

A tensor artificial viscosity using a finite element approach [☆]

Tz.V. Kolev ^a, R.N. Rieben ^{b,*}

^a Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94551, United States

^b Weapons and Complex Integration, B-Division, Lawrence Livermore National Laboratory, Livermore, CA 94551, United States

ARTICLE INFO

Article history:

Received 26 January 2009

Received in revised form 5 August 2009

Accepted 13 August 2009

Available online 21 August 2009

Keywords:

Shock hydrodynamics
Lagrangian hydrodynamics
Numerical methods
Finite element methods

ABSTRACT

We derive a tensor artificial viscosity suitable for use in a 2D or 3D unstructured arbitrary Lagrangian–Eulerian (ALE) hydrodynamics code. This work is similar in nature to that of Campbell and Shashkov [1]; however, our approach is based on a finite element discretization that is fundamentally different from the mimetic finite difference framework. The finite element point of view leads to novel insights as well as improved numerical results. We begin with a generalized tensor version of the Von Neumann–Richtmyer artificial viscosity, then convert it to a variational formulation and apply a Galerkin discretization process using high order Gaussian quadrature to obtain a generalized nodal force term and corresponding zonal heating (or shock entropy) term. This technique is modular and is therefore suitable for coupling to a traditional staggered grid discretization of the momentum and energy conservation laws; however, we motivate the use of such finite element approaches for discretizing each term in the Euler equations. We review the key properties that any artificial viscosity must possess and use these to formulate specific constraints on the total artificial viscosity force term as well as the artificial viscosity coefficient. We also show, that under certain simplifying assumptions, the two-dimensional scheme from [1] can be viewed as an under-integrated version of our finite element method. This equivalence holds on general distorted quadrilateral grids. Finally, we present computational results on some standard shock hydro test problems, as well as some more challenging problems, indicating the advantages of the new approach with respect to symmetry preservation for shock wave propagation over general grids.

© 2009 Elsevier Inc. All rights reserved.

1. Introduction and motivation

This work is motivated by the desire to make the numerical algorithms which are used in solving the equations of shock hydrodynamics in a Lagrangian frame more robust with respect to grid motion. Our goal is to improve the Lagrangian hydrodynamics algorithms to prevent spurious grid distortions and, more importantly, to eliminate artificial symmetry breaking in problems with irregular/non-orthogonal grids. In a typical Lagrangian calculation such symmetry breaking and mesh distortion can come from a variety of sources, including: inaccuracies in the pressure gradient operator, mismatches in the spatial centering of acceleration and mass, hourglass modes and deficiencies in the artificial viscosity term. An example of this is given in Fig. 1, where we show a typical result obtained by running the canonical Noh test problem [2] on an initially Cartesian grid using a traditional staggered-grid hydro (SGH) method with a scalar artificial viscosity. The initial conditions in the Noh problem have radial symmetry, and this should be preserved for all time by the numerical solution. When the compu-

[☆] This work performed under the auspices of the US Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344, LLNL-JRNL-409868.

* Corresponding author. Tel.: +1 925 422 3783.

E-mail addresses: tzanio@llnl.gov (Tz.V. Kolev), rieben1@llnl.gov (R.N. Rieben).

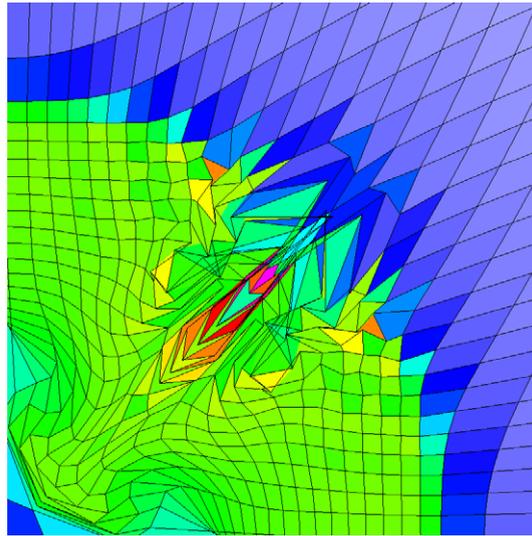


Fig. 1. Example of spurious grid distortion encountered when applying standard SGH methods to the Noh problem on an initially orthogonal mesh. There are multiple sources of this grid distortion including inaccuracies in the pressure gradient operator, discretization of the artificial viscosity term and hourglass instabilities. Each of these errors can amplify each other over time, leading to a rapid tangling of the grid.

tational grid is of the polar type (with equally distributed angles) and is aligned with the initial velocity conditions, then radial symmetry is preserved. However, when the mesh is not aligned with the shock flow as in Fig. 1, spurious vorticity and general grid distortion are generated.

Grid distortion of this type is usually blamed entirely on the so-called “hourglass” modes. The hourglass modes are very specific numerical instabilities caused by the inability of an SGH method to resolve the gradient of the highest frequency spatial mode of a given computational grid (the so-called “checkerboard” pressure mode). Stated another way, one can add an arbitrarily scaled checkerboard mode to the pressure field, as well as an arbitrarily scaled hourglass mode to the velocity field, and not affect the numerical solution. In the context of hyperbolic problems, the hourglass/checkerboard modes are triggered by point (or delta function) sources, an example of which is given in Fig. 2. If left unchecked, such modes can grow indefinitely in a time dependent problem leading to spurious grid distortion. To address this instability, most SGH codes introduce forces in the momentum equation which are designed to resist the offending modes or use more advanced

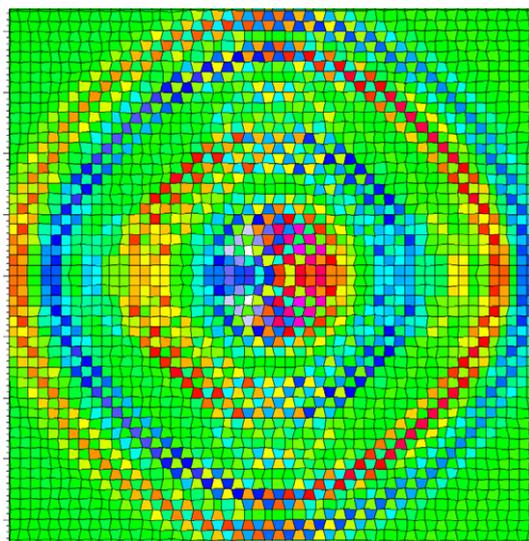


Fig. 2. Example of a “checkerboard” instability in the pressure field and corresponding “hourglass” instability in velocity field; excited by applying a time dependent perturbation to a single node at the center. Instability exists at highest spatial frequency of the underlying grid irrespective of mesh resolution and is a result of a fundamental instability in the discrete representation of velocity, pressure and the spatial differential operators which relate them.

techniques such as the sub-zonal pressure method of [3]. The fundamental cause of this instability lies in the discrete representation of the pressure and velocity fields and the spatial differential operators which relate them (i.e. Div and $Grad$). It is important to distinguish between the very specific hourglass/checkerboard instabilities as demonstrated in Fig. 2 and more general grid instabilities, such as those of Fig. 1. In this paper, we will not directly address the case of hourglass/checkerboard instabilities.

Based on our experience, the artificial viscosity term is usually the biggest culprit when it comes to spurious grid distortion and symmetry breaking for shock driven problems. Therefore, in this paper, we focus on improving the artificial viscosity term. Our approach is inspired by, and similar to, the work of Campbell and Shashkov [1]; however, our ultimate numerical formulation is fundamentally different. We use a finite element (FE) approach because it is general, well founded in theory, valid on unstructured grids and readily extendable to higher order methods. Our goal is to develop an artificial viscosity term that can be used in the context of traditional SGH methods which use node centered kinematic variables and zone centered thermodynamic variables. We show that with a particular choice of FE basis functions this is possible. However, it is important to note that the FE approach we present is general enough to use in non-traditional (possibly higher order) discretizations of the Euler equations, which we plan to investigate in a future publication.

The method of artificial viscosity, as originally introduced by Von Neumann and Richtmyer [4], is still the most popular method for treating shock waves in a Lagrangian description of hydrodynamics. Essentially, their approach for one-dimensional problems was to add a diffusion term with a special non-linear coefficient. Generalizing this idea to multiple dimensions has proved to be challenging over the years. The current state of the art is to use a Van Leer type “monotonic limiter” [5] in conjunction with a Von Neumann–Richtmyer type artificial viscosity along with an additional linear term due to Landshoff [6] to keep the artificial diffusion length of the shock front to a minimum while preventing spurious “ringing” (overshoots and undershoots). The application of such monotonicity limiters is closely related to the use of approximate Riemann solvers to track shock discontinuities [7]. For 1D calculations or cases where the computational grid is perfectly aligned with the flow, such methods give very good results. For general meshes which are not aligned with the flow (in 2D and 3D), such methods are known to cause symmetry breaking (e.g. the Noh problem as shown in Fig. 1). It is clear that the artificial viscosity formulation is the biggest (though not the only) culprit in breaking symmetry when shock waves pass over non-aligned grids. Based on numerical evidence, the use of monotonic limiters seems to amplify the symmetry breaking in such cases (see for example [8]). Furthermore, the artificial viscosity is typically implemented as a scalar term (with units of pressure) which is added to the bulk pressure in the momentum and energy conservation equations. This approach is valid only for the special case of non-vortical flows (i.e. it ignores a shear term). The more general case requires a tensor viscosity. Stated another way, the artificial viscous force term involves the $Div(Grad)$ operator acting on the velocity field.

To resolve the deficiencies of using a simple scalar viscosity, the authors in [1,9] show how to discretize the $Div(Grad)$ operator using a mimetic finite difference (MFD) approach and turn this into a tensor artificial viscosity term. We show how to discretize the same operator using a generic finite element approach in 2D and 3D. We show that for a specific choice of finite element basis functions and quadrature rule, we can exactly recover the method of [1]. Furthermore, we show that by using higher order quadrature, a more accurate method can be derived. The FE approach supports further improvements in the treatment of the non-linear viscosity coefficient, which leads to improvements in symmetry preservation for general shock flow problems as well as a novel way to implement an analog to the so called “limiter” function (smoothness sensor) to eliminate the effects of the artificial viscosity for “smooth” flow. Finally, we emphasize again that the general FE approach is valid not just for the standard SGH choices (i.e. node centered kinematic variables and zone centered thermodynamics), but is readily extendable to higher order methods.

The remainder of this paper is organized as follows: in Section 2, we list the notation used for continuous and discrete variables. Then, in Section 3, we motivate our choice for the viscous force and energy terms in the Euler equations. The variational formulation and finite element discretization are described in Sections 4 and 5 respectively. In Section 6, we show that we can recover the method of [1] if we use a lower-order approximation of the bi-linear finite element stiffness matrix. Several options for the nonlinear viscosity coefficient are discussed and motivated in Section 7. Finally, an extensive set of numerical examples on model Lagrangian hydro problems is presented in Section 8.

2. Notational conventions

Before we begin our discussion of the Euler equations, we introduce the notational conventions that will be used throughout this document. For *continuous* fields, we will adopt the conventions as outlined in Table 1. All vector fields will be designated with an arrow, with individual spatial components denoted with subscripts x , y and z (for 3D).

As we introduce the FE discretization process and the notion of *discrete* fields, we will use the notation conventions as outlined in Table 2. For this case, we will use **bold faced** letters to denote “arrays” (i.e. matrices and vectors) where capital letters will denote matrices and lower case letters will denote vectors. Subscripts will be reserved for indexing spatial quantities. We will adopt a notation similar to that of [10,11] for describing “nodal” values (subscript p) and “zonal” values (subscript “ Z ”). We also introduce the notion of a discrete basis for scalar fields which we will denote as ϕ_i . The subscript in this case will be a “degree of freedom” (DOF) index which spans the number of unique degrees of freedom in the basis function expansion.

Table 1

Notational conventions used for continuous variables.

Notation/symbol	Description
$\vec{x} = \{x, y, z\}$	Lagrangian coordinate
$\vec{v} = \{v_x, v_y, v_z\}$	Velocity vector field
$\vec{f} = \{f_x, f_y, f_z\}$	Force vector field
ρ	Density (mass/volume)
m	Mass
V	Volume
e	Internal energy per mass
P	Scalar pressure
μ	Viscosity coefficient
Ω	Spatial domain
$\partial\Omega$	Boundary of spatial domain
\vec{n}	A unit vector normal to the surface $\partial\Omega$

Table 2

Notational conventions used for discrete variables.

Notation/symbol	Description
$-z$	A subscript denoting a particular “zonal” value
$-p$	A subscript denoting a particular point (a.k.a “nodal”) value
$-i_j$	Subscripts denoting “degree of freedom” indexes for inner products
$\hat{\cdot}$	An accent denoting definition with respect to a reference coordinate system
Ω_Z	A discrete volume of space (“zone”)
\vec{x}_i	Coordinates of the vertices of the zone
ϕ_i	A discrete basis for scalar fields
\vec{v}_p	Velocity for node p
\vec{f}_p	Force due to artificial viscosity acting on node p
\vec{v}_Z	An array containing the nodal velocities for zone Z
\vec{f}_Z	A “corner force” array containing the nodal artificial viscosity force contributions from zone Z
\mathbf{J}_Z	The Jacobian matrix for zone Z
\mathbf{S}_Z	The “stiffness” matrix for zone Z
μ_Z	The non-linear artificial viscosity coefficient for zone Z
c_Z	The sound speed for zone Z
ℓ_Z	A measure of the length scale of zone Z
C_Z	A measure of the compressibility of zone Z
\mathcal{V}_Z	A measure of the vorticity of zone Z

3. Generalized viscous force and energy terms

Consider the Euler equations in a co-moving Lagrangian reference frame [12]. We begin by adding a generalized viscous force term to the momentum conservation equation and a corresponding energy term to the energy conservation equation, to obtain

$$\rho \frac{\partial \vec{v}}{\partial t} = -\vec{\nabla} P + \vec{\nabla} \cdot (\mu \vec{\nabla} \vec{v}), \quad (1)$$

$$\rho \frac{\partial e}{\partial t} = -P \vec{\nabla} \cdot \vec{v} + (\mu \vec{\nabla} \vec{v}) : \vec{\nabla} \vec{v}, \quad (2)$$

where μ is a generalized tensor coefficient which determines the artificial diffusion “length scale” set by the computational grid and $:$ denotes tensor contraction. Note that the sign of the artificial viscous force term in the momentum equation is opposite that of the pressure gradient force to ensure that it resists hydrodynamic motion. Note also that the viscous force is computed as the divergence of the stress-like tensor quantity $\mu \vec{\nabla} \vec{v}$; i.e. we have made use of the second order spatial differential operator $Div(Grad)$.

Let us assume that μ is given by a piece-wise constant function with respect to a set of discrete “zones” which decompose the computational domain. The artificial viscous force term is a vector field, and on each zone we can apply a Helmholtz decomposition and reduce the $Div(Grad)$ operation to obtain

$$\vec{f} \equiv \vec{\nabla} \cdot (\mu \vec{\nabla} \vec{v}) = \vec{\nabla} (\mu \vec{\nabla} \cdot \vec{v}) - \vec{\nabla} \times (\mu \vec{\nabla} \times \vec{v}). \quad (3)$$

In other words, the $Div(Grad)$ operator can be decomposed into a combination of the $Grad(Div)$ and $Curl(Curl)$ operators. If we ignore the $Curl(Curl)$ term (an assumption which is only valid when $\vec{\nabla} \times \vec{v} = 0$, i.e. zero vorticity), and define the following scalar value

$$q \equiv \mu \vec{\nabla} \cdot \vec{v},$$

then we can rewrite the generalized momentum and energy equations of (1) and (2) as

$$\rho \frac{\partial \vec{v}}{\partial t} = -\vec{\nabla}(P + q), \quad (4)$$

$$\rho \frac{\partial e}{\partial t} = -(P + q) \vec{\nabla} \cdot \vec{v}. \quad (5)$$

Note that this is precisely the form of the equations used in traditional staggered grid hydro (SGH) codes which make use of a scalar artificial viscosity [4,13].

4. Variational formulation

Rather than ignoring the $\text{Curl}(\text{Curl})$ term, our goal is to directly discretize the full tensor form of the artificial viscosity term in the momentum (1) and energy (2) conservation equations. To do so, we apply a variational formulation to the momentum equation by multiplying both sides of the equation by a vector valued “test” function and integrate over the problem domain Ω . If we ignore the pressure gradient term (for the sake of clarity), we have the following variational form

$$\int_{\Omega} \left(\rho \frac{\partial \vec{v}}{\partial t} \right) \cdot \{ \phi_x, \phi_y, \phi_z \} = \int_{\Omega} (\vec{\nabla} \cdot (\mu \vec{\nabla} \vec{v})) \cdot \{ \phi_x, \phi_y, \phi_z \}. \quad (6)$$

Each component of the vector $\{ \phi_x, \phi_y, \phi_z \}$ above is an arbitrary finite element “test” function. Next, we can split (6) in three independent equations, and perform integration by parts on the right-hand side using the Gauss divergence theorem to obtain

$$\int_{\Omega} \rho \frac{\partial \vec{v}}{\partial t} \cdot \{ \phi_x, 0, 0 \} = - \int_{\Omega} \mu \vec{\nabla} v_x \cdot \vec{\nabla} \phi_x + \int_{\partial \Omega} (\mu \vec{\nabla} v_x \phi_x) \cdot \vec{n}, \quad (7)$$

$$\int_{\Omega} \rho \frac{\partial \vec{v}}{\partial t} \cdot \{ 0, \phi_y, 0 \} = - \int_{\Omega} \mu \vec{\nabla} v_y \cdot \vec{\nabla} \phi_y + \int_{\partial \Omega} (\mu \vec{\nabla} v_y \phi_y) \cdot \vec{n}, \quad (8)$$

$$\int_{\Omega} \rho \frac{\partial \vec{v}}{\partial t} \cdot \{ 0, 0, \phi_z \} = - \int_{\Omega} \mu \vec{\nabla} v_z \cdot \vec{\nabla} \phi_z + \int_{\partial \Omega} (\mu \vec{\nabla} v_z \phi_z) \cdot \vec{n}, \quad (9)$$

where \vec{n} is the outward pointing unit normal vector of the surface $\partial \Omega$. The Div operator has now been transferred to a Grad operator acting on the scalar components of the vector test function. We can simplify things further, by ignoring the above boundary integrals. This is justified for the majority of the problems we consider in this paper and is discussed in more detail in Remark 1, Section 5.

Applying a Galerkin process, we define a basis $\{ \phi_i \}$ for the test function space and expand the velocity field in this basis. Specifically, we have the following expansions for each component

$$v_x(x, y, z) \approx \sum_i \alpha_i \phi_i(x, y, z), \quad (10)$$

$$v_y(x, y, z) \approx \sum_i \beta_i \phi_i(x, y, z), \quad (11)$$

$$v_z(x, y, z) \approx \sum_i \gamma_i \phi_i(x, y, z), \quad (12)$$

where the coefficients of expansion, α_i , β_i and γ_i , are independent of the spatial coordinates. Inserting these expressions into the variational form, dropping the boundary term, and testing with the basis functions yields

$$\sum_i \frac{\partial \alpha_i}{\partial t} \int_{\Omega} \rho \phi_i \phi = - \sum_i \alpha_i \int_{\Omega} \mu \vec{\nabla} \phi_i \cdot \vec{\nabla} \phi, \quad (13)$$

$$\sum_i \frac{\partial \beta_i}{\partial t} \int_{\Omega} \rho \phi_i \phi = - \sum_i \beta_i \int_{\Omega} \mu \vec{\nabla} \phi_i \cdot \vec{\nabla} \phi, \quad (14)$$

$$\sum_i \frac{\partial \gamma_i}{\partial t} \int_{\Omega} \rho \phi_i \phi = - \sum_i \gamma_i \int_{\Omega} \mu \vec{\nabla} \phi_i \cdot \vec{\nabla} \phi. \quad (15)$$

We now have a suitable starting point for a finite element discretization of the momentum equation. However, in this paper, we are only concerned with discretizing the force term of (3). Therefore, if we know the coefficients α_i , β_i and γ_i for the velocity field component expansions, then the generalized viscous force term can be approximated in variational form as

$$f_x \approx - \sum_i \alpha_i \int_{\Omega} \mu \vec{\nabla} \phi_i \cdot \vec{\nabla} \phi, \tag{16}$$

$$f_y \approx - \sum_i \beta_i \int_{\Omega} \mu \vec{\nabla} \phi_i \cdot \vec{\nabla} \phi, \tag{17}$$

$$f_z \approx - \sum_i \gamma_i \int_{\Omega} \mu \vec{\nabla} \phi_i \cdot \vec{\nabla} \phi. \tag{18}$$

5. Finite element discretization

Up to this point, we have not specified the basis we use for our velocity expansion. The finite element framework is very general and permits several choices for this basis. For example, we could use high order polynomial basis functions such as quadratic or cubic to achieve high levels of accuracy. Furthermore, we have not made any explicit assumptions about the element (or zone) topology we will use (e.g. tetrahedral or hexahedral elements). To remain compatible with a traditional staggered grid hydrodynamics code (i.e. node centered velocities), we now make the explicit choice of a tri-linear (bi-linear in 2D) basis. In a typical 3D staggered grid discretization, the x -, y - and z -components of the velocity field are known at the 8 vertices of a hexahedral zone. These values are now understood to be the values of the velocity coefficients α_i , β_i and γ_i from (10)–(12). Implicit in this choice is the notion of a tri-linear representation of velocity within a given hexahedral zone.

We choose to define this basis with respect to a standard reference frame. All hexahedral zones (including zones with curved boundaries) in a physical mesh are topologically equivalent to a reference hexahedral zone. In order to make integration over the reference zone as simple as possible, we adopt a standard Cartesian coordinate system with an origin at the point (0,0,0) as our reference coordinate system. Throughout the remainder of this paper, all objects explicitly defined with respect to this reference coordinate system will be accented with a *hat* symbol. Let $\hat{\Omega}_z$ denote the unit hexahedron such that

$$\hat{\Omega}_z \equiv \{(\hat{x}, \hat{y}, \hat{z}); 0 \leq \hat{x}, \hat{y}, \hat{z} \leq 1\}. \tag{19}$$

The tri-linear basis functions defined with respect to this reference hexahedron are listed in Table 3. The variational approximation of the generalized viscous force term in (16)–(18) requires the gradients of the basis functions. These are given in Table 4.

There exists a mapping Φ from the reference zone $\hat{\Omega}_z$ to an actual mesh zone Ω_z . This mapping is functionally written as

$$\vec{x} = \Phi(\vec{\hat{x}}). \tag{20}$$

This mapping is referred to as the local-to-global (a.k.a parametric) mapping and is defined by the geometry of the actual mesh zone. For example, if we restrict ourselves to the specific case of a hexahedron consisting of 8 vertices connected by straight lines, then this is equivalent to a tri-linear parametric mapping. For the purposes of this work, this is exactly the mapping we will assume (since this is precisely the type of hexahedral zone that is permitted in most SGH codes); however, we point out that higher order mappings exist which permit curved surfaces (such as those used in [14]), and the formulation we present is general enough to support such elements. For the specific case of a tri-linear parametric map, we have

$$\Phi(\vec{\hat{x}}) = \sum_{i=1}^8 \vec{x}_i \hat{\phi}_i(\vec{\hat{x}}), \tag{21}$$

where \vec{x}_i denotes the Lagrangian coordinate of vertex i and $\hat{\phi}_i$ are the tri-linear basis functions defined in Table 3. A schematic depiction of this mapping, along with our particular labeling convention for the vertices, is given in Fig. 3. The fact that our parametric mapping Φ makes use of the same basis as our velocity expansion is purely a coincidence; for general FE discretizations, this is not necessarily true.

Based on (20), we define basis functions on Ω_z by transferring the reference basis functions with Φ :

Table 3
Tri-linear basis functions on a reference hexahedron.

Basis function	Reference value
$\hat{\phi}_1$	$(1 - \hat{x})(1 - \hat{y})(1 - \hat{z})$
$\hat{\phi}_2$	$\hat{x}(1 - \hat{y})(1 - \hat{z})$
$\hat{\phi}_3$	$\hat{x}\hat{y}(1 - \hat{z})$
$\hat{\phi}_4$	$(1 - \hat{x})\hat{y}(1 - \hat{z})$
$\hat{\phi}_5$	$(1 - \hat{x})(1 - \hat{y})\hat{z}$
$\hat{\phi}_6$	$\hat{x}(1 - \hat{y})\hat{z}$
$\hat{\phi}_7$	$\hat{x}\hat{y}\hat{z}$
$\hat{\phi}_8$	$(1 - \hat{x})\hat{y}\hat{z}$

Table 4
Gradient of tri-linear basis functions on a reference hexahedron.

Basis gradient	x-Component	y-Component	z-Component
$\vec{\nabla} \hat{\phi}_1$	$(\hat{y} - 1)(1 - \hat{z})$	$(\hat{x} - 1)(1 - \hat{z})$	$(\hat{x} - 1)(1 - \hat{y})$
$\vec{\nabla} \hat{\phi}_2$	$(1 - \hat{y})(1 - \hat{z})$	$-\hat{x}(1 - \hat{z})$	$-\hat{x}(1 - \hat{y})$
$\vec{\nabla} \hat{\phi}_3$	$\hat{y}(1 - \hat{z})$	$\hat{x}(1 - \hat{z})$	$-\hat{x}\hat{y}$
$\vec{\nabla} \hat{\phi}_4$	$-\hat{y}(1 - \hat{z})$	$(1 - \hat{x})(1 - \hat{z})$	$(\hat{x} - 1)\hat{y}$
$\vec{\nabla} \hat{\phi}_5$	$(\hat{y} - 1)\hat{z}$	$(\hat{x} - 1)\hat{z}$	$(1 - \hat{x})(1 - \hat{y})$
$\vec{\nabla} \hat{\phi}_6$	$(1 - \hat{y})\hat{z}$	$-\hat{x}\hat{z}$	$\hat{x}(1 - \hat{y})$
$\vec{\nabla} \hat{\phi}_7$	$\hat{y}\hat{z}$	$\hat{x}\hat{z}$	$\hat{x}\hat{y}$
$\vec{\nabla} \hat{\phi}_8$	$-\hat{y}\hat{z}$	$(1 - \hat{x})\hat{z}$	$(1 - \hat{x})\hat{y}$

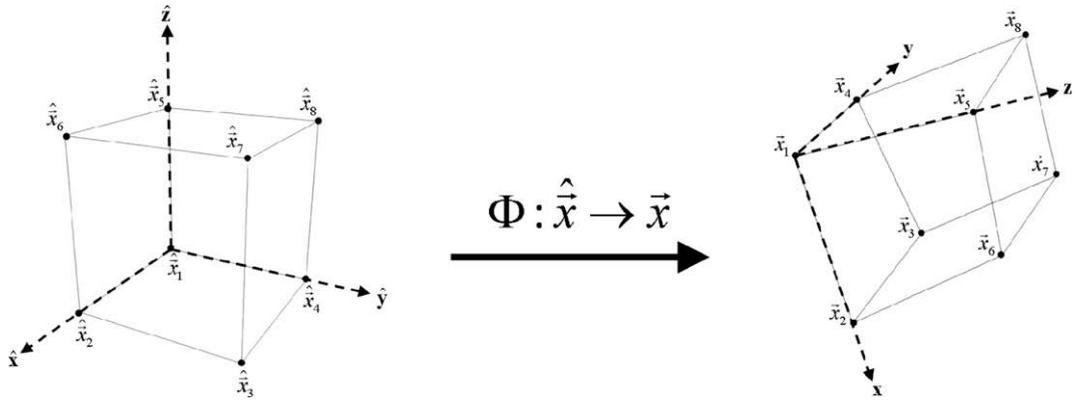


Fig. 3. Schematic depiction and vertex labeling convention for the parametric transformation process that maps a point from the reference zone to an actual mesh zone using tri-linear interpolation.

$$\phi_i(\vec{x}) = \hat{\phi}_i(\Phi^{-1}(\vec{x})). \quad (22)$$

Note that on distorted grids Φ^{-1} , and therefore $\{\hat{\phi}_i\}$, are no longer polynomial. Nevertheless, linear functions can always be represented exactly by expansion in the above basis. We define the Jacobian matrix (or metric tensor) for the parametric mapping as

$$(\mathbf{J}_Z)_{ij} = \frac{\partial x_j}{\partial \hat{x}_i}. \quad (23)$$

For the specific case of a tri-linear parametric map in 3D, the Jacobian matrix has the form

$$\mathbf{J}_Z = \begin{pmatrix} J_{1,1} & J_{1,2} & J_{1,3} \\ J_{2,1} & J_{2,2} & J_{2,3} \\ J_{3,1} & J_{3,2} & J_{3,3} \end{pmatrix}, \quad (24)$$

where

$$\begin{aligned} J_{1,1} &= (x_2 - x_1)(1 - \hat{y})(1 - \hat{z}) + (x_3 - x_4)\hat{y}(1 - \hat{z}) + (x_6 - x_5)(1 - \hat{y})\hat{z} + (x_7 - x_8)\hat{y}\hat{z}, \\ J_{1,2} &= (y_2 - y_1)(1 - \hat{y})(1 - \hat{z}) + (y_3 - y_4)\hat{y}(1 - \hat{z}) + (y_6 - y_5)(1 - \hat{y})\hat{z} + (y_7 - y_8)\hat{y}\hat{z}, \\ J_{1,3} &= (z_2 - z_1)(1 - \hat{y})(1 - \hat{z}) + (z_3 - z_4)\hat{y}(1 - \hat{z}) + (z_6 - z_5)(1 - \hat{y})\hat{z} + (z_7 - z_8)\hat{y}\hat{z}, \\ J_{2,1} &= (x_4 - x_1)(1 - \hat{x})(1 - \hat{z}) + (x_3 - x_2)\hat{x}(1 - \hat{z}) + (x_8 - x_5)(1 - \hat{x})\hat{z} + (x_7 - x_6)\hat{x}\hat{z}, \\ J_{2,2} &= (y_4 - y_1)(1 - \hat{x})(1 - \hat{z}) + (y_3 - y_2)\hat{x}(1 - \hat{z}) + (y_8 - y_5)(1 - \hat{x})\hat{z} + (y_7 - y_6)\hat{x}\hat{z}, \\ J_{2,3} &= (z_4 - z_1)(1 - \hat{x})(1 - \hat{z}) + (z_3 - z_2)\hat{x}(1 - \hat{z}) + (z_8 - z_5)(1 - \hat{x})\hat{z} + (z_7 - z_6)\hat{x}\hat{z}, \\ J_{3,1} &= (x_5 - x_1)(1 - \hat{x})(1 - \hat{y}) + (x_6 - x_2)\hat{x}(1 - \hat{y}) + (x_7 - x_3)\hat{x}\hat{y} + (x_8 - x_4)(1 - \hat{x})\hat{y}, \\ J_{3,2} &= (y_5 - y_1)(1 - \hat{x})(1 - \hat{y}) + (y_6 - y_2)\hat{x}(1 - \hat{y}) + (y_7 - y_3)\hat{x}\hat{y} + (y_8 - y_4)(1 - \hat{x})\hat{y}, \\ J_{3,3} &= (z_5 - z_1)(1 - \hat{x})(1 - \hat{y}) + (z_6 - z_2)\hat{x}(1 - \hat{y}) + (z_7 - z_3)\hat{x}\hat{y} + (z_8 - z_4)(1 - \hat{x})\hat{y}, \end{aligned}$$

and x_i, y_i and z_i denote the x, y and z components of the Lagrangian coordinate of vertex i . The determinant of the 3×3 Jacobian matrix is given by

$$\det \mathbf{J}_Z = J_{1,2} J_{2,3} J_{3,1} - J_{1,3} J_{2,2} J_{3,1} + J_{1,3} J_{2,1} J_{3,2} - J_{1,1} J_{2,3} J_{3,2} - J_{1,2} J_{2,1} J_{3,3} + J_{1,1} J_{2,2} J_{3,3}. \tag{25}$$

Using the chain rule and (22) we get the following transformation rule for the basis function gradients

$$\vec{\nabla} \phi_i = \mathbf{J}_Z^{-1} \vec{\nabla} \hat{\phi}_i. \tag{26}$$

The inverse Jacobian matrix is computed as

$$\mathbf{J}_Z^{-1} = \frac{1}{\det \mathbf{J}_Z} \begin{pmatrix} J_{2,2} J_{3,3} - J_{2,3} J_{3,2} & J_{1,3} J_{3,2} - J_{1,2} J_{3,3} & J_{1,2} J_{2,3} - J_{1,3} J_{2,2} \\ J_{2,3} J_{3,1} - J_{2,1} J_{3,3} & J_{1,1} J_{3,3} - J_{1,3} J_{3,1} & J_{1,3} J_{2,1} - J_{1,1} J_{2,3} \\ J_{2,1} J_{3,2} - J_{2,2} J_{3,1} & J_{1,2} J_{3,1} - J_{1,1} J_{3,2} & J_{1,1} J_{2,2} - J_{1,2} J_{2,1} \end{pmatrix}. \tag{27}$$

An example of the transformation process for one of the basis function gradients is shown in Fig. 4. It is important to note that the tri-linear Jacobian matrix (24), its determinant (25) and its inverse (27) are all functions of \hat{x}, \hat{y} and \hat{z} and therefore vary inside of a zone.

Given the previous definitions, we can discretize the variational form of the generalized viscous force term from (16)–(18). For every zone in our computational mesh, we calculate a finite element “stiffness” matrix of the form

$$(\mathbf{S}_Z)_{ij} = \int_{\hat{\Omega}_Z} \mu_Z (\mathbf{J}_Z^{-1} \vec{\nabla} \hat{\phi}_i) \cdot (\mathbf{J}_Z^{-1} \vec{\nabla} \hat{\phi}_j) |\det \mathbf{J}_Z|, \tag{28}$$

where \mathbf{J}_Z denotes the Jacobian matrix for zone Z . This 8×8 matrix is computed by performing the integral over the reference element $\hat{\Omega}_Z$ and making use of the parametric mapping. Note that, with the exception of μ_Z , every term under the integral is explicitly a function of the reference coordinates \hat{x}, \hat{y} and \hat{z} . The artificial viscosity coefficient μ_Z can be written as a function of the local coordinates as well by simply making use of the parametric mapping (20); however, it is often convenient to approximate this term by a zone-averaged quantity (see Section 7). We can therefore pull this term out from under the integral in (28).

Remark 1. In general, the stiffness matrix should include the boundary integrals corresponding to (7)–(9). This means that we have to compute additional terms when Ω_Z shares a part of the domain boundary.

Assume that this is the case, and denote with $\Gamma_Z = \{F_Z\}$ the collection of boundary faces in $\partial\Omega_Z \cap \partial\Omega$. Let $\hat{\Gamma}_Z = \{\hat{F}_Z\}$ be the corresponding collection of reference element faces and \mathbf{J}_{F_Z} be the Jacobian of the two-dimensional transformation from \hat{F}_Z to F_Z . Then

$$(\mathbf{S}_Z)_{ij} = \int_{\hat{\Omega}_Z} \mu_Z (\mathbf{J}_Z^{-1} \vec{\nabla} \hat{\phi}_i) \cdot (\mathbf{J}_Z^{-1} \vec{\nabla} \hat{\phi}_j) |\det \mathbf{J}_Z| - \sum_{\hat{F}_Z \in \hat{\Gamma}_Z} \int_{\hat{F}_Z} \mu_Z (\mathbf{J}_Z^{-1} \vec{\nabla} \hat{\phi}_j) \cdot \vec{n}_{F_Z} \hat{\phi}_i |\det \mathbf{J}_{F_Z}|, \tag{29}$$

where \vec{n}_{F_Z} is the outward unit normal to the face F_Z .

In this paper, we use formula (28) instead of (29), since the above boundary integrals are zero in the applications we consider, due to given boundary conditions, or the fact that shocks do not reach certain parts of the boundary (and therefore $\mu = 0$ there for all time). Indeed, the force term can be safely ignored on the (Dirichlet) part of the boundary where the velocity has prescribed values. The other common boundary condition is that of (Neumann) “symmetry plane”, where we restrict $\vec{v} \cdot \vec{n} = 0$ on a coordinate-aligned plane. This usually implies that one of the velocity components is zero, while the other two satisfy $\vec{\nabla} \vec{v} \cdot \vec{n} = 0$, annihilating the boundary terms in (7)–(9).

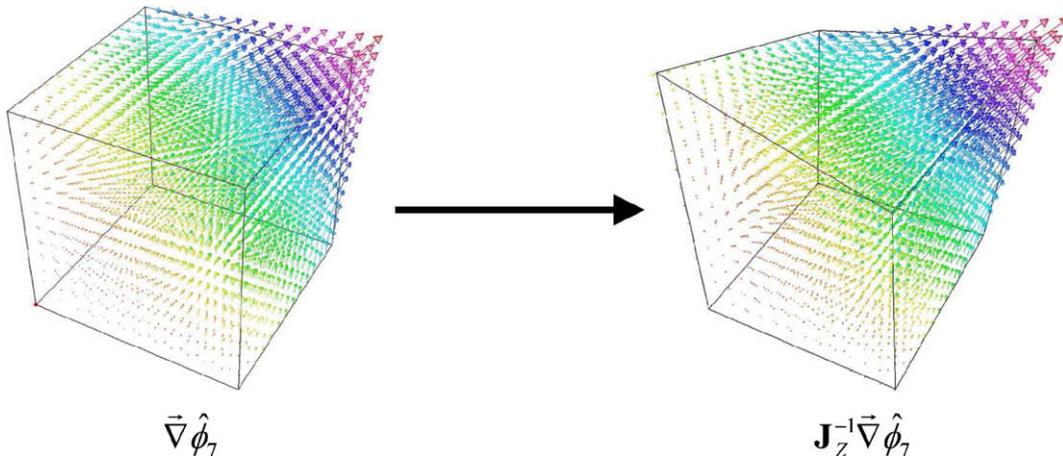


Fig. 4. Example of transformation process applied to one of the basis function gradients.

The integral (28) is approximated using Gaussian quadrature of a specified order, and can be written as a weighted sum of the form

$$(\mathbf{S}_Z)_{ij} \approx \mu_Z \sum_{n=1}^{k^3} w_n \left\{ (\mathbf{J}_Z^{-1} \vec{\nabla} \hat{\phi}_i) \cdot (\mathbf{J}_Z^{-1} \vec{\nabla} \hat{\phi}_j) |\det \mathbf{J}_Z| \right\}_{\hat{\mathbf{x}}=\hat{\mathbf{q}}_n}, \quad (30)$$

where w_n are the quadrature weights and $\hat{\mathbf{q}}_n$ are quadrature points where the integrand is sampled at. The quadrature rule is first defined in 1D by a set of k weights w_n and points q_n , where k denotes the order of the 1D quadrature rule. Specifically, we use Gauss–Lobatto quadrature where the endpoints of the 1D reference domain $[0,1]$ are included in the points q_n . The one dimensional Gauss–Lobatto weights and points on the interval $[0,1]$ for $k = 1, \dots, 4$ are defined in Table 5. To generate the corresponding 3D (or 2D) weights and points for the reference zone $\hat{\Omega}_Z$, we simply take a direct tensor product of the 1D weights and points, generating the total set of k^3 (k^2 in 2D) weights and points used in (30). In Section 6, we show that for the particular choice of $k = 2$ using a 2D method, we can exactly recover the original method of [1]. However, this is technically an “under-sampled” quadrature rule for the specific integrand of (28); we show that greater accuracy can be achieved if we use a higher order rule.

Once we have constructed this matrix for each zone, the zonal contribution to the nodal viscous force is given by

$$\vec{\mathbf{f}}_Z = -\mathbf{S}_Z \vec{\mathbf{v}}_Z. \quad (31)$$

These quantities are known as “corner forces.” The total force acting on node p is simply the sum of the corner forces at node p from all of the zones that share node p

$$\vec{\mathbf{f}}_p = \sum_{Z \ni p} \vec{\mathbf{f}}_Z. \quad (32)$$

The corresponding change in internal energy (i.e. shock heating) can be calculated by

$$m_Z \frac{\partial e_Z}{\partial t} = \vec{\mathbf{v}}_Z \cdot (\mathbf{S}_Z \vec{\mathbf{v}}_Z), \quad (33)$$

which is simply a discrete version of (2) without the pressure term. Note that the energy update term of (33) is in the so-called “compatible” form (e.g. [15]).

For the special case of 2D axisymmetric calculations (i.e. for bodies of revolution about some axis), the proposed artificial viscosity can be adapted in a straightforward manner using the so-called Petrov–Galerkin (or area weighting) approach. The nodal viscous force term of (32) that is added to the right-hand side of the discrete momentum equation is unchanged (equivalent to using area weighting in solving the momentum equation) while the “corner force” terms used in the energy update are weighted by the radial value of their associated vertex.

6. Connection with the method of Campbell and Shashkov

In this section, we consider a two-dimensional version of our tensor viscosity approach and establish a connection with the mimetic finite difference scheme of Campbell and Shashkov. Specifically, we show that in the model case $\mu = 1$, the *Div(-Grad)* operator from [1] coincides with an under-integrated version of the bi-linear finite element stiffness matrix. Thus, the tensor viscosity proposed in this paper can be viewed as a more accurate generalization of the one in [1].

We will describe the method and prove the equivalence on a fixed zone Ω_Z with counter-clockwise oriented vertices $\vec{\mathbf{x}}_1, \vec{\mathbf{x}}_2, \vec{\mathbf{x}}_3$ and $\vec{\mathbf{x}}_4$. Let \mathbf{S}_Z be the local stiffness matrix on Ω_Z , as in (28),

Table 5
Weights and points for 1D Gauss–Lobatto quadrature rules of order 1 through 4.

<i>Order 1</i>	
$w_1 = 1$	$q_1 = 0.5$
<i>Order 2</i>	
$w_1 = 1/2$	$q_1 = 0.0$
$w_2 = 1/2$	$q_2 = 1.0$
<i>Order 3</i>	
$w_1 = 1/6$	$q_1 = 0.0$
$w_2 = 4/6$	$q_2 = 0.5$
$w_3 = 1/6$	$q_3 = 1.0$
<i>Order 4</i>	
$w_1 = 1/12$	$q_1 = 0.0$
$w_2 = 5/12$	$q_2 = 0.276393$
$w_3 = 5/12$	$q_3 = 0.723607$
$w_4 = 1/12$	$q_4 = 1.0$

$$(\mathbf{S}_Z)_{ij} = \int_{\Omega_Z} \vec{\nabla} \phi_i \cdot \vec{\nabla} \phi_j = \int_{\hat{\Omega}_Z} (\mathbf{J}_Z^{-1} \vec{\nabla} \hat{\phi}_i) \cdot (\mathbf{J}_Z^{-1} \vec{\nabla} \hat{\phi}_j) |\det \mathbf{J}_Z|, \tag{34}$$

where \mathbf{J}_Z is the Jacobian of the transformation from $\hat{\Omega}_Z$ to Ω_Z . In two dimensions, $\hat{\Omega}_Z$ is the reference square with vertices

$$\hat{\mathbf{x}}_1 = (0, 0), \quad \hat{\mathbf{x}}_2 = (1, 0), \quad \hat{\mathbf{x}}_3 = (1, 1), \quad \hat{\mathbf{x}}_4 = (0, 1),$$

and corresponding basis functions

$$\hat{\phi}_1(\hat{x}, \hat{y}) = (1 - \hat{x})(1 - \hat{y}), \quad \hat{\phi}_2(\hat{x}, \hat{y}) = (1 - \hat{x})\hat{y}, \quad \hat{\phi}_3(\hat{x}, \hat{y}) = \hat{x}\hat{y}, \quad \hat{\phi}_4(\hat{x}, \hat{y}) = \hat{x}(1 - \hat{y}).$$

The Jacobian matrix has the form

$$\mathbf{J}_Z(\hat{x}, \hat{y}) = \begin{pmatrix} J_{1,1} & J_{1,2} \\ J_{2,1} & J_{2,2} \end{pmatrix} = \begin{pmatrix} \vec{x}_2 - \vec{x}_1 + (\vec{x}_3 - \vec{x}_2 - \vec{x}_4 + \vec{x}_1)\hat{y} \\ \vec{x}_4 - \vec{x}_1 + (\vec{x}_3 - \vec{x}_2 - \vec{x}_4 + \vec{x}_1)\hat{x} \end{pmatrix}, \tag{35}$$

and its inverse can be expressed in terms of its adjugate:

$$\mathbf{J}_Z^{-1} = \frac{1}{\det \mathbf{J}_Z} \mathbf{J}_Z^\perp \quad \text{where} \quad \mathbf{J}_Z^\perp = \begin{pmatrix} J_{2,2} & -J_{1,2} \\ -J_{2,1} & J_{1,1} \end{pmatrix}.$$

Note that the counter-clockwise orientation and (35) imply that the determinant of \mathbf{J}_Z is a *positive linear function*, and its values at the reference vertices are given by

$$\det \mathbf{J}_Z(\hat{\mathbf{x}}_1) = 2S_{412}, \quad \det \mathbf{J}_Z(\hat{\mathbf{x}}_2) = 2S_{123}, \quad \det \mathbf{J}_Z(\hat{\mathbf{x}}_3) = 2S_{234}, \quad \det \mathbf{J}_Z(\hat{\mathbf{x}}_4) = 2S_{341}. \tag{36}$$

Here S_{abc} stands for the area of the triangle with vertices \vec{x}_a, \vec{x}_b and \vec{x}_c .

The accurate computation of the entries of the stiffness matrix (34) requires high order quadrature rules since the integrand is, in general, a rational function. However, if we choose to sacrifice some accuracy, we can consider using a simple vertex-based 4-point quadrature, corresponding to the case $k = 2$ in Table 5. With this choice of approximation we get

$$\mathbf{S}_Z = \frac{1}{4} \sum_{k=1}^4 \frac{1}{\det \mathbf{J}_Z(\hat{\mathbf{x}}_k)} \mathbf{S}_{Z,k} \quad \text{where} \quad (\mathbf{S}_{Z,k})_{ij} = \vec{\nabla} \hat{\phi}_i(\hat{\mathbf{x}}_k)^T (\mathbf{J}_Z^\perp(\hat{\mathbf{x}}_k))^T \mathbf{J}_Z^\perp(\hat{\mathbf{x}}_k) \vec{\nabla} \hat{\phi}_j(\hat{\mathbf{x}}_k). \tag{37}$$

Similar to the notation in [1], let θ_k denote the angle at vertex \vec{x}_k ; l_{km} be the length of the edge connecting \vec{x}_k and \vec{x}_m ; and W_k be equal to one-half of the area of the triangle containing θ_k divided by the area of the zone, $|\Omega_Z|$.

With this notation, (36) implies

$$\frac{1}{\det \mathbf{J}_Z(\hat{\mathbf{x}}_1)} = \frac{|\Omega_Z|}{l_{41}^2 l_{12}^2} \frac{W_1}{\sin^2 \theta_1}, \quad \frac{1}{\det \mathbf{J}_Z(\hat{\mathbf{x}}_2)} = \frac{|\Omega_Z|}{l_{12}^2 l_{23}^2} \frac{W_2}{\sin^2 \theta_2}, \quad \frac{1}{\det \mathbf{J}_Z(\hat{\mathbf{x}}_3)} = \frac{|\Omega_Z|}{l_{23}^2 l_{34}^2} \frac{W_3}{\sin^2 \theta_3}, \quad \frac{1}{\det \mathbf{J}_Z(\hat{\mathbf{x}}_4)} = \frac{|\Omega_Z|}{l_{34}^2 l_{41}^2} \frac{W_4}{\sin^2 \theta_4}.$$

For example,

$$W_1 = \frac{1}{2} \frac{S_{412}}{|\Omega_Z|} \quad \text{by definition, so} \quad l_{41}^2 l_{12}^2 \sin^2 \theta_1 = S_{412}^2 = (2S_{412}) \left(\frac{1}{2} S_{412} \right) = (\det \mathbf{J}_Z(\hat{\mathbf{x}}_1)) (W_1 |\Omega_Z|).$$

Straightforward calculations using (35) reveal that

$$\begin{aligned} \mathbf{J}_Z^\perp(\hat{\mathbf{x}}_1)^T \mathbf{J}_Z^\perp(\hat{\mathbf{x}}_1) &= \begin{pmatrix} l_{41}^2 & -l_{41} l_{12} \cos \theta_1 \\ -l_{41} l_{12} \cos \theta_1 & l_{12}^2 \end{pmatrix}, \\ \mathbf{J}_Z^\perp(\hat{\mathbf{x}}_2)^T \mathbf{J}_Z^\perp(\hat{\mathbf{x}}_2) &= \begin{pmatrix} l_{23}^2 & l_{23} l_{12} \cos \theta_2 \\ l_{23} l_{12} \cos \theta_2 & l_{12}^2 \end{pmatrix}, \\ \mathbf{J}_Z^\perp(\hat{\mathbf{x}}_3)^T \mathbf{J}_Z^\perp(\hat{\mathbf{x}}_3) &= \begin{pmatrix} l_{23}^2 & -l_{23} l_{34} \cos \theta_3 \\ -l_{23} l_{34} \cos \theta_3 & l_{34}^2 \end{pmatrix}, \\ \mathbf{J}_Z^\perp(\hat{\mathbf{x}}_4)^T \mathbf{J}_Z^\perp(\hat{\mathbf{x}}_4) &= \begin{pmatrix} l_{41}^2 & l_{41} l_{34} \cos \theta_4 \\ l_{41} l_{34} \cos \theta_4 & l_{34}^2 \end{pmatrix}. \end{aligned}$$

In addition, note that for a given velocity field at the vertices, $\vec{\mathbf{v}}_Z = (\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{v}_4)$, we have

$$\sum_{k=1}^4 \vec{v}_k \vec{\nabla} \hat{\phi}_k(\vec{\mathbf{x}}_1) = \begin{pmatrix} \vec{v}_2 - \vec{v}_1 \\ \vec{v}_4 - \vec{v}_1 \end{pmatrix}, \quad \sum_{k=1}^4 \vec{v}_k \vec{\nabla} \hat{\phi}_k(\vec{\mathbf{x}}_2) = \begin{pmatrix} \vec{v}_2 - \vec{v}_1 \\ \vec{v}_3 - \vec{v}_2 \end{pmatrix}, \quad \sum_{k=1}^4 \vec{v}_k \vec{\nabla} \hat{\phi}_k(\vec{\mathbf{x}}_3) = \begin{pmatrix} \vec{v}_3 - \vec{v}_4 \\ \vec{v}_3 - \vec{v}_2 \end{pmatrix}, \quad \sum_{k=1}^4 \vec{v}_k \vec{\nabla} \hat{\phi}_k(\vec{\mathbf{x}}_4) = \begin{pmatrix} \vec{v}_3 - \vec{v}_4 \\ \vec{v}_4 - \vec{v}_1 \end{pmatrix}.$$

Thus,

$$\frac{1}{\det \mathbf{J}_Z(\hat{\mathbf{x}}_1)} \bar{\mathbf{v}}_Z \cdot (\mathbf{S}_{Z,1} \bar{\mathbf{v}}_Z) = \frac{1}{\det \mathbf{J}_Z(\hat{\mathbf{x}}_1)} \begin{pmatrix} \bar{v}_2 - \bar{v}_1 \\ \bar{v}_4 - \bar{v}_1 \end{pmatrix} \cdot \begin{pmatrix} l_{41}^2 & -l_{41} l_{12} \cos \theta_1 \\ -l_{41} l_{12} \cos \theta_1 & l_{12}^2 \end{pmatrix} \begin{pmatrix} \bar{v}_2 - \bar{v}_1 \\ \bar{v}_4 - \bar{v}_1 \end{pmatrix} \quad (38)$$

$$= |\Omega_Z| \frac{W_1}{\sin^2 \theta_1} \left\{ \frac{\|\bar{v}_1 - \bar{v}_4\|^2}{l_{41}^2} + \frac{\|\bar{v}_2 - \bar{v}_1\|^2}{l_{12}^2} + 2 \cos \theta_1 \frac{(\bar{v}_1 - \bar{v}_4) \cdot (\bar{v}_2 - \bar{v}_1)}{l_{12} l_{41}} \right\}. \quad (39)$$

Summing up analogous identities over the four vertices, we get a formula for $\bar{\mathbf{v}}_Z \cdot (\mathbf{S}_Z \bar{\mathbf{v}}_Z)$ which agrees with (35) from [1] in the case $\mu = 1$. Let p be a fixed point in Ω_Z , then the corner artificial viscosity force associated with p is simply given by $\bar{\mathbf{e}}_p \cdot (\mathbf{S}_Z \bar{\mathbf{v}}_Z)$, where $\bar{\mathbf{e}}_p$ is a vector which is one at p and zero in all the other vertices of the zone. For example,

$$-\bar{\mathbf{e}}_1 \cdot (\mathbf{S}_Z \bar{\mathbf{v}}_Z) = \frac{|\Omega_Z|}{4} \left(\frac{W_2}{\sin^2 \theta_2} \left\{ \frac{(\bar{v}_2 - \bar{v}_1)}{l_{12}^2} + \cos \theta_2 \frac{(\bar{v}_3 - \bar{v}_2)}{l_{12} l_{23}} \right\} + \frac{W_1}{\sin^2 \theta_1} \left\{ \frac{(\bar{v}_2 - \bar{v}_1)}{l_{12}^2} + \cos \theta_1 \frac{(\bar{v}_1 - \bar{v}_4)}{l_{12} l_{41}} \right\} \right. \\ \left. - \frac{W_1}{\sin^2 \theta_1} \left\{ \frac{(\bar{v}_1 - \bar{v}_4)}{l_{41}^2} + \cos \theta_1 \frac{(\bar{v}_2 - \bar{v}_1)}{l_{12} l_{41}} \right\} - \frac{W_4}{\sin^2 \theta_4} \left\{ \frac{(\bar{v}_1 - \bar{v}_4)}{l_{14}^2} + \cos \theta_4 \frac{(\bar{v}_4 - \bar{v}_3)}{l_{34} l_{41}} \right\} \right).$$

This is identical with the Ω_Z -contribution in (40) from [1], when $\mu = 1$. Therefore, we have established the following result.

Theorem 1. The artificial viscosity term obtained by (37) is the same as the one given by (40) in [1]. In other words, by using a vertex-based quadrature rule in the stiffness matrix, we recover the mimetic $\text{Div}(\text{Grad})$ tensor viscosity of Campbell and Shashkov. This equivalence holds on general unstructured quadrilateral grids.

Remark 2. The finite element point of view (38) makes it clear that all the required quantities are directly obtainable from the coordinates of the vertices. For example,

$$\det \mathbf{J}_Z(\hat{\mathbf{x}}_1) = \begin{vmatrix} \bar{\mathbf{x}}_2 - \bar{\mathbf{x}}_1 \\ \bar{\mathbf{x}}_4 - \bar{\mathbf{x}}_1 \end{vmatrix}, \quad l_{12}^2 = (\bar{\mathbf{x}}_2 - \bar{\mathbf{x}}_1) \cdot (\bar{\mathbf{x}}_2 - \bar{\mathbf{x}}_1), \quad l_{41} l_{12} \cos \theta_1 = (\bar{\mathbf{x}}_4 - \bar{\mathbf{x}}_1) \cdot (\bar{\mathbf{x}}_2 - \bar{\mathbf{x}}_1).$$

Remark 3. The finite element approach can also be used to derive a similar discretization in three dimensions. To outline this extension, let

$$\hat{\mathbf{x}}_1 = (0, 0, 0), \quad \hat{\mathbf{x}}_2 = (1, 0, 0), \quad \hat{\mathbf{x}}_3 = (1, 1, 0), \quad \hat{\mathbf{x}}_4 = (0, 1, 0), \quad \hat{\mathbf{x}}_5 = (0, 0, 1), \quad \hat{\mathbf{x}}_6 = (1, 0, 1), \quad \hat{\mathbf{x}}_7 = (1, 1, 1), \quad \hat{\mathbf{x}}_8 = (0, 1, 1)$$

be the vertices of the reference unit cube $\hat{\Omega}_Z$, cf. Fig. 3. Then

$$\det \mathbf{J}_Z(\hat{\mathbf{x}}_1) = \begin{vmatrix} \bar{\mathbf{x}}_2 - \bar{\mathbf{x}}_1 \\ \bar{\mathbf{x}}_4 - \bar{\mathbf{x}}_1 \\ \bar{\mathbf{x}}_5 - \bar{\mathbf{x}}_1 \end{vmatrix} = 6V_{1245},$$

where V_{abcd} stands for the volume of the tetrahedron with vertices $\bar{\mathbf{x}}_a, \bar{\mathbf{x}}_b, \bar{\mathbf{x}}_c$ and $\bar{\mathbf{x}}_d$. Furthermore,

$$\mathbf{J}_Z^{-1} = \frac{1}{\det \mathbf{J}_Z} \mathbf{J}_Z^\perp \quad \text{where, e.g.} \quad \mathbf{J}_Z^\perp(\bar{\mathbf{x}}_1) = \begin{pmatrix} (\bar{\mathbf{x}}_4 - \bar{\mathbf{x}}_1) \times (\bar{\mathbf{x}}_5 - \bar{\mathbf{x}}_1) \\ (\bar{\mathbf{x}}_5 - \bar{\mathbf{x}}_1) \times (\bar{\mathbf{x}}_2 - \bar{\mathbf{x}}_1) \\ (\bar{\mathbf{x}}_2 - \bar{\mathbf{x}}_1) \times (\bar{\mathbf{x}}_4 - \bar{\mathbf{x}}_1) \end{pmatrix}^T.$$

This means that,

$$\mathbf{J}_Z^\perp(\bar{\mathbf{x}}_1)^T \mathbf{J}_Z^\perp(\bar{\mathbf{x}}_1) = 4 \begin{pmatrix} S_{145}^2 & S_{125} S_{145} \cos \theta_1^{xy} & S_{124} S_{145} \cos \theta_1^{xz} \\ S_{145} S_{125} \cos \theta_1^{yx} & S_{125}^2 & S_{124} S_{125} \cos \theta_1^{yz} \\ S_{145} S_{124} \cos \theta_1^{zx} & S_{125} S_{124} \cos \theta_1^{zy} & S_{124}^2 \end{pmatrix},$$

where θ_1^{xy} denotes the angle between the inwardly-pointing normals of the x - and y -face of Ω_Z at $\bar{\mathbf{x}}_1$. Using these results, a mimetic-type tensor artificial viscosity can be developed based on formulas analogous to (38).

7. Non-linear viscosity coefficient

In this section we detail the treatment of the generalized tensor artificial viscosity coefficient μ_Z . This coefficient is a function of the velocity and the local grid size, making the artificial viscous force term non-linear in nature.

7.1. Guiding principles

Based on the properties suggested in [16], we formulate the following requirements for the artificial viscosity force and zonal heating in terms of the stiffness matrix:

- (S0) $\mathbf{S}_z \vec{v}_z$ should be invariant under orthogonal transformation of Ω_z and \vec{v}_z .
- (S1) $\vec{v}_z \cdot (\mathbf{S}_z \vec{v}_z) \geq 0$ (dissipativity, i.e. 2nd law of thermodynamics).
- (S2) $\mathbf{S}_z(\vec{v}_z + \vec{c}_z) = \mathbf{S}_z \vec{v}_z$, for any constant vector field \vec{c}_z (Galilean invariance).
- (S3) $\mathbf{S} \vec{v} = 0$, when \vec{v} is linear (related to self-similar motion invariance).
- (S4) $\mathbf{S} \vec{v}$ should preserve radial symmetry (related to wave-front invariance).
- (S5) $\mathbf{S}_z \vec{v}_z = 0$ if $\Delta \vec{v}_z > 0$ and $\mathbf{S}_z \vec{v}_z \rightarrow 0$ when $\Delta \vec{v}_z \rightarrow 0$ (viscous force continuity).

Here, $\Delta \vec{v}_z$ is a zonal function which is correlated with the compression of Ω_z in the direction of the shock wave propagation: $\Delta \vec{v}_z > 0$ when the zone is expanding, while $\Delta \vec{v}_z < 0$ when the zone is being compressed.

Note that the above list is not precisely the one given in [1]. In particular, the stiffness matrix automatically satisfies $\lim_{\vec{v}_z \rightarrow \vec{c}_z} \mathbf{S}_z \vec{v}_z = 0$, which was originally a part of (S2). Furthermore, (S3) implies that the force will be zero for global uniform contraction or rigid rotation which is the true definition of self-similar motion invariance. We have also replaced the general wave front invariance property with the more tractable requirement that the acceleration produced by $\mathbf{S} \vec{v}$ is radially symmetric whenever \vec{v} is. Finally, we added the condition of orthogonal transformation invariance, which we now define in more detail. Let $\Phi(\vec{x})$ be a linear orthogonal transformation, i.e. a combination of translations, rotations and reflections represented by

$$\Phi(\vec{x}) = A\vec{x} + \vec{b} \quad \text{where} \quad AA^T = I. \tag{40}$$

Let $\tilde{\Omega}_z = \Phi(\Omega_z)$ be the transformed zone, and $\tilde{\vec{v}}_z = \Phi(\vec{v}_z)$ be the transformed velocity at each point, $\Phi(\vec{v}_z) = A\vec{v}_z$. Denote with $\tilde{\mathbf{S}}_z$ the local stiffness matrix based on $\tilde{\Omega}_z$ and $\tilde{\vec{v}}_z$. Then (S0) states that the viscosity force should also be obtained by transformation with Φ , i.e., $\tilde{\mathbf{S}}_z \tilde{\vec{v}}_z = \Phi(\mathbf{S}_z \vec{v}_z)$. Since we use the same stiffness matrix for the x -, y - and z -components of the velocity, it is straightforward to see that the above condition can be equivalently restated as follows: \mathbf{S}_z should be invariant under orthogonal transformation of Ω_z and \vec{v}_z , i.e. $\tilde{\mathbf{S}}_z = \mathbf{S}_z$.

Remark 4. Note that, in principle, we could have considered different viscosity coefficients for the different components of the velocity, resulting in different stiffness matrices. This however would have broken the radial symmetry preservation property (S4), even on continuous level, e.g. $(\vec{\nabla} \cdot \mu_x \vec{\nabla} v_x, \vec{\nabla} \cdot \mu_y \vec{\nabla} v_y)$ preserves radial symmetry if and only if $\mu_x = \mu_y$ is radially symmetric. The same holds true for $(\vec{\nabla} \cdot \mu \vec{\nabla} v_x, \vec{\nabla} \cdot \mu \vec{\nabla} v_y)$ with $\mu = \begin{pmatrix} \mu_x & 0 \\ 0 & \mu_y \end{pmatrix}$. This can be seen, in both cases, by considering μ_x and μ_y to be constants (or functions of x and y) and picking a particular velocity field, such as $\vec{v} = (x^2 + y^2) \begin{pmatrix} x \\ y \end{pmatrix}$. Similar arguments, involving (S4) and (S0), rule out the possibility of using a general matrix instead of a scalar coefficient for μ . For example, using the matrix $\mu = \vec{\nabla} \vec{v}$ will preserve radial symmetry, but is not invariant with respect to the orthogonal linear transformations.

To further elaborate on the condition $\tilde{\mathbf{S}}_z = \mathbf{S}_z$, note that $\tilde{\mathbf{J}}_z = \mathbf{J}_z A^T$, which implies due to (40) and (28) that

$$(\tilde{\mathbf{S}}_z)_{ij} = \int_{\tilde{\Omega}_z} \tilde{\mu}_z (A \mathbf{J}_z^{-1} \vec{\nabla} \hat{\phi}_i) \cdot (A \mathbf{J}_z^{-1} \vec{\nabla} \hat{\phi}_j) |\det A| |\det \mathbf{J}_z| = \int_{\tilde{\Omega}_z} \tilde{\mu}_z (\mathbf{J}_z^{-1} \vec{\nabla} \hat{\phi}_i) \cdot (\mathbf{J}_z^{-1} \vec{\nabla} \hat{\phi}_j) |\det \mathbf{J}_z|. \tag{41}$$

The right-hand side clearly equals $(\mathbf{S}_z)_{ij}$ when $\tilde{\mu}_z = \mu_z$.

Given the above considerations and the fact that \mathbf{S}_z is a discretization of $\text{Div}(\text{Grad})$ the requirements (S0)–(S5) can be reduced to the following set of conditions on the viscosity coefficient:

- (M0) μ_z should be invariant under orthogonal transformation of Ω_z and \vec{v}_z .
- (M1) $\mu_z(\vec{v}_z) \geq 0$.
- (M2) $\mu_z(\vec{v}_z + \vec{c}_z) = \mu_z(\vec{v}_z)$ for constant \vec{c}_z .
- (M3) $\mu(\vec{v}) = 0$ for linear \vec{v} .
- (M4) $\mu(\vec{v})$ should be radially symmetric if \vec{v} is.
- (M5) $\mu_z(\vec{v}_z) = 0$ if $\Delta \vec{v}_z > 0$ and $\mu_z(\vec{v}_z) \rightarrow 0$ when $\Delta \vec{v}_z \rightarrow 0$.

The first three conditions (M0)–(M2) simply mean that μ_z should be a positive number in each zone which uses only derivatives/differences of the velocity, and involves only topologically “local” (or zonal) information. These are generally easy to satisfy and do not need special considerations.

Traditionally, (M3) is addressed through the use of limiter functions which compare ratios of $\Delta \vec{v}_z$ in neighboring zones, cf. [7,1]. In this paper, we develop a novel finite element approach for satisfying (M3). We do this by first constructing a nodal value that is zero at a vertex p whenever \vec{v} is locally harmonic in a neighborhood of p . This number is a measure of the local

smoothness of the velocity and could be directly applied to the total viscous force at the point. This will make the force zero whenever \vec{v} is locally linear, which results in the following point-wise version of the (S3) property:

$$(S3)' \quad (\mathbf{S}\vec{v})_p = 0, \text{ when } \vec{v} \text{ is locally harmonic around the vertex } p.$$

This straightforward approach however, would have negative implications for energy conservation. Instead, we turn the nodal smoothness detector into a zone based analog of a limiter to apply directly to μ_z ; the details of this are given in Section 7.4.

We already used the continuous version of (M4) to deduce that μ should be a scalar coefficient, see Remark 4. In practice however, radial symmetry is preserved exactly only in some very special cases. For example, if the mesh consists of radial slices which can be obtained from each other by rotation (as on the left in Fig. 10), then (S0) will ensure (S4). In general, we only get approximate symmetry preservation as will be shown in the numerical results in Section 8.2.

Finally, condition (M5) follows directly once an appropriate definition for $\Delta\vec{v}_z$ has been chosen. This, together with the other ingredients necessary to define μ is discussed in the following section.

7.2. General form of the viscosity coefficient

Starting from the requirements listed in the previous section, we next determine what should be the overall form of μ_z . We do this by analogy with the standard scalar viscosity approach, which was summarized in Section 3. We begin with the classical Von Neumann–Richtmyer artificial viscosity coefficient [4], with a linear term due to Landshoff [6]:

$$q_z = \psi \rho_z \Delta\vec{v}_z (q_{quad} \Delta\vec{v}_z - q_{lin} c_z).$$

Here $\Delta\vec{v}_z$, ρ_z and c_z are the zone-based jump in the velocity, density and sound speed respectively and ψ is a binary switch setting $q_z = 0$, when $\Delta\vec{v}_z > 0$. The coefficients q_{lin} and q_{quad} allow for independent scaling of the linear and quadratic terms and are additionally used in artificial viscosity formulations.

One of the simplest, and most typical choices for the velocity jump in a multi-dimensional context is $\Delta\vec{v}_z = (\vec{\nabla} \cdot \vec{v})_z \ell_z$, where ℓ_z denotes a “characteristic length scale”, while $(\vec{\nabla} \cdot \vec{v})_z$ stands for the divergence of the velocity field evaluated in the center of the zone. This leads to following zonal heating term in (5)

$$-q_z (\vec{\nabla} \cdot \vec{v})_z = -[\psi \rho_z \ell_z (q_{quad} \Delta\vec{v}_z - q_{lin} c_z)] (\vec{\nabla} \cdot \vec{v})_z (\vec{\nabla} \cdot \vec{v})_z,$$

which can be viewed as an approximation of $-\vec{v}_z \cdot (\mathbf{S}_h^q \vec{v}_z)$, where \mathbf{S}_h^q is the *div–div* stiffness matrix

$$(\mathbf{S}_h^q)_{ij} = \int_{\Omega_z} [\psi \rho_z \ell_z (q_{quad} \Delta\vec{v}_z - q_{lin} c_z)] (\vec{\nabla} \cdot \vec{\phi}_i) (\vec{\nabla} \cdot \vec{\phi}_j).$$

The difference between this term and the one in (33) is simply in the bilinear form used to generate the stiffness matrix. Therefore, a natural starting point for μ_z is the expression

$$\mu_z = -\psi \rho_z \ell_z (q_{quad} \Delta\vec{v}_z - q_{lin} c_z).$$

This form is analogous to the connection between (28) and (5) in [1], and allows for easy computation of the related scalar coefficient (which may be needed in other parts of a general hydro code) by the formula

$$q_z = \mu_z (\vec{\nabla} \cdot \vec{v})_z. \quad (42)$$

Motivated by this analysis and the conditions of the previous section, we propose the following general form for the non-linear viscosity coefficient:

$$\mu_z = \psi_0 \psi_1 \rho_z \ell_z (q_{quad} \ell_z |C_z| + \psi_2 q_{lin} c_z), \quad (43)$$

where

ρ_z, c_z	zonal density and sound speed
q_{lin}, q_{quad}	independent linear and quadratic scaling parameters
$\ell_z = \ell_z(\Omega_z)$	measure of the diameter of Ω_z (in the direction of the shock)
$C_z = C_z(\Omega_z, \vec{v}_z)$	measure of compressibility (in the direction of the shock)
$\mathcal{V}_z = \mathcal{V}_z(\Omega_z, \vec{v}_z)$	measure of the vorticity of the velocity
$\psi_0 = \psi_0(\Omega, \vec{v})$	measure of smoothness in $[0, 1]$: $\psi_0 = 0$ for locally harmonic velocity field, while $\psi_0 = 1$ for non-smooth \vec{v}
$\psi_1 = \psi_1(C_z)$	compressibility switch: $\psi_1 = 0$ if $C_z \geq 0$, otherwise $\psi_1 = 1$.
$\psi_2 = \psi_2(C_z, \mathcal{V}_z)$	vorticity/compression measure in $[0, 1]$: $\psi_2 = 0$ when $\mathcal{V}_z \gg C_z$ and $\psi_2 = 1$ when $\mathcal{V}_z \ll C_z$

Note that we have introduced three switch-like terms, ψ_0 , ψ_1 and ψ_2 . The first term ψ_0 is the zone based coefficient that is used to drive the viscosity coefficient to zero whenever the velocity is locally harmonic, i.e. this term acts as a “smoothness” sensor. This is analogous to the monotonic limiters frequently used in Lagrangian hydro codes. The details on how this term is computed are given in Section 7.4. The second switch, ψ_1 , is designed to ensure that shock heating only occurs for zones in

compression, while ψ_2 is designed to turn off the linear term of the artificial viscosity for “shear dominated” zones. This term is also required to satisfy the viscous force continuity property of (M5). In practice, we have found that ψ_2 is essential to prevent the spurious suppression of vorticity. In Section 8.8 we give a specific example of this behavior. Specifically, we use the form (cf. [17])

$$\psi_2(C_Z, \nu_Z) = \frac{C_Z}{C_Z + \alpha \nu_Z}, \tag{44}$$

where α is a scalar coefficient which we set equal to 1. Other possible choices for ψ_2 , with different rates of transition from 0 to 1, are as follows:

$$\psi_2(C_Z, \nu_Z) = 1 - \exp\left(-\frac{C_Z}{\alpha \nu_Z}\right) \quad \text{and} \quad \psi_2(C_Z, \nu_Z) = \frac{2}{\pi} \arctan\left(\frac{C_Z}{\alpha \nu_Z}\right).$$

Note that (43) automatically satisfies (M1) and (M5) with $\Delta \vec{v}_Z = \ell_Z C_Z$. Condition (M2) gets transferred to C_Z and ν_Z , while (M0) has to be satisfied by ℓ_Z, C_Z and ν_Z .

7.3. Artificial viscosity models

In this subsection we propose three artificial viscosity coefficients obtained by specific choices for ℓ_Z, C_Z and ν_Z in (43). First, we define a zone centered velocity gradient tensor of the form

$$\vec{\nabla} \vec{v}_Z = \begin{pmatrix} (\vec{\nabla} v_x)_Z \\ (\vec{\nabla} v_y)_Z \\ (\vec{\nabla} v_z)_Z \end{pmatrix}, \tag{45}$$

by evaluating the gradient of the finite element basis functions in the center of the zone:

$$\begin{aligned} (\vec{\nabla} v_x)_Z &= \sum_{i=1}^8 \alpha_i (\mathbf{J}_Z^{-1} \vec{\nabla} \hat{\phi}_i)(0.5, 0.5, 0.5), \\ (\vec{\nabla} v_y)_Z &= \sum_{i=1}^8 \beta_i (\mathbf{J}_Z^{-1} \vec{\nabla} \hat{\phi}_i)(0.5, 0.5, 0.5), \\ (\vec{\nabla} v_z)_Z &= \sum_{i=1}^8 \gamma_i (\mathbf{J}_Z^{-1} \vec{\nabla} \hat{\phi}_i)(0.5, 0.5, 0.5). \end{aligned}$$

The zone centered divergence can now be evaluated by simply taking the trace of this 3×3 (2×2 in 2D) tensor:

$$(\vec{\nabla} \cdot \vec{v})_Z = \text{tr}(\vec{\nabla} \vec{v}_Z). \tag{46}$$

Likewise, the cell centered curl of the velocity, $(\vec{\nabla} \times \vec{v})_Z$, can be evaluated by using the nine components of this tensor.

It is natural to consider $(\vec{\nabla} \cdot \vec{v})_Z$ as a measure of compression and $\|(\vec{\nabla} \times \vec{v})_Z\|$ as a measure of vorticity. Coupling this with the area-based “characteristic length scale” Δh , that is typically used in the CFL-based time stepping control, we get our first viscosity model.

Model 1	ℓ_Z	C_Z	ν_Z
$\mu = \mu_{div}$	Δh	$(\vec{\nabla} \cdot \vec{v})_Z$	$\ (\vec{\nabla} \times \vec{v})_Z\ $

Note that (M4) can clearly be violated in this model, since $(\vec{\nabla} \cdot \vec{v})_Z$ is proportional to the change in volume, and therefore is not generally radial. On the other hand, (M0) is satisfied, since

$$\vec{\nabla} \cdot \vec{v} = \sum_k \nabla \phi_k \cdot \vec{\nabla}_{Z,k} = \sum_k \mathbf{J}_Z^{-1} \nabla \hat{\phi}_k \cdot \vec{\nabla}_{Z,k} \quad \text{and} \quad \vec{\nabla} \times \vec{v} = \sum_k \nabla \phi_k \times \vec{\nabla}_{Z,k} = \sum_k \mathbf{J}_Z^{-1} \nabla \hat{\phi}_k \times \vec{\nabla}_{Z,k}$$

are transformed similarly to (41):

$$\vec{\nabla} \cdot \vec{v} = \sum_k \mathbf{A} \mathbf{J}_Z^{-1} \nabla \phi_k \cdot \mathbf{A} \vec{\nabla}_{Z,k} \quad \text{and} \quad \vec{\nabla} \times \vec{v} = \sum_k \mathbf{A} \mathbf{J}_Z^{-1} \nabla \phi_k \times \mathbf{A} \vec{\nabla}_{Z,k}.$$

Since orthogonal transformations preserve the magnitudes and angles between vectors, it follows that

$$(\vec{\nabla} \cdot \vec{v})_Z = (\vec{\nabla} \cdot \vec{v})_Z \quad \text{and} \quad \|(\vec{\nabla} \times \vec{v})_Z\| = \|(\vec{\nabla} \times \vec{v})_Z\|.$$

The rest of the conditions, (M1), (M2) and (M5) are also satisfied due to the form of C_Z, ν_Z and ψ_0 .

We next consider a viscosity coefficient having different components in the topological i -, j - and k -directions in the zone. Specifically, let $\Delta\vec{i}, \Delta\vec{j}, \Delta\vec{k}$ be the vectors connecting the mid-points of the opposite faces of Ω_Z . The change in the velocity in direction $\Delta\vec{i}$ is simply given by $\Delta_i \vec{v} = \vec{v}(\vec{B}) - \vec{v}(\vec{A})$, where \vec{A} and \vec{B} are the face mid-points used to define $\Delta\vec{i} = \vec{B} - \vec{A}$ (see Fig. 5 for a 2D illustration). With this notation, we can define ℓ_z and C_z specific for the i -direction, resulting in coefficient μ_i that is sensitive only to compression in this one logical direction. Similarly we can define μ_j and μ_k and set the overall viscosity coefficient to be the sum of the three. This gives us a second, logically directional, model.

Model 2	ℓ_z	C_z	ν_z
$\mu = \mu_i+$	$\ \Delta\vec{i}\ $	$(\Delta_i \vec{v} \cdot \Delta\vec{i}) / (\Delta\vec{i} \cdot \Delta\vec{i})$	$\ (\vec{\nabla} \times \vec{v})_z\ $
μ_j+	$\ \Delta\vec{j}\ $	$(\Delta_j \vec{v} \cdot \Delta\vec{j}) / (\Delta\vec{j} \cdot \Delta\vec{j})$	$\ (\vec{\nabla} \times \vec{v})_z\ $
μ_k	$\ \Delta\vec{k}\ $	$(\Delta_k \vec{v} \cdot \Delta\vec{k}) / (\Delta\vec{k} \cdot \Delta\vec{k})$	$\ (\vec{\nabla} \times \vec{v})_z\ $

Note that (M0)–(M2) and (M5) are clearly satisfied by this model, while (M4) looks promising since potentially one-dimensional shocks can be detected in the direction of their propagation. This can be further improved by straightening the edges using the associated cubeoctahedron (parallelogram in 2D).

The final model involves explicitly an approximation of the shock direction. First, we compute a vector $\Delta\vec{s}$ by averaging the velocity in the zone center. Since this gives the average direction in which the zone is moving, it is reasonable to suggest that it will be close to the actual shock direction. Let, as above, $\Delta\vec{s} = \vec{B} - \vec{A}$, where \vec{A} and \vec{B} are the points where the line through the center of the zone in the averaged velocity direction intersects the boundary, see Fig. 5. Then, $\Delta_s \vec{v} = \vec{v}(\vec{B}) - \vec{v}(\vec{A})$, and we can define our third, shock-aware, model.

Model 3	ℓ_z	C_z	ν_z
$\mu = \mu_s$	$\ \Delta\vec{s}\ $	$(\Delta_s \vec{v} \cdot \Delta\vec{s}) / (\Delta\vec{s} \cdot \Delta\vec{s})$	$\ (\vec{\nabla} \times \vec{v})_z\ $

Again, all properties except the radial symmetry preservation are clearly satisfied, and (M4) should be also satisfied since the computation of $\Delta\vec{s}$ should preserve the symmetry. Note that the determination of $\Delta\vec{s}$ is more complicated compared with the other models and requires a “ray-tracing” algorithm for computing the intersection points A and B .

Each of the three proposed models has its advantages in certain zonal configurations. Some of the differences are illustrated in Fig. 5, where we consider the deformation of two simple two-dimensional zones. When the volume of the zone is not changed, as in the first case, Model 1 will detect no compression, even though the zone is clearly squeezed in the direction of $\Delta\vec{s}$, and therefore Model 3 will indicate compression. Model 2 will detect expansion in i -direction and compression in j -direction, which also seems appropriate. In the second test, Model 1 will give compression, despite the fact that the sides of the trapezoid are being stretched. This will be detected as expansion in j -direction by Model 2, with some small compression in i -direction. Model 3, in contrast, will only indicate expansion.

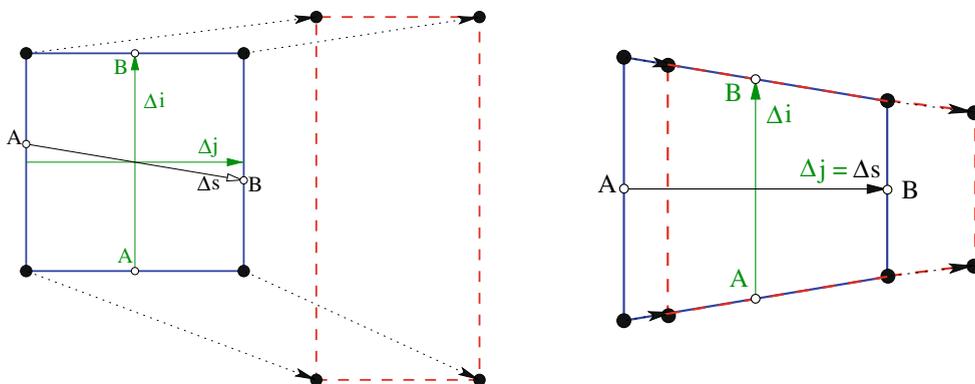


Fig. 5. Comparison of the different measures of compressibility in two model cases. On the left, a square zone is deformed into a rectangle of the same area. On the right, an equilateral trapezoid is stretched into one with a smaller area. In both cases the original zone is in solid blue, the new zone is in dashed red and the velocity is in dotted black. The $\Delta\vec{i}$ and $\Delta\vec{j}$ vectors associated with Model 2 are given in green, while $\Delta\vec{s}$ from Model 3 is in solid black. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

7.4. FEM-based smoothness detector

Let \mathbf{S}^0 be the stiffness matrix corresponding to $\mu = 1$ (i.e. the unscaled stiffness matrix) and consider the associated corner forces

$$\vec{\mathbf{f}}_Z^0 = \mathbf{S}_Z^0 \vec{\mathbf{v}}_Z.$$

Now, define the sum of these unscaled corner forces at a point p to be

$$\vec{\mathbf{f}}_p^0 = \sum_{Z \ni p} \vec{\mathbf{f}}_Z^0$$

and assume that \vec{v} is locally smooth. Then, using the Gauss divergence theorem, we obtain

$$\vec{\mathbf{f}}_p^0 = \int_{\Omega} \vec{\nabla} \vec{v} \cdot \vec{\nabla} \phi_p = - \int_{\Omega_p} \Delta \vec{v} \phi_p.$$

In other words, $\vec{\mathbf{f}}_p^0 = 0$ when \vec{v} is harmonic in Ω_p . This implies that $\vec{\mathbf{f}}_p^0$ is a useful measure of the local “smoothness” of the velocity field in the neighborhood of point p . This is the first ingredient of our zone based smoothness measure ψ_0 . In order to turn this value into a smooth function which varies in the range $[0,1]$, we need a second quantity to compare against to determine when $\vec{\mathbf{f}}_p^0$ is sufficiently large to describe a local shock. To be consistent, this quantity needs to be dimensionally equivalent to $\vec{\mathbf{f}}_p^0$. Specifically, consider the term

$$\mathbf{g}_p = \sum_{Z \ni p} \frac{|\Omega_Z| c_Z}{\ell_Z^2},$$

where $|\Omega_Z|$ is the volume (area in 2D) of the zone Z , c_Z is local sound speed and ℓ_Z is the length scale measure of zone Z . Given these two values, we now define the point-wise function

$$\psi_0(\vec{\mathbf{f}}_p^0, \mathbf{g}_p) = 1 - \exp\left(-\frac{\|\vec{\mathbf{f}}_p^0\|}{\alpha |\mathbf{g}_p|}\right).$$

This function is defined for every node in the mesh. To construct the zone based version, we simply take the maximum value in the neighborhood of the zone, specifically

$$(\psi_0)_Z = \max_{p \in Z} \{ (\psi_0)_p \}.$$

In practice, we have found that a value of $\alpha = 0.005$ works quite well for a large range of calculations. We remark that the computations required for ψ_0 can be incorporated in the momentum and energy update without significant additional effort.

8. Numerical results

In this section we present results from a sequence of numerical experiments that increase in complexity. The problems are designed to exercise features of our formulation on simple verification problems as well as more complicated shock driven hydrodynamics calculations. We perform our tests on a variety of computational grids and compare results to multiple methods. The numerical shock hydro experiments were conducted in the framework of an existing ALE hydro code, cf. [18,19], in order to showcase the plug-in character of the proposed tensor artificial viscosity. Therefore, the quality of the full simulation results below is limited by the inaccuracies of the existing staggered-grid method, such as the HEMP operator for computing the pressure gradient and the approach for computing nodal momentum using simple nodal masses. Developing a comprehensive FEM-based discretization for the whole hydro system is an ongoing effort and will be addressed in future publications.

Each example below uses the following consistent settings: for the linear and quadratic coefficients, we use $q_{lin} = q_{quad} = 1.0$, for the FEM quadrature rule order we use $k = 3$, the FEM-based smoothness measure ψ_0 has a coefficient $\alpha = 0.005$ while the vorticity/compression measure in the linear term ψ_2 has a coefficient $\alpha = 1.0$. Unless otherwise specified, we use the tensor artificial viscosity Model 1. Note that tuning these parameters for specific problems may result in further improvements, but we choose to use the above baseline values since in our experience they produce uniformly good results.

8.1. Verification of discrete Div(Grad) operator

In this example we verify our discretization of the *Div(Grad)* operator for the simple case of $\mu = 1$ on a 2D grid. We also provide direct numerical evidence of the equivalence of our method under a particular quadrature rule to that of Campbell and Shashkov as derived in Section 6. Consider a 2D velocity vector field of the form

$$\vec{v} = \{ \cos(\pi(x+y)), \sin(\pi(x-y)) \}, \tag{47}$$

which has both a well defined curl and divergence. Applying the $Div(Grad)$ operator to this field yields

$$\vec{\nabla} \cdot (\vec{\nabla} \vec{v}) = \{-2\pi^2 \cos(\pi(x+y)), -2\pi^2 \sin(\pi(x-y))\}. \quad (48)$$

A 2D vector plot of this field is shown in Fig. 8.

We begin with a randomly distorted 12×12 , 2D grid of the unit box and generate two refined versions of this grid as shown in Fig. 6. We initialize the velocity at every node in the grid to that of (47). We now compute the action of $Div(Grad)$ on this field using three methods: the mimetic finite difference (MFD) method of Campbell and Shashkov as described in [1], the finite element method (FEM) using a low order quadrature rule of order $k = 2$ and the finite element method using a higher order quadrature rule of order $k = 3$. In addition, we also consider the full finite element method (FEM-EP) where higher order quadrature rules are used to evaluate both the stiffness and the full “mass matrix” corresponding to the left-hand side of (13)–(15). Unlike the other approaches, this requires discrete velocity \vec{v}_z defined by a global elliptic projection onto the finite element space (rather than simple point-wise evaluation of the initial velocity field at the nodes). To remain compatible with existing staggered-grid hydro methods, we have focused primarily on the case of a diagonal mass matrix, but we nevertheless include the above result in order to demonstrate the potential benefit of using a consistent finite element discretization for all terms in the Euler equations.

We compute the error of each method using the L_2 integral norm on each of the three meshes and use this data to compute convergence plots as shown in Fig. 7. The numerical results obtained with the MFD method and the FEM with $k = 2$ are the same to machine precision, confirming the results of Section 6. Furthermore, a smaller absolute error and a faster error convergence rate is achieved by using a higher order $k = 3$ (i.e. nine points per zone) quadrature rule. The full finite element method with elliptic projection gives an optimal, second order convergence rate in L_2 .

Next we consider a fully randomized 24×24 grid as shown in Fig. 8 and apply both the MFD method and the FEM with order $k = 3$ quadrature. In Fig. 9 we visually compare the results obtained with each method by plotting the computed vector field. It is clear from the plots that the FEM with higher order quadrature is superior at reproducing the analytic field as plotted in Fig. 8.

8.2. Verification of radial symmetry preservation

In this section we numerically investigate properties (S4) and (M4), i.e. to what extent the artificial viscosity term preserves radial symmetry. In addition, we demonstrate the invariance of the proposed schemes under orthogonal transformations, confirming the theoretically established properties (S0) and (M0). For simplicity, we consider two-dimensional problems using the polar meshes depicted in Fig. 10 and an initial velocity field given by

$$\vec{v}(x, y) = (x^2 + y^2)(x, y), \quad \text{or in polar coordinates,} \quad \vec{v}(r, \theta) = r^3 \vec{r}.$$

The mesh and the above velocity are then transformed by an orthogonal linear transformation consisting of a rotation on 30° , followed by reflection with respect to the y -axis and translation on $(\frac{2}{3}, 1)$. The initial and final configurations are illustrated in Fig. 11.

Even though this section is focused on the momentum equation, radial symmetry needs to also be preserved in the internal energy update. This requires that $\vec{v}_z \cdot (\mathbf{S}_z \vec{v}_z)$ is radially symmetric whenever \vec{v} is. Note that when the angular spacing is uniform (as in the mesh on the left of Fig. 11) property (S0) guarantees that the “corner forces” $\mathbf{S}_z \vec{v}_z$ will be radially symmetric. This is due to the fact that zones at the same radial distance are obtained from each other by an orthogonal transformation (rotation at a fixed angle). This means that both the acceleration and the internal energy update will be radially symmetric in this case. In general, when there is a variation in the angles (as in the mesh on the right of Fig. 11) property (S0)

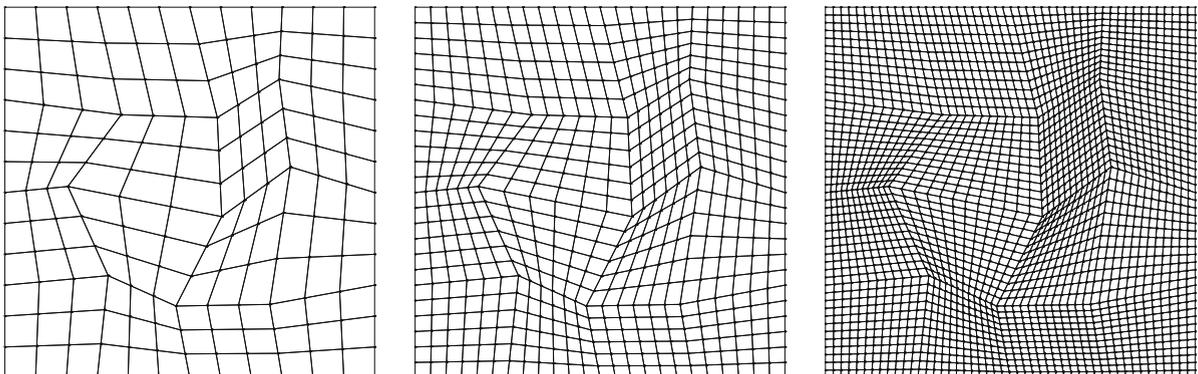


Fig. 6. A randomly distorted 2D grid along with two levels of refinement.

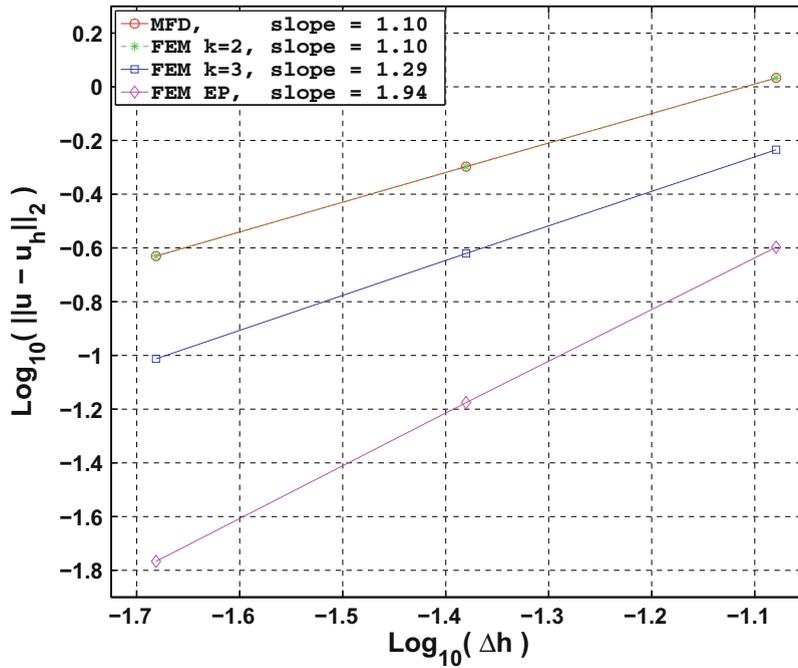


Fig. 7. Convergence of the approximation error using the L_2 norm and four different methods. Note that the mimetic finite difference (MFD) and the finite element method (FEM) with a low order ($k = 2$) quadrature rule yield identical results. A smaller absolute error with a faster convergence rate can be achieved by using a higher order ($k = 3$) quadrature rule for the finite element stiffness matrix. Finally, by applying a full finite element treatment using initial velocity given by a global elliptic projection and solving a linear system involving a mass matrix, we recover an optimal second order convergence rate with a much smaller absolute error.

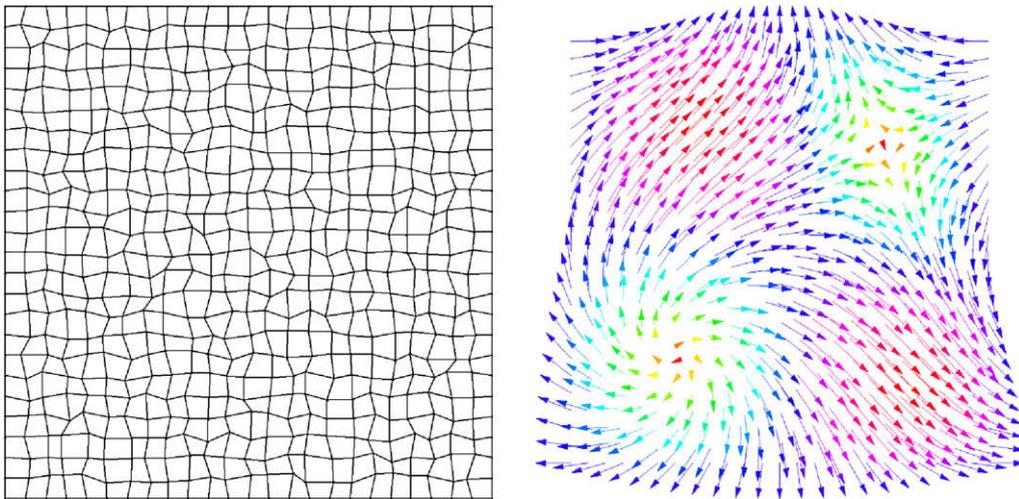


Fig. 8. Fully randomized 24×24 grid (left) and exact value of $\nabla \cdot (\nabla \vec{v})$ plotted at each vertex (right).

does not apply and we will see that radial symmetry is lost in the momentum equation. This implies (even though we don't show it here) corresponding loss of symmetry preservation in the internal energy equation.

Since in this paper we focus on the case of a diagonal mass matrix, the acceleration is computed simply by rescaling of the force term: $\vec{a}_p = (\mathbf{S}\vec{v})_p/m_p$ at each point p . Here m_p denotes the nodal mass, which is computed by lumping the rows of the full mass matrix \mathbf{M} which is in turn computed with a quadrature of order k (the same as the stiffness matrix). We proceed to investigate the preservation of radial symmetry in the interior nodes, by considering the following two components of radial symmetry:

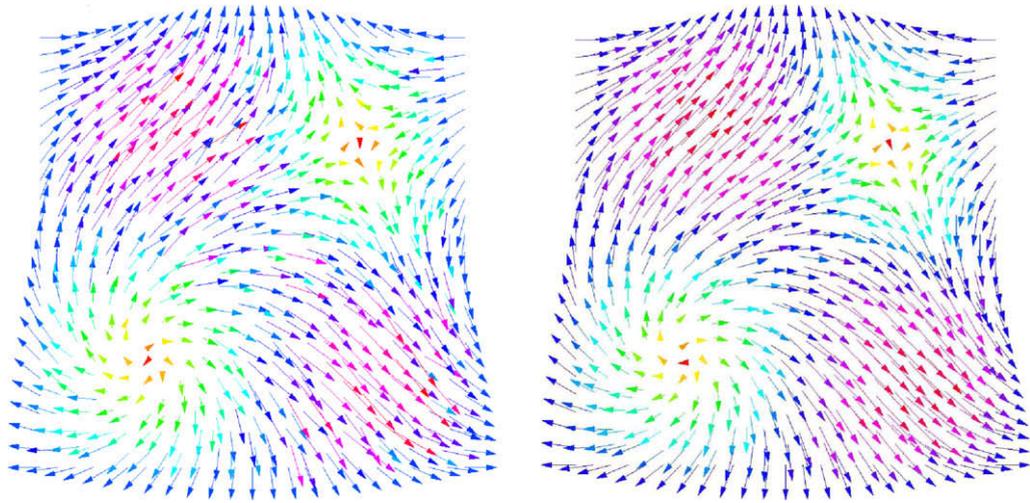


Fig. 9. Computed values of $\bar{\nabla} \cdot (\bar{\nabla} \bar{v})$ using the MFD method (left) and the FEM with $k = 3$ (right).

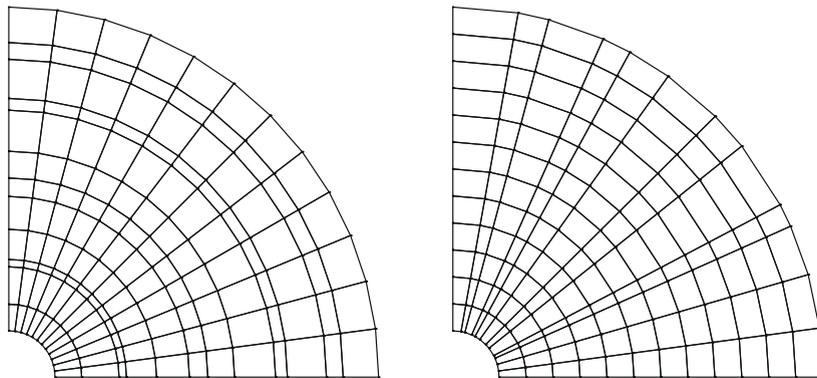


Fig. 10. Computational meshes used in the numerical experiments verifying properties (M4) and (S4). All meshes are obtained from an initial uniform polar mesh which is perturbed in radial direction only (left), or in angular direction only (right).

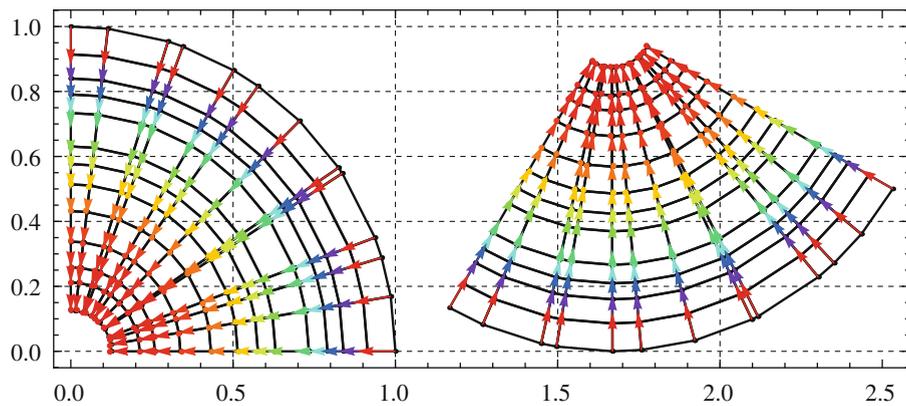


Fig. 11. Initial polar mesh with radial velocity centered at the origin, together with their image under orthogonal transformation.

1. Preservation of the direction of the vector field, i.e. to what extent the computed acceleration field \bar{a} and the original velocity \bar{v} point in the same direction. We quantify this using the following measure:

$$\alpha = \max_p |\alpha_p(\bar{a}, \bar{v})|, \quad \text{where} \quad \alpha_p(\bar{a}, \bar{v}) = \arcsin \left(\frac{|\bar{a}_p \times \bar{v}_p|}{\|\bar{a}_p\| \|\bar{v}_p\|} \right).$$

2. Preservation of the r -uniform magnitude of the vector field, i.e. to what extent \vec{a}_p has the same magnitude for points with the same polar radius r . This can be measured as the maximum standard deviation of the magnitudes over the parallel circular arcs having the same value of r :

$$\sigma = \max_r \sigma_r(\vec{a}), \quad \text{where} \quad \sigma_r(\vec{a}) = \text{StdDev}(\{\|\vec{a}_p\|\}_{|p|=r}).$$

It is clear that the smaller the values of α and σ , the better the radial symmetry is being preserved. In particular, the acceleration is exactly radially symmetric if and only if $\alpha = \sigma = 0$.

In Table 6 we compare the numerical results for these measures corresponding to different artificial viscosity models as well as the benchmark case $\mu = 1$. For simplicity, these do not include the FEM-based smoothness detector from Section 7.4, and we have set $\psi_0 = 1$. The left part of Table 6 clearly shows that, as expected, when the mesh is only perturbed in radial direction all models preserve radial symmetry up to machine precision. This is independent of the order of integration rule used. It is straightforward to see that (M4) is satisfied by all models in this case.

The situation is worse when there is a perturbation in the angular direction. In this case, Model 1 and Model 2 do not satisfy condition (M4), which explains why they give inferior results. Examining some of the computed acceleration fields, shown in Fig. 12, we see that the maximal angle deviation is worse in Model 1, while Model 2 exhibits a larger variation in the velocity magnitude. Both results are significantly worse than those for Model 3, which satisfies (M4). However, even in this case (or the simpler case $\mu = 1$), we see that we only get an approximate symmetry preservation. This is purely a discretization phenomenon: in the continuous case (M4) implies (S4). The results in Table 6 also indicate, that under-integration may help with symmetry preservation, though the difference is not very significant, except in the case $\mu = 1$.

Note that, one can prove that the acceleration produced with $\mu = 1$ and $k = 2$ will point in radial direction, i.e. will have $\alpha = 0$, even for polar meshes which are perturbed in both radial and angular directions. However, this does not mean that radial symmetry is preserved exactly, since $\sigma \neq 0$ as seen on the right of Table 6. Furthermore, this property is no longer true when the interior nodes are independently perturbed as in Fig. 8.

Finally, as mentioned earlier, Model 2 can be improved to satisfy (M4) by straightening of the zone edges using its associated parallelogram. This however was found to be sensitive to roundoff errors, and not to be beneficial in truly unstructured settings.

8.3. The Coggeshall adiabatic compression problem #2

In this example, we verify that property (M3) is satisfied using our FEM based smoothness detection algorithm from Section 7.4. We consider a variation of a problem originally described in [20]; in particular we apply the analytic solution described as problem #2. This problem describes an adiabatic compression (i.e. no shock waves generated) in one spatial coordinate, r . We run the problem in 2D x - y geometry on a quarter symmetry polar “ring” mesh with an inner radius $R_i = 0.1$ and an outer radius $R_o = 1.0$. We initialize the density, internal energy and velocity of the problem according to the solution provided in [20] and use a simple ideal gas equation of state with $\gamma = 5/3$. Specifically, we use the following initial conditions

$$\begin{aligned} \vec{v}(r, t = 0) &= -\frac{3r}{5} \vec{r}, \\ \rho(r, t = 0) &= \rho_0 r, \\ e(r, t = 0) &= \frac{3}{25} r^2. \end{aligned}$$

Since the initial velocity has components that are linear in x and y , property (M3) requires that the computed artificial viscous force be exactly zero. In Fig. 13 we show normalized plots of the zonal value $\vec{v}_z \cdot (\mathbf{S}_z \vec{v}_z)$, which represents the change in energy generated by the artificial viscosity, after one time step for two separate cases: one in which the FEM-based smoothness sensor ψ_0 is off (i.e. $\psi_0 = 1$ for all time) and one in which it is on. It is clear from the plots that in the absence of the ψ_0 term, the (M3) property is not satisfied and the artificial viscosity is doing work over the entire problem domain. When the ψ_0 term is on, property (M3) is satisfied exactly.

Table 6

Measures of radial symmetry preservation for a transformed polar mesh perturbed in radial direction only (left), or in angular direction only (right). Listed are the different artificial viscosity models from Section 7.3 against the order of the integration rule.

Model	$k = 2$		$k = 3$		Model	$k = 2$		$k = 3$	
	α	σ	α	σ		α	σ	α	σ
$\mu = 1$	2.6e-14	2.4e-13	1.8e-14	1.5e-13	$\mu = 1$	2.1e-14	0.00913	0.04167	0.00434
1	1.3e-14	5.2e-15	4.8e-15	3.8e-15	1	0.49434	0.00059	0.58381	0.00058
3	2.3e-13	2.6e-14	7.7e-14	1.4e-14	2	0.36655	0.00259	0.41049	0.00260
3	1.0e-14	8.3e-15	8.9e-15	5.4e-15	3	0.00361	0.00050	0.03859	0.00050

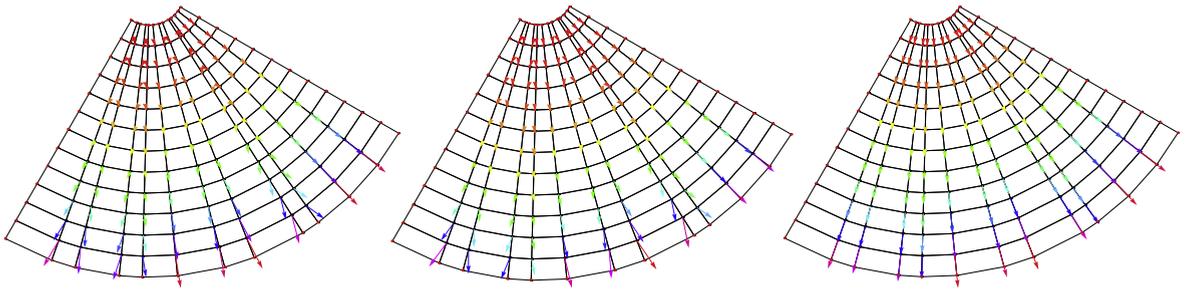


Fig. 12. Acceleration for the polar mesh perturbed in angular direction with $k = 2$ and using viscosity Model 1 (left), Model 2 (center) and Model 3 (right).

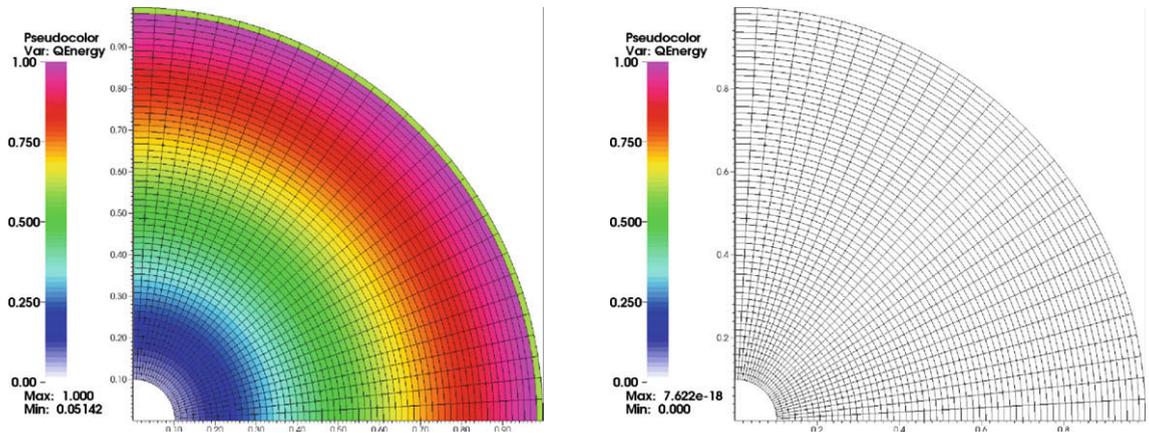


Fig. 13. Snapshot of Lagrangian mesh and pseudo-colors of artificial viscosity zonal “heating” for the 2D Coggeshall adiabatic compression problem using the tensor artificial viscosity *without* the ψ_0 term (left) and *with* the ψ_0 term (right). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

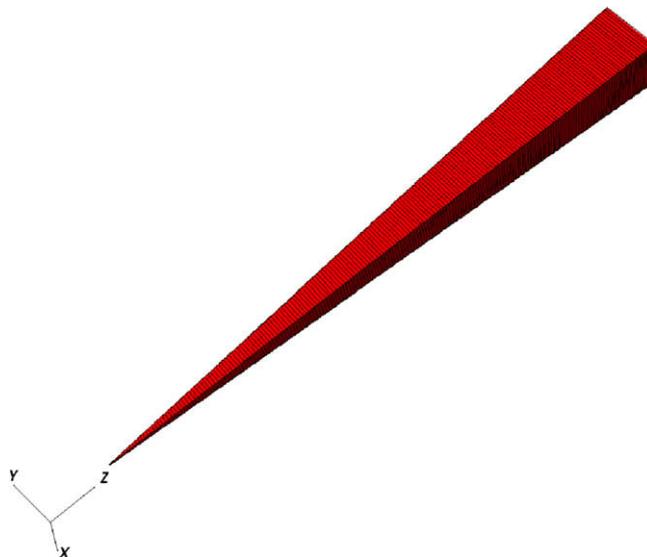


Fig. 14. Computational 3D “spike” mesh with 200 radial zones used for the spherical Noh test problem.

8.4. The Noh test problem

In these examples we consider the canonical Noh [2] implosion test problem on various mesh types. In each case, the problem domain consists of an ideal gas with $\gamma = 5/3$ and an initial density of $\rho_0 = 1$. The velocity of the grid is initialized to a radial vector pointing toward the origin, $\vec{v} = -\vec{r}$. The initial velocity directed at the origin will generate a stagnation shock wave that propagates radially outward and produces a peak post shock density of $\rho = 16$ for 2D problems and $\rho = 64$ for 3D problems.

8.4.1. Simple benchmark on a 3D “Spike” Mesh

We begin with a simple benchmark calculation. The purpose of this test is to verify the diffusive properties of the proposed FEM tensor artificial viscosity in a simple 1D context by comparing to some well established formulations. We consider the 3D spherical Noh implosion test problem on a 3D x - y - z “spike mesh” as shown in Fig. 14. The computational grid consists of 200 zones distributed along the z -axis over the range $r = [r_{min}, 1]$ where r_{min} is some small, non-zero value; this is required since the finite element method does not support degenerate zones (i.e. zones with coincident vertices). The initial

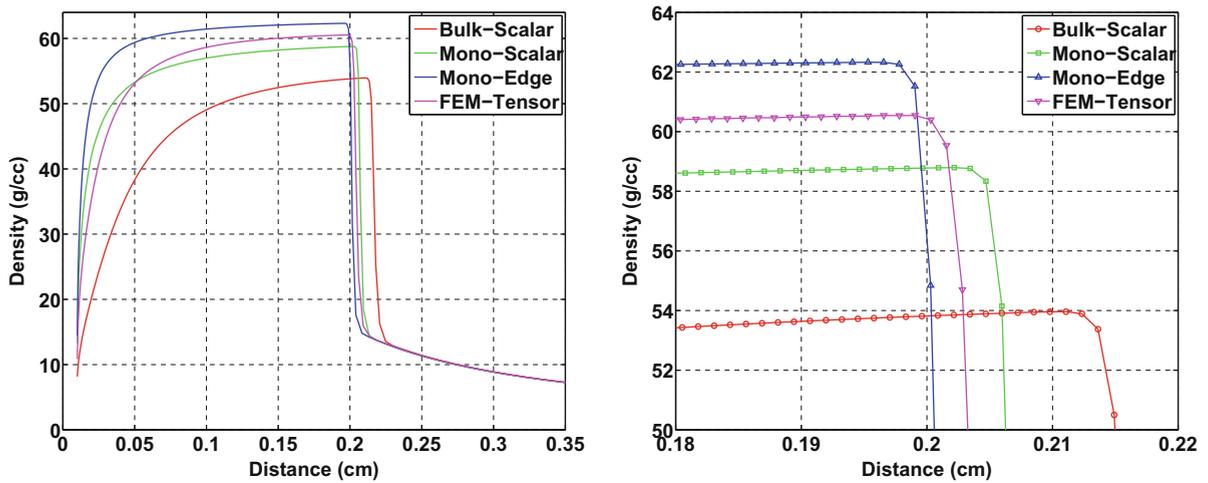


Fig. 15. Density versus distance at time $t = 0.6$ for the Noh test problem on a 3D x - y - z spike mesh using various artificial viscosity formulations (left) and a zoomed in view of the data at the shock front (right).

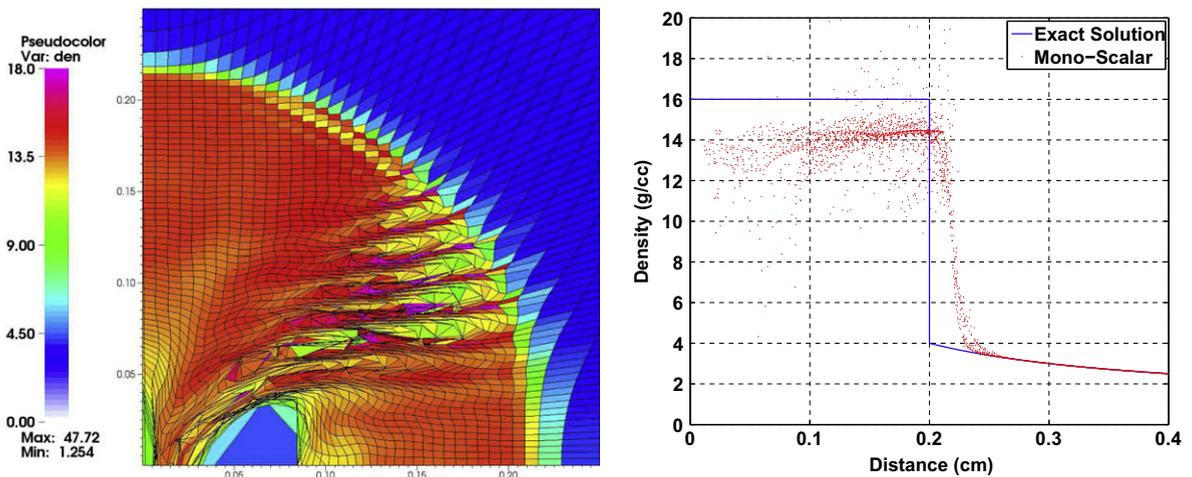


Fig. 16. Snapshot of Lagrangian mesh and density pseudo-color (left) and scatter plot of zone radius vs. zone density (right) at time $t = 0.6$ for the 2D Noh problem on a 40 by 80 initial grid using the monotonic-scalar artificial viscosity.

velocity directed at the origin will generate a stagnation shock wave that propagates radially outward and produces a peak post shock density of $\rho = 64$. We run the problem to a final time of $t = 0.6$ and compare values of the density using various artificial viscosity formulations.

We compare the FEM tensor artificial viscosity against three commonly used formulations: a monotonically limited scalar artificial viscosity (Mono-Scalar), a monotonically limited “edge” artificial viscosity (Mono-Edge), and a simple scalar “bulk” artificial viscosity (Bulk-Scalar). The so-called edge formulation is similar to that of [16] while the scalar monotonic artificial viscosity is due to [7] and a detailed description in 2D can be found in [13]. The bulk artificial viscosity is similar to the original Von Neumann–Richtmyer formulation; it is based on a cell centered value of the velocity divergence in every cell using local difference formulas based on the zone vertices, it does not invoke the use of a monotonic limiter. The cell centered divergence is scaled by a “characteristic length” of the zone to determine the velocity jump in the zone. Anecdotal evidence indicates that such a formulation is generally quite reliable in terms of symmetry preservation for general shock wave propagation over multi-dimensional grids. The main drawback to the Bulk-Scalar method is that it is too “diffusive,” implying that it gives up too much kinetic energy into irreversible shock heating by smearing the shock front over too many zones. It is precisely this limitation that has prompted the development of the monotonically limited artificial viscosity variants.

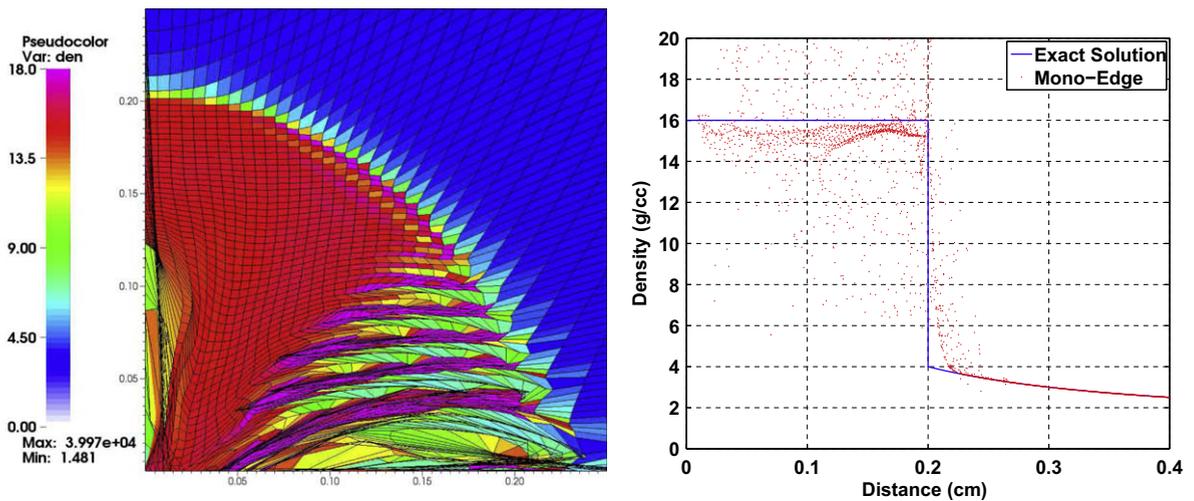


Fig. 17. Snapshot of Lagrangian mesh and density pseudo-color (left) and scatter plot of zone radius vs. zone density (right) at time $t = 0.6$ for the 2D Noh problem on a 40 by 80 initial grid using the monotonic-edge artificial viscosity.

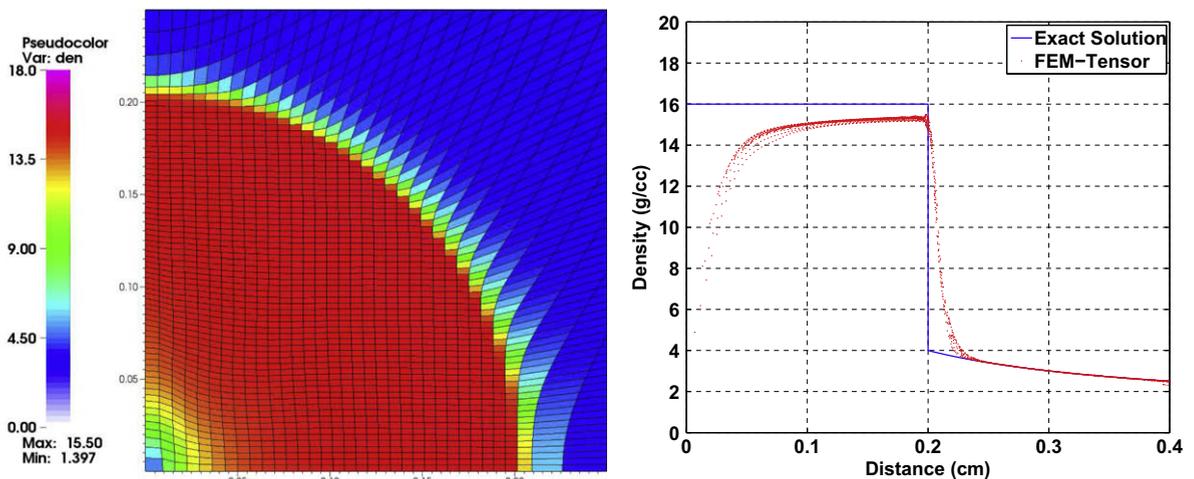


Fig. 18. Snapshot of Lagrangian mesh and density pseudo-color (left) and scatter plot of zone radius vs. zone density (right) at time $t = 0.6$ for the 2D Noh problem on a 40 by 80 initial grid using the FEM-tensor artificial viscosity.

The results for a “line-out” of the density field along the z -axis at the final time $t = 0.6$ using various artificial viscosities are shown in Fig. 15. Of all the formulations, the bulk scalar viscosity results in the lowest post-shock density, indicating its overly diffusive nature. The monotonic-scalar viscosity performs better with a higher post-shock density while the monotonic-edge viscosity performs the best for this particular test problem since the shock flow is perfectly aligned with the meshes. However, as we will see in the following sections, the edge viscosity does not perform very well when the shock

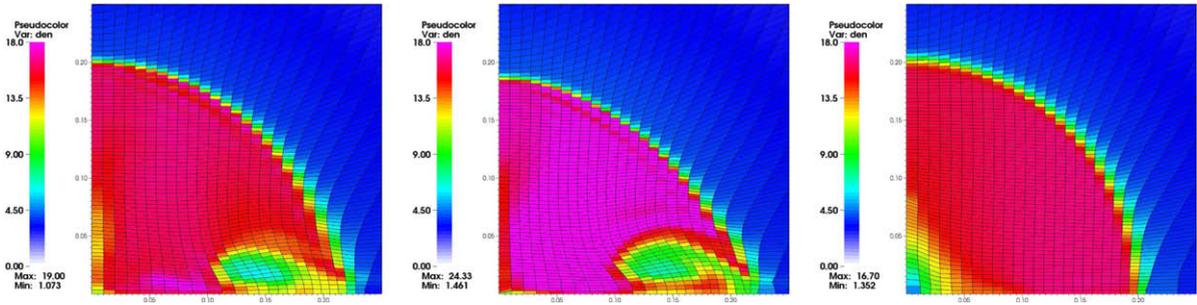


Fig. 19. Snapshots of ALE mesh and density pseudo-colors at time $t = 0.6$ for the 2D Noh problem on a 30 by 90 initial grid using the monotonic-scalar (left), monotonic-edge (center) and FEM-tensor (right) artificial viscosities. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

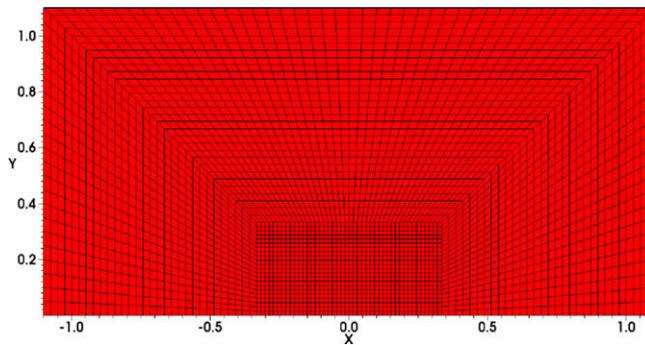


Fig. 20. Initial unstructured butterfly mesh used for the Sedov test problem.

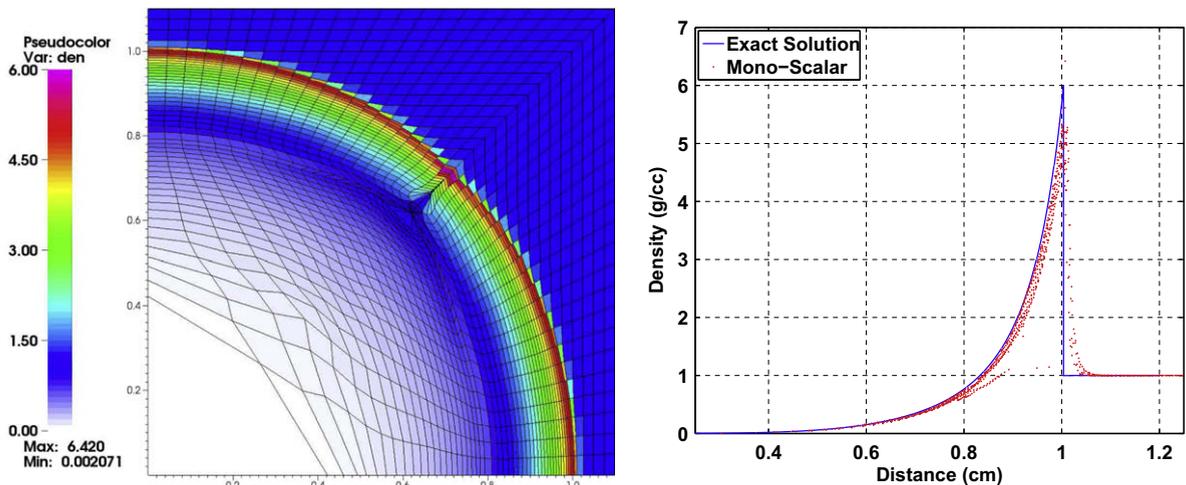


Fig. 21. Snapshot of Lagrangian mesh and density pseudo-color (left) and scatter plot of zone radius vs. zone density (right) at time $t = 1.0$ for the 2D Sedov problem on an unstructured grid using the monotonic-scalar artificial viscosity.

flow is not perfectly aligned with the mesh. In both cases, the FEM tensor artificial viscosity produces a higher post shock density than the monotonic-scalar, but not as high as the monotonic-edge viscosity.

8.4.2. Symmetry preservation

We now transition to the 2D variant of the Noh problem on an initially Cartesian x - y mesh to demonstrate the affect of artificial viscosity with respect to symmetry preservation on a mesh that is not aligned with the shock flow. In these examples, we make use of the tensor artificial viscosity Model 2 of Section 7. This particular problem is typically run using an initially uniform Cartesian mesh where each zone has unit aspect ratio (e.g. in [1]); In these examples, we introduce a more challenging variant, where the initial aspect ratio of each zone is not unity. We first consider a mesh with 40 zones in the x -direction and 80 zones in the y -direction, giving each cell an initial aspect ratio of 2.

In Figs. 16–18 we show plots of the Lagrangian mesh and pseudo-color plots of the density at time $t = 0.6$ for the case of the monotonic-scalar, monotonic-edge and the FEM tensor artificial viscosity Model 1. We also show scatter plots of the zone radius vs. the zone density and compare these to the analytic solution. The scatter plots provide a simple way of determining the quality of symmetry preservation. It is clear from the plots that the tensor viscosity using the Model 2 coefficient does a superior job of maintaining the symmetry of the Lagrangian mesh as well as the state variables, even on a grid with non-

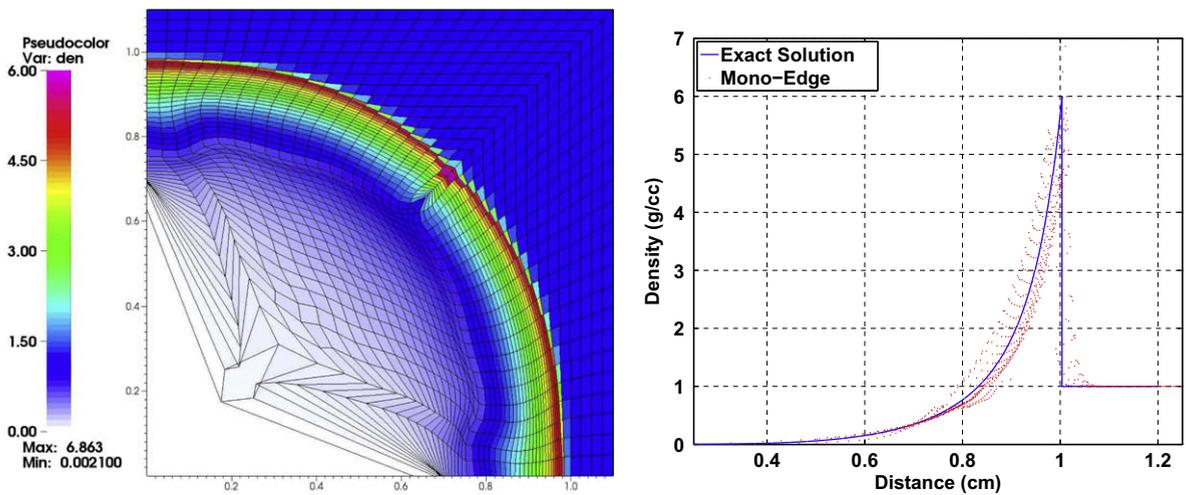


Fig. 22. Snapshot of Lagrangian mesh and density pseudo-color (left) and scatter plot of zone radius vs. zone density (right) at time $t = 1.0$ for the 2D Sedov problem on an unstructured grid using the monotonic-edge artificial viscosity.

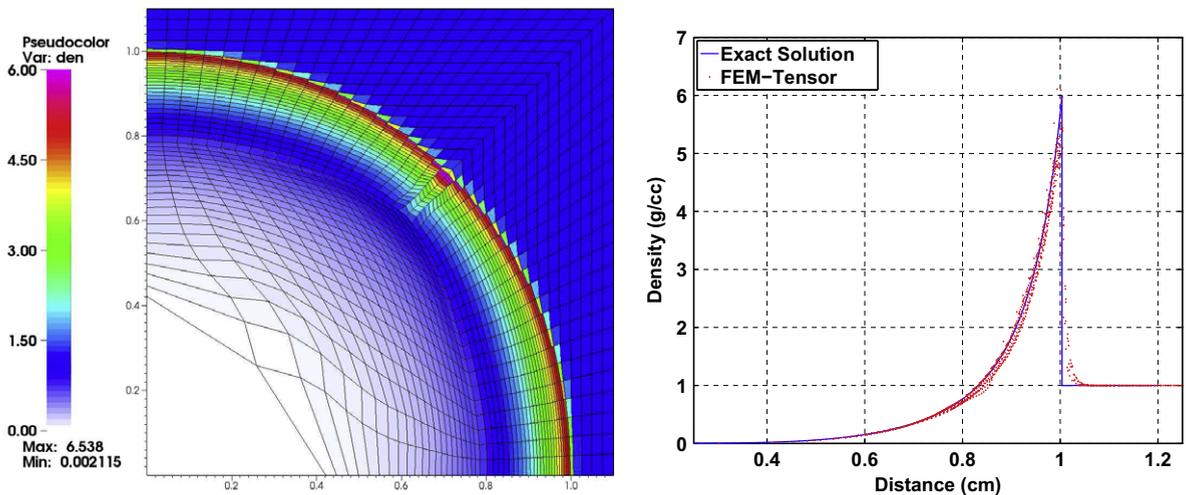


Fig. 23. Snapshot of Lagrangian mesh and density pseudo-color (left) and scatter plot of zone radius vs. zone density (right) at time $t = 1.0$ for the 2D Sedov problem on an unstructured grid using the FEM-tensor artificial viscosity.

uniform aspect ratio. We note that for this particular problem, Model 1 and Model 3 do a much better job at preserving symmetry than the monotonic-scalar and monotonic-edge viscosities, but not as well as Model 2.

Most modern hydro codes are of the arbitrary Lagrangian–Eulerian (ALE) [18] variety; where severe grid distortion is corrected by a mesh relaxation algorithm followed by a remapping of the state variables onto the new grid. However, such mesh relaxation approaches will not address the issue of symmetry preservation if the underlying methods used in the Lagrange

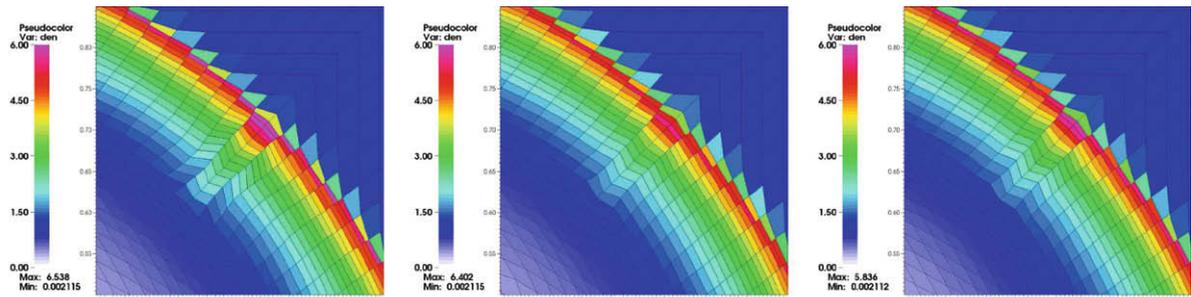


Fig. 24. Close-up view around reduced connectivity point of Lagrangian mesh and density pseudo-colors at time $t = 1.0$ for the 2D Sedov problem on an unstructured grid using the FEM-tensor artificial viscosity, with the Model 1 (left), Model 2 (center) and Model 3 (right) coefficients. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

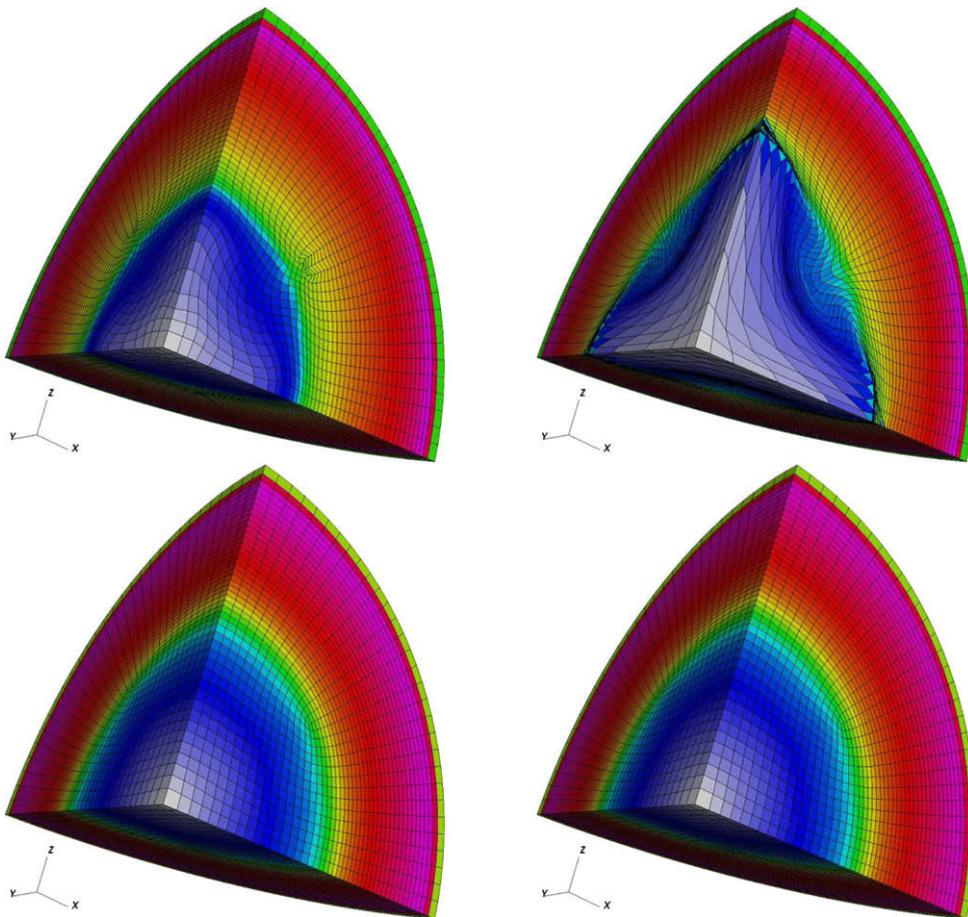


Fig. 25. Snapshots of Lagrangian mesh and density pseudo-colors at time $t = 0.65$ for the 3D spherically driven shock problem on an unstructured mesh using the monotonic-scalar (top-left), monotonic-edge (top-right), FEM-tensor $k = 2$ (bottom-left) and FEM-tensor $k = 3$ (bottom-right) artificial viscosities.

step do not respect the symmetry. To demonstrate this, we re-run the problem in an ALE fashion where at every cycle, the computational grid is relaxed via an iterative equipotential method followed by a remapping of the state variables to the new mesh. In addition, we now use a mesh with 30 zones in the x -direction and 90 zones in the y -direction, giving each cell an initial aspect ratio of 3. In Fig. 19 we show plots of the ALE mesh and pseudo-color plots of the density at time $t = 0.6$ for the case of the monotonic-scalar, monotonic-edge and the FEM-tensor artificial viscosity. In each case, the ALE algorithm does an excellent job of keeping the mesh uniform and free from spurious distortions; however, the symmetry in the underlying state variables is not preserved for the case of the monotonic-scalar and monotonic-edge viscosities. The FEM-tensor viscosity using the Model 2 coefficient does a superior job of maintaining the symmetry of the state variables.

8.5. The Sedov test problem

Here we investigate the Sedov [21] explosion test problem in planar x - y geometry. The problem consists of an ideal gas ($\gamma = 1.4$) with a delta function source of internal energy deposited at the origin. The sudden release of the energy will create an expanding shock wave, converting the initial internal energy into kinetic energy over time. Typically, this problem is run on a Cartesian mesh; we introduce a more challenging version where the mesh has an unstructured “butterfly” topology as shown in Fig. 20. The delta function energy source is approximated by setting the internal energy per mass variable in the zones to the left and right of the origin to a large value such that the total integrated energy over the problem domain is $1EU$; however, since we are only meshing half of the entire domain, we need to scale this number by a factor of $1/2$.

In Figs. 21–23 we show plots of the Lagrangian mesh and pseudo-color plots of the density at time $t = 1.0$ for the case of the monotonic-scalar, monotonic-edge and the FEM tensor artificial viscosity. We also show scatter plots of the zone radius vs. the zone density and compare these to the analytic solution. For the case of the monotonic-scalar and edge viscosities, the mesh gets highly distorted as the shock wave passes over the “reduced connectivity” point of the grid. As before, the FEM tensor artificial viscosity is better at preserving the radial symmetry of the problem, even when the underlying mesh is unstructured with sharp transitions. In Fig. 24, we show close up plots of the Lagrangian mesh and density pseudo-colors

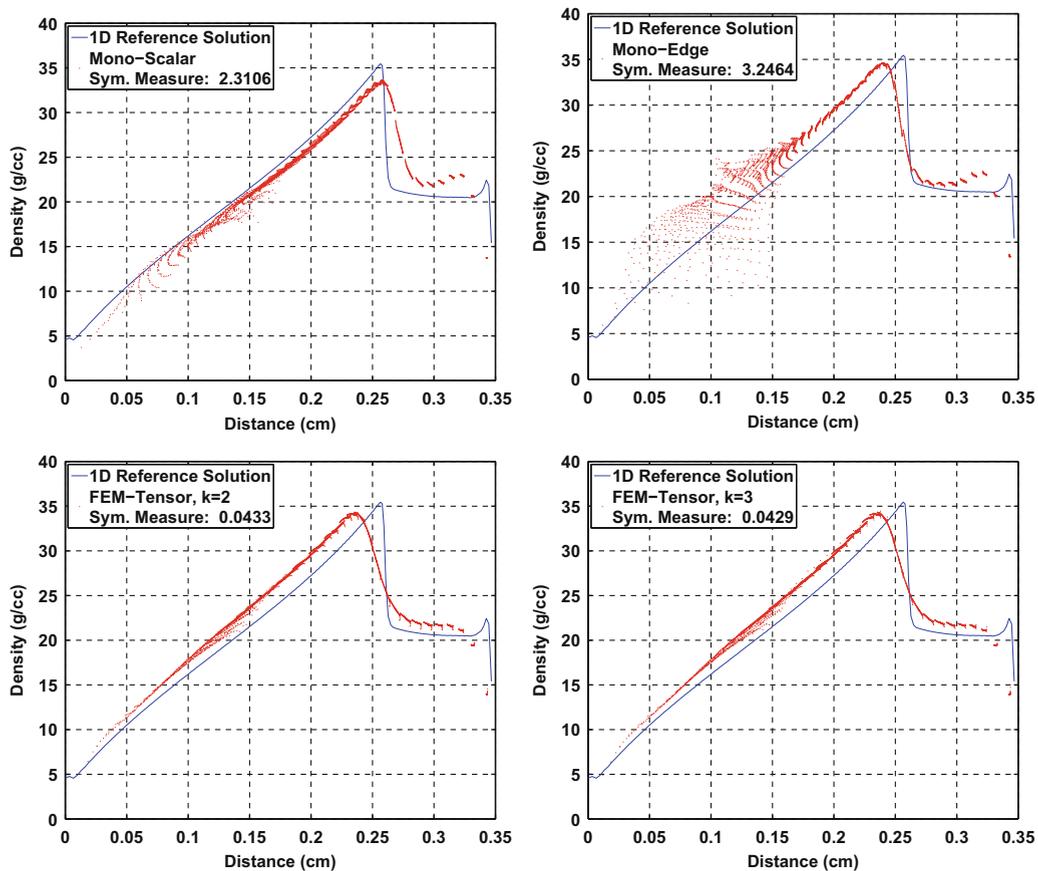


Fig. 26. Scatter plots of zone radius vs. zone density at $t = 0.65$ along with a single “symmetry measure” value for the 3D spherically driven shock problem on an unstructured mesh using the monotonic-scalar (top-left), monotonic-edge (top-right), FEM-tensor $k = 2$ (bottom-left) and FEM-tensor $k = 3$ (bottom-right) artificial viscosities.

around the reduced connectivity point for each of the tensor artificial viscosity models. Note that for this particular test problem, each model does a good job of propagating the shock wave over this mesh transition point, though there are some subtle differences.

8.6. Spherically driven shock wave on 3D unstructured mesh

In this problem we drive a spherical shock wave through an unstructured 3D octant mesh. The mesh has a so-called “butterfly” topology consisting of four separate mesh blocks stitched together at a “reduced connectivity” point. The sphere consists of a simple ideal gas with $\gamma = 5/3$ and an initial density of $\rho_0 = 1$. We drive the shock wave by applying a unit radial velocity boundary condition to the outermost spherical surface for all time. We run the problem to a final time of $t = 0.75$ which is enough time for the shock wave to bounce off of the origin as well as bounce off of the incoming piston boundary and return radially inward.

In Fig. 25 we show plots of the Lagrangian mesh and pseudo-color plots of the density at time $t = 0.65$ for the case of the monotonic-scalar, monotonic-edge and the tensor artificial viscosity using both $k = 2$ and $k = 3$. In Fig. 26 we show scatter plots of the zone radius vs. the zone density and compare these to a 1D reference solution. In addition, we compute a single number for each scatter plot data set (the “symmetry measure”) which corresponds to the maximal standard deviations in the clustering of the scatter plot values averaged over cluster sizes between 1 and 100 (note that the exterior of the domain has 45 “layers” of elements). At this point in time, the inwardly traveling shock wave has bounced from the origin and is now traveling radially outward. For the case of the monotonic-scalar and edge viscosities, we begin to notice significant deviations from spherical symmetry as the shock bounces from the origin and the Lagrangian mesh begins to develop spurious vorticity. In Fig. 27 we show plots of the Lagrangian mesh and pseudo-color plots of the density at time $t = 0.75$ and the corresponding scatter plots and symmetry measure values in Fig. 28. At this point in time, the outwardly traveling shock

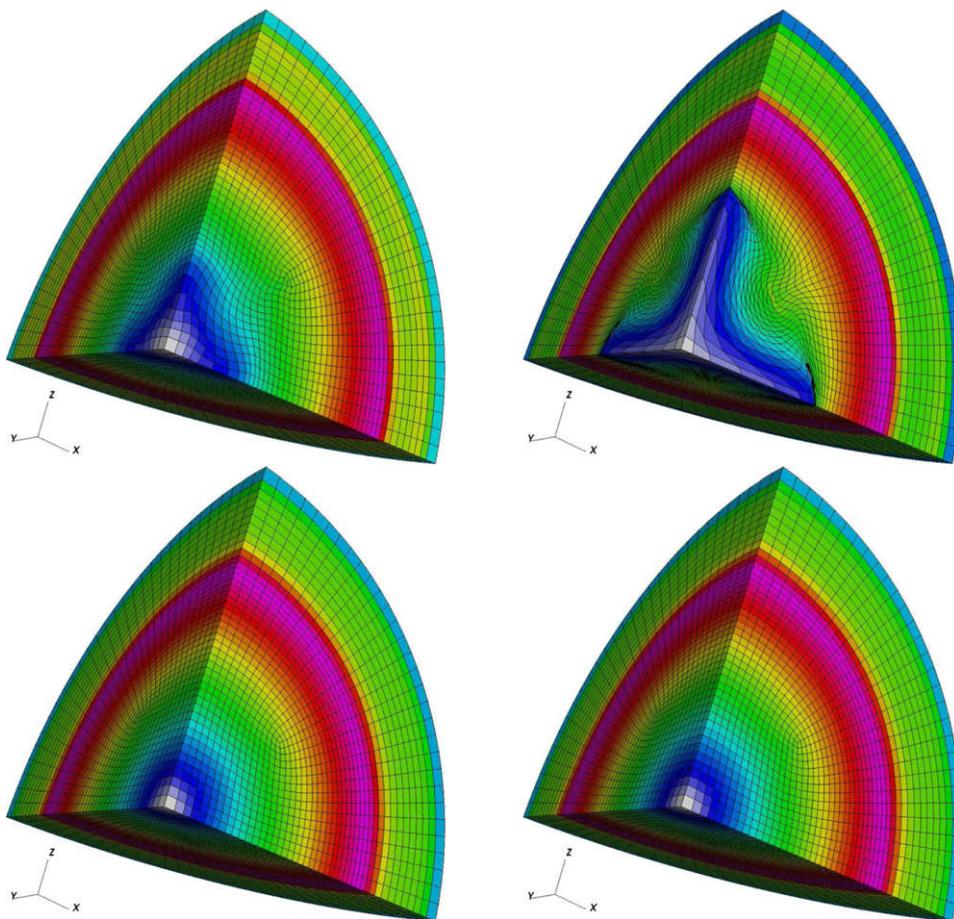


Fig. 27. Snapshots of Lagrangian mesh and density pseudo-colors at time $t = 0.75$ for the 3D spherically driven shock problem on an unstructured mesh using the monotonic-scalar (top-left), monotonic-edge (top-right), FEM-tensor $k = 2$ (bottom-left) and FEM-tensor $k = 3$ (bottom-right) artificial viscosities.

wave has bounced from the outermost driven boundary and is now traveling radially inward again. For the case of the monotonic scalar and edge viscosities, the deviations from spherical symmetry have increased substantially while both instances of the tensor viscosity do a much better job at maintaining the spherical symmetry of the problem.

8.7. The Saltzman piston test problem

The Saltzman piston is a 1D driven shock problem run on a distorted 2D mesh. In [1], the authors demonstrate how the mimetic finite difference tensor artificial viscosity can generate excellent results on this test problem. In this example, we

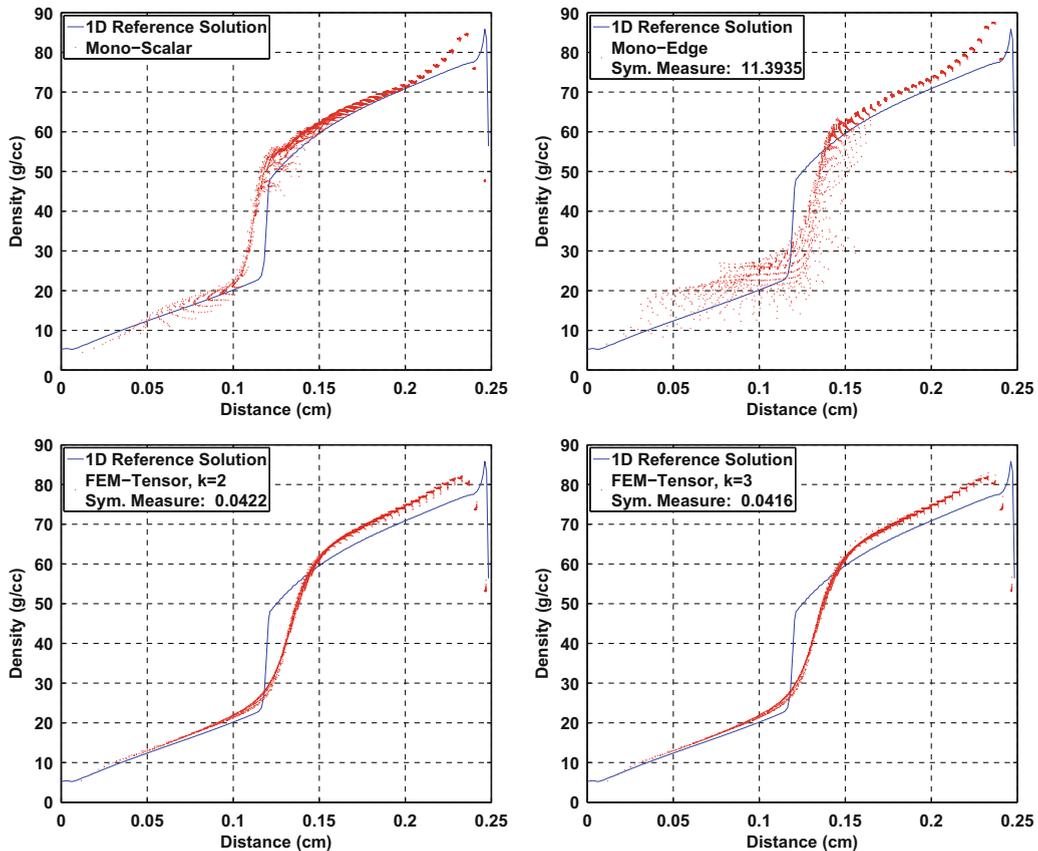


Fig. 28. Scatter plots of zone radius vs. zone density at $t = 0.75$ along with a single “symmetry measure” for the 3D spherically driven shock problem on an unstructured mesh using the monotonic-scalar (top-left), monotonic-edge (top-right), FEM-tensor $k = 2$ (bottom-left) and FEM-tensor $k = 3$ (bottom-right) artificial viscosities.

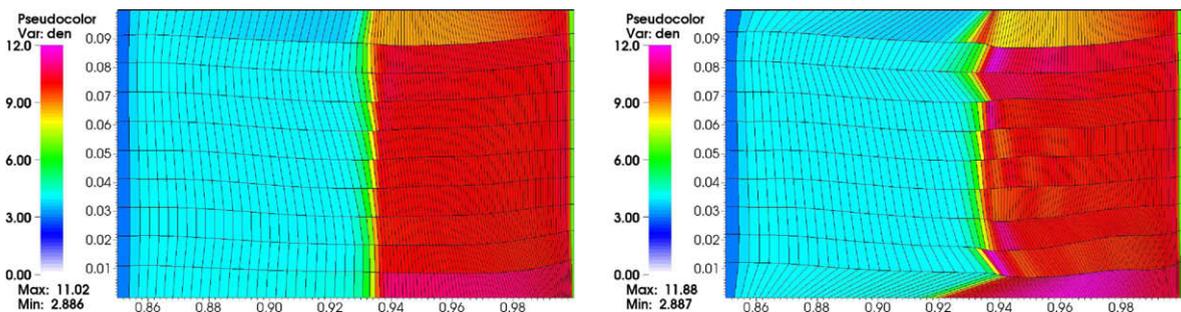


Fig. 29. Snapshots of Lagrangian mesh and density pseudo-colors at time $t = 0.85$ for the 2D Saltzman piston problem using the tensor artificial viscosity without a ψ_2 on the linear term (left) and with ψ_2 on the linear term (right).

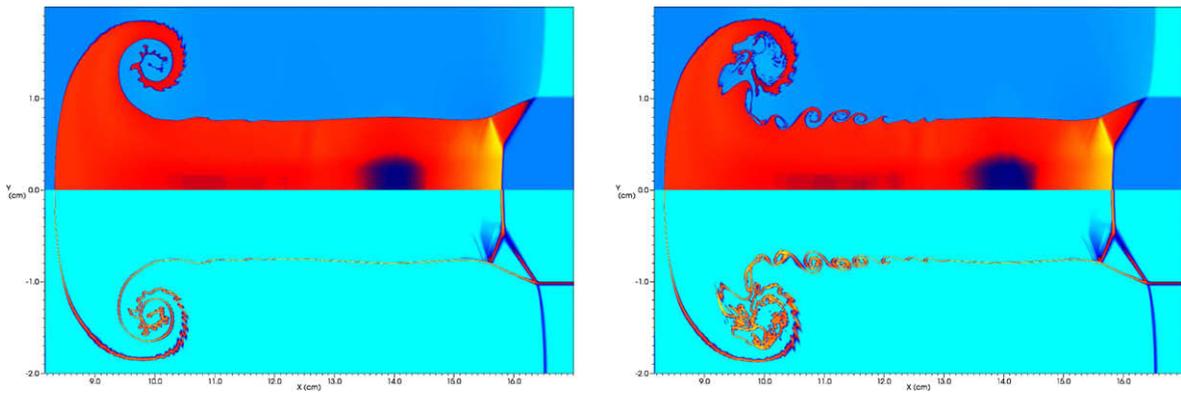


Fig. 30. Density (*top plane*) and log density gradient (*bottom plane*) plots for the shock driven Kelvin–Helmholtz instability problem using the tensor artificial viscosity *without* ψ_2 on the linear term (*left*) and *with* ψ_2 on the linear term (*right*).

run the same problem using our proposed FEM-based tensor artificial viscosity and point out an interesting feature concerning the ψ_2 measure which is applied to the linear term in the artificial viscosity coefficient. This coefficient is designed to satisfy the viscous force continuity property by forcing the linear coefficient to zero whenever the local vorticity dominates the local compression of a zone (for the case of perfect shear flow, the compression is exactly zero). In Fig. 29 we show plots of the Lagrangian mesh and pseudo-color plots of the density at time $t = 0.85$ (after the shock wave has bounced from the left wall) using the FEM tensor viscosity with and without the ψ_2 term. In the absence of ψ_2 (i.e. $\psi_2 = 1$ for all time), we obtain results that are of the same quality as those shown in [1]; however, when the ψ_2 coefficient is on, the solution generates spurious vorticity leading to subsequent mesh tangling.

This is because the artificial viscosity is not the sole source of vorticity error in this particular test problem [22]; the discretization of the pressure gradient term as well as the use of simple nodal masses in the momentum equation also generate spurious vorticity. In the absence of the ψ_2 term, the linear term of the tensor artificial viscosity is acting to suppress the spurious vorticity generated in the post shock region; which turns out to be a good thing for this particular test problem. However; we point out that in general, this is a very undesirable property since the tensor artificial viscosity will always act to counter any vorticity in a non-shocked region, whether it is spurious or physical. In the following Section 8.8, we give a specific example of this behavior. The fundamental reason for this is that without the ψ_2 term, the linear component of the tensor artificial viscosity does not satisfy the viscous force continuity property (S5) of Section 7.

8.8. Shock driven Kelvin–Helmholtz instability

The purpose of this example is to demonstrate the practical necessity of the ψ_2 coefficient which is applied to the linear term in the artificial viscosity coefficient. In this test problem, we propagate a Mach 1.5 planar shock wave normal to a two material interface (upper plane with density of ρ , lower plane with density of 2ρ). The difference in shock wave speeds in the two materials as well as the density gradient causes the interface between the two regions to be Kelvin–Helmholtz unstable [23]. In Fig. 30 we show pseudo-color plots of the density (upper plane) as well as the log of the density gradient (lower plane) using the FEM tensor viscosity with and without the ψ_2 term. Note that this is a full multi-material ALE calculation. Without ψ_2 , the interface remains fairly unperturbed. This is because the linear component of the artificial viscous force is acting in the opposite direction of the Kelvin–Helmholtz vorticity in the post-shock region of the problem. When the ψ_2 coefficient is on, viscous force continuity is restored in the post shock region and the tensor artificial viscous force no longer resists the motion of the physical instability.

9. Conclusions

We have derived and presented a finite element-based tensor artificial viscosity suitable for use in a 2D or 3D unstructured arbitrary Lagrangian–Eulerian (ALE) hydrodynamics code. This approach was shown to be a generalization of the mimetic tensor viscosity proposed by Campbell and Shashkov in the two-dimensional model case of unit viscosity. Several different finite element-based forms of the non-linear viscosity coefficient were proposed, based on theoretical guiding principles and employing measures of length scale, compressibility, vorticity and smoothness of the velocity field. We demonstrated that each of the above models outperforms the rest in specific simulations, and while none of them emerged as a clear winner, we generally recommend the use of Model 1 and Model 2 in practical computations.

Acknowledgments

The authors would like to acknowledge the input of Panayot Vassilevski, Misha Shashkov, Andy Barlow, Bop Tipton, Tom McAbee and Brian Pudliner.

References

- [1] J.C. Campbell, M.J. Shashkov, A tensor artificial viscosity using a mimetic finite difference algorithm, *J. Comput. Phys.* 172 (2001) 739–765.
- [2] W.F. Noh, Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux, *J. Comput. Phys.* 72 (1) (1987) 78–120.
- [3] E.J. Caramana, M.J. Shashkov, Elimination of artificial grid distortion and hourglass-type motions by means of Lagrangian subzonal masses and pressures, *J. Comput. Phys.* 142 (2) (1998) 521–561.
- [4] J. VonNeumann, R.D. Richtmyer, A method for the numerical calculation of hydrodynamic shocks, *J. Appl. Phys.* 21 (1950) 232–237.
- [5] B. VanLeer, Towards the ultimate conservative difference scheme, *J. Comput. Phys.* 32 (1) (1979) 101–136.
- [6] R. Landshoff, A numerical method for treating fluid flow in the presence of shocks, Technical Report LA-1930, Los Alamos National Laboratory, 1955.
- [7] R.B. Christensen, Godunov methods on a staggered mesh – an improved artificial viscosity, Technical Report UCRL-JC-105269, Lawrence Livermore National Laboratory, 1991.
- [8] R. Loubere, On the effect of the different limiters for the tensor artificial viscosity for the compatible lagrangian hydrodynamics scheme, Technical report, Theoretical Division, T-7, Los Alamos National Laboratory, 2005.
- [9] J.C. Campbell, J.M. Hyman, M. Shashkov, Mimetic finite difference operators for second-order tensors on unstructured grids, *Comput. Math. Appl.* 44 (2002) 157–173.
- [10] D.E. Burton, Multidimensional discretizations of conservation laws for unstructured polyhedral grids, Technical Report UCRL-JC-118306, Lawrence Livermore National Laboratory, 1994.
- [11] 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.
- [12] F.H. Harlow, A.A. Amsden, Fluid dynamics: a LASL monograph, Technical Report LA-4700, Los Alamos Scientific Laboratory, 1971.
- [13] R. Tipton, CALE Lagrange step, Technical report, Lawrence Livermore National Laboratory, Unpublished, October 1990.
- [14] L. Margolin, M. Shashkov, Using a curvilinear grid to construct symmetry-preserving discretizations for Lagrangian gas dynamics, *J. Comput. Phys.* 149 (1999) 389–417.
- [15] A.J. Barlow, A compatible finite element multi-material ALE hydrodynamics algorithm, *Int. J. Numer. Meth. Fluids* 56 (2008) 953–964.
- [16] E.J. Caramana, M.J. Shashkov, P.P. Whalen, Formulations of artificial viscosity for multi-dimensional shock wave computations, *J. Comput. Phys.* 144 (1998) 70–97.
- [17] J.M. Owen, A tensor artificial viscosity for SPH, *J. Comput. Phys.* 201 (2) (2004) 601–629.
- [18] R.M. Darlington, Large eddy simulation of Rayleigh–Taylor instability using the arbitrary Lagrangian–Eulerian method, Technical Report UCRL-LR-136160, Lawrence Livermore National Laboratory, 1999.
- [19] R.M. Darlington, T.L. McAbee, G. Rodrigue, A study of ALE simulations of Rayleigh–Taylor instability, *Comp. Phys. Comm.* 135 (1) (2001) 58–73.
- [20] S.V. Coggshall, Analytic solutions of hydrodynamics equations, *Phys. Fluids A* 3 (5) (1991) 757–769.
- [21] L.I. Sedov, *Similarity and Dimensional Methods in Mechanics*, 10th ed., Springer, 1993.
- [22] J.K. Dukowicz, B.J. Meltz, Vorticity errors in multidimensional lagrangian codes, *J. Comput. Phys.* 99 (1) (1992) 115–134.
- [23] J.A. Greenough, W.Y. Crutchfield, C.A. Rendleman, Numerical simulation of a wave-guide mixing layer on a Cray C-90, in: 26th American Institute of Aeronautics and Astronautics Fluid Dynamics Conference, Number UCRL-JC-118610, San Diego, CA, June 1995.