Available online at www.sciencedirect.com
ScienceDirect

Comput. Methods Appl. Mech. Engrg. 300 (2016) 402–426

Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Entropy–viscosity method for the single material Euler equations in Lagrangian frame[☆]

Jean-Luc Guermond[a], Bojan Popov[a], Vladimir Tomov[b,*]

[a] Department of Mathematics, Texas A&M University, 3368 TAMU, College Station, TX 77843, USA
[b] Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P. O. Box 808, L-561, Livermore, CA 94551, USA

Received 11 June 2015; received in revised form 31 October 2015; accepted 5 November 2015
Available online 8 December 2015

Abstract

A new finite element method for solving the Euler equations in Lagrangian coordinates is proposed. The method is stabilized by adding artificial diffusion terms compatible with positivity of mass and internal energy, a minimum principle on the specific entropy, and all generalized entropy inequalities. Two options of first-order artificial diffusion are considered. One is in the spirit of the Eulerian based method (Guermond et al., 2011 [23, 22]; Zingan et al., 2013) and the other is similar to existing viscosity stabilizations in Lagrangian frame, e.g., Dobrev et al. (2012). The method is verified to be high-order for smooth solutions even with active viscosity terms. This is achieved by using high-order finite element spaces and an entropy-based viscosity stabilization that degenerates the first-order viscous terms. This stabilization automatically distinguishes smooth and singular regions. The formal accuracy and convergence properties of the proposed methods are tested on a series of benchmark problems. This is the first result extending the entropy–viscosity methodology to the Lagrangian hydrodynamics.

Keywords: Conservation equations; Lagrangian hydrodynamics; Parabolic regularization; Entropy–viscosity; Finite element method

1. Introduction

We are interested in the present paper in simulating the 2D and 3D compressible Euler equations by using a Lagrangian method, i.e., one where the computational mesh moves with the fluid. These methods preserve material interface very well, thus they are the preferred approach for solving problems dealing with compressible multi-phase and multi-material flows, for example numerical simulations of inertial confinement fusion (ICF) and impact problems. We refer the readers to [1] for a review on hydrodynamics methods and detailed comparisons between Lagrangian and Eulerian methods.

[☆] This material is based upon work supported in part by the National Science Foundation grants DMS-1015984 and DMS-1217262, by the Air Force Office of Scientific Research, USAF, under grant/contract number FA99550-12-0358. This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344, LLNL-JRNL-679098.

[*] Corresponding author.
E-mail address: tomov2@llnl.gov (V. Tomov).

http://dx.doi.org/10.1016/j.cma.2015.11.009
0045-7825/© 2015 Elsevier B.V. All rights reserved.

There are roughly three major approaches for solving the Lagrangian hydrodynamics equations: staggered grid hydrodynamics (SGH); cell-centered hydrodynamics (CCH); finite element methods (FEM). SGH schemes represent the thermodynamic variables (density, energy, pressure) as constants over the mesh cells, and the kinematic variables (position, velocity) are evolved at the mesh nodes; see [2–5]. CCH methods define all the flow variables as cell-averages quantities and nodal velocities are reconstructed by using approximate Riemann solvers; see [6–11]. Approaches that combine both SGH and CCH concepts are derived in [12,13].

The purpose of the present paper is to further explore the FEM route. The use of FEM in Lagrangian hydrodynamics has recently re-gained interest after the work of Dobrev et al. [14], where the authors have shown that high-order FEM have excellent properties: geometry representation, symmetry preservation, resolution of shock fronts within one computational cell, high-order convergence for smooth solutions. The FEM formalism has been combined with SGH methods in [15–17] and with CCH methods in [18] in the form of a Discontinuous Galerkin scheme. Another approach that uses curvilinear NURBS (Non-Uniform Rational B-splines) meshes in order to obtain symmetry preservation can be found in [19].

Since hydrodynamics calculations solve problems containing shock waves, special treatment is required in order to avoid post-shock oscillations. One way to resolve this issue is the artificial viscosity approach as originally suggested by Von Neumann and Richtmyer [20]. When combined with FEM, this regularization is uniform with respect to the space dimension, the mesh type and the polynomial degree. High-order convergence can be achieved as long as the added diffusion terms are active only in the singular regions of the solution. Further requirements and specific artificial viscosity terms for the Lagrangian case have been proposed by Caramana et al. [21], Caramana and Loubére [22], Shashkov and Campbell [23], Kolev and Rieben [24], Lipnikov and Shashkov [25].

In this paper we propose a new artificial-viscosity-based high-order finite element method for solving the Euler equations of the compressible gas dynamics in Lagrangian coordinates. We generalize to the Lagrangian framework the ideas presented in [26–28] and derive a regularized Lagrangian system that is compatible with all the generalized entropy inequalities, see [29,28], and the minimum principle on the specific entropy, see [30]. In addition to the standard viscous regularization of the momentum equation, artificial diffusion is added to both the mass and the energy equations, thereby allowing a control of density and energy oscillations near shocks and contact regions. Note that although the mesh moves with the fluid velocity, adding mass viscosity allows material to flow through the mesh and this might create technical difficulties when dealing with multi-material flow problems. This issue is not addressed here since we are only concerned in this paper with single phase flows. This approach is nonetheless useful in non-ideal gas simulations where contacts transform into composite waves. The amount of applied nonlinear dissipation is based on the local residual of the entropy equation; see [31]. One immediate consequence of this choice is that the method automatically distinguishes between smooth and singular regions. The proposed method combines the following features: (i) the equations are regularized in a way that provides control over oscillations around contact discontinuities as well as oscillations in shock regions. (ii) The method produces high-order convergence rates for smooth solutions even with active viscosity terms. This is achieved by using high-order finite element spaces and, more importantly, an entropy-based viscosity that automatically distinguishes between smooth and singular regions; (iii) the proposed diffusion terms are in agreement with the general requirements stated by Kolev and Rieben [24] for artificial tensor viscosities. This work is motivated and influenced by the idea of entropy-production-based artificial viscosity which was developed in [27,32,33] in the Eulerian frame of reference, and the application of high-order finite elements in Lagrangian Hydrodynamics presented in [14]. The finite element framework is similar to that in [14], however, our approach is based on different viscous regularizations, viscosity coefficients, and finite element spaces.

This paper is organized as follows: in Section 2, we discuss the basic entropy principles that motivate the specific form of the viscous regularization, and derive the continuous form of the regularized system in the Lagrangian frame of reference. In Section 3, we describe the finite element discretization and the viscosity coefficients. Numerical test on various model Lagrangian problems are presented in Section 4.

2. Preliminaries

In this section we state the problem and give the theoretical background that motivates the particular form of the viscous regularization that is chosen. We formulate the Lagrangian system of equations that serves as a starting point for the finite element discretization.

We start with the compressible Euler equations in Eulerian coordinates:

$$\frac{\partial}{\partial t}\rho + \nabla\cdot(\boldsymbol{u}\rho) = 0, \tag{1}$$

$$\frac{\partial}{\partial t}(\rho\boldsymbol{u}) + \nabla\cdot(\rho\boldsymbol{u}\otimes\boldsymbol{u}) + \nabla p = 0, \tag{2}$$

$$\frac{\partial}{\partial t}(\rho E) + \nabla\cdot(\boldsymbol{u}\rho E + \boldsymbol{u}p) = 0, \tag{3}$$

where the dependent variables are the density ρ, the momentum $(\rho u_1 \ldots \rho u_d)^T$ and the total energy ρE. We use the usual convention for vector and tensor operations: for column vectors $\boldsymbol{a} = (a_1 \ldots a_d)^T, \boldsymbol{b} = (b_1 \ldots b_d)^T$ and order 2 tensors $\mathbb{g}$, $\mathbb{h}$ with entries g_{ij}, h_{ij} where $i, j = 1 \ldots d$, we have:

$$\boldsymbol{a}\otimes\boldsymbol{b} = \boldsymbol{a}\boldsymbol{b}^T, \qquad (\nabla\boldsymbol{a})_{ij} = \frac{\partial a_j}{\partial x_i}, \qquad (\nabla\cdot\mathbb{g})_i = \sum_{j=1}^{d}\frac{\partial g_{ji}}{\partial x_j},$$

$$(\boldsymbol{a}\cdot\mathbb{g})_i = \sum_{j=1}^{d} a_j g_{ji}, \qquad (\mathbb{g}\cdot\boldsymbol{a})_i = \sum_{j=1}^{d} g_{ij}a_j, \qquad \mathbb{g}:\mathbb{h} = \sum_{ij=1}^{d} g_{ij}h_{ij}.$$

The Euclidean norm in $\mathbb{R}^d$ and $\mathbb{R}^{d\times d}$ is denoted $\|\cdot\|_{\ell^2}$ indifferently. The following standard identities are going to be used throughout this document:

$$\nabla\cdot(\boldsymbol{a}\otimes\boldsymbol{b}) = \boldsymbol{a}\cdot\nabla\boldsymbol{b} + \boldsymbol{b}\nabla\cdot\boldsymbol{a}, \qquad \nabla\cdot(\mathbb{g}\cdot\boldsymbol{a}) = \mathbb{g}:\nabla\boldsymbol{a} + (\nabla\cdot\mathbb{g})\cdot\boldsymbol{a},$$

$$\left[\nabla\left(\frac{\boldsymbol{a}^2}{2}\right)\right]\cdot\boldsymbol{b} = \boldsymbol{a}\cdot(\boldsymbol{b}\cdot\nabla\boldsymbol{a}).$$

We focus our attention to the equation of state of an ideal gas, although most of the arguments in the present paper are independent of the exact nature of the equation of state. The pressure p is given by

$$p = (\gamma - 1)\rho e, \tag{4}$$

where γ is the ratio of constant pressure and constant volume heat capacities. Instead of using the conservation equation for the total energy (3), a common practice in Lagrangian gas dynamics consists of working with the balance equation for the internal energy:

$$\frac{\partial}{\partial t}(\rho e) + \nabla\cdot(\boldsymbol{u}\rho e) + p\nabla\cdot\boldsymbol{u} = 0. \tag{5}$$

Throughout this document we also use the vector form of the system (1)–(3) to simplify the notation:

$$\frac{\partial\boldsymbol{v}}{\partial t} + \sum_{m=1}^{d}\frac{\partial\boldsymbol{f}^{\boldsymbol{m}}(\boldsymbol{v})}{\partial x_m} = 0, \tag{6}$$

where $\boldsymbol{v} = (\rho, \rho u_1, \ldots, \rho u_d, \rho E)^T$, and $\boldsymbol{f} = (\boldsymbol{f}^{\boldsymbol{1}} \ldots \boldsymbol{f}^{\boldsymbol{d}})^T$ are the fluxes for each dimension.

2.1. Entropy principles

Due to the nonlinearities in (6), solutions that are initially smooth may become discontinuous within finite time. Hence, solutions are defined globally in time by considering the equations in the distributional sense. However, the resulting weak problem may admit infinitely many weak solutions, and a selection mechanism for determining whether a weak solution is physical or not must be invoked. A weak solution $\boldsymbol{v}$ of (6) is called admissible in the vanishing viscosity sense if it is the limit of solutions $\boldsymbol{v}^\delta$ in $L^1_{x,t,\text{loc}}$ as $\delta \to 0$ where $\boldsymbol{v}^\delta$ are the solutions of the viscous equation

$$\frac{\partial\boldsymbol{v}^\delta}{\partial t} + \nabla\cdot\boldsymbol{f}(\boldsymbol{v}^\delta) = \delta\Delta\boldsymbol{v}^\delta, \qquad \boldsymbol{v}^\delta(\cdot, 0) = \boldsymbol{v}^0. \tag{7}$$

If a sequence of solutions of (7) can be proved to converge in $L^1_{x,t,\mathrm{loc}}$, then one can show that the limit is a weak solution. While the convergence has been established in the scalar case and for some specific systems, see [34–38], the result for the case of the Euler equations is still an open problem. In the rest of this paper, we assume that the solutions of (7) converge as $\delta \to 0$ and we use the viscous limit as the criterion for selecting a "physical" weak solution. This result is known to hold for any hyperbolic systems in one space dimensions with small initial data, see [39] for details. With this assumption, Eq. (7) is used to deduce other conditions which are easier to verify numerically. This is done through a function called "entropy", which is a generalization of the thermodynamical entropy.

Definition 1. A function $S(\boldsymbol{v})$ is an entropy function for the system (6), if:

1. There is vector-valued function $\boldsymbol{F}(\boldsymbol{v})$, called entropy flux, such that the following identity holds $\partial_{\boldsymbol{v}} S^T(\partial_{\boldsymbol{v}} \boldsymbol{f}^m) = (\partial_{\boldsymbol{v}} F^m)^T, m = 1...d$,
2. S is a convex function of $\boldsymbol{v}$.

Now we derive an entropy S that is compatible with Definition 1. Note that the first condition in the above definition implies (after taking a dot product of (6) with $\partial_{\boldsymbol{v}} S$) that:

$$\frac{\partial S(\boldsymbol{v})}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{v}) = 0 \tag{8}$$

for every smooth solution $\boldsymbol{v}$.

We follow the approach first introduced by Harten [40]. For the EOS $p = (\gamma - 1)\rho e$ and $T = (\gamma - 1)e$, we define the quantity

$$s(\rho, e) := \log(p\rho^{-\gamma}) = \log((\gamma - 1)e\rho^{1-\gamma}) = \log((\gamma - 1)e) + (1 - \gamma) \log \rho, \tag{9}$$

which we call "specific entropy". Notice that this is not exactly the physical specific entropy (it does not satisfy $Tds = de - \frac{p}{\rho^2} d\rho$). Its derivatives with respect to density and energy are

$$\frac{\partial s}{\partial \rho} = \frac{1 - \gamma}{\rho}, \qquad \frac{\partial s}{\partial e} = \frac{1}{e}. \tag{10}$$

For smooth solutions the specific entropy s satisfies $\frac{\partial s(\boldsymbol{v})}{\partial t} + \boldsymbol{u} \cdot \nabla s(\boldsymbol{v}) = 0$. Note that we abuse the notation by using indifferently $s(\boldsymbol{v})$ or $s(\rho, e)$. For any scalar differentiable function $f(s)$, the conservation of mass implies that

$$\frac{\partial}{\partial t}(\rho f(s)) + \nabla \cdot (\boldsymbol{u} \rho f(s)) = 0. \tag{11}$$

We set

$$S := -\rho f(s), \qquad \boldsymbol{F} := -\boldsymbol{u} \rho f(s) \tag{12}$$

and arrive at the entropy equation (8). It is shown in [29] that the function S is a convex function of the conservative variables if and only if

$$f'(s) > 0, \qquad f'(s) \frac{\gamma - 1}{\gamma} + f''(s) > 0. \tag{13}$$

A family of functionals that satisfy (13) is

$$f(s) = \frac{\gamma + \alpha}{\gamma - 1} \exp\left(\frac{s}{\gamma + \alpha}\right) \Rightarrow f'(s) = \frac{1}{\rho(\gamma - 1)} (p\rho^\alpha)^{\frac{1}{\gamma + \alpha}},$$

where $\alpha > 0$ is some constant. Then we have the following family of entropies:

$$S = -\rho f(s) = \frac{\gamma + \alpha}{1 - \gamma} (p\rho^\alpha)^{\frac{1}{\gamma + \alpha}}, \tag{14}$$

so that S satisfies both conditions of Definition 1 for any $\alpha > 0$.

Based on the results in [29,30], the following "physical" properties are satisfied by strong viscosity limits:

- Positivity of density and internal energy.
- Entropy inequality. A weak solution of (6) is entropy-admissible if it satisfies the inequality
$$\frac{\partial S(\boldsymbol{v})}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{v}) \leq 0, \tag{15}$$
in the distribution sense for every pair of entropy pair $(S, \boldsymbol{F})$.
- Minimum principle on the specific entropy, derived by Tadmor [30]: if $\boldsymbol{v}(\boldsymbol{x}, t)$ is a weak solution of (6) that satisfies the entropy inequality (15), then for the specific entropy s defined in (9) we have
$$\operatorname*{Ess\,inf}_{\boldsymbol{x}\in\Omega} s(\boldsymbol{x}, t) \geq \operatorname*{Ess\,inf}_{\boldsymbol{x}\in\Omega+tU} s(\boldsymbol{x}, 0), \tag{16}$$
for all $t > 0$ and any domain Ω, where U is the maximum of $\|\boldsymbol{u}\|_{\ell^2}$ in Ω.

Remark 2.1. When there is a source Q of internal energy, the entropy equation (11) becomes: $\frac{\partial}{\partial t}(\rho f(s)) + \nabla \cdot (\rho f(s)) - Q s_e f'(s) = 0$, i.e.,

$$\frac{\partial S}{\partial t} + \nabla \cdot S + \frac{Q f'(s)}{e} = 0. \tag{17}$$

Remark 2.2. The amount of violation of (11) or (17) is called "entropy production". It can be non-zero only in regions of discontinuities of the solution. We will use this quantity to scale the artificial viscosity terms, i.e., it will be our shock detector (and hence the name "Entropy–Viscosity method").

Remark 2.3. We usually choose $\alpha = 1$ in the definition of S in (14). For smaller values of α the explicit density dependence decreases, for example $\alpha = 0$ would imply that S is smooth in contact regions (since the pressure is continuous across contact discontinuities). In that case the entropy production there is zero and the corresponding shock detector would not add viscosity in contacts. The key reason for this behavior is that the function S is not strictly convex when $\alpha = 0$.

Remark 2.4. We have a minus sign in the definitions of S and $\boldsymbol{F}$, because we want to be consistent with (13). The specific entropy used by Harten et al. [29] is that in (9) multiplied by a negative constant, namely it is $\log(\rho e^{1/(1-\gamma)})$ (which is minus the physical specific entropy).

2.2. *Viscous regularization*

Artificial viscous terms must be introduced to be consistent with entropy dissipation, see [28], and to preserve invariant domains, see [41,42]. A common way to regularize (1)–(3) consists of adding diffusion terms that are similar to the viscosity and thermal diffusion in the Navier–Stokes equations. This approach, however, is in agreement with the minimum principle of the specific entropy (16) only if the thermal diffusion is zero, see Theorem 8.2.3 in [43]. If the thermal diffusion is absent, there is no mechanism to control contact discontinuities, since in contact regions the velocity is constant (i.e., there is no compression) and the viscosity is inactive since the gradient of the velocity is zero across the contact. In the Eulerian frame, one would see uncontrolled oscillations resulting from the Gibbs effect as these contact discontinuities move through the computational mesh. This problem can be concealed in Lagrangian methods by aligning the initial contact discontinuities with the cell boundaries and by using discontinuous spaces, but the above problem nevertheless re-appears if contact lines or surfaces form in time.

Our goal is to satisfy the entropy inequality (15) and minimum principle on the specific entropy (16). It is shown in [28] that to satisfy both these requirement, one needs to add mass and thermal viscosity, in addition to the standard momentum viscosity. This makes the method more diffusive than those using the Navier–Stokes regularization approach only, but it is essential for removing oscillation both in contact and shock regions. A regularization that takes into account all the above considerations is described in [28] and has the form

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = \nabla \cdot \boldsymbol{w}, \tag{18}$$

$$\frac{\partial}{\partial t}(\rho \boldsymbol{u}) + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) + \nabla p = \nabla \cdot \mathrm{g}, \tag{19}$$

$$\frac{\partial}{\partial t}(\rho E) + \nabla \cdot (\rho E \boldsymbol{u} + p \boldsymbol{u}) = \nabla \cdot (\boldsymbol{h} + \mathrm{g} \cdot \boldsymbol{u}), \tag{20}$$

where the viscous terms are:

$$\boldsymbol{w} = \lambda \nabla \rho, \qquad \mathrm{g} = \nu \rho \nabla \boldsymbol{u} + \boldsymbol{w} \otimes \boldsymbol{u}, \qquad \boldsymbol{h} = \lambda \nabla (\rho e) - \frac{\boldsymbol{u}^2}{2} \boldsymbol{w}.$$

Here λ and ν are coefficients that control the amount of added diffusion. These coefficients must have units of (speed × distance).

2.3. Lagrangian formulation

We want to solve the system (18)–(20) in the Lagrangian frame of reference. We can think of the medium as a set of particles having original positions $\boldsymbol{x}_0$. In the Lagrangian setting these particles move with the fluid velocity, namely

$$\frac{d}{dt} \boldsymbol{x}(\boldsymbol{x}_0, t) := \boldsymbol{u}(\boldsymbol{x}, t).$$

Then the material derivative (also called total, Lagrangian, convective, etc.) of a scalar- or vector-valued function $\beta = \beta(\boldsymbol{x}(\boldsymbol{x}_0, t), t)$ is

$$\frac{d}{dt} \beta(\boldsymbol{x}(\boldsymbol{x}_0, t), t) = \frac{\partial \beta(\boldsymbol{x}, t)}{\partial t} + \boldsymbol{u}(\boldsymbol{x}, t) \cdot \nabla_{\boldsymbol{x}} \beta(\boldsymbol{x}, t).$$

Now we rewrite Eqs. (18)–(20) by using the total derivatives. The conservation of mass equation (18) becomes

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \boldsymbol{u} + \nabla \cdot (\lambda \nabla \rho).$$

The velocity equation (19) becomes

$$\rho \frac{d\boldsymbol{u}}{dt} = -\nabla p + \nabla \cdot (\nu \rho \nabla \boldsymbol{u}) + (\lambda \nabla \rho) \cdot (\nabla \boldsymbol{u}). \tag{21}$$

The equation for total energy (20) becomes

$$\rho \frac{dE}{dt} = -\nabla \cdot (p \boldsymbol{u}) + \nabla \cdot (\lambda \rho \nabla e + \nu \rho \nabla \boldsymbol{u} \cdot \boldsymbol{u}) + (\lambda \nabla \rho) \cdot \nabla E. \tag{22}$$

Working with the equation for the internal energy instead of (22) is more common in Lagrangian formulation. Taking the dot product of (21) with $\boldsymbol{u}$ and subtracting the result from (22) we derive

$$\rho \frac{de}{dt} = -p \nabla \cdot \boldsymbol{u} + \nabla \cdot (\lambda \rho \nabla e) + \nu \rho \nabla \boldsymbol{u} : \nabla \boldsymbol{u} + (\nabla e) \cdot (\lambda \nabla \rho).$$

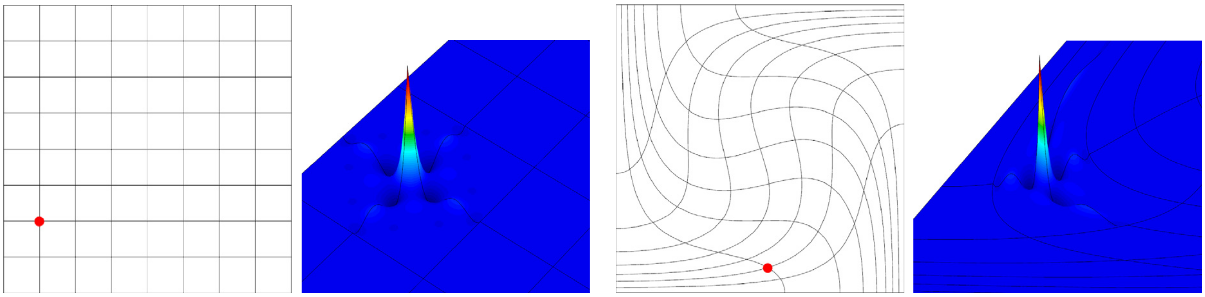
Then the final Lagrangian system that we propose is:

$$\frac{d}{dt} \boldsymbol{x}(\boldsymbol{x}_0, t) = \boldsymbol{u}(\boldsymbol{x}, t), \tag{23}$$

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \boldsymbol{u} + \nabla \cdot (\lambda \nabla \rho), \tag{24}$$

$$\rho \frac{d\boldsymbol{u}}{dt} = -\nabla p + \nabla \cdot (\nu \rho \nabla \boldsymbol{u}) + (\lambda \nabla \rho) \cdot (\nabla \boldsymbol{u}), \tag{25}$$

$$\rho \frac{de}{dt} = -p \nabla \cdot \boldsymbol{u} + \nabla \cdot (\lambda \rho \nabla e) + \nu \rho \nabla \boldsymbol{u} : \nabla \boldsymbol{u} + (\nabla e) \cdot (\lambda \nabla \rho). \tag{26}$$



(a) Original mesh and Q_4 shape function.

(b) Transformed mesh using the Taylor–Green vortex motion and Q_4 shape function.

Fig. 1. Example of a Q_4 shape function on the original mesh and transformed mesh.

The dependent variables are the position $\boldsymbol{x}(\boldsymbol{x}_0, t)$, the density $\rho(\boldsymbol{x}, t)$, velocity $\boldsymbol{u}(\boldsymbol{x}, t)$, the specific internal energy $e(\boldsymbol{x}, t)$. We recall that the equation of state (4) is $p = (\gamma - 1)\rho e$. Note that the system (24)–(26) is written in non-conservative form, which is standard in the Lagrangian hydrodynamics literature.

3. Discretization details

In this subsection we derive a fully-discrete finite element method for the system (23)–(26). We propose a semi-discrete form with high-order continuous finite element spaces and discuss the notions of mesh representation, mappings, length scale, viscosity coefficients, and time discretization.

3.1. Notation

Let Ω_0 be the domain at time 0 and $\Omega(t)$ be the domain at time t. Then $\{\mathcal{K}_h\}_{h>0}$ is a mesh sequence with no hanging nodes that discretizes Ω_0. As we progress in time, the initial mesh is deformed by moving the Lagrange nodes of the mesh according to the rule (23) where the $\boldsymbol{u}$ is made explicit in time. The reference cell is denoted $\widehat{K}$. Given a cell $K_0 \in \mathcal{K}_h$, we denote by $\Phi_{K_0} : \widehat{K} \to K_0$ the geometric transformation. As time increases, K_0 is deformed into K and we denote $\Phi_K : \widehat{K} \to K$ the corresponding geometric transformation; details on how these transformations are constructed are given in Section 3.3. Note that Φ_K and K are time-dependent, but we henceforth abuse the notation by omitting the time index. A natural mapping from K_0 to K is given by $\Phi_K \circ \Phi_{K_0}^{-1} : K_0 \to K$; note that $\Phi_K \circ \Phi_{K_0}^{-1}$ is the identity at $t = 0$. To simplify the notation we introduce $\boldsymbol{x} := \Phi_K(\widehat{\boldsymbol{x}})$ and $\boldsymbol{x}_0 := \Phi_{K_0}(\widehat{\boldsymbol{x}})$ for any $\widehat{\boldsymbol{x}} \in \widehat{K}$.

We introduce a scalar-valued nodal finite element space defined with respect to the initial mesh K_0 as follows:

$$Q_k = \{v \in C^0(\Omega_0);\, v|_{K_0} \circ \Phi_{K_0} \in \mathbb{Q}_k, \forall K_0 \in \mathcal{K}_h\}$$

where $\mathbb{Q}_k$ is the set of multivariate polynomials of partial degree at most k. Let $\{\boldsymbol{a}_1, \dots, \boldsymbol{a}_N\}$ be the Lagrange nodes of the mesh and let $\{\varphi_1, \dots, \varphi_N\}$ be the associated nodal shape functions, i.e., $\varphi_i(\boldsymbol{a}_j) = \delta_{ij}$. Since the mesh nodes move, we define the time-dependent basis functions by

$$\varphi(\boldsymbol{x}, t) := \varphi(\boldsymbol{x}_0) \qquad \left(\text{equivalent to } \varphi(\boldsymbol{x}, 0) := \varphi(\boldsymbol{x}_0),\ \frac{d}{dt}\varphi(\boldsymbol{x}, t) := 0\right), \tag{27}$$

where by convention $\boldsymbol{x} = \Phi_K \circ \Phi_{K_0}^{-1}(\boldsymbol{x}_0)$. Note that here again we abuse the notation by using the same symbol for the reference shape function and the transformed shape function. Examples of Q_4 basis functions on the original and transformed meshes are shown in Figs. 1(a) and 1(b). Note that approximation using such functions stays H^1-conforming in time. Shape functions on the reference cell are denoted by $\widehat{\varphi}_i(\widehat{\boldsymbol{x}})$, $i = 1 \dots \widehat{N}$.

From this point forward, by $\rho, \boldsymbol{u}, e, \boldsymbol{x}, p, S$ we refer to the discrete versions of the dependent variables in Q_k. Quantities that only depend on $\boldsymbol{x}_0$ are defined at the initial time, for example $\rho(\boldsymbol{x}_0)$ is the approximation of the initial density.

Remark 3.1. The methodology presented in this paper works in principle for any type of continuous finite elements (it is not restricted to quadrangular and hexahedral meshes) but in this paper we only consider Q_k approximations in the implementation of the method.

3.2. Semi-discrete form

We obtain a semi-discrete form of the problem by multiplying every equation in (24)–(26) with a test function and integrating over $\Omega(t)$. We use spaces of same polynomial degree for all the dependent variables, namely we seek $(\rho, \boldsymbol{u}, e, \boldsymbol{x}) \in (Q_k \times Q_k^d \times Q_k \times Q_k^d)$. Our tests did not show any significant benefits in using different polynomial degrees for the kinematic and thermodynamic spaces.

- Density—for every $j = 1 \dots N$ we have

$$\int_{\Omega(t)} \frac{d\rho(\boldsymbol{x},t)}{dt}\varphi_j(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x} = -\int_{\Omega(t)} \rho(\boldsymbol{x},t)\nabla\cdot\boldsymbol{u}(\boldsymbol{x},t)\varphi_j(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x} - \int_{\Omega(t)} \lambda\nabla\rho(\boldsymbol{x},t)\cdot\nabla\varphi_j(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x}. \tag{28}$$

- Velocity—for every dimension $m = 1 \dots d$ and every $j = 1 \dots N$ we have

$$\begin{aligned}\int_{\Omega(t)} \rho(\boldsymbol{x},t)\frac{du_m(\boldsymbol{x},t)}{dt}\varphi_j(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x} &= -\int_{\Omega(t)} \frac{\partial p(\boldsymbol{x},t)}{\partial x_m}\varphi_j(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x}\\ &- \int_{\Omega(t)} \nu\rho\nabla u_m(\boldsymbol{x},t)\cdot\nabla\varphi_j(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x} + \int_{\Omega(t)} \lambda\nabla u_m(\boldsymbol{x},t)\cdot\nabla\rho(\boldsymbol{x},t)\varphi_j(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x}.\end{aligned} \tag{29}$$

- Internal energy—for every $j = 1 \dots N$ we have

$$\begin{aligned}\int_{\Omega(t)} \rho(\boldsymbol{x},t)\frac{de(\boldsymbol{x},t)}{dt}\varphi_j(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x} &= -\int_{\Omega(t)} p(\boldsymbol{x},t)\nabla\cdot\boldsymbol{u}(\boldsymbol{x},t)\varphi_j(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x}\\ &- \int_{\Omega(t)} \lambda\rho(\boldsymbol{x},t)\nabla e(\boldsymbol{x},t)\cdot\nabla\varphi_j(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x} + \int_{\Omega(t)} \lambda\nabla e(\boldsymbol{x},t)\cdot\nabla\rho(\boldsymbol{x},t)\varphi_j(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x}\\ &+ \int_{\Omega(t)} \nu\rho(\boldsymbol{x},t)\nabla\boldsymbol{u}(\boldsymbol{x},t) : \nabla\boldsymbol{u}(\boldsymbol{x},t)\varphi_j(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x}.\end{aligned} \tag{30}$$

And the position function $\boldsymbol{x}(\boldsymbol{x}_0, t)$ is simply evolved by solving $\frac{d}{dt}\boldsymbol{x}(\boldsymbol{x}_0, t) = \boldsymbol{u}(\boldsymbol{x}, t)$ with the initial condition $\boldsymbol{x}(\boldsymbol{x}_0, 0) = \boldsymbol{x}_0$ using the chosen time integrator.

Remark 3.2. One drawback of adding mass viscosity is that the resulting mass matrices are time-dependent. For example, the left-hand side of (30) and the corresponding matrix are

$$\int_{\Omega(t)} \rho(\boldsymbol{x},t)\frac{d}{dt}\left(\sum_{i=1}^{N} e_i(t)\varphi_i(\boldsymbol{x},t)\right)\varphi_j(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x} = \sum_{i=1}^{N}\frac{de_i(t)}{dt}\int_{\Omega(t)} \rho(\boldsymbol{x},t)\varphi_i(\boldsymbol{x},t)\varphi_j(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x},$$

$$(M_e)_{ij} = \int_{\Omega_0} \rho(\boldsymbol{x}(\boldsymbol{x}_0,t),t)\varphi_i(\boldsymbol{x}_0)\varphi_j(\boldsymbol{x}_0)|\det J_{\boldsymbol{x}_0\to\boldsymbol{x}}|\,\mathrm{d}\boldsymbol{x}_0.$$

It is not true here that $\rho(\boldsymbol{x}(\boldsymbol{x}_0,t),t)|\det J_{\boldsymbol{x}_0\to\boldsymbol{x}}| = \rho(\boldsymbol{x}_0)$, which would otherwise be the case if we did not have the additional viscous term in the mass equation, see [14].

Remark 3.3. Note that no special effort is made here to make the method exactly conservative. Actually the method, semi-discretized in space as explained above but with continuous time, is conservative. More precisely, summing (28) over j and using the partition of unity property, we have $\int_{\Omega(t)} \frac{d\rho(\boldsymbol{x},t)}{dt}\,\mathrm{d}\boldsymbol{x} + \int_{\Omega(t)} \rho(\boldsymbol{x},t)\nabla\cdot\boldsymbol{u}(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x} = 0$, which, owing to Reynolds Theorem, gives that the time derivative of $\int_{\Omega(t)} \rho(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x}$ is zero; whence conservation of mass. This property does not hold though when time discretization is applied; strictly speaking the fully discrete algorithm as described in Section 3.7 is not conservative.

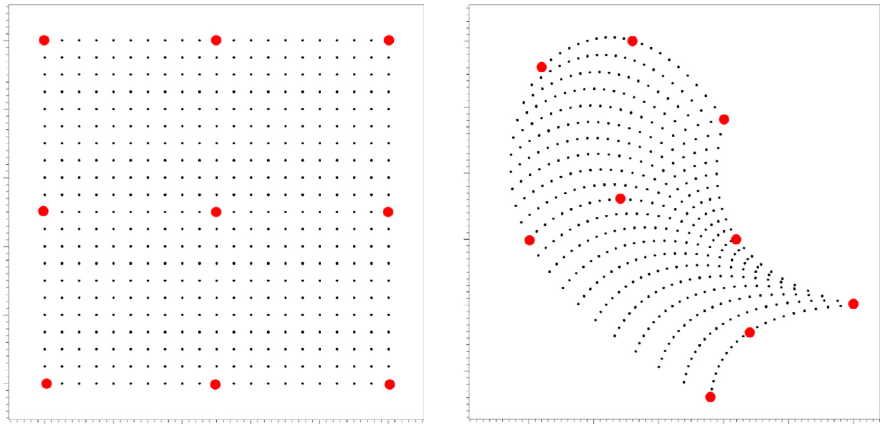


Fig. 2. Example of a Q_2 mapping between the reference and the current element.

3.3. Mesh representation and mappings

Assuming again that the velocity field $\boldsymbol{u}(\boldsymbol{x},t)$ is known, the motion of the mesh is controlled by the motion of the nodes $\{\boldsymbol{a}_1 \dots \boldsymbol{a}_N\}$, i.e., we define the functions $\{\boldsymbol{X}_1(t) \dots \boldsymbol{X}_N(t)\}$ by

$$\frac{\mathrm{d}\boldsymbol{X}_i(t)}{\mathrm{d}t} = \boldsymbol{u}(\boldsymbol{X}_i(t),t), \qquad \boldsymbol{X}_i(0) = \boldsymbol{a}_i. \tag{31}$$

Then the trajectory of a particle with original position $\boldsymbol{x}_0$ is obtained by using the finite element expansion

$$\boldsymbol{x}(\boldsymbol{x}_0,t) = \sum_{j=1}^{N} \boldsymbol{X}_j(t)\varphi_j(\boldsymbol{x}_0). \tag{32}$$

The use of high-order polynomial basis functions implies that the nodes are connected by high-order curves and the mesh is curvilinear. Notice that in order to obtain the position of any point in the computational mesh, we only need the positions of the nodes. This approach is very efficient since it does not involve any complicated curve reconstructions.

Let $j : \mathcal{K}_h \times \{1 \dots \widehat{N}\} \ni (K,i) \longmapsto j(K,i) \in \{1 \dots N\}$ be the connectivity array associating the reference shape functions with the global shape functions over each cell, i.e., $\varphi_{j(K,i)}(\Phi_K(\widehat{\boldsymbol{x}})) = \widehat{\varphi}_i(\widehat{\boldsymbol{x}})$ for all $\widehat{\boldsymbol{x}} \in \widehat{K}$ for all $t \geq 0$. The formula (32) provides a straightforward way to define the time-dependent position mapping $\Phi_K(\cdot,t) : \widehat{K} \ni \widehat{\boldsymbol{x}} \longmapsto \boldsymbol{x} \in K$ from the reference cell $\widehat{K}$ to an actual cell of interest K:

$$\Phi_K(\widehat{\boldsymbol{x}},t) := \boldsymbol{x}(\widehat{\boldsymbol{x}},t) = \sum_{i=1}^{\widehat{N}} \boldsymbol{X}_{j(K,i)}(t)\widehat{\varphi}_i(\widehat{\boldsymbol{x}}), \tag{33}$$

where recall that $\widehat{\varphi}_1 \dots \widehat{\varphi}_{\widehat{N}}$ are the basis functions on the reference cell. The time-dependent Jacobian matrix of this mapping is then simply

$$J(\widehat{\boldsymbol{x}},t) := \frac{\partial \boldsymbol{x}}{\partial \widehat{\boldsymbol{x}}} = \sum_{i=1}^{\widehat{N}} \boldsymbol{X}_{j(K,i)}(t) \otimes \nabla\widehat{\varphi}_i(\widehat{\boldsymbol{x}}). \tag{34}$$

Note that with this definition J has the usual form:

$$J_{kl} = \frac{\partial x_k}{\partial \widehat{x}_l}, \qquad k,l = 1 \dots d. \tag{35}$$

At $t = 0$, we have $\Phi_{K_0} := \Phi_K(\cdot,0) : \widehat{\boldsymbol{x}} \to \boldsymbol{x}_0$ with Jacobian $J_0(\widehat{\boldsymbol{x}}) := J(\widehat{\boldsymbol{x}},0)$. Later in the text, we also use the mapping $\Phi_K \circ \Phi_{K_0}^{-1} : \boldsymbol{x}_0 \to \boldsymbol{x}$ with Jacobian JJ_0^{-1}. A realization of the mapping $(\Phi_K)_{K\in\mathcal{K}_k}$ using the Q_2 space is shown in Fig. 2. For instance, looking at this figure, we understand that (33) tells us that we can obtain the position of the black points on the right side of the current element by using only the positions of the red dots on the right side of the reference cell.

3.4. Length scales

Artificial viscosity coefficients (λ, ν) must scale like a speed times a distance. This is usually done in Eulerian methods by defining (i) a mesh dependent length scale and (ii) a shock-capturing quantity, for example the entropy production. We are going to adapt this technique to the Lagrangian situation. The key difference is that now the mesh moves and the notion of meshsize is dynamic and therefore must be revised. In this subsection we define three different length scales and we match them with specific coefficients in the next subsection. All the scales are defined pointwise in order to be computable at Gauss points and be compatible with the usage high order polynomial spaces.

We start by defining a smooth initial mesh length scale function $h_0(\boldsymbol{x})$: (i) On each initial cell K_0, we define $h^*(\boldsymbol{x}) = \frac{1}{k}|K_0|^{\frac{1}{d}}$ for all $\boldsymbol{x} \in K_0$ and set $h^*_{\min} = \min_{\boldsymbol{x}\in\Omega_0} h^*(\boldsymbol{x})$, where recall that k is the degree of the polynomials in Q_k. (ii) If $h^*(\boldsymbol{x})$ is smooth (i.e., the initial mesh is uniform), then we set $h_0 = h^*$, otherwise h_0 is computed by a smoothing procedure with some smoothing constant ε:

$$\int_{\Omega_0} h_0(\boldsymbol{x})\varphi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + (\varepsilon h^*_{\min})^2 \int_{\Omega_0} \nabla h_0(\boldsymbol{x}) \cdot \nabla\varphi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int_{\Omega_0} h^*(\boldsymbol{x})\varphi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}. \tag{36}$$

We approximate h_0 in the same finite element space as the dependent variables.

Once $h_0(\boldsymbol{x})$ is known, we define three length scales by using the mapping $\Phi_K \circ \Phi_{K_0}^{-1} : \boldsymbol{x}_0 \to \boldsymbol{x}$ with Jacobian JJ_0^{-1}:

1. Recalling that $\|\cdot\|_{\ell^2}$ is the Euclidean norm in $\mathbb{R}^d$, $h_1(\boldsymbol{x})$ is defined as a perturbation of the initial mesh in the direction of the current motion $\boldsymbol{u}(\boldsymbol{x}, t)$:
$$h_1(\boldsymbol{x}(\boldsymbol{x}_0, t)) = h_0(\boldsymbol{x}_0)\frac{\|JJ_0^{-1}(\boldsymbol{x}_0)\boldsymbol{u}(\boldsymbol{x}, t)\|_{\ell^2}}{\|\boldsymbol{u}(\boldsymbol{x}, t)\|_{\ell^2}}. \tag{37}$$
2. $h_2(\boldsymbol{x})$ is defined as in [14], i.e., it is a perturbation of the initial mesh in the direction of maximal compression:
$$h_2(\boldsymbol{x}(\boldsymbol{x}_0, t)) = h_0(\boldsymbol{x}_0)\frac{\|JJ_0^{-1}(\boldsymbol{x}_0)\boldsymbol{s}(\boldsymbol{x})\|_{\ell^2}}{\|\boldsymbol{s}(\boldsymbol{x})\|_{\ell^2}} \tag{38}$$
where $\boldsymbol{s}(\boldsymbol{x})$ is the eigenvector corresponding to the smallest eigenvalue $\mu(\boldsymbol{x})$ of $\nabla^s\boldsymbol{u}(\boldsymbol{x})$. Here $\mu(\boldsymbol{x})$ is a measure of the maximal compression (or minimal expansion if the value is positive), and $\boldsymbol{s}(\boldsymbol{x})$ is the direction in which this compression occurs.
3. $h_3(\boldsymbol{x})$ is defined to be the measure of the volume change at $\boldsymbol{x}$:
$$h_3(\boldsymbol{x}(\boldsymbol{x}_0, t)) = h_0(\boldsymbol{x}_0)|\det(JJ_0^{-1}(\boldsymbol{x}_0))|. \tag{39}$$

Note that h_1 and h_2 are direction-dependent, whereas h_3 is not. These functions are time-dependent and are not represented in the finite element space, their values are computed whenever needed (usually at quadrature points).

3.5. Viscosity coefficients

As discussed in the previous subsection, the artificial viscosity coefficients (λ, ν) must be an appropriate combination of a length scale and a shock detector, and they should be computed at each quadrature point. We first define the “first-order” viscous coefficients. Let $\boldsymbol{x}^n$ be the position of the quadrature point of interest at time t_n, and let $c_{\text{visc}} \geq 0$ be a tunable constant. We propose two options corresponding to h_1 and h_2:

- **Option 1:**

$$\begin{aligned}\lambda_1^{\text{visc}} &:= c_{\text{visc}} h_1(\boldsymbol{x}^n)\|\boldsymbol{u}^n(\boldsymbol{x}^n)\|_{\ell^2},\\ \nu_1^{\text{visc}} &:= c_{\text{visc}} h_1(\boldsymbol{x}^n)\left(a(\boldsymbol{x}^n) + \|\boldsymbol{u}^n(\boldsymbol{x}^n)\|_{\ell^2}\right),\end{aligned} \tag{40}$$

where $a(\boldsymbol{x}^n)$ is the sound speed. These quantities are not shock detectors since diffusion is active everywhere the velocity is nonzero.

- **Option 2:** (which is used in [14]):

$$\begin{aligned}\lambda_2^{\mathrm{visc}} &:= c_{\mathrm{visc}} h_2^2(\boldsymbol{x}^n)|\mu(\boldsymbol{x}^n)|,\\ \nu_2^{\mathrm{visc}} &:= \begin{cases} c_{\mathrm{visc}} h_2(\boldsymbol{x}^n)\left(a(\boldsymbol{x}^n)+h_2(\boldsymbol{x}^n)|\mu(\boldsymbol{x}^n)|\right) & \mu(\boldsymbol{x}^n)<0,\\ c_{\mathrm{visc}} h_2^2(\boldsymbol{x}^n)|\mu(\boldsymbol{x}^n)| & \text{otherwise,}\end{cases}\end{aligned} \tag{41}$$

where recall that $\mu(\boldsymbol{x}) \leq 0$ in compressive regions. Note that the above definition is formally less diffusive than the first one, but it is nevertheless first-order in compressive regions. Note also that both $\lambda_2^{\mathrm{visc}}$ and ν_2^{visc} are small when the velocity is close to a constant, which is the case, for instance, in contact regions.

Then we define "non-linear" viscosity coefficients by using the property that the entropy production is zero in regions where the solution is smooth (at the continuous level) and non-zero in singular regions. In the Lagrangian frame of reference, Eq. (11) becomes

$$\frac{dS}{dt}+S\nabla\cdot\boldsymbol{u}=0,$$

where S is defined in (12). Then at the discrete level, we define

$$D := \frac{S^n-S^{n-1}}{t_n-t_{n-1}}+S^n\nabla\cdot\boldsymbol{u}^n(\boldsymbol{x}^n), \tag{42}$$

where $S^n = S(\rho^n, e^n)$ is the entropy functional at t_n. If there is an energy source as in (17) then (42) becomes

$$D := \frac{S^n-S^{n-1}}{t_n-t_{n-1}}+S^n\nabla\cdot\boldsymbol{u}^n(\boldsymbol{x}^n)+\frac{Q}{f'(s^n)}e^n, \tag{43}$$

where $s^n = s(\rho^n, e^n)$ is the specific entropy at t_n and Q is the source contribution at the quadrature point of interest. The time derivative of S can be approximated by higher-order backward differencing; this is what is actually done in the convergence tests reported below for smooth solutions. The high-order viscosity coefficient is defined to be

$$\nu^{\mathrm{entr}} := \frac{c_{\mathrm{entr}} h_3^2(\boldsymbol{x}^n)|D|}{\|S^n-\overline{S^n}\|_{L^\infty(\Omega(t_n))}+\epsilon\overline{S^n}}, \qquad \overline{S^n} := \frac{1}{|\Omega(t_n)|}\int_{\Omega(t_n)} S^n dx, \tag{44}$$

where c_{entr} is a tunable constant and $\epsilon = 10^{-10}$. Finally the so-called entropy–viscosity is obtained by taking the minimum of the first-order and high-order viscosities. We consider two options:

- **Option 1:**

$$\lambda_1^n := \min(\lambda_1^{\mathrm{visc}}, \nu^{\mathrm{entr}}), \qquad \nu_1^n := \min(\nu_1^{\mathrm{visc}}, \nu^{\mathrm{entr}}). \tag{45}$$

- **Option 2:**

$$\lambda_2^n := \min(\lambda_2^{\mathrm{visc}}, \nu^{\mathrm{entr}}), \qquad \nu_2^n := \min(\nu_2^{\mathrm{visc}}, \nu^{\mathrm{entr}}). \tag{46}$$

Note that the definitions (45) and (46) are dimensionally coherent since the quantities $\lambda_1^{\mathrm{visc}}$, $\lambda_2^{\mathrm{visc}}$, ν_1^{visc}, ν_2^{visc}, and ν^{entr} scale like a speed times a length. The heuristics for the above definitions is that we expect the first-order viscosity to be the active in shocks and contacts and the high-order vanishing viscosity to be active in the smooth regions.

Using appropriate combinations of length scales and shock detectors is essential for avoiding incorrect mesh behavior, see definitions (40), (41) and (44). The first-order shock detectors depend on the relative orientation of the velocity with respect to the mesh, whereas the entropy production coefficient and its associated length scale are direction-independent. In Option 1, the length scales in (44) are h_1 and h_3, but they are h_2 and h_3 in Option 2. And example of incorrect combination of length scales is shown in Fig. 3. The left panel shows the mesh obtained by solving the Sedov problem (see Section 4.3 for details) using Option 2 with the correct combination of length scales (i.e., h_2 in (41)). The right panel shows the result obtained by using h_1 instead on h_2 in (41).

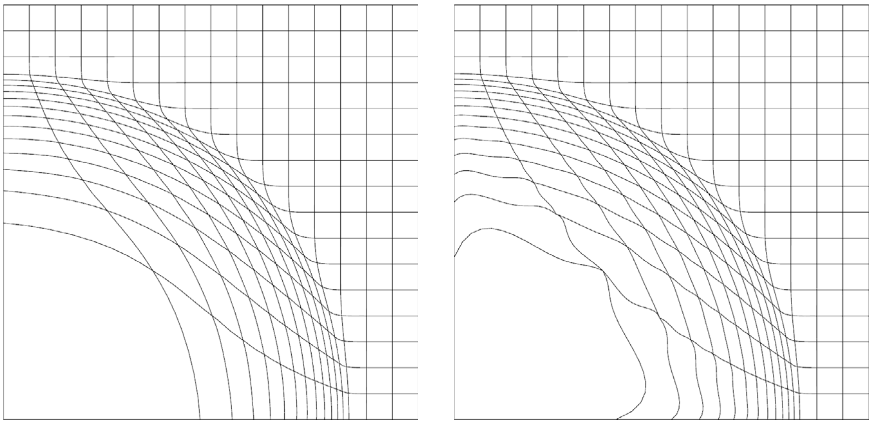


Fig. 3. Resulting meshes from applying the Option 2 coefficients to a Q_4 position function with h_3 (left panel) and h_1 (right panel) in Eq. (44).

Table 1
Butcher tableau for a r-stage explicit RK method.

a_1					
a_2	b_{21}				
a_3	b_{31}	b_{32}			
$\dots$	$\dots$				
a_r	b_{r1}	b_{r2}	$\dots$	$b_{r,r-1}$	
	c_1	c_2	$\dots$	c_{r-1}	c_r

3.6. Consistency with general viscosity requirements

Here we comment on the compatibility of the viscosity tensors with the general requirements for artificial tensor viscosities stated in [23] and [24]. We use the terminology from [24]:

- *The viscous terms must be invariant under orthogonal transformations of $\boldsymbol{x}$ and $\boldsymbol{u}$*: the method satisfies this requirement at the discrete level; the details are given in Appendix A.1.
- *For the specific entropy s, see (9), we must have $\frac{ds}{dt} \geq 0$*: the regularization satisfies a weaker version of this requirement, namely the minimum principle on the specific entropy (16) on continuous level. This result is established in [28, Theorem 3.5].
- *The regularized system must be Galilean invariant*: the system in Lagrangian form (24)–(26) is Galilean invariant. All the viscosity coefficients in option 2, the entropy production D, h_2, h_3 satisfy the requirement at the discrete level. Option 1 and h_1 are not Galilean invariant, though, since they depend explicitly on $\boldsymbol{u}$.
- *The artificial viscosity must preserve radial symmetry*: the symmetry preservation of our method is presented by the numerical tests in Sections 4.3–4.5. In general, achieving radial symmetry on fully-discrete level is a difficult task, because that would require the artificial viscosity, and in particular its length scale part, to be radially symmetric. Length scales, however, typically depend on initial cell sizes and directions, which are not in general radially symmetric. Exact radial symmetry can be achieved in special cases, for example by constructing the initial mesh with curved radial cells obtained from each other by orthogonal transformation, see Section 8.2 in [24].
- *The artificial dissipation in the momentum equation must be zero for linear velocity (uniform contractions, rigid rotations) and all the artificial viscosity terms must be zero in regions of expansion*: these requirements are not satisfied exactly but are satisfied up to the consistency error of the method. More precisely the entropy–viscosity (44) is $\mathcal{O}((\Delta t+h)h^2)$ in smooth regions by construction. (It can be made $\mathcal{O}(\Delta t^2+h)h^2$ by using Crank–Nicolson's or BDF2 approximation in (43), see Section 4.1.)

3.7. Time discretization

We discretize the time derivatives of $(\rho, \boldsymbol{u}, e, \boldsymbol{x})$ by using a standard explicit Runge–Kutta method (i.e., either RK3 or RK4). To be generic, we assume the method is composed of r stages and be fully described by its Butcher tableau (Table 1). Let $\boldsymbol{v} = (\rho, \boldsymbol{u}, e, \boldsymbol{x})$ be the discrete solution at time t. We introduce the discrete operators W_ρ, $\boldsymbol{W_u}$, W_e, $\boldsymbol{W_x}$

defined by solving the following discrete weak versions of (28)–(30):

$$\begin{aligned}
\int_{\Omega(t)} W_\rho(\boldsymbol{v},t)\varphi\,\mathrm{d}\boldsymbol{x} &= \int_{\Omega(t)}\Big(-\rho\nabla\cdot\boldsymbol{u}\varphi - (\lambda\nabla\rho)\cdot\nabla\varphi\Big)\,\mathrm{d}\boldsymbol{x},\\
\int_{\Omega(t)} \rho[\boldsymbol{W_u}(\boldsymbol{v},t)]_m\varphi\,\mathrm{d}\boldsymbol{x} &= \int_{\Omega(t)}\Big(\lambda\nabla u_m\cdot\nabla\rho\varphi - \nu\rho\nabla u_m\cdot\nabla\varphi - \frac{\partial p}{\partial x_j}\varphi\Big)\mathrm{d}\boldsymbol{x}, \qquad m = 1...d,\\
\int_{\Omega(t)} \rho W_e(\boldsymbol{v},t)\varphi\,\mathrm{d}\boldsymbol{x} &= \int_{\Omega(t)}\Big(\nu\rho\|\nabla\boldsymbol{u}\|^2_{\ell^2}\varphi - \lambda\rho\nabla e\cdot\nabla\varphi + (\lambda\nabla e\cdot\nabla\rho)\varphi - p\nabla\cdot\boldsymbol{u}\varphi\Big)\mathrm{d}\boldsymbol{x},\\
\boldsymbol{W_x}(\boldsymbol{v},t) &= \boldsymbol{u}.
\end{aligned}$$

Then upon denoting $\boldsymbol{W}(\boldsymbol{v},t) := (W_\rho, \boldsymbol{W_u}, W_e, \boldsymbol{W_x})$, the solution $\boldsymbol{v}$ is updated at time t_{n+1} by

$$\boldsymbol{v}^{n+1} = \boldsymbol{v}^n + \Delta t\sum_{i=1}^{r} c_i\boldsymbol{k}_i, \qquad \boldsymbol{k}_i := \boldsymbol{W}\left(\boldsymbol{v}^n + \Delta t\sum_{j=1}^{r} b_{ij}\boldsymbol{k}_j, t_n + a_i\Delta t\right).$$

3.8. Time step control

Because of the finite speed of propagation in hyperbolic problems, the time step for explicit methods is restricted by a CFL condition. We then introduce the CFL number c and control the time step by using the following heuristic technique:

1. Given $(\rho^0, \boldsymbol{u}^0, e^0, \boldsymbol{x}^0)$ at time t_0, compute the first time step:

$$\Delta t = \min_{\boldsymbol{x}\in\Omega_n} \frac{c_{\mathrm{cfl}} h_2(\boldsymbol{x})}{\sqrt{\gamma T^0(\boldsymbol{x})} + |\boldsymbol{u}^0(\boldsymbol{x})|}. \tag{47}$$

2. Use Δt to compute the update $(\rho^{n+1}, \boldsymbol{u}^{n+1}, e^{n+1}, \boldsymbol{x}^{n+1})$ and compute:

$$\Delta t^* = \min_{\boldsymbol{x}\in\Omega_{n+1}} \frac{c_{\mathrm{cfl}} h_2(\boldsymbol{x})}{\sqrt{\gamma T^{n+1}(\boldsymbol{x})} + |\boldsymbol{u}^{n+1}(\boldsymbol{x})|}.$$

3. If $\Delta t^* < \Delta t \Rightarrow \Delta t = 0.9\Delta t$, discard the update and go to step 2 (i.e., repeat the step).
4. Else, set $n = n + 1$. If $\Delta t^* > 1.25\Delta t$ then set $\Delta t = 1.02\Delta t$. Set $t = t + \Delta t$ and go to step 2.

4. Numerical tests

In this section we illustrate the behavior of the method on standard Lagrangian hydrodynamics test cases with known exact solutions. First we use a smooth solution to estimate the convergence properties of the method. Then, we use a one-dimensional Riemann problem to demonstrate the shock-capturing properties of the entropy production-based viscosity coefficients and the non-oscillating behavior of the method in contact regions. Finally, we turn to two-dimensional shock problems to test the symmetry properties of the method.

For all the test cases, we solve the linear systems associated with the mass matrix by using a conjugate gradient algorithm with a diagonal Jacobi preconditioner. The list of the parameters used in each of the test cases is reported in Appendix A.2.

The method is made parallel by using the parallel finite element method library [44]. The results are visualized by using the OpenGL visualization tool [45].

4.1. 2D Taylor–Green vortex

The goal of this test case is to demonstrate that the method achieves high-order convergence rates on smooth solutions. This is essentially a consistency test. All the simulations are done by keeping the viscosity terms active in Eqs. (28)–(30) (i.e., we do not set them to zero explicitly). This confirms the convergence to zero of the entropy production-based viscosity coefficients on discrete level.

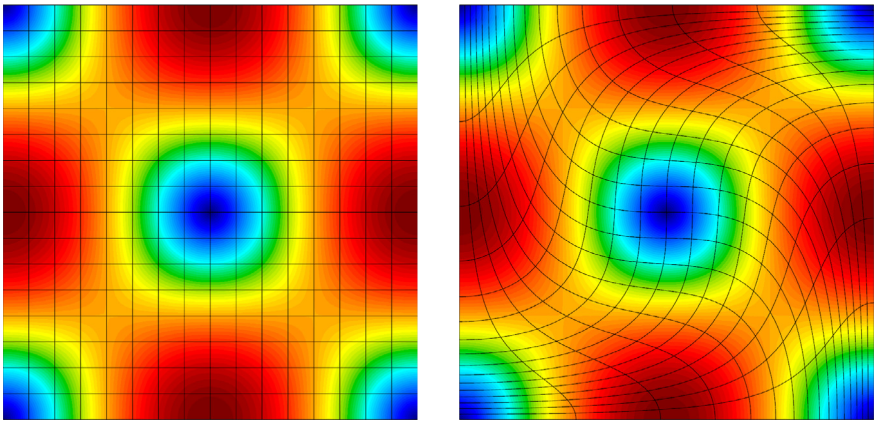


Fig. 4. Velocity magnitude on the initial mesh (left panel), and on the final mesh (right panel) computed with the Q_4 FE space using 16 × 16 cells for the 2D Taylor–Green vortex problem.

Table 2
L^1-norm of the error on the velocity and convergence rates for the 2D Taylor–Green vortex problem, $c_{\text{cfl}} = 0.1$.

	Q_1		Q_2		Q_3		Q_4	
h_0	L^1 error	rate	L^1 error	rate	L^1 error	rate	L^1 error	rate
1/4	4.08e−1		3.61e−2		1.87e−2		6.22e−3	
1/8	7.39e−2	2.46	6.13e−3	2.55	2.53e−3	2.88	2.35e−4	4.72
1/16	1.07e−2	2.78	1.54e−3	1.99	1.71e−4	3.88	8.47e−6	4.79
1/32	1.96e−3	2.44	3.99e−4	1.94	6.69e−6	4.67	7.14e−7	3.56
1/64	4.26e−4	2.20	1.00e−4	1.99	4.65e−7	3.84	4.89e−8	3.86
1/128	1.00e−4	2.09	2.42e−5	2.04	4.89e−8	3.24	3.23e−9	3.92

The 2D Taylor–Green vortex test case is designed so that the solution in Eulerian coordinates is smooth and time-independent. This is done by introducing a manufactured source of internal energy Q

$$\rho \frac{de}{dt} = -p \nabla \cdot \boldsymbol{u} + Q.$$

Using the following initial conditions and source

$$\begin{aligned} \rho_0(\boldsymbol{x}_0) &= 1, \qquad \boldsymbol{u}_0(\boldsymbol{x}_0) = (\sin(\pi x_0)\cos(\pi y_0) - \cos(\pi x_0)\sin(\pi y_0)), \\ p_0(\boldsymbol{x}_0) &= \frac{\rho}{4}(\cos(2\pi x_0) + \cos(2\pi y_0)) + 1, \qquad \gamma = \frac{5}{3}, \\ Q &= \frac{3\pi}{8}(\cos(3\pi x)\cos(\pi y) - \cos(\pi x)\cos(3\pi y)), \end{aligned}$$

the exact solution is $\rho(\boldsymbol{x}, t) = \rho_0(\boldsymbol{x})$, $\boldsymbol{u}(\boldsymbol{x}, t) = \boldsymbol{u}_0(\boldsymbol{x})$, $e(\boldsymbol{x}, t) = e_0(\boldsymbol{x})$ for all $\boldsymbol{x} \in \Omega(t)$.

For smooth solutions, the order of accuracy of the method depends on the polynomial degree of the approximation space and on how fast the artificial viscosity terms converge to zero. In order to minimize the consistency error in the entropy residual equation (43), we approximate the time derivative of the entropy by using BDF2. This, together with the h_3^2 scaling in (44) produces a consistency error of 4th order, i.e., the artificial viscosity is formally fourth order. One can use backward differencing with more points in (43) in order to achieve orders higher than 4.

We run the problem up to $t = 0.5$ with the boundary condition $\boldsymbol{u} \cdot \boldsymbol{n} = 0$. The initial and final meshes and the corresponding velocity magnitudes with a Q_4 finite element space are shown in Fig. 4. We compare a Q_1 and a Q_4 simulation with similar number of degrees of freedom in Fig. 5. Although it may not be clearly visible in the figure, the convergence rates reported in Table 2 show that the resolution of the Q_4 space is far superior to that of the Q_1 space. In Table 2 we show L^1-convergence rates for the velocity computed with finite element spaces of different

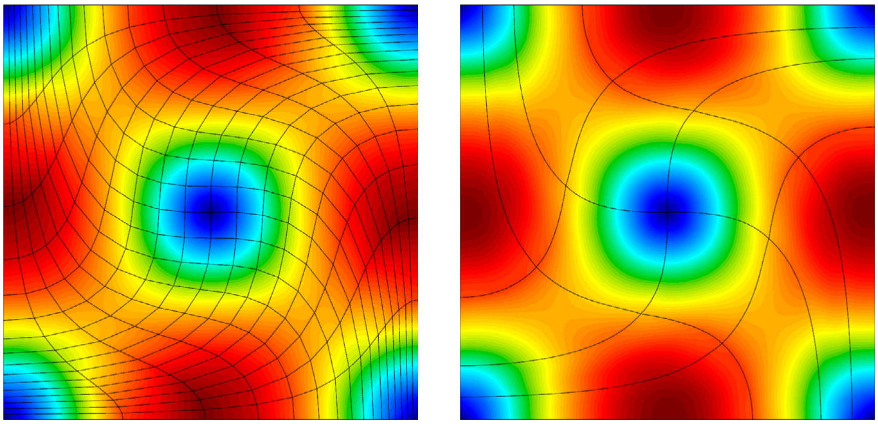


Fig. 5. Final mesh and velocity magnitude for a Q_1 simulation with 16×16 cells (left panel), and for a Q_4 simulation with 4×4 cells (right panel) for the 2D Taylor–Green vortex problem.

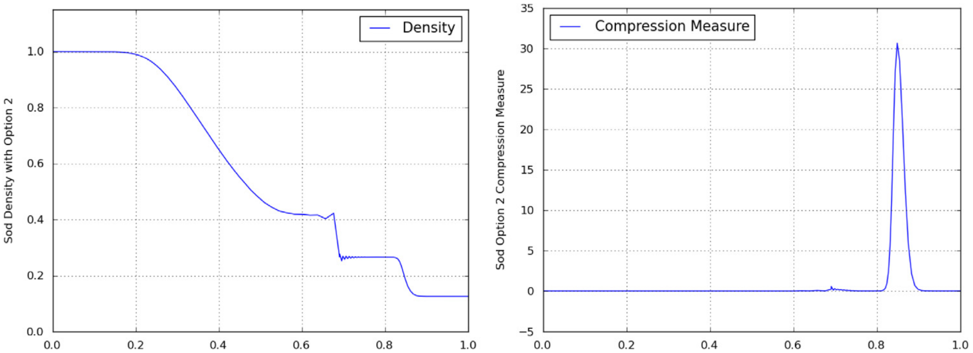


Fig. 6. 1D Sod tube problem using Option 2 for the viscosity coefficients. Density field (left panel) and compression measure (right panel).

order. The expected high-order convergence rates are observed. Note that the Q_4 approximation is limited to fourth order since we used BDF2 to evaluate the time derivative of the entropy in the entropy residual.

4.2. 1D Sod tube

The goal of this test is to demonstrate the capabilities of the method to capture shocks and contact discontinuities properly. This is a 1D Riemann problem developing a rarefaction, a contact and a shock wave. The initial computational domain is [0, 1]. The initial conditions are:

$$\rho(x_0) = \begin{cases} 1.0 & x_0 \leq 0.5, \\ 0.125 & \text{otherwise}, \end{cases}, \qquad \boldsymbol{u}(x_0) = 0, \qquad p(x_0) = \begin{cases} 1.0 & x_0 \leq 0.5, \\ 0.1 & \text{otherwise}, \end{cases}, \qquad \gamma = 1.4.$$

The final time is 0.2 and we enforce $\boldsymbol{u} \cdot \boldsymbol{n} = 0$ at the boundary.

We first discuss the effects of using the viscosity coefficients defined in Option 2, see (41). In Fig. 6 we show density field (on the left panel) and the magnitude of the compression measure $\mu(\boldsymbol{x})$ at the final time, see (41). The simulation is done with $\mathbb{Q}_1$ FE on a mesh composed of 128 cells. We observe that Option 2 provides sufficient dissipation in the shock region, but almost none in the contact. The reason for this is that the compression measure $\mu(\boldsymbol{x})$ is based only on the gradient of the velocity, and the velocity is almost constant in the region of the contact. The oscillations in density, pressure and energy around the contact cause some small velocity gradients, but the generated compression is too small to dampen those oscillations. Next we present in Fig. 7 results obtained by using Option 1 for the viscosity. On the left panel of the figure we show the density field computed with Q_1 and Q_4 FE spaces using the first-order viscosity only, i.e., $c_{\text{entr}} = \infty$ in (40). There are 256 and 1024 cells for the Q_4 and Q_1 spaces, respectively, so that the numbers of degrees of freedom are almost identical for the two spaces. We observe that for both the Q_1 and Q_4

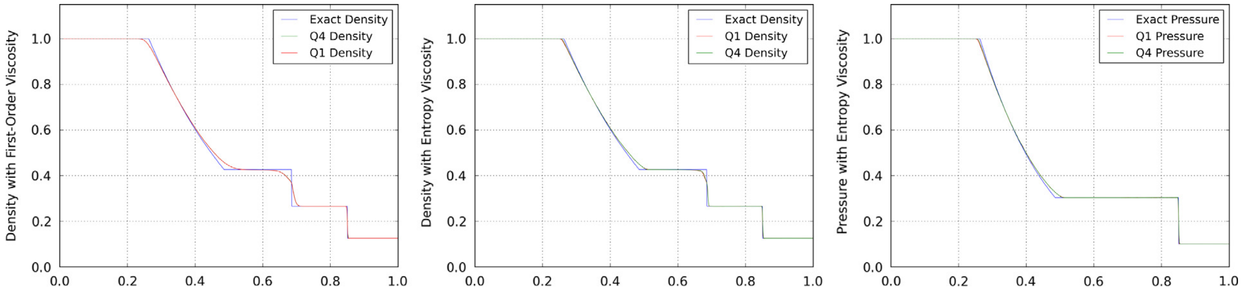


Fig. 7. Sod shocktube, Q_1 vs. Q_4 approximation, both with Option 1. Density field computed with the first-order viscosity (left panel). Density (center panel) and pressure (right panel) computed with the entropy–viscosity.

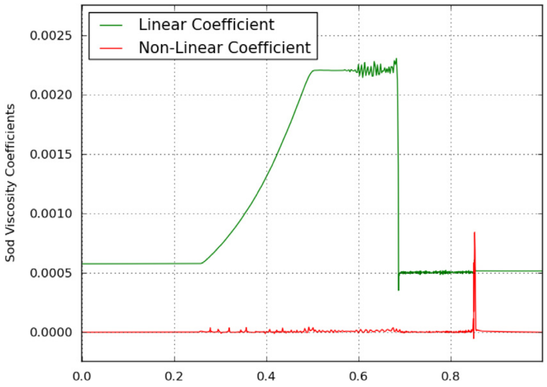


Fig. 8. 1D Sod tube problem. Comparison between the L^2-projections of the first-order viscosity and the high-order viscosity in Option 1.

Table 3
L^1-norm of error on density and convergence rates for the 1D Sod tube problem.

	Q_1			Q_2			Q_4	
h_0	L^1 error	rate	h_0	L^1error	rate	h_0	L^1error	rate
1/64	0.02194		1/32	0.02216		1/16	0.02190	
1/128	0.01319	0.73	1/64	0.01335	0.73	1/32	0.01264	0.79
1/256	0.00723	0.86	1/128	0.00747	0.83	1/64	0.00701	0.85
1/512	0.00382	0.92	1/256	0.00394	0.92	1/128	0.00383	0.87
1/1024	0.00199	0.94	1/512	0.00204	0.95	1/256	0.00207	0.88

approximations, the definition (40) provides enough diffusion throughout the domain. In the center and in the right panel of Fig. 7 we show the density and pressure fields obtained with the entropy–viscosity, see (45). The results are sharper than those generated by the first-order viscosity and there are no oscillations in the contact region, meaning that the entropy production provides sufficient dissipation. On the left figure, where we use the first-order viscosity, we observe that the results for the Q_1 and Q_4 FE spaces are essentially the same.

In Fig. 8 we compare the L^2-projections of the first order viscosity field (40), which was used to generate the results in the left panel of Fig. 7, and the entropy-based viscosity field (44) which was used to generate the results in the center and right panels of Fig. 7. We observe that in the smooth regions the entropy production coefficient is the active part of the minimum in (45), and the first order coefficient is the active part in the shock, thereby confirming the heuristic argument that leads to the notion of entropy-based viscosity.

We show in Table 3 the convergence rates in the L^1-norm for the density computed with Q_1, Q_2 and Q_4 spaces. The rows are aligned so that the rates in each row are computed with approximately the same number of degrees of freedom. We observe that the errors and rates for all the finite element spaces are similar, they are close to the optimal rate of 1.

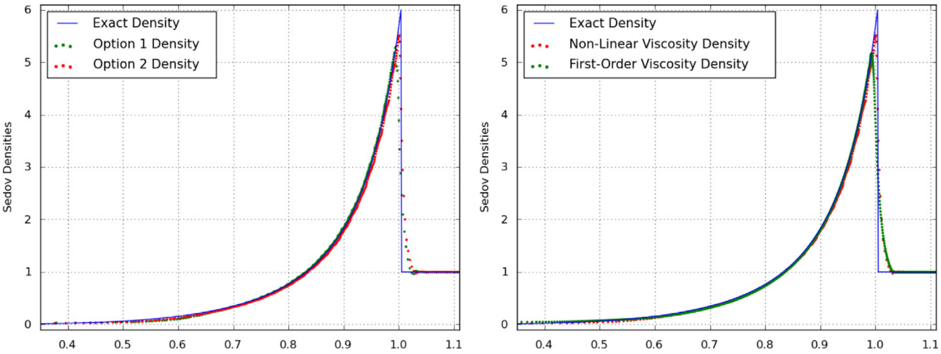


Fig. 9. 2D Sedov problem at $t = 1$. Density vs. radius for entropy–viscosity from Option 1 to Option 2 (left panel). Density vs. for radius for entropy–viscosity and first-order viscosity from Option 2 (right panel).

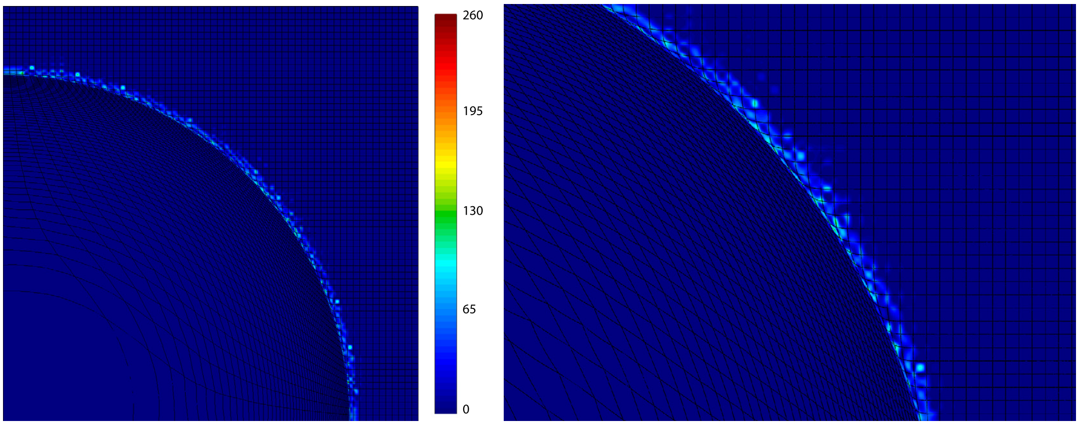


Fig. 10. 2D Sedov problem at $t = 1$. Mesh and L^2-projection of the entropy residual $|D|$ (left panel). Zoom at the shock region (right panel).

4.3. 2D Sedov explosion

The Sedov explosion, introduced in [46], is a standard problem to test the ability of codes to preserve the radial symmetry of shocks. The initial conditions are

$$\rho(\boldsymbol{x}_0) = 1, \qquad \boldsymbol{u}(\boldsymbol{x}_0) = 0, \qquad e(\boldsymbol{x}_0) = \delta_0, \qquad \int_{\Omega_0} e(\boldsymbol{x}_0)\,\mathrm{d}\boldsymbol{x}_0 = E_{\text{tot}}, \qquad \gamma = 1.4$$

where δ_0 is the Dirac measure supported at $\boldsymbol{x}_0 = 0$, and $E_{\text{tot}} = 1$ if the origin $\boldsymbol{x}_0 = 0$ is in the interior of Ω_0. The internal energy deposited at the origin converts into kinetic energy thereby creating an expanding shock wave. The problem is usually run up to $t = 1.0$.

First we discuss results obtained by using Q_2 FE spaces on a Cartesian uniform mesh for the sector $\Omega_0 = [0, 1.2] \times [0, 1.2]$ with 64×64 cells. Note that since one quarter of the Dirac measure is supported in Ω_0, the initial energy is $E_{\text{tot}} = \frac{1}{4}$ for this case. We enforce the symmetry condition $\boldsymbol{u} \cdot \boldsymbol{n} = 0$ on the left and bottom boundaries. In the left panel of Fig. 9 we compare the exact density and the approximate density fields obtained by using the Option 1, see (45), and Option 2, see (46), for the viscosity coefficients. This figure shows the nodal value of the density as a function of the radial distance to the origin for all the nodes in the mesh. This representation is meant to evaluate the radial symmetry of the approximate solutions. There is no clear difference between the two options. The results shown in the right panel compare the density obtained with the first order viscosity (41) and the entropy–viscosity from Option 2, (46). We see that the entropy-based viscosity (46) is sharper than the first-order one (the maximum is higher).

In Fig. 10 we show the L^2-projection of the entropy residual, i.e., $|D|$ in (44). Note that the residual $|D|$ focuses very well in the shock as expected.

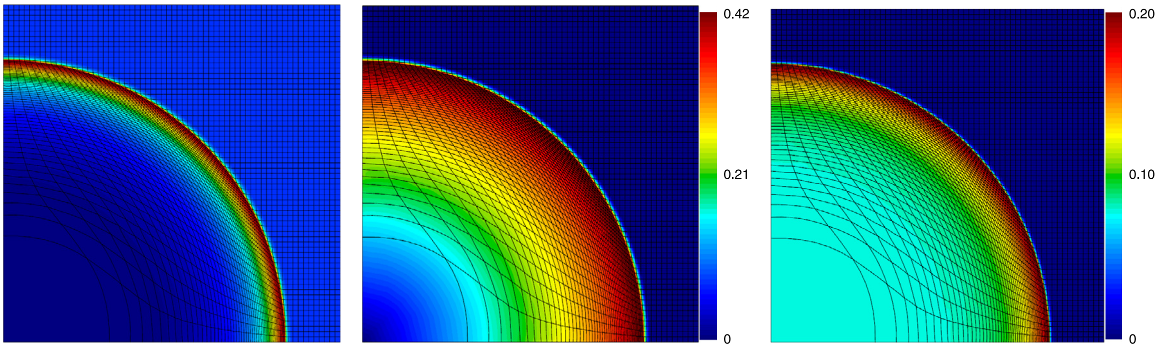


Fig. 11. 2D Sedov explosion problem. Entropy–viscosity solution at $t = 1$ using Option 2: density (left), velocity magnitude (center), and pressure (right).

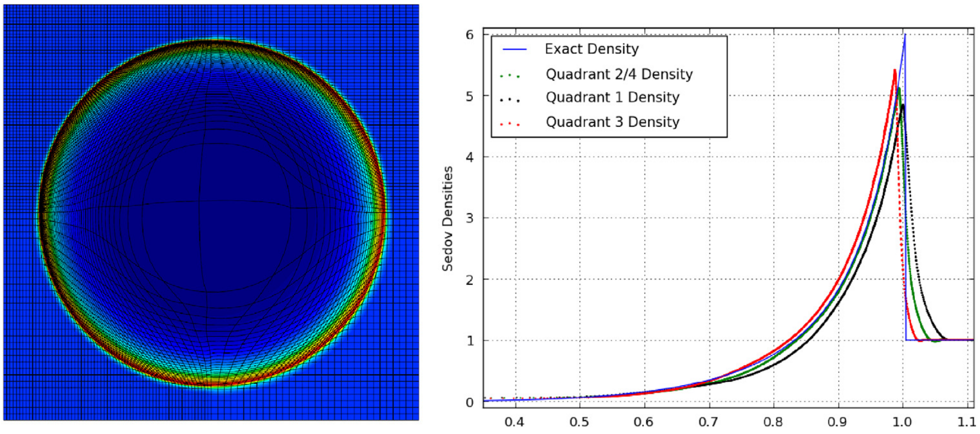


Fig. 12. 2D Sedov problem on a non-uniform mesh. Density and mesh at $t = 1$ (left panel), and density vs. radius along the lines $|x| = |y|$ (right panel).

In Fig. 11 we show the mesh, the density, the velocity magnitude and pressure obtained by the Option 2 viscosity (46). The shock is properly captured and the motion of the mesh is correct.

Next we consider a Cartesian non-uniform mesh for $\Omega_0 = [-1.2, 1.2] \times [-1.2, 1.2]$. In this case the initial energy is $E_{\text{tot}} = 1.0$ since the Dirac measure at the origin is entirely supported in Ω_0. We use 32×64 cells in quadrant #1, 64×64 cells in quadrant #2, 64×32 cells in quadrant #3, and 32×32 cells in quadrant #4. The purpose of this test is to evaluate the ability of the scheme to preserve the radial symmetry of the solution in the presence of non uniformities in the mesh. The computation is done up to $t = 1$ with Q_2 finite elements. We show in the left panel of Fig. 12 the density and the mesh at $t = 1$. We show in the right panel of the figure the distribution of the density as a function of the distance to the origin along the lines $|x| = |y|$ for each quadrant. The method performs satisfactorily since the radial symmetry is well preserved and the mesh motion is correct.

4.4. 2D Noh implosion

The Noh implosion, introduced in [47], is another benchmark problem tailored for testing the ability of numerical schemes to preserve symmetry. The initial conditions are

$$\rho(\boldsymbol{x}_0) = 1, \qquad \boldsymbol{u}(\boldsymbol{x}_0) = -\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_{\ell^2}}, \qquad e(\boldsymbol{x}_0) = 0, \qquad \gamma = \frac{5}{3}.$$

The initial velocity generates an outward traveling shock wave. This test is usually run up to $t = 0.6$.

First we discuss results obtained by using Q_2 FE spaces on a uniform Cartesian mesh for the quadrant $\Omega_0 = [0, 1.2] \times [0, 1.2]$. We enforce $\boldsymbol{u} \cdot \boldsymbol{n} = 0$ on the left and bottom boundaries of Ω_0. All the simulations reported

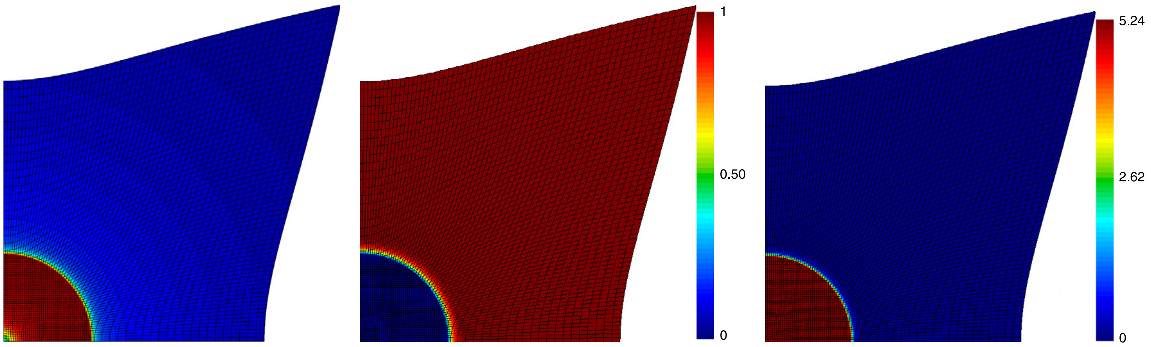


Fig. 13. 2D Noh problem at $t = 1$. Density (left panel), velocity magnitude (center panel), pressure (right panel) and Q_2 mesh with 64 × 64 cells.

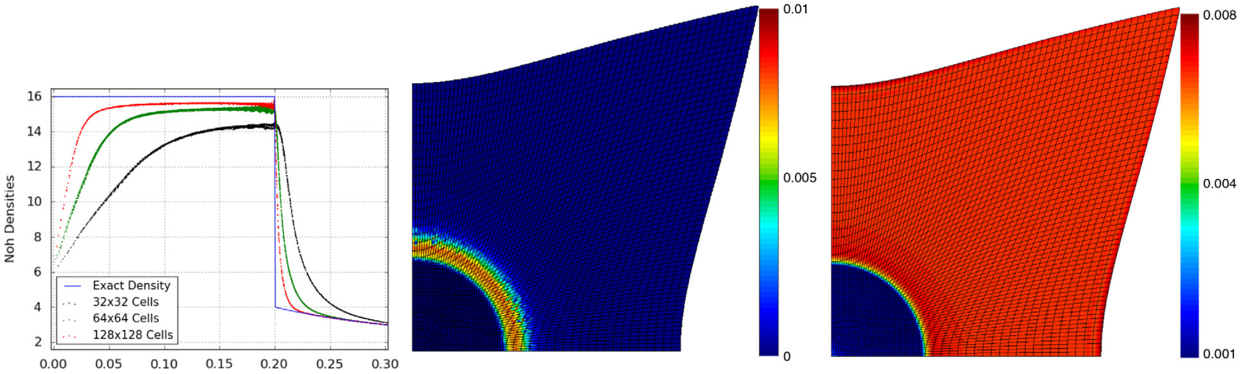


Fig. 14. 2D Noh problem at $t = 1$. Density vs. radius on different Q_2 meshes, 32 × 32, 64 × 64 and 128 × 128 (left panel). L^2-projections of the high-order viscosity (center panel) and first-order viscosity (right panel) from Option 1, 64 × 64 cells.

hereafter are done with the Option 1 of the viscosity (40). When using the same parameters for both options, Option 2 tangles the mesh and the simulation fails. It is possible to stabilize the computation with Option 2 by increasing c_{entr}, but in all the cases the results obtained with Option 1 are always less oscillatory and therefore the results from Option 2 are not reported. Fig. 13 shows the mesh, the density (left), the velocity magnitude (center) and the pressure (right) obtained at $t = 1$ by using the entropy–viscosity from Option 1 (45) on a mesh composed of 64 × 64 cells. The left panel of Fig. 14 shows the exact density and the densities computed on meshes composed of 32 × 32, 64 × 64 and 128 × 128 cells. The method is slightly oscillatory but convergent. In the two other panels are shown the L^2-projections of the high-order viscosity (center panel) and the first-order coefficient (right panel). Again we observe that the high-order viscosity is almost zero away from the shock region and it dominates the first order viscosity in the shock. This justifies the heuristic definition of the entropy–viscosity in (45) and (46).

Next we consider a non-uniform Cartesian mesh with $\Omega_0 = [-1.2, 1.2]\times[-1.2, 1.2]$. We use the Q_2 approximation and put 32 × 64 cells in quadrant #1, 64 × 64 cells in quadrant #2, 64 × 32 cells in quadrant #3, and 32 × 32 cells in quadrant #4. The purpose of this mesh is to test again the ability of the scheme to preserve radial symmetry for radial shocks. The left panel of Fig. 15 shows the density and the mesh at $t = 1$. The right panel of the figure shows the density as a function of the distance to the origin for the four quadrants. Some motion is observed in the middle of the domain. This problem is caused by the combination of wall heating, see [47], and the presence of cells of different size in the middle of the domain. These heterogeneities produce a nonuniform density in the different quadrants which causes the more dense material to push the less dense one. One way to limit this effect consists of smoothing the initial length scale h_0. The initial length scale that is used in the computation is shown in the left panel of Fig. 16. Although a large smoothing constant $\varepsilon = 30.0$ has been used in the smoothing procedure (36), the large gradient on h_0 in the middle of the domain causes mesh deformation. Using a uniform initial length scale is not appropriate either, since it would cause adding too much (or too little) artificial viscosity to one of the quadrants. The right panel of Fig. 16 shows the pressure field. The wall heating already mentioned above is clearly visible in the middle of the domain.

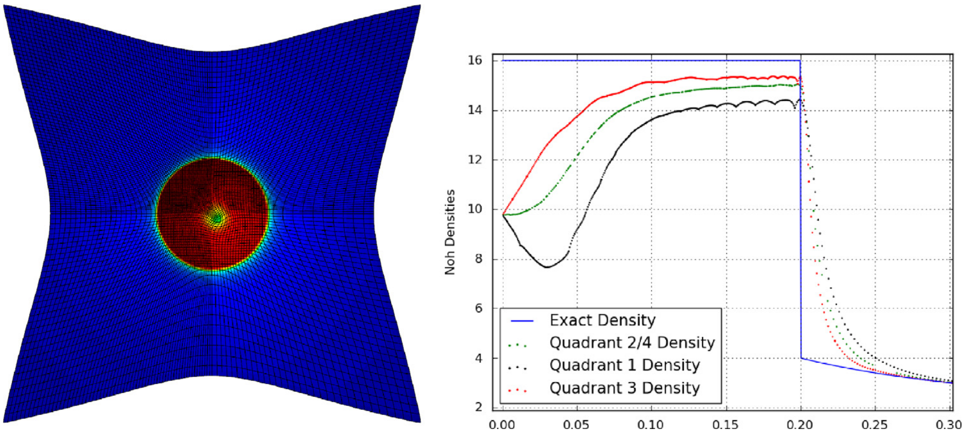


Fig. 15. 2D Noh problem on a non-uniform mesh at $t = 0.6$. Density and mesh (left panel), density vs. radius (right panel).

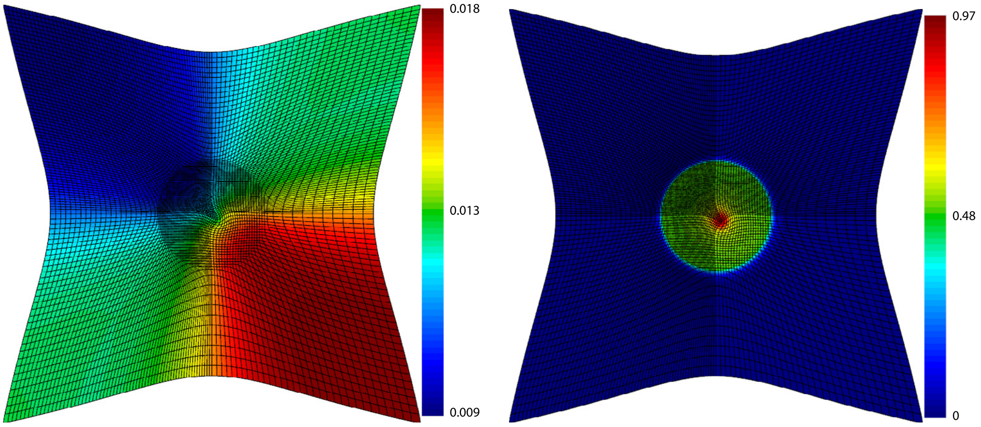


Fig. 16. 2D Noh problem on a non-uniform mesh at $t = 0.6$. Length scale field h_0 (left panel) and pressure (right panel).

4.5. 3D Noh implosion and parallel performance

Here we demonstrate the capabilities of the method to perform parallel 3D calculations. We consider the three-dimensional Noh problem with the initial conditions

$$\rho(\boldsymbol{x}_0) = 1, \qquad \boldsymbol{u}(\boldsymbol{x}_0) = -\frac{\boldsymbol{r}}{\|\boldsymbol{r}\|_{\ell^2}}, \qquad e(\boldsymbol{x}_0) = 0, \qquad \gamma = \frac{5}{3}.$$

Again we run this test up to $t = 0.6$ with the entropy–viscosity from Option 1.

Although the purpose of the present paper is not to detail the computational aspects of the method; we have programmed the algorithm in parallel and in three space dimensions to give some ideas on the performances in realistic situations. Communication between parallel tasks occurs for the following procedures:

- Computation of the time step (47).
- Computation of the global entropy production normalization constant used in the denominator of Eq. (44).
- Assembly of global matrices for density, velocity and specific internal energy at each Runge–Kutta sub-stage.
- Solving the mass matrix linear systems for density, velocity and specific internal energy, and redistribution of the new solution to the different MPI tasks at each Runge–Kutta sub-stage.

Figs. 17 and 18 show the density and the mesh using Q_2 approximation on a uniform Cartesian mesh composed of 32 cells in each space direction. The initial domain is $\Omega_0 = [-1.2, 1.2]^3$. The center panel of Fig. 18 shows the mesh division between 64 parallel tasks. We also present a strong scalability test for Q_2 (on 32^3 cells) and Q_4 (on 16^3 cells) simulations. We have performed 20 time steps for all the scalability simulations.

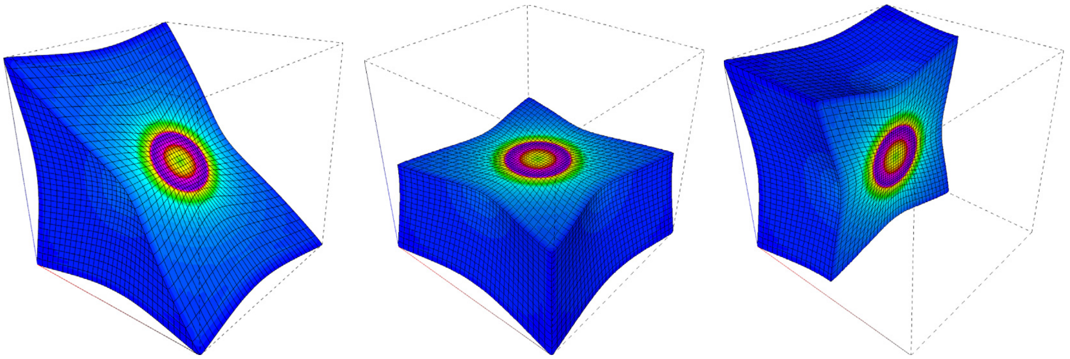


Fig. 17. Density cuts for the 3D Noh problem at $t = 0.6$.

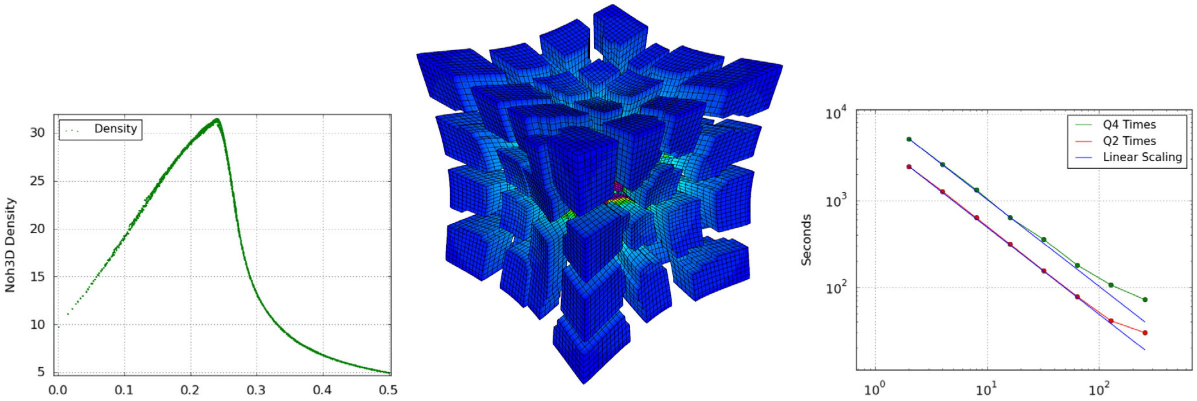


Fig. 18. 3D Noh problem at $t = 0.6$. Density vs. radius (left panel), 64 MPI tasks division (center panel), and strong scalability test from 1 to 256 processors (right panel).

5. Conclusion

We have presented a new high-order curvilinear finite element method for Lagrangian hydrodynamics that is non-oscillatory in contacts, has a good shock detection capability, is compatible at the continuous level with generalized entropy inequalities (15) and the minimum principle on the specific entropy (16). The artificial viscosity is also compatible with the requirements from [24]. The method is general with respect to the polynomial degree of the finite element approximation in space and the order of the time integration. The results reported confirm that the artificial viscosity converges to zero for smooth solutions and the convergence rate is optimal. The method achieves convergence with a rate close to one for standard shock wave problems, which is optimal. Proper radial symmetry preservation is observed on uniform and non-uniform meshes. The method has the ability to represent details of the flow within one single zone for high-order polynomial degrees. All these features come with the price of adding artificial viscosity in all the equations and using continuous finite element spaces for all variables. Being more diffusive, Option 1 is overall slightly more robust than Option 2. Our next objective is to replace the notion of artificial viscosity based on differential operators like $-\nabla \cdot (\nu \nabla)$ by a graph Laplacian approach outlined in http://arxiv.org/abs/1509.07461.

Appendix

A.1. *Orthogonal transformations invariance*

Here we show that the artificial viscosity is invariant under any orthogonal transformation of $\boldsymbol{x}$ and $\boldsymbol{u}$ as stated in Section 3.6.

Proposition 1. *The scalar diffusion in the mass equation (24) and the energy equation (26) are invariant by orthogonal transformation of $\boldsymbol{x}$ and $\boldsymbol{u}$. The viscous force in the momentum equation (25) is transformed the same way as $\boldsymbol{x}$ and $\boldsymbol{u}$.*

Proof. Since this argument is discrete, we introduce the notation:

$$\begin{aligned}
\rho(\boldsymbol{x},t) &= \sum_{j=1}^{N} \rho_j(t)\varphi_j(\boldsymbol{x},t), \qquad e(\boldsymbol{x},t) = \sum_{j=1}^{N} e_j(t)\varphi_j(\boldsymbol{x},t),\\
u_m(\boldsymbol{x},t) &= \sum_{j=1}^{N} U_{jm}(t)\varphi_j(\boldsymbol{x},t), \quad m = 1...d, \qquad \boldsymbol{U}_j = (U_{j1}...U_{jd})^T,\\
x_m(\boldsymbol{x}_0,t) &= \sum_{j=1}^{N} X_{jm}(t)\varphi_j(\boldsymbol{x}_0), \quad m = 1...d, \qquad \boldsymbol{X}_j = (X_{j1}...X_{jd})^T,
\end{aligned}$$

where the details of the mesh representation can be found in (32)–(35). Let A be unitary matrix and consider the orthogonal transformation

$$\boldsymbol{X}_j^* = A\boldsymbol{X}_j, \qquad \boldsymbol{U}_j^* = A\boldsymbol{U}_j, \quad \text{where } A^T = A^{-1}.$$

By convention all quantities used with upper index $*$ are defined in the transformed frame, i.e., are functions of $\boldsymbol{x}^*$. The Jacobian J^* of the transformation $\widehat{\boldsymbol{x}} \to \boldsymbol{x}^*$ can be expressed by the Jacobian of $\widehat{\boldsymbol{x}} \to \boldsymbol{x} \to \boldsymbol{x}^*$:

$$J^* = AJ, \qquad \det(J^*) = \det(J).$$

Then the gradient of the finite element shape functions are obtained as follows:

$$\nabla\varphi^* = (J^*)^{-T}\nabla\widehat{\varphi} = AJ^{-T}\nabla\widehat{\varphi}$$

and the gradient of $\boldsymbol{u}^*$ is

$$\nabla\boldsymbol{u}^* = \sum_{j=1}^{N} \boldsymbol{U}_j^* \otimes \nabla\varphi_j^* = \sum_{j=1}^{N} (A\boldsymbol{U}_j) \otimes (AJ^{-T}\nabla\widehat{\varphi}_j) = \sum_{j=1}^{N} A\boldsymbol{U}_j(\nabla\varphi_j)^T A^T = A\nabla\boldsymbol{u}A^T.$$

Let us examine the contribution of the artificial viscosity to a node j in the mass equation (28). After mapping everything to the reference frame and applying a quadrature rule with weights and nodes $(w_q, \widehat{\boldsymbol{x}}_q)_{1\le q\le Q}$, the contribution from an arbitrary cell K^* is computed by

$$-\sum_{q=1}^{Q}\left(w_q|\det(J_q^*)|\lambda_q^* \sum_{i=1}^{N} \rho_i(AJ_q^{-T}\nabla\widehat{\varphi}_{iq})\cdot(AJ_q^{-T}\nabla\widehat{\varphi}_{jq})\right),$$

where $\nabla\widehat{\varphi}_{iq} := \nabla\widehat{\varphi}_i(\widehat{\boldsymbol{x}}_q)$, $J_q := J(\widehat{\boldsymbol{x}}_q)$, $\lambda_q^* := \lambda^*(\boldsymbol{x}_q^*)$, etc. But this is the same as

$$-\sum_{q=1}^{Q}\left(w_q|\det(J_q)|\lambda_q^* \sum_{i=1}^{N} \rho_i(J_q^{-T}\nabla\widehat{\varphi}_{iq})\cdot(J^{-T}\nabla\widehat{\varphi}_{jq})\right),$$

which is the contribution from the corresponding cell K if $\lambda^*(\boldsymbol{x}_q^*) = \lambda(\boldsymbol{x}_q)$. Now we look at the artificial force term in the momentum equation (29). The contribution of the Cartesian component m from a cell K^* to a node j is

$$\begin{aligned}
&\sum_{q=1}^{Q} w_q|\det(J_q^*)|\Bigg(-\nu_q^*\widehat{\rho}_q \sum_{i=1}^{N} U_{im}^*(AJ_q^{-T}\nabla\widehat{\varphi}_{iq})\cdot(AJ_q^{-T}\nabla\widehat{\varphi}_{jq})\\
&\qquad + \lambda_q^*\widehat{\varphi}_{jq}\sum_{k=1}^{N}\sum_{i=1}^{N} U_{km}^*\rho_i(AJ_q^{-T}\nabla\varphi_{kq})\cdot(AJ^{-T}\nabla\varphi_{iq})\Bigg)\\
&\quad = \sum_{q=1}^{Q} w_q|\det(J_q)|\Bigg(-\nu_q^*\widehat{\rho}_q \sum_{i=1}^{N} A_m\cdot\boldsymbol{U}_i(J_q^{-T}\nabla\varphi_{iq})\cdot(J_q^{-T}\nabla\varphi_{jq})\\
&\qquad + \lambda_q^*\widehat{\varphi}_{jq}\sum_{k=1}^{N}\sum_{i=1}^{N} A_m\cdot\boldsymbol{U}_k\rho_i(J_q^{-T}\nabla\widehat{\varphi}_{kq})\cdot(J_q^{-T}\nabla\widehat{\varphi}_{iq})\Bigg),
\end{aligned}$$

where A_m is the mth row of A. Hence the artificial viscosity force is transformed the same way as $\boldsymbol{u}$ and $\boldsymbol{x}$ are if $\lambda^*(\boldsymbol{x}_q^*) = \lambda_q(\boldsymbol{x}_q)$ and $\nu^*(\boldsymbol{x}_q^*) = \nu(\boldsymbol{x}_q)$. The viscous terms in the internal energy equation (30) can be handled similarly and we reach the same conclusion. The only non trivial term there is $\|\nabla\boldsymbol{u}^*\|_{\ell^2}^2$; but we have $\nabla\boldsymbol{u}^* : \nabla\boldsymbol{u}^* = \nabla\boldsymbol{u} : \nabla\boldsymbol{u}$ since the Frobenius norm is invariant under orthogonal transformations. Hence the orthogonal transformation invariance holds if $\lambda_q^* = \lambda_q$ and $\nu_q^* = \nu_q$.

Now we consider the shock detectors. For the entropy production (42) we have $D^* = D$ since it depends only on scalars and $\nabla \cdot \boldsymbol{u}^*$:

$$\begin{aligned}\nabla \cdot \boldsymbol{u}^* &= \sum_{j=1}^{N} \nabla\varphi_j^* \cdot \boldsymbol{U}_j^* = \sum_{j=1}^{N} (AJ^{-T}\nabla\widehat{\varphi}_j) \cdot (A\boldsymbol{U}_j) \\ &= \sum_{j=1}^{N} (A\nabla\varphi_j) \cdot (A\boldsymbol{U}_j) = \sum_{j=1}^{N} \nabla\varphi_j \cdot \boldsymbol{U}_j = \nabla \cdot \boldsymbol{u}.\end{aligned}$$

The first-order viscosity of Option 1 in (40) is also invariant since $\|\boldsymbol{u}^*\|_{\ell^2} = \|A\boldsymbol{u}\|_{\ell^2} = \|\boldsymbol{u}\|_{\ell^2}$. For Option 2, see (41), we have $\nabla^s\boldsymbol{u}^* = A\nabla^s\boldsymbol{u}A^T$, hence $\nabla^s\boldsymbol{u}^*$ and $\nabla^s\boldsymbol{u}$ are similar matrices; thereby implying that $\mu^* = \mu$ and $\boldsymbol{s}^* = A\boldsymbol{s}$, where recall that $(\mu, \boldsymbol{s})$ is an eigenpair of $\nabla^s\boldsymbol{u}$. Hence the first-order viscosity is invariant under orthogonal transformation in both Option 1 and Option 2.

We now investigate how the length scales are transformed. It is clear that $h_0^* = h_0$ since lengths and angles are invariant by orthogonal transformation. Consider the Jacobian of the transformation $\boldsymbol{x}_0^* \to \boldsymbol{x}^*$, that is to say, the Jacobian of $\boldsymbol{x}_0^* \to \boldsymbol{x}_0 \to \boldsymbol{x} \to \boldsymbol{x}^*$, which is $AJJ_0^{-1}A^T$, then we have

$$\det(A) = 1 \Rightarrow \det(AJJ_0^{-1}A^T) = \det(JJ_0^{-1}) \Rightarrow h_3^* = h_3.$$

$$\frac{\|AJJ_0^{-1}A^T\boldsymbol{u}^*\|_{\ell^2}}{\|\boldsymbol{u}^*\|_{\ell^2}} = \frac{\|AJJ_0^{-1}\boldsymbol{u}\|_{\ell^2}}{\|A\boldsymbol{u}\|_{\ell^2}} = \frac{\|JJ_0^{-1}\boldsymbol{u}\|_{\ell^2}}{\|\boldsymbol{u}\|_{\ell^2}} \Rightarrow h_1^* = h_1.$$

$$\frac{\|AJJ_0^{-1}A^T\boldsymbol{s}^*\|_{\ell^2}}{\|\boldsymbol{s}^*\|_{\ell^2}} = \frac{\|AJJ_0^{-1}\boldsymbol{s}\|_{\ell^2}}{\|A\boldsymbol{s}\|_{\ell^2}} = \frac{\|JJ_0^{-1}\boldsymbol{s}\|_{\ell^2}}{\|\boldsymbol{s}\|_{\ell^2}} \Rightarrow h_2^* = h_2.$$

This implies that $\lambda^* = \lambda$ and $\nu^* = \nu$ for all the options.

A.2. Parameter list

In this section we list all the parameters that have been used in the computations reported in the paper. Are listed the viscosity options, the polynomial degree of the finite element space, the size of the initial mesh, the Runge–Kutta method, the CFL constant c_{cfl}, the viscosity constants c_{visc}, c_{entr}, and the final time t.

- 2D Taylor–Green problem, all figures ($c_{\text{cfl}} = 0.5$) and convergence tables ($c_{\text{cfl}} = 0.1$): Option 2 (with h_3 on $|D|$), RK4, $c_{\text{visc}} = 1.0$, $c_{\text{entr}} = 0.5$, $t = 0.5$.
- 1D Sod tube, Fig. 6: Option 2 (with h_3 on $|D|$), Q_1, 128 cells, RK4, $c_{\text{cfl}} = 0.2$, $c_{\text{visc}} = 1.0$, $t = 0.2$.
- 1D Sod tube, Table 3, Fig. 7, Fig. 8: Option 1 (with h_3 on $|D|$), 256 cells for Q_4, 1024 cells for Q_1, RK4, $c_{\text{cfl}} = 0.4$, $c_{\text{visc}} = 0.5$, $c_{\text{entr}} = 20.0$, $t = 0.2$.
- 1D Sod tube, Fig. 8: 1024 Q_1 cells with the same parameters as above.
- 2D Sedov explosion, Fig. 9, Fig. 11: Option 2 (with h_3 on $|D|$), Q_2, RK3, 64 × 64, $c_{\text{cfl}} = 0.2$, $c_{\text{visc}} = 2.0$, $c_{\text{entr}} = 5.0$, $t = 1.0$.
- 2D Sedov explosion, Fig. 12: Option 2 (with h_3 on $|D|$), Q_2, RK3, 64 × 64 cells in quadrant #2, 32 × 64 in #1, 64 × 32 in #3, 32 × 32 in #4, $c_{\text{cfl}} = 0.15$, $c_{\text{visc}} = 2.0$, $c_{\text{entr}} = 20.0$, $t = 1.0$, $\varepsilon = 1.0$ for h_0 smoothing.
- 2D Noh implosion, Fig. 13, Fig. 14: Option 1 (with h_1 on $|D|$), Q_2, RK4, 64 × 64, $c_{\text{cfl}} = 0.05$, $c_{\text{visc}} = 0.75$, $c_{\text{entr}} = 20.0$, $t = 0.6$.
- 2D Noh implosion, Fig. 15, Fig. 16: Option 1 (with h_1 on $|D|$), Q_2, RK4, 64 × 64 cells in quadrant #2, 32 × 64 in #1, 64 × 32 in #3, 32 × 32 in #4, $c_{\text{cfl}} = 0.05$, $c_{\text{visc}} = 0.75$, $c_{\text{entr}} = 20.0$, $t = 0.6$, $\varepsilon = 30.0$ for h_0 smoothing.
- 3D Noh implosion, Fig. 17, Fig. 18: Option 1 (with h_1 on $|D|$), Q_2/Q_4, $RK4$, 32 × 32 × 32/16 × 16 × 16, $c_{\text{cfl}} = 0.05$, $c_{\text{visc}} = 0.75$, $c_{\text{entr}} = 20.0$, $t = 0.6$.

References

[1] D.J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Comput. Methods Appl. Mech. Engrg. 99 (1992) 235–394.

[2] E. Caramana, D. Burton, M. Shashkov, The construction of compatible hydrodynamics algorithms utilizing conservation of total energy, J. Comput. Phys. 146 (1) (1998) 227–262.

[3] D.E. Burton, Multidimensional Discretizations of Conservation Laws for Unstructured Polyhedral Grids, Technical Report UCRL-JC-118306, Lawrence Livermore National Laboratory, 1994.

[4] D. Burton, Exact conservation of energy and momentum in staggered-grid hydrodynamics with arbitrary connectivity, in: Advances in the Free-Lagrange Method Including Contributions on Adaptive Gridding and the Smooth Particle Hydrodynamics Method, vol. 395, Springer, Berlin Heidelberg, 1991, pp. 7–19.

[5] M.L. Wilkins, Methods in Computational Physics, vol. 3, Academic Press, 1964.

[6] A. Claisse, B. Després, E. Labourasse, F. Ledoux, A new exceptional points method with application to cell-centered Lagrangian schemes and curved meshes, J. Comput. Phys. 231 (11) (2012) 4324–4354.

[7] G. Carré, S. Del Pino, B. Després, E. Labourasse, A cell-centered lagrangian hydrodynamics scheme on general unstructured meshes in arbitrary dimension, J. Comput. Phys. 228 (14) (2009) 5160–5183.

[8] P.-H. Maire, A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes, J. Comput. Phys. 228 (7) (2009) 2391–2425.

[9] P.-H. Maire, J. Breil, A second-order cell-centered lagrangian scheme for two-dimensional compressible flow problems, Internat. J. Numer. Methods Fluids 56 (8) (2008) 1417–1423.

[10] P.-H. Maire, R. Abgrall, J. Breil, J. Ovadia, A cell-centered Lagrangian scheme for two-dimensional compressible flow problems, SIAM J. Sci. Comput. 29 (4) (2007) 1781–1824.

[11] B. Després, C. Mazeran, Lagrangian gas dynamics in two dimensions and Lagrangian systems, Arch. Ration. Mech. Anal. 178 (3) (2005) 327–372.

[12] R. Loubère, P.-H. Maire, P. Váchal, 3d staggered lagrangian hydrodynamics scheme with cell-centered Riemann solver-based artificial viscosity, Internat. J. Numer. Methods Fluids 72 (1) (2013) 22–42.

[13] P.-H. Maire, R. Loubère, P. Vachal, Staggered Lagrangian discretization based on cell-centered Riemann solver and associated hydrodynamics scheme, Commun. Comput. Phys. 10 (2011) 940–978.

[14] V. Dobrev, T. Kolev, R. Rieben, High-order curvilinear finite element methods for Lagrangian hydrodynamics, SIAM J. Sci. Comput. 34 (5) (2012) B606–B641.

[15] A. Barlow, A compatible finite element multi-material ALE hydrodynamics algorithm, Internat. J. Numer. Methods Fluids 56 (2007) 953–964.

[16] G. Scovazzi, Stabilized shock hydrodynamics: II. Design and physical interpretation of the SUPG operator for Lagrangian computations, Comput. Methods Appl. Mech. Engrg. 196 (2007) 966–978.

[17] G. Scovazzi, E. Love, M. Shashkov, Multi-scale Lagrangian shock hydrodynamics on Q1/P0 finite elements: Theoretical framework and two-dimensional computations, Comput. Methods Appl. Mech. Engrg. 197 (2008) 1056–1079.

[18] F. Vilar, P.-H. Maire, R. Abgrall, A discontinuous Galerkin discretization for solving the two-dimensional gas dynamics equations written under total Lagrangian formulation on general unstructured grids, J. Comput. Phys. 276 (0) (2014) 188–234.

[19] Y. Bazilevs, I. Akkerman, D. Benson, G. Scovazzi, M. Shashkov, Isogeometric analysis of lagrangian hydrodynamics, J. Comput. Phys. 243 (0) (2013) 224–243.

[20] J. Von Neumann, R. Richtmyer, A method for the numerical calculation of hydrodynamic shocks, J. Appl. Phys. 21 (1950) 232–237.

[21] E. Caramana, M. Shashkov, P. Whalen, Formulations of artificial viscosity for multi-dimensional shock wave computations, J. Comput. Phys. 144 (1) (1998) 70–97.

[22] E.J. Caramana, R. Loubère, Curl-q: a vorticity damping artificial viscosity for essentially irrotational Lagrangian hydrodynamics calculations, J. Comput. Phys. 215 (2) (2006) 385–391.

[23] M. Shashkov, J. Campbell, A tensor artificial viscosity using a mimetic finite difference algorithm, J. Comput. Phys. 172 (2) (2001) 739–765.

[24] T. Kolev, R. Rieben, A tensor artificial viscosity using a finite element approach, J. Comput. Phys. 228 (22) (2009) 8336–8366.

[25] K. Lipnikov, M. Shashkov, A framework for developing a mimetic tensor artificial viscosity for Lagrangian hydrocodes on arbitrary polygonal meshes, J. Comput. Phys. 229 (20) (2010) 7911–7941.

[26] J.-L. Guermond, On the use of the notion of suitable weak solutions in CFD, Internat. J. Numer. Methods Fluids 57 (9) (2008) 1153–1170.

[27] J.-L. Guermond, R. Pasquetti, B. Popov, Entropy viscosity method for nonlinear conservation laws, J. Comput. Phys. 230 (11) (2011) 4248–4267.

[28] J.-L. Guermond, B. Popov, Viscous regularization of the Euler equations and entropy principles, SIAM J. Appl. Math. 74 (2) (2014) 284–305.

[29] A. Harten, P. Lax, D. Levermore, W. Morokoff, Convex entropies and hyperbolicity for general Euler equations, SIAM J. Numer. Anal. 35 (6) (1998) 2117–2127.

[30] E. Tadmor, A minimum entropy principle in the gas dynamics equations, Appl. Numer. Math. 2 (3–5) (1986) 211–219.

[31] P. Lax, Shock Waves and Entropy in Contributions to Nonlinear Functional Analysis, Academic Press, New York, NY, 1971.

[32] J.-L. Guermond, M. Nazarov, B. Popov, Implementation of the Entropy Viscosity Method, Technical Report 4015, KTH, Numerical Analysis, Sweden, Stockholm, 2011a. QC 20110720.

[33] V. Zingan, J.-L. Guermond, J. Morel, B. Popov, Implementation of the entropy viscosity method with the discontinuous Galerkin method, Comput. Methods Appl. Mech. Engrg. 253 (2013) 479–490.

[34] S. Krushkov, First order quasilinear equations with several independent variables, Math. USSR-Sb. 10 (1970) 217–243.

[35] P. Lax, Weak solutions of nonlinear hyperbolic equations and their numerical computation, Comm. Pure Appl. Math. 7 (1954) 159–193.

[36] X. Ding, G. Chen, P. Luo, Convergence of the Lax-Friedrichs scheme for isentropic gas dynamics I, Acta Math. Sci. 5 (1985) 415–432.

[37] R.J. DiPerna, Convergence of approximate solutions to conservation laws, Arch. Ration. Mech. Anal. 82 (1) (1983) 27–70.

[38] R.J. DiPerna, Convergence of the viscosity method for isentropic gas dynamics, Comm. Math. Phys. 91 (1) (1983) 1–30.

[39] S. Bianchini, A. Bressan, Vanishing viscosity solutions of nonlinear hyperbolic systems, Ann. of Math. (2) 161 (1) (2005) 223–342.

[40] A. Harten, On the symmetric form of systems of conservation laws with entropy, J. Comput. Phys. 49 (1) (1983) 151–164.

[41] H. Frid, Maps of convex sets and invariant regions for finite-difference systems of conservation laws, Arch. Ration. Mech. Anal. 160 (3) (2001) 245–269.

[42] J.-L. Guermond, B. Popov, Invariant domains and first-order continuous finite element approximation for hyperbolic systems, SIAM J. Numer. Anal. (2015).

[43] D. Serre, Systèmes de lois de conservation I: hyperbolicité, entropies, ondes de choc, Diderot Editeur, Paris, 1996.

[44] MFEM. Modular finite element methods library. URL http://mfem.googlecode.com.

[45] GLVis. OpenGL visualization tool. URL http://glvis.googlecode.com.

[46] L. Sedov, Similarity and Dimensional Methods in Mechanics, CRC Press, Boca Raton, FL, 1993.

[47] W. Noh, Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux, J. Comput. Phys. 72 (1) (1987) 78–120.