describe the predictor part of the time integration – for full details of time integration, one can refer to [3]. The Lagrangian stage starts with the computation of common pressures in the multimaterial cells

$$\tilde{p}_z^n = \sum_{i \in M(z)} \alpha_{z,i}^n p_{z,i}^n, \quad (1)$$

where $M(z)$ is the set of all materials in the multimaterial cell, including void material. The common pressure in the single-material cells is just equal to the material pressure and the common pressure in the void cells is zero.

Next, the corner forces in each cell acting on each of the cells points are computed using common zonal pressures.

This allows the new velocities of the *real-material points* to be computed.

The *void points* cannot be moved using the momentum equation because their mass is zero and there are no forces acting on them. This means that the mesh would tangle if only the Lagrangian stage was used. However, the Eulerian = Lagrange + Remap algorithm, which is used in this paper, avoids this situation. For a general ALE method, a special algorithm for the movement of such points at the Lagrangian stage may be necessary.

Next, the coordinates of all *real-material points*, are updated, which allows the new volumes of all zones to be computed. Let us note that the volumes of the pure void cells may also change if at least one of the vertices of the void cell is a *real-material point*.

Using the new volume we can update the density, specific internal energy and pressure in all the pure cells containing real materials.

For pure void cells we do not need to do anything.

Next, we need to update the densities and specific internal energies for all the real materials in the multimaterial cells. If a void was present in the multimaterial cell at t^n , we also need to decide if it will close. The objective of closure models for multimaterial cells is to answer all these questions. Our closure model is described in the next section.

4. Void closure model

In this section, where it does not lead to misunderstanding, the zonal index z is dropped because only a single cell is considered. Additionally, in most cases the time index is also dropped.

Our void closure model consists of two stages – bulk stage and sub-scale stage.

4.1. Bulk stage

At the bulk stage we use the equal compressibility model, whereby it is assumed that the volume fraction does not change in time, i.e. $\alpha_{z,i}^{n+1} = \alpha_{z,i}^n$. At bulk stage, the total change in the volume of the multimaterial cell, ΔV^{n+1} , is distributed between all constituents in the multimaterial cell, including void, proportionally to its volume fraction from the previous time step

$$\Delta V_i^{n+1, bulk} = \alpha_i^n \Delta V^{n+1}. \quad (2)$$

Therefore, void cannot close at this stage.

The update of density, specific internal energy and pressure for the real materials in the multimaterial cell is the same as described in [3] for the case when the multimaterial cell only contains real materials. In particular the update of the internal energies for each real material is done using individual internal energy equations and expression (2) for the change in the material volume

$$m_i (\varepsilon_i^{n+1} - \varepsilon_i^n) = -p_i^n \Delta V_i^{n+1, bulk}. \quad (3)$$

4.2. Sub-scale stage

The sub-scale stage takes into account the interactions between the materials inside the multimaterial cell. It is assumed that the topology of the materials inside a multimaterial cell is known from the interface reconstruction algorithm. Each material in the multimaterial cell is represented by a pure sub-polygon. The set of the materials in the multimaterial cell which have a face in common with material i is denoted by $M(i)$. At the sub-scale stage each material, including void, interacts in a pair-wise fashion with all the materials that it shares a common boundary with.

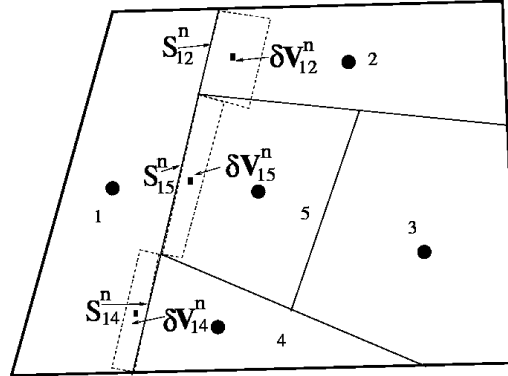


Fig. 1. Multimaterial cell: definition of the relative volume exchange.

4.2.1. Volume change model

To take into account the difference in pressure of the materials which share the face, the volume exchange between the neighboring materials is introduced in flux form as

$$\Delta V_i = \Delta V_i^{bulk} + \sum_{k \in M(i)} \delta V_{i,k}, \quad \delta V_{i,k} = -\delta V_{k,i}. \quad (4)$$

The volume exchange term is constructed as follows:

$$\delta V_{i,k} = \Psi_{i,k} \delta V_{i,k}^{max}, \quad (5)$$

where $\delta V_{i,k}^{max}$ is the maximum allowed volume exchange between adjacent materials, or real material and void, due to the imbalance in pressure, where $0 \leq \Psi_{i,k} \leq 1$ are the limiters which are chosen in such a way that the overall model does not violate physically justifiable constraints – for example, the positivity of the volume of any real material sub-cell. The obvious requirement $0 \leq \Psi_{i,k}$ ensures that the limiter does not reverse the direction of the “physical” force. The definition of the upper bound $\Psi_{i,k} \leq 1$ is more questionable, however, it is sensible to not increase the magnitude of the sub-scale fluxes.

The expression for the exchange term $\delta V_{i,k}^{max}$ between real materials is estimated using the acoustic Riemann solver (see [3] for details).

In the case of the volume exchange between real material and void we define $\delta V_{i,v}^{max}$ as follows

$$\delta V_{i,v}^{max} = \frac{p_i}{\rho_i c_i} S_{i,v} \Delta t, \quad (6)$$

where c_i is the adiabatic speed of sound in the material i , and $S_{i,v}$ is the area of the interface between materials i and void, see Fig. 1.

In Fig. 1, a multimaterial cell with five materials is presented with a graphical representation of $\delta V_{i,k}$ (the same notation is used for its volume and the corresponding rectangle). For the material #1 ($i = 1$) there are three neighbors: {2, 4, 5}. In the situation presented p_1 is larger than p_2 and p_5 and smaller than p_4 .

Bounds for material volume

Following [3], the following constraints are imposed on the volume of a real material sub-cell:

$$V_i^{n+1} \geq \kappa_{bot} V_i^{n+1,bulk}, \quad 1 \geq \kappa_{bot} > 0. \quad (7)$$

Here, the parameter κ_{bot} controls how close to zero the volume of the real material may reach. In the results presented in this paper κ_{bot} is presumed to be constant in time for all materials.

The constraints for real materials (7) represent a system of linear inequalities with respect to the limiters $\Psi_{i,k}$:

$$V_i^{n+1,bulk} + \sum_{k \in M(i)} \Psi_{i,k} \delta V_{i,k}^{max} \geq \kappa_{bot} V_i^{n+1,bulk}. \quad (8)$$

For void, the volume constraint can be expressed as

$$V_v^{n+1} \geq 0, \quad (9)$$

that is, void closes if $V_v^{n+1} = 0$ at the end of the subscale stage.

For void, the constraint (9) leads to the following linear inequality (with respect to $\Psi_{i,k}$):

$$V_v^{n+1,bulk} + \sum_{k \in M(v)} \Psi_{v,k} \delta V_{v,k}^{max} \geq 0. \tag{10}$$

4.2.2. Specific internal energy evolution

In the IA-SSD closure model, each real material has a separate approximate $p dV$ equation. In [3] we use the following equation which guarantees the conservation of total energy:

$$m_i (\varepsilon_i^{n+1} - \varepsilon_i^n) = -p_i^n \alpha_i^n \Delta V_i^{n+1} - \sum_{k \in M(i)} p_{i,k}^* \Psi_{i,k} \delta V_{i,k}^{max}, \tag{11}$$

where the pressure $p_{i,k}^*$ is obtained from the one-dimensional acoustic Riemann problem between materials i and k .

Since the time of publishing [3], we have found that better results can be obtained by using a non-conservative approximation of the internal energy equation in the following form:

$$m_i (\varepsilon_i^{n+1} - \varepsilon_i^n) = -p_i^n \Delta V_i^{n+1}. \tag{12}$$

Here, the material pressure is multiplied by the overall change of the material volume. We have performed experiments similar to those presented in [3] and have found that the conservation of total energy is violated on the level of less than one percent. A detailed justification of our algorithmic choices, including the approximation (12) of the internal energy equation, is presented in Section 5.

Positivity of internal energy

The inequality required for the positivity of internal energy of the real material on the new time step can be written as

$$\Delta \mathcal{E}_i^{bulk} \geq p_i^n \sum_{k \in M(i)} \Psi_{i,k} \delta V_{i,k}^{max}, \tag{13}$$

where

$$\Delta \mathcal{E}_i^{bulk} = m_i \varepsilon_i^n - p_i^n \Delta V_i^{bulk}. \tag{14}$$

This is another linear inequality constraint with respect to limiters $\Psi_{i,k}$. The specific internal energy of void is always zero, therefore this constraint does not apply to a void sub-cell.

4.2.3. Controlling pressure equilibration

Another important property of closure models is the manner in which pressure equilibration is achieved in time.¹ Pressure equilibration should be achieved without oscillations and in such a way that the pressures of different materials approach equilibrium in a smooth fashion. The linear inequality constraints related to smooth pressure equilibration are essentially the same as for the case of the multimaterial cells with real materials [3]. The only difference is that the pressure of void is always zero.

Consider the evolution of the material pressure during a single time step. If the time step is small enough it can be assumed that the entropy is constant during the time step² and the following approximation for material pressure evolution may be used:

$$\tilde{p}_i^{n+1}(\Psi_{i,k}) = p_i^n - \rho_i^n (a_i^n)^2 \Delta V_i^{n+1} / V_i^n = \tilde{p}_i^{bulk,n+1} - \frac{\rho_i^n (a_i^n)^2}{V_i^n} \sum_{k \in M(i)} \Psi_{i,k} \delta V_{i,k}^{max}, \tag{15}$$

where the following pressures are given:

$$\tilde{p}_i^{bulk,n+1} = p_i^n - \frac{\rho_i^n (a_i^n)^2}{V_i^n} \Delta V_i^{bulk,n+1}.$$

The second term on the RHS of (15) is again a linear function of $\Psi_{i,k}$. For void, the pressure is zero by definition and therefore $\tilde{p}_i^{bulk,n+1} = 0$.

To achieve smooth equilibration, a temporarily targeted equilibrated pressure, \bar{p} , is computed from known quantities. The material pressures are intended to relax towards this target pressure in this particular time moment. In general, material

¹ Here we consider a situation in which no new forces are applied to the multimaterial cell after a certain transition time and contact is established.
² Strictly speaking the assumption of isentropy is admissible providing that the shock does not cross the multimaterial cell. Therefore, we can consider such an assumption as a simplifying approximation which leads to linearization of equation of state.

pressures $p_i^{bulk,n+1}$ may be higher or lower than \bar{p} . However, the following inequality must always hold: $\max_i \tilde{p}_i^{bulk,n+1} \geq \bar{p} \geq \min_i \tilde{p}_i^{bulk,n+1}$. If a multimaterial cell does not contain void then we always choose

$$\bar{p} = \sum_i \alpha_i^n \tilde{p}_i^{bulk,n+1}. \quad (16)$$

If a multimaterial cell contains void then we have more options.

As the pressure in void is always zero and cannot change. Therefore, it is natural, to relax all material pressures towards zero. This means that in every cell containing void we set $\bar{p} = 0$. In Section 5.2 we will present a comparison of calculations which have been performed with $\bar{p} = 0$ and with \bar{p} defined by the expression (16) in cells with voids – it turns out that these calculations are not sensitive to which of these options is used.

The limiters $\Psi_{i,k}$ are chosen in such a way that \tilde{p}_i^{n+1} defined by equation (15) will, if possible, relax towards (or at least not diverge from) \bar{p} . The zero void pressure is fixed and as such is not affected by the limiters.

A derivation similar to the one presented in [3] leads to another system of linear inequalities with respect to the limiters – which is summarized in the next Section.

4.2.4. Constrained optimization framework

For each multimaterial cell z , the following minimization problem is solved

$$\min_{\Psi_{i,k}} \left\{ \sum_i \left[\sum_{k \in M(i)} (1 - \Psi_{i,k})^2 \right] \right\}. \quad (17)$$

The linear constraints come from the volume bounds, positivity of internal energy and pressure equilibration conditions described in the previous sections. For each material i , the constraints for the limiters $\Psi_{i,k}$ are:

- The constraint related to the volume of a real material i :

$$\sum_{k \in M(i)} \Psi_{i,k} \delta V_{i,k}^{max} \geq (\kappa_{bot} - 1) V_i^{n+1,bulk}. \quad (18)$$

- For void, we have the following volume related constraint:

$$\sum_{k \in M(v)} \Psi_{v,k} \delta V_{v,k}^{max} \geq -V_v^{n+1,bulk}. \quad (19)$$

- The constraint related to positivity of real material internal energy (13):

$$m_i \epsilon_i^n - p_i^n \Delta V_i^{n+1,bulk} \geq p_i^n \sum_{k \in M(i)} \Psi_{i,k} \Delta V_{i,k}^{max}. \quad (20)$$

- The constraints related to controlling pressure equilibration

$$\frac{V_i^n}{\rho_i^n (c_i^n)^2} (\tilde{p}_i^{bulk,n+1} - \bar{p}) \geq \sum_{k \in M(i)} \Psi_{i,k} \delta V_{i,k}^{max} \geq 0, \quad \text{if } \tilde{p}_i^{bulk,n+1} \geq \bar{p}; \quad (21)$$

$$\frac{V_i^n}{\rho_i^n (c_i^n)^2} (\tilde{p}_i^{bulk,n+1} - \bar{p}) \leq \sum_{k \in M(i)} \Psi_{i,k} \delta V_{i,k}^{max} \leq 0, \quad \text{if } \tilde{p}_i^{bulk,n+1} \leq \bar{p}. \quad (22)$$

Remember that for multimaterial cell with void \bar{p} is zero.

The constrained-optimization problem for finding the limiters $\Psi_{i,k}$ is evaluated as a quadratic optimization problem with linear inequality constraints [16]. In the current implementation, the “QL” software [21] is used to solve this problem.

4.2.5. Void closure detection

After we solve the constrained optimization problem and limiter values are determined we need to update all material quantities. In particular, the updated volume of void sub-cells is computed. If it is “zero” then it means that the void has closed. Therefore, we need to update the status of the multimaterial cell as it now does not contain void and the corresponding sub-cell must be removed.

More details can be found in [5]. In particular, in [5] we present explicit solution of the constrained optimization problem in the case when multimaterial cell contains one real material and void.

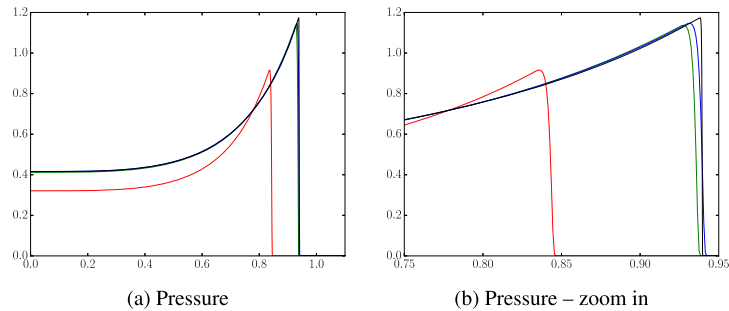


Fig. 2. Pressure in the Sedov-like problem demonstrating the importance of the kinetic energy fix. Comparing exact solution (–), no KE fix (–), KE fix with switch (–), full KE fix (–). (For interpretation of the colors in the figure(s) (and caption(s)), the reader is referred to the web version of this article.)

5. Algorithmic choices

5.1. Kinetic energy fix in arbitrary Lagrangian–Eulerian methods

An important aspect of ALE methods is the techniques used for remapping variables from the Lagrangian to the rezoned mesh. In this paper we use Eulerian = Lagrange + Remap version of ALE, such that after Lagrangian step data is transferred back to the initial Eulerian mesh.

The remap has to *conserve* the same quantities and in the same form as Lagrangian step does – conservation of mass, momentum and total energy – to ensure the conservation of the overall ALE method.

If the total energy after remap is computed as the sum of the conservatively remapped internal energy and the kinetic energy computed from the remapped velocity, which is computed from conservatively remapped momentum and remapped mass, then it will not be conservative, because some dissipation is introduced during the velocity remap.

Some codes ignore this discrepancy. Other codes (see, for example, [9,6,18–20,8,4]) use an approach often termed the *kinetic energy fix* to restore total energy conservation completely or locally near shock front. The kinetic energy fix is critical for correct computing flows with shock waves. In this approach one also conservatively remaps the kinetic energy. The full discrepancy (or part of it) between actual kinetic energy computed from remapped velocity and remapped kinetic energy is added to the internal energy, such that the total energy is conserved exactly (or total energy imbalance is decreased).

To demonstrate the importance of conservation of total energy we consider the computation of a Sedov-like shock. The initial conditions are $\rho = 1$, $u = 0$ everywhere. The internal energy is initialized as $\varepsilon = 100$ in the source region from 0 to 0.01 and $\varepsilon = 0.0001$ in the rest of the real material. This is not a standard Sedov problem with point source but rather a *finite size* source Sedov-like problem. In our experiments we choose a number of cells such that one of the nodes always lies at $x = 0.01$. This means that for refined meshes there are more cells in the region with high internal energy but the initial total energy in the high energy part of the domain is always the same and equal to $0.01 \times 100 = 1$. The computational domain is the segment $[0 : 1]$ and both boundaries are solid walls. The final time is $t = 0.5$. In Fig. 2 we present four graphs – the reference solution obtained on a very fine grid, and the results of three calculations with different treatments of the kinetic energy discrepancy on a mesh of 2500 cells.

The first treatment is the full kinetic energy fix – this is a completely conservative method – there is no loss of total energy. Such an approach produces numerical results which are very close to the reference solution. The second approach is no kinetic energy fix at all. In this approach we lose approximately 28% of the total energy and the results are absolutely wrong. The final approach is to perform the kinetic energy fix only for cells in which the ratio of the artificial viscosity to the pressure is greater than 1%. In this case we lose approximately 1.3% of the total energy and the results are still very accurate. Therefore, when modeling flows with shocks we have to use some form of kinetic energy fix. As will be shown in the next Section the use of the artificial viscosity/pressure switch for the kinetic energy fix is critical for smooth flows.

5.2. Algorithmic choices in closure model

To apply the closure model described in Section 4 we have to make several algorithmic choices. We set κ_{bot} in inequality (7) equal to 0.1.

In Fig. 3 we present the results for the different algorithmic choices for the expansion in to vacuum problem. This problem is also termed the Riemann problem in the presence of a vacuum [22]. We consider the case where an ideal gas on the left of the origin is initially at rest. The gas has a ratio of constant volume specific heat capacity to constant pressure specific heat capacity ($\gamma = 1.4$), a constant density $\rho_L = 1$ and constant pressure $p_L = 0.4$. The vacuum is on the right of the origin.

If we denote the adiabatic speed of sound of the initial state by a_L , where $a_L = \sqrt{\gamma(\gamma - 1)\varepsilon_L}$, then the solution, $\mathbf{W}(x, t) = (\rho, u, p)$ is

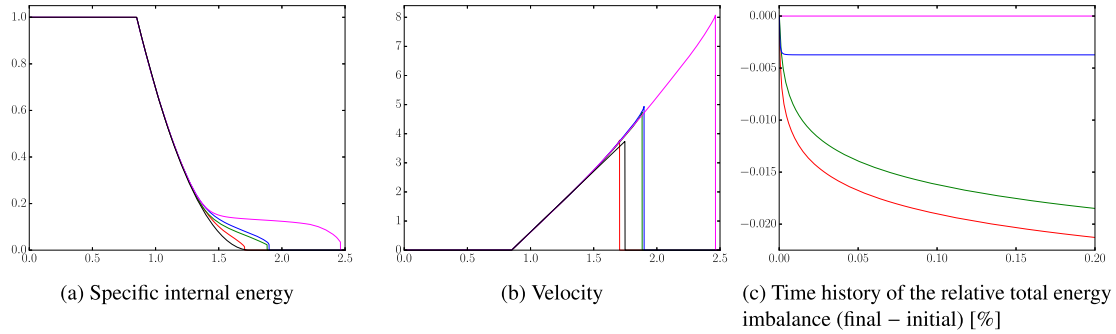


Fig. 3. Expansion into vacuum problem, different algorithmic choices: switch on, $p^* = p_{mat}$ (—), switch on, $p^* = 0$ (—), switch off, $p^* = p_{mat}$ (—), switch off, $p^* = 0$ (—), exact solution (—).

$$\mathbf{W}(x, t) = \begin{cases} \mathbf{W}_L & \text{if } \frac{x}{t} \leq -a_L \\ \mathbf{W}_{Lfan} & \text{if } -a_L < \frac{x}{t} < \frac{2a_L}{\gamma-1} \\ \mathbf{W}_0 & \text{if } \frac{x}{t} \geq \frac{2a_L}{\gamma-1}, \end{cases} \quad (23)$$

where

$$\mathbf{W}_{Lfan}(x, t) = \begin{cases} \rho = \rho_L \left[\frac{2}{\gamma+1} - \frac{\gamma-1}{(\gamma+1)a_L} \frac{x}{t} \right]^{\frac{2}{\gamma-1}} \\ u = \frac{2}{\gamma+1} \left[a_L + \frac{x}{t} \right] \\ p = p_L \left[\frac{2}{\gamma+1} - \frac{\gamma-1}{(\gamma+1)a_L} \frac{x}{t} \right]^{\frac{2\gamma}{\gamma-1}}. \end{cases} \quad (24)$$

The boundary between the gas and vacuum is moving with the velocity

$$u_{free} = \frac{2}{\gamma-1} a_L = \frac{2\sqrt{\gamma(\gamma-1)}\varepsilon}{\gamma-1} = 3.741657.$$

The exact solution of this problem conserves total energy.

In our numerical calculations we use the domain $[0 : 2.5]$ and the real gas is contained within the subregion $[0 : 1]$ and the rest of the domain is occupied by the vacuum (void in the calculations). The extensive numerical results for problem are presented in Section 6.1.2, in this section we just present some results which justify our algorithmic choices for the void closure model.

In Fig. 3a we show graphs for the specific internal energy at $t = 0.5$ with 10000 cells for different algorithmic choices. For all the calculations $\bar{p} = 0$. In the figure caption “switch off” means that we are performing the full kinetic energy fix and “switch on” meant that we use the switch based on the ratio of the artificial viscosity and pressure. The $p^* = 0$ means that we use internal energy update in the conservative form (11) with $p^* = 0$; the $p^* = p_{mat}$ means that we use internal energy update in the non-conservative form (12). The first conclusion from these results is that we have to use the switch, because completely conservative calculations (“switch off” and $p^* = 0$) give very inaccurate results. The velocity graphs presented in Fig. 3b confirm this conclusion.

From these figures we can also conclude that the best results are achieved when we use $p^* = p_{mat}$. Clearly, using the switch option for kinetic energy fix and $p^* = p_{mat}$ makes our method not completely conservative. The influence of these choices on conservation is demonstrated in Fig. 3c – the maximum loss of the total energy is less than 0.025%.

Finally we will demonstrate that our model is not sensitive to the choice of \bar{p} . In Fig. 4 we present results for “switch on”, $p^* = p_{mat}$ and two options for $\bar{p} = 0$ and \bar{p} computed as volume fraction average, (16). To see the difference between these options we need to zoom in on the solution. In the calculations with $\bar{p} = 0$ the front of free boundary moves faster.

To summarize our algorithmic choices for closure model we use the kinetic energy fix with the artificial viscosity/pressure switch which is on when this ratio is greater than 1%; we use the non-conservative internal energy update – (12); and we use $\bar{p} = 0$.

Let us note, that we do not claim that our algorithmic choices are optimal. This is why this paper has the word “framework” in its title. The optimal choice of the parameters may be problem dependent and will require more investigation.

6. Numerical examples

Three one-dimensional test problems are presented in Section 6.1. The first problem is a *sanity check* problem which involves the movement and collision of cold blocks of gas – Section 6.1.1. It demonstrates that our void closure treatment is

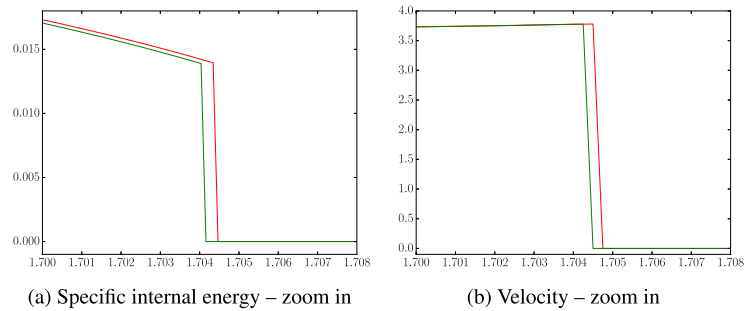


Fig. 4. Comparison of options for \bar{p} for expansion in to vacuum problem. Comparing \bar{p} computed as volume fraction average (–), $\bar{p} = 0$ (–).

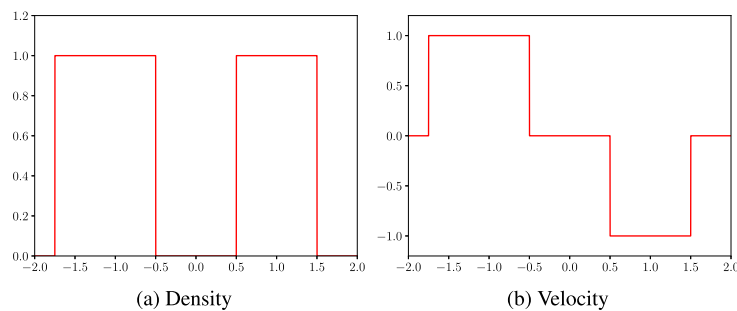


Fig. 5. Cold blocks collision problem – the initial conditions for density and velocity.

working and gives overall plausible results. In Section 6.1.2 we present results for the expansion of an ideal gas into vacuum and the subsequent collision of the free boundary with a wall. The expansion into vacuum (until the free boundary hits the wall) has an analytical solution. It is observed that the position of the free boundary is not accurate, however, after the free boundary hits the wall the solution converges to a reference solution. In the last one-dimensional example, Section 6.1.3, we explore how a Sedov-like shock wave interacts with the free boundary. The conclusions are the same as for the expansion in to vacuum problem.

The two-dimensional tests are presented in Section 6.2. In Section 6.2.1 we present a diagonal projectile-plate impact. Here we demonstrate the robustness of our method and give some comparisons with using an artificial “light” gas environment instead of void. This numerical example confirms our speculations in the Introduction about the problems arising with the usage of an artificial gas. In the following Section 6.2.2 we present an example of a shock wave interacting with a spherical void cavity. It gives plausible and the expected results – the void completely closes and the shock front is accelerated in the location of the void cavity. In Section 6.2.3 we present the results for a simplified model of a shaped-charge-like problem. It is intentionally set up so that it shows some of the features seen in real shaped charge simulations, but uses a simplified geometry and does not include more complex material features (only ideal gas EOS). That way the interested reader can attempt to reproduce our results or compare her/his own method for void closure. Finally, in Section 6.2.4 we present a sneak peek of the application of our new method for solving a realistic shaped charge problem with appropriate models for high-explosives and solids. This test is the unconfined shaped charge from [23,2].

6.1. One-dimensional tests

6.1.1. Movement and collision of cold blocks of gas

In this Section we present a *sanity check* example. The statement of the problem is as follows:

The computational domain is the segment $[-2, 2]$, the left and right boundaries are walls. There are two blocks of cold ideal gas, both with the following properties: $\gamma = 5/3$, $\rho = 1$, $\varepsilon = 1.0 \cdot 10^{-9}$. The first, longer, block, initially occupies the segment $[-1.75, -0.5]$ and its velocity is 1. The second, shorter, block occupies the segment $[0.5, 1.5]$ and its velocity is -1 . The rest of the computational domain is filled with void cells – see Fig. 5.

We perform the calculations using our new treatment of void closure using 10000 cells. In Fig. 6 we present the density and velocity at $t = 0.25$. It demonstrates that our new void treatment allows us to reproduce translation of solid bodies exactly.

In Fig. 7 we present the density, velocity and pressure at $t = 0.501$, that is, right after the collision. All void cells between the blocks are closed and the density and the pressure start to grow near the collision point.

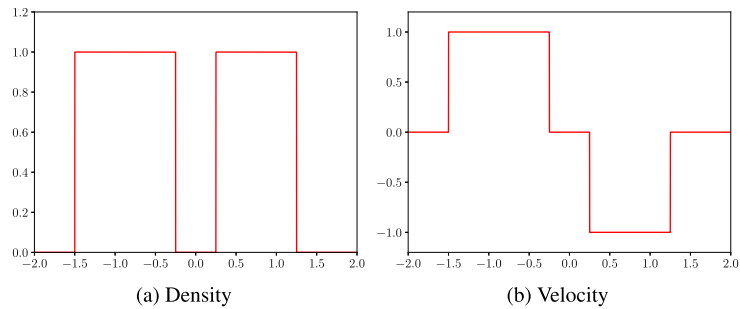


Fig. 6. Cold blocks collision problem, $t = 0.25$ – intermediate stage.

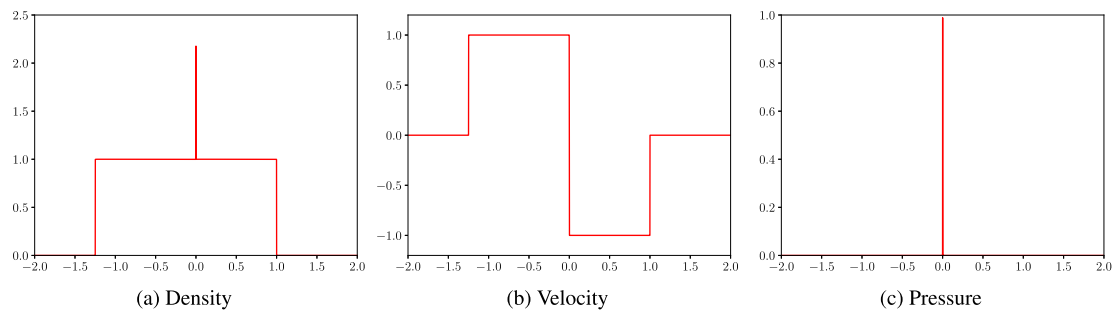


Fig. 7. Cold blocks collision problem, $t = 0.501$ – just after collision.

After the collision, two shocks form traveling in opposite directions within each block. In Fig. 8 we present the density, velocity, specific internal energy and pressure plots for the time moment $t = 1.2$ when the shock front in the right block has almost reached its right boundary. The standard wall heating phenomenon can be observed, which manifests itself as a dip in the density and an overshoot in the specific internal energy, while the velocity and pressure are flat. After the shocks have reached the free boundaries of both blocks, the expansion into vacuum starts. This moment, $t = 1.5$, is shown in Fig. 9. The expansion has obviously started earlier for the shorter right hand block.

In Fig. 10 time moment $t = 2.0$ is shown where the expansion wave from the right block has reached the right wall and all the voids are closed on the right side.

Finally, in Fig. 11, we show the time moment $t = 2.5$ where all the void is closed, and shock waves are starting to form from the reflections at both the left and right walls.

6.1.2. Expansion into vacuum

The second problem is the expansion of ideal gas into vacuum, which was fully defined in Section 5.2.

First we perform a Lagrangian simulation with only pure cells by explicitly imposing free boundary conditions at the right boundary.

The results for velocities and pressures at different mesh resolutions are compared with the exact solution at $t = 0.25$ in Fig. 12. It is clear from Fig. 12 that the convergence (if any) of the Lagrangian calculations is very slow and the position of the free boundary is very inaccurate even for such high resolutions as 32000 cells. This is a very well known result which is usually attributed to the presence of a very big cell next to the free boundary (the interested reader can refer to a recent presentation [17]).

However, to understand the accuracy issues one also needs to look not just at the position of the free boundary but also at the values of the density, specific internal energy and pressure near the free boundary. In Fig. 13 we show a zoom in of the density, specific internal energy and pressure profiles near free boundary for two spatial resolutions as well as the profile of the exact solution. As one can see from Fig. 13, the values of the Lagrangian numerical solution are slightly larger compared to the exact solution. However, in general the values of the pressure are small near the free boundary. The real question is what will happen after the free boundary interacts with some obstacle. Clearly, the start of the interaction will be delayed in comparison with exact solution, but it is not clear how it will affect the resulting flow afterwards – pressure in the exact solution near the free boundary goes to zero. We will address this question later in this section.

Now, we will use our new void closure method as described in the previous sections to solve the same problem. Let us remember that we are using an Eulerian = Lagrange + Remap approach. Therefore we need to use a longer computational domain because the interface between real gas and vacuum will move throughout the domain. In our calculation the

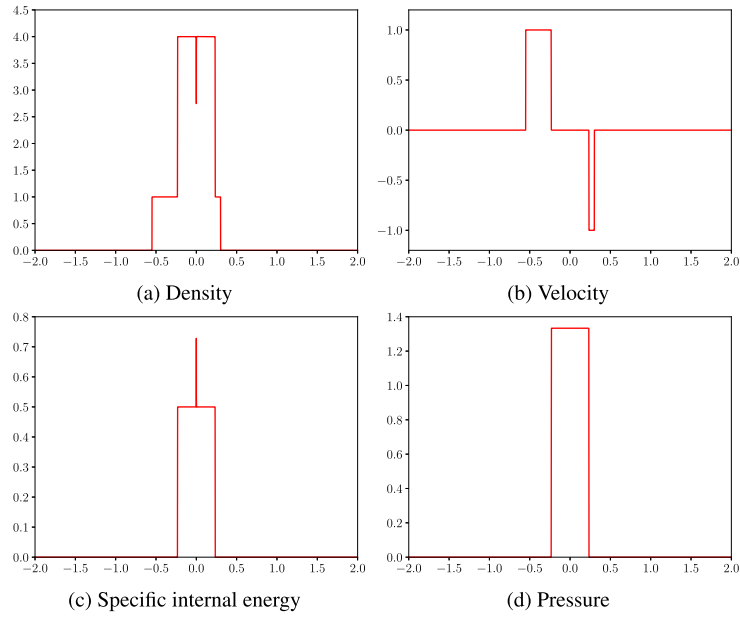


Fig. 8. Cold blocks collision problem, $t = 1.2$ – the shock has almost reached the right boundary of the right block.

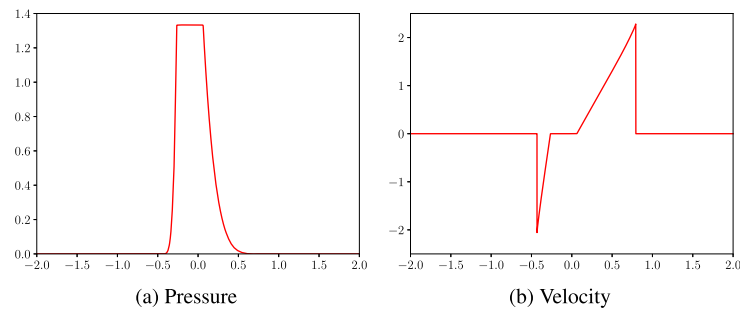


Fig. 9. Cold blocks collision problem, $t = 1.5$ – expansion into vacuum is starting.

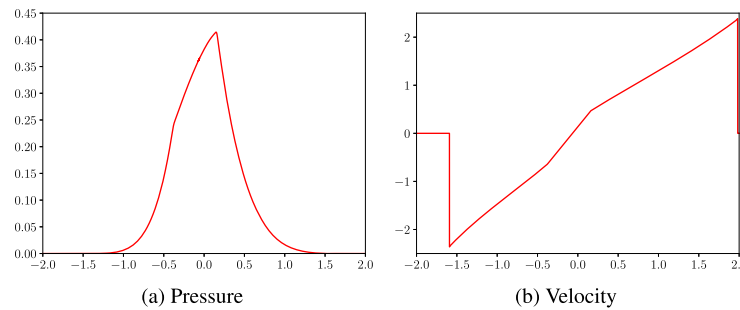


Fig. 10. Cold blocks collision problem, $t = 2.0$ – the right end of the right block has reached the right wall.

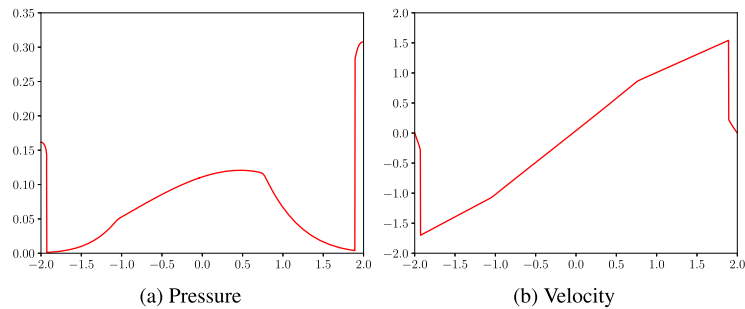


Fig. 11. Cold blocks collision problem, $t = 2.5$ – the void is completely closed in the entire domain and the reflection at both walls is visible.

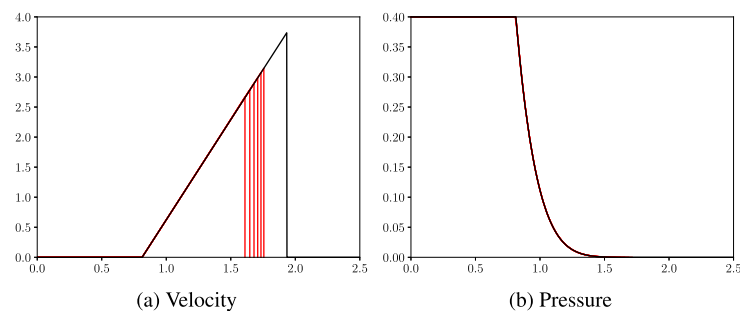


Fig. 12. Expansion in vacuum – exact solution (–) and the Lagrangian calculation with free boundary (–) using 1000, 2000, 4000, 8000, 16000, 32000 cells (higher resolutions produce a faster boundary), $t = 0.25$.

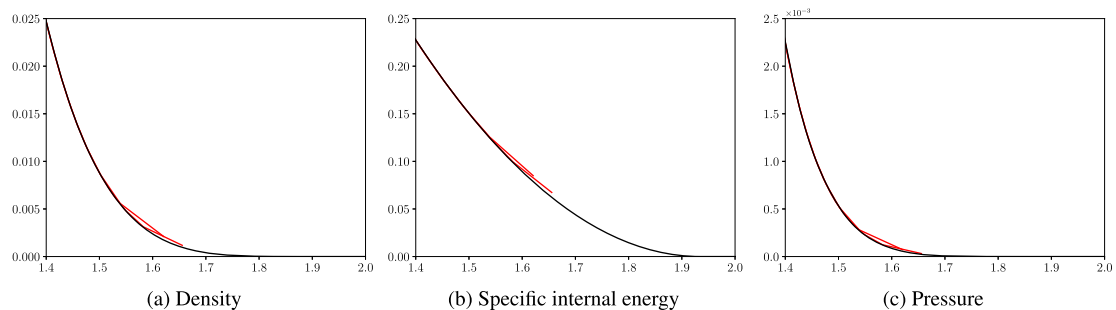


Fig. 13. Zoom of the density, specific internal energy and pressure profiles near the free boundary. Two spatial resolutions – the Lagrangian calculation using 4000 and 8000 cells (–) and the exact solution (–).

computational domain is $[0, 2.5]$. The gas initially occupies the segment $[0, 1]$ as in the Lagrangian calculations with free boundary. The rest of the domain, $[1, 2.5]$, is initially filled with void. In Fig. 14 we show the velocity and pressure obtained by our new method at $t = 0.25$ for different spatial resolutions so that these pictures can be compared with pictures in Fig. 12. The number of cells is chosen so that the number of cells initially containing the real gas is the same as in the pure Lagrangian calculation. The total number of cells is then 2.5 times larger than in the Lagrangian calculations.

The position of the interface between gas and void, which corresponds to the position of the free boundary in pure Lagrangian calculations, apparently moves much faster compared to the Lagrangian calculations. It also appears that it does not converge to the exact solution and the interface is moving faster and faster each time the spatial resolution increases.

Let us now analyze convergence for specific internal energy, density, pressure. Because our method is not exactly conservative it is also important to show how the imbalance in total energy changes with increase in resolution. The most sensitive quantity is specific internal energy. We show graphs of specific internal energy at the final time, $t = 0.25$, for different mesh resolutions in the left and middle panels in Fig. 15. In the right panel of this figure we show the total energy discrepancy as function of time for the different mesh resolutions. We have also made a quantitative analysis. In Table 1 we present discrete $L1$ norms for the errors in density, specific internal energy and pressure and the imbalance of total energy

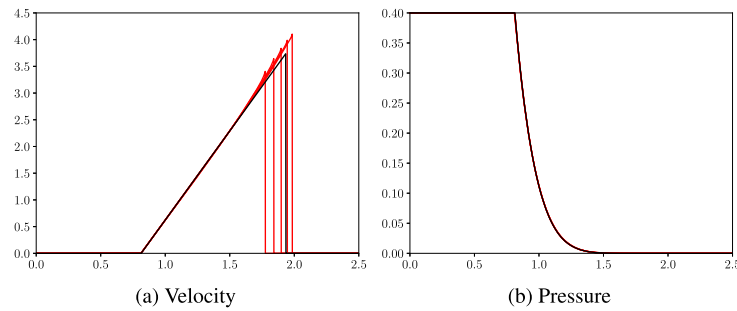


Fig. 14. Expansion in vacuum – exact solution (–) and Eulerian = Lagrange + Remap calculations with the void closure treatment (–) using 2500, 5000, 10000, 20000, 40000 cells (higher resolutions produce faster boundary), $t = 0.25$.

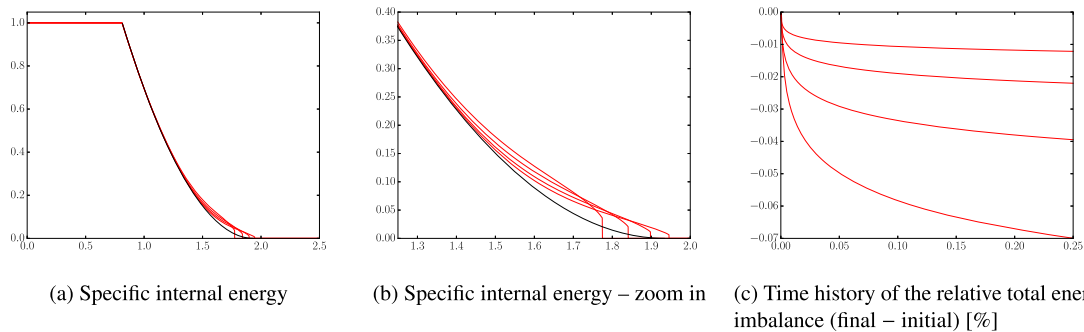


Fig. 15. Expansion in vacuum – convergence demonstration. Exact solution (–) and Eulerian = Lagrange + Remap calculations with the void closure treatment (–) using 2500, 5000, 10000, 20000 cells (higher resolutions produce faster boundary and lower total energy imbalance), $t = 0.25$.

Table 1

Expansion in to vacuum – convergence rates. Discrete L1 norms for errors in density, specific internal energy and pressure and imbalance of total energy at final time $t = 0.25$. Last row – estimated convergence rate based on two finest meshes.

Number of cells	Density	Inter. ener.	Pres.	Disb. tot. en.
2500	1.47E–3	1.66E–2	5.81E–4	6.99E–2
5000	8.62E–4	1.33E–2	3.28E–4	3.94E–2
10000	4.91E–4	1.09E–2	1.81E–4	2.20E–2
20000	2.81E–4	8.92E–3	1.001E–4	1.21E–2
Conv. rate	0.81	0.29	0.85	0.86

at the final time $t = 0.25$. In the last row of this table we give estimates for the convergence rate based on results for two finest meshes. As expected the worse convergence rate is for specific internal energy, for other quantities it is close to first order.

We give more details relating to this problem below. In Fig. 16 we present a zoom of the density, specific internal energy and pressure near the position of the interface. The values of all the parameters in the void are zero. The conclusion from Fig. 16 is similar to the one from Fig. 13. That is, clearly the numerical values are not very accurate, especially the values for the specific internal energy. However, the pressure values are very low and it is not clear how it will affect the flow if the interface collides with some obstacle.

To answer this question we have performed the following experiment. The exact solution at the time $t = 0.4009$ was computed. This is the time when the free boundary reaches the wall at $x = 2.5$ and is shown in Fig. 17.

We use this exact solution as the initial data for our Eulerian = Lagrange + Remap calculations to create a reference solution at $t = 0.8$, which corresponds to the solution at $t = 1.2009 = 0.4009 + 0.8$ starting with the position of the interface at $x = 1$. The reference solution is computed using 80000 cells. Then we compute our numerical solution in which the free boundary initially at $x = 1$ and run it till $t = 1.2009$. At this moment all voids are closed, the wave is reflected from the wall and we can compare this solution with our reference solution. This comparison is shown for various resolutions in Fig. 18. We only show the pressure profiles here.

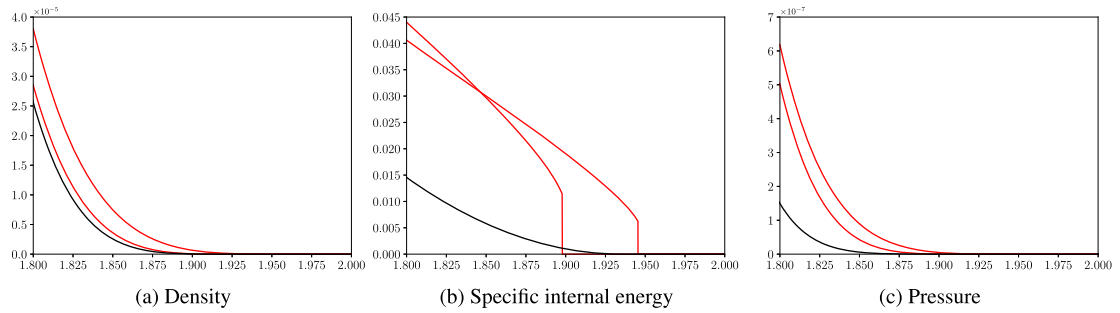


Fig. 16. Expansion in vacuum – exact solution (–) and Eulerian + Lagrange + Remap calculations with the void closure treatment (–) using 10000 and 20000 cells (higher resolution produces results closer to the exact solution and a faster boundary), $t = 0.25$. Zoom for density, specific internal energy and pressure.

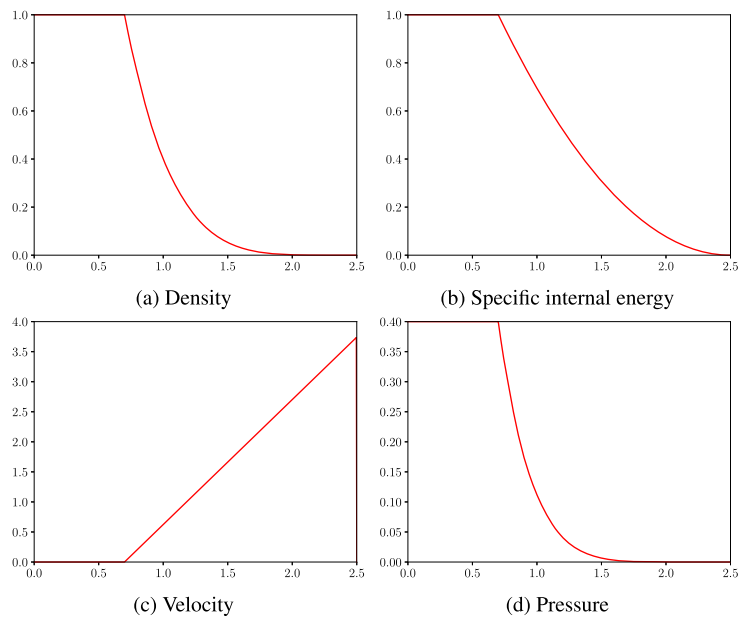


Fig. 17. Exact solution for expansion in vacuum at $t = 0.4009$, when the free boundary reaches the right boundary of the domain (wall).

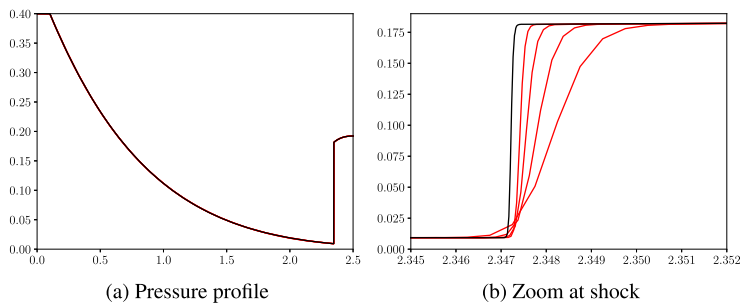


Fig. 18. Convergence after the reflection from the wall. Reference solution (–) and Eulerian + Lagrange + Remap calculations with the void closure treatment (–) using 5000, 10000, 20000, 40000 cells (higher resolutions are closer to the reference solution), $t = 1.2009$.

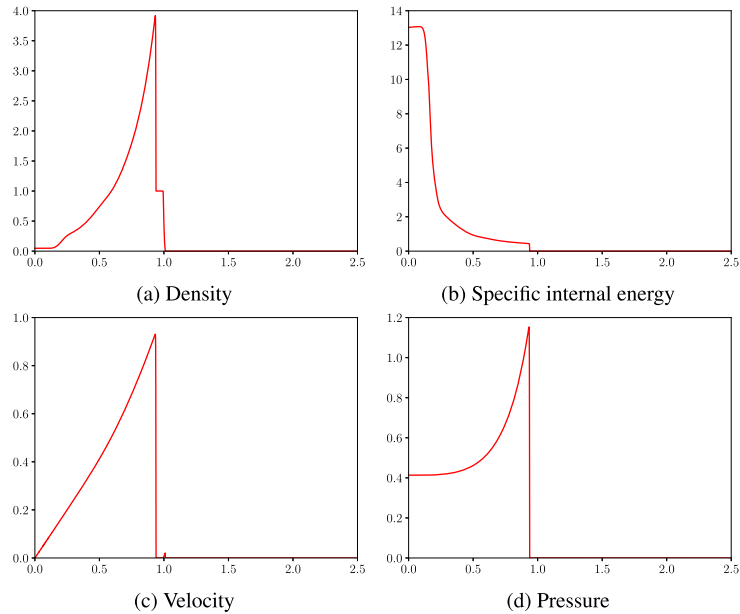


Fig. 19. Sedov-like wave formed by the finite size source before it reaches the free boundary.

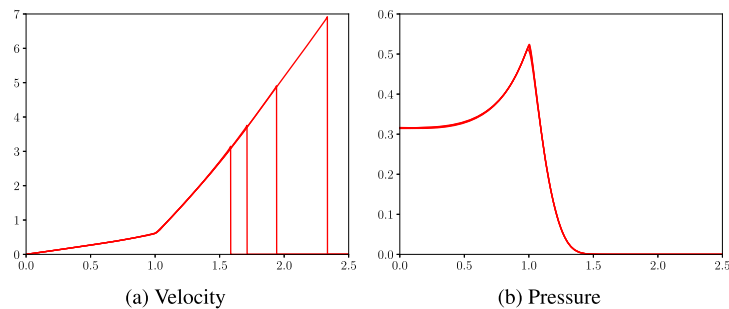


Fig. 20. Sedov-like problem – Eulerian = Lagrange + Remap calculations with void closure treatment, 2500, 5000, 10000, 20000 cells (higher resolutions create faster interfaces), $t = 0.75$.

One can only see the differences in the zoomed in picture on the right panel in Fig. 18. One can clearly see that the numerical solutions obtained using void closure treatment do converge to the reference solution and the convergence is visually close to first order.

6.1.3. Sedov-like release in to vacuum

In this section we consider the interaction of a Sedov-like shock as described in Section 5.1, with a free boundary and the reflection of the rarefaction wave that is formed with the wall. The geometry of the mesh and the boundary conditions are the same as for the expansion in to vacuum problem. The initial condition are the same as described in Section 5.1.

The initial conditions produce a Sedov-like wave which is shown in Fig. 19 for $t = 0.5$. A small velocity can be seen in the gas near the interface between the gas and the void, which is due to the initially small but non-zero specific internal energy in the gas (which creates a non-zero pressure gradient at the interface).

After the wave reaches the first cell containing void (the free boundary), the rarefaction wave moves to the left while reducing the peak of the Sedov-wave and the free boundary moves to the right, as seen in Fig. 20. We are not aware of an analytical solution for this problem.

First we run until $t = 0.75$, when the Sedov-like wave has reached the free boundary and a rarefaction wave has started to move to the left (and the free boundary has started moving to the right). In Fig. 20 and Fig. 21 we present the results using our new void closure treatment for different mesh resolutions. The results are similar to those presented in the

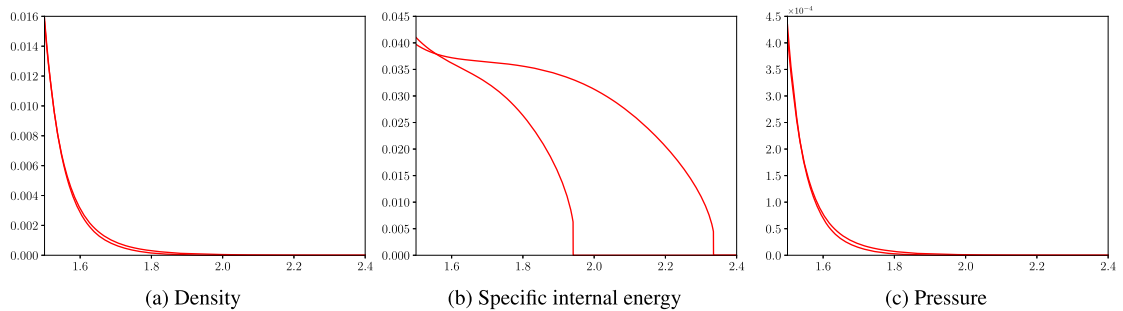


Fig. 21. Sedov-like problem – zoom of the Eulerian = Lagrange + Remap calculations with void closure treatment, 10000 and 20000 cells (higher resolutions create faster interfaces with slightly higher densities and pressures in the neighboring real gas), $t = 0.75$.

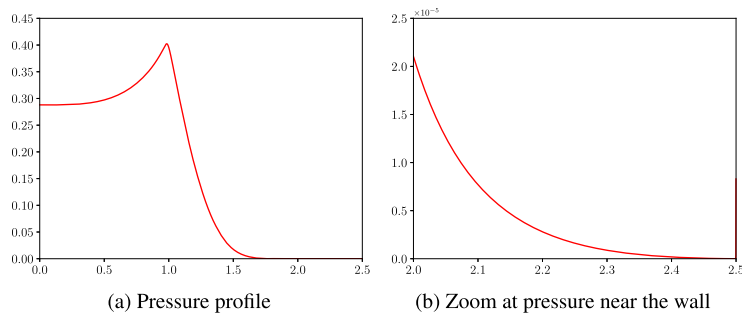


Fig. 22. Sedov-like problem – calculation with 10000 cells, at $t \approx 0.862$. All the void cells have just closed.

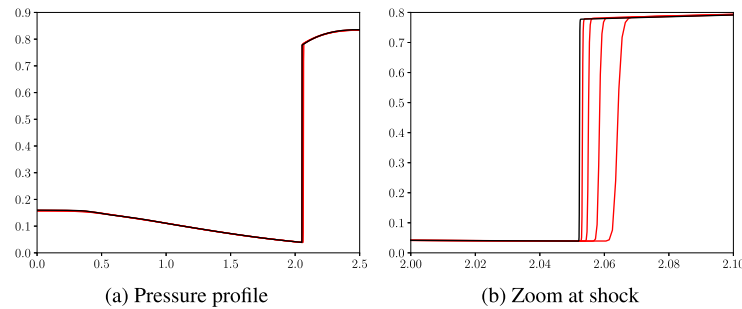


Fig. 23. Convergence after reflection from the wall, $t = 2.00$. Reference solution obtained by calculation on 40000cells (–) and solutions obtained by using the void closure treatment (–) using 2500, 5000, 10000, 20000 cells. Higher resolutions are closer to the reference.

previous section for the expansion into vacuum – there is again no convergence in the position of the free boundary, but the pressure profiles are very close and the pressure near the free boundary is very small.

In Fig. 22 we present the profile of pressure at the moment $t \approx 0.862$ for 10000 cells, when all the void sub-cells have closed and the wave is about to start to reflect from the wall. This can be compared with Fig. 17.

In Fig. 23 we demonstrate the convergence for the Sedov-like problem after the reflection from the wall at time $t = 2.00$, we present only pressure profiles.

One can clearly see that the numerical solution obtained by using the void closure treatment converges to the reference solution.

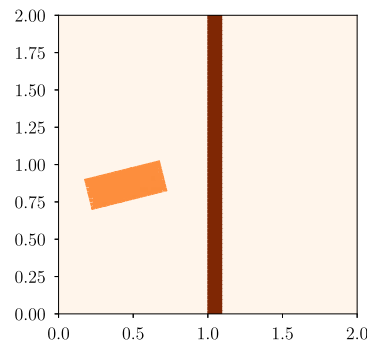


Fig. 24. Initial geometry of the projectile-plate impact test. ■ "projectile", ■ "plate", ■ void or artificial "light gas".

6.2. Two-dimensional tests

6.2.1. Diagonal projectile-plate impact

In this section we present a simplified projectile-plate impact problem in the $(x, y) \in [0 : 2] \times [0 : 2]$ computational domain. An 80×80 computational mesh is used for most of the simulations of this problem. Both the projectile and the plate are modeled as ideal gases with different properties:

- **Projectile** – $\gamma = 50$, $\rho = 20$, $p = 1 \cdot 10^{-7}$, $\mathbf{u} = (0.2, 0.05)$, initially occupies the region:

$$(x, y) \in ((0.225, 0.7), (0.725, 0.825), (0.675, 1.025), (0.175, 0.9))$$

- **Plate** – $\gamma = \frac{5}{3}$, $\rho = 15$, $p = 1 \cdot 10^{-7}$, $\mathbf{u} = (0, 0)$, initially in the region

$$(x, y) \in (1, 0) \times (1.1, 2)$$

The rest of the domain is filled with void or artificial gas – the initial geometry is shown in Fig. 24.

Using this example we will demonstrate the problems of using artificial gas to model void and show the robustness of our new closure model for voids. Let us note that for the artificial gas we use the IA-SSD closure model for multimaterial cells described in [3] with the modification for the treatment of the internal energy equation at the sub-scale stage as described in Section 4.2.2.

In this problem we can distinguish two phases. In the first phase (approx. $t \in [0, 1]$), the projectile moves freely through the void (or artificial gas). In the second phase the projectile interacts with the plate. We compare the performance of our method with a simulation using an artificial gas as a replacement for the void in both phases.

We will start by presenting the results obtained with our new method for the treatment of voids in multimaterial cells. In Fig. 25 we present results for different time moments. The left column shows the material and the interface reconstruction plots. We also present density plots in the middle column and velocity plots in the right column. At $t = 3$ the void between both materials is completely closed and no cavities remain. At $t = 4.5$, fragmentation of the projectile material begins and an instability is seen on the boundary. Then, at $t = 6$ the results are shown before the plate hits the boundary.

We now compare the results of computations using the new void treatment with three different types of artificial gas. Since we do not know what parameters to use for the artificial gas. Ideally, the artificial gas should not affect the shapes of the real materials and the simulations should be robust. We use the following initial parameters for the artificial gases:

1. $\gamma = \frac{5}{3}$, $\rho = 10^{-3}$ – high gamma/high density,
2. $\gamma = \frac{5}{3}$, $\rho = 10^{-7}$ – high gamma/low density,
3. $\gamma = 1.001$, $\rho = 10^{-7}$ – low gamma/low density.

The first type of gas has air-like properties, while the other two are artificial materials. In all cases the initial pressure is 10^{-7} both in the projectile and the plate (and in the artificial gas as well). From a theoretical point of view one may prefer artificial gas #3 – low gamma/low density – because it is very light and compressible, however, we will see that using such an artificial gas can cause robustness problems for the code.

At $t = 1$, the velocity distribution is shown in Fig. 26 for different artificial gases and void. Lower 40×40 resolution simulations are presented in this Figure to improve the readability of the velocity vector plots. We can see how the artificial gas is affected by the motion of the projectile (the shape of the projectile itself is not visibly altered). We would like to mention again that one needs to solve the full system of hydrodynamics equations for the artificial gas representing void, which can be quite expensive. In addition to the expense it also causes a time step drop in the artificial gas materials with

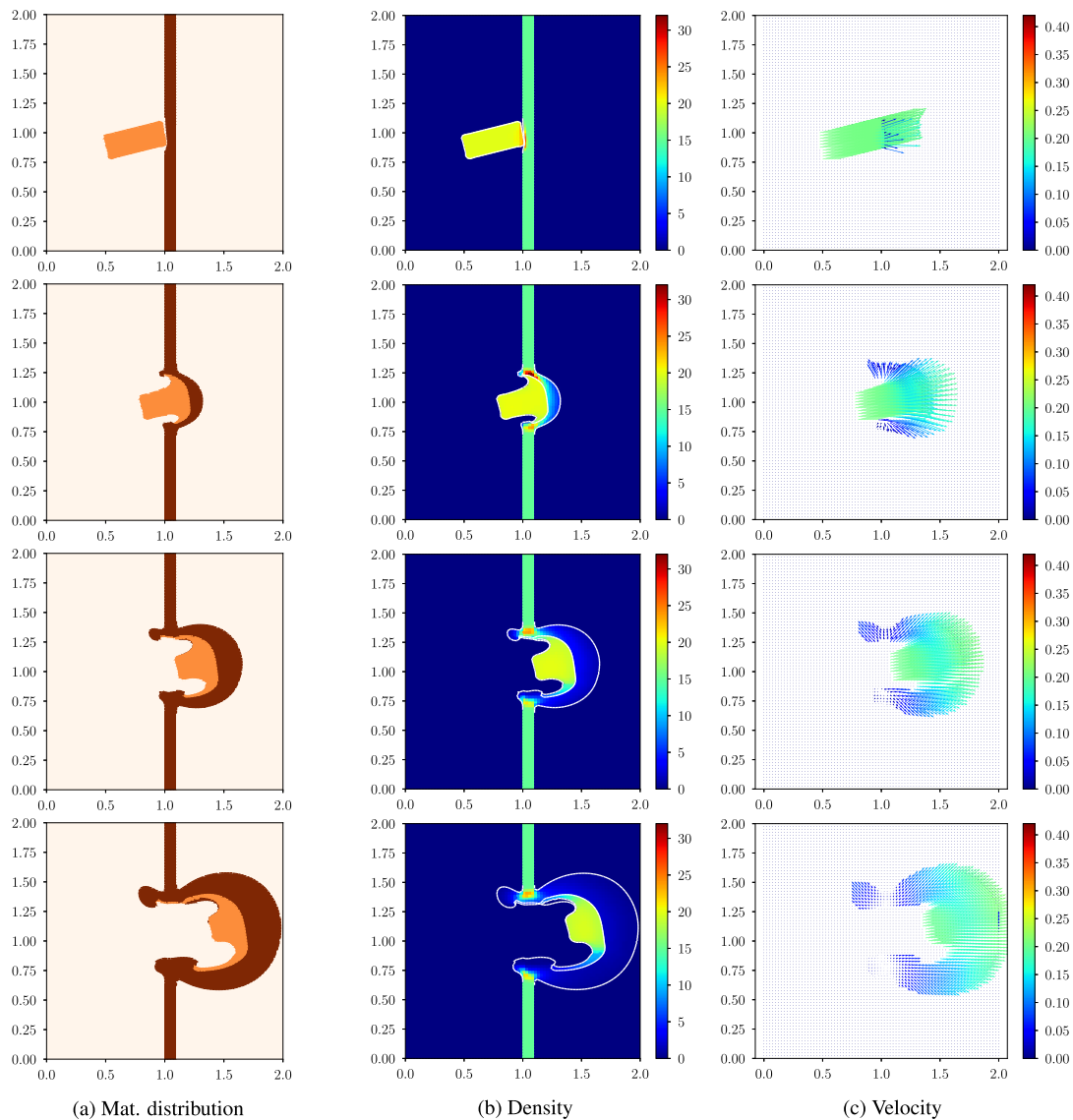


Fig. 25. Diagonal projectile-plate impact, new void treatment, 80×80 mesh, $t = 1.5, 3, 4.5, 6$ from top to bottom.

very low densities – see Fig. 27 for the time step behavior of the different artificial gases and the void treatment. One can also observe that the time step for the high gamma/high density artificial gas is even larger than in the void case. However, in this case the shapes of the real materials are visibly altered after the impact (see Fig. 28).

At $t = 3$, the artificial gas that remains between the projectile and the target are clearly visible in the left panel in Fig. 29.

At later time moments we can see that when using artificial gas the shape of the deformed projectile is slightly different than in the void case. Also the expanding plate material is propagating more slowly due to the resistance of the artificial gas. In the velocity plot, we can see that the void calculation gives a smoother velocity profile with less artifacts.

For the other two types of artificial gas the time step degenerates after the impact and the simulation fails.

The conclusion for this is that although an artificial light gas computation with reasonable accuracy is indeed possible, our example illustrates the ambiguity of choosing parameters for the artificial gas. This ambiguity is eliminated by using voids.

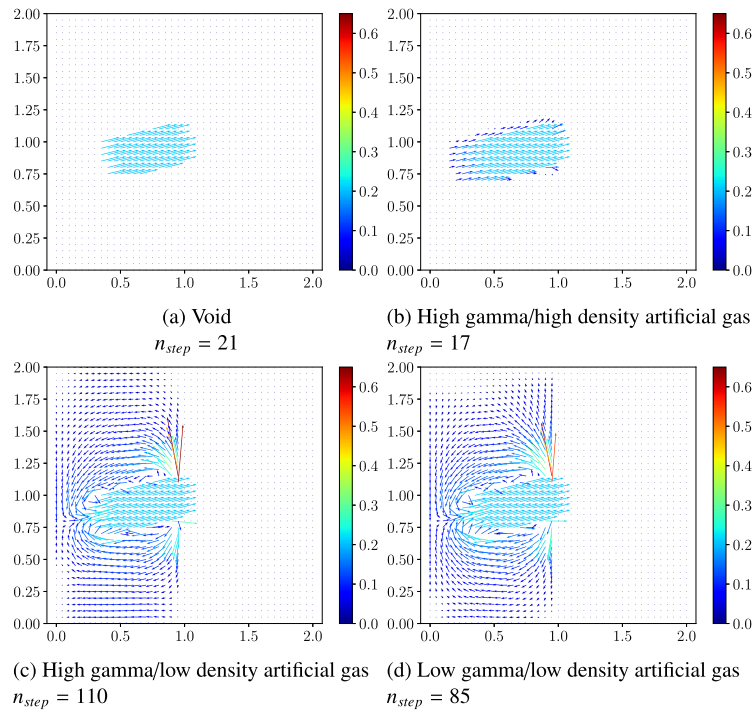


Fig. 26. Diagonal projectile-plate impact – comparison of the velocity distribution at $t = 1$ (before impact) for void and various light gas environments, 40×40 mesh.

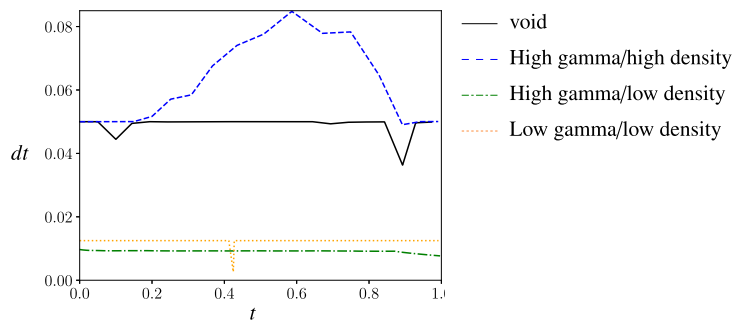


Fig. 27. Diagonal projectile-plate impact – time step comparison for void and light gas computations during the initial flight phase, 40×40 mesh.

6.2.2. Shock wave interacting with a spherical void cavity

In this test we demonstrate the ability to simulate the closure of a void cavity by a shock wave. The domain is $(-1.5, -0.5) \times (1.5, 0.5)$. All boundaries are walls. The void bubble is centered at $(0, 0)$ with a radius of $r = 0.3$. The rest of the domain is filled with two regions of ideal gas as follow:

$$\gamma = 1.4, \quad \begin{cases} \rho = 1, \quad p = 1 \cdot 10^{-5}, \quad \mathbf{u} = (0, 0) & \text{for } x > -0.5 \\ \rho = 4, \quad p = 1.3334, \quad \mathbf{u} = (1, 0) & \text{elsewhere.} \end{cases}$$

This generates a shock wave, with the front located initially at $x = -0.5$. The shock is moving to the right. The density and velocity fields are displayed in Figs. 30 and 31 – the shock front is visibly deformed after passing through the void bubble. At the time $t = 0.5$ the void is completely closed and the simulation continues in the single-material regime.

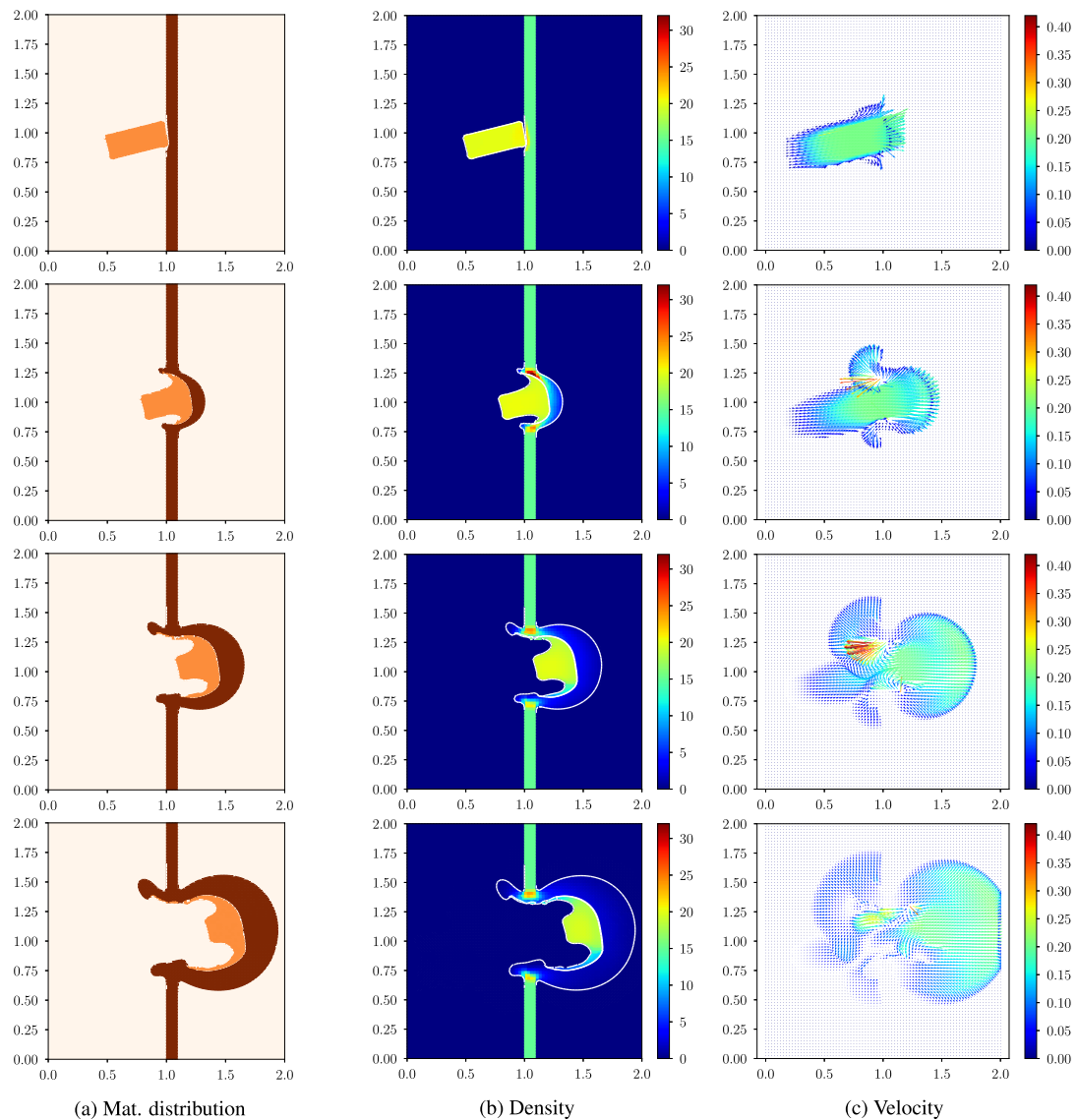


Fig. 28. Diagonal projectile-plate impact – results using high gamma/high density artificial gas for various time moments, 80×80 mesh, $t = 1.5, 3, 4.5, 6$ from top to bottom.

6.2.3. Simplified shaped-charge-like problem

This test is a simplified shaped charge problem – we call it shaped-charge-like because it exhibits some of the features of a realistic shaped charge experiment. A simple geometry and the ideal gas equation of state has been used to define the problem, so that interested researchers can easily reproduce our results.

There are two regions filled with ideal gas (see Fig. 32 for the material distribution):

- “high explosive”:

$$\gamma = 1.4, \quad \rho = 1, \quad p = 1 \cdot 10^{-7}, \quad \mathbf{u} = (0, 0),$$

- “shell”:

$$\gamma = 50, \quad \rho = 10, \quad p = 1 \cdot 10^{-7}, \quad \mathbf{u} = (0, 0).$$

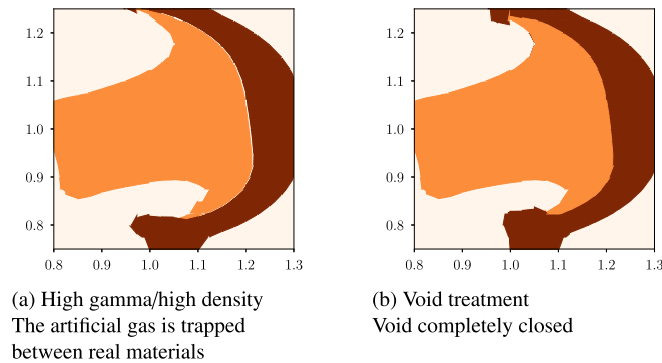


Fig. 29. Diagonal projectile-plate impact – zoom of the impact zone for 80×80 mesh at $t = 3$.

The domain occupied by “high explosive” is defined by a polygon with the following vertices:

$$(-1.0, -0.5), (2.5, -0.5), (0.0, 0.0), (2.5, 0.5), (-1.0, 0.5);$$

the shell domain is defined by polygon

$$(2.5, -0.5), (3.0, -0.5), (0.5, 0.0), (3.0, 0.5), (2.5, 0.5), (0.0, 0.0);$$

and the rest is filled with void:

$$(3.0, -0.5), (5.0, -0.5), (5.0, 0.5), (3.0, 0.5), (0.5, 0.0).$$

The boundaries of the domain are walls. Calculations are performed on 20×120 cell mesh.

To imitate the detonator we set a high internal energy, $\varepsilon = 2.5e6$, in the two cells in the “high explosive” that are located adjacent to the left boundary at its center. In Fig. 33 we show the isolines for pressure at different time moments.

In Fig. 34 we show the shape of the shaped charge (material interfaces) and a color-map for density. The corresponding velocity field is also shown in Fig. 35. We can see that a jet of material has formed which is similar to a shaped charge jet. However, since this is an artificial problem there is no experimental data to compare against.

6.2.4. Unconfined shaped charge

In this section we present our first attempt to model a real problem, the so-called unconfined shaped charge. We have loosely followed the description in [23]. The BRL [2] 105 mm unconfined shaped charge is possibly the simplest shaped charge problem available with a good set of experimental data. The geometry and materials in the experiment are shown in Fig. 36 and are taken from [23]. There are only two real materials in the experiment, a high explosive (Composition B) and the copper liner. The high explosive is detonated at the left hand end. The copper liner is then deformed and convergence of the flow in the liner on to the axis produces a jet that can travel at velocities comparable to the detonation velocity of the explosive. The simulation is performed in the two dimensional cylindrical (rz) geometry, using programmed burn to model the detonation of the explosive and with material strength included in the copper liner. All the space around the explosive and liner material is treated as void. An ALE mesh motion strategy is used to increase the resolution in the liner material by applying a weighted equipotential mesh movement strategy. Two simulations have been performed using different multimaterial cell closure models. The first was performed using the new void closure capability described in this paper. The second was performed using the equal volumetric strain closure model. The intention being to use the two simulations to give an indication of the sensitivity of the shaped charge simulation to the details of the closure model. The initial geometry for the simulation is shown in Fig. 37. We do not describe the exact specifications for the equations of state of the materials used because there is no intention to make a quantitative comparison with the experimental data in this paper.

In Fig. 38 we present the results of our simulation using the equal volumetric strain closure model and in Fig. 40 with the new void closure capability for different time moments. A zoom in showing the jet tip and mesh resolution through the jet tip for the equal volumetric strain simulation is given in Fig. 39 and for the new void closure capability in Fig. 41. The jet tip produced by the new void closure treatment is flatter and blunter than that obtained with equal volumetric strain.

The numerical results are presented only to show that the void closure algorithm described in this paper can be applied to real problems with real materials, including solids. However, we would like to mention that both the sets of results we present are in reasonable agreement with the results presented in [23]. The main differences observed between these three sets of results is in the shape of the jet tip. The new void closure model we have described in this paper produces the most blunt jet tip and Pagosa code, [23], simulation produces the sharpest jet tip, while the equal volumetric strain simulation

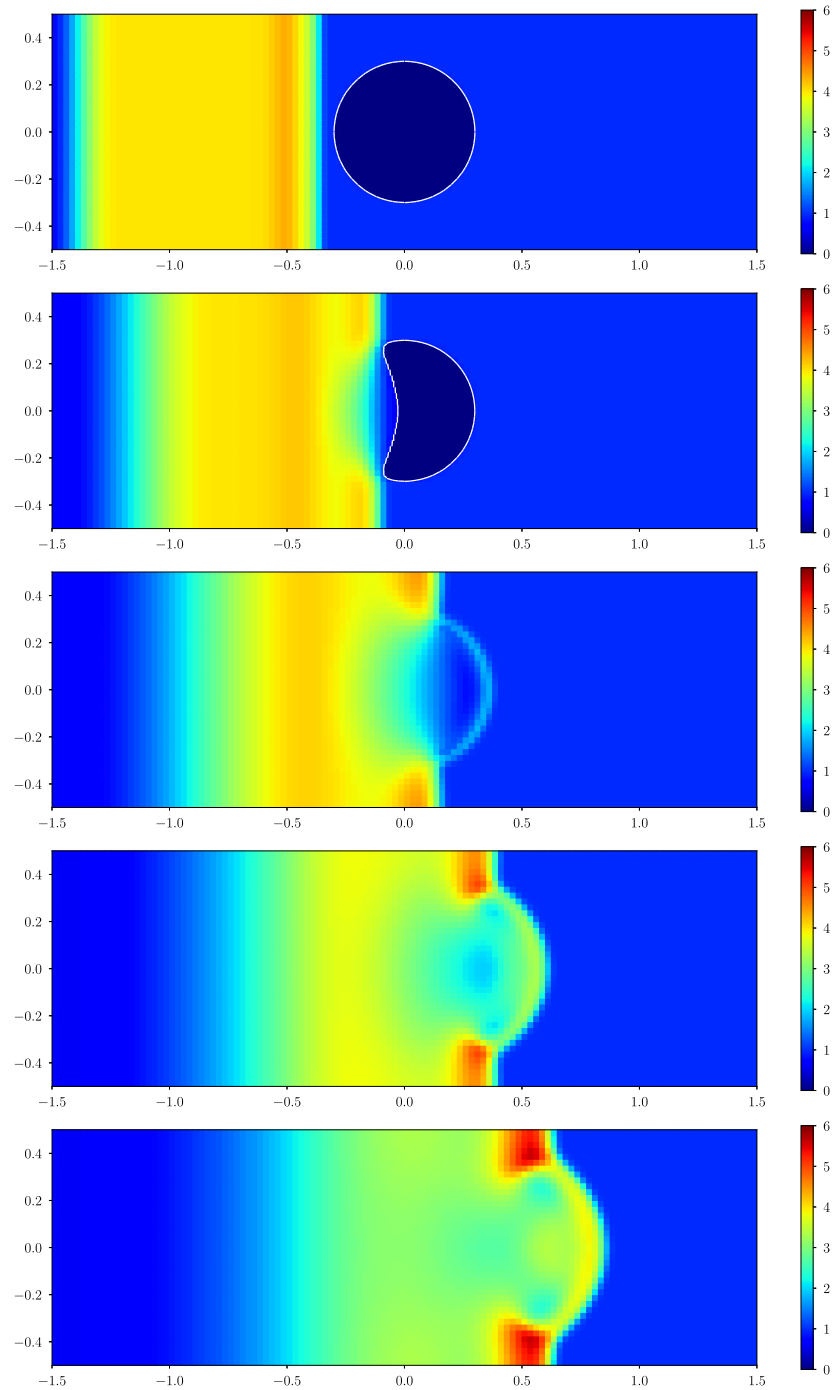


Fig. 30. Shock wave interacting with a spherical void cavity, density profile, 120×40 mesh, $t = 0.1, 0.3, 0.5, 0.7, 0.9$ from top to bottom. On the top two pictures we also show boundary of the void bubble.

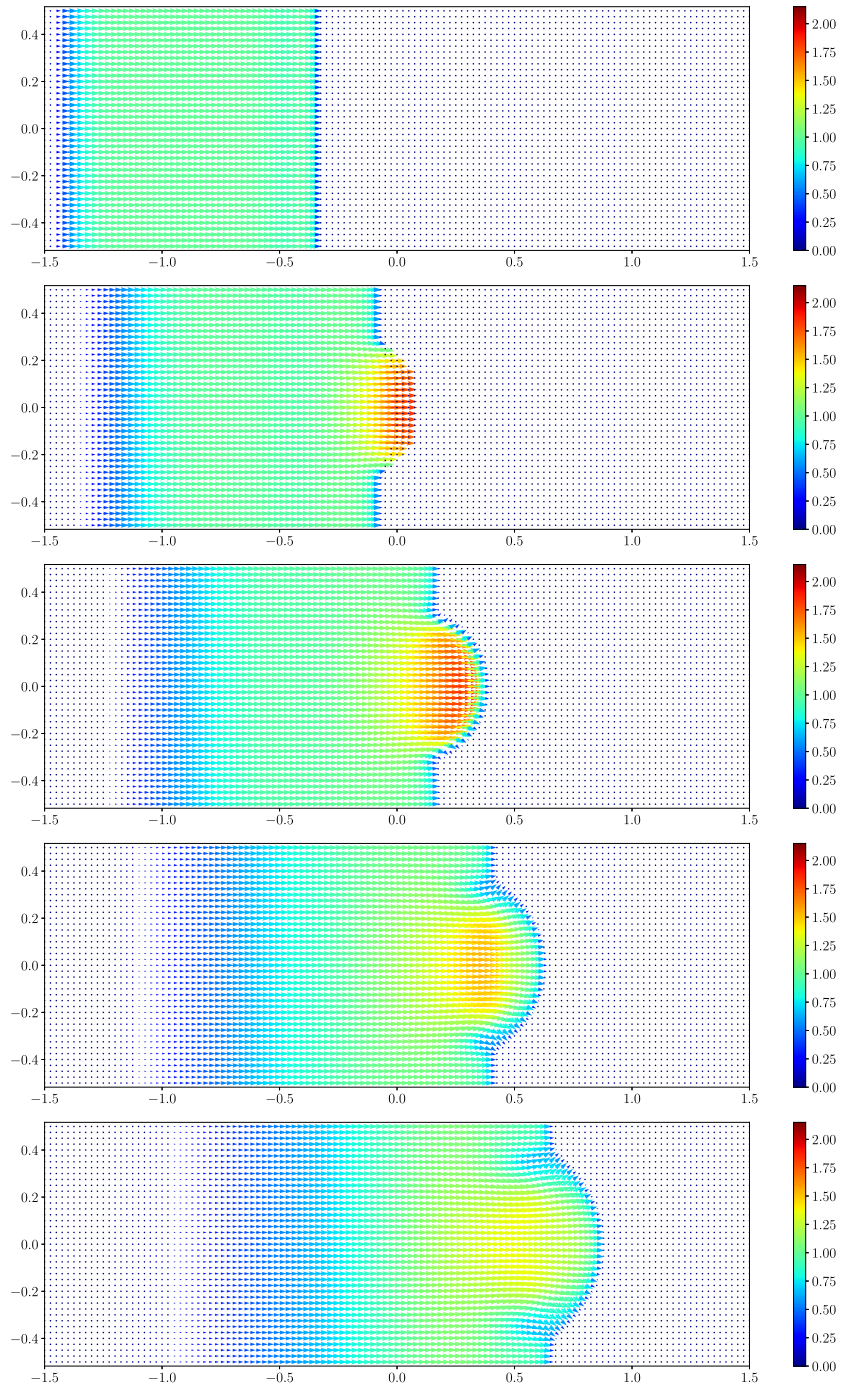


Fig. 31. Shock wave interacting with a spherical void cavity, velocity field, 120×40 mesh, $t = 0.1, 0.3, 0.5, 0.7, 0.9$ from top to bottom.

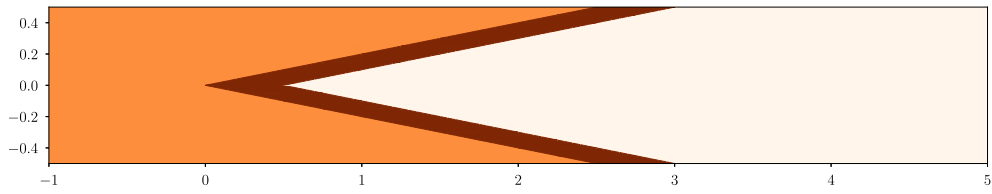


Fig. 32. Simplified shaped charge problem, initial material distribution. ■ "high explosive", ■ "shell", ■ void. 120×20 mesh, $t = 0$.

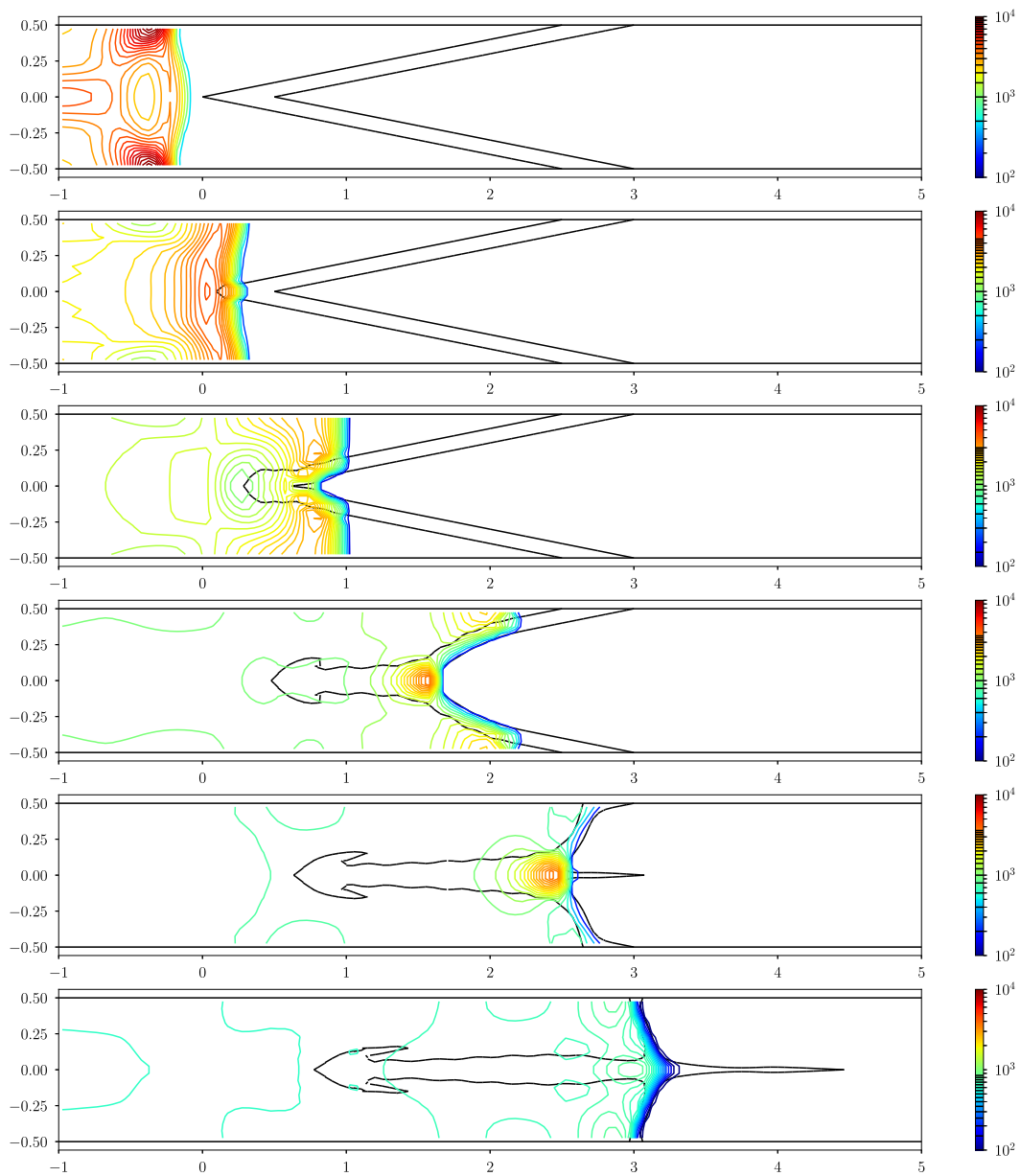


Fig. 33. Simplified shaped-charge-like problem, isolines of pressure, $t = 5, 10, 20, 40, 60, 80$ from top to bottom.

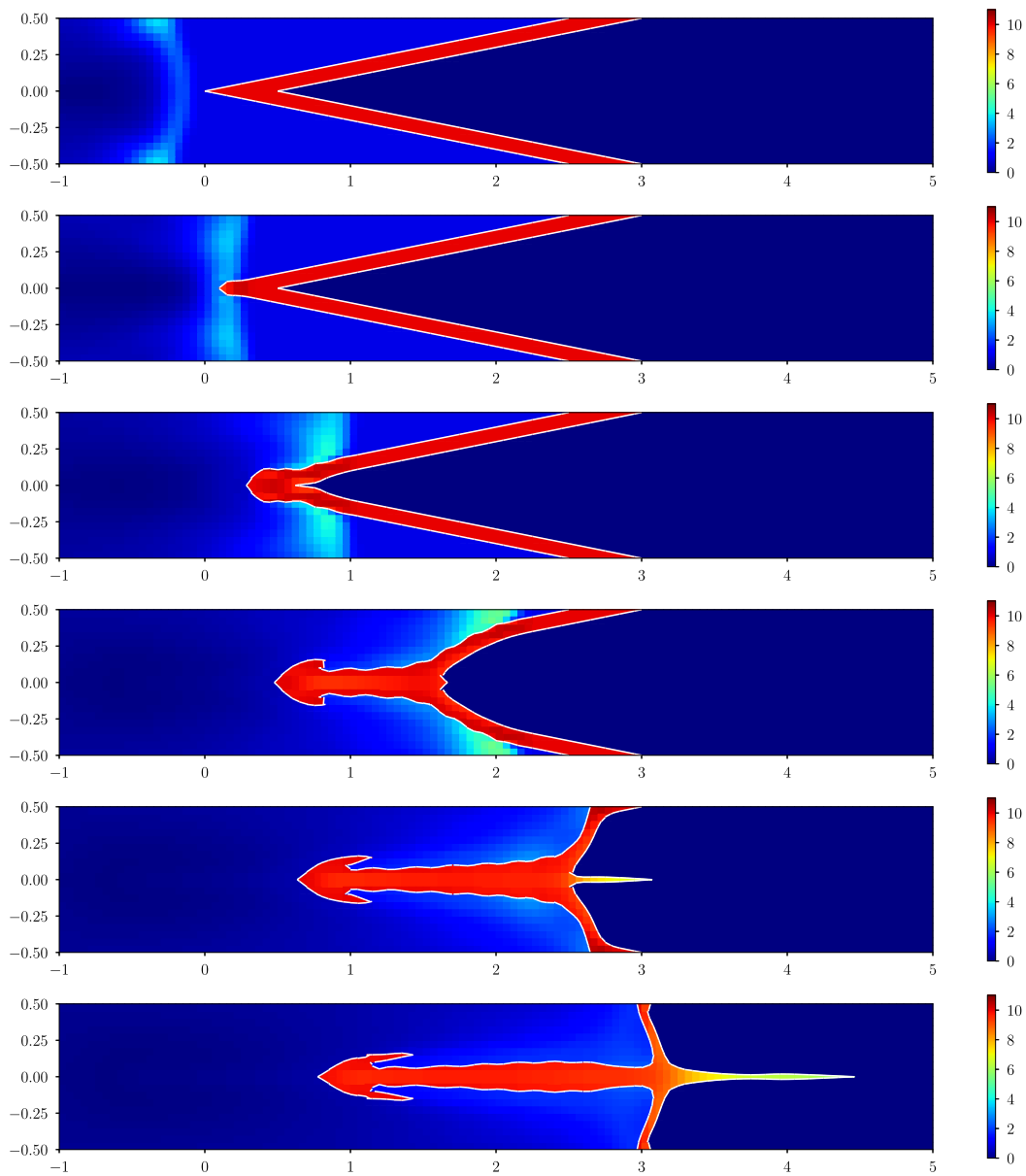


Fig. 34. Simplified shaped-charge-like problem, material interfaces and a density color-map, $t = 5, 10, 20, 40, 60, 80$ from top to bottom.

lies somewhere between the other two. This suggests the structure of the shaped charge jet tip is sensitive to the details of the multimaterial cell closure model used. A blunter jet tip such as that produced by the new void closure model is considered to be more physically justifiable and what is usually observed experimentally as can be seen in [25].

7. Conclusions

In this paper we have extended the constrained optimization-based framework for the interface-aware sub-scale dynamics [3] closure model to support multimaterial cells containing void with the ability to close.

The IA-SSD closure model for the Lagrangian stage of ALE consists of two stages. At the first, *bulk*, stage we use the equal compressibility model. Void cannot close at this stage. During the second, *sub-scale*, stage, the interactions of the materials

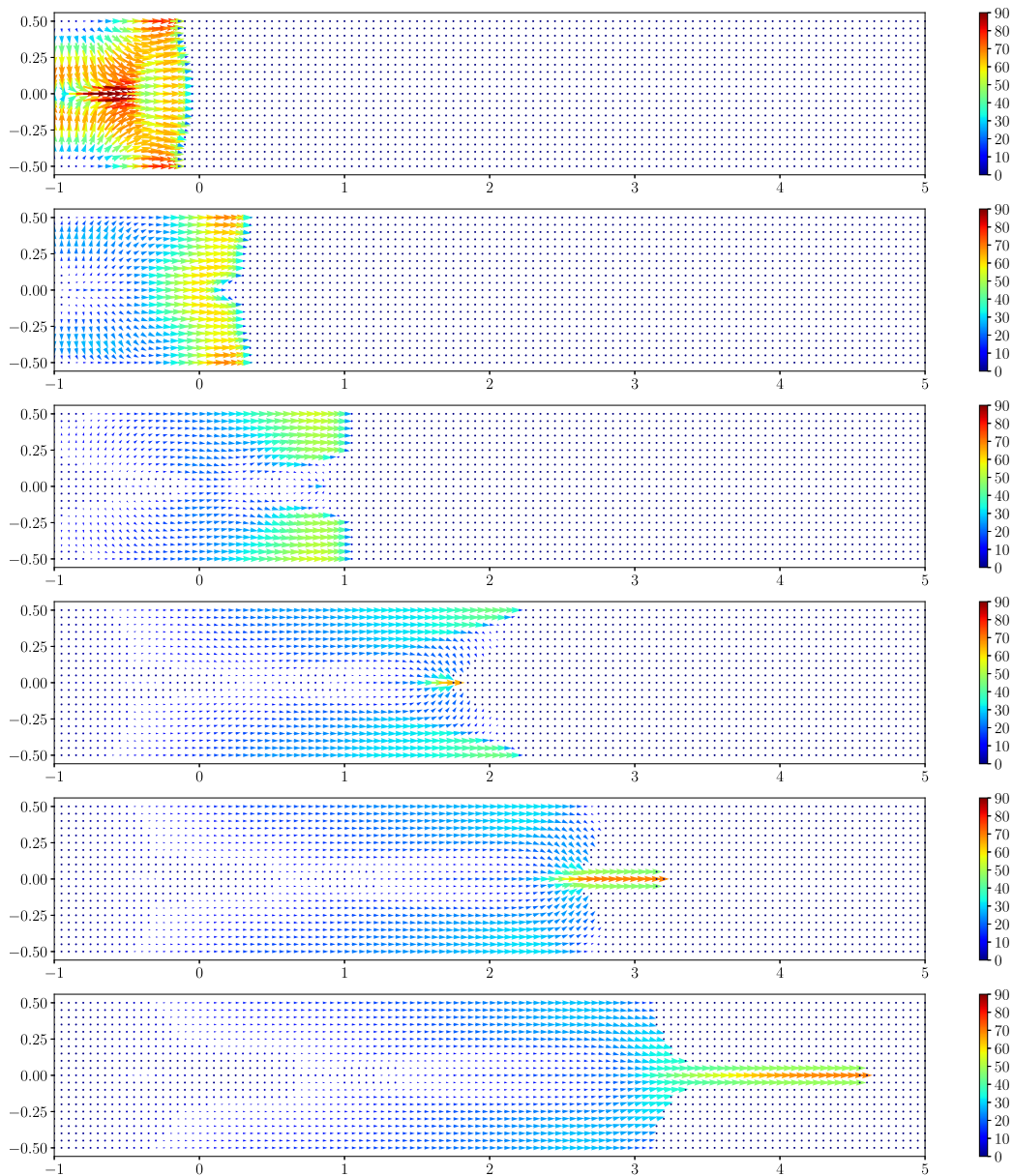


Fig. 35. Simplified shaped-charge-like problem, velocity field, $t = 5, 10, 20, 40, 60, 80$ from top to bottom.

inside the multimaterial cell are taken into account. Each material including void interacts in a pair-wise fashion with the materials with which it has a common boundary.

The pair-wise interactions of the real materials and void are based on the solution of the acoustic Riemann problem between each pair and are limited using physically justified constraints. The constraints applied are: positivity of volume for real materials, non-negativity of the volume of void, positivity of internal energy and controlled rate of pressure relaxation. The volume of the void can be zero as a result of the sub-scale stage, which indicates that the void has closed. To determine the values of the limiter coefficients, a constrained-optimization framework is employed using a quadratic objective function with linear constraints.

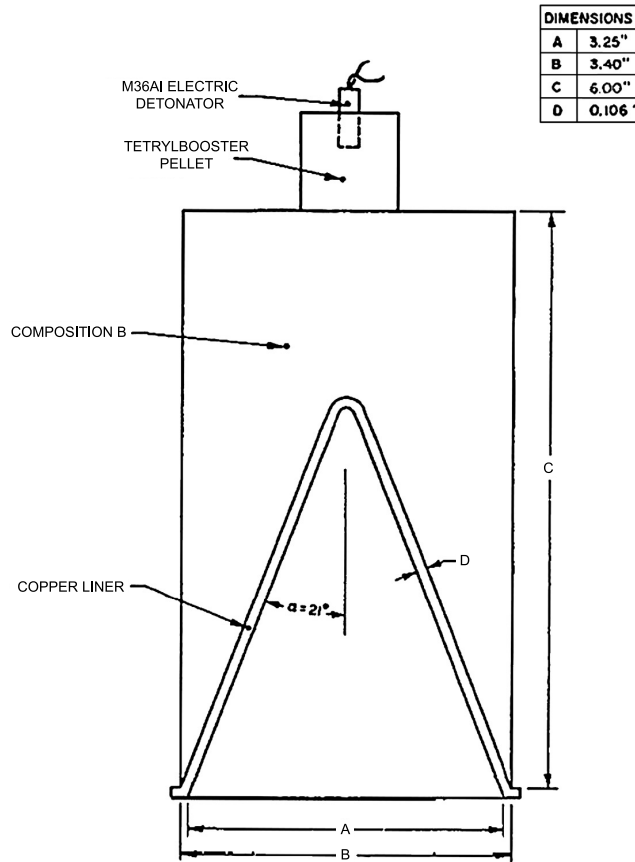


Fig. 36. Drawing of the BRL 105 mm unconfined shaped charge – copied from [23].

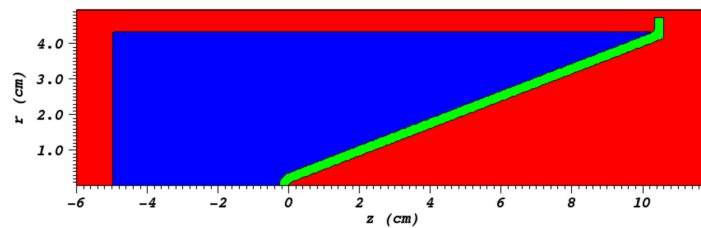


Fig. 37. Initial configuration for the shaped charge problem. ■ high-explosive, ■ copper liner, ■ void.

We have demonstrated that the new void closure model can be successfully used to simulate problems including; two-body impacts, collapse of a cavity containing a vacuum and shock-driven expansion into vacuum. Moreover, the new method eliminates the ambiguity of choosing parameters for an artificial gas material used to mock up a vacuum, which can often lead to a compromise between robustness, efficiency and accuracy. This is shown on various 1D and 2D numerical examples.

In the future we plan to describe our algorithm for the case of solid-void interactions and present a quantitative comparison of our numerical results with experimental data for problems like shaped-charges. We also plan to introduce a procedure for void opening and explore phenomena such as multimaterial sliding and friction.

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

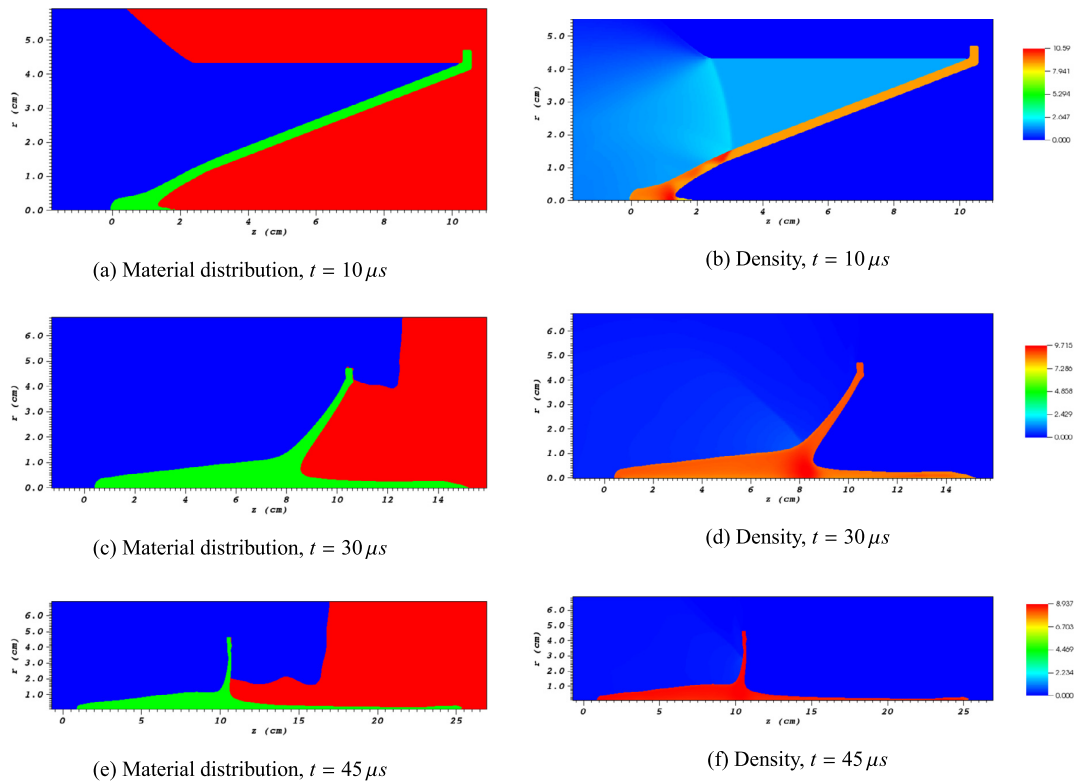


Fig. 38. Unconfined BRL shaped-charge example, equal volumetric strain model.

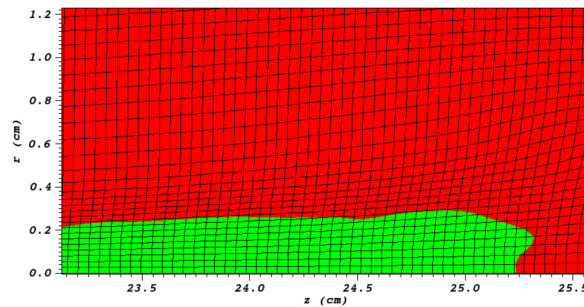


Fig. 39. Unconfined BRL shaped-charge example, equal volumetric strain model. Material distribution, zoom at the tip shape, $t = 45 \mu s$.

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 work was partially supported by the Czech Technical University grant SGS16/247/OHK4/3T/14, the Czech Science Foundation project 18-20962S and the Czech Ministry of Education project RVO68407700.

The authors thank D. Burton, A. Harrison, J. Fung, T. Canfield, M. Francois, D. Youngs, M. Kucharik, R. Hill, M. Dumbser, N. Morgan, R. Liska, D. Benson, Y. Yanilkin, Y. Bondarenko, B. Wendroff, J. Grove, B. Despres, R. Loubere, P. Vachal, T. Masser, E. Dendy, B. Rider, D. Miller, R. Abgrall, for many useful discussion over the years.

Special thanks to E. Kikinzon for allowing us to use X-MOF interface reconstruction code.

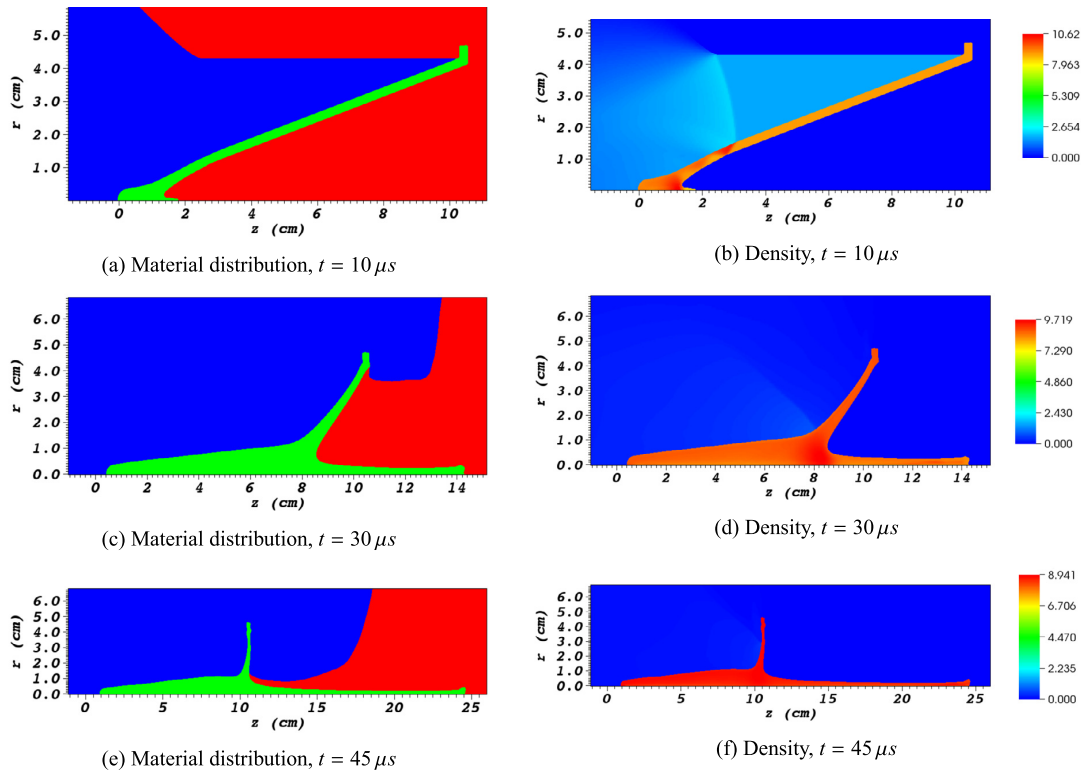


Fig. 40. Unconfined BRL shaped-charge example, IASSD model with void closure.

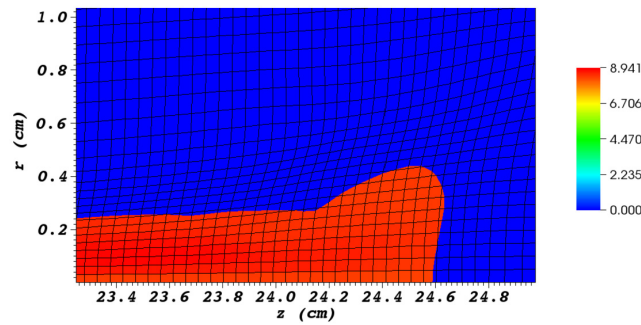


Fig. 41. Unconfined BRL shaped-charge example, IASSD model with void closure. Density, zoom at the tip shape, $t = 45 \mu s$.

References

- [1] H. Ahn, M. Shashkov, Multi-material interface reconstruction on generalized polyhedral meshes, *J. Comput. Phys.* 226 (2007) 2096–2132.
- [2] F. Allison, R. Vitali, An Application of the Jet Formation Theory to a 105 mm Shaped Charge, Report No. 1165, Ballistics Research Laboratory, March 1962.
- [3] A. Barlow, R. Hill, M. Shashkov, Constrained optimization framework for interface-aware sub-scale dynamics closure model for multimaterial cells in Lagrangian and arbitrary Lagrangian Eulerian hydrodynamics, *J. Comput. Phys.* 276 (2014) 92–135.
- [4] A.J. Barlow, P.-H. Maire, W.J. Rider, R.N. Rieben, M.J. Shashkov, Arbitrary Lagrangian–Eulerian methods for modeling high-speed compressible multimaterial flows, *J. Comput. Phys.* 322 (C) (2016) 603–665.
- [5] A. Barlow, M. Klima, M. Shashkov, Constrained Optimization Framework for Interface-Aware Sub-Scale Dynamics Closure Models with Voids for Multimaterial Cells in Lagrangian Hydrodynamics, Technical Report of Los Alamos National Laboratory, LA-UR-17-28814. Available at <https://www.researchgate.net/publication/321136813>.
- [6] D. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, *Comput. Methods Appl. Mech. Eng.* 99 (2–3) (1992) 235–394.

- [7] 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.
- [8] R.A. Carson, O. Sahni, Effect of channel geometry on blast overpressure attenuation for a large caliber cannon, in: R.G. Amesa, R.D. Boeka (Eds.), *Proceedings of the 28th International Symposium on Ballistics*, Atlanta, Georgia, USA, 22–26 September, 2014, vol. 2, DEStech Publications, Inc., 2014, pp. 1711–1721.
- [9] R.B. DeBar, *Fundamentals of the KRAKEN Code*, Report of Lawrence Livermore Laboratory, UCID-17366, 1974, <https://www.osti.gov/scitech/biblio/7227630>.
- [10] V. Dyadechko, M. Shashkov, Reconstruction of multi-material interfaces from moment data, *J. Comput. Phys.* 227 (2008) 5361–5384.
- [11] C. Hirt, A. Amsden, J.J.L. Cook, An arbitrary Lagrangian–Eulerian computing method for all flow speeds, *J. Comput. Phys.* 14 (3) (1974) 227–253.
- [12] X.Y. Hu, B.C. Khoo, Kinetic energy fix for low internal energy flows, *J. Comput. Phys.* 193 (2003) 243–259.
- [13] E. Kikinzon, Yu. Kuznetsov, K. Lipnikov, M. Shashkov, Approximate static condensation algorithm for solving multi-material diffusion problems on meshes non-aligned with material interfaces, *J. Comput. Phys.* 347 (2017) 416–436.
- [14] M. Kuchafik, M. Shashkov, Conservative multi-material remap for staggered multi-material arbitrary Lagrangian–Eulerian methods, *J. Comput. Phys.* 258 (2014) 268–304.
- [15] C.-D. Munz, A tracking method for gas flow into vacuum based on the vacuum Riemann problem, *Math. Methods Appl. Sci.* 17 (1994) 597–612.
- [16] J. Nocedal, S. Wright, *Numerical Optimization*, Springer, 1999.
- [17] W.J. Rider, Void’s unphysical response in hydrocodes, Presentation at Nuclear Explosives Code Development Conference held October 20–24, 2014 in Los Alamos, NM, SAND2014-18762PE. Available at <https://www.osti.gov/scitech/biblio/1241862>.
- [18] W. Rider, A. Robinson et al. Algorithm developments in ALEGRA guided by testing, Presentation at the Conference on Numerical Methods for Multi-Material Fluid Flows, Czech Technical University, Prague, Czech Republic, September 10–14, 2007. Available at www.cs.sandia.gov/ALEGRA/Presentations/Prague2007AlegraAlgorithmTesting.pdf.
- [19] A.C. Robinson, O.E. Strack et al., ALEGRA Update: Modernization and Resilience Progress, Report of Sandia National Laboratory SAND2013-7960C, 477209, <https://www.osti.gov/scitech/biblio/1115141>.
- [20] A.C. Robinson, J.H.J. Niederhaus, V.G. Weirs, E. Love, Arbitrary Lagrangian–Eulerian 3D ideal MHD algorithms, *Int. J. Numer. Methods Fluids* 65 (2011) 1438–1450.
- [21] K. Schittowski, QI: a Fortran Code for Convex Quadratic Programming – User’s Guide, Tech. rep., University of Bayreuth, Dept. of Computer Science, 2011, www.klaus-schittkowski.de/software.htm.
- [22] E.F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Practical Introduction, second edition, Springer, 1999.
- [23] W.N. Weseloh, PAGOSA Sample Problem: Unconfined Shaped Charge, LA-UR-16-20590, 2016.
- [24] J. Zukas, *Introduction to Hydrocodes*, Stud. Appl. Mech., vol. 49, Elsevier, 2004.
- [25] J. Zukas, W.P. Walters, *Fundamentals of Shaped Charges*, Wiley Interscience, 1989.



Appendix B

Original Research Article: An interface-aware sub-scale dynamics multi-material cell model for solids with void closure and opening at all speeds

Authors: Matej Klima, Andrew Barlow, Milan Kucharik, Mikhail Shashkov
Published in *Computers & Fluids*. 2020; vol. 208, 104578.



An interface-aware sub-scale dynamics multi-material cell model for solids with void closure and opening at all speeds



Matej Klima^{b,*}, Andrew Barlow^{a,1}, Milan Kucharik^b, Mikhail Shashkov^c

^a Computational Physics Group, AWE Aldermaston, Reading, Berkshire, RG7 4PR, UK

^b Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Břehova 7, Praha 1, 115 19, Czech Republic

^c X-Computational Physics, XCP-4, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

ARTICLE INFO

Article history:

Received 29 December 2019

Revised 18 March 2020

Accepted 7 May 2020

Available online 16 May 2020

Keywords:

Lagrangian hydrodynamics

Solid mechanics

Multi-material flows

Interface-aware sub-scale dynamics closure model

Void closure

Void opening

ABSTRACT

We present a multi-material cell model (closure model) for demanding arbitrary Lagrangian-Eulerian (ALE) simulations of fluids and solids. It is based on the interface-aware sub-scale dynamics (IASSD) approach which utilizes the exact material interface geometry within the computational cell to calculate internal material interactions. Our formulation of the closure model also aims to improve the accuracy in low-speed impact events.

Voids are used to represent ambient vacuum and internal free boundaries of the distinct materials. Void regions can close and open at contact surfaces, allowing a transition from contact physics to free motion in vacuum.

The coupling of void closure and opening with a new formulation of the IASSD model for solids is tested on several one- and two-dimensional numerical examples, ranging from gas expansion in vacuum to planar and round object impacts at various speeds.

© 2020 Published by Elsevier Ltd.

1. Introduction

Multi-material Arbitrary Lagrangian-Eulerian (ALE) simulations of fluid flows represent a state-of-the-art approach in current CFD. These types of methods typically employ an indirect ALE framework consisting of

- a staggered [1] or cell-centered [2] Lagrangian solver advancing the fluid quantities and the computational mesh in time;
- a mesh rezoning algorithm [3–5] improving the geometric quality of the mesh;
- a remapping method [6–9] transferring the fluid quantities conservatively from the Lagrangian to the rezoned mesh.

This construction is widely used nowadays [10] because it benefits from the moving computational mesh, naturally tracking the motion of the fluid and the waves inside, while keeping the mesh valid and preserving its high geometric quality. Many papers addressing various issues of multi-material ALE algorithms have been

published in recent years [11–15]. When multiple materials are present in a single cell (which is unavoidable when the full ALE algorithm is employed) material reconstruction [16–18] is used in order to determine the exact location of the material interfaces.

Material interaction during the Lagrangian step needs to be handled through a particular closure model² Typically, the closure model assumes the compressibility of each material and based on this assumption, the change of the material volume during one time step is computed together with the resulting average cell pressure, which is needed for the computation of velocities defining the Lagrangian motion of nodes of the computational mesh. The simplest and often used model is the equal compressibility (EC) model [11] (often also termed the “equal volumetric strain” model in literature), in which all materials change their volumes by the same relative amount. This simple approximation is known to cause robustness issues because of the error in approximating material quantities in the multi-material cell (these can lead to spurious sound speeds). Also it is not possible to introduce any meaningful representation of the interface physics.

* Corresponding author.

E-mail addresses: klimamat@jfifi.cvut.cz (M. Klima), Andy.Barlow@awe.co.uk (A. Barlow), kucharik@newton.jfifi.cvut.cz (M. Kucharik), shashkov@lanl.gov (M. Shashkov).

¹ ©British Crown Copyright 2020/AWE.

² The term *closure model* is frequently used in the literature due to the requirement of closure of the multi-material fluid equation system. In this paper we also use the synonymous term “multi-material cell model” when needed to avoid confusion with *void closure*, a physical phenomena of closing vacuum gaps.

One of the examples of an advanced multi-material cell model is the Tipton closure model [19,20], representing the pressure relaxation approach where a common pressure is used for all materials in the multi-material cell. This pressure is used also in the internal energy equation of each material and a corresponding volume change is found. The knowledge of exact interface geometry is not required for this model.

Another technique is used in the Interface-Aware Sub-Scale-Dynamics (IASSD) model [21], which splits the computation in two phases. In the first, *bulk* stage, the momentum equation is solved using average pressure and the new locations of the cell nodes are computed. A simplified material model which does not utilize exact material interface geometry (such as the equal compressibility model) is used at this stage. Then, in the *sub-scale* stage, the motion of each material interface inside the cell is treated using an approximate Riemann solver defining its velocity. This interface treatment attempts to replicate separate Lagrangian sub-zones. The final material distribution at the end of the time step results from material reconstruction, which leads to a small increase in computational expense as the reconstruction must be done also after the bulk phase of the model.

For many applications, the fluid approximation is not sufficient and a full elastic model needs to be incorporated. In the Lagrangian step, the material pressure must be replaced by the full stress tensor, and the temporal advance of its deviatoric terms must be provided. In this paper, we use the linear-elastic Wilkins model [22,23], describing the deviatoric term by a constitutive law expressing the Jaumann stress rate in terms of strain rate tensor components. The plastic yield is defined by the von Mises criterion. In the remapping step, the remap of the whole stress tensor must be done properly, without violating the interplay between the tensor components, such as the tensor invariants. Especially stress tensor limiting is the crucial issue, which has been addressed in several papers recently [24–28].

Another important feature is the treatment of vacuum in the simulation. It has been shown previously [29], that the often used technique of approximation of vacuum by a low-density gas strongly affects the simulation results. In a typical situation of two approaching solids, the low-density gas is compressed, its temperature and pressure is increased to unrealistic values and this persisting thin gas layer in between prevents any realistic contact simulation. Therefore, vacuum models (usually known as voids) have been developed to allow vacuum to be incorporated in the form of a zero-strength material more realistically, including void opening and closure.

The material-void interface resembles an internal free boundary and therefore it is possible to represent some of the contact problems discussed here with multiple computational grids coupled with the sliding line framework [30,31]. We believe that the non-diffuse interface multi-material approach with voids is better suited for impact problems because there are no limitations on the complexity of the interface geometry (including jets, fragments etc.). But unlike slide lines, our model was not designed to handle situations with strong interface shear.

In this paper, we focus on the generalization of the IASSD closure model for elastic/plastic material in the presence of voids in 2D. We follow the two-step structure of the standard IASSD model and demonstrate, that a natural extension of the model for fluids provides wrong results for low-velocity solid material interactions. We propose a modification of the first step (bulk stage) of the closure model called proportional compressibility (PC) model here, allowing full absorption of the volume change in the void material. This better describes the usual situation of materials starting their interaction after all void has been closed and the materials touch each other.

The rest of the paper is organized as follows. In Section 2, the Lagrangian solver for solids is described, employing the Wilkins model in the staggered discretization. In Section 3, the IASSD multi-material cell model is described in detail, including the PC model and its combination with the EC model via a directional factor. The treatment of void material is detailed in Section 4, describing the mechanics of void closure and opening. The behavior of the new model in the context of multi-material ALE algorithm is demonstrated in Section 5. The paper is concluded in Section 6.

2. Lagrangian formulation for solids

We solve the Euler equations in Lagrangian form with the Cauchy stress tensor σ :

$$\rho \frac{d}{dt} \left(\frac{1}{\rho} \right) - \nabla \cdot \mathbf{u} = 0, \quad (1)$$

$$\rho \frac{d}{dt} \mathbf{u} - \nabla \sigma = 0, \quad (2)$$

$$\rho \frac{d}{dt} E - \nabla \cdot (\sigma \mathbf{u}) = 0. \quad (3)$$

The system of equations is solved with a compatible hydrodynamics algorithm [1] using predictor-corrector time integration and staggered spatial discretization. The cells of the computational mesh can be further divided into polygonal sub-cells containing a single material, representing exact material partitioning of the domain. All cell-centered state quantities can be expressed either as *average* values – which means averaged over the computational cell – or material values. We will summarize the algorithm of one time step (the predictor part), for a detailed description see [21].

1. The corner forces acting on each node of the computational mesh are calculated by summing the contributions of the Cauchy stress acting on the neighboring half-edge vectors:

$$\mathbf{F}_{p,c} = l_1 \sigma_c \mathbf{n}_1 + l_2 \sigma_c \mathbf{n}_2, \quad (4)$$

which are combined to yield the total nodal force,

$$\mathbf{F}_p = \sum_{c \in N(p)} \mathbf{F}_{p,c}. \quad (5)$$

where $N(p)$ is a set containing all neighboring cells of node p . l_1, l_2 are equal to the half of the respective cell edge length and $\mathbf{n}_1, \mathbf{n}_2$ are the unit normal vectors. See Fig. 1 for details.

2. The forces are used to determine the velocities of mesh nodes using the momentum equation.

$$\mathbf{u}_p^{n+1} = \mathbf{u}_p^n + (\mathbf{F}_p + \mathbf{F}_q) \Delta t / m_p, \quad (6)$$

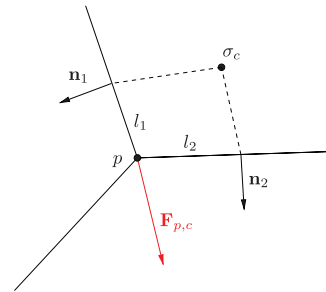


Fig. 1. Cell c to node p sub-zonal elastic force $\mathbf{F}_{p,c}$ construction with half-edge lengths l_1, l_2 , normals $\mathbf{n}_1, \mathbf{n}_2$ and the cell-centered stress tensor σ_c .

where \mathbf{F}_η are artificial viscosity forces (edge-centered artificial viscosity is used to stabilize the solution [32]) and m_p is the node-averaged mass. If the node is surrounded by cells containing void only, its mass is effectively zero and the velocity is not defined. We will consider such nodes static for the purpose of the Lagrangian step (however, nodes can and should be moved during the rezone + remap phase).

- The coordinates of mesh nodes are updated with a simple motion equation:

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + (\mathbf{u}_p^{n+1} + \mathbf{u}_p^n) \Delta t / 2, \quad (7)$$

- The volumes and average densities of all computational cells are updated.

- The closure model determines new sub-cell volume, material internal energy and centroid coordinates** (Section 3).

- An interface reconstruction algorithm is used to obtain the exact geometry of sub-cells in multi-material cells at the new time level. The Volume of fluid method is used at this stage [16].

- Material pressure is updated using the equation of state. In this paper the ideal gas equation of state is used for inviscid fluids:

$$p = \rho \epsilon (\gamma - 1), \quad (8)$$

where γ is the adiabatic gas constant (ratio of specific heats). For computations with solids the Mie-Grüneisen equation of state [33] is used in the following form:

$$p = \rho_0 c_0^2 f(\eta) + \rho_0 \Gamma_0 \epsilon, \quad \eta = \frac{\rho}{\rho_0}, \quad (9)$$

$$f(\eta) = \frac{(\eta - 1)(\eta - \frac{1}{2}\Gamma_0(\eta - 1))}{(\eta - s(\eta - 1))^2}, \quad (10)$$

where ρ_0 is the density at the reference state, c_0 the bulk speed of sound, s is the linear Hugoniot slope coefficient and Γ_0 is the Grüneisen's constant at the reference state.

- The Cauchy stress tensor is a material-based cell-centered quantity, which can be split into the isotropic pressure part and the deviatoric stress tensor:

$$\sigma = p\mathbf{I} - \mathbf{S}. \quad (11)$$

The deviatoric stress tensor is updated using the constitutive model [23]. First we determine the strain rate (D) and spin (W) tensors:

$$D = \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T), \quad W = \frac{1}{2}(\nabla \mathbf{u} - (\nabla \mathbf{u})^T). \quad (12)$$

The velocity gradient is a tensor, hence a cell-centered quantity in our scheme. It is estimated numerically for each control volume – computational cell c :

$$(\nabla \mathbf{u})_c = \frac{\int_c \nabla \mathbf{u} dV}{\int_c 1 dV} = \frac{1}{V_c} \oint \mathbf{u} \otimes \mathbf{n} dl, \quad (13)$$

where V_c is the volume of cell c . To calculate the integral over the cell boundary, velocities of the nodes are used. This is justified in [23] – even though the cell-centered representation is used, velocity vectors are recalculated at nodes.

We now express the Hooke's law using the objective Jaumann deviatoric stress rate and the deviatoric part of the strain rate:

$$\overset{\circ}{\mathbf{S}} = \frac{d\mathbf{S}}{dt} + \mathbf{S}W - W\mathbf{S} = 2\mu \left(D - \frac{1}{3} \text{tr}(D)\mathbf{I} \right), \quad (14)$$

μ is the shear modulus, a material constant. From this incremental stress-strain relationship we can determine the \mathbf{S} in each time step in the Lagrangian part of the simulation.

This model is supplemented with the von Mises yield condition describing material plasticity by limiting the magnitude of the deviatoric stress:

$$|\mathbf{S}| \leq \sqrt{\frac{2}{3}} Y^0, \quad |\mathbf{S}| = \sqrt{\text{tr}(\mathbf{S}^T \mathbf{S})}. \quad (15)$$

The constant Y^0 is the yield strength – it determines the point at which the material begins to deform plastically.

3. IASSD multi-material cell model

In this paper we will present an evolution of the IASSD two-step closure model, which was first presented in [21] for fluids and then extended to fluids in the presence of voids in [29]. Our goal is to enable impact problem simulation with the IASSD model, which requires some modification to account for solid materials with a generally non-isotropic Cauchy stress tensor. If considering also solid material interactions with voids, the possibility of tension forces on a material interface implies a treatment of void opening is required, which is detailed in the next section.

This model consists of two stages – in the first one, the *bulk stage*, we consider the movement of the whole computational cell and use a rougher approximation of internal processes, either ignoring any interaction completely, or creating a simple compressibility-based approximation of the sub-cell volume changes. This is followed by a *sub-scale stage*, which treats the pairwise interactions of material sub-cells, while the boundaries of the computational cell now considered as static. The structure of both stages will be described in the following subsections.

Because the closure model is local to a single computational cell, in this section we drop the cell index c from all variables. Subscript k represents a material quantity, while two subscripts such as k, l define a quantity on the interface of neighboring material sub-cells.

3.1. Bulk stage

We will consider a more general formulation of the bulk stage in this paper. Unlike in [21,29], we will allow the volume fraction to change during this part of the algorithm. Our numerical experiments have shown (see Section 5.3) that this approach is much better suited for solid-void interactions (and also for other cases when the compressibility of the interacting materials differs by orders of magnitude) at low speeds. Imagine two solid plates separated by void and approaching each other within one computational cell – in such case we need the void to take in ideal case all the compressibility, while the solid materials should remain undisturbed. To be able to do this, we need to review the definition of material volume changes and subsequently how the averaged quantities (pressure, stress, artificial viscosity) are calculated inside the computational cell. Following the ideas in [34], we will introduce compressibility factors:

$$\beta_k = \left| \frac{\Delta V_k^{bulk}}{\Delta V} \right|. \quad (16)$$

These compressibility factors will then serve as weights for the calculation of averaged quantities:

$$p = \sum_k \beta_k p_k, \quad S = \sum_k \beta_k S_k, \quad q = \sum_k \beta_k q_k. \quad (17)$$

3.1.1. Equal compressibility (EC) model

The equal compressibility model used in [21,29] presumes, that all volume fractions are constant during the bulk stage. This means, that the total volume change is distributed to all materials in the

same ratio as defined by the volume fractions, which are therefore equal to the compressibility factors:

$$\beta_k = \alpha_k = \frac{V_k}{V}. \quad (18)$$

This makes it compatible with our general formulation of the bulk stage as defined in (16). From our experience, the equal compressibility model inhibits correct impact simulations at relatively low speeds (when many time steps are needed for the interface to cross the computational cell). The solid bodies start interacting already before the void is closed and the shock timing can be distorted.

3.1.2. Proportional compressibility (PC) model

Assuming that during the bulk stage the sign of all material volume changes is the same as the sign of the total volume change (ie. all materials will either only compress or only expand), we can apply relation (16) inversely and compute the volume change from an approximate compressibility factor. This is the basis of what we call the ‘‘proportional compressibility’’ (PC) model:

$$\Delta V_k^{bulk,unlim} = \beta_k^{PC} \Delta V, \quad \beta_k^{PC} = \frac{\frac{\alpha_k}{K_k}}{\sum_k \frac{\alpha_k}{K_k}}, \quad (19)$$

where K_k is the bulk modulus of material k . If a computational cell contains void material, we get to an infinite compressibility limit in the β_k^{PC} parameter:

$$\beta_k^{PC} = \begin{cases} 1 & \text{if } k \text{ is a void material} \\ 0 & \text{if } k \text{ is a non-void material,} \end{cases} \quad (20)$$

resulting in the fact, that all volume is preferentially given or taken from the void sub-cell (we assume that at this stage there is never more than one void sub-cell in each computational cell).

The approximate compressibility factors β_k^{PC} are guaranteed to sum to one in each computational cell, a condition necessary for the bulk model to preserve the total cell volume. The volume change given by (19) unfortunately does not consider actual material volume bounds and if applied directly, it could lead to non-physical results. A limiter is therefore required to keep the solution within bounds – it is described further in Section 3.1.3.

3.1.3. Limiting bulk stage material volume change

The limiting condition is described by the following formula:

$$V_k^{bulk} = V_k^n + \beta_k \Delta V \geq V_k^{min.}, \quad (21)$$

where $V_k^{min.}$ represents the minimum admissible volume for a certain material sub-cell. First, it is important to note that for cases of cell expansion $\Delta V > 0$, this condition is always satisfied due to the assumption of the same signs of all material and total volume changes. For the compression of a non-void material, we allow the sub-cell volume to decrease up to a certain proportion of the original sub-cell volume in one time step, while for void sub-cells we must allow the volume to approach zero:

$$V_k^{min.} = \begin{cases} 0 & \text{if } k \text{ is a void material,} \\ \kappa_{bulk} V_k^n & \text{if } k \text{ is a non-void material.} \end{cases} \quad (22)$$

We will further show that for the equal compressibility model, the minimum volume condition is always satisfied. In our code there is a time step condition that effectively allows the cell volume to decrease to at most $\kappa_{dV} = 0.8$ of the original cell volume [1]. If we substitute this condition in (21) we obtain the following rule:

$$V_k^n - \alpha_k(1 - \kappa_{dV})V^n = \kappa_{dV}V_k^n \geq \kappa_{bulk}V_k^n \geq 0, \quad (23)$$

which implies that as long as $\kappa_{dV} \geq \kappa_{bulk}$, the equal compressibility bulk stage will never exceed the minimum bounds of the material

volume. We choose $\kappa_{bulk} = 0.75$ as a value which satisfies this condition and is not unnecessarily restrictive.

Unfortunately, for the proportional compressibility model this minimum volume condition can be violated. To summarize – we have one model which satisfies volume bounds, but is not accurate enough, and a more accurate model for which bounds preservation is not guaranteed. We can interpolate between those two in a similar way as FCT-like limiters [35,36] interpolate between low and high order solutions to get the most accurate solution while still preserving bounds. For each computational cell we will define the actual compressibility factors as convex combinations of equal and proportional compressibility factors using a limiter parameter θ :

$$\beta_k = (1 - \theta)\alpha_k + \theta \beta_k^{PC}, \quad 0 \leq \theta \leq 1. \quad (24)$$

For any value of θ , the total volume preservation is satisfied as it is satisfied for each compressibility factor alone:

$$\sum_k \beta_k = (1 - \theta) \sum_k \alpha_k + \theta \sum_k \beta_k^{PC} = 1. \quad (25)$$

We will calculate the cell-based limiter by substituting β_k definition (24) in the bulk stage material volume definition (21):

$$V_k^n + ((1 - \theta)\alpha_k + \theta \beta_k^{PC})(V^{n+1} - V^n) \geq V_k^{min.}, \quad (26)$$

further transforming the inequality specifically for the cell compression case,

$$\theta(\beta_k^{PC} - \alpha_k) \leq \frac{V_k^{min.} - V_k^n}{V^{n+1} - V^n} - \alpha_k. \quad (27)$$

Now to express the inequality for θ , we need to consider how the two parameters compare. If $\beta_k^{PC} < \alpha_k$, then the bulk stage material volume change (26) is equal to or less than material volume change as calculated by the equal compressibility model (23). And as we have shown, for such model the minimal volume condition is fulfilled, therefore it is valid also in this case and any value of θ in the $[0, 1]$ is acceptable.

In the second case when $\beta_k^{PC} > \alpha_k$ we can transform (27) to express the maximum possible value for the limiter parameter to guarantee volume lower bound preservation:

$$\theta^{lim.} \leq \frac{1}{(\beta_k^{PC} - \alpha_k)} \left(\frac{V_k^n - V_k^{min.}}{V^n - V^{n+1}} - \alpha_k \right). \quad (28)$$

3.1.4. Direction-aware hybrid model

In the two-dimensional setting, the proportional compressibility model works well when interfaces are perpendicular to the effective strain principal direction, but suffers from symmetry distortions in areas where it is aligned differently. If the strain is parallel to the interface, the equal compressibility model seems to be a more accurate description.

Therefore we can use a similar approach to blend those two models as in Section 3.1.3, but with a directional factor. If the cell deforms in the normal direction of the majority of interfaces (such as in contact cells), we prefer to use the PC model. While if the deformation is mainly parallel with the interface direction, we revert to the EC model. To achieve this we define the following target value of θ :

$$\theta^{dir.} = \frac{\sum_{k,l} |\mathbf{n}_{k,l}^T D \mathbf{n}_{k,l}| \mathbf{S}_{k,l}}{\|D\|^2 \sum_{k,l} \mathbf{S}_{k,l}}, \quad (29)$$

where $\mathbf{n}_{k,l}$ is the unit normal of the interface between material sub-polygons k, l , $\mathbf{S}_{k,l}$ is the interface length and D is the cell centered strain rate tensor. $\|D\|$ is its spectral norm, which can be easily calculated as follows:

$$\|D\| = \sqrt{\max(\lambda_1, \lambda_2)}, \quad (30)$$

$$\lambda_{1,2} = \frac{1}{2} \left(D_{xx} + D_{yy} \pm \sqrt{(D_{xx} - D_{yy})^2 + 4D_{xy}^2} \right). \quad (31)$$

This ensures that the contributions to the directional factor are interface-length weighted and always normalized to yield a number between 0 and 1.

If the strain rate tensor norm is small or even very close to zero, the directional parameter (29) loses accuracy as the denominator approaches zero. In such cases (negligible deformation of the computational cell) we want to revert to the equal compressibility model by setting θ close to zero. For such cases we need to have some measure of the strain rate magnitude, which we can derive from the strain required to reach full plastic loading from the fully unloaded state in one time step. The following limiting parameter will be equal to or greater than one if the strain magnitude exceeds one tenth of such measure:

$$\theta^{norm} = 10 \|D\| / \min_k \frac{Y_{0,k} \alpha_k}{\Delta t \mu_k \sqrt{6}}, \quad (32)$$

while for smaller strain rates, it will start to bias towards the equal compressibility model as its value will approach zero.

The final value of θ will be a minimum of the limiter defined in Section 3.1.3, the norm factor (32) and the directional factor as defined by (29):

$$\theta = \min \left(1, \theta^{lim}, \theta^{norm}, \frac{1}{2} \left(1 + \tanh(20(\theta^{dir} - \frac{1}{2})) \right) \right). \quad (33)$$

For the directional factor a hyperbolic tangent steepener function is used to avoid activating the switch only for angles which are close to perpendicular. We are aware that the setting of the parameters of such model may depend on the actual problem being solved, however, this particular configuration was used in both low and high speed impact tests presented further in this paper with satisfactory results.

3.2. Sub-scale dynamics stage

The sub-scale stage takes into account pair-wise interactions between the material sub-polygons inside the multi-material cell (it is assumed that the exact geometry is known from the interface reconstruction algorithm). The boundaries of the computational cell are fixed in this part of the algorithm. The overview of an example sub-cell geometry is shown in Fig. 2. It also shows approximations of the volume fluxes $\delta V_{k,l}$ between material 1 and its neighbors.

The basic principles of the sub-scale stage are presented in [21,29] and we refer the interested reader there for the detailed description of the constrained optimized limiter implementation and

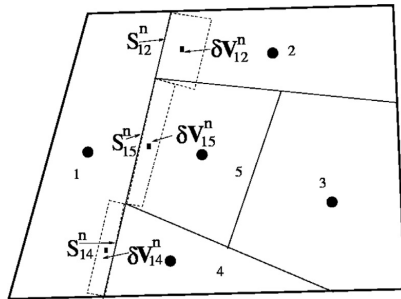


Fig. 2. Multi-material cell: an example of the sub-cell geometry and relative volume exchange approximation.

other considerations. We will further describe the modification of such algorithm to allow for representation of solid materials with strength model. We found that we need to replace pressure equilibration with normal stress equilibration.

Each pair-wise interaction generates a volume exchange between the two materials that we define as follows:

$$\Delta V_k = \Delta V_k^{bulk} + \sum_{l \in M(k)} \delta V_{k,l}, \quad \delta V_{k,l} = \Psi_{k,l} \delta V_{k,l}^{max}, \quad (34)$$

where $\delta V_{k,l}^{max}$ is the volume exchange calculated using the interface velocity obtained from the approximate Riemann solver. $0 \leq \Psi_{k,l} \leq 1$ are the limiter coefficients chosen so that the resulting material distribution does not violate physically justified constraints. The set of the materials in the multi-material cell which have a face in common with material k is denoted by $M(k)$. Unlike in [21,29], the interaction is driven not only by pressure difference, but also by the resulting interface stress tensor. In our approximation we neglect interface shear motion and use the normal component of the interface stress only. We approximate its value after the bulk stage:

$$\tilde{\sigma}_{k,l}^{bulk,n+1} = p_k^n - \mathbf{n}_{k,l} S_k^n \mathbf{n}_{k,l} - \frac{\rho_k^n (c_k^n)^2}{V_k^n} \Delta V_k^{bulk,n+1}, \quad (35)$$

where $\mathbf{n}_{k,l}$ is the unit normal vector on the interface between materials k and l and S_k^n is the material deviatoric stress tensor. Note that $\tilde{\sigma}_{k,l}^{bulk,n+1}$ determines the normal interface stress in material k in the normal direction of its interface with material l and therefore $\tilde{\sigma}_{k,l}^{bulk,n+1} \neq \tilde{\sigma}_{l,k}^{bulk,n+1}$. The material normal stresses are used to compute interface velocity:

$$u_{k,l}^* = \frac{\tilde{\sigma}_{l,k}^{bulk,n+1} - \tilde{\sigma}_{k,l}^{bulk,n+1}}{\rho_l c_l + \rho_k c_k}, \quad (36)$$

where c_l, c_k are the respective material sound speeds corrected for the elastic wave speed:

$$c_k = \sqrt{a_k^2 + \frac{4\mu_k}{3\rho_k}}, \quad (37)$$

where a_k is the thermodynamic sound speed. If one of the materials is void, then the corresponding density, sound speed and stress values are set to zero. The resulting approximate volume exchange is computed as follows:

$$\delta V_{k,l}^{max} = u_{k,l}^* S_{k,l} \Delta t. \quad (38)$$

There is also an energy exchange associated with the material interface movement:

$$\delta E_{k,l}^{max} = \sigma_{k,l}^* \delta V_{k,l}^{max}, \quad (39)$$

where $\sigma_{k,l}^*$ is the interfacial normal stress, which is defined by solving the Riemann jump condition with a viscosity-like term:

$$\sigma_{k,l}^* = \frac{\tilde{\sigma}_{l,k}^{bulk,n+1} \rho_k c_k + \tilde{\sigma}_{k,l}^{bulk,n+1} \rho_l c_l}{\rho_l c_l + \rho_k c_k} + \frac{\rho_l a_l \rho_k a_k V_c (\nabla \cdot \mathbf{u})}{\rho_l a_l + \rho_k a_k}. \quad (40)$$

The velocity divergence is calculated for the whole cell from the node-based velocities. If one of the materials is void, $\sigma_{k,l}^*$ is set to the normal stress value of the non-void material. This approach mimics what is done with pressure in [29], where we show that it yields better expansion velocity approximation at the cost of a slight total energy conservation violation.

3.2.1. Constraints on material volume change

Similarly as in the original IASSD model, there are three types of constraints imposed on material volume change by the $\Psi_{k,l}$ limiter. The first one is a minimum sub-cell volume constraint:

$$V_k^{n+1,bulk} + \sum_{l \in M(k)} \Psi_{k,l} \delta V_{k,l}^{max} \geq \kappa_{bot} V_k^{n+1,bulk}, \quad 1 \geq \kappa_{bot} > 0, \quad (41)$$

where κ_{dot} is a parameter controlling the decrease of volume during one time step. κ_{dot} is presumed to be constant and in this paper we use the value $\kappa_{dot} = 0.1$. This is set much lower than the corresponding limit in the bulk stage on purpose – larger volume exchanges are expected to take place in the sub-scale stage.

For void, the volume constraint can be expressed as

$$V_v^{n+1, bulk} + \sum_{l \in M(v)} \Psi_{v,l} \delta V_{v,l}^{max} \geq 0. \quad (42)$$

therefore void volume is allowed to reach zero for void closure to be possible.

The second set of constraints ensures the positivity of internal energy on the new time level. It is relevant only for real materials (as void energy is always zero) and can be written as:

$$m_k e_k^{bulk} + \sum_{l \in M(k)} \sigma_{k,l}^* \Psi_{k,l} \delta V_{k,l}^{max} > 0, \quad (43)$$

The last constraints are related to the stress equilibration – this is slightly different than the pressure equilibration constraints used in [21,29]. In general, the Cauchy stress is what drives interface movement and numerical experiments show that pressure equilibration alone is not sufficient for solids (see Section 5.1).

It is important to equilibrate the normal stress in a controlled way, therefore the limiter must ensure not only that the value moves towards the equilibrium but also that it does so in a smooth and non-oscillatory manner. We presume that the time step is controlled so that the entropy can be considered constant and the following approximation may be used:

$$\bar{\sigma}_{k,l}^{n+1} \simeq \bar{\sigma}_{k,l}^{bulk, n+1} - \frac{\rho_k^n (c_k^n)^2}{V_k^n} \sum_{l \in M(k)} \Psi_{k,l} \delta V_{k,l}^{max}. \quad (44)$$

We use this to predict the stress change as a linear function of the $\Psi_{k,l}$ limiters with a prescribed maximal volume change. The limiters are to be chosen in such a way that each material normal stress $\bar{\sigma}_{k,l}^{n+1}$ defined by Eq. (44) will, if possible, relax towards a pre-defined target stress based on the compressibility factor averaging used in the bulk stage:

$$\bar{\sigma}_{k,l} = \sum_j \beta_j \left(p_j^n - \mathbf{n}_{k,l} S_j^n \mathbf{n}_{k,l} - \frac{\rho_j^n (c_j^n)^2}{V_j^n} \Delta V_j^{bulk, n+1} \right). \quad (45)$$

Here, $\bar{\sigma}_{k,l}$ is not a target tensor defined for the whole computational cell, but a scalar target stress value specific for each interface normal direction. Note that this is different from [21,29], where a single target pressure with volume fraction averaging was used. In our case, compressibility factors are consistent with the bulk stage

formulation and are fully consistent with voids as well (which do not require a special treatment here).

In the worst-case scenario, the normal stress should not diverge from the target. It should also not overshoot the target (which could lead to spurious oscillations). This implies the following set of constraints for normal stress at each material interface:

- If $\bar{\sigma}_{k,l}^{bulk, n+1} \geq \bar{\sigma}_{k,l}$, then:

$$\frac{V_k^n}{\rho_k^n (c_k^n)^2} (\bar{\sigma}_{k,l}^{bulk, n+1} - \bar{\sigma}_{k,l}) \geq \sum_{l \in M(k)} \Psi_{k,l} \delta V_{k,l}^{max} \geq 0, \quad (46)$$

- If $\bar{\sigma}_{k,l}^{bulk, n+1} \leq \bar{\sigma}_{k,l}$, then:

$$\frac{V_k^n}{\rho_k^n (c_k^n)^2} (\bar{\sigma}_{k,l}^{bulk, n+1} - \bar{\sigma}_{k,l}) \leq \sum_{l \in M(k)} \Psi_{k,l} \delta V_{k,l}^{max} \leq 0, \quad (47)$$

The actual computation of $\Psi_{k,l}$ limiter values is based on solving the following minimization problem in each computational cell:

$$\min_{\Psi_{k,l}} \left\{ \sum_k \left[\sum_{l \in M(k)} (1 - \Psi_{k,l})^2 \right] \right\}, \quad (48)$$

which will give us the least restrictive limiter values respecting the aforementioned constraints (41,42,43,46,47). The constrained-optimization problem for finding the limiters $\Psi_{k,l}$ is evaluated as a quadratic optimization problem with linear inequality constraints [37]. In the current implementation, the “QL” software [38] is used to solve this problem.

4. Void treatment

4.1. Closure

Void closure treatment was thoroughly described in [29]. Void is considered closed if its volume fraction falls lower than a pre-set threshold (here 10^{-6}) – if that happens, the void sub-cell is deleted and the volume fractions of the remaining materials are re-normalized to keep the total volume of all materials in the cell consistent.

In the original IASSD model, void closure was considered only in the sub-scale phase. Due to the nature of the modified multi-material cell model, we must now consider also the possibility of void closure at the bulk stage (because the volume fractions can now change in this part of the algorithm). The detection threshold as well as the requirement of re-normalization are the same as for the sub-scale stage closure.

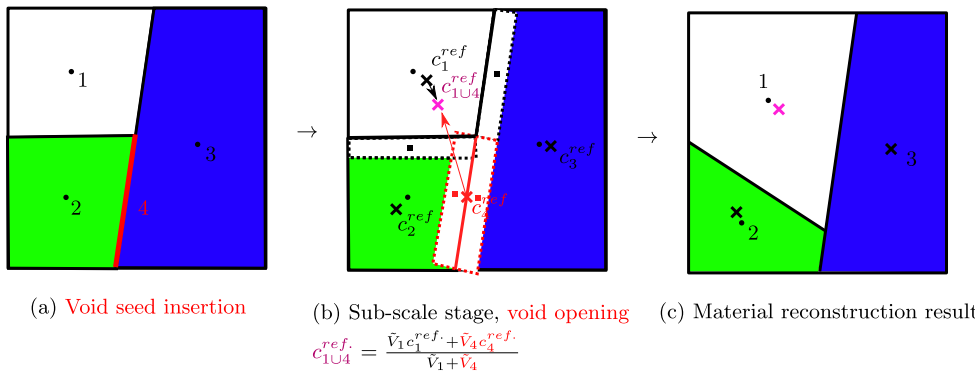
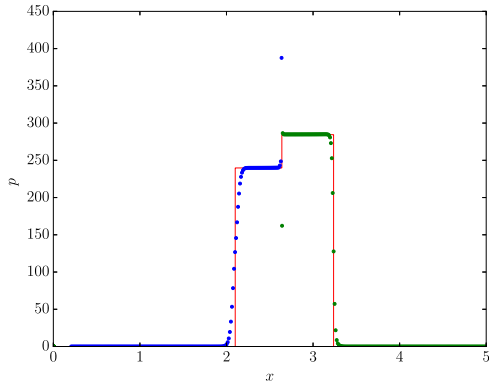
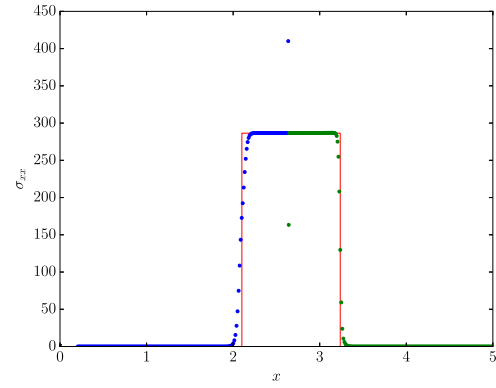


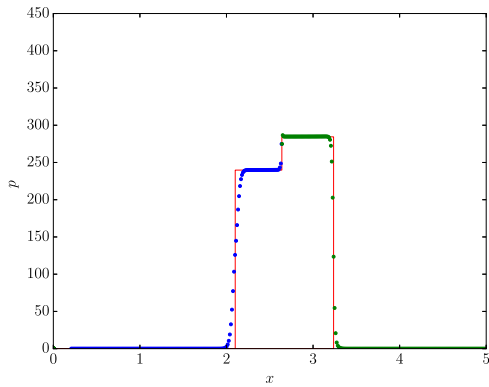
Fig. 3. Example of the addition of a newly opened void sub-cell prior to material reconstruction.



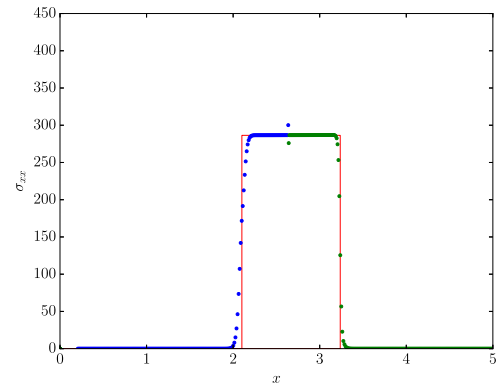
(a) Pressure [kbar], EC only



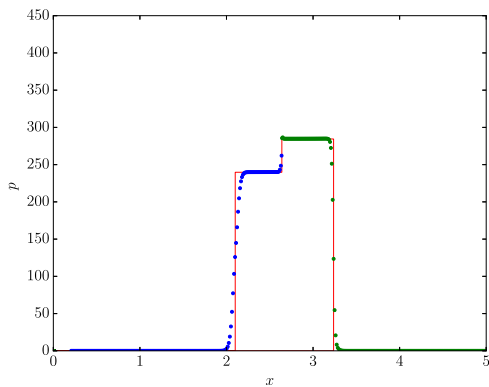
(b) Cauchy stress [kbar], EC only



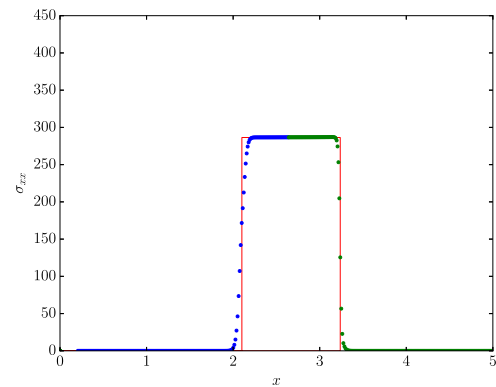
(c) Pressure [kbar], IASSD, pressure eq.



(d) Cauchy stress [kbar], IASSD, press. eq.



(e) Pressure [kbar], IASSD, stress eq.



(f) Cauchy stress [kbar], IASSD, stress eq.

Fig. 4. Impact of a steel plate on an aluminum plate at $t = 1 \mu\text{s}$, 401 cells, $u_{\text{mit.}} = 2000 \text{ ms}^{-1}$. Comparison of equal compressibility computation with an IASSD computation with pressure equilibration or normal stress equilibration. Exact solution is marked with a red line. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

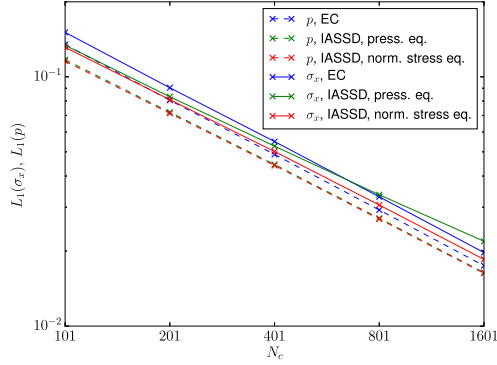


Fig. 5. Impact of a steel plate on an aluminum plate at $t = 1 \mu\text{s}$, $u_{\text{init}} = 2000 \text{ ms}^{-1}$. Convergence plot of total L1 error of Cauchy stress (solid) and pressure (dashed). Comparison of equal compressibility computation with an IASSD computation with pressure equilibration or normal stress equilibration.

If the proportional compressibility bulk model is used, in cells containing void all volume change is initially distributed only to the void material. Clearly, this volume change can be significantly larger than the actual volume of the void sub-cell. This means that the volume change predicted by the bulk model is used only as an input to the closure detection. After the void sub-cells are removed, the re-normalization of volume fractions gives us the actual material volume changes that can be different than those predicted by the bulk model:

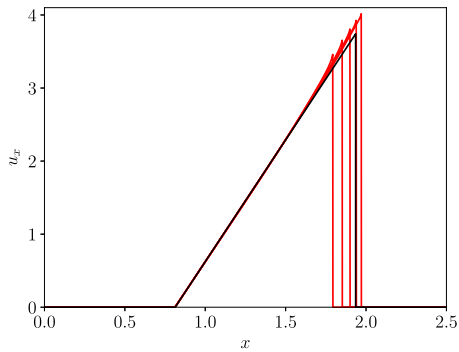
$$\alpha_{c,k}^{n+1} = \alpha_{c,k}^{\text{post-closure}} / \sum_k \alpha_{c,k}^{\text{post-closure}}, \quad (49)$$

$$V_{c,k}^{n+1} = \alpha_{c,k}^{\text{norm}} V_c^{n+1}, \quad dV_{c,k} = \alpha_{c,k}^{n+1} V_c^{n+1} - \alpha_{c,k}^n V_c^n. \quad (50)$$

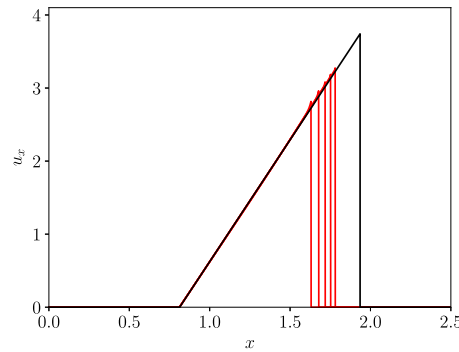
We'll use this material volume change to define the actual weight factors for averaging pressure, stress and artificial viscosity and other material quantities:

$$\beta_{c,k} = \frac{dV_{c,k}}{dV_c} = \frac{\alpha_{c,k}^{n+1} V_c^{n+1} - \alpha_{c,k}^n V_c^n}{V_c^{n+1} - V_c^n}, \quad (51)$$

An infinitesimal cell volume change means that the volume fraction change is also approaching zero. In this limiting case therefore $\beta_{c,k} \approx \alpha_{c,k}$.



(a) EC bulk stage



(b) PC bulk stage

Fig. 6. Expansion of an ideal gas in vacuum, velocity, $t = 0.25 \text{ ms}$. – numerical solution for $N = 2500, 5000, 10000, 20000, 40000$ cells (left to right), – analytical solution.

4.2. Opening

In this paper we deal with void opening on contact surfaces – this means we are aware of the location of the interface between two materials, which determines the location where the void can be created. So far we are not able to construct an algorithm that could create voids arbitrarily and separate areas with the same logical material, which would be required for accurate fracture and damage simulation.

We neglect the concept of interface bonding energy and either consider a material pair as inseparable or as a “contact pair”, which means that a void can open on interfaces between the selected materials. The conditions on which the void will open are usually described as tension between the contacting materials. Therefore we opt to evaluate void opening in the sub-scale stage where the interface-based stress is already being calculated. Allowing void opening in the bulk stage would pose additional algorithmic challenges, such as how to locate the opened void correctly if interface and centroid location is not known exactly. For this reason we assume that void opening is evaluated only in the sub-scale stage.

Due to the presence of limiting conditions in the sub-scale part of the algorithm and also a void volume fraction lower threshold, we can not simply compute the volume of the opened void from the interface tension. For this reason, we rather calculate all sub-cell interactions as if an infinitesimal void was already present at all contact pairs, and if it fails to open we perform an a posteriori correction for which the void is removed. This results in the following algorithm:

1. We insert a *void seed*, a sub-cell located in between the contact pair materials with zero width and volume. The connectivity is changed so that the neighboring sub-cells are **material 1 – zero volume void – material 2**.
2. The sub-scale stage is performed as described in Section 3.2.
3. Now we evaluate if the void has opened (exceeded the minimum volume fraction threshold for void closure). If it did, then we proceed to material reconstruction to obtain the sub-mesh geometry in the computational cell and the sub-scale step is completed. If it did not open, then we proceed to the next step.
4. If the void did not open, then we need to reevaluate the sub-scale step because a different process is happening on the material boundary – not void opening, but a contact problem. Insufficiently developed void sub-cells are removed and the connectivity is restored to the original state **material 1 – material 2**.

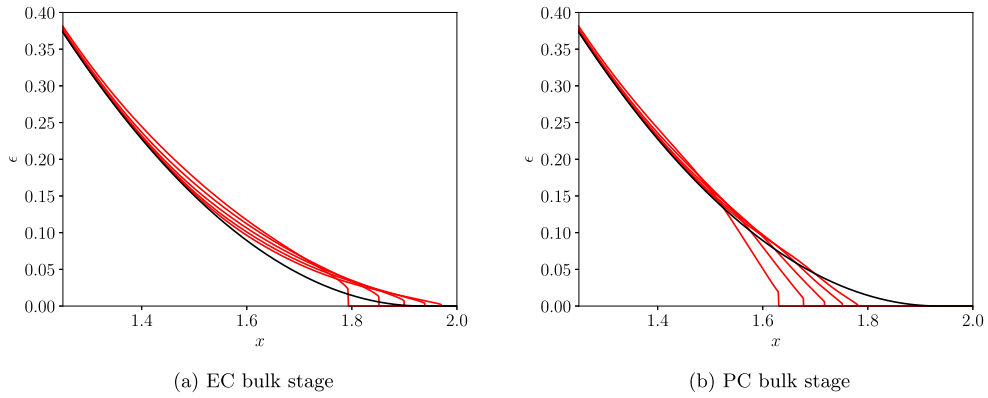


Fig. 7. Expansion of an ideal gas in vacuum, specific internal energy, expansion area closeup, $t = 0.25$ ms. - numerical solution for $N = 2500, 5000, 10000, 20000, 40000$ cells (left to right), - analytical solution.

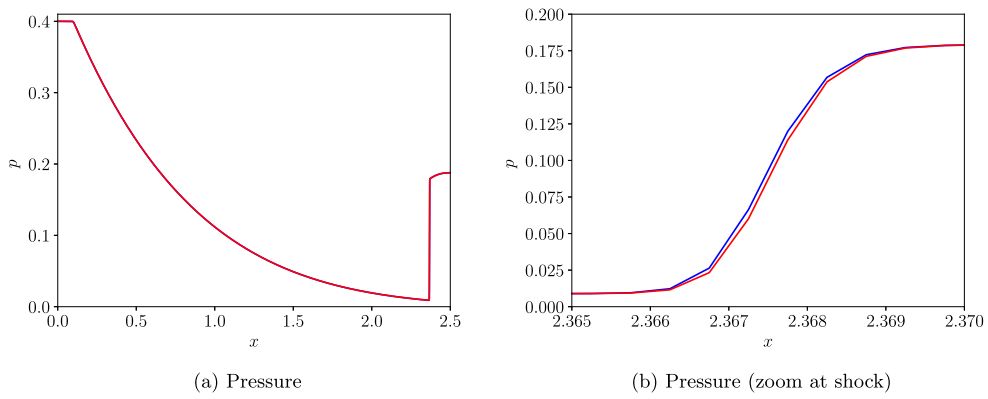


Fig. 8. Expansion of an ideal gas in vacuum, comparison of pressure profiles after void closure, - EC, - PC, $t = 1.2099$ ms. $N = 5000$ cells.

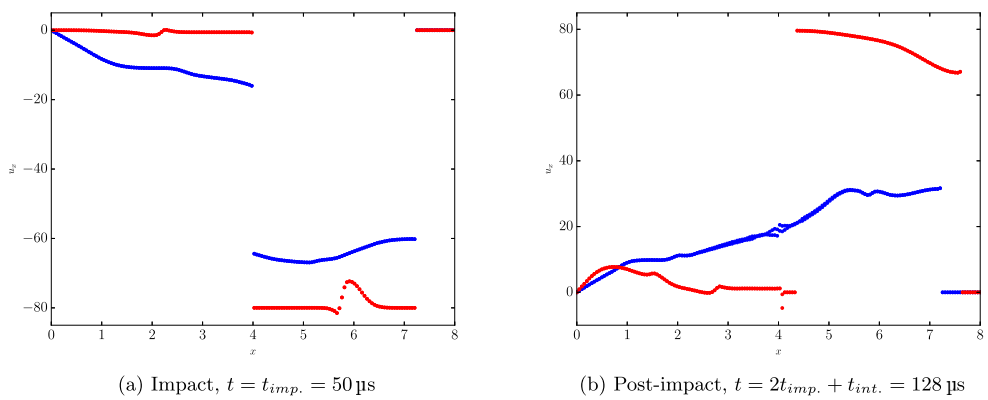


Fig. 9. Moving plate impacting on a fixed plate, comparison of impact-time and rebound velocity profiles [ms^{-1}], $u_{\text{init.}} = 80 \text{ ms}^{-1}$, 161 cells, ● EC bulk stage, ● PC bulk stage.

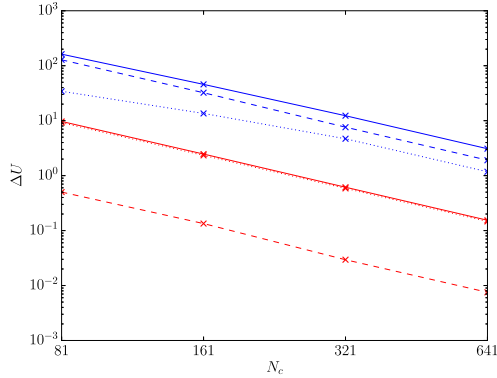


Fig. 10. Moving plate impacting the fixed plate, internal energy convergence plot. $t = t_{imp.} = 50 \mu\text{s}$, $u_{init.} = 80 \text{ ms}^{-1}$. ● EC bulk stage, ● PC bulk stage, - total internal energy, - - static plate internal energy, - · - moving plate internal energy.

5. Now we perform the sub-scale stage again as in Section 3.2, but this time without the void seeds that have no potential for opening. We will perform material reconstruction and this is the final result of the computational step.

One of the important questions is – what happens if there is a void already present in the computational cell? This will happen for example in two-dimensional simulations of impacting bodies in areas where the contact zone meets the surrounding void environment. Neglecting opening in such cells leads to artificial bonding of the materials in such corner zones. On the other hand, creation of several independent void sub-cells within one multi-material cell is not feasible, as it could lead to a new void sub-cell being created at each time step. Instead, we allow a second void sub-cell to emerge *only* during void opening in the sub-scale stage. After the sub-scale stage is completed, both void sub-cells are merged so that in all other parts of the algorithm only one void subcell can be present in any computational cell at a time.

An example of this process is depicted in Fig. 3. In the beginning of the sub-scale stage, a void sub-cell 1 is already present, but a new void sub-cell opens with the index 4. For the sake of material reconstruction, their volumes are summed and a weighted average of their reference centroids $c_{1\cup 4}^{ref}$ is used. It is to be noted that

this does not guarantee instantaneous void opening in one computational step as seen in Fig. 3c. This approach will therefore generate a small amount of artificial bonding that, however, does not seem to alter the numerical results as significantly as if opening was not considered at all here. To eliminate this inaccuracy, a general sub-cell conglomeration algorithm would be required, which is out of the scope of this paper (although such approach is indeed possible and has been described in [39]).

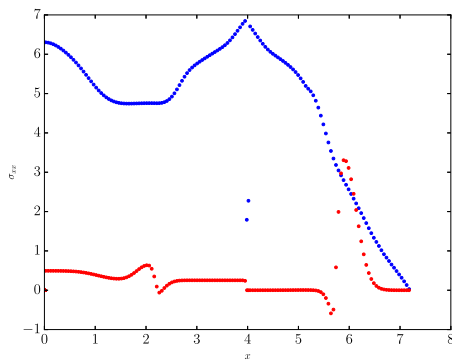
5. Numerical results

In this section we present selected numerical examples. There are three one-dimensional tests with planar symmetry starting from a Lagrangian impact problem to evaluate the performance of the overall IASSD model on simple problems involving solids. Then we return to the case of gas expanding in vacuum and show that our model with the modified bulk stage is applicable for fluid-void interactions as well. The last one-dimensional problem represents a more complex solid plate impact case which involves void closure and opening. Then there are two more two-dimensional tests – an implosion of a cylindrical structure with multiple layers separated by void (involving a simplified fluid representation on a symmetrical polar mesh). The last example is a round impactor hitting a flat surface at low and high speeds with void closure, void opening and including material strength. We did not include standard fluid flow testing problems because the design options discussed in this paper seem to have little effect on fluid-only interactions (even with vortical flow patterns).

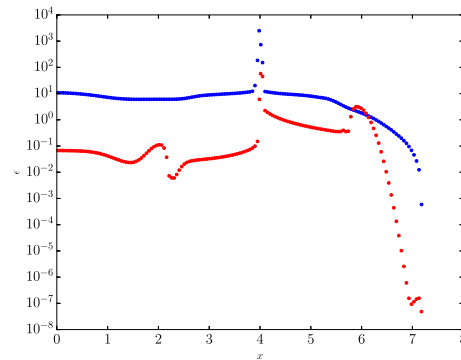
Except for the first subsection, the Eulerian as Lagrange + Remap approach is used in all tests, therefore after each computational step the mesh is rezoned to its initial state and remapping is performed.

5.1. Lagrangian computation of a steel plate impacting on an aluminum plate

The first test serves as an example of an idealized plate collision computation in Lagrangian mode with one multi-material cell in the middle. Void closure or opening is not considered. The domain is $[0, 5]$, the left half represents the steel plate moving with high velocity ($u_{init.} = 2000 \text{ ms}^{-1}$). The steel material is modeled using the Mie-Grüneisen equation of state (9) with linear elastic-plastic model with Von Mises yield condition (15). The following material



(a) Normal comp. of the Cauchy stress $\sigma_{x,x}$ [kbar]



(b) Specific internal energy $[\text{J g}^{-1}]$

Fig. 11. Moving plate impacting on a fixed plate, $t = t_{imp.} = 50 \mu\text{s}$, $u_{init.} = 80 \text{ ms}^{-1}$, 161 cells, ● EC bulk stage, ● PC bulk stage.

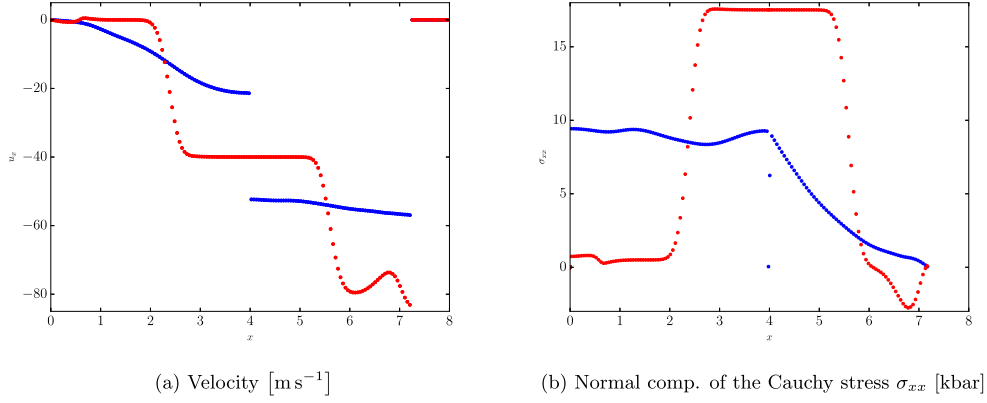


Fig. 12. Moving plate 3 μ s after impacting on a fixed plate, $t = t_{imp.} + 3 \mu$ s = 53 μ s, $u_{mit.} = 80$ ms⁻¹, 161 cells, ● EC bulk stage, ● PC bulk stage.

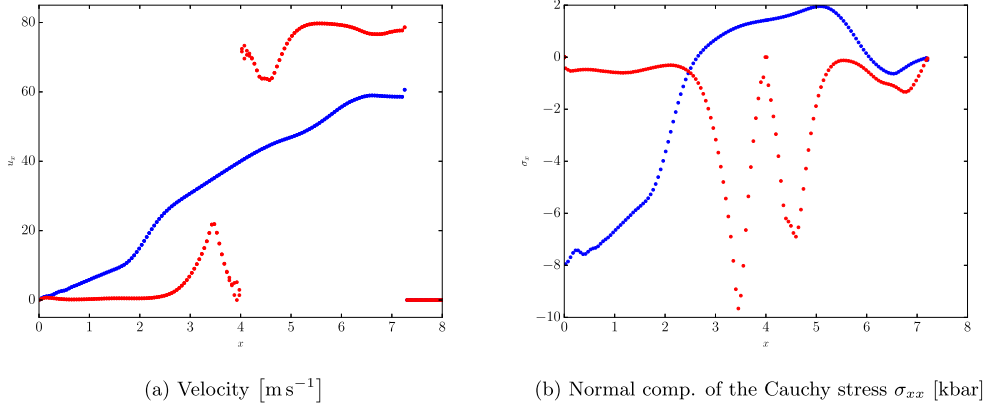


Fig. 13. Moving plate immediately before bouncing off the fixed plate, $t = t_{imp.} + t_{int.} - 1 \mu$ s = 77 μ s, $u_{mit.} = 80$ ms⁻¹, 161 cells, ● EC bulk stage, ● PC bulk stage.

parameters are used:

$$\rho_0 = 7.905 \text{ g cm}^{-3}, \mu = 547 \text{ kbar}, Y_0 = 70 \text{ kbar}, c_0 = 4570 \text{ ms}^{-1}, \gamma = 2, S_\alpha = 1.49. \quad (52)$$

The right plate is stationary and we use the following values for the aluminum material:

$$\rho_0 = 2.785 \text{ g cm}^{-3}, \mu = 276 \text{ kbar}, Y_0 = 3 \text{ kbar}, c_0 = 5328 \text{ ms}^{-1}, \gamma = 2, S_\alpha = 1.338. \quad (53)$$

This testing problem is useful for evaluating the performance of the whole multi-material cell model. In Fig. 4 we compare the IASSD computation with a simple equal compressibility model in a short time after impact. As shocks propagate in both plates in opposite directions, the EC model creates material pressure and stress values that are out of bounds defined by neighboring single material states. This also shows the justification for normal stress equilibration in the IASSD sub-scale stage for solids. On the interface of two different solid materials a pressure discontinuity can develop, but this discontinuity results in violating bounds in the stress value. If we equilibrate the normal stress, the material pressures match the corresponding up-/downstream values correctly and the normal stress bounds are preserved. We also show the exact solution for this problem in Fig. 4. To obtain the solution we

have followed the approach outlined in [40]. In this case, however, the impact velocity is high enough for the plastic wave to move faster than the elastic wave, so the solution is even simpler – we solve the system of shock jump relations on both waves with known yield stresses for both materials and equations of state. The resulting pressure and stress values are:

$$p_L = 239.757 \text{ kbar}, p_R = 284.424 \text{ kbar}, \sigma_x = 286.424 \text{ kbar}, \quad (54)$$

In Fig. 5, we show a convergence plot of the total relative L1 error of the normal stress and pressure values. We can see that the convergence rate is almost equal for both EC and IASSD with normal stress equilibration – the EC model is inaccurate mostly in the mixed cell. Interestingly, when using IASSD with pressure equilibration, the convergence rate is slightly lower in the deviatoric stress component.

5.2. Expansion of an ideal gas in vacuum

This testing problem was used in [29] as a basic benchmark of the one-dimensional fluid – void interaction. It is our first test of choice to compare the influence of the modified bulk stage on computations with fluids. The initial conditions represent a special case of the Riemann problem [41] where the right material is void and the left material is an ideal gas with constant properties:

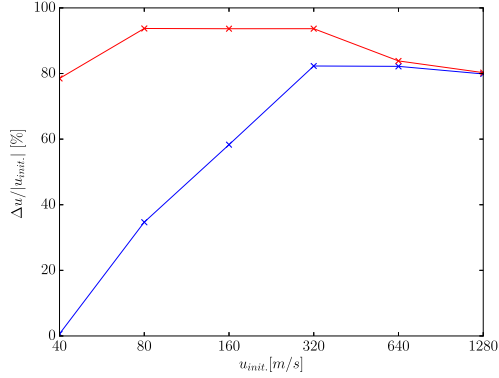


Fig. 14. Moving plate impacting the fixed plate, average moving velocity after bounce at $t = 2 * t_{imp} + t_{int.}$ as a percentage of the initial plate velocity. ● EC bulk stage, ● PC bulk stage.

$\gamma = 1.4$, $\rho_L = 1.0$, $p_L = 0.4$. The ideal gas is contained within the $[0, 1]$ interval of the computational domain $[0, 2.5]$. Both boundaries are fixed, resulting in two phases of this test – in the first phase ($t < t_{close}$, $t_{close} \sim 0.4009$ ms) the gas expands into void. The second phase ($t > t_{close}$) allows for testing the behavior of void closure and a wave reflection from the wall.

In Fig. 6, the velocity profile is shown during expansion and compared with an exact solution of $u_{exact} = 3.741657$ (see [29] for the derivation of the analytical solution for this test case). The original EC bulk stage results in higher velocities, which do not converge, but instead overshoot the exact solution. The PC bulk stage, however, behaves more closely to results obtained by pure cell Lagrangian computation with free boundary [29,42] – that is, much slower than exact solution, with very slow convergence (if any). This is expected, because in such case all expansion of the multi-material cell will be transformed into expansion of the non-void material at the bulk stage, therefore the material interface will behave closely to the free boundary in pure Lagrangian computation.

We compare the internal energy profiles in Fig. 7, where we can again see slow convergence behavior for the PC bulk stage. Unlike in the EC model, the specific internal energy does not overshoot the exact solution in the multi-material cell value.

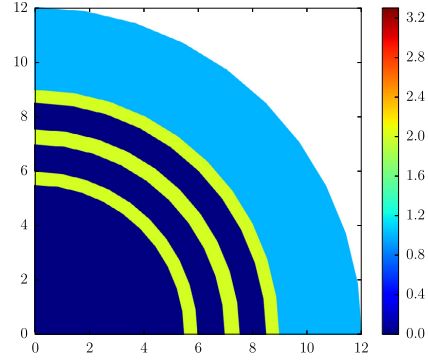


Fig. 15. Implosion of cylindrical layers, initial state at $t = 0$, density distribution $[\text{g cm}^{-3}]$.

Table 1

Average moving plate velocity at time $t = 2 * t_{imp} + t_{int.} = 128 \mu\text{s}$ for different resolutions, $u_{init.} = 80 \text{ ms}^{-1}$, PC bulk stage.

Resolution	$\frac{\sum V_{c,plate}(u_c + u_{c+1})/2}{\sum V_{c,plate}}$
81	72.133
161	74.982
321	76.600
641	77.685

In the pre-closure phase we have seen significant inaccuracies in the interface velocity approximations. But at the same time the expanding regions contain little mass with small internal energy. In Fig. 8 we want to show that these inaccuracies do not result in different wave shapes or timing after the void closes (as seen in [29]). This is valid for the PC bulk stage as well – both methods give very similar pressure profiles after void closure even for lower resolutions. Therefore we can conclude that the possibly inaccurate positioning of the gas-void interface will not impair the accuracy in simulating subsequent interactions.

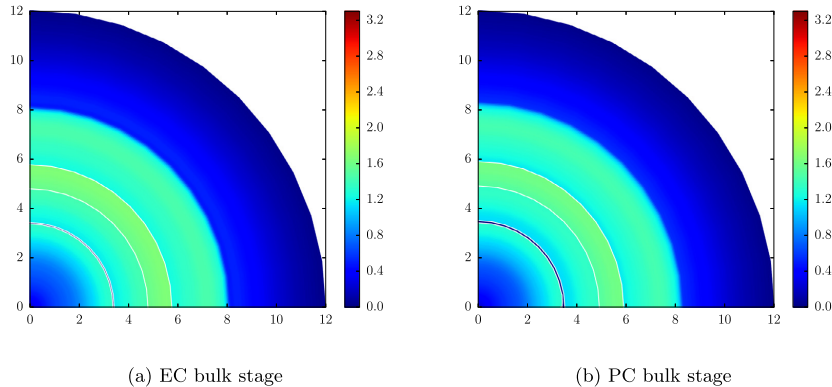


Fig. 16. Implosion of cylindrical layers at $t = 9$ ms, density distribution $[\text{g cm}^{-3}]$.

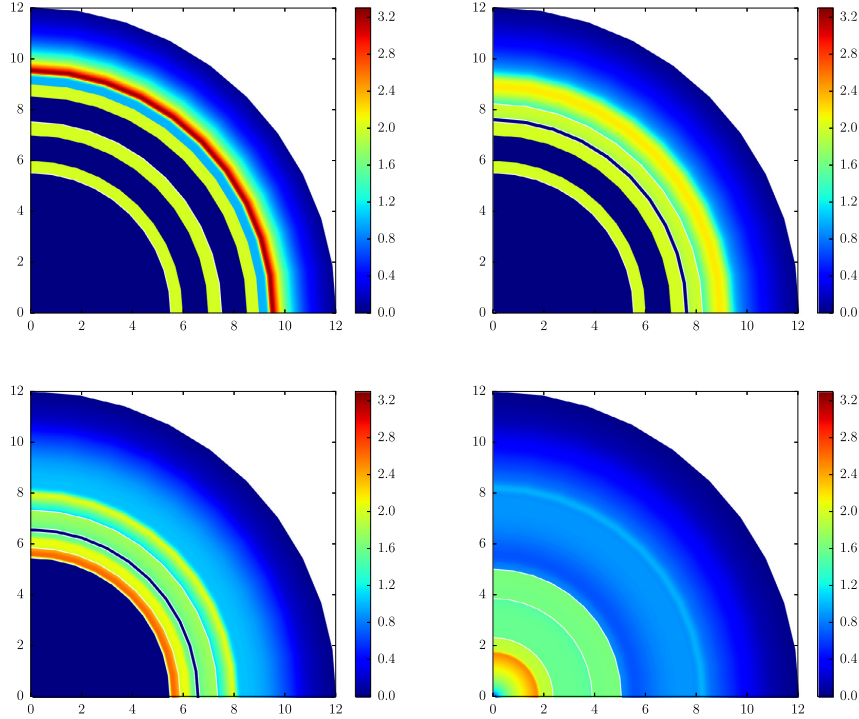


Fig. 17. Implosion of cylindrical layers at several time levels $t = 1, 2, 3, 6$ ms, PC bulk stage, density distribution [g cm^{-3}].

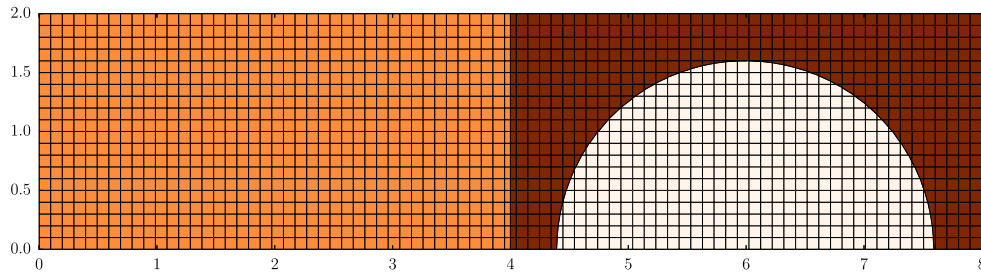


Fig. 18. Round object impacting a fixed plate, initial logical material distribution, $t = 0 \mu\text{s}$, 81×20 cells.

5.3. Impact of a moving steel plate on a fixed steel plate

The next test case is focused on the one dimensional simulation of impacting solid bodies at a relatively low speed $u_{init.} \ll c_{elast.}$. A fixed steel plate is positioned in the $[0, 4]$ part of the domain, while a moving plate is positioned at $[4.4, 7.6]$ initially. The used steel material properties are specified in (52).

This problem has a more complex nature with several stages that will allow us to evaluate the performance of different model functionalities:

1. Non-interacting translation phase – the plate moves through void undisturbed
2. Impact – void closure, the time duration of this phase is $t_{imp.} = 0.4\text{cm}/u_{init.}$
3. Post-impact interaction (shocks propagating through both plates), we can approximate the duration of this phase as $t_{int.} = 4 \cdot 3.2\text{cm}/c_0 \sim 28 \mu\text{s}$
4. Void opening on stress drop in interface region and subsequent rebound of the moving plate away from the fixed plate.

Our goal is to have as little disturbance as possible for all quantities at or immediately before the impact time. The velocity profile at that time is displayed in Fig. 9a, where we can see that there is a substantial decrease in the overall velocity for computation with equal compressibility bulk stage. For the PC model, the velocity is still very close to the initial value with a small disturbance created possibly on the entry of the material interface in the interaction zone.

In Fig. 11 we see the consequences of the premature interaction caused by the EC bulk stage – namely pre-loading both plates

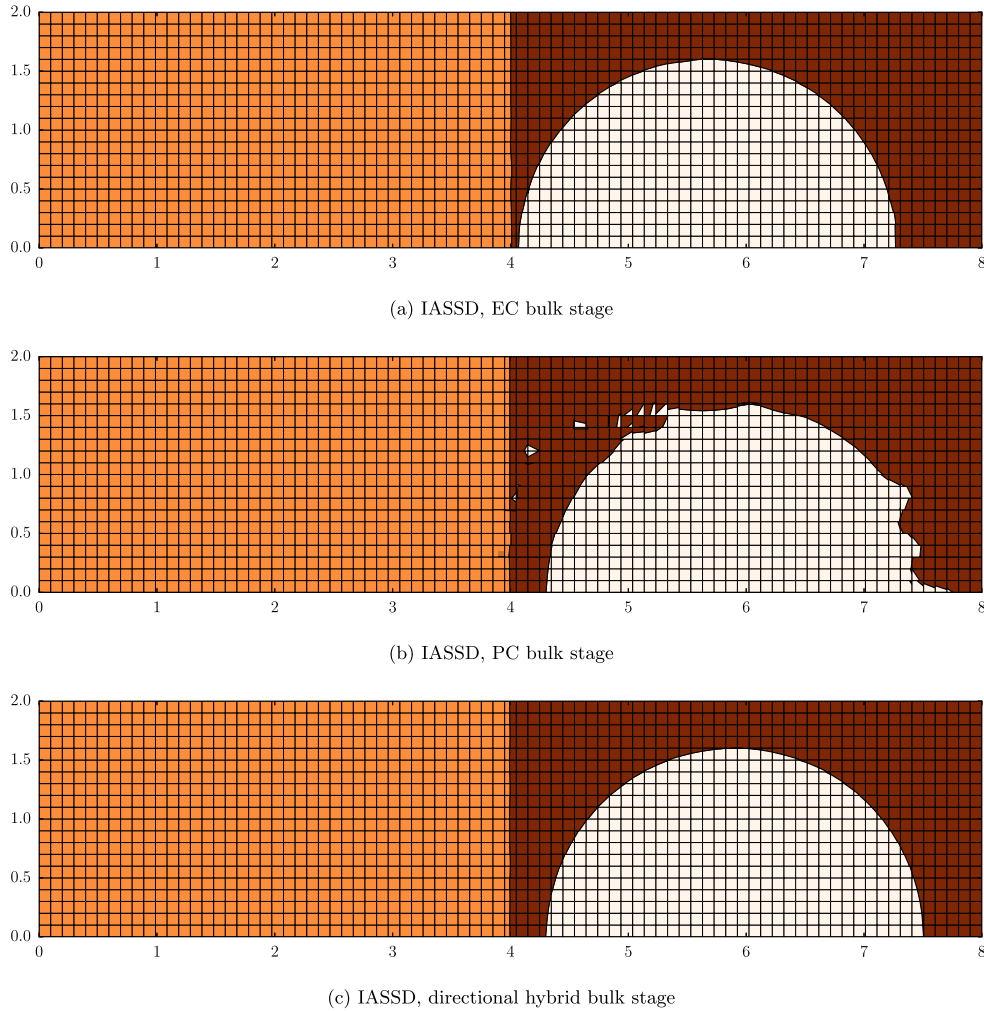


Fig. 19. Round object after impacting a fixed plate, material distribution at $t = 128 \mu\text{s}$, initial velocity $u_{\text{init.}} = 80 \text{ ms}^{-1}$, 81×20 cells.

with nonphysical stress, delaying the actual time of void closure and preventing the formation of a sharp shock profile. We can also measure the excessive internal energy deposited in both impacting bodies for increasing computational resolution, as shown in Fig. 10. Although both methods converge, still we can see that the EC bulk stage results in an order of magnitude higher energy deposit consistently for coarse and finer meshes. The difference is even bigger if we compare only the energy deposited in the static plate.

This is even more evident if we move further $3 \mu\text{s}$ from the impact time (displayed in Fig. 12). For the EC bulk stage, the void still hasn't closed and the velocity and stress profiles do not resemble the expected shock patterns which are clearly visible in the results using the PC bulk stage.

Moving even further, in Fig. 13, the consequence of the distorted shock timing for the EC bulk stage prevents the onset of the tension wave and makes void opening and plate separation impossible. The final velocity profile is shown in Fig. 9b, showing that

for the EC simulation the void has not opened and the velocity is continuous at the interface (with a small oscillation).

It seems that increasing resolution only might not be effective as a way of avoiding problems with the EC model. In Table 1 we can see an increase in accuracy in terms of the average reflected plate velocity at $t = 128 \mu\text{s}$ using the PC bulk stage. For the EC model the computation fails for 321 and 641 cells after the unsuccessful void opening (which is in fact often a repeated succession of void closures and openings). For lower resolution it fails to sustain opening as well, so it is not included in the comparison.

We can also evaluate the dependence of the bounce velocity on the initial velocity. In Fig. 14 we can visualize the dissipation of momentum in the moving plate during impact – the bounce velocity is displayed as a percentage of the initial velocity. We can see that the relative error of the equal compressibility bulk stage diminishes with increasing initial velocity (actual opening and separation with this method happens for velocities of 160 ms^{-1} and

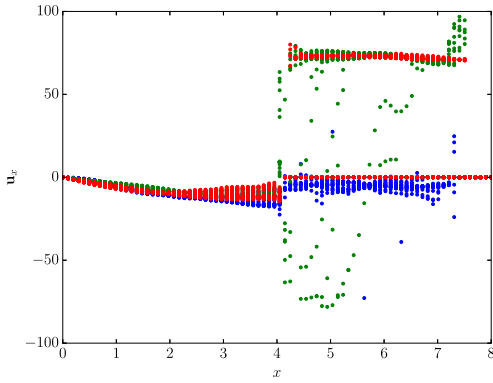


Fig. 20. Round object after impacting a fixed plate, velocity scatter plot (x-axis projection) at $t = 128 \mu s$, initial velocity $u_{init.} = 80 \text{ ms}^{-1}$, 81×20 cells. ● EC bulk stage, ● PC bulk stage, ● hybrid bulk stage.

larger). At approximately 1/4 of the thermodynamic sound speed, the problem starts to be dominated by plastic deformation and both methods provide comparable accuracy. This shows that the proportional compressibility model is equally useful for high speed impact simulations as the equal compressibility model. However, if lower velocities are present the PC model clearly gives superior results.

5.4. Cylindrical implosion of several void-separated layers

In the first two dimensional test we assess the performance of the model in the case of three imploding cylindrical layers of a low-compressibility material separated by void gaps. A simplified constitutive model without material strength is used and the computations are performed on a polar mesh with 250 radial layers and 10 axial segments with a radial domain of $[0.1, 12]$ with wall boundary conditions. The layers are located between the radii $[5.5, 6]$, $[7, 7.5]$, $[8.5, 9]$ and the following stiffened gas equation is used:

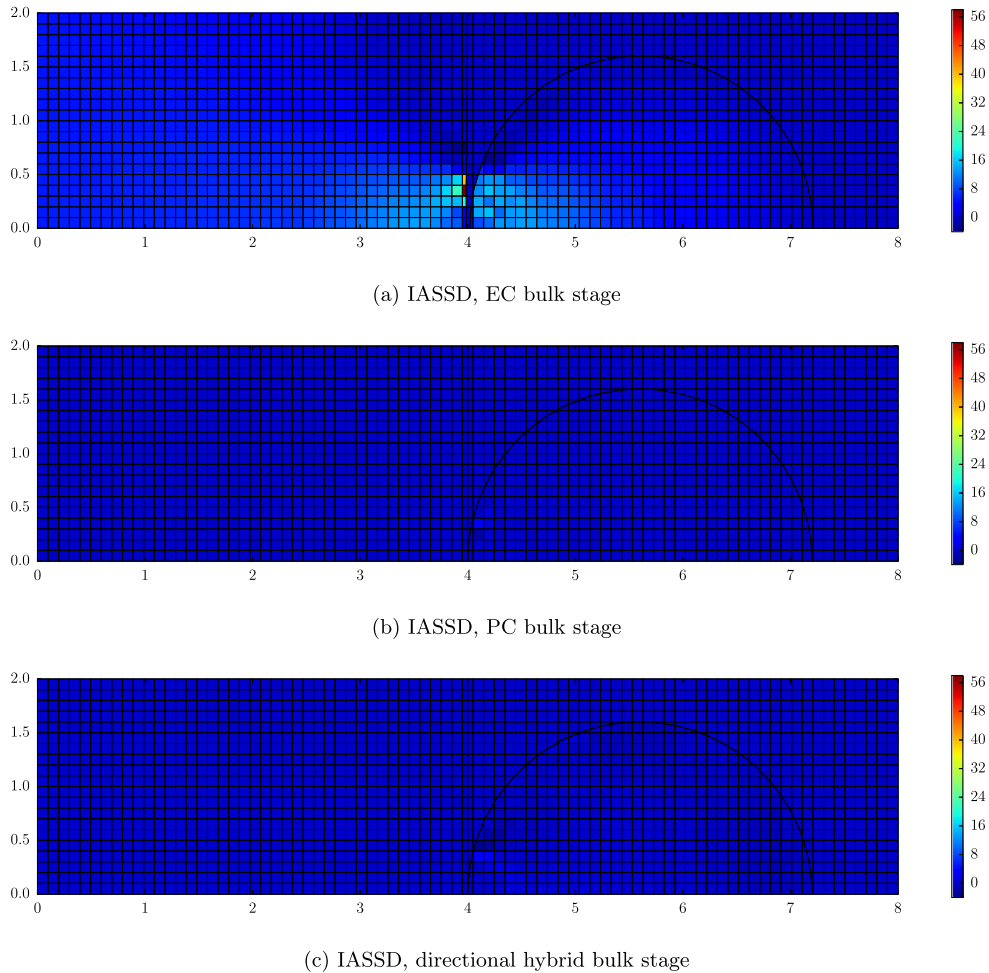
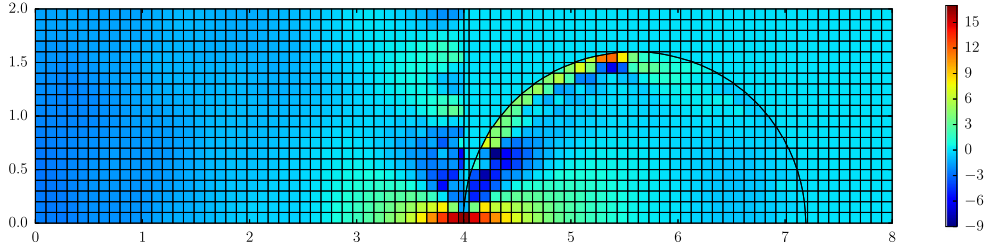
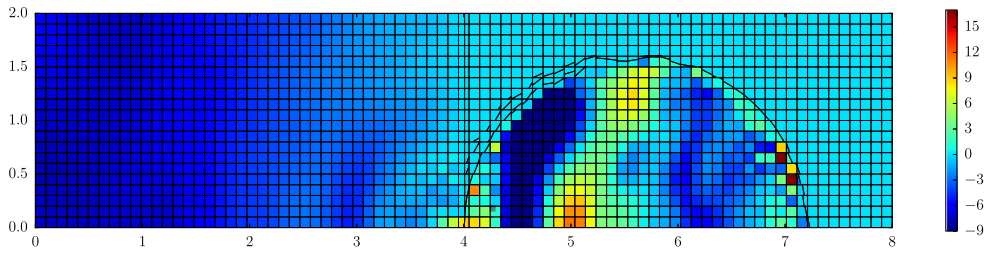


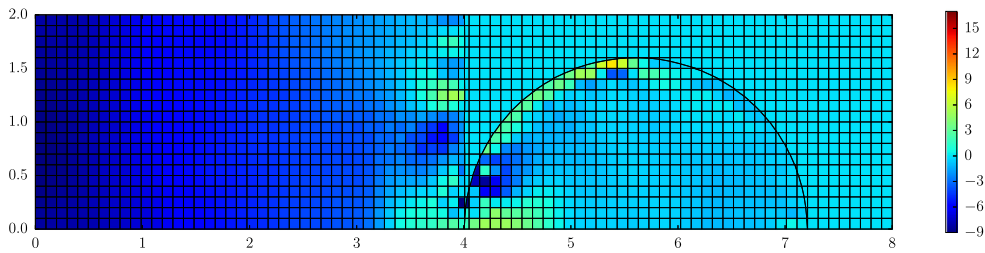
Fig. 21. Round object impacting a fixed plate, x-axis Cauchy stress projection σ_{xx} [kbar] at $t = 50 \mu s$, initial velocity $u_{init.} = 80 \text{ ms}^{-1}$, 81×20 cells.



(a) IASSD, EC bulk stage



(b) IASSD, PC bulk stage



(c) IASSD, directional hybrid bulk stage

Fig. 22. Round object after impacting a fixed plate, x-axis Cauchy stress projection σ_{xx} [kbar] at $t = 87\mu\text{s}$, immediately before void opening. Initial velocity $u_{init.} = 80\text{ ms}^{-1}$, 81×20 cells.

$$p = (\gamma - 1)\rho\varepsilon - \gamma p_{\infty}, \quad \gamma = 4.4, \quad p_{\infty} = 1\text{ kbar}. \quad (55)$$

The layers are initially at rest and the initial conditions are the following:

$$p_{init.} = 0\text{ kbar}, \quad \rho_{init.} = 2\text{ g cm}^{-3}. \quad (56)$$

In the outermost region defined by radius $r > 9$, an ideal gas layer is located with the following parameters:

$$\gamma = 5/3 \quad \varepsilon_{init.} = 10 \times 10^{-6}\text{ J g}^{-1}, \quad \rho_{init.} = 1\text{ g cm}^{-3}, \quad (57)$$

with the exception of the outermost layer of computational cells, where the specific internal energy is set to 1 kJ g^{-1} to initiate an inward shock, triggering the compression and collisions of the stiffened gas layers. The rest of the domain is filled with void. The initial setting is shown in Fig. 15.

In Fig. 16 we compare the results computed using the EC and PC bulk stage models at a late time, after all layers have been compressed towards the center. The layers rebound outward and we

can see a void gap opening in between. The results computed using the PC model show a slightly faster outward movement, but otherwise they are very similar. The hybrid model is not included in the comparison as we are not using material strength and the problem exhibits radial symmetry – in this case the strain is always perpendicular to the material interface. The time evolution of the density distribution is shown in Fig. 17, displaying the subsequent closure of all void gaps up to the center of the domain.

5.5. Impact of a round object on a fixed steel plate

The ability of this closure model to reproduce an elastic collision of solid bodies in two dimensions is tested on the simulation of a round impactor 3.2 cm in diameter, centered at $(6, 0)$ bouncing off a steel wall (between $x \in [0, 4]$). Initial configuration is shown in Fig. 18 (the ball and wall are different logical materials with the same physical properties), the steel material parameters are identical as in the previous test case, see (52). This problem has a similar timeline as the one-dimensional plate impact in Section 5.3, but

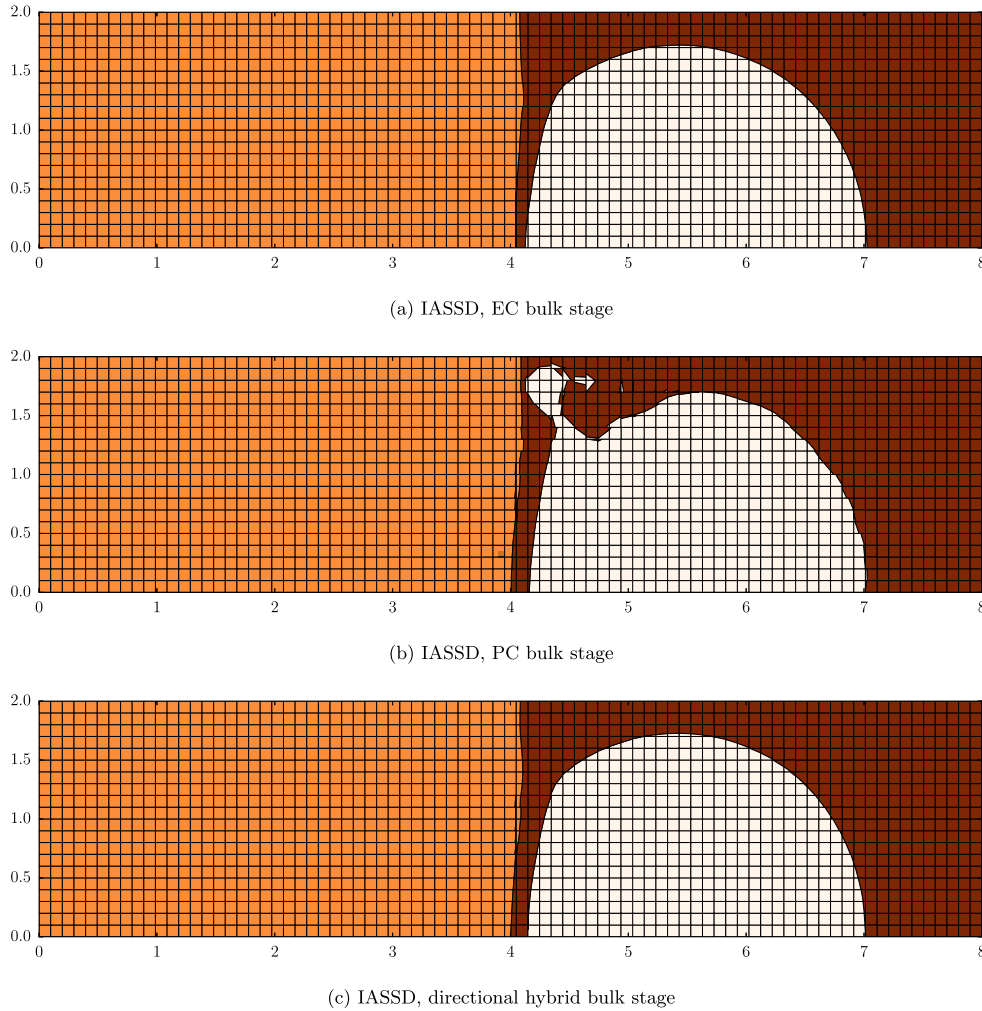


Fig. 23. Round object after impacting a fixed plate, material distribution at $t = 41 \mu\text{s}$, initial velocity $u_{\text{init.}} = 640 \text{ ms}^{-1}$, 81×20 cells.

the underlying physics is slightly different as the shock movement interaction is much more complex in the round body. Also most of the impact interface interaction is centered in approximately 4 computational cells, a small percentage of the total interface length of the impactor.

In Fig. 19 we show the material distribution at $t = 128 \mu\text{s}$. At this time, the impactor should be in the post-opening phase, moving away from the fixed plate. We can see that this is not the case for the EC bulk stage, where a very limited opening gap is observed. Using the PC bulk stage results in a completed opening phase and the impactor retains much of its original momentum. Unfortunately its symmetry is distorted severely and fragmentation occurs at the upper part, where the velocity is close to parallel with the interface direction. The hybrid model combines the best of both approaches, where most of the momentum is recovered with no visible deformation of the impactor shape after it bounces.

In Fig. 20 we show the comparison of the velocity x-axis component at $t = 128 \mu\text{s}$, showing clearly the loss of momentum for the EC bulk stage. Also note that the hybrid model does not incur a significant decrease of the bounce velocity compared to the pure PC bulk stage.

To evaluate the extent of the pre-impact interaction, we compare the x-axis compressive/tensile component of the Cauchy stress tensor at the ideal impact time $t = 50 \mu\text{s}$. This is shown in Fig. 21 – you can see significant stress loading for the EC model already at the impact time, which causes an already visible momentum dissipation and void gap compared to the PC and hybrid models which are much closer to an ideal contact. There is no visible interface deformation yet at that time for the non-hybrid PC bulk stage.

In Fig. 22 we show how the stress distribution changes at time $t = 87 \mu\text{s}$, when void opening and impactor rebound is imminent. For the EC bulk stage we can see higher compressive stress loading

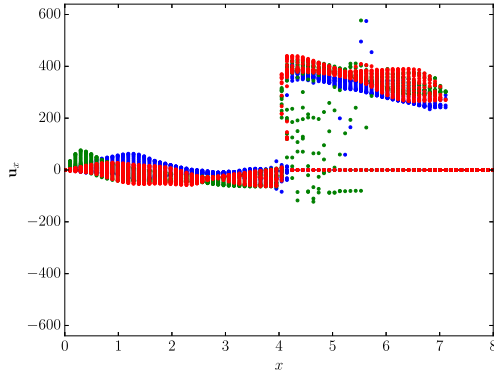


Fig. 24. Round object after impacting a fixed plate, velocity scatter plot (x-axis projection) at $t = 41 \mu\text{s}$, initial velocity $u_{\text{init.}} = 640 \text{ ms}^{-1}$, 81×20 cells. ● EC bulk stage, ● PC bulk stage, ● hybrid bulk stage.

(which inhibits opening) in the interaction area as well as lower tensile stress in the fixed plate overall. In this phase interface distortion sets in for the non-hybrid PC model.

In the 1D impact simulation we have observed that when we increase the impact velocity, the PC and EC bulk stage models converge. In the 2D case, however, plastic deformation starts to determine the behavior at higher velocities. In Fig. 23 we compare the aftermath of an impact with initial velocity increased to $u_{\text{init.}} = 640 \text{ ms}^{-1}$, which will cause a slight deformation of both the target and the impactor which is followed by a void opening and rebound with reduced velocity. We can see that the material distribution is visually very similar for the EC and hybrid models. The pure PC bulk stage again produces an interface deformation in the form of a material jet. This distortion is located where the projectile is most “flattened”. This is in line with our hypothesis that such model will not be accurate if the deformation runs parallel to the interface.

If we compare the bounce velocity distribution for the fast projectile (Fig. 24) we see that the EC and hybrid bulk stage models deliver comparable performance. For the pure PC model, offshoots in velocities are visible due to the distortion of the projectile shape.

6. Conclusion

In this paper we have presented an improvement of the Interface-Aware Sub-Scale Dynamics closure model [29] developed for simulating impacts and other interactions of solid materials at low or high speeds. This new extension of the IASSD model consists of:

- A new formulation of the sub-scale stage of the model for solid materials including a limiter for normal stress equilibration at material interfaces.
- A new formulation of the bulk stage of the model, which improves accuracy of low speed solid impact events. It was shown to converge to the results of the original model for fluids or high speed impacts.
- A void opening algorithm to simulate not only contact, but also the rebound after impacting.

We have evaluated the performance of this IASSD extension on several one- and two-dimensional numerical tests involving interactions of fluids, solids and voids. It was shown that the new model improves the quality of solution for impacts at low speeds while not distorting the results for impacts at higher speeds.

In the future, we plan to extend the model to cylindrical geometry. Other possible future developments include exploring more complex void opening conditions which would allow for representing not only separation of bodies, but introducing voids within materials. This would allow us to simulate fracture initiation and propagation. Another interesting option is to investigate ways of including interface sliding in the multi-material cell model.

In memoriam

This paper is dedicated to the memory of Dr. Douglas Nelson Woods (*January 11th 1985 - †September 11th 2019), a promising young scientist and post-doctoral research fellow at Los Alamos National Laboratory. Our thoughts and wishes go to his wife Jessica, to his parents Susan and Tom, to his sister Rebecca and to his brother Chris, whom he left behind.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Matej Klima: Writing - original draft, Software, Visualization. **Andrew Barlow:** Writing - review & editing, Software, Conceptualization. **Milan Kucharik:** Writing - original draft, Conceptualization. **Mikhail Shashkov:** Writing - review & editing, Conceptualization.

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.

This work was partially supported by the Czech Technical University grant SGS19/191/OHK4/3T/14, the Czech Science Foundation project 18-20962S, the Czech Ministry of Education project RVO 68407700 and the European Regional Development Fund project CZ.02.1.01/0.0/0.0/16_019/0000778.

References

- [1] Caramana EJ, Burton DE, Shashkov MJ, Whalen PP. The construction of compatible hydrodynamics algorithms utilizing conservation of total energy. *J. Comput. Phys.* 1998;146(1):227–62.
- [2] Maire PH. A high-order cell-centered lagrangian scheme for two-dimensional compressible fluid flow on unstructured meshes. *J. Comput. Phys.* 2009;228(7):2391–425.
- [3] Knupp P, Steinberg S. *Fundamentals of grid generation*. Boca Raton: CRC Press; 1993.
- [4] Winslow AM. Equipotential zoning of two-dimensional meshes. *Tech Rep UCRL-7312*, Lawrence Livermore National Laboratory 1963.
- [5] Knupp PM, Margolin LG, Shashkov MJ. Reference jacobian optimization-based rezoning strategies for arbitrary lagrangian eulerian methods. *J. Comput. Phys.* 2002;176(1):93–128.
- [6] Dukowicz JK, Baumgardner JR. Incremental remapping as a transport/advection algorithm. *J. Comput. Phys.* 2000;160(1):318–35.
- [7] Margolin LG, Shashkov M. Second-order sign-preserving remapping on general grids. *Tech Rep LA-UR-02-525*, Los Alamos National Laboratory 2002.
- [8] Loubere R, Shashkov M. A subcell remapping method on staggered polygonal grids for arbitrary-lagrangian-eulerian methods. *J. Comput. Phys.* 2005;209(1):105–38.

- [9] Kucharik M, Shashkov M. Conservative multi-material remap for staggered multi-material arbitrary lagrangian–eulerian methods. *J. Comput. Phys.* 2014;258:268–304.
- [10] Barlow AJ, Maire P-H, Rider WJ, Rieben RN, Shashkov MJ. Arbitrary lagrangian-eulerian methods for modeling high-speed compressible multimaterial flows. *J. Comput. Phys.* 2016;322:603–65.
- [11] Benson DJ. Computational methods in lagrangian and eulerian hydrocodes. *Comput. Methods Appl. Mech. Eng.* 1992;99(2–3):235–394.
- [12] Margolin LG. Introduction to “an arbitrary lagrangian-Eulerian computing method for all flow speeds”. *J. Comput. Phys.* 1997;135(2):198–202.
- [13] Anderson RW, Elliott NS, Pember RB. An arbitrary lagrangian-eulerian method with adaptive mesh refinement for the solution of the euler equations. *J. Comput. Phys.* 2004;199(2):598–617.
- [14] Galera S, Maire P-H, Breil J. A two-dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction. *J. Comput. Phys.* 2010;229(16):5755–87.
- [15] Breil J, Galera S, Maire PH. Multi-material ALE computation in inertial confinement fusion code CHIC. *Computers & Fluids* 2011;46(1):161–7.
- [16] Youngs DL. Time dependent multi-material flow with large fluid distortion. In: Morton KW, Baines MJ, editors. *Numerical Methods for Fluid Dynamics*. Academic Press; 1982. p. 273–85.
- [17] Dyadechko V, Shashkov M. Reconstruction of multi-material interfaces from moment data. *J. Comput. Phys.* 2008;227(11):5361–84.
- [18] Kucharik M, Garimella R, Schofield S, Shashkov M. A comparative study of interface reconstruction methods for multi-material ALE simulations. *J. Comput. Phys.* 2010;229(7):2432–52.
- [19] Tipton RE. CALE Mixed zone pressure relaxation model. Private communication; 1989.
- [20] Shashkov M. Closure models for multimaterial cells in arbitrary lagrangian–eulerian hydrocodes. *Int. J. Numer. Methods Fluids* 2008;56(8):1497–504.
- [21] Barlow A, Hill R, Shashkov M. Constrained optimization framework for interface-aware sub-scale dynamics closure model for multimaterial cells in lagrangian and arbitrary lagrangian-eulerian hydrodynamics. *J. Comput. Phys.* 2014;276:92–135.
- [22] Wilkins M. Calculation of elastic-plastic flow. 1963. Tech. Rep. UCRL-7322, California. Univ., Livermore. Lawrence Radiation Lab.
- [23] Maire P-H, Abgrall R, Breil J, Loubere R, Rebourec B. A nominally second-order cell-centered lagrangian scheme for simulating elastic-plastic flows on two-dimensional unstructured grids. *J. Comput. Phys.* 2013;235:626–65.
- [24] Burton DE, Morgan NR, Charest MRJ, Kenamond MA, Fung J. Compatible, energy conserving, bounds preserving remap of hydrodynamic fields for an extended ALE scheme. *J. Comput. Phys.* 2018;355:492–533.
- [25] Maire PH. Contribution to the numerical modeling of inertial confinement fusion. 2011. Université Bordeaux I, Habilitation.
- [26] Lohmann C. Flux-corrected transport algorithms preserving the eigenvalue range of symmetric tensor quantities. *J. Comput. Phys.* 2017;350:907–26.
- [27] Luttwak G, Falcovitz J. Slope limiting for vectors: a novel vector limiting algorithm. *Int. J. Numer. Methods Fluids* 2011;65(11–12):1365–75.
- [28] Klima M, Kucharik M, Velechovsky J, Shashkov M. Second-invariant-preserving remap of the 2d deviatoric stress tensor in ALE methods. *Computers & Mathematics with Applications* 2019;78(2):654–69.
- [29] Barlow A, Klima M, Shashkov M. Constrained optimization framework for interface-aware sub-scale dynamics models for voids closure in lagrangian hydrodynamics. *J. Comput. Phys.* 2018;371:914–44.
- [30] Kucharik M, Loubere R, Bednarik L, Liska R. Enhancement of lagrangian slide lines as a combined force and velocity boundary condition. *Computers & Fluids* 2013;83:3–14.
- [31] Caramana E. The implementation of slide lines as a combined force and velocity boundary condition. *J. Comput. Phys.* 2009;228(11):3911–16.
- [32] Caramana E, Shashkov M, Whalen P. Formulations of artificial viscosity for multi-dimensional shock wave computations. *J. Comput. Phys.* 1998;144(1):70–97.
- [33] Menikoff R. *Equations of state and fluid dynamics*. 2007. Tech. Rep. LA-UR-07-3989, Los Alamos National Laboratory.
- [34] Barlow AJ. 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.
- [35] Kuzmin D, Lohner R. *Flux corrected transport: principles, algorithms, and applications*, scientific computation series. New York, LLC: Springer-Verlag; 2005.
- [36] Zalesak ST. Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.* 1979;31(3):335–62.
- [37] Nocedal J, Wright S. *Numerical optimization*. Springer; 1999.
- [38] Schittkowski K. *QL: A fortran code for convex quadratic programming – user’s guide*. 2011. Tech. rep., University of Bayreuth, Dept. of Computer Science, <http://www.klaus-schittkowski.de/software.htm>, Accessed 21.11.2016.
- [39] Jemison M, Sussman M, Shashkov M. Filament capturing with the multimaterial moment-of-fluid method. *J. Comput. Phys.* 2015;285:149–72.
- [40] Udaykumar H, Tran L, Belk D, Vanden K. An eulerian method for computation of multimaterial impact with eno shock-capturing and sharp interfaces. *J. Comput. Phys.* 2003;186(1):136–77.
- [41] Toro EF. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Berlin, Heidelberg: Springer Verlag; 1997. ISBN 3-540-61676-4
- [42] Rider W. Void’s unphysical response in hydrocodes, presentation at nuclear explosives code development conference held october 20–24. 2014/2014 in Los Alamos, NM, sAND2014-18762PE. <https://www.osti.gov/scitech/biblio/1241862>.



Appendix C

Original Research Article: Local Error Analysis and Comparison of the Swept- and Intersection-Based Remapping Methods

Authors: Matej Klima, Milan Kucharik, Mikhail Shashkov
Published in Communications in Computational Physics. 2017; vol. 21, pp.
526-558.

Local Error Analysis and Comparison of the Swept- and Intersection-Based Remapping Methods

Matej Klima¹, Milan Kucharik^{1,*} and Mikhail Shashkov²

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]:

$$\tilde{f}_c \tilde{V}_c = f_c V_c + \sum_{c' \in C(c)} F_{c' \rightarrow c} \quad (2.1)$$

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). $F_{c' \rightarrow c}$ 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 $F_{c' \rightarrow c}$ represent mass fluxes between the cells.

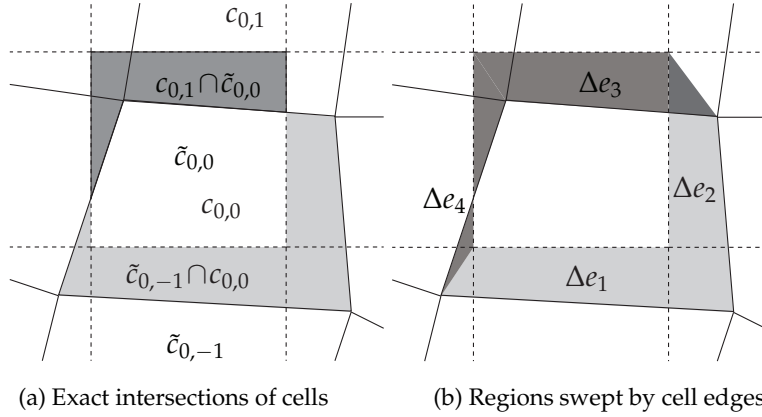


Figure 1: Comparison of flux construction in the remapping methods, dashed line – new mesh, solid line – old mesh, ■ positive flux, ■ negative flux.

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):

$$F_{c' \rightarrow c} = F_{c' \cap \tilde{c}} - F_{c \cap \tilde{c}'}, \quad (2.2)$$

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,

$$F_{c' \cap \tilde{c}} = \int_{c' \cap \tilde{c}} f_c^{\text{rec}}(x, y) dV. \quad (2.3)$$

In the second-order method presented here, a piecewise linear reconstruction is computed in the intersecting region:

$$f_c^{\text{rec}}(x, y) = f_{c'} + \left(\frac{\partial f}{\partial x} \right)_{c'} (x - x_{c'}) + \left(\frac{\partial f}{\partial y} \right)_{c'} (y - y_{c'}), \quad (2.4)$$

where $x_{c'}$, $y_{c'}$ are the coordinates of the geometrical centroid of the respective cell,

$$x_{c'} = \frac{\int_{c'} x dV}{\int_{c'} 1 dV}, \quad y_{c'} = \frac{\int_{c'} y dV}{\int_{c'} 1 dV}. \quad (2.5)$$

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:

$$\tilde{f}_c \tilde{V}_c = f_c V_c + \sum_{e \in E(c)} \int_{\Delta e} f_{c(\Delta e)}^{\text{rec}}(x, y) dV, \quad (2.6)$$

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:

$$f_c = \frac{1}{V_c} \int_c \left(f(\mathbf{r}_c) + \nabla f(\mathbf{r}_c) \cdot (\mathbf{r} - \mathbf{r}_c) + \frac{1}{2} (\mathbf{r} - \mathbf{r}_c)^T \mathbf{H} (\mathbf{r} - \mathbf{r}_c) + \mathcal{O}(\mathbf{r}^3) \right) dV, \quad (3.1)$$

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:

$$f_c = f(\mathbf{r}_c) + \frac{1}{2V_c} \int_c (\mathbf{r} - \mathbf{r}_c)^T \mathbf{H} (\mathbf{r} - \mathbf{r}_c) + \mathcal{O}(\mathbf{r}^3) dV. \quad (3.2)$$

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:

$$I_c^{\mathbf{H}}(\mathbf{r}_c) = \int_c (\mathbf{r} - \mathbf{r}_c)^T \mathbf{H} (\mathbf{r} - \mathbf{r}_c) dV. \quad (3.3)$$

This integral can be evaluated by expanding the quadratic form:

$$I_c^{\mathbf{H}}(\mathbf{r}_c) = \int_c \mathbf{r}^T \mathbf{H} \mathbf{r} - 2\mathbf{r}^T \mathbf{H} \mathbf{r}_c + \mathbf{r}_c^T \mathbf{H} \mathbf{r}_c dV, \quad (3.4)$$

due to the fact that the Hessian is symmetric. The Hessian can be diagonalized in the following way:

$$I_c^{\mathbf{H}}(\mathbf{r}_c) = \int_c \mathbf{r}^T \mathbf{P} \mathbf{A} \mathbf{P}^T \mathbf{r} - 2\mathbf{r}^T \mathbf{H} \mathbf{r}_c + \mathbf{r}_c^T \mathbf{H} \mathbf{r}_c dV, \quad (3.5)$$

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 $\mathbf{u} = \mathbf{P}^T \mathbf{r}$ 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:

$$I_c^{\mathbf{H}}(\mathbf{r}_c) = \int_c \mathbf{u}^T \mathbf{\Lambda} \mathbf{u} dV + \sum_{i=0}^n \begin{vmatrix} x_i & x_{i+1} \\ y_i & y_{i+1} \end{vmatrix} \left(\frac{\mathbf{r}_c}{2} - \frac{\mathbf{r}_i + \mathbf{r}_{i+1}}{3} \right)^T \mathbf{H} \mathbf{r}_c, \quad (3.6)$$

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:

$$\int_c \mathbf{u}^T \mathbf{\Lambda} \mathbf{u} dV = \frac{1}{12} \sum_{i=0}^n \begin{vmatrix} u_{x,i} & u_{x,i+1} \\ u_{y,i} & u_{y,i+1} \end{vmatrix} \begin{pmatrix} u_{x,i}^2 + u_{x,i} u_{x,i+1} + u_{x,i+1}^2 \\ u_{y,i}^2 + u_{y,i} u_{y,i+1} + u_{y,i+1}^2 \end{pmatrix} \cdot \begin{pmatrix} k_1 \\ k_2 \end{pmatrix}, \quad (3.7)$$

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} :

$$\int_c \mathbf{r}^T \mathbf{H} \mathbf{r} dV = \frac{1}{12} \sum_{i=0}^n \begin{vmatrix} x_i & x_{i+1} \\ y_i & y_{i+1} \end{vmatrix} \left(\mathbf{r}_i^T \mathbf{H} \mathbf{r}_i + \mathbf{r}_i^T \mathbf{H} \mathbf{r}_{i+1} + \mathbf{r}_{i+1}^T \mathbf{H} \mathbf{r}_{i+1} \right) \quad (3.8)$$

and use this result in (3.6). After simplification, we obtain:

$$I_c^{\mathbf{H}}(\mathbf{r}_c) = \frac{1}{12} \sum_{i=0}^n \|\mathbf{r}_i \times \mathbf{r}_{i+1}\|^2 \left((\mathbf{r}_i - \mathbf{r}_c)^T \mathbf{H} (\mathbf{r}_i - \mathbf{r}_c) + (\mathbf{r}_i - 2\mathbf{r}_c)^T \mathbf{H} (\mathbf{r}_{i+1} - 2\mathbf{r}_c) + (\mathbf{r}_{i+1} - \mathbf{r}_c)^T \mathbf{H} (\mathbf{r}_{i+1} - \mathbf{r}_c) \right). \quad (3.9)$$

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:

$$\epsilon_{\text{remap}} = \tilde{f}_c - \frac{1}{\tilde{V}_c} \int_{\tilde{c}} f(\mathbf{r}) dV = \frac{1}{\tilde{V}_c} \left(f_c V_c + \int_F f_c^{\text{rec}}(\mathbf{r}) dV - \int_{\tilde{c}} f(\mathbf{r}) dV \right), \quad (3.10)$$

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:

$$\epsilon_{\text{remap}} = \frac{1}{\tilde{V}_c} \left(\int_F f_{c'}^{\text{rec}}(\mathbf{r}) dV - \int_F f(\mathbf{r}) dV \right). \quad (3.11)$$

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:

$$\epsilon_{\text{remap}} = \frac{1}{\tilde{V}_c} \left(\int_F f_{c'} - f(\mathbf{r}_{c'}) - \frac{1}{2}(\mathbf{r} - \mathbf{r}_{c'})^T \mathbf{H}(\mathbf{r} - \mathbf{r}_{c'}) dV \right). \quad (3.12)$$

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:

$$\epsilon_{\text{remap}} = \frac{V_F}{2\tilde{V}_c} \left(\frac{I_{c'}^{\mathbf{H}}(\mathbf{r}_{c'})}{V_{c'}} - \frac{I_F^{\mathbf{H}}(\mathbf{r}_{c'})}{V_F} \right), \quad (3.13)$$

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. $|\mathbf{H}| < 0 \dots$ 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 ($|\mathbf{H}| < 0$):

$$\epsilon_{\text{remap}} \sim \frac{V_F}{2\tilde{V}_c} \left(\frac{I_{c'}^{\mathbf{H}}(\mathbf{r}_{c'})}{V_{c'}} - (\mathbf{r}_F - \mathbf{r}_{c'})^T \mathbf{H}(\mathbf{r}_F - \mathbf{r}_{c'}) \right). \quad (3.14)$$

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 re-zoning 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:

$$\epsilon_{\text{sw}} = \frac{1}{2\tilde{V}_{c_{0,0}}} \left(\frac{V_{F_N+F_{NE1}}}{V_{c_{0,1}}} I_{c_{0,1}}^{\mathbf{H}}(\mathbf{r}_{c_{0,1}}) - I_{F_N+F_{NE1}}^{\mathbf{H}}(\mathbf{r}_{c_{0,1}}) + \frac{V_{F_E+F_{NE2}}}{V_{c_{1,0}}} I_{c_{1,0}}^{\mathbf{H}}(\mathbf{r}_{c_{1,0}}) - I_{F_E+F_{NE2}}^{\mathbf{H}}(\mathbf{r}_{c_{1,0}}) \right), \quad (3.15)$$

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:

$$\epsilon_{\text{int}} = \frac{1}{2\tilde{V}_{c_{0,0}}} \left(\frac{V_{F_N}}{V_{c_{0,1}}} I_{c_{0,1}}^{\mathbf{H}}(\mathbf{r}_{c_{0,1}}) - I_{F_N}^{\mathbf{H}}(\mathbf{r}_{c_{0,1}}) + \frac{V_{F_{NE1}+F_{NE2}}}{V_{c_{1,1}}} I_{c_{1,1}}^{\mathbf{H}}(\mathbf{r}_{c_{1,1}}) - I_{F_{NE1}+F_{NE2}}^{\mathbf{H}}(\mathbf{r}_{c_{1,1}}) + \frac{V_{F_E}}{V_{c_{1,0}}} I_{c_{1,0}}^{\mathbf{H}}(\mathbf{r}_{c_{1,0}}) - I_{F_E}^{\mathbf{H}}(\mathbf{r}_{c_{1,0}}) \right). \quad (3.16)$$

Now, we would like to express the difference between the two errors to analyze, which method is more accurate:

$$\Delta\epsilon = |\epsilon_{\text{int}}| - |\epsilon_{\text{sw}}|, \quad |\Delta\epsilon| \leq |\epsilon_{\text{int}} - \epsilon_{\text{sw}}|. \quad (3.17)$$

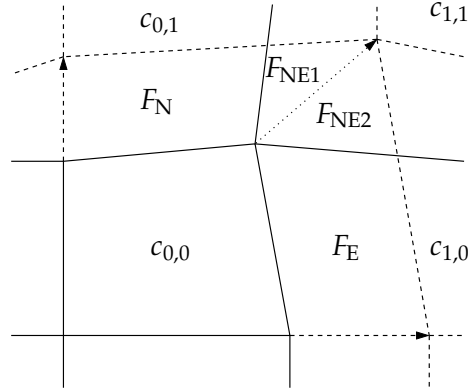


Figure 2: Upper right corner movement, dashed line – new mesh, solid line – old mesh.

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:

$$|\Delta\epsilon| \leq \frac{1}{2\tilde{V}_{c_{0,0}}} \left| \frac{V_{F_{NE1}+F_{NE2}}}{V_{c_{1,1}}} I_{c_{1,1}}^{\mathbf{H}}(\mathbf{r}_{c_{1,1}}) - \frac{V_{F_{NE1}}}{V_{c_{0,1}}} I_{c_{0,1}}^{\mathbf{H}}(\mathbf{r}_{c_{0,1}}) - \frac{V_{F_{NE2}}}{V_{c_{1,0}}} I_{c_{1,0}}^{\mathbf{H}}(\mathbf{r}_{c_{1,0}}) - I_{F_{NE1}+F_{NE2}}^{\mathbf{H}}(\mathbf{r}_{c_{1,1}}) + I_{F_{NE1}}^{\mathbf{H}}(\mathbf{r}_{c_{0,1}}) + I_{F_{NE2}}^{\mathbf{H}}(\mathbf{r}_{c_{1,0}}) \right|. \quad (3.18)$$

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 .

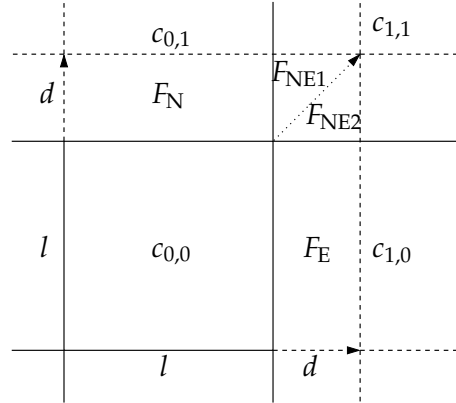


Figure 3: Diagonal movement of orthogonal corner, dashed line – new mesh, solid line – old mesh.

The difference between the methods is dependent on the area of the corner flux regions $V_{F_{NE1}+F_{NE2}}$ 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:

$$\epsilon_{sw} = -\frac{d}{12\tilde{V}_{c_{0,0}}} [(2d^3 + ld^2 - 2dl^2 + l^3)(\rho_{xx} + \rho_{yy}) + (3d^3 + 2ld^2 - 3dl^2)\rho_{xy}], \quad (3.19)$$

$$\epsilon_{int} = -\frac{d}{12\tilde{V}_{c_{0,0}}} [(2d^3 - ld^2 - 2dl^2 + l^3)(\rho_{xx} + \rho_{yy}) + (3d^3 - 6ld^2 + 3dl^2)\rho_{xy}], \quad (3.20)$$

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:

$$x' = \frac{1}{\sqrt{2}}(x+y), \quad y' = \frac{1}{\sqrt{2}}(y-x). \quad (3.21)$$

We can transform the second derivatives similarly by multiplying them by the Jacobian matrix of the transformation,

$$\rho_{x'x'} = \frac{1}{2}(\rho_{xx} + \rho_{yy}) + \rho_{xy}, \quad \rho_{x'y'} = \frac{1}{2}(\rho_{xx} - \rho_{yy}), \quad \rho_{y'y'} = \frac{1}{2}(\rho_{xx} + \rho_{yy}) - \rho_{xy}. \quad (3.22)$$

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 \mathbf{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

$$\epsilon_{\text{sw}} = -\frac{d}{24\tilde{V}_{c_{0,0}}} [(7d^3 + 4ld^2 - 7dl^2 + 2l^3)\rho_{x'x'} + (d^3 - dl^2 + 2l^3)\rho_{y'y'}], \quad (3.23)$$

$$\epsilon_{\text{int}} = -\frac{d}{24\tilde{V}_{c_{0,0}}} [(7d^3 - 8ld^2 - dl^2 + 2l^3)\rho_{x'x'} + (d^3 + 4ld^2 - 7dl^2 + 2l^3)\rho_{y'y'}]. \quad (3.24)$$

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):

$$\Delta\epsilon = |K_1| (|g(t^3 - t + 2) + 7t^3 + 4t^2 - 7t + 2| - |g(t^3 + 4t^2 - 7t + 2) + 7t^3 - 8t^2 - t + 2|). \quad (3.25)$$

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:

$$|\Delta\epsilon| \leq \frac{d^2l}{12\tilde{V}_{c_{0,0}}} |(3l - 6d)\rho_{x'x'} + (2d - 3l)\rho_{y'y'}|. \quad (3.26)$$

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

$$|\Delta\epsilon| \leq \frac{V_{F_{\text{NE}}}}{12\tilde{V}_{\tilde{c}}} |3(V_{F_{\text{NE}}} + 2V_{c_{0,0}} - \tilde{V}_{c_{0,0}})\rho_{x'x'} - (\tilde{V}_{c_{0,0}} - 4V_{c_{0,0}} - V_{F_{\text{NE}}})\rho_{y'y'}|. \quad (3.27)$$

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.

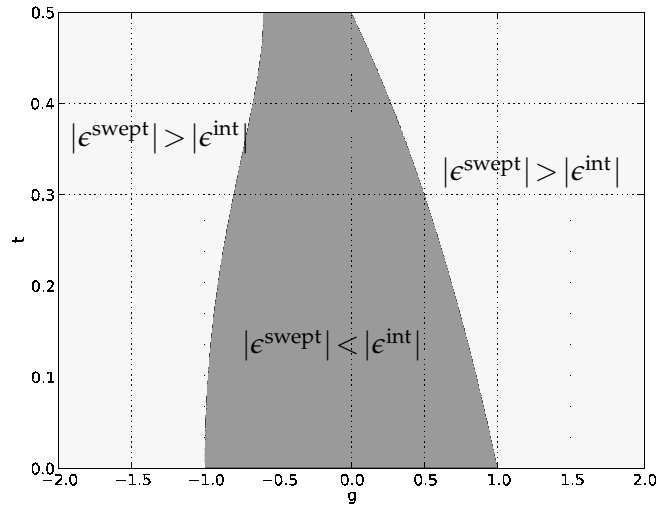


Figure 4: Two grid-lines motion – method preference. Different colors show regions where each method is more accurate.

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

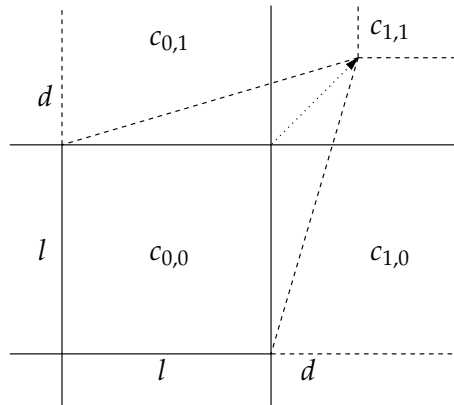


Figure 5: Diagonal one point corner movement, dashed line – new mesh, solid line – old mesh.

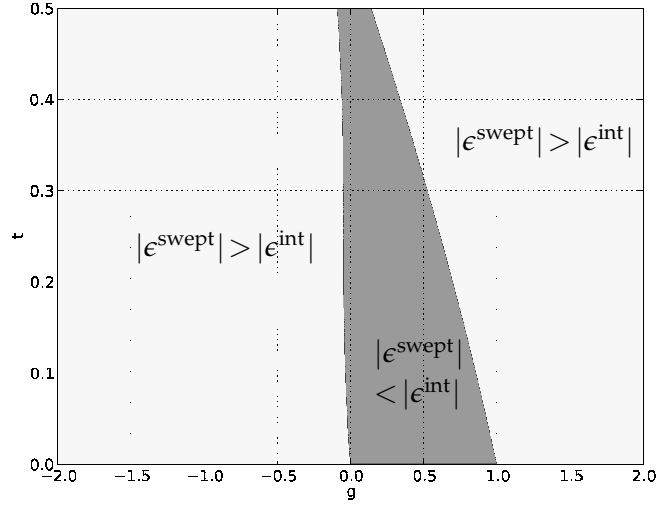


Figure 6: Single vertex movement – method preference. Different colors show regions where each method is more accurate.

process as described in (3.22):

$$\epsilon_{\text{sw}} = -\frac{d}{12(l+d)} [(2d^2 - ld)\rho_{x'x'} + l^2\rho_{y'y'}], \quad (3.28)$$

$$\epsilon_{\text{int}} = -\frac{d}{12(l+d)^3} [(2d^4 - d^3l - 3d^2l^2 + 2dl^3)\rho_{x'x'} + (l^4 - dl^3)\rho_{y'y'}]. \quad (3.29)$$

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:

$$\Delta\epsilon = |K_2| [(1+t)^2|(2t^2 - t + g)| - |g(t-1) - 2t^4 + t^3 + 3t^2 - 2t|]. \quad (3.30)$$

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:

$$|\Delta\epsilon| \leq \frac{d^2l}{12(l+d)^3} |(3l^2 - 3ld - 4d^2)\rho_{x'x'} - (3l^2 + ld)\rho_{y'y'}|, \quad (3.31)$$

and after substituting for volumes where possible, we can write

$$|\Delta\epsilon| \leq \frac{1}{24} \left(\frac{V_{\text{FNE}}}{V_{\tilde{C}}} \right)^{\frac{3}{2}} |(9V_{c_{0,0}} - 3\tilde{V}_{c_{0,0}} - 5V_{\text{FNE}})\rho_{x'x'} - (\tilde{V}_{c_{0,0}} + 5V_{c_{0,0}} - V_{\text{FNE}})\rho_{y'y'}|. \quad (3.32)$$

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 pattern 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.

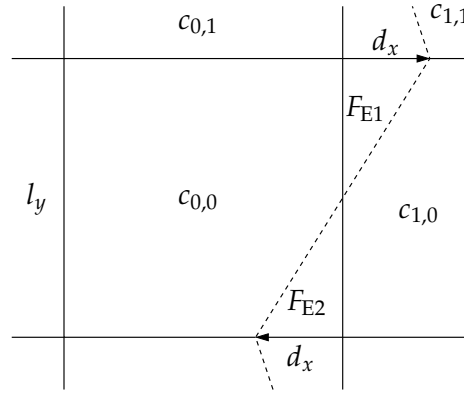


Figure 7: Hourglass movement, dashed line – new mesh, solid line – old mesh.

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:

$$\epsilon_{sw} = \frac{1}{2\tilde{V}_{c_{0,0}}} \left(\frac{V_{F_{E1}-F_{E2}}}{V_{c_{1,0}}} I_{c_{1,0}}^{\mathbf{H}}(\mathbf{r}_{c_{1,0}}) - I_{F_{E1}-F_{E2}}^{\mathbf{H}}(\mathbf{r}_{c_{1,0}}) \right). \quad (3.33)$$

The swept method is constructed so that it perceives the self-intersecting polygon $F_{E1} - F_{E2}$ 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:

$$\epsilon_{int} = \frac{1}{2\tilde{V}_{c_{0,0}}} \left(\frac{V_{F_{E1}}}{V_{c_{1,0}}} I_{c_{1,0}}^{\mathbf{H}}(\mathbf{r}_{c_{1,0}}) - I_{F_{E1}}^{\mathbf{H}}(\mathbf{r}_{c_{1,0}}) - \frac{V_{F_{E2}}}{V_{c_{0,0}}} I_{c_{0,0}}^{\mathbf{H}}(\mathbf{r}_{c_{0,0}}) + I_{F_{E2}}^{\mathbf{H}}(\mathbf{r}_{c_{0,0}}) \right). \quad (3.34)$$

We can also express the error difference magnitude, similarly as for the corner movement:

$$|\Delta\epsilon| \leq \frac{V_{F_{E2}}}{2\tilde{V}_{c_{0,0}}} \left| \frac{I_{c_{1,0}}^{\mathbf{H}}(\mathbf{r}_{c_{1,0}})}{V_{c_{1,0}}} - \frac{I_{F_{E2}}^{\mathbf{H}}(\mathbf{r}_{c_{1,0}})}{V_{F_{E2}}} - \frac{I_{c_{0,0}}^{\mathbf{H}}(\mathbf{r}_{c_{0,0}})}{V_{c_{0,0}}} + \frac{I_{F_{E2}}^{\mathbf{H}}(\mathbf{r}_{c_{0,0}})}{V_{F_{E2}}} \right|. \quad (3.35)$$

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 $I_{c_{1,0}}^{\mathbf{H}}(\mathbf{r}_{c_{1,0}}) \approx I_{c_{0,0}}^{\mathbf{H}}(\mathbf{r}_{c_{0,0}})$. 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 $I_{F_{E1}}^{\mathbf{H}}(\mathbf{r}_{c_{1,0}}) = I_{F_{E2}}^{\mathbf{H}}(\mathbf{r}_{c_{0,0}})$. 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:

$$\epsilon_{\text{sw}} = \frac{d_x}{12} (d_x \rho_{xx} + l_y \rho_{xy}), \quad \epsilon_{\text{int}} = 0, \quad (3.36)$$

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:

$$\epsilon_{\text{sw}} = \frac{d_x}{6} (d_x \rho_{xx} + l_y \rho_{xy}). \quad (3.37)$$

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.

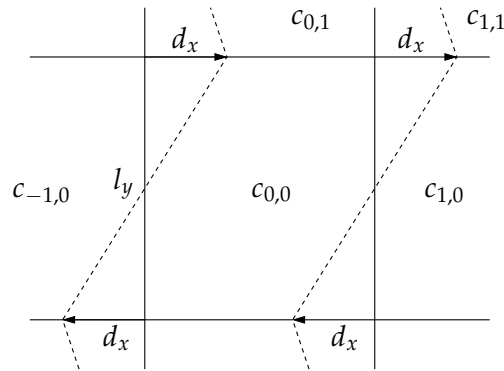


Figure 8: Symmetric cell shear, dashed line – new mesh, solid line – old mesh.

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

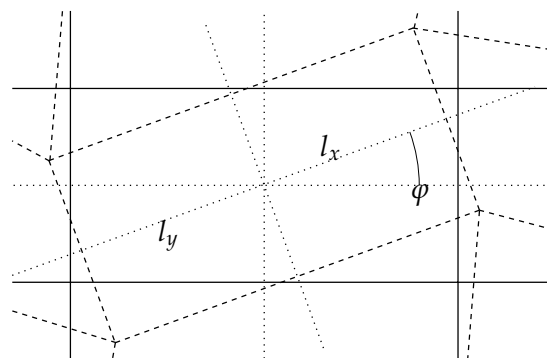


Figure 9: Cell rotation around its centroid by angle φ , dashed line – new mesh, solid line – old mesh.

respectively:

$$\epsilon_{\text{int}} = \frac{1}{24} (l_x^2 - l_y^2) \sin \varphi \left((\rho_{xx} - \rho_{yy}) \sin \varphi - \rho_{xy} \cos \varphi \right), \quad (3.38)$$

$$\epsilon_{\text{sw}} = \frac{1}{12} (1 - \cos \varphi) \left((\rho_{xx} l_x^2 + \rho_{yy} l_y^2) \cos \varphi + \rho_{xy} (l_x^2 - l_y^2) \sin \varphi \right). \quad (3.39)$$

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 \mathbf{H} is diagonal). We presume that \mathbf{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:

$$(\rho_{xx} - \rho_{yy}) \sin 2\psi = 2\rho_{xy} \cos 2\psi, \quad \psi \in \left(-\frac{\pi}{4}, \frac{\pi}{4} \right). \quad (3.40)$$

Note that this transformation can change the order of the eigenvalues. The derivatives in the principal directions ρ_{dd}, ρ_{pp} (the eigenvalues of \mathbf{H}) can be expressed as follows:

$$\rho_{dd} = \rho_{xx} \cos^2 \psi + 2\rho_{xy} \sin \psi \cos \psi + \rho_{yy} \sin^2 \psi, \quad (3.41)$$

$$\rho_{pp} = \rho_{xx} \sin^2 \psi - 2\rho_{xy} \sin \psi \cos \psi + \rho_{yy} \cos^2 \psi. \quad (3.42)$$

By substituting these transformations in (3.38), (3.39), we yield the following error terms:

$$\epsilon_{\text{int}} = \frac{1}{24} (l_x^2 - l_y^2) (\rho_{dd} - \rho_{pp}) \sin \varphi \sin(\varphi - 2\psi), \quad (3.43)$$

and

$$\begin{aligned} \epsilon_{\text{sw}} = \frac{1}{24} (1 - \cos \varphi) & \left((\rho_{dd} + \rho_{pp}) (l_x^2 + l_y^2) \cos \varphi \right. \\ & \left. + (\rho_{dd} - \rho_{pp}) (l_x^2 - l_y^2) \cos(\varphi - 2\psi) \right). \end{aligned} \quad (3.44)$$

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:

$$\epsilon_{\text{sw},\square} = \frac{1}{12} l^2 (\rho_{dd} + \rho_{pp}) (1 - \cos \varphi) \cos \varphi, \quad (3.45)$$

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:

$$L_1 = \frac{\sum_{c \in C} |f_c - \bar{f}_c|}{\sum_{c \in C} \bar{f}_c}, \quad \bar{f}_c = \frac{\int_c f(\mathbf{r}) dV}{V_c}, \quad (4.1)$$

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

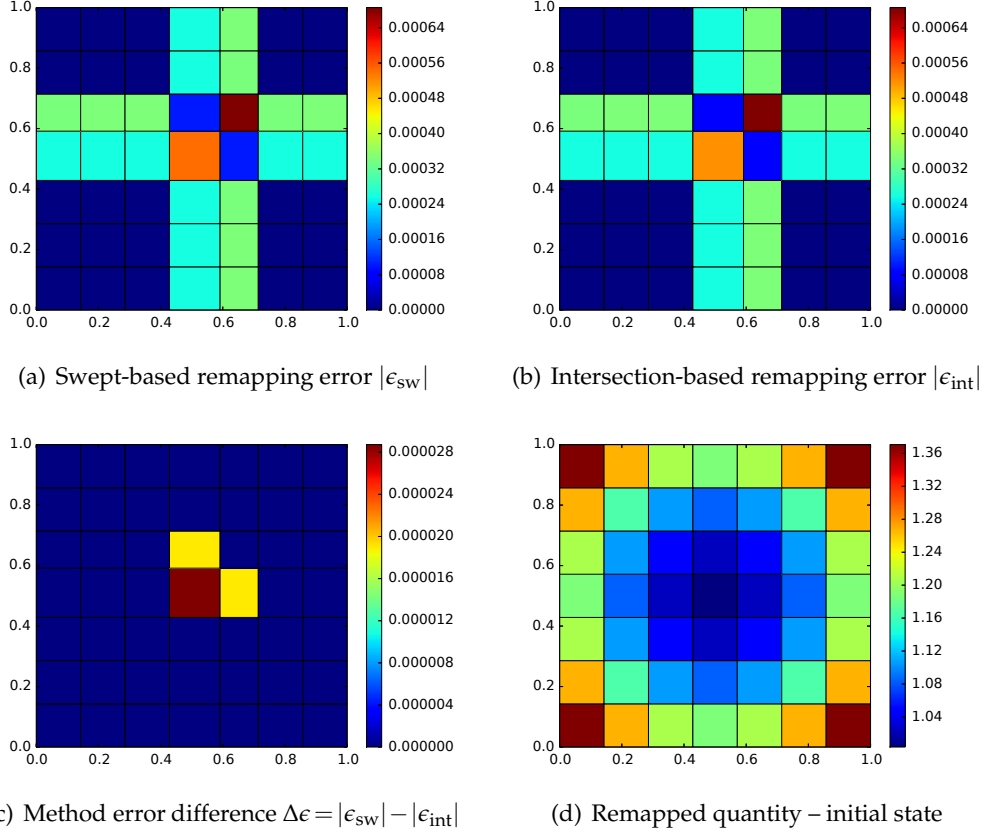


Figure 10: Numerical error of both remapping methods compared. Initial distribution function $(x-1/2)^2 + (y-1/2)^2 + 1$, 7×7 mesh, one step of a diagonal movement.

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}
$c_{4,4}$	$-5.460 \cdot 10^{-4}$	$-5.173 \cdot 10^{-4}$
$c_{5,4}$	$1.033 \cdot 10^{-4}$	$8.421 \cdot 10^{-5}$
$c_{4,5}$	$1.033 \cdot 10^{-4}$	$8.421 \cdot 10^{-5}$
$c_{5,5}$	$6.857 \cdot 10^{-4}$	$6.857 \cdot 10^{-4}$

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:

$$\begin{aligned} x_i^n &= x_i^0(1-d^n) + (x_i^0)^3 d^n, & y_i^n &= y_i^0(1-d^n) + (y_i^0)^2 d^n, \\ d^n &= \frac{1}{2}\sin(2\pi t^n), & t^n &= n/n_{\max}, & n_{\max} &= 100, \end{aligned} \quad (4.2)$$

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 $5.786 \cdot 10^{-6}$, which is slightly worse than the numerical error of the intersection-based method $4.920 \cdot 10^{-6}$. 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:

$$\frac{\partial f}{\partial x}(0) = \frac{\partial f}{\partial x}(1) = 2x+2, \quad \frac{\partial f}{\partial y}(0) = \frac{\partial f}{\partial y}(1) = 0. \quad (4.3)$$

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.

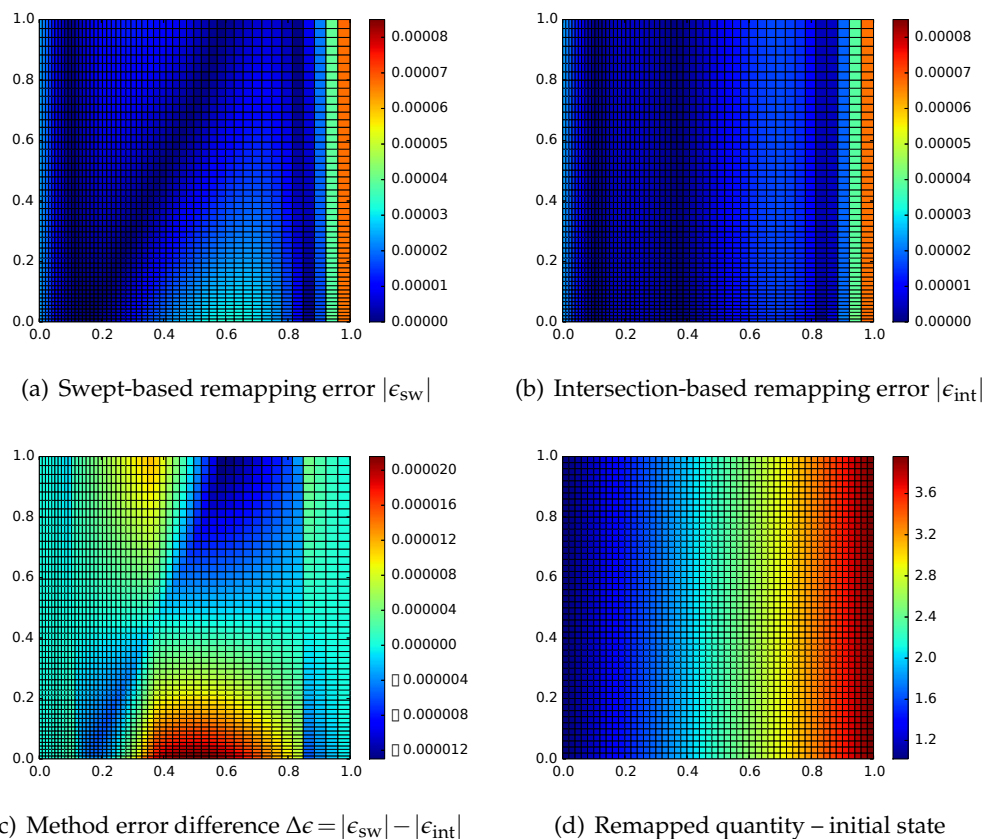


Figure 11: Numerical remapping error of both methods compared. Initial distribution function $(x+1)^2$, 50×50 mesh, step $n=25$ of the tensor product movement.

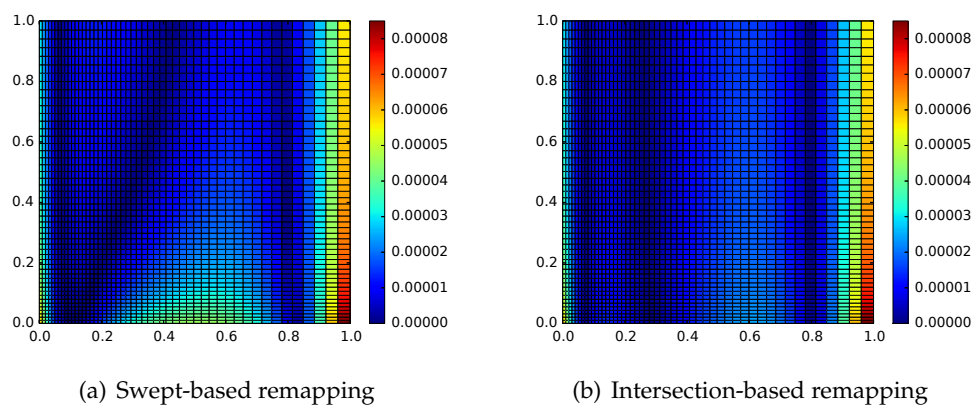


Figure 12: Estimate of the second-order remapping error. Initial distribution function $(x+1)^2$, 50×50 mesh, step $n=25$ of the tensor product movement.

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:

$$f = 10 \left(\text{sinc}(24r - 8) + 2 \right) = 10 \left(\frac{\sin(24r - 8)}{24r - 8} + 2 \right), \quad (4.4)$$

$$r = \sqrt{(x - 0.5)^2 + (y - 0.5)^2}. \quad (4.5)$$

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 $9.532 \cdot 10^{-3}$ for the swept-based method and $9.586 \cdot 10^{-3}$ 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 $m = \text{sw}/\text{int}$ 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 $10(\text{sinc}(24r - 8) + 2)$, 50×50 mesh, step $n = 100$ of the tensor product movement.

m	sw	int
$L_{1,m}^1$	$9.351 \cdot 10^{-3}$	$1.045 \cdot 10^{-2}$
$L_{1,m}^2$	$9.305 \cdot 10^{-3}$	$8.676 \cdot 10^{-3}$
$L_{1,m}^3$	$8.056 \cdot 10^{-3}$	$8.892 \cdot 10^{-3}$
$L_{1,m}^4$	$1.142 \cdot 10^{-2}$	$1.033 \cdot 10^{-2}$
$ \max_{Q=1..4} (L_{1,m}^Q) - \min_{Q=1..4} (L_{1,m}^Q) $	$3.364 \cdot 10^{-3}$	$1.774 \cdot 10^{-3}$

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

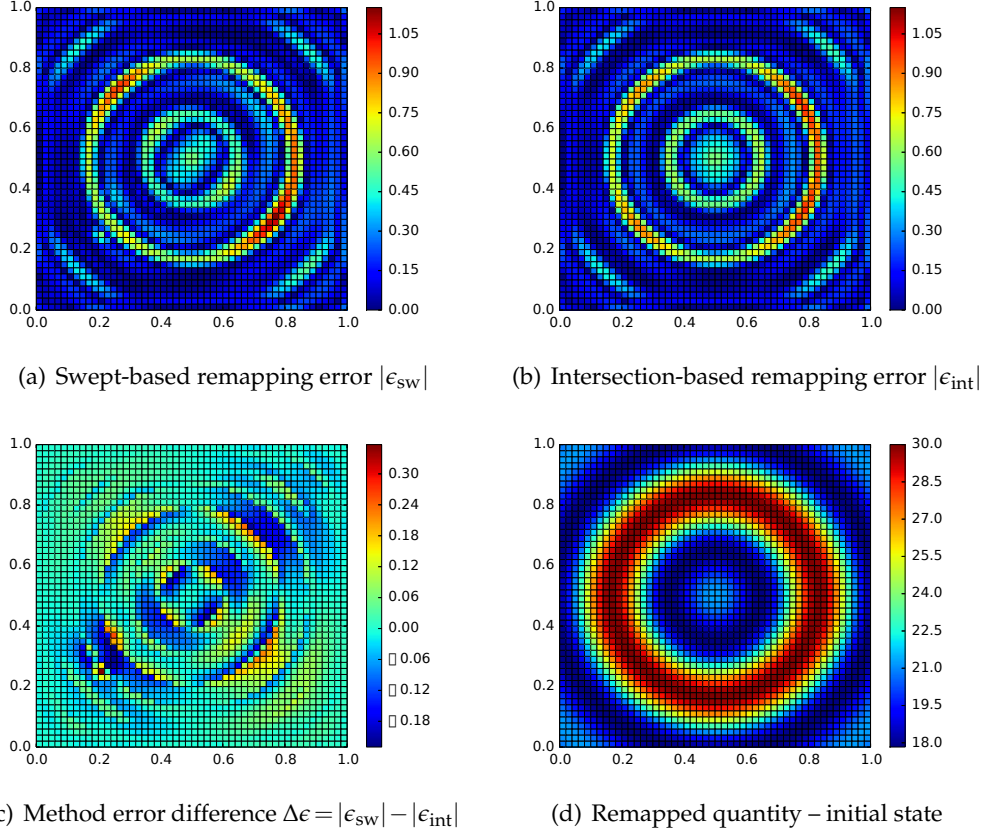


Figure 13: Remapping error of both methods compared. Initial distribution function $10(\text{sinc}(24r-8)+2)$, 50×50 mesh, step $n=100$ of the tensor product movement.

Table 3: L_1 remapping error and its σ_{L_1} standard deviation (in %) for smooth function $10(\text{sinc}(24r-8)+2)$ 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	$9.532 \cdot 10^{-3}$	$9.586 \cdot 10^{-3}$	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,

$$\sigma_{L_1}^m = \frac{1}{L_{1,m}} \sqrt{\frac{1}{4} \sum_{Q=1}^4 (L_{1,m}^Q - L_{1,m})^2}, \quad (4.6)$$

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 $10(\text{sinc}(24r-8)+2)$ 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 of 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,

$$f(r) = \begin{cases} 1 + e^{10r}, & \text{for } x \leq 1/4, \\ 1 + e^{6r-1/4}, & \text{otherwise,} \end{cases} \quad (4.7)$$

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,

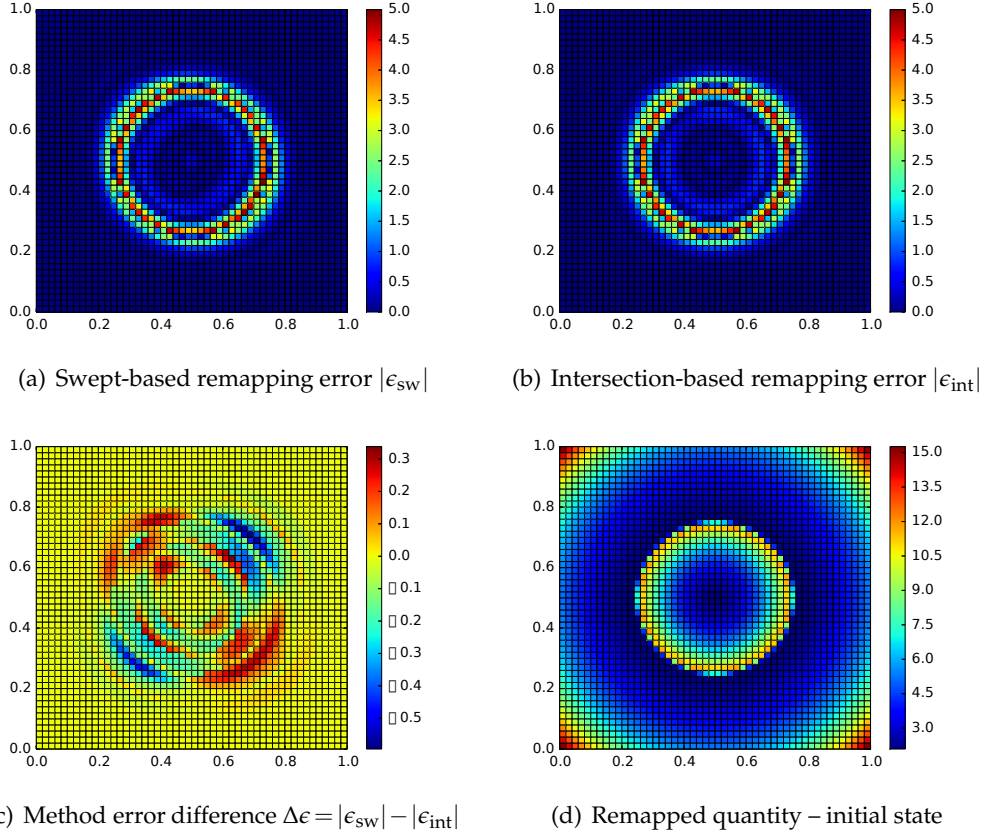


Figure 14: Remapping error of both methods compared. Initial distribution function – double exponential, 50×50 mesh, step $n=100$ of the tensor product movement.

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	$7.702 \cdot 10^{-2}$	$7.873 \cdot 10^{-2}$	9.4	2.7
100×100	200	$4.859 \cdot 10^{-2}$	$4.940 \cdot 10^{-2}$	10.7	2.9
200×200	400	$2.930 \cdot 10^{-2}$	$2.982 \cdot 10^{-2}$	10.9	2.9
400×400	800	$1.750 \cdot 10^{-2}$	$1.782 \cdot 10^{-2}$	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:

$$\frac{1}{\rho}\rho_t + \nabla \cdot \mathbf{u} = 0, \quad (4.8)$$

$$\rho(\mathbf{u}_x)_t + \nabla p = 0, \quad \rho(\mathbf{u}_y)_t + \nabla p = 0, \quad (4.9)$$

$$\rho\varepsilon_t + p\nabla \cdot \mathbf{u} = 0, \quad (4.10)$$

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,

$$p = \rho\varepsilon(\gamma - 1). \quad (4.11)$$

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 $\varepsilon_{0,0}$ 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 $\langle 0,1 \rangle^2$ 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.

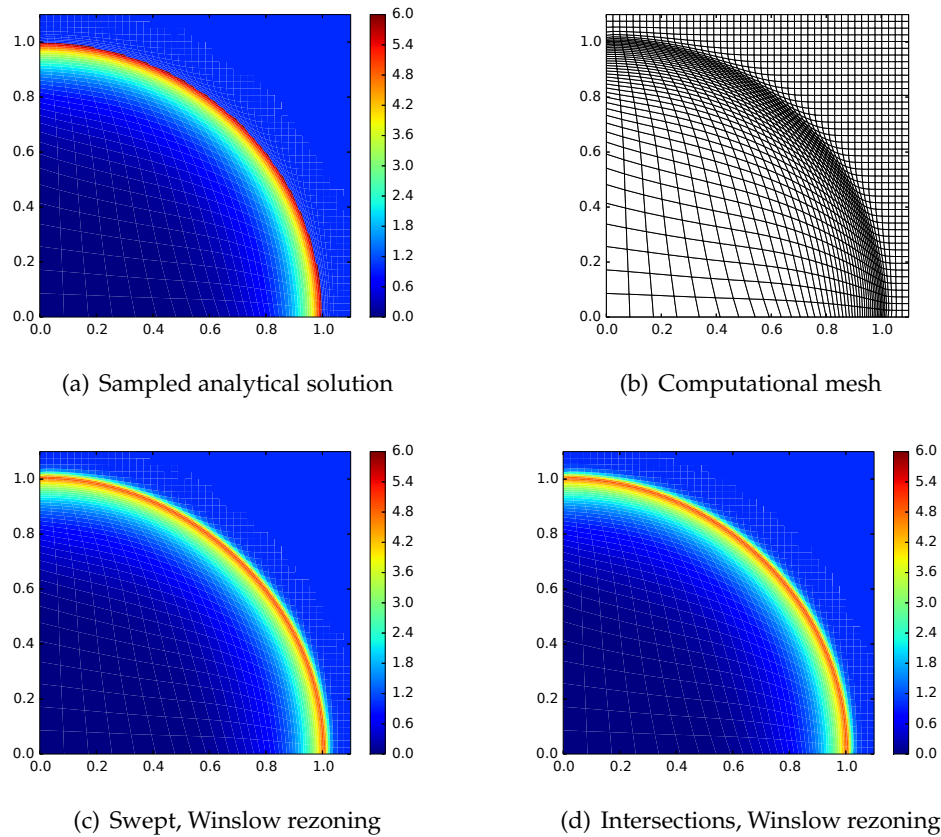
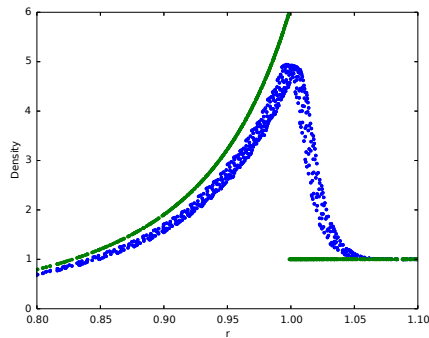


Figure 15: Sedov problem, 2D density profile (analytical and numerical), 45×45 mesh, $t = 1.0$.

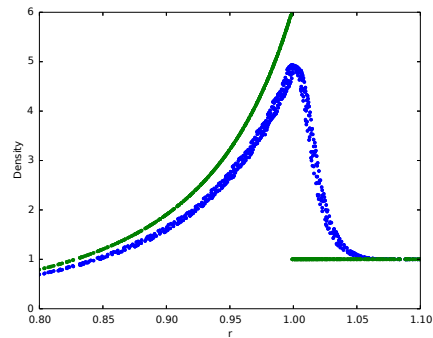
5 Conclusion

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.

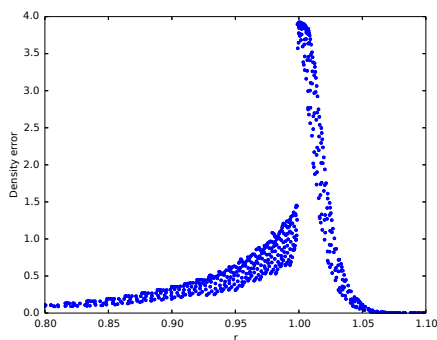
If the initial quantity distribution is symmetric, then although the total remapping error is similar, its variance and spatial distribution is apparently less favorable in the case



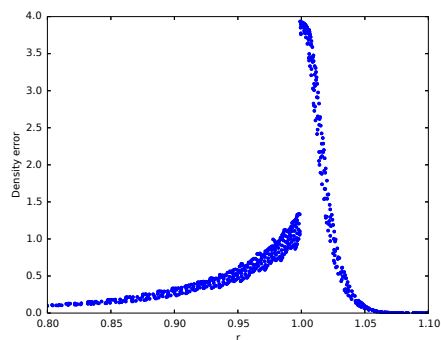
(a) Density, swept-based remapping



(b) Density, intersection-based remapping

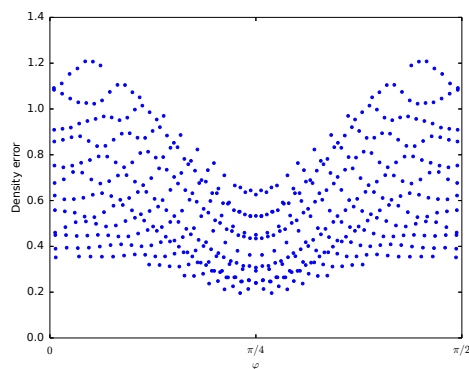


(c) Error, swept-based remapping

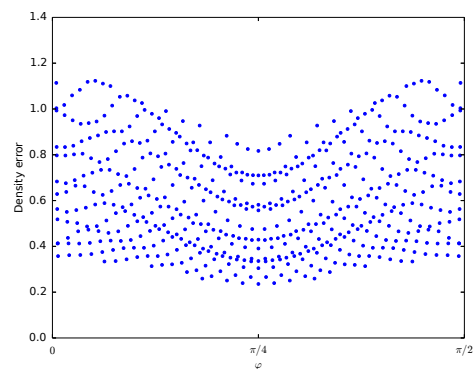


(d) Error, intersection-based remapping

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, Raphaël 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.






Appendix D

Original Research Article: Combined swept region and intersection-based single-material remapping method

Authors: Matej Klima, Milan Kucharik, Mikhail Shashkov
Published in International Journal of Numerical Methods in Fluids. 2017;
vol. 85, pp. 363-382.

Combined swept region and intersection-based single-material remapping method

Matej Klima¹  | Milan Kucharik¹  | Mikhail Shashkov² 

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

²XCP-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 regions⁴ 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

$$\rho_{\tilde{c}} V_{\tilde{c}} = \rho_c V_c + \sum_{c' \in S(c)} F_{c,c'}^m, \quad F_{c,c'}^m = -F_{c',c}^m, \quad (1)$$

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 $F_{c,c'}^m$ 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:

$$m_{\tilde{c}} = \sum_{c' \in S(c) \cup \{c\}} m_{\tilde{c} \cap c'}. \quad (2)$$

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:

$$m_{\tilde{c}} = m_c + \sum_{c' \in S(c)} m_{\tilde{c} \cap c'} - \sum_{c' \in S(\tilde{c})} m_{c' \cap c}. \quad (3)$$

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:

$$m_{\tilde{c} \cap c'} = \iint_{c' \cap \tilde{c}} \left[\rho_{c'} + \left(\frac{\partial \rho}{\partial x} \right)_{c'} (x - x_{c'}) + \left(\frac{\partial \rho}{\partial y} \right)_{c'} (y - y_{c'}) \right] dx dy, \quad (4)$$

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.

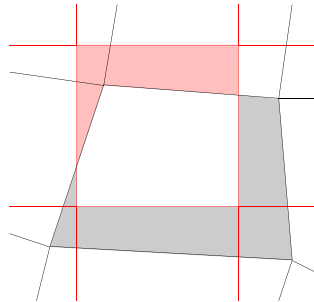


FIGURE 1 Flux construction using intersections of old and new cells. — new mesh, — old mesh, ■ positive flux, ■ negative flux [Colour figure can be viewed at wileyonlinelibrary.com]

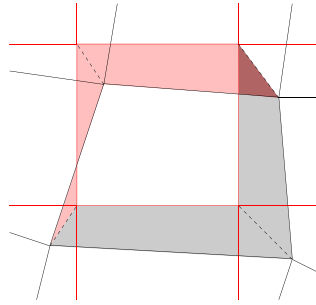


FIGURE 2 Flux construction using regions swept by the cell edges. — new mesh, — old mesh, ■ positive flux, ■ negative flux, ■ overlapping regions [Colour figure can be viewed at wileyonlinelibrary.com]

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}:

$$m_{\bar{e}} = m_c + \sum_{e \in E(c)} \iint_{\Delta e} \left[\rho_{c'} + \left(\frac{\partial \rho}{\partial x} \right)_{c'} (x - x_{c'}) + \left(\frac{\partial \rho}{\partial y} \right)_{c'} (y - y_{c'}) \right] dx dy. \quad (5)$$

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 process¹⁰ 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 in²⁴ – 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

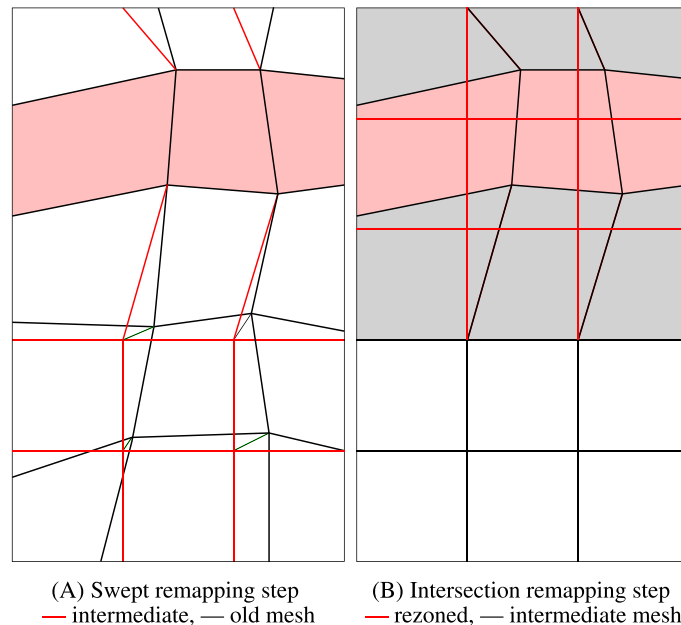


FIGURE 3 Mesh construction in two steps of our adaptive hybrid remapping method. A, Swept remapping step and B, intersection remapping step mesh. ■ cells marked for intersections, ■ buffer cells [Colour figure can be viewed at wileyonlinelibrary.com]

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 methods¹⁵ 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:

$$\frac{\sqrt{\left(\frac{\partial \rho}{\partial x}\right)_c^2 + \left(\frac{\partial \rho}{\partial y}\right)_c^2}}{\rho_{\max}} > \alpha_{GS}, \quad (6)$$

where ρ_{\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):

$$\frac{\sqrt{|\det \mathbf{H}(\rho)|}}{\rho_{\max}} > \alpha_{HS}. \quad (7)$$

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:

$$\rho_{\mathbf{d}\mathbf{d}} = \frac{\mathbf{d} \cdot \nabla (\mathbf{d} \cdot \nabla \rho)}{\|\mathbf{d}\|^2} = \frac{\mathbf{d} \mathbf{H}(\rho) \mathbf{d}^T}{\mathbf{d}^2}, \quad \rho_{\mathbf{p}\mathbf{p}} = \frac{\mathbf{p} \mathbf{H}(\rho) \mathbf{p}^T}{\mathbf{p}^2}, \quad (8)$$

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:

$$\frac{4 V_d}{V_c} \left| \frac{\rho_{\mathbf{p}\mathbf{p}}}{\rho_{\mathbf{d}\mathbf{d}}} \right| > \alpha_{DDS}, \quad (9)$$

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 $\rho_{\mathbf{p}\mathbf{p}} = 0$, we always use swept-based remap, even when $\rho_{\mathbf{d}\mathbf{d}} = 0$ as well (constant gradient, no curvature). If $\rho_{\mathbf{p}\mathbf{p}} \neq 0$ and $\rho_{\mathbf{d}\mathbf{d}} = 0$, 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:

$$\rho_{\mathbf{g}\mathbf{g}} = \frac{\mathbf{g} \mathbf{H}(\rho) \mathbf{g}^T}{\mathbf{g}^2}, \quad \mathbf{g} = \mathbf{x}_i - \mathbf{x}_c, \quad (10)$$

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:

$$\frac{4 V_d |\rho_{\mathbf{g}\mathbf{g}}|}{V_c \rho_{\max}} > \alpha_{DGS}. \quad (11)$$

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 from⁷, 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:

$$f(x, y) = \begin{cases} 1 + e^{10\sqrt{(x-1/2)^2+(y-1/2)^2}} & \text{for } \sqrt{(x-1/2)^2+(y-1/2)^2} \leq 1/4, \\ 1 + e^{6\sqrt{(x-1/2)^2+(y-1/2)^2}-1/4} & \text{in other cases.} \end{cases} \quad (12)$$

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 perpendicular⁴. The inner mesh vertices thus move in diagonal directions. This movement is defined as follows:

$$\begin{aligned} x_i^n &= x_i^0 (1 - d^n) + (x_i^0)^3 d^n, \quad y_i^n = y_i^0 (1 - d^n) + (y_i^0)^2 d^n, \\ d^n &= \frac{\sin(2\pi t^n)}{2}, \quad t^n = n/n_{\max}, \end{aligned} \quad (13)$$

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.

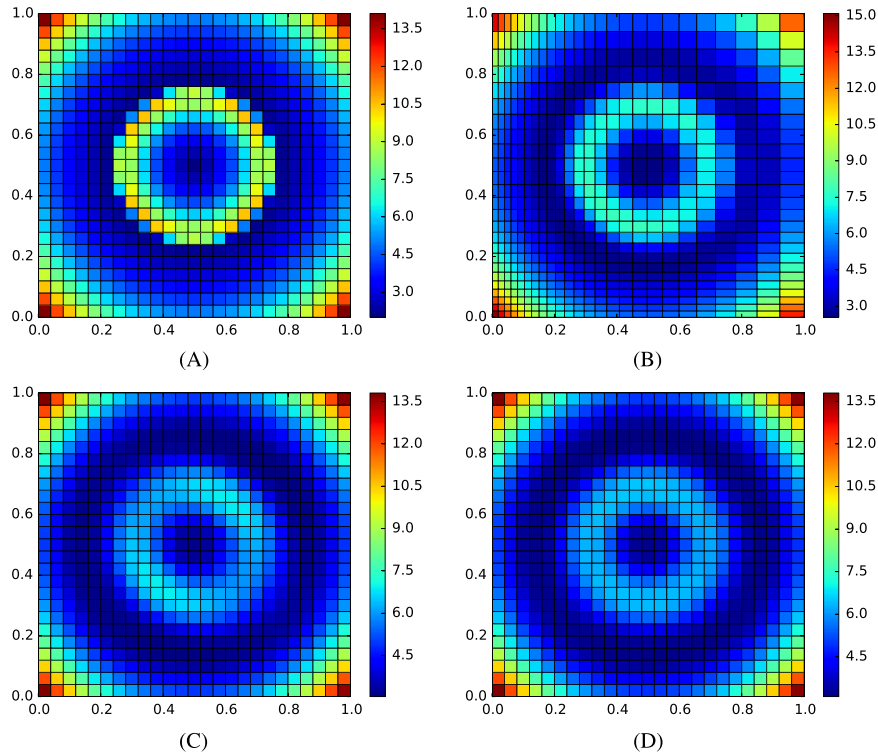


FIGURE 4 Density profile during the cyclic remapping test, 25×25 mesh, discontinuous exponential initial distribution, tensor product rezone, $n_{\max} = 50$. A, Initial state; B, $n = 12$, swept region remapping; C, final state, swept region remapping; D, final state, exact intersection remapping [Colour figure can be viewed at wileyonlinelibrary.com]

To measure the total error produced by the remapping process, we use the L1 norm:

$$L_{1m} = \frac{\sum_c m_c^n - m_c^0}{\sum_c m_c^0}, \quad (14)$$

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 ($\sim 3.3\%$) and intersection-based ($\sim 5\%$) 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:

$$\sigma_{L_{1m}} = \frac{\sqrt{\frac{1}{4} \sum_{q=1}^4 (L_{1m}^q - L_{1m})^2}}{L_{1m}}, \quad (15)$$

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 hybrid 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.

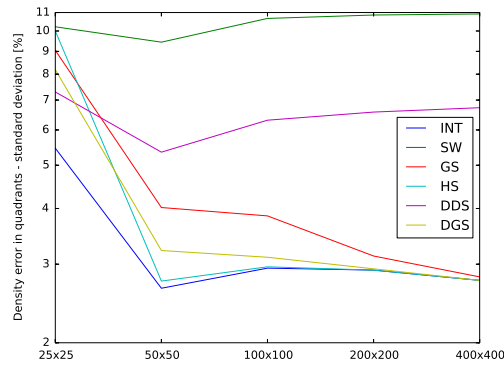


FIGURE 5 Cyclic remapping – variance of L1 mass errors in domain quadrant comparison for various mesh resolutions, discontinuous exponential initial distribution, and tensor product rezone. INT, intersections; SW, swept regions; GS, gradient switch; HS, Hessian switch; DDS, directional second derivative; DGS, diagonal second derivative [Colour figure can be viewed at wileyonlinelibrary.com]

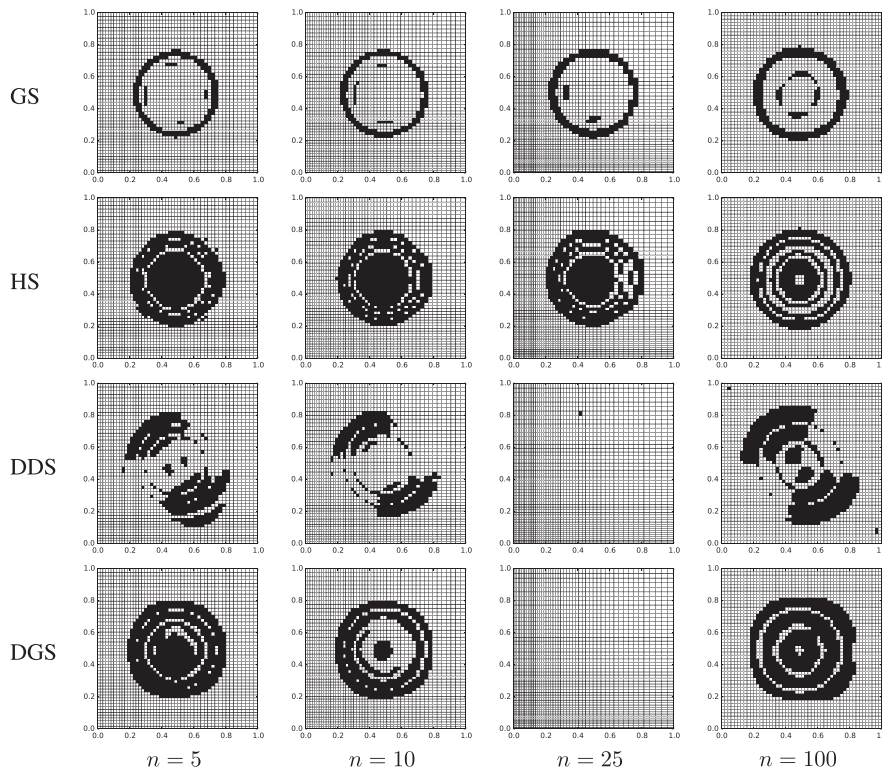


FIGURE 6 Switching functions pattern during cyclic remapping, 50×50 mesh, discontinuous exponential initial distribution, tensor product rezone, and adaptive hybrid remapping. Switch pattern: black, intersections; white, swept regions. GS, gradient switch; HS, Hessian switch; DDS, directional second 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.

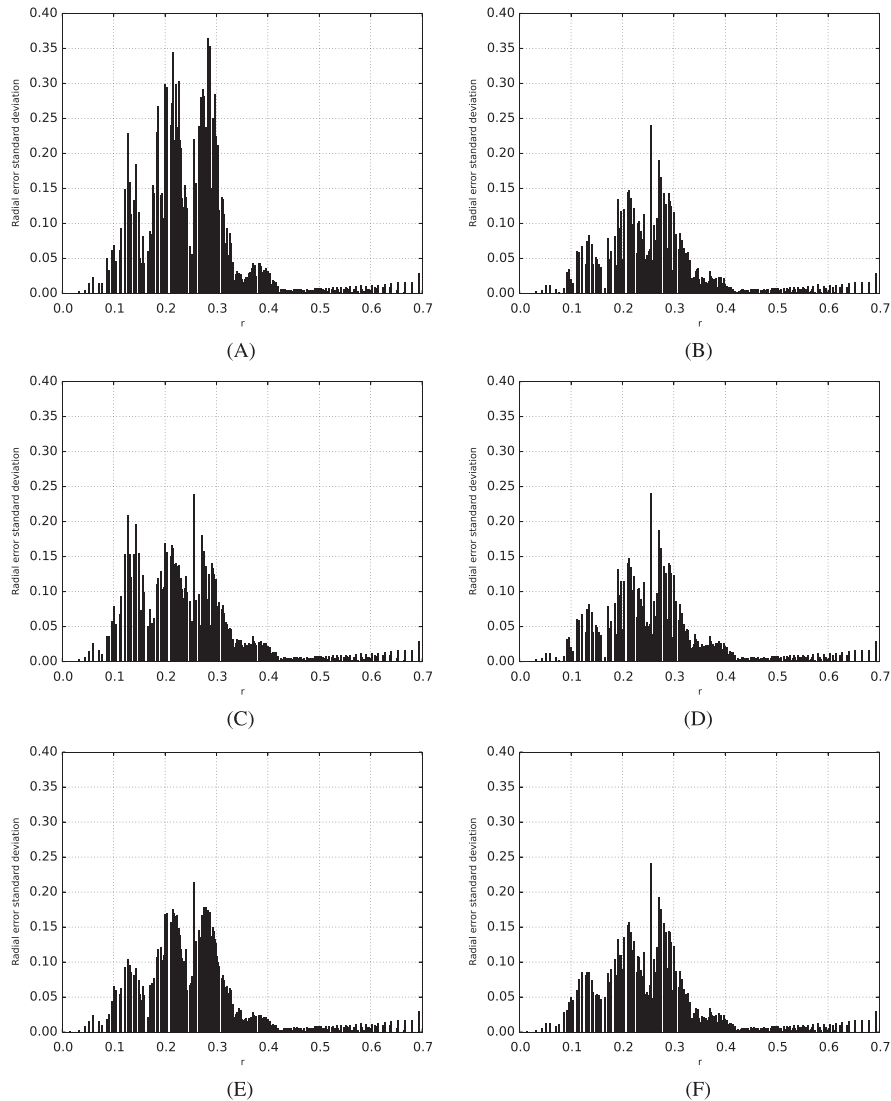


FIGURE 7 Cyclic remapping – L1 mass error radial variance comparison for different remapping methods, 50×50 mesh, discontinuous exponential initial distribution, and tensor product rezone, $n = 100$. A, Swept regions; B, intersections; C, gradient switch; D, Hessian switch; E, directional second derivative; 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 solver²⁵ with predictor–corrector time integration. The remapping stage uses the Winslow mesh smoothing algorithm²⁶ every 10 Lagrangian steps. The initial conditions of the Sedov test²² represent an energy release from the origin (here in cell $(0, 0)$), while other quantities are initially constant:

$$\rho = 1.0, p = 0.4 \cdot 10^{-14}, \mathbf{u} = \mathbf{0}, \gamma = 1.4, \epsilon_{0,0} = 409.7, t = 1.0. \quad (16)$$

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

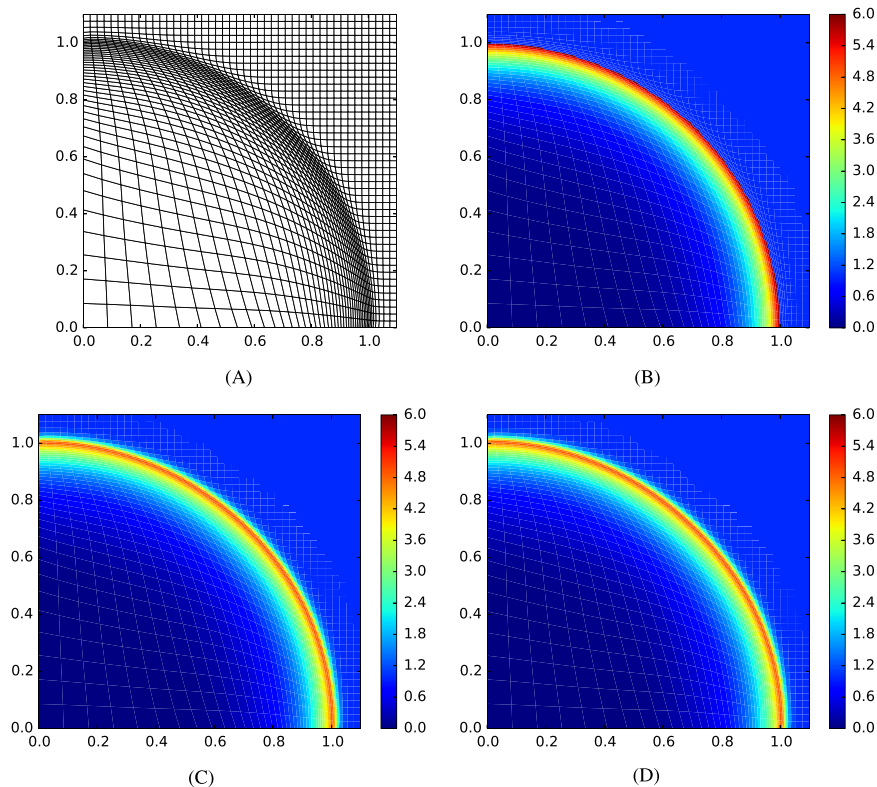


FIGURE 8 Sedov problem – density profile, ALE simulation with Winslow rezoning every 10 time steps, 45×45 cells, $t = 1.0$. A, Computational mesh; B, sampled analytical solution; C, numerical solution (intersection fluxes); D, numerical solution (swept region fluxes) [Colour figure can be viewed at wileyonlinelibrary.com]

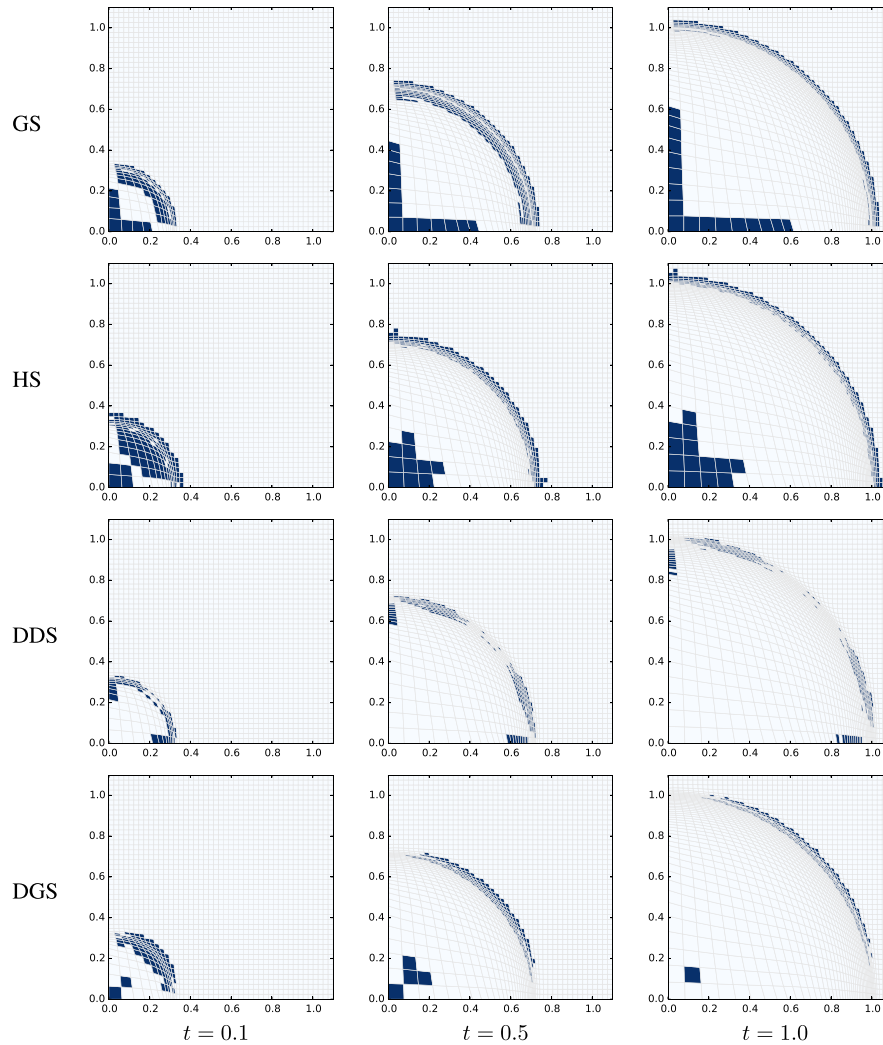


FIGURE 9 Sedov problem – switch pattern time evolution, ALE simulation with Winslow rezoning every 10 time steps, adaptive hybrid remapping, 45×45 cells, $t = 1.0$. Switch pattern: blue, intersections; white, swept regions. GS, gradient switch; HS, Hessian switch; DDS, directional second derivative; 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.

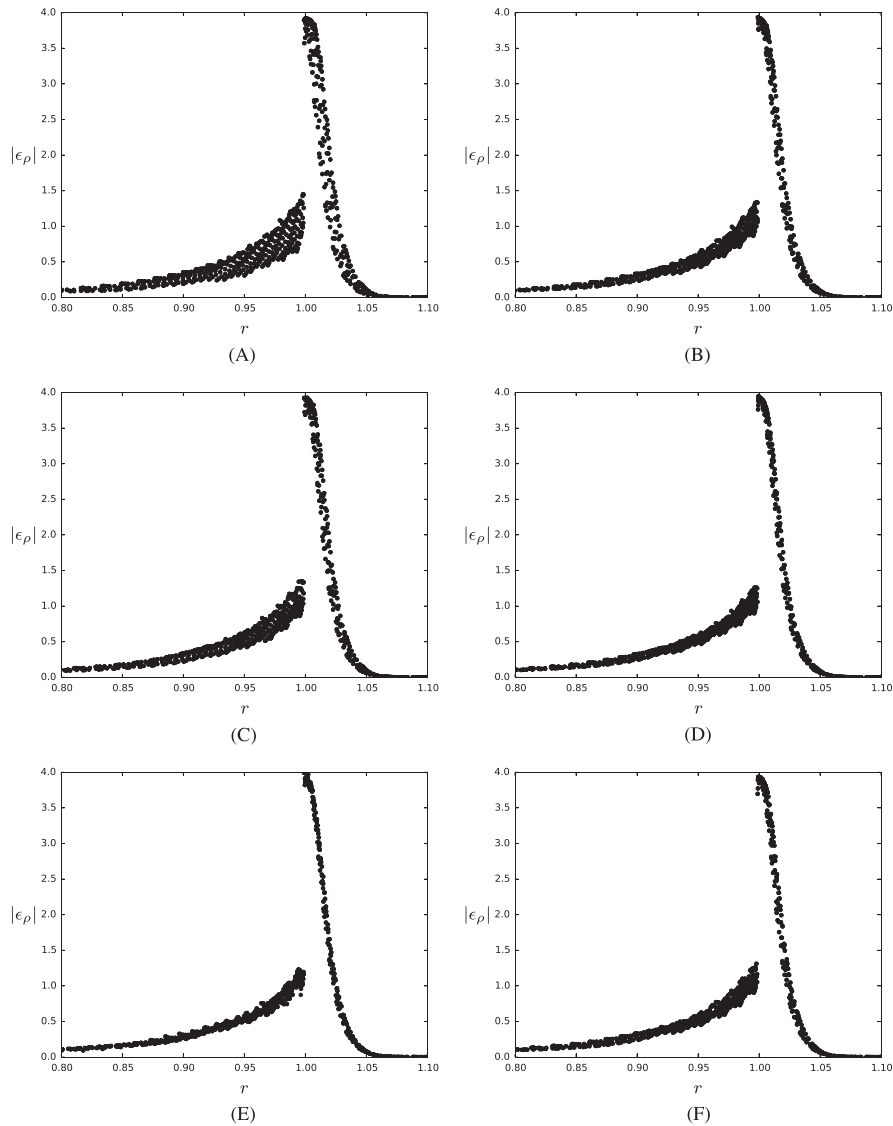


FIGURE 10 Sedov problem – radial density error distribution near the shock front compared by the remapping method, ALE simulation with Winslow rezoning every 10 time steps, 45×45 cells, $t = 1.0$. A, Swept regions; B, intersections; C, gradient switch; D, Hessian switch; E, directional second derivative; 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 comparing 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

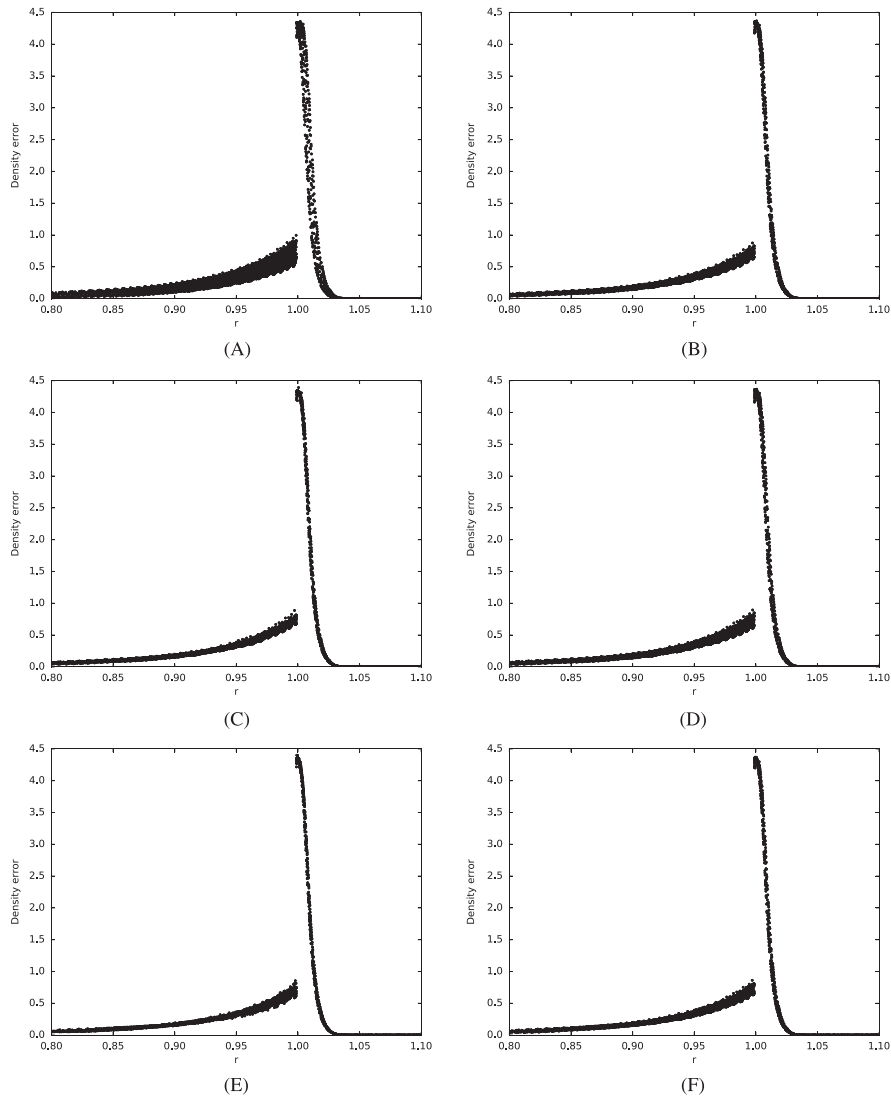


FIGURE 11 Sedov problem – radial density error distribution near the shock front compared by the remapping method, ALE simulation with Winslow rezoning every 10 time steps, 90×90 cells, $t = 1.0$. A, Swept regions; B, intersections; C, gradient switch; D, Hessian switch; E, directional second derivative; F, diagonal second derivative

(90×90 cells) than that in the original test²² 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 vortex²⁸ with compressible ideal gas – a standard test that involves rotational flow inside the domain:

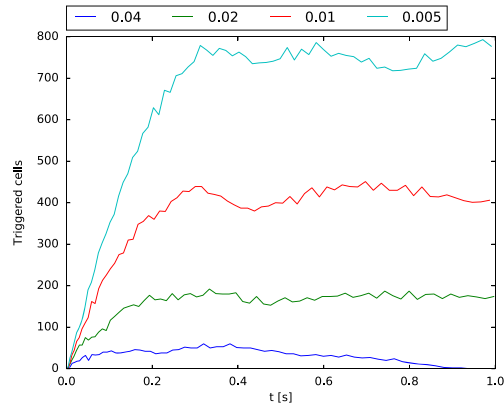


FIGURE 12 Sedov problem – time history of the total number of cells triggered for intersection remapping. Computed using the adaptive hybrid method with the diagonal second derivative switch. Comparison of different threshold settings. ALE simulation with Winslow rezoning every 10 time steps, 45×45 cells [Colour figure can be viewed at wileyonlinelibrary.com]

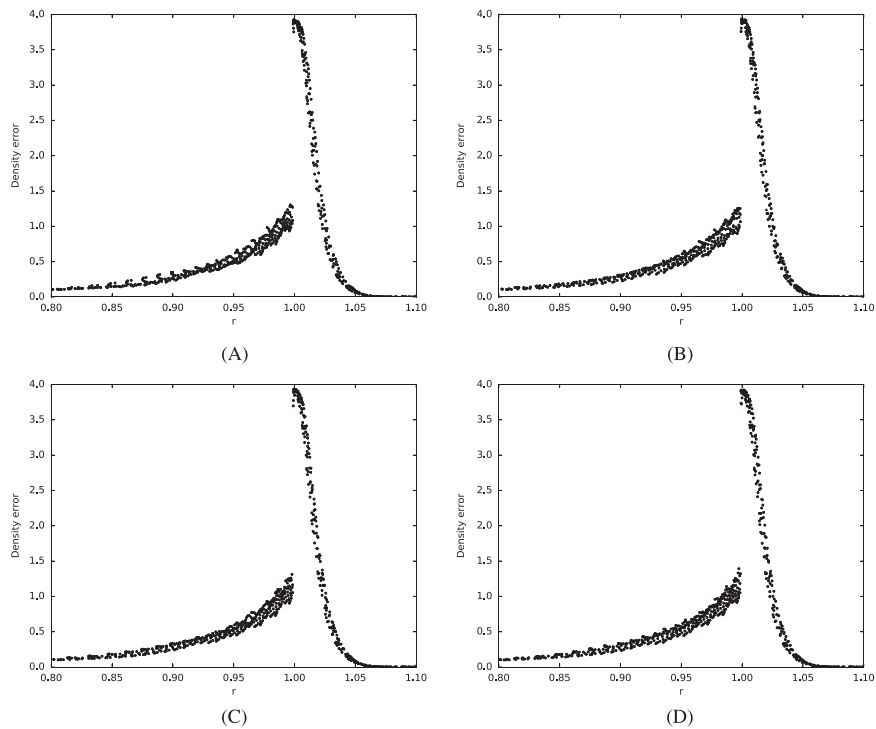


FIGURE 13 Sedov problem – radial density error distribution near the shock front using the adaptive hybrid remapping method with the DGS switch. Comparison of different threshold settings. ALE simulation with Winslow rezoning every 10 time steps, 45×45 cells, $t = 1.0$. A, $a_{DGS} = 0.04$; B, $a_{DGS} = 0.02$; 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.

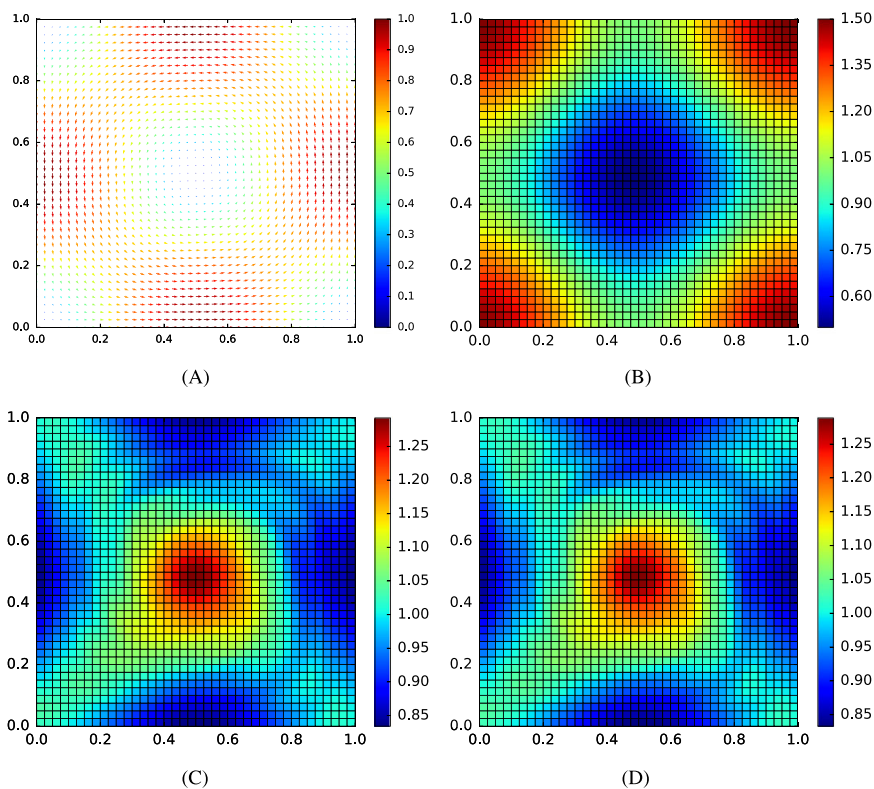


FIGURE 14 Taylor–Green vortex. ALE simulation with initial mesh rezoning on every time step, 40×40 cells. A, Velocity field at $t = 0$; B, pressure at $t = 0$; C, density at $t = 4$, intersection remapping; D, density at $t = 4$, swept-based remapping [Colour figure can be viewed at wileyonlinelibrary.com]

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_{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.

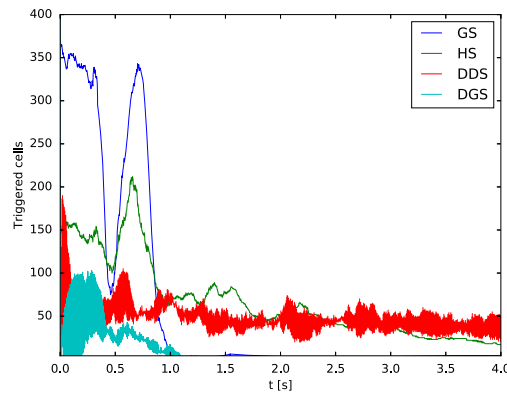


FIGURE 15 Time history of the total number of cells triggered for intersection remapping. Comparison of different switches in the adaptive hybrid remapping method. The Taylor–Green compressible vortex test, ALE simulation with rezoning to initial mesh on each step, 40×40 mesh. GS, gradient switch; HS, 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 project 14-21318S, 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. fId.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;85:363–382. <https://doi.org/10.1002/fld.4384>



Appendix E

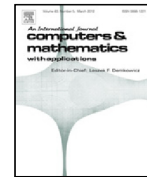
Original Research Article: Second-invariant-preserving Remap of the 2D deviatoric stress tensor in ALE methods

Authors: Matej Klima, Milan Kucharik, Jan Velechovsky, Mikhail Shashkov
Published in Computers & Mathematics with Applications. 2019; vol. 78, pp.
654-669.



Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

Second-invariant-preserving Remap of the 2D deviatoric stress tensor in ALE methods

Matěj Klíma^{a,*}, Milan Kuchařík^a, Jan Velechovský^b, Mikhail Shashkov^b

^a Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Břehová 7, Praha 1, 115 19, Czech Republic

^b X-Computational Physics Group, Los Alamos National Laboratory, Los Alamos, NM 87545, USA



ARTICLE INFO

Article history:
Available online 20 June 2018

Keywords:
ALE
Remapping
Limiter
Stress tensor
Symmetry preservation

ABSTRACT

For numerical simulations of impact problems or fluid–solid interactions, the ALE (Arbitrary Lagrangian–Eulerian) approach is a useful tool due to its ability to keep the computational mesh smooth and moving with the fluid. The elastic–plastic extension of the compressible fluid model requires tensor variables for the description of non-volumetric (deviatoric) mechanical stress. While Lagrangian numerical schemes based on the evolution equation of the stress tensor are well developed, tensor remap is still a relatively unexplored territory.

We propose a new approach to deviatoric stress remapping, where the second invariant J_2 (a conservative scalar quantity related to the strain energy) is remapped independently of the tensor components. These are re-scaled to match the remapped invariant value, effectively using only the principal directions and eigenvalue ratio from the component-wise remap. This approach is frame invariant, preserves J_2 invariant bounds and conserves the total invariant. We compare our method with component-based remapping using a simple synchronized limiter or a specialized stress tensor limiter described in the literature.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

The Arbitrary Lagrangian–Eulerian (ALE) framework [1,2] was originally used for demanding simulations with high speed shear flows in hydrodynamics. Such methods allow arbitrary movement of the computational mesh during the simulation. In the indirect ALE formulation described here, each time step is split into the following phases:

- **the Lagrangian step** [3], which advances the solution and the mesh in time,
- **the rezoning step** – if needed, the computational mesh is smoothed to maintain its geometric quality,
- **the remapping step** – if the mesh is rezoned, all quantities that are conserved in the Lagrangian step must be conservatively interpolated onto the rezoned mesh.

Currently, the use of the ALE methods is being investigated in the fields of solid mechanics and fluid–solid interactions as well. Lagrangian schemes for elastic–plastic flows have been used for decades already [4]. If this is extended to ALE, all quantities must be remapped onto the new mesh. An established remapping method for staggered meshes uses a second-order flux-form interpolation which guarantees conservation of the remapped scalar quantity [5]. This must be supplemented by a limiter, which will prevent oscillations at discontinuities by constraining the remapped quantity within

* Corresponding author.

E-mail addresses: klimamat@fjfi.cvut.cz (M. Klíma), kucharik@newton.fjfi.cvut.cz (M. Kuchařík), jan@lanl.gov (J. Velechovský), shashkov@lanl.gov (M. Shashkov).

the bounds defined by its local extrema (local monotonicity preservation). Such methods are applicable for remapping scalar fields, mainly density and internal energy, while nodal mass fluxes on the dual mesh are used to remap momentum, with a correction to satisfy conservation of total energy [6].

Unlike with fluid quantities, remapping of the stress tensor (and tensor fields in general) is still a relatively unexplored territory. Tensor components are not considered a conservative quantity, instead, preserving internal relations between components (e.g. invariants or principal directions) is key. To our best knowledge, there are currently a few attempts at tensor remap in the realm of cell-centered discretization. Burton et al. [7] proposed to remap the stress tensor with a limiter designed to preserve the bounds of generated elastic forces. Another approach is to transform the stress tensor to principal coordinates of the velocity gradient prior to limiting (originally designed for vectors by Maire [8]).

In general, the monotonicity of the reconstructed scalar fields is often ensured by slope limiters in the remapping step. These were later extended to vector quantities [9]. Extensions of scalar approaches (such as the Barth–Jespersen limiter [10] applied component-wise or in a synchronized manner) to tensors are claimed to cause symmetry distortions [7]. Another alternative scheme, inspired by the Vector Image Polygon limiter [11] applied to tensor components [12] was investigated, constraining the tensor components within a convex hull constructed in the tensor component space. This was observed to be too diffusive in practical applications and difficult to implement robustly due to the need of a robust multidimensional convex hull construction.

A new approach to limiting tensors was proposed by Lohmann [13]. This method is intended for orientation tensor transport applications, although it may work for any 3D tensors where the primary objective is to limit the eigenvalue range. In the case of stress tensors, we are interested more in constraining invariants, which is a different requirement.

A few methods have been proposed recently by Sambasivan et al. [14] specifically for stress tensor limiting. These approaches were constructed with frame invariance and preservation of bounds of tensor invariants in mind. We show that such approach can be used for stress tensor remapping as well, although only preservation of local maxima will be guaranteed. Nevertheless, this work has highlighted the important idea of second invariant boundedness, which we will further use in this paper as well.

Building on the ideas presented in [14], we are convinced that in the case of the deviatoric stress tensor in a hypo-elastic framework with the von Mises yield condition, the conservation and monotonicity of the second invariant J_2 are most important. Some of the arguments for this are:

- Preservation of J_2 maximum guarantees that the stress state does not enter a nonphysical state outside of the yield surface.
- J_2 is proportional to elastic strain energy density in the linear hypo-elastic model, therefore making it a conservative quantity.
- J_2 is calculated as a Frobenius matrix norm squared, which allows us to scale the tensor in a frame-invariant way.
- The von Mises stress (square root of J_2) is in general an important measure of stress magnitude.

In this paper, we propose a novel method of deviatoric stress tensor remapping based on these requirements. We remap the J_2 invariant independently of the tensor components, which are scaled afterwards to fit the remapped invariant value. The invariant is remapped as a scalar quantity and a readily available slope limiter guarantees its monotonicity. The total J_2 invariant (proportional to elastic energy in the linear hypo-elastic framework) is conserved and frame invariance is preserved as well.

Such method can unfortunately distort the principal directions of the stress tensor in specific cases (critical points, discontinuities in directions). A relaxation mechanism is presented to detect the discontinuities and employ component-based remap if this behavior is detected. The transition between both remapping algorithms is ensured by a smooth function of the principal direction relative angle.

The performance of our method is compared with component-wise remapping (with minimum or scaling limiters) on 1D cylindrical and 2D (tensor field rotation) static remapping-only tests. A preview of a result of a full elastic ALE test case is also presented. This paper extends the preliminary research published in [12].

2. Governing equations – the Lagrangian step

In this section we briefly introduce the underlying model that is discretized in the Lagrangian step of the ALE computation. Although the Lagrangian part itself is not the objective of this paper, it is needed to describe the intended application of the presented remapping method. We solve the time-dependent Euler equations in the Lagrangian form, extended to a general elastic–plastic continuum [4,15]:

$$\rho \frac{dv}{dt} - \nabla \cdot \mathbf{u} = 0, \quad (1)$$

$$\rho \frac{d\mathbf{u}}{dt} - \nabla \cdot \boldsymbol{\sigma} = 0, \quad (2)$$

$$\rho \frac{d\epsilon}{dt} - \boldsymbol{\sigma} : \nabla \mathbf{u} = 0, \quad (3)$$

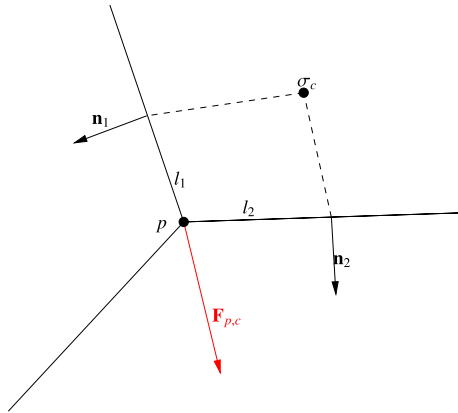


Fig. 1. Cell c to node p sub-zonal elastic force $\mathbf{F}_{p,c}$ construction with half-edge lengths l_1, l_2 , normals $\mathbf{n}_1, \mathbf{n}_2$ and the cell-centered stress tensor σ_c .

where ρ represents density, $v = \frac{1}{\rho}$ specific volume, σ the Cauchy stress tensor, \mathbf{u} velocity vector, and ϵ being the specific internal energy. The tensor double dot product is defined as $A : B = \text{tr}(A^T B)$. The Cauchy stress tensor is symmetric and can be decomposed as

$$\sigma = -pI + S, \quad (4)$$

where p is hydrodynamic pressure, I the identity matrix and S the deviatoric stress tensor. An equation of state is required for the closure of the system, such as the Mie–Grüneisen EOS [16] in the case of metallic solids.

The system is solved by a numerical scheme based on compatible discretization [3]. The movement of the computational mesh is calculated using nodal force vectors while the discrete stress tensor is defined in cell centers along with the discrete pressure. The elastic part of the cell-to-node sub-zonal forces is defined as

$$\mathbf{F}_{p,c} = l_1 \sigma_c \cdot \mathbf{n}_1 + l_2 \sigma_c \cdot \mathbf{n}_2, \quad (5)$$

and then combined to yield the total nodal elastic force,

$$\mathbf{F}_p = \sum_{c \in N(p)} \mathbf{F}_{p,c}, \quad (6)$$

where $N(p)$ is a set containing all neighboring cells of node p . l_1, l_2 is equal to the half of the respective cell edge length and $\mathbf{n}_1, \mathbf{n}_2$ are the unit normal vectors. See Fig. 1 for details. Pressure-based (hydrodynamic) forces are computed similarly using a scalar relation and include an artificial viscosity contribution.

The Euler equations must be supplemented with a constitutive stress–strain equation – here we use the linear-elastic Wilkins model [4,15] supplemented by a von Mises yield criterion for plasticity:

$$|S| = \sqrt{S : S} = \sqrt{\text{tr}(S^T S)} \leq \sqrt{\frac{2}{3}} Y_0, \quad (7)$$

where Y_0 is the yield strength, a material constant determining the stress measure at which the material starts to deform plastically.

Basic properties of the deviatoric stress tensor are described further. In two-dimensional planar geometry it has the following shape:

$$S = \begin{pmatrix} S_{xx} & S_{xy} & 0 \\ S_{xy} & S_{yy} & 0 \\ 0 & 0 & -(S_{xx} + S_{yy}) \end{pmatrix}. \quad (8)$$

It is necessary to use the full 3×3 representation [15], where the third diagonal term enforces the deviatoric property $\text{tr}(S) = 0$. The characteristic equation of the tensor defines its three invariants:

$$\lambda^3 + J_1 \lambda^2 + J_2 \lambda + J_3 = 0, \quad (9)$$

$$J_1 = \text{tr}(S) = 0, \quad J_2 = \frac{1}{2}(S : S) = \frac{1}{2} \text{tr}(S^T S), \quad J_3 = \det(S). \quad (10)$$

We are especially interested in the J_2 invariant, as its square root is present in (7) in the form of the von Mises stress. This justifies the enforcing of positivity and upper bound (plastic threshold) preservation for the second invariant to represent the yield surface correctly. Also the conservation of total invariant is important for the linear elasticity model, in which it is proportional to the elastic strain energy density:

$$e_{\text{elast.}} = \frac{1}{2\mu} J_2, \tag{11}$$

where μ is the shear modulus, a material constant. The J_2 invariant is also equivalent to the Frobenius norm of the tensor squared – this allows us to scale an arbitrary deviatoric stress tensor S with a scalar parameter α to change its second invariant in the following way:

$$J_2(\alpha S) = \alpha^2 J_2(S) \tag{12}$$

3. Component-wise remap of the deviatoric stress tensor

This section describes the intuitive way of remapping the deviatoric stress tensor by treating the individual components of the tensor as independent scalar variables. Unit-wise the stress component remap is equivalent to pressure remap (used in some multimaterial computations [6]), which aims to conserve the pressure–volume work. The stress tensor in the cell c is remapped as follows (the tilde symbol is used to denote the cells and quantities related to the rezoned mesh):

$$\tilde{S}^c \tilde{V}^c = S^c V^c + \sum_{c' \in \mathcal{N}(c)} (F_{c' \cap \tilde{c}}^S - F_{c \cap \tilde{c}'}^S), \quad F_{c' \cap \tilde{c}}^S = \iint_{c' \cap \tilde{c}} S(\mathbf{x}) dV, \tag{13}$$

where $\mathcal{N}(c)$ is the set of all neighboring computational cells, $F_{c' \cap \tilde{c}}^S$ is the tensor flux from neighboring cell c' to cell \tilde{c} . This is calculated by integrating an approximate reconstruction of the remapped quantity over the exact intersection of the old cell c' and rezoned cell \tilde{c} [17,5]. The second-order tensor reconstruction $S(\mathbf{x})$ can be expressed in terms of the independent tensor components as:

$$S(\mathbf{x}) = \begin{pmatrix} S_{xx} \\ S_{xy} \\ S_{yy} \end{pmatrix} (\mathbf{x}) = \begin{pmatrix} S_{xx}^c \\ S_{xy}^c \\ S_{yy}^c \end{pmatrix} + (\mathbf{x} - \mathbf{x}_c) \begin{pmatrix} \psi \nabla S_{xx} \\ \psi \nabla S_{xy} \\ \psi \nabla S_{yy} \end{pmatrix}, \tag{14}$$

where ∇S is the tensor gradient and its components can be obtained using the least squares optimization [18,19] on all neighboring cells. \mathbf{x}_c is the geometric centroid of the computational cell and ψ is a scalar limiting coefficient. In particular, the Barth–Jespersen procedure [18,10] is used here:

$$\psi_{xx}^p = \begin{cases} \min \left(\frac{S_{xx}^{\text{max}} - S_{xx}^c}{S_{xx}^p - S_{xx}^c}, 1.0 \right) & \text{if } S_{xx}^p > S_{xx}^c \\ \min \left(\frac{S_{xx}^{\text{min}} - S_{xx}^c}{S_{xx}^p - S_{xx}^c}, 1.0 \right) & \text{if } S_{xx}^p < S_{xx}^c \\ 1.0 & \text{otherwise,} \end{cases} \tag{15}$$

$$\psi_{xx} = \min_{p \in \mathcal{P}(c)} (\psi_{xx}^p), \quad S_{xx}^p = S_{xx}^c + (\mathbf{x}_p - \mathbf{x}_c) \nabla S_{xx}^c, \tag{16}$$

where $\mathcal{P}(c)$ is the set of all vertices of the cell c , \mathbf{x}_p is the position of the vertex (point) p and S_{xx}^p is the unlimited reconstruction in the corresponding point. S_{xx}^{max} and S_{xx}^{min} are tensor component maximum and minimum calculated on the same 9-cell stencil as is used for the gradient computation. The same procedure is also used for the other independent tensor components S_{xy} and S_{yy} .

To reduce the symmetry distortions caused by different limiters for each component, a single final limiter is often calculated in a synchronized manner, i.e. a common scalar limiting value is used (as mentioned in [7]):

$$\psi = \min (\psi_{xx}, \psi_{xy}, \psi_{yy}). \tag{17}$$

In the rest of the paper, this method is referred to as the *minimum limiter*.

3.1. J_2 invariant scaling limiter

Using a minimum of component limiters is an inherently frame-dependent approach which will not preserve symmetry when the coordinate system is rotated. An alternative to this is using a limiter formulated specifically for the deviatoric stress tensor [14]. The *scaling limiter* is based on an assumption that the monotonicity of the J_2 invariant (proportional to elastic

energy) is more important than monotonicity of the particular tensor components. The monotonicity condition leading to a Barth–Jespersen-like limiter coefficient for the J_2 invariant can be described as:

$$J_2^{\min} - J_2^c \leq J_2^p - J_2^c \leq J_2^{\max} - J_2^c \quad \forall p \in \mathcal{P}(c), \quad (18)$$

where J_2^{\min} and J_2^{\max} are determined on the set of neighboring cells. A single scaling factor (see property (12)) is then used for the reconstructed tensor:

$$\psi(\mathbf{x}) = \sqrt{\psi_{J_2} + (1 - \psi_{J_2}) \frac{J_2^c}{J_2(S(\mathbf{x}))}}, \quad \psi_{J_2} = \text{Barth-Jespersen}[J_2(S)], \quad (19)$$

If the denominator is vanishing we set the scaling factor $\psi(\mathbf{x}) = 1$ as the near-zero J_2 value implies that the tensor itself is close to zero and maximum violation is not possible. Also scaling a near-zero tensor by any means can introduce a disproportionate numerical error. The resulting limited tensor value is defined as follows:

$$S^{\text{lim}}(\mathbf{x}) = \psi(\mathbf{x}) (S^c + (\mathbf{x} - \mathbf{x}_c) \nabla S), \quad (20)$$

$$F_{c' \cap \tilde{c}}^S = \iint_{c' \cap \tilde{c}} S(\mathbf{x}) dV \sim S^{\text{lim}}(\mathbf{x}_{c' \cap \tilde{c}}) V^{c' \cap \tilde{c}}. \quad (21)$$

This approach is relatively fast, simple to implement and guarantees the preservation of local J_2 maximum by design. We are aware that the preservation of lower bound is not guaranteed by (18) due to the quadratic nature of J_2 . However, for component-wise remapping it is not guaranteed regardless of the limiter used (as shown in the next section). We still believe that such condition may be useful in situations such as near-linear tensor distributions with aligned principal directions. For a detailed description of this limiter scheme, see [14].

3.2. Shortcomings of the component-based remapping

The limiters described in this section are all utilized with the component-based remapping of the stress tensor. The scaling limiter guarantees that the local maxima of J_2 are not violated, but in certain situations it fails to preserve local minima.

Assume a remapping configuration with two tensors in neighboring cells c, c' that have the same value, except for opposite signs $S^c = -S^{c'}$. The common edge between the two cells moves towards c' so that there is a single non-self-overlapping intersection with volume $V^{c' \cap \tilde{c}} < \min(V^c, V^{c'})$. We can write the first order remap for such situation – Eq. (13) with piece-wise constant reconstruction:

$$\tilde{S}^c \tilde{V}^c = S^c V^c + S^{c'} V^{c' \cap \tilde{c}} = S^c (V^c - V^{c' \cap \tilde{c}}) \quad (22)$$

$$\tilde{S}^c = S^c \left(\frac{V^c - V^{c' \cap \tilde{c}}}{\tilde{V}^c} \right) = S^c \left(\frac{V^c - V^{c' \cap \tilde{c}}}{V^c + V^{c' \cap \tilde{c}}} \right) \quad (23)$$

Tensors $S^c, S^{c'}$ have the same J_2 values, only the orientation differs. The J_2 of the remapped tensor has the following value:

$$J_2(\tilde{S}^c) = J_2 \left(S^c \frac{V^c - V^{c' \cap \tilde{c}}}{V^c + V^{c' \cap \tilde{c}}} \right) < J_2(S^c) = J_2(S^{c'}) \quad (24)$$

This indicates that the component-based approach itself is not able to guarantee J_2 invariant monotonicity, regardless of the limiting method used. Also the density-like remapping approximation is not physically justified – stress components are not a conservative quantity.

4. J_2 invariant remapping

In the linear-elastic model, the J_2 invariant is proportional to the elastic strain energy density (11) and as such is justifiable as a conservative quantity. Therefore, remapping of the J_2 invariant satisfies conservation and allows to use a limiter to enforce J_2 bounds preservation directly.

Following [12], the remap of the J_2 invariant can be performed independently of S and expressed in a flux form:

$$\tilde{J}_2^c \tilde{V}^c = J_2^c V^c + \sum_{c' \in \mathcal{N}(c)} F_{c' \cap \tilde{c}}^{J_2} - F_{c \cap \tilde{c}'}^{J_2}, \quad (25)$$

with second order fluxes and a scalar Barth–Jespersen limiter used in the J_2 reconstruction. Then, S^c is remapped component-wise with a first order algorithm (no limiting is required) and the resulting tensor is scaled (12) by multiplying by the ratio of the remapped invariant \tilde{J}_2^c and $J_2(\tilde{S}^c)$ calculated from remapped \tilde{S}^c :

$$\tilde{\tilde{S}}^c = \tilde{S}^c \sqrt{\alpha + (1 - \alpha) \frac{\tilde{J}_2^c}{J_2(\tilde{S}^c)}}. \quad (26)$$

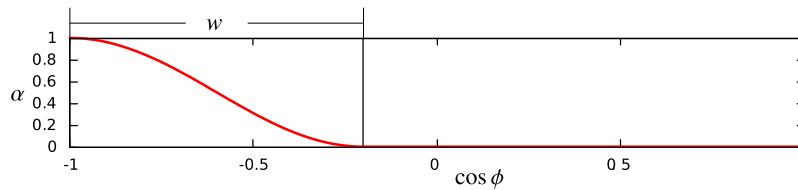


Fig. 2. Dependence of the relaxation parameter α on the cosine of the tensor relative direction $\cos \phi$ and width parameter $w = 0.8$.

A non-zero α value is used in situations where the strict re-scaling of the remapped tensor causes distortions of the resulting principal directions. The parameter is then defined as:

$$\alpha = \frac{\cos\left(\pi \frac{1+\cos\phi}{w}\right) + 1}{2} \quad \text{for } \cos\phi < 1 - w, \quad \alpha = 0 \text{ otherwise} \tag{27}$$

$$\cos\phi = \min_{c' \in \mathcal{N}(c)} \frac{S^c : S^{c'}}{\|S^c\| \|S^{c'}\|}, \tag{28}$$

where $\cos\phi$ is the directional parameter corresponding to a cosine of the maximum angle difference between two neighboring tensor principal coordinate systems (calculated as a normalized double dot product). w is the width parameter – a threshold which defines when will the relaxation be activated (see Fig. 2). This serves as a smooth transition from the component-wise to the fully re-scaled remap of the deviatoric stress. In practical applications, we typically use $w = 1$.

The usage of the low-order (donor) remapping in (26) is justified by the fact that the component-wise remap of S primarily determines the principal directions of the tensor and not its magnitude. Also, according to our observation, low-order remapping is sufficient here with negligible influence on the overall accuracy of the stress tensor profile.

5. Numerical results

In this section we show three test cases to compare J_2 remapping with previously described approaches. The first cyclic remapping test involves sequential remapping steps at with artificial grid movement only. The tensor distribution, mesh geometry and movement are all radially symmetric. If the investigated method is frame-invariant, this type of test will behave as a 1D simulation. The second problem is a more general tensor equivalent of the solid body rotation problem, in which there is a simplistic Lagrangian-like phase in each step and the tensor field is then remapped on the initial mesh. The third test demonstrates a possible application of J_2 remapping in the context of an ALE (Lagrangian+Remap) simulation of elastic-plastic flow on a simplified cylindrical shock testing problem.

5.1. Cyclic remapping of a radially discontinuous distribution of the deviatoric stress tensor

This test is aimed mainly at comparing the performance of the tensor remapping methods on a radially symmetric case with a discontinuity. This discontinuity involves a reversal in the orientation of the deviatoric stress tensor, which is initialized on an annular mesh (see Fig. 3) with the following distribution:

$$S_c = \begin{cases} \frac{1}{2} \begin{pmatrix} \cos(2\varphi_c) & \sin(2\varphi_c) \\ \sin(2\varphi_c) & -\cos(2\varphi_c) \end{pmatrix} & \text{for } r_c < 0.55 \\ -\begin{pmatrix} \cos(2\varphi_c) & \sin(2\varphi_c) \\ \sin(2\varphi_c) & -\cos(2\varphi_c) \end{pmatrix} & \text{for } r_c > 0.55, \end{cases} \tag{29}$$

where r_c, φ_c are the polar coordinates of the center of cell c . The internal and external radii of the mesh are $r_l = 0.1, r_r = 1.0$.

This distribution of the deviatoric stress is initialized and repeatedly remapped without any influence of the hydrodynamics. The testing movement of the mesh follows the radial symmetry along with the quantity distribution and mesh. This means any observed symmetry violation in the result is caused by the remapping algorithm itself. The exact definition of the mesh rezoning is a variation of the “tensor-product” cyclic remapping test [5] applied only in the radial direction:

$$r_n = r_l + \left[\frac{r_0 - r_l}{r_r - r_l} (1 - d_n) + \left(\frac{r_0 - r_l}{r_r - r_l} \right)^3 d_n \right] (r_r - r_l), \tag{30}$$

$$\varphi_n = \varphi_0, \quad d_n = \frac{1}{2} \sin\left(\frac{\pi n}{n_{\max}}\right),$$

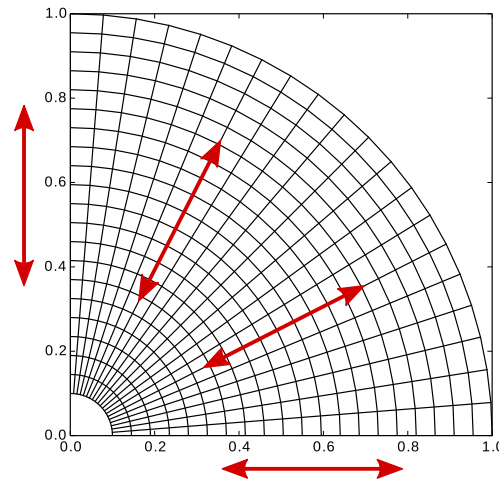


Fig. 3. Initial 20×20 computational mesh with indicated directions of the rezoning movement.

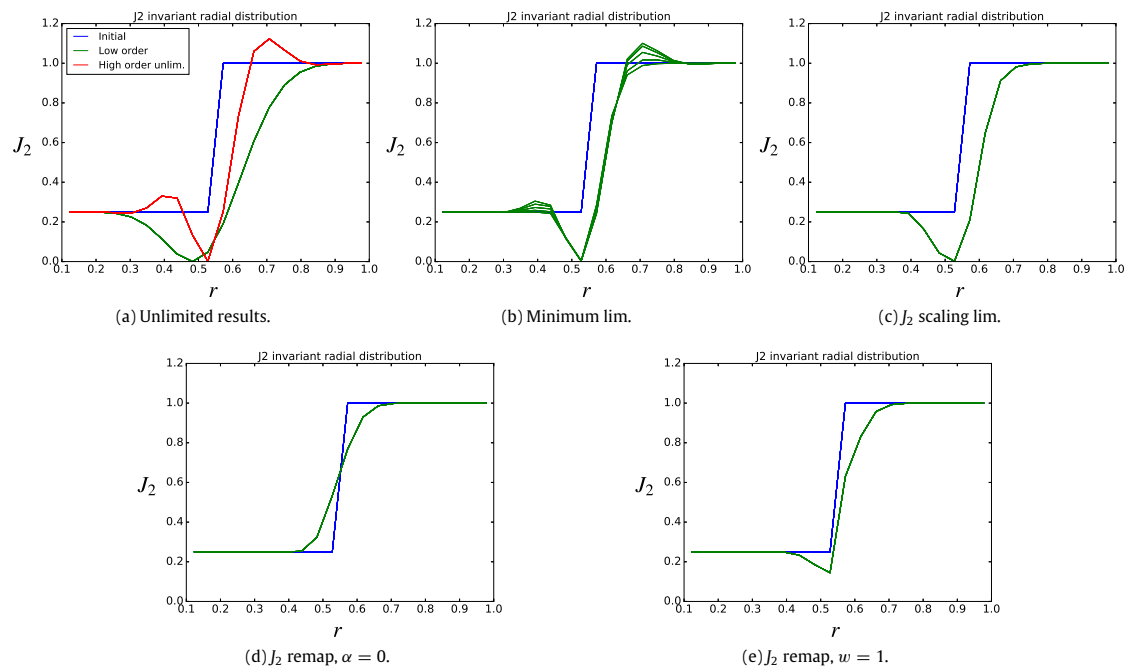


Fig. 4. Radial distribution of the J_2 invariant after $n_{\max} = 80$ cyclic remapping steps, 20×20 mesh.

where r_0, φ_0 are the initial nodal polar coordinates, n is the current remapping step and n_{\max} is the total number of remapping steps. This represents a cyclic movement of nodes in the radial direction where the initial ($n = 0$) and final ($n = n_{\max}$) meshes are identical, see Fig. 3.

The final J_2 distribution in the r direction for various tensor remapping methods is shown in Fig. 4. We can see that high-order component-wise remapping violates both local minimum and maximum. The low-order remapping violates only the local minimum, suggesting that this is an intrinsic property of the component-wise remapping method (see Fig. 4a, explanation of minimum violation in Section 3.2). Both methods show no symmetry violation.

Table 1

Various properties of the resulting J_2 invariant distribution after full 2π rotation in $n_{\max} = 640$ steps, compared by the remapping method used. Violations of local bounds are marked in red color.

	Init.	Low order	High order	Min. lim.	Scaling lim.	J_2 remap, $\alpha = 0$	J_2 remap, $w = 1$
Total J_2 relative discrepancy		0.832	0.209	0.385	0.264	$3.8 \cdot 10^{-10}$	0.044
s^4 : Relative L_1 error of J_2		0.998	0.218	0.448	0.260	0.259	0.242
s^4 : Relative Frob. norm error of S		0.966	0.166	0.295	0.175	0.480	0.269
s^2 : J_2 maximum, $r < 0.175$	1.0	0.035	1.172	0.709	0.874	0.999	0.979
s^1 : J_2 minimum, $r < 0.08$	0.5	0.143	0.413	0.535	0.534	0.540	0.540
simulation time [s]		18	140	192	144	149	164

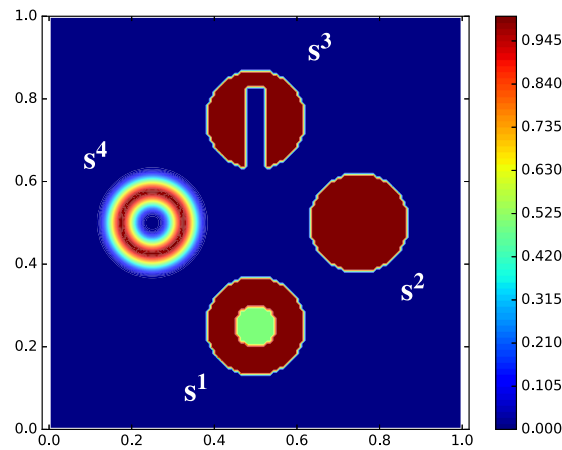


Fig. 5. J_2 invariant distribution, initial state, 128×128 mesh.

For comparison, in Fig. 4b we show results using the component-wise remapping method with a common scalar limiter that is calculated as a minimum of Barth–Jespersen limiters for each component of the deviatoric stress tensor. This shows that such method is inherently flawed, as it fails to prevent violations of extrema in the J_2 invariant and introduces violation of symmetry in the result as well.

Fig. 4c shows results using component-wise remapping with the J_2 scaling limiter [14]. It preserves radial symmetry and eliminates all violations of local maxima. Unfortunately, it is unable to preserve the local minimum.

Our J_2 remapping approach (Fig. 4d) preserves all extrema, does not violate symmetry and also preserves the total integral of the invariant (conserves the total elastic energy). Slight minimum violation can, however, be introduced if the relaxation approach is used (Fig. 4e).

5.2. Tensor advection on rotating meshes

This test is a tensor adaptation of the solid body rotation test. Unlike in the previous section, the tensor distribution, mesh and the rezoning movement are not aligned here. A tensor field with a complex shape is initialized, then in each step, all internal nodes of the mesh are rotated by an angle $2\pi/n_{\max}$ with the rotation center in (0.5, 0.5). The deviatoric stress is then remapped onto the initial mesh. This effectively represents the transport of the tensor in a rotating velocity field.

A similar test was originally designed in [13] with eigenvalues in mind. In our case we focus more on the initial J_2 invariant field. The four “solid bodies” with different stress tensor distributions are initialized as follows:

- s^1 – lower/upper bound test with two different principal directions, piece-wise constant J_2 ,

$$s_{c=(0.5,0.25)}^1 = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & a \end{pmatrix} \quad a = \begin{cases} 1 & \text{for } 0.05 < r < 0.12, \\ 0 & \text{for } r < 0.05, \end{cases}$$

- s^2 – upper bound test with radially symmetric principal directions, constant J_2 ,

$$s_{c=(0.75,0.5)}^2 = - \begin{pmatrix} \cos(2\phi) & \sin(2\phi) \\ \sin(2\phi) & -\cos(2\phi) \end{pmatrix} \quad \text{for } r < 0.12,$$

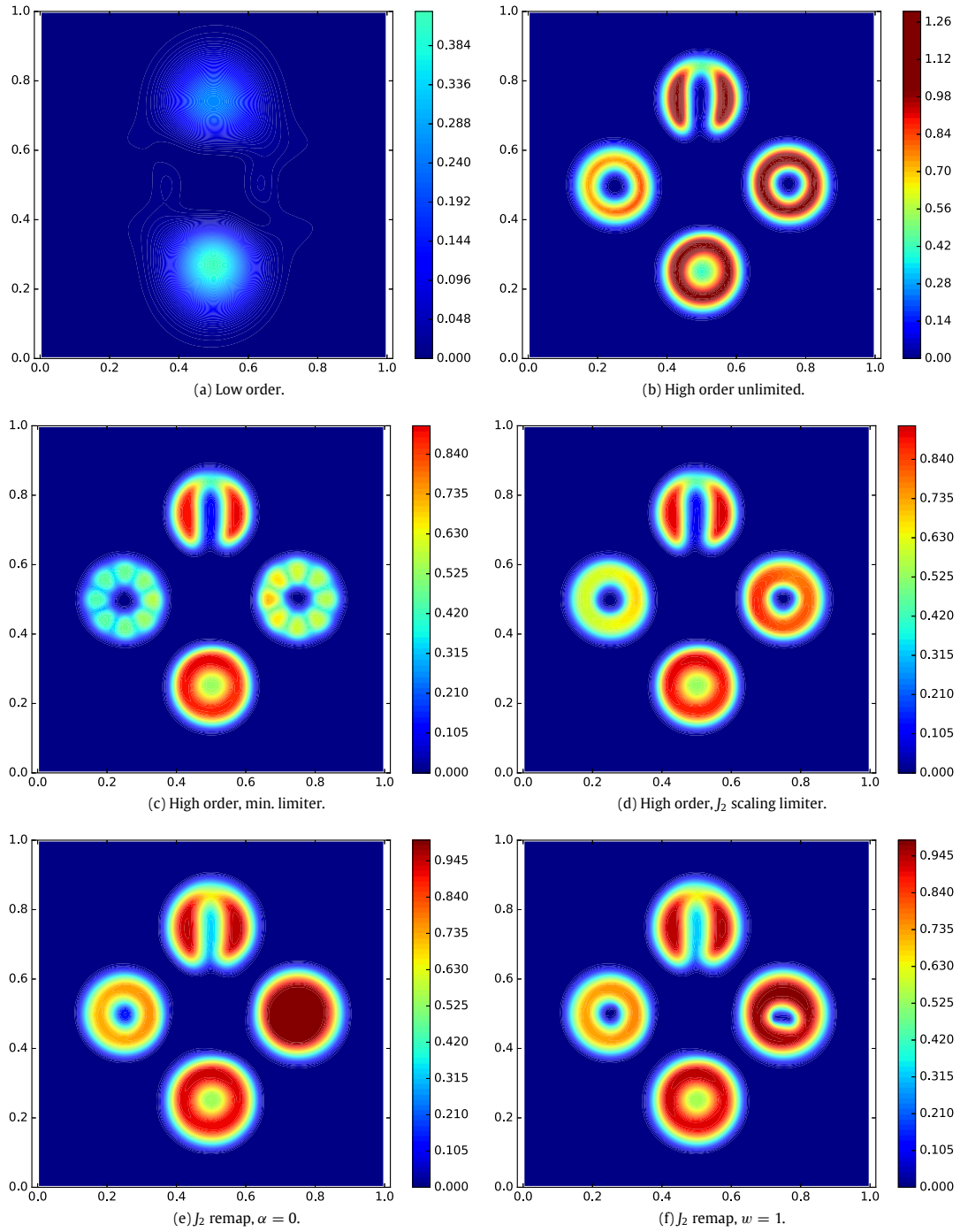


Fig. 6. J_2 invariant distribution after full 2π rotation in $n_{\max} = 640$ steps.

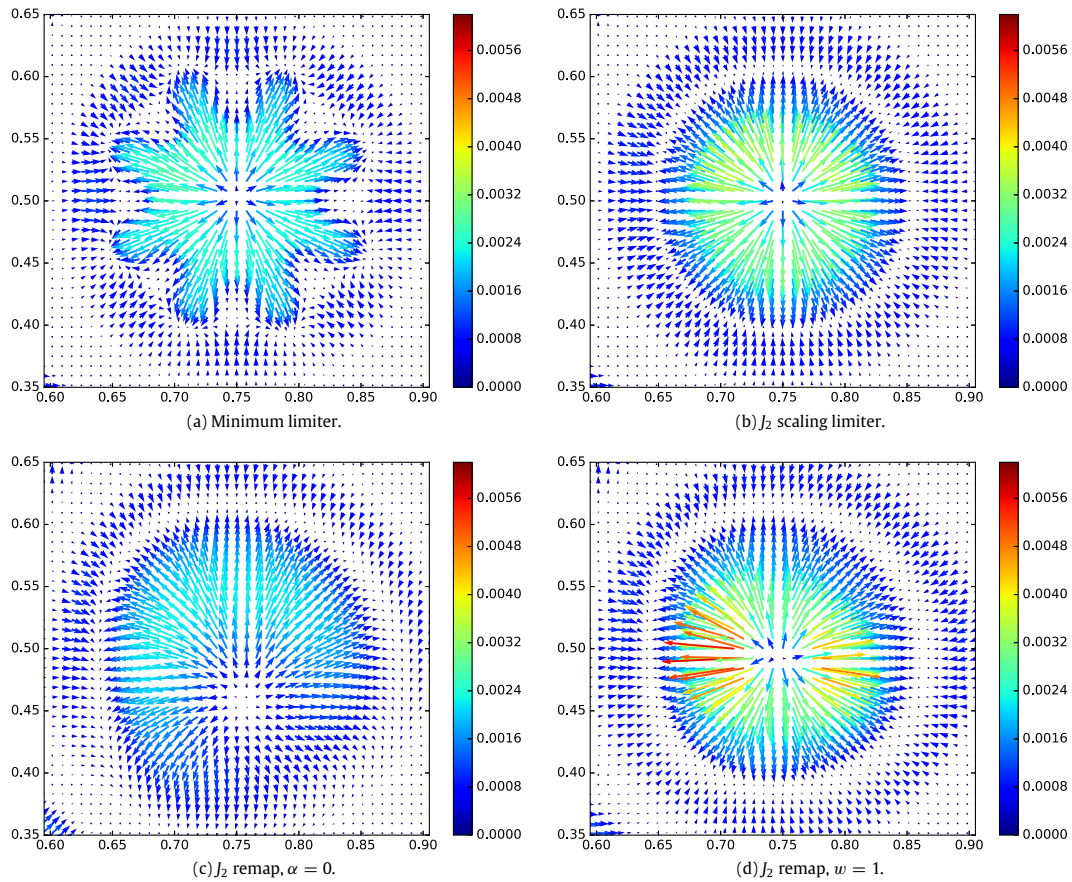


Fig. 7. Nodal elastic forces generated by the deviatoric stress field after full 2π rotation in $n_{\max} = 640$ steps. Zoom at the s_2 object.

- s^3 – slotted cylinder shape, unidirectional, constant J_2 ,

$$s_{c=(0.5,0.75)}^3 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \text{ for } r < 0.12 \cap (|x| > 0.02 \cup y > 0.08),$$

- s^4 – smooth ring distribution of J_2 with radially symmetric principal directions,

$$s_{c=(0.25,0.5)}^4 = \left(-1 - \sin\left(\frac{2\pi r}{0.15} - \frac{\pi}{2}\right) \right) \begin{pmatrix} \cos(2\phi) & \sin(2\phi) \\ \sin(2\phi) & -\cos(2\phi) \end{pmatrix} \text{ for } r < 0.15,$$

where r is the radius from the center of each object, ϕ is the corresponding angle. A zero stress tensor is initialized everywhere in the rest of the domain. The initial setup is shown in Fig. 5.

Fig. 6 shows the results after one complete rotation for various tensor remapping methods. Clearly for this kind of test, the low-order remapping has an insufficient accuracy, as the shapes are no longer visible. High-order component-wise remapping violates J_2 extrema by a significant margin. There is an undershoot in the minimum in s_1 too, for exact values see Table 1.

In the constant-direction tests the minimum and scaling limiters perform similarly. However, if radial principal directions are introduced, the minimum limiter fails to preserve the symmetry of the problem significantly, while the scaling limiter treats these distributions correctly.

One very prominent feature of all results is the large dip created in the center of s_2 in the point (0.75,0.5). This behavior is seen in all results employing component-wise remapping, even if no limiter is used. This is due to the radial symmetry of the principal directions – even though the J_2 distribution is initially constant, the tensor still has a critical point in the center of s_2 . This has similar effects as the direction reversal presented in the previous section. The J_2 remapping method

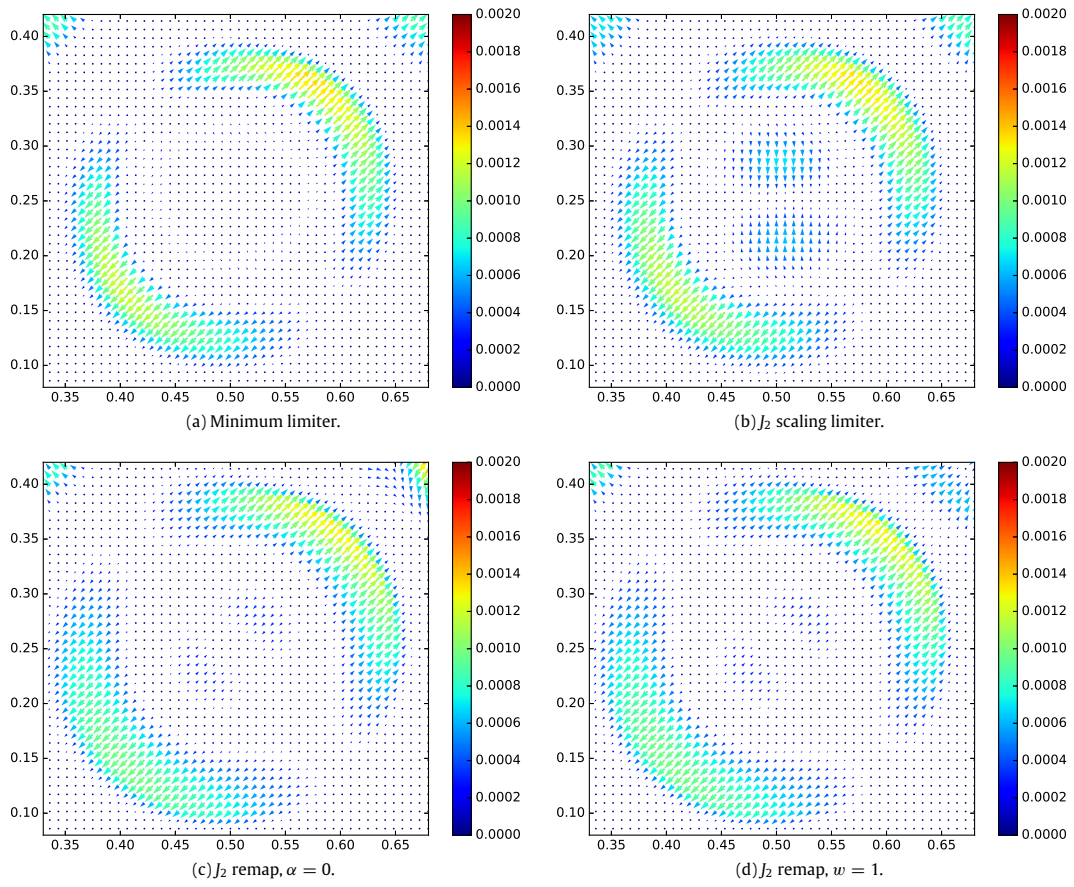


Fig. 8. Nodal elastic forces generated by the deviatoric stress field after full 2π rotation in $n_{\max} = 640$ steps. Zoom at the s_1 object.

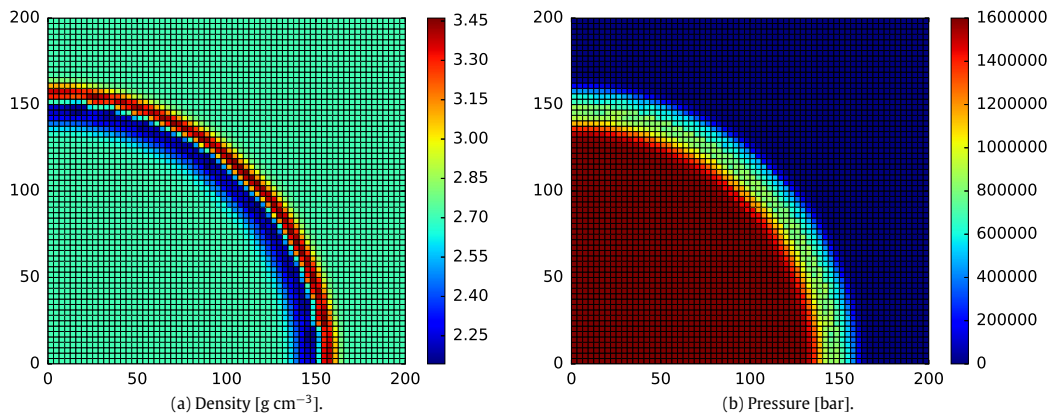


Fig. 9. Cylindrical shock in aluminum at $10 \mu\text{s}$, J_2 remapping, 64×64 mesh.

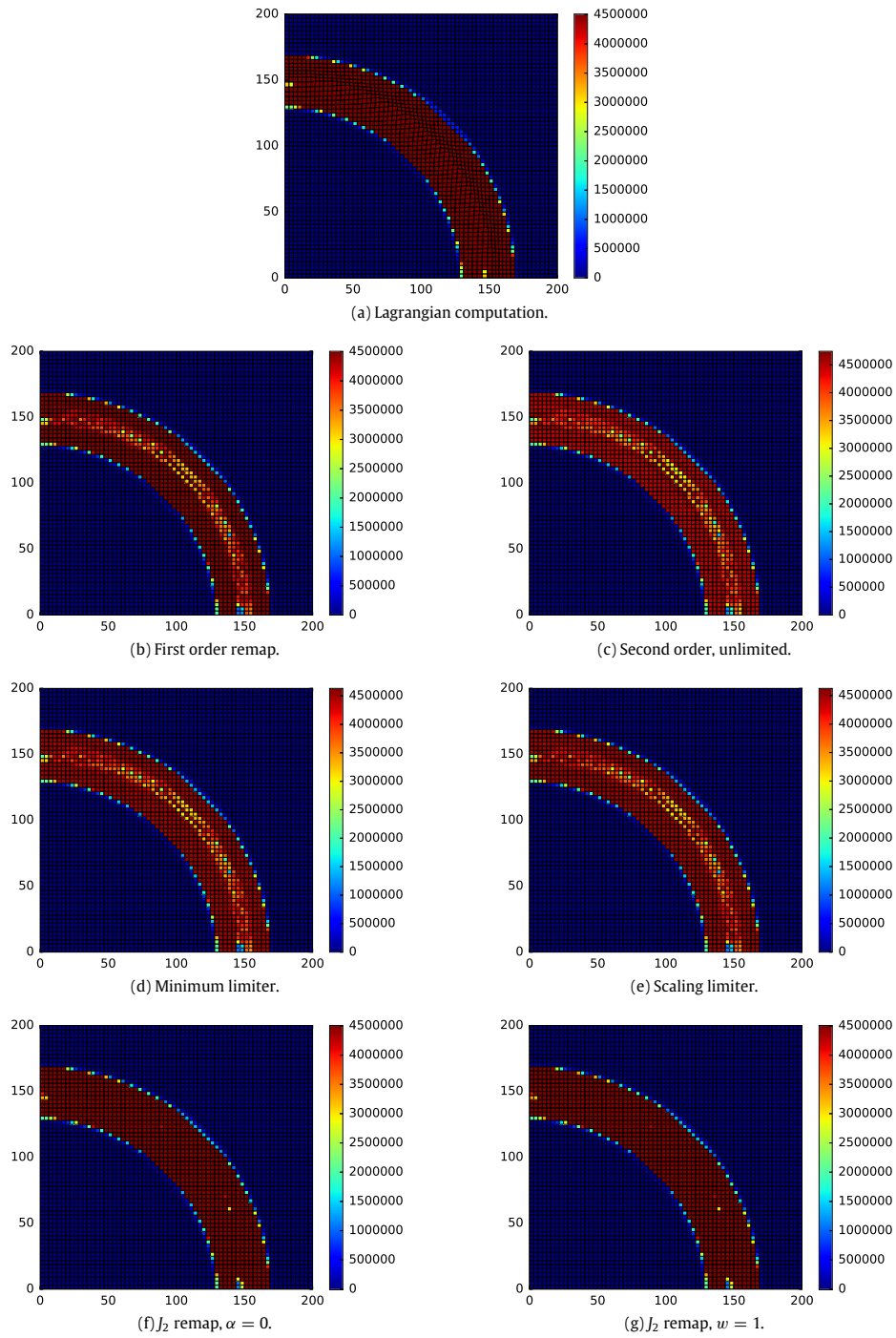


Fig. 10. J_2 invariant [bar^2] distribution, cylindrical shock in aluminum at $10 \mu\text{s}$, 64×64 mesh.

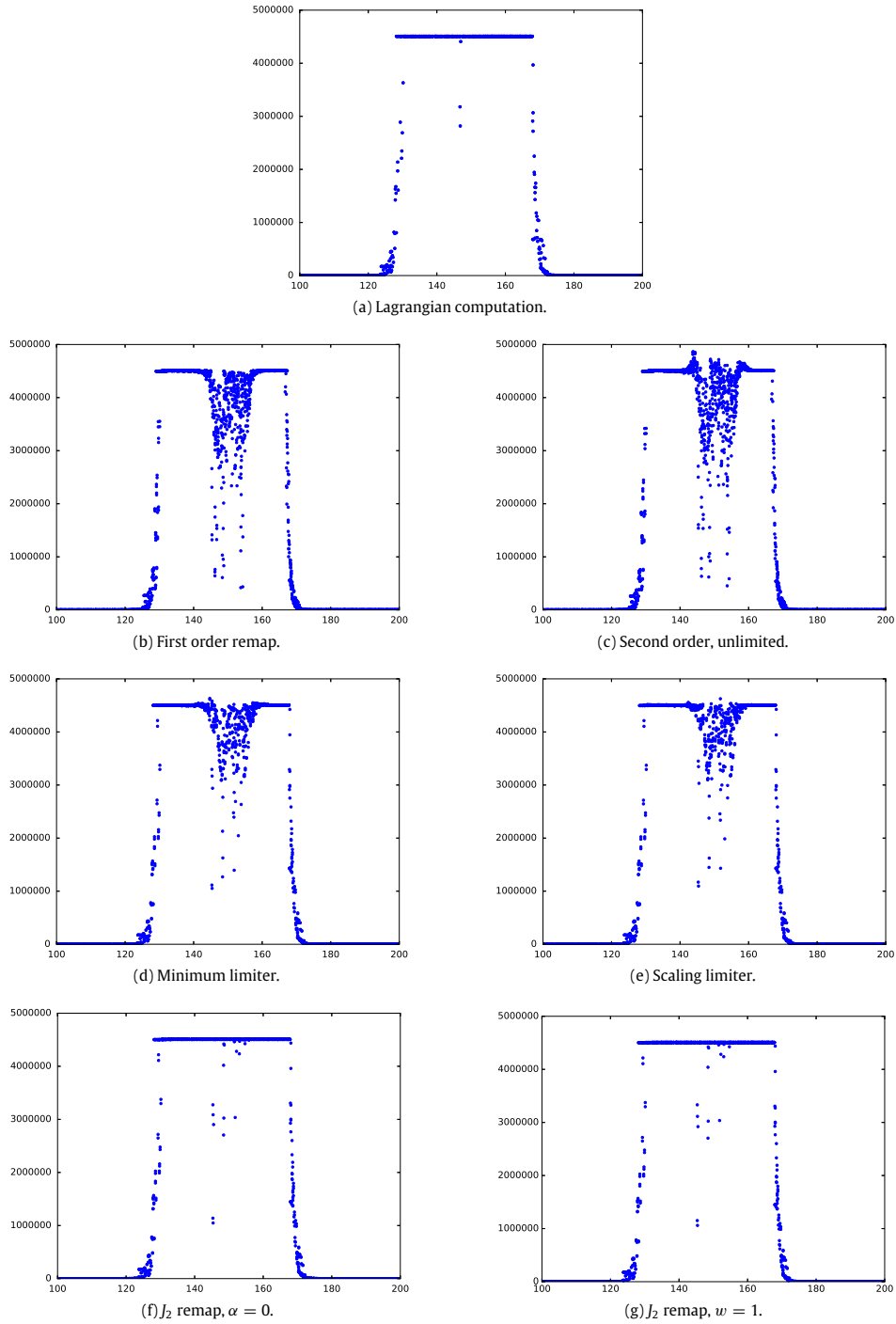


Fig. 11. Radial distribution of the J_2 invariant, cylindrical shock in aluminum at $10 \mu s$, 64×64 mesh.

without relaxation is able to represent the J_2 distribution correctly. A trade-off of this is the distortion of the principal directions distribution which cannot be seen in the J_2 plot, but can affect the elastic forces produced by the stress field gradient significantly (see Fig. 7). Using J_2 remap without relaxation leads to a deformed elastic force profile (the tensor critical point is shifted significantly). Using the relaxation with $w = 1$ is able to preserve the location of the critical point, but the resulting distribution still shows some asymmetry with forces perpendicular to the direction of rotation increased in magnitude.

If J_2 remapping is used with relaxation of the tensor re-scaling, we get a result with a slight dip in the s_2 center and better representation of the tensor principal directions, although still not as symmetric as when using component-wise remapping with the scaling limiter.

This behavior is seen in the case of remapping a J_2 distribution with diverging principal directions. For example, in Fig. 8 we show the remapped unidirectional distribution object (s_1), where no signs of such symmetry distortions are present for the J_2 remapping method. The scaling limiter even performs worse in this case.

The first row of Table 1 shows the relative discrepancy of the total J_2 summed over the whole domain, essentially a measure of violation of the linear elastic energy conservativity:

$$J_2^\Sigma - J_{2,\text{init}}^\Sigma = \sum_c V^c J_2(S^c) - \sum_c V_{\text{init}}^c J_2(S_{\text{init}}^c).$$

The J_2 remap obviously satisfies this requirement and even if ($w = 1$) relaxation is used, we can still have an order of magnitude increase in the total J_2 accuracy.

The second row shows the L_1 error of J_2 in the smooth and non-linear part of the problem (s_4):

$$L_1^{s_4} [J_2] = \frac{\sum_{c \in s_4} V^c |J_2(S^c) - J_2(S_{\text{init}}^c)|}{\sum_{c \in s_4} V^c J_2(S_{\text{init}}^c)}$$

In this case all high-order methods perform similarly, except for the minimum limiter, which is significantly less accurate. Also it is slightly more expensive in terms of computational time, probably due to the independent calculation of Barth–Jespersen limiters for all three components of the stress tensor.

In the third row, the Frobenius norm of the error is displayed to quantify component-based deviation of the tensor from the initial values:

$$L_F^{s_4} [S] = \frac{\sum_{c \in s_4} \|S^c - S_{\text{init}}^c\|_F}{\sum_{c \in s_4} \|S_{\text{init}}^c\|_F}$$

Here the J_2 remap results are worse than component-based remap, possibly indicating the error caused by principal direction distortion. It also shows that using the relaxation mechanism in J_2 remap can improve the component-based accuracy. However, it is to be noted that the component-wise metric itself is ambiguous in the case of the deviatoric stress tensor – it can include J_2 error as well as orientation-based error and J_3 invariant discrepancy, which is not detailed in this paper.

In the fourth and fifth rows of Table 1 we can see the actual extremal values in the center of s_2 (maximum) and s_1 (minimum). Unlimited high order method violates both bounds, while other methods using limiters guarantee bounds preservation. However, the J_2 remapping method is closest to the actual maximum, the scaling limiter diminishes it by more than 10% and the minimum limiter by almost 30%.

5.3. Strong cylindrical shock in aluminum

We present here an application of the tensor remapping algorithm in a full ALE simulation. In the Lagrangian phase the Wilkins hypo-elastic model [4,15] is used along with the compatible discretization [3] of the elastic–plastic Euler equations. The Mie–Grüneisen equation of state [16] is employed to close the system.

In each step of the computation, the Lagrangian results are remapped onto the initial mesh (the Eulerian as Lagrange+Remap regime). Density and internal energy are remapped in all cases using the second order approach with the Barth–Jespersen limiter and the standard kinetic energy fix. Momentum is remapped using the optimization-based inter-nodal mass fluxes [20]. For a detailed description of the whole remapping algorithm see [6].

We use the following material parameters for aluminum:

$$\Gamma = 2.0, \quad \rho_0 = 2.71 \text{ g cm}^{-3}, \quad \mu = 286 \text{ kbar}, \quad Y_0 = 2.6 \text{ kbar},$$

$$s = 1.34, \quad c_0 = 5.33 \text{ km s}^{-1},$$

where the Grüneisen parameter Γ , reference density and sound speed ρ_0, c_0 and the Hugoniot slope coefficient s are all parameters of the Mie–Grüneisen equation of state. The remaining parameters are used in the linear elastic–plastic model: μ is the shear modulus and Y_0 the von Mises yield strength,

Similarly to the “cylindrical shock tube” [21] test, the pressure in the Al block is initialized in two regions:

$$r < 150 \text{ cm} : p = 1.6 \text{ Mbar}, \quad r > 150 \text{ cm} : p = 0 \text{ bar},$$

$$\rho = \rho_0, \quad \mathbf{u} = \mathbf{0} \quad \text{everywhere.}$$

r indicates the distance from the origin of the coordinate system. In our case much higher pressure than in the original test is used to get a state with large plastic deformation, where remapping effects are clearly visible.

We display the resulting density and pressure using our J_2 remapping algorithm at $t = 10 \mu\text{s}$ in Fig. 9. The cylindrical shock front moves outward, while a rarefaction wave moves inward leaving a contact discontinuity in the middle. The profile of these quantities is similar for all methods, so a comparison was omitted here.

Fig. 10 shows the comparison of the resulting J_2 distribution for all methods. We can see the Lagrangian result (no mesh rezoning) displaying only small artifacts in the directions of the axes (not related to the boundary conditions as they are present in full radial domain simulations as well). The component-based remapping methods create a disturbance in the stress profile in place of the contact discontinuity. As this is caused on the discontinuity in principal directions, using second order remapping does not appear to affect the magnitude of this error considerably. Using J_2 remapping, only a few artifacts are present, probably due to the discontinuity in the principal directions on the contact surface. The plastic threshold is preserved.

Similar results can be seen in the radial plots of the resulting J_2 in Fig. 11. We can see that even the scaling limiter fails to prevent all overshoots, although its results are probably the best from all component-based remapping methods. In this simulation, if J_2 remapping is used there is no visible difference in the results based on whether the relaxation parameter is used or not.

6. Conclusion

We have designed a new method for remapping the 2D deviatoric stress tensor in ALE methods, where the tensor and its second invariant are remapped independently. The tensor is then scaled to match the remapped invariant. Without much overhead, this method is shown to preserve the bounds of the second invariant, while being conservative in the total second invariant (preserving the total strain energy).

This method is frame-invariant and compared to component-based remapping with specialized limiters, it is able to prevent glitches in the stress tensor on contact surfaces under high pressure. Up to a certain point, it can also eliminate dips in stress magnitude in the vicinity of tensor critical points at the cost of principal direction symmetry. A relaxation mechanism was proposed in this paper, which can locally transition to low-order remap and improve the symmetry. However, exact J_2 conservation is violated in such approach.

The topics of future research might contain extending this approach to a full 3D stress tensor (5 independent components), more complex elasticity models and multimaterial remap. There is also room for improvement of the tensor principal directions accuracy in our J_2 remapping scheme.

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 18-20962S and by the Czech Ministry of Education project RVO 68407700. Supported by the project CZ.02.1.01/0.0/0.0/16_019/0000778 from European Regional Development Fund.

References

- [1] 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.
- [2] L.G. Margolin, Introduction to “An arbitrary Lagrangian-Eulerian computing method for all flow speeds”, *J. Comput. Phys.* 135 (2) (1997) 198–202.
- [3] 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.
- [4] M. Wilkins, Calculation of Elastic-Plastic Flow, Tech. Rep. UCRL-7322, California. Univ., Livermore. Lawrence Radiation Lab., 1963.
- [5] L.G. Margolin, M. Shashkov, Second-Order Sign-Preserving Remapping on General Grids, Tech. Rep. LA-UR-02-525, Los Alamos National Laboratory, 2002.
- [6] M. Kucharik, M. Shashkov, Conservative multi-material remap for staggered multi-material Arbitrary Lagrangian–Eulerian methods, *J. Comput. Phys.* 258 (2014) 268–304.
- [7] D.E. Burton, N.R. Morgan, M.R.J. Charest, M.A. Kenamond, J. Fung, Compatible, energy conserving, bounds preserving remap of hydrodynamic fields for an extended ALE scheme, *J. Comput. Phys.* 355 (2018) 492–533.
- [8] P.-H. Maire, Contribution to the numerical modeling of inertial confinement fusion, L’Université Bordeaux I, Habilitation, 2011.
- [9] J. Velechovsky, M. Kucharik, R. Liska, M. Shashkov, Symmetry-preserving momentum remap for ALE hydrodynamics, *J. Phys.: Conf. Ser.* 454 (1) (2013) 012003 IOP Publishing.

- [10] T.J. Barth, Numerical methods for gasdynamic systems on unstructured meshes, in: C.R.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.
- [11] G. Luttwak, J. Falcovitz, Slope limiting for vectors: A novel vector limiting algorithm, *Internat. J. Numer. Methods Fluids* 65 (11–12) (2011) 1365–1375.
- [12] M. Klíma, M. Kuchařík, M. Shashkov, J. Velechovský, Bound-Preserving Reconstruction of Tensor Quantities for Remap in ALE Fluid Dynamics, *Tech. Rep. LA-UR-17-20068*, Los Alamos National Laboratory, 2017.
- [13] C. Lohmann, Flux-corrected transport algorithms preserving the eigenvalue range of symmetric tensor quantities, *J. Comput. Phys.* 350 (2017) 907–926.
- [14] S.K. Sambasivan, M. Shashkov, D.E. Burton, Exploration of new limiter schemes for stress tensors in Lagrangian and ALE hydrocodes, *Comput. Fluids* 83 (2013) 98–114.
- [15] P.-H. Maire, R. Abgrall, J. Breil, R. Loubere, B. Rebourecet, A nominally second-order cell-centered Lagrangian scheme for simulating elastic–plastic flows on two-dimensional unstructured grids, *J. Comput. Phys.* 235 (2013) 626–665.
- [16] R. Menikoff, *Equations of State and Fluid Dynamics*, *Tech. Rep. LA-UR-07-3989*, Los Alamos National Laboratory, 2007.
- [17] J.K. Dukowicz, J.R. Baumgardner, Incremental remapping as a transport/advection algorithm, *J. Comput. Phys.* 160 (1) (2000) 318–335.
- [18] D.J. Mavriplis, Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes, in: *AIAA 2003-3986*, 16th AIAA Computational Fluid Dynamics Conference, June 23–26, Orlando, Florida, NASA, 2003.
- [19] M. Kucharik, *Arbitrary Lagrangian-Eulerian (ALE) Methods in Plasma Physics*, Ph.D. thesis, Czech Technical University in Prague, 2006.
- [20] J.M. Owen, M. Shashkov, Arbitrary Lagrangian Eulerian remap treatments consistent with staggered compatible total energy conserving Lagrangian methods, *J. Comput. Phys.* 273 (2014) 520–547.
- [21] S.K. Sambasivan, M.J. Shashkov, D.E. Burton, A finite volume cell-centered Lagrangian hydrodynamics approach for solids in general unstructured grids, *Internat. J. Numer. Methods Fluids* 72 (7) (2013) 770–810.