

MONOTONICITY-PRESERVING FINITE ELEMENT SCHEMES WITH ADAPTIVE MESH REFINEMENT FOR HYPERBOLIC PROBLEMS

JESÚS BONILLA^{1,2} AND SANTIAGO BADIA^{2,3}

ABSTRACT. This work is focused on the extension and assessment of the monotonicity-preserving scheme in [3] and the local bounds preserving scheme in [5] to hierarchical octree adaptive mesh refinement (AMR). Whereas the former can readily be used on this kind of meshes, the latter requires some modifications. A key question that we want to answer in this work is whether to move from a linear to a nonlinear stabilization mechanism pays the price when combined with shock-adapted meshes. Whereas nonlinear (or shock-capturing) stabilization leads to improved accuracy compared to linear schemes, it also negatively hinders nonlinear convergence, increasing computational cost. We compare linear and nonlinear schemes in terms of the required computational time versus accuracy for several steady benchmark problems. Numerical results indicate that, in general, nonlinear schemes can be cost-effective for sufficiently refined meshes. Besides, it is also observed that it is better to refine further around shocks rather than use sharper shock capturing terms, which usually yield stiffer nonlinear problems. In addition, a new refinement criterion has been proposed. The proposed criterion is based on the graph Laplacian used in the definition of the stabilization method. Numerical results show that this shock detector performs better than the well-known Kelly estimator for problems with shocks or discontinuities.

Keywords: Adaptive mesh refinement, Shock capturing, Euler equations, Hyperbolic problems, Discrete maximum principle

CONTENTS

1. Introduction	2
2. Preliminaries	3
2.1. Continuous problem	3
2.2. Discretization	3
2.3. Stability properties	5
3. Nonlinear stabilization	6
3.1. Differentiable stabilization	10
4. Adaptive mesh refinement	12
4.1. Error estimators	12
4.2. Refinement strategy	13
5. Nonlinear solver	13
6. Numerical results	14
6.1. Convergence	15
6.2. Linear discontinuity	16
6.3. Circular discontinuity	16
6.4. Compression corner	18
6.5. Reflected shock	22
7. Conclusions	23
Acknowledgments	23
References	25

¹ Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain.

² Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE), Esteve Terradas 5, 08860 Castelldefels, Spain.

³ School of Mathematics, Monash University, Clayton, Victoria, 3800, Australia.

1. INTRODUCTION

Natural phenomena can develop shock waves in different scenarios. A classical example is the shock wave generated by an object traveling faster than sound. The numerical modeling of problems with shocks is still a challenge, especially when the admissible physical solution has some physical constraints, e.g., positivity or non-negativity, that must be preserved at the discrete level to have well-posedness; E.g., the fluid density and temperature are positive quantities in a compressible flow.

Several numerical schemes have been proposed so far to approximate this kind of problems by combining finite volume methods (FVM) or discontinuous Galerkin (dG) finite elements (FEs) for space discretization with explicit time integrators (see [19, 23, 42, 57]). Explicit time integrators are only stable under a Courant-Friedrichs-Levy (CFL) restriction over the time step size, which implies to capture all time scales. Thus, explicit methods are not suitable for problems in which the smallest time scales are not of interest. For instance, the fastest time scales at a confined plasma in a nuclear fusion reactor are not of engineering interest whereas explicit time integration is unaffordable in practical simulations [32].

Implicit monotonicity-preserving (or at least positivity-preserving) methods are still scarce. As proved by Godunov [25], linear monotonicity-preserving schemes can be at most first-order accurate. For scalar problems (and under some mesh restrictions), Burman and Ern [18], Barrenechea and co-workers [12, 13], Kuzmin and co-workers [35, 38, 44], and Badia and Hierro [6, 7] have proposed nonlinear schemes that preserve monotonicity and can presumably attain higher order accuracy.¹ However, these properties come at the cost of solving a very stiff nonlinear problem [33]. The authors [3, 4] have proposed differentiable schemes that improve the nonlinear convergence behavior of previous methods.

For hyperbolic systems of equations, numerical methods are less well developed. For explicit time integration, Guermond and Popov [28] have recently proposed a continuous Galerkin (cG) FE scheme that preserves positivity of density and energy under certain CFL-like condition. More recently, Kuzmin [34] has extended the previous work to monolithic convex limiting. This allows one to use implicit time integration while preserving positivity. Another approach is flux corrected transport (FCT) [38, 45]. The schemes in [47, 48] combine the diffusion operators in FCT with novel shock-detection techniques to obtain a nonlinear monolithic scheme. Those methods have been shown experimentally to be robust, but lack of a theoretical analysis. Besides, this strategy also yields very stiff nonlinear problems. Differentiable schemes for compressible flows have been proposed in [5] to alleviate (but not eliminate) this problem.

Shocks are non-smooth and localized and thus suitable for AMR [21, 58]. AMR allows one to increase the mesh resolution only in the vicinity of shocks or discontinuities. In brief, the AMR process can be divided into two main ingredients. On the one hand, to estimate the error at each element. On the other hand, to decide which elements need to be refined or coarsened. This iterative process provides a mesh locally adapted to the features of the problem at hand. As a result, it is a nonlinear approximation scheme which tries to minimize the error for a target computational cost. In some situations, the optimal order of convergence can be achieved even for solutions with limited regularity using AMR [21], whereas convergence is limited by the regularity of the solution for uniform mesh refinements.

In this context, a key question is whether it is computationally more effective to consider a nonlinear high-order scheme (with the nonlinear convergence issues) or a cheaper linear (first-order) scheme in a more refined mesh. The motivation of this work is to shed light on this issue. First, we adapt the schemes developed in [3, 5] to hierarchical octree AMR [9, 55]. Next, we propose a refinement criterium that relies on information already present in the stabilization technique; nonlinear stabilization methods include a shock detector to activate the artificial diffusion only close to discontinuities. We propose to use a modification of the shock detector in [3] to drive the AMR process.

This paper is structured as follows. First, we introduce the problem, its discretization, and monotonicity properties for scalar problems and hyperbolic systems in Sect. 2. Then, the stabilization

¹In this work, schemes with nonlinear stabilization are also referred to as high-order and linear stabilization schemes as low or first-order.

techniques are introduced in Sect. 3. Sect. 4 is devoted to the AMR strategy. We introduce the non-linear solvers in Sect. 5. Finally, we show numerical experiments in Sect. 6 and draw some conclusions in Sect. 7.

2. PRELIMINARIES

2.1. Continuous problem. Let us consider an open bounded and connected domain, $\Omega \subset \mathbb{R}^d$, where d is the number of spatial dimensions. Let $\partial\Omega$ be the Lipschitz continuous boundary of Ω . The conservative form of a first order hyperbolic problem reads

$$\begin{cases} \partial_t \mathbf{u} + \nabla \cdot \mathbf{f}(\mathbf{u}) = \mathbf{g}, & \text{in } \Omega \times (0, T], \\ u^\beta(x, t) = \bar{u}^\beta(x, t), & \text{on } \Gamma_{\text{in}}^\beta \times (0, T], \beta = 1, \dots, m, \\ \mathbf{u}(x, 0) = \mathbf{u}_0(x), & x \in \Omega, \end{cases} \quad (1)$$

where $\mathbf{u} = \{u^\beta\}_{\beta=1}^m$ are $m \geq 1$ conserved variables, \mathbf{f} is the physical flux, $\bar{u}^\beta(x, t)$ are the boundary values for the β th-component of \mathbf{u} , $\mathbf{u}_0(x)$ are the initial conditions, and $\mathbf{g}(x, t)$ is a function defining the body forces. Note that the flux, $\mathbf{f} : \mathbb{R}^m \rightarrow \mathbb{R}^{m \times d}$, is composed of $\mathbf{f} = \{\mathbf{f}_i\}_{i=1}^d$, where $\mathbf{f}_i : \mathbb{R}^m \rightarrow \mathbb{R}^m$ is the flux in the i th spatial direction. We denote by $\mathbf{f}' : \mathbb{R}^m \rightarrow \mathbb{R}^{m \times m \times d}$ the flux Jacobian. Let $\mathbf{n} \in \mathbb{R}^d$ be any direction vector. Since the system is hyperbolic, the flux Jacobian in any direction is diagonalizable and has only real eigenvalues, i.e., $\mathbf{f}'(\mathbf{u}) \cdot \mathbf{n} = \sum_{i=1}^d \mathbf{f}'_i(\mathbf{u}) n_i$ is diagonalizable with real eigenvalues $\{\lambda_\beta\}_{\beta=1}^m$. These eigenvalues might have different multiplicities and different signs. Hence, for a given direction \mathbf{n} , each characteristic variable might be convected forward (along \mathbf{n}) or backwards (along $-\mathbf{n}$). Therefore, it is convenient to define inflow and outflow boundaries for each component. The inflow boundary for component β is defined as $\Gamma_{\text{in}}^\beta \doteq \{\mathbf{x} \in \partial\Omega : \lambda_\beta(\mathbf{f}'(\mathbf{u}) \cdot \mathbf{n}_{\partial\Omega}) \leq 0\}$, where $\mathbf{n}_{\partial\Omega}$ is the unit outward normal to the boundary and λ_β is the β th-eigenvalue of the flux Jacobian. We define the outflow boundary as $\Gamma_{\text{out}}^\beta \doteq \partial\Omega \setminus \Gamma_{\text{in}}^\beta$. We refer the reader to [23, 29, 57] for a detailed discussion on boundary conditions for hyperbolic problems. In the present study, we will also consider the steady counterpart of (1), which is obtained by dropping the time derivative term and the initial conditions.

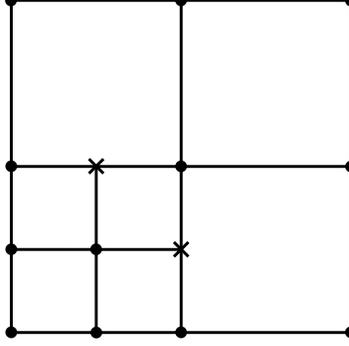
In this work, we work with both scalar convection equations and Euler equations. Taking $m = 1$ and $\mathbf{f}(u) \doteq \mathbf{v}u$ with \mathbf{v} a divergence-free convection field, we recover the well known scalar transport problem. On the other hand, Euler equations for ideal gases are recovered by defining $m = d + 2$ and

$$\mathbf{u} \doteq \begin{pmatrix} \rho \\ \mathbf{m} \\ \rho E \end{pmatrix}, \quad \mathbf{f} \doteq \begin{pmatrix} \mathbf{m} \\ \mathbf{m} \otimes \mathbf{v} + p\mathbb{I} \\ \mathbf{v}(\rho E + p) \end{pmatrix}, \quad \text{and} \quad \mathbf{g} \doteq \begin{pmatrix} 0 \\ \mathbf{b} \\ \mathbf{b} \cdot \mathbf{v} + r \end{pmatrix},$$

where ρ is the density, E is the total energy, p is the pressure, $\mathbf{m} = \{m_1, \dots, m_d\}$, where $m_i = \rho v_i$, is the momentum, $\mathbf{v} = \{v_1, \dots, v_d\}$ is the velocity, $\mathbf{b} = \{b_1, \dots, b_d\}$ are the body forces, r is an energy source term per unit mass, and \mathbb{I} is the identity matrix of dimension $d \times d$. In addition, the system is equipped with the ideal gas equation of state $p = (\gamma - 1)\rho\iota$, where $\iota = E - \frac{1}{2}\|\mathbf{v}\|^2$ is the internal energy and γ is the adiabatic index.

2.2. Discretization. The discretization used in this work is able to adapt its local size to the features of the problem at hand. In particular, it is a hierarchically refined octree-based hexahedral mesh [55]. This type of discretizations are constructed hierarchically. At every step of the refinement process, marked cells are refined into four (eight) cells in 2D (3D). The adaptation of the mesh to the problem at hand is achieved by only marking for refining a targeted amount of cells. This results in a mesh with different refinement levels at different regions. *Hanging* nodes appear at the interface between cells at different refinement levels. These are nodes that only belong to the cells at a higher refinement level (see Fig. 1). In our case, the meshes used are *2:1 balanced*. This restriction implies that there can only be a difference of one refinement level between neighboring cells. This restriction is a trade-off between implementation complexity and performance gain that has been adopted by many AMR codes [55].

Hanging nodes need to be treated carefully in the case of working with conforming FE discretizations. Otherwise, associating a regular degree of freedom (DOF) to a hanging node may lead to discontinuities

FIGURE 1. Example of a mesh with *hanging* nodes.

in the approximated solution. To preserve continuity of the FE space, *hanging DOFs* values are not included in the assembled system of equations but obtained by interpolating the values of the neighboring *regular* DOFs. For more details in the definitions of these conformity constraints we refer the reader to [8–10].

Let \mathcal{T}_h be a hierarchical octree-based partition of Ω . Consider a Lagrangian (nodal) FE space on top of this mesh. The set of all nodes in the FE space is represented with $\tilde{\mathcal{N}}_h$. For every node $i \in \tilde{\mathcal{N}}_h$, \mathbf{x}_i stands for the node coordinates. We can split $\tilde{\mathcal{N}}_h$ into two subsets, namely the set of hanging nodes \mathcal{N}_h^{hg} and the set of conforming nodes $\mathcal{N}_h \doteq \tilde{\mathcal{N}}_h \setminus \mathcal{N}_h^{hg}$. We denote by $N \doteq \text{card}(\mathcal{N}_h)$ the total number of conforming nodes. The set of nodes belonging to a particular element $K \in \mathcal{T}_h$ is defined as $\mathcal{N}_h(K) \doteq \{i \in \mathcal{N}_h : \mathbf{x}_i \in K\}$. Moreover, Ω_i is the macroelement composed by the union of elements that contain node i , i.e., $\Omega_i \doteq \bigcup_{K \in \mathcal{T}_h, \mathbf{x}_i \in K} K$. To simplify the discussion below, we abuse notation and use i for both the node and its associated index.

We restrict the present work to first order FEs and define the FE space as follows. We define $\mathbf{V}_h \doteq \{\mathbf{v}_h \in (\mathcal{C}^0(\Omega))^m : \mathbf{v}_h|_K \in (Q_1(K))^m \forall K \in \mathcal{T}_h\}$, where $Q_1(K)$ is the space of polynomials of partial degree less than or equal to one. Furthermore, we define the space $\mathbf{V}_{h0} \doteq \{\mathbf{v}_h \in \mathbf{V}_h : \mathbf{v}_h(\mathbf{x}) = 0 \forall \mathbf{x} \in \Gamma_{\text{in}}\}$. The functions $\mathbf{v}_h \in \mathbf{V}_h$ can be constructed as a linear combination of the basis $\{\varphi_i\}_{i \in \tilde{\mathcal{N}}_h}$ and nodal values \mathbf{v}_i , where φ_i is the shape function associated to the node i . Hence, $\mathbf{v}_h = \sum_{i \in \tilde{\mathcal{N}}_h} \varphi_i \mathbf{v}_i$.

We use standard notation for Sobolev spaces. The $L^2(\omega)$ scalar product is denoted by $(\cdot, \cdot)_\omega$ for $\omega \subset \Omega$. However, we omit the subscript for $\omega \equiv \Omega$. The L^2 norm is denoted by $\|\cdot\|$.

The method of lines is applied in combination with the FE spaces described above for the spatial discretization. We approximate the solution $\mathbf{u} \approx \mathbf{u}_h = \sum_{i \in \tilde{\mathcal{N}}_h} \varphi_i \mathbf{u}_i$. In addition, we make use of the group–FEM approximation [24]. Hence, fluxes are discretized in the same FE space as the unknown, i.e. $\mathbf{f} \approx \mathbf{f}_h = \sum_{i \in \tilde{\mathcal{N}}_h} \varphi_i \mathbf{f}(\mathbf{u}_i)$. For simplicity in the exposition, we use the Backward Euler (BE) scheme for the time discretization; higher order time discretizations can be achieved using strong stability preserving (SSP)–Runge Kutta (RK) methods (see [26]). In the latter case, a CFL-like condition must be satisfied to enjoy the monotonicity properties in Sect. 2.3 (see [37, 41]).

The semi-discrete Galerkin FE approximation of the weak form of (1) reads: find $\mathbf{u}_h \in \mathbf{V}_h$ such that $u_h^\beta = \bar{u}_h^\beta$ on Γ_{in}^β , $\mathbf{u}_h = \mathbf{u}_{0h}$ at $t = 0$, and

$$(\partial_t \mathbf{u}_h, \mathbf{v}_h) - (\mathbf{u}_h, \mathbf{f}'_h(\mathbf{u}_h) : \nabla \mathbf{v}_h) + (\mathbf{u}_h, \mathbf{n}_{\Gamma_{\text{out}}} \cdot \mathbf{f}'_h(\mathbf{u}_h) \mathbf{v}_h)_{\Gamma_{\text{out}}} = (\mathbf{g}, \mathbf{v}_h), \text{ for all } \mathbf{v}_h \in \mathbf{V}_{h0},$$

where \bar{u}_h^β and \mathbf{u}_{0h} are admissible FE approximations of the boundary and initial conditions \bar{u}^β and \mathbf{u}_0 . In this context, we consider admissible any approximation that satisfies the maximum principle, i.e., it does not introduce new extrema. Notice that boundary conditions are strongly imposed. For transonic, or complex problems, this strategy might lead to convergence issues. However, in the present paper we use this strategy for the sake of simplicity. As previously mentioned, we refer the reader to [23, 29, 57] for a detailed discussion on boundary conditions for hyperbolic problems. Note that the double contraction is applied as $\mathbf{f}'_h(\mathbf{u}_h) : \nabla \mathbf{v}_h = \sum_{k, \gamma} \mathbf{f}'_h(\mathbf{u}_h)_k^{\beta \gamma} \mathbf{v}_{h, \gamma, k}$.

As commented above, we need to apply constraints to all hanging DOFs to keep conformity. The value of the hanging DOF needs to be equal to the value of the interpolation of the unknown at the neighboring coarser elements. That is, given $i \in \mathcal{N}_h^{hg}$ and one of its neighboring (coarse) FE $K \in \mathcal{T}_h$, $\mathbf{v}_i = \sum_{j \in \mathcal{N}_h(K)} \varphi_j(\mathbf{x}_i) \mathbf{v}_j$. In general, we will represent this constraint with $\mathbf{v}_i = \sum_{j \in \overline{\mathcal{M}}(i)} C_{ij} \mathbf{v}_j$, where $\overline{\mathcal{M}}(i)$ is the set of DOFs constraining DOF i , and $C_{ij} \doteq \varphi_j(\mathbf{x}_i)$. It is also useful to define $\mathcal{M}(i)$, which is the set of DOFs constraint by i . For details of the implementation of this kind of constraints see [8–10].

Finally, to obtain the fully discrete problem, we consider a partition of the time domain $(0, T]$ into n^{ts} sub-intervals $(t^n, t^{n+1}]$ of length Δt_{n+1} . Then, at every time step $n = 0, \dots, n^{ts} - 1$, the discrete problem consists in solving

$$\mathbf{M} \delta_t \mathbf{U}^{n+1} + \mathbf{K} \mathbf{U}^{n+1} = \mathbf{G}, \quad (2)$$

where $\mathbf{U}^{n+1} \doteq [\mathbf{u}_1^{n+1}, \dots, \mathbf{u}_N^{n+1}]^T$ is the vector of nodal values at time t^{n+1} , $\delta_t(\mathbf{U}) \doteq \Delta t_{n+1}^{-1}(\mathbf{U}^{n+1} - \mathbf{U}^n)$, and $\Delta t_{n+1} \doteq (t^{n+1} - t^n)$. The $m \times m$ -matrices relating nodes $i, j \in \mathcal{N}_h$ are given by

$$\begin{aligned} \mathbf{M}_{ij}^{\beta\gamma} &\doteq (\varphi_j, \varphi_i) \delta_{\beta\gamma} + \mathcal{M}_{ij}^{\beta\gamma}, \\ \mathbf{K}_{ij}^{\beta\gamma} &\doteq -(\varphi_j \delta_{\beta\xi}, \mathbf{f}'_k(\mathbf{u}_j^{n+1})^{\xi\eta} \cdot \partial_k \varphi_i \delta_{\eta\gamma}) + (\varphi_j \delta_{\beta\xi}, n_k \cdot \mathbf{f}'_k(\mathbf{u}_j^{n+1})^{\xi\eta} \varphi_i \delta_{\eta\gamma})_{\Gamma_{\text{out}}} + \mathcal{K}_{ij}^{\beta\gamma}, \\ \mathbf{G}_i^\beta &\doteq (\mathbf{g}^\beta, \varphi_i) + \mathcal{G}_i^\beta, \end{aligned}$$

where Einstein summation applies, $\beta, \gamma, \xi, \eta \in \{1, \dots, m\}$ are the component indices, $\delta_{\beta\gamma}$ is the Kronecker delta, and \mathcal{M} , \mathcal{K} , and \mathcal{G} are the terms arising from applying the conformity constraints in the mass, flux and body forces terms.

2.3. Stability properties. Finally, let us review some concepts required for discussing the stabilization method used in the subsequent sections. Let us recall some definitions used for scalar problems.

Definition 2.1 (Local discrete extremum). *The function $v_h \in V_h$ has a local discrete minimum (resp. maximum) on $i \in \mathcal{N}_h$ if $u_i \leq u_j$ (resp. $u_i \geq u_j$) $\forall j \in \mathcal{N}_h(\Omega_i)$.*

Definition 2.2 (Local discrete maximum principle (DMP)). *A solution $u_h \in V_h$ satisfies the local discrete maximum principle if for every $i \in \mathcal{N}_h$*

$$\min_{j \in \mathcal{N}_h(\Omega_i) \setminus \{i\}} u_j \leq u_i \leq \max_{j \in \mathcal{N}_h(\Omega_i) \setminus \{i\}} u_j.$$

Definition 2.3 (LED). *A scheme is local extremum diminishing if, for every u_i that is a local discrete maximum (resp. minimum),*

$$\frac{du_i}{dt} \leq 0, \quad \left(\text{resp. } \frac{du_i}{dt} \geq 0 \right),$$

is satisfied.

One possible strategy to satisfy the above properties consists in designing a scheme that yields a positive diagonal mass matrix and a stiffness matrix that satisfies

$$\sum_j \mathbf{A}_{ij} = 0, \quad \text{and} \quad \mathbf{A}_{ij} \leq 0, \quad i \neq j. \quad (3)$$

In this case, it is possible to rewrite the system as

$$m_i \delta_t u^{n+1} + \sum_{j \in \mathcal{N}_h(\Omega_i) \setminus \{i\}} \mathbf{A}_{ij} (u_j^{n+1} - u_i^{n+1}) = 0, \quad \forall i \in \mathcal{N}_h.$$

As shown in [41], such a scheme is local extremum diminishing (LED). Moreover, for the steady scheme obtained by dropping the transient term, property (3) leads to solutions that satisfy the local DMP [20].

Stability properties for hyperbolic systems can be based on the extension of the above to hyperbolic systems in characteristic variables. In this direction, we define local bounds preserving schemes as follows.

Definition 2.4. *The discrete scheme*

$$\sum_j \mathbf{M}_{ij} \delta_t \mathbf{u}_i^{n+1} + \sum_{j \neq i} \mathbf{A}_{ij} (\mathbf{u}_j^{n+1} - \mathbf{u}_i^{n+1}) = \mathbf{0}$$

is said to be local bounds preserving if \mathbf{M} is diagonal with positive entries (i.e., $\mathbf{M}_{ij} = m_i \delta_{ij} I_{m \times m}$), \mathbf{A}_{ij} has non-positive eigenvalues for every $j \neq i$, and $\sum_j \mathbf{A}_{ij} = \mathbf{0}$.

Unfortunately, to the best of our knowledge, satisfying this definition does not ensure positivity of density, internal energy, or non-decreasing entropy. In any case, numerical schemes based on this definition have shown good numerical behavior [36, 39, 43, 48].

Several stabilization strategies have been defined based on the previous ideas. One of the most simple strategies consists in adding a scalar artificial diffusion term proportional to the spectral radius of \mathbf{A}_{ij} [36, 46]. This strategy is usually called Rusanov artificial diffusion, since the scheme results in the Rusanov Riemann solver for linear FEs in one dimension [36, 57]. Without any special treatment, the resulting scheme is only first order accurate. The key for recovering high-order convergence is to modulate the action of the artificial diffusion term, and restrict its action to the vicinity of discontinuities. In the present work, our stabilization term for systems of equations is based on Rusanov artificial diffusion and a differentiable shock detector recently developed for scalar problems [3, 15].

Finally, it is also important to define the concept of linearity preservation.

Definition 2.5. *Given $\mathbf{u}_h \in \mathbf{V}_h$ and \mathcal{J} the set of conservative variables that are used to detect inadmissible values of u_h , a stabilization scheme is said to be linearity preserving if the stabilization vanishes at any region such that $u_h^\beta \in P_1(\Omega) \forall \beta \in \mathcal{J}$.*

3. NONLINEAR STABILIZATION

In this section, we describe the additional terms used for the stabilization of problem (2). In particular, we use the stabilization terms defined in [3] for the scalar problem and [5] for Euler:

$$B_h(\mathbf{w}_h; \mathbf{u}_h, \mathbf{v}_h) \doteq \begin{cases} \sum_{i \in \mathcal{N}_h} \sum_{j \in \mathcal{N}_h(\Omega_i)} \nu_{ij}(w_h) v_i u_j \ell(i, j), & \text{for } m = 1, \\ \sum_{K_e \in \mathcal{T}_h} \sum_{i, j \in \mathcal{N}_h(K_e)} \nu_{ij}^e(\mathbf{w}_h) \ell(i, j) \mathbf{v}_i \cdot I_{m \times m} \mathbf{u}_j, & \text{for } m > 1, \end{cases} \quad (4)$$

for any $\mathbf{w}_h, \mathbf{u}_h, \mathbf{v}_h \in \mathbf{V}_h$. Here, ℓ is the graph-Laplacian operator defined as $\ell(i, j) \doteq 2\delta_{ij} - 1$ (see [3, 27]). In the case of a scalar problem, $m = 1$, the nodal artificial diffusion $\nu_{ij}(w_h)$ is defined as

$$\begin{aligned} \nu_{ij}(w_h) &\doteq \max\{\alpha_i(w_h) \mathbf{K}_{ij}, 0, \alpha_j(w_h) \mathbf{K}_{ji}\} \quad \text{for } j \in \mathcal{N}_h(\Omega_i) \setminus \{i\}, \\ \nu_{ii}(w_h) &\doteq \sum_{j \in \mathcal{N}_h(\Omega_i) \setminus \{i\}} \nu_{ij}(w_h). \end{aligned}$$

We denote by $\alpha(w_h)$ the scalar shock detector used for computing the artificial diffusion parameter. In the case of the Euler equations, the element-wise artificial diffusion $\nu_{ij}^e(\mathbf{w}_h)$ is defined as

$$\begin{aligned} \nu_{ij}^e(\mathbf{w}_h) &\doteq \max(\boldsymbol{\alpha}_i(\mathbf{w}_h) \lambda_{ij}^{\max}, \boldsymbol{\alpha}_j(\mathbf{w}_h) \lambda_{ji}^{\max}) \\ &+ \sum_{k \in \mathcal{M}(i)} C_{ki} \max(\boldsymbol{\alpha}_k(\mathbf{w}_h) \lambda_{kj}^{\max}, \boldsymbol{\alpha}_j(\mathbf{w}_h) \lambda_{jk}^{\max}) \\ &+ \sum_{k \in \mathcal{M}(j)} C_{kj} \max(\boldsymbol{\alpha}_i(\mathbf{w}_h) \lambda_{ik}^{\max}, \boldsymbol{\alpha}_k(\mathbf{w}_h) \lambda_{ki}^{\max}) \\ &+ \sum_{k \in \mathcal{M}(i) \cap \mathcal{M}(j)} C_{ki} C_{kj} \boldsymbol{\alpha}_k(\mathbf{w}_h) \lambda_{kk}^{\max}, \quad \text{for } j \in \mathcal{N}_h(\Omega_i) \setminus \{i\}, \\ \nu_{ii}^e(\mathbf{w}_h) &\doteq \sum_{j \in \mathcal{N}_h(\Omega_i) \setminus \{i\}} \nu_{ij}^e(\mathbf{w}_h), \end{aligned} \quad (5)$$

where λ_{ij}^{\max} is the spectral radius of the elemental convection matrix relating nodes $i, j \in \mathcal{N}_h$, i.e., $\rho(\mathbf{f}'(\mathbf{u}_{ij}) \cdot (\nabla \varphi_j, \varphi_i)_{K_e})$, where \mathbf{u}_{ij} is the Roe average between \mathbf{u}_i and \mathbf{u}_j (see (7)). Notice that for ν_{ij}^e and $k \in \mathcal{N}_h^{hg}$, λ_{ik}^{\max} is actually the spectral radius of $\rho(\mathbf{f}'(\mathbf{u}_{ij}) \cdot (\nabla \varphi_k, \varphi_i)_{K_e})$. This is a direct

consequence of using a FE approximation of the fluxes. As previously discussed, this artificial diffusion term is based on Rusanov scalar diffusion [38]. It is important to mention that, given any direction vector \mathbf{n} , the eigenvalues $\mathbf{f}'(\mathbf{u}_{ij}) \cdot \mathbf{n}$ of the Roe-averaged flux Jacobian in that direction can be easily computed as

$$\lambda_{1,\dots,d} = \mathbf{v}_{ij} \cdot \mathbf{n}, \quad \lambda_{d+1} = \mathbf{v}_{ij} \cdot \mathbf{n} - a_{ij} \|\mathbf{n}\|, \quad \lambda_{d+2} = \mathbf{v}_{ij} \cdot \mathbf{n} + a_{ij} \|\mathbf{n}\|, \quad (6)$$

where the velocity \mathbf{v}_{ij} and sound speed a_{ij} are computed using the Roe-averaged values

$$a_{ij} = \sqrt{(\gamma - 1) \left(H_{ij} - \frac{\|\mathbf{v}_{ij}\|^2}{2} \right)}, \quad \mathbf{v}_{ij} = \frac{\mathbf{v}_i \sqrt{\rho_j} + \mathbf{v}_j \sqrt{\rho_i}}{\sqrt{\rho_i} + \sqrt{\rho_j}}, \quad (7)$$

$$H_{ij} = \frac{H_i \sqrt{\rho_i} + H_j \sqrt{\rho_j}}{\sqrt{\rho_i} + \sqrt{\rho_j}}, \quad H_i = E_i + \frac{p_i}{\rho_i}, \quad \text{and} \quad \rho_{ij} = \sqrt{\rho_i \rho_j}.$$

This property greatly simplifies the computation of $\rho(\mathbf{f}'(\mathbf{u}_{ij}) \cdot (\nabla \varphi_j, \varphi_i)_{K_e})$.

We denote by $\alpha_i(\mathbf{w}_h)$ the shock detector used for modulating the action of the artificial diffusion term. The idea behind the definition of this detector is to minimize the amount of artificial diffusion introduced while stabilizing any oscillatory behavior. We want to ensure that the resulting stabilized scheme satisfies Def. 2.4 in regions where the local DMP is violated (see Def. 2.2) by a given set of components. $\alpha_i(\mathbf{w}_h)$ must be a positive real number that takes value 1 when $\mathbf{u}_h(\mathbf{x}_i)$ is an inadmissible value of \mathbf{u}_h , and smaller than 1 otherwise. To this end, we define

$$\alpha_i(\mathbf{u}_h) \doteq \max\{\alpha_i(u_h^\beta)\}_{\beta \in \mathcal{J}}, \quad \forall i \in \mathcal{N}_h \quad (8)$$

where \mathcal{J} is the set of components that are used to detect inadmissible values of \mathbf{u}_h , e.g. density and total energy in the case of Euler equations. For simplicity, we restrict ourselves to the components of \mathbf{u}_h . However, derived quantities such as the pressure or internal energy can be also used. For scalar equations, since the stabilization is defined for the assembled system, the shock detector α_i only needs to be defined for $i \in \mathcal{N}_h$. However, for the elemental definition used for Euler equations, it is also required for $i \in \mathcal{N}_h^{hg}$. In that case, we use the maximum of its constraining nodes, i.e.,

$$\alpha_k(\mathbf{u}_h) \doteq \max_{j \in \mathcal{M}(k)} \alpha_j(\mathbf{u}_h) \quad \text{for } k \in \mathcal{N}_h^{hg}.$$

Let us recall some useful notation from [3] to introduce the scalar shock detector $\alpha_i(w_h)$. Let $\mathbf{r}_{ij} = \mathbf{x}_j - \mathbf{x}_i$ be the vector pointing from node \mathbf{x}_i to \mathbf{x}_j with $i, j \in \mathcal{N}_h$ and $\hat{\mathbf{r}}_{ij} \doteq \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|}$. Recall that the set of points \mathbf{x}_j for $j \in \mathcal{N}_h(\Omega_i) \setminus \{i\}$ define the macroelement Ω_i around node \mathbf{x}_i . Let $\mathbf{x}_{ij}^{\text{sym}}$ be the point at the intersection between $\partial\Omega_i$ and the line that passes through \mathbf{x}_i and \mathbf{x}_j that is not \mathbf{x}_j (see Fig. 2). The set of all $\mathbf{x}_{ij}^{\text{sym}}$ for all $j \in \mathcal{N}_h(\Omega_i) \setminus \{i\}$ is represented with $\mathcal{N}_h^{\text{sym}}(\Omega_i)$. We define $\mathbf{r}_{ij}^{\text{sym}} \doteq \mathbf{x}_{ij}^{\text{sym}} - \mathbf{x}_i$. We define $\mathbf{u}_{ij}^{\text{sym}}$ as the value of \mathbf{u}_h at $\mathbf{x}_{ij}^{\text{sym}}$, i.e., $\mathbf{u}_h(\mathbf{x}_{ij}^{\text{sym}})$.

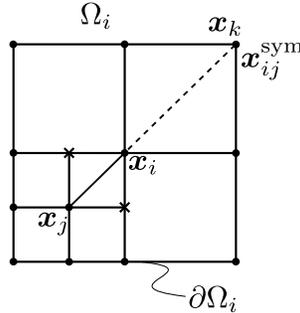


FIGURE 2. u^{sym} drawing

Both $\mathbf{u}_{ij}^{\text{sym}}$ and $\mathbf{x}_{ij}^{\text{sym}}$ are only required to construct a linearity preserving shock detector. Let us define the jump and the mean of a linear approximation of component β of the unknown gradient at

node \mathbf{x}_i in direction \mathbf{r}_{ij} as

$$\left[\left[\nabla u_h^\beta \right] \right]_{ij} \doteq \frac{u_j^\beta - u_i^\beta}{|\mathbf{r}_{ij}|} + \frac{u_{ij}^{\text{sym},\beta} - u_i^\beta}{|\mathbf{r}_{ij}^{\text{sym}}|}, \quad (9)$$

$$\left\{ \left| \nabla u_h^\beta \cdot \hat{\mathbf{r}}_{ij} \right| \right\}_{ij} \doteq \frac{1}{2} \left(\frac{|u_j^\beta - u_i^\beta|}{|\mathbf{r}_{ij}|} + \frac{|u_{ij}^{\text{sym},\beta} - u_i^\beta|}{|\mathbf{r}_{ij}^{\text{sym}}|} \right). \quad (10)$$

In the present work, for each component in \mathcal{J} , we use the same shock detector developed in [3]. Let us recall its definition

$$\alpha_i(u_h^\beta) \doteq \begin{cases} \left[\frac{\left| \sum_{j \in \mathcal{N}_h(\Omega_i)} \left[\nabla u_h^\beta \right]_{ij} \right|}{\sum_{j \in \mathcal{N}_h(\Omega_i)} 2 \left\{ \left| \nabla u_h^\beta \cdot \hat{\mathbf{r}}_{ij} \right| \right\}_{ij}} \right]^q & \text{if } \sum_{j \in \mathcal{N}_h(\Omega_i)} \left\{ \left| \nabla u_h^\beta \cdot \hat{\mathbf{r}}_{ij} \right| \right\}_{ij} \neq 0, \\ 0 & \text{otherwise} \end{cases}, \quad (11)$$

where $q > 1$ is a parameter that minimizes the amount of artificial diffusion introduced. We know from [3, Lm. 3.1] that (11) gets values between 0 and 1, and it is only equal to one if $u_h^\beta(\mathbf{x}_i)$ is a local discrete extremum. Since the linear approximations of the unknown gradients are exact for $u_h^\beta \in \mathcal{P}_1$, the shock detector vanishes when the solution is linear. Thus, it is also linearly preserving for every component in \mathcal{J} . This result follows directly from [3, Th. 4.5].

The shock detector is properly defined for any interior node. However, u_{ij}^β might not seem easy to compute for nodes at the boundary. For nodes at Dirichlet boundaries, $\mathbf{x}_i \in \Gamma_{\text{in}}$, the shock detector is set to 0, since the value of the unknown is fixed. For weak Dirichlet boundary conditions, we refer the reader to [4], where appropriate definitions are developed such that monotonicity and linearity are preserved. In the case of $\mathbf{x}_i \in \Gamma_{\text{out}}$, $\mathbf{x}_{ij}^{\text{sym}}$ might lay on top of node \mathbf{x}_i for some directions. Hence, the second fraction in (9) and (10) becomes undefined. In these cases, this term is dropped and we use

$$\left[\left[\nabla u_h^\beta \right] \right]_{ij} \doteq \frac{u_j^\beta - u_i^\beta}{|\mathbf{r}_{ij}|} \quad \text{and} \quad \left\{ \left| \nabla u_h^\beta \cdot \hat{\mathbf{r}}_{ij} \right| \right\}_{ij} \doteq \frac{1}{2} \left(\frac{|u_j^\beta - u_i^\beta|}{|\mathbf{r}_{ij}|} \right).$$

This definition still ensures that the shock detector takes value 1 for extreme values at \mathbf{x}_i (see [3, Lm. 3.1]). However, linearity preservation in the direction normal to Γ_{out} is lost unless the unknown is constant in that direction.

The final stabilized problem in matrix form reads as follows. Find $\mathbf{u}_h \in \mathbf{V}_h$ such that $u_h^\beta = \bar{u}_h^\beta$ on Γ_{in}^β , $\mathbf{u}_h = \mathbf{u}_{0h}$ at $t = 0$, and

$$\bar{\mathbf{M}}(\mathbf{u}_h^{n+1}) \delta_t \mathbf{U}^{n+1} + \bar{\mathbf{K}}_{ij}(\mathbf{u}_h^{n+1}) \mathbf{U}^{n+1} = \mathbf{G} \quad (12)$$

for $n = 1, \dots, n^{ts}$, where

$$\bar{\mathbf{M}}_{ij}(\mathbf{u}_h^{n+1}) \doteq [1 - \max(\alpha_i, \alpha_j)] \mathbf{M}_{ij} + \delta_{ij} \sum_{k \in \mathcal{N}_h} \max(\alpha_i, \alpha_k) \mathbf{M}_{ik},$$

$\mathbf{M}_i^L = \sum_j \mathbf{M}_{ij}$, and $\bar{\mathbf{K}}_{ij}(\mathbf{u}_h^{n+1}) \doteq \mathbf{K}_{ij} + \mathbf{B}_{ij}$, where \mathbf{B}_{ij} is the stabilization matrix that takes the form

$$\mathbf{B}_{ij}(\mathbf{w}_h) \doteq \begin{cases} \nu_{ij}(\mathbf{w}_h) \ell(i, j), & \text{for } m = 1, \\ \sum_{K_e \in \mathcal{T}_h} \nu_{ij}^e(\mathbf{w}_h) \ell(i, j) I_{m \times m}, & \text{for } m > 1, \end{cases}.$$

Let us show that adapted non-conforming meshes do not jeopardize any of the stability properties defined in Sect. 2.3 and proved for conforming meshes in [4, 8].

Corollary 3.1 (DMP). *The solution of the discrete problem (12) with $m = 1$ and using the shock detector (11) satisfies the local DMP in Def. 2.2 if $g = 0$ and, for every control point $i \in \mathcal{N}_h$ such that u_i is a local discrete extremum, it holds:*

$$\bar{\mathbf{K}}_{ij}(u_h) \leq 0, \quad \forall j \in \mathcal{N}_h(\Omega_i) \setminus \{i\}, \quad \sum_{j \in \mathcal{N}_h(\Omega_i)} \bar{\mathbf{K}}_{ij}(u_h) = 0.$$

Moreover, the resulting scheme is linearity-preserving as defined in Def. 2.5, i.e., $\mathbf{B}_{ij}(\mathbf{u}_h) = 0$ for $\mathbf{u}_h \in \mathcal{P}_1(\Omega_i)$.

Proof. The stabilization scheme for scalar problems is defined on the assembled system. Hence, the modifications introduced in the assembly procedure do not affect the reasoning in the proof of [4, Thm. 5.2]. \square

Lemma 3.1 (Local bounds preservation). *Consider $\mathbf{u}_h \in \mathbf{V}_h$ with component β in the set of tracked variables \mathcal{J} . The stabilized problem (12), with $\mathbf{G} = 0$, is local bounds preserving as defined in Def. 2.4 at any region where u_h^β has extreme values.*

Proof. If component $\beta \in \mathcal{J}$ of \mathbf{u}_h has an extremum at \mathbf{x}_i , we know from [3, Lm. 3.1] that $\alpha_i(u_h^\beta) = 1$. Moreover, it can be checked from (8) that $\alpha_i(\mathbf{u}_h) = 1$. In this case, $\bar{\mathbf{M}}_{ij}(\mathbf{u}_h) = \delta_{ij} \sum_j \mathbf{M}_{ij}$. Hence, $\bar{\mathbf{M}}_{ij}(\mathbf{u}_h) = 0$ for $j \neq i$ and $\bar{\mathbf{M}}_{ii}(\mathbf{u}_h) = m_i$. Therefore, we can rewrite the system as follows

$$\begin{aligned} m_i \partial_t \mathbf{u}_i + \sum_{j \in \mathcal{N}_h(\Omega_i) \setminus \{i\}} \bar{\mathbf{K}}_{ij}(\mathbf{u}_{ij})(\mathbf{u}_j - \mathbf{u}_i) = \\ m_i \partial_t \mathbf{u}_i + \sum_{j \in \mathcal{N}_h(\Omega_i) \setminus \{i\}} \sum_{K_e \in \mathcal{T}_h} \left[(\mathbf{f}'(\mathbf{u}_{ij})) \cdot (\nabla \varphi_j, \varphi_i)_{K_e} \right. \\ \left. + \sum_{k \in \mathcal{M}(j)} C_{kj} (\nabla \varphi_k, \varphi_i)_{K_e} \right. \\ \left. + \sum_{k \in \mathcal{M}(i)} C_{ki} (\nabla \varphi_j, \varphi_k)_{K_e} \right. \\ \left. + \sum_{k \in \mathcal{M}(i) \cap \mathcal{M}(j)} C_{ki} C_{kj} (\nabla \varphi_k, \varphi_k)_{K_e} \right] - \nu_{ij}^e I_{m \times m} (\mathbf{u}_j - \mathbf{u}_i) = \mathbf{0}. \end{aligned}$$

We need to prove that the eigenvalues of $\bar{\mathbf{K}}_{ij}(\mathbf{u}_{ij})$ are non-positive. To this end, let us show that the following inequality holds

$$\begin{aligned} \sum_{K_e \in \mathcal{T}_h} \left(\rho(\mathbf{f}'(\mathbf{u}_{ij}) \cdot (\nabla \varphi_j, \varphi_i)_{K_e}) \right. \\ \left. + \sum_{k \in \mathcal{M}(j)} C_{kj} \rho(\mathbf{f}'(\mathbf{u}_{ij}) \cdot (\nabla \varphi_k, \varphi_i)_{K_e}) \right. \\ \left. + \sum_{k \in \mathcal{M}(i)} C_{ki} \rho(\mathbf{f}'(\mathbf{u}_{ij}) \cdot (\nabla \varphi_j, \varphi_k)_{K_e}) \right. \\ \left. + \sum_{k \in \mathcal{M}(i) \cap \mathcal{M}(j)} C_{ki} C_{kj} \rho(\mathbf{f}'(\mathbf{u}_{ij}) \cdot (\nabla \varphi_k, \varphi_k)_{K_e}) \right) \geq \rho(\mathbf{f}'(\mathbf{u}_{ij}) \cdot (\nabla \varphi_j, \varphi_i)). \end{aligned}$$

One can observe from (6) that $\rho(\mathbf{f}'(\mathbf{u}_{ij}) \cdot (\nabla \varphi_j, \varphi_i)_{K_e}) = |\mathbf{v}_{ij} \cdot \mathbf{c}_{ij}^e| + a_{ij} \|\mathbf{c}_{ij}^e\|$, where $\mathbf{c}_{ij}^e = (\nabla \varphi_j, \varphi_i)_{K_e}$. We have that

$$\mathbf{c}_{ij} = (\nabla \varphi_j, \varphi_i) = \sum_{K_e \in \mathcal{T}_h} \left(\mathbf{c}_{ij}^e + \sum_{k \in \mathcal{M}(j)} C_{kj} \mathbf{c}_{ik}^e + \sum_{k \in \mathcal{M}(i)} C_{ki} \mathbf{c}_{kj}^e + \sum_{k \in \mathcal{M}(i) \cap \mathcal{M}(j)} C_{ki} C_{kj} \mathbf{c}_{kk}^e \right).$$

Thus,

$$\begin{aligned} \sum_{K_e \in \mathcal{T}_h} \left(|\mathbf{v}_{ij} \cdot \mathbf{c}_{ij}^e| + \sum_{k \in \mathcal{M}(j)} C_{kj} |\mathbf{v}_{ij} \cdot \mathbf{c}_{ik}^e| \right. \\ \left. + \sum_{k \in \mathcal{M}(i)} C_{ki} |\mathbf{v}_{ij} \cdot \mathbf{c}_{kj}^e| \right. \\ \left. + \sum_{k \in \mathcal{M}(i) \cap \mathcal{M}(j)} C_{ki} C_{kj} |\mathbf{v}_{ij} \cdot \mathbf{c}_{kk}^e| \right) \geq |\mathbf{v}_{ij} \cdot \mathbf{c}_{ij}|, \end{aligned}$$

and

$$\begin{aligned} \sum_{K_e \in \mathcal{T}_h} \left(a_{ij} \|\mathbf{c}_{ij}^e\| + \sum_{k \in \mathcal{M}(j)} C_{kj} a_{ij} \|\mathbf{c}_{ik}^e\| + \sum_{k \in \mathcal{M}(i)} C_{ki} a_{ij} \|\mathbf{c}_{kj}^e\| \right. \\ \left. + \sum_{k \in \mathcal{M}(i) \cap \mathcal{M}(j)} C_{ki} C_{kj} a_{ij} \|\mathbf{c}_{kk}^e\| \right) \geq a_{ij} \|\mathbf{c}_{ij}\|. \end{aligned}$$

Therefore, $\sum_e \rho(\mathbf{K}_{ij}^e(\mathbf{u}_{ij})) \geq \rho(\mathbf{K}_{ij}(\mathbf{u}_{ij}))$. Moreover, by definition (see (5)),

$$\begin{aligned} \nu_{ij}^e &\geq \rho(\mathbf{f}'(\mathbf{u}_{ij}) \cdot (\nabla \varphi_j, \varphi_i)_{K_e}) + \sum_{k \in \mathcal{M}(j)} C_{kj} \rho(\mathbf{f}'(\mathbf{u}_{ij}) \cdot (\nabla \varphi_k, \varphi_i)_{K_e}) \\ &+ \sum_{k \in \mathcal{M}(i)} C_{ki} \rho(\mathbf{f}'(\mathbf{u}_{ij}) \cdot (\nabla \varphi_j, \varphi_k)_{K_e}) \\ &+ \sum_{k \in \mathcal{M}(i) \cap \mathcal{M}(j)} C_{ki} C_{kj} \rho(\mathbf{f}'(\mathbf{u}_{ij}) \cdot (\nabla \varphi_k, \varphi_k)_{K_e}) \quad \text{for } j \neq i. \end{aligned}$$

Furthermore, one can infer from (4) that $\rho(\mathbf{B}_{ij}^e(\mathbf{u}_{ij})) \geq \rho(\mathbf{K}_{ij}^e(\mathbf{u}_{ij}))$, where $\mathbf{B}_{ij}^e(\mathbf{u}_{ij})$ and $\mathbf{K}_{ij}^e(\mathbf{u}_{ij})$ are the elemental stabilization and stiffness matrices, respectively. Hence, $\rho(\mathbf{B}_{ij}(\mathbf{u}_{ij})) \geq \rho(\mathbf{K}_{ij}(\mathbf{u}_{ij}))$. Finally, since $\bar{\mathbf{K}}_{ij} = \mathbf{K}_{ij} + \mathbf{B}_{ij}$ and $\mathbf{B}_{ij} = \sum_e \mathbf{B}_{ij}^e = -\sum_e \nu_{ij}^e I_{m \times m}$ for all $j \neq i$. Then, the maximum eigenvalue of $\bar{\mathbf{K}}_{ij}(\mathbf{u}_{ij})$ is non-positive, which completes the proof. \square

Notice that it is essential to apply properly the constraints at the flux FE approximation, i.e., $\mathbf{f}'(\mathbf{u}_k) = \sum_{i \in \bar{\mathcal{M}}(k)} C_{ki} \mathbf{f}'(\mathbf{u}_i)$. Otherwise, it is not possible to formally prove local bound preservation. However, experimental results in the present work show that using $\mathbf{f}'(\mathbf{u}_k)$ does not affect the overall performance of the scheme.

3.1. Differentiable stabilization. In the case of steady, or implicit time integration, differentiability plays a role in the convergence behavior of the nonlinear solver. This is especially important if one wants to use Newton's method. The authors show in [3–5] that nonlinear convergence can be improved after few modifications to make the scheme twice-differentiable. In this section, we introduce a set of regularizations applied to all non-differentiable functions present in the stabilized scheme introduced above. In order to regularize these functions, we follow the same strategy as in [3–5]. Absolute values are replaced by

$$|x|_{1, \varepsilon_h} = \sqrt{x^2 + \varepsilon_h}, \quad |x|_{2, \varepsilon_h} = \frac{x^2}{\sqrt{x^2 + \varepsilon_h}},$$

where ε_h is a small positive value. Note that $|x|_{2, \varepsilon_h} \leq |x| \leq |x|_{1, \varepsilon_h}$. Next, we also use the smooth maximum function

$$\max_{\sigma_h}(x, y) \doteq \frac{|x - y|_{1, \sigma_h} + x + y}{2} \geq \max(x, y),$$

where σ_h is a small positive value. In addition, we need a smooth function to limit the value of any given quantity to one. To this end, we use

$$Z(x) \doteq \begin{cases} 2x^4 - 5x^3 + 3x^2 + x, & x < 1, \\ 1, & x \geq 1. \end{cases}$$

The set of twice-differentiable functions defined above allows us to redefine the stabilization term introduced in Sect. 3. In particular, we define

$$\tilde{B}_h(\mathbf{w}_h; \mathbf{u}_h, \mathbf{v}_h) \doteq \begin{cases} \sum_{i \in \mathcal{N}_h} \sum_{j \in \mathcal{N}_h(\Omega_i)} \tilde{v}_{ij}(\mathbf{w}_h) v_i u_j \ell(i, j), & \text{for } m = 1, \\ \sum_{K_e \in \mathcal{T}_h} \sum_{i, j \in \mathcal{N}_h(K_e)} \tilde{v}_{ij}^e(\mathbf{w}_h) \ell(i, j) \mathbf{v}_i \cdot I_{m \times m} \mathbf{u}_j, & \text{for } m > 1, \end{cases} \quad (13)$$

where

$$\begin{aligned} \tilde{v}_{ij}(\mathbf{w}_h) &\doteq \max_{\sigma_h} \left(\max_{\sigma_h} (\alpha_{\varepsilon_h, i}(\mathbf{w}_h) \mathbf{K}_{ij}, \alpha_{\varepsilon_h, j}(\mathbf{w}_h) \mathbf{K}_{ji}), 0 \right) \quad \text{for } j \in \mathcal{N}_h(\Omega_i) \setminus \{i\}, \\ \tilde{v}_{ii}(\mathbf{w}_h) &\doteq \sum_{j \in \mathcal{N}_h(\Omega_i) \setminus \{i\}} \tilde{v}_{ij}(\mathbf{w}_h), \end{aligned}$$

and

$$\begin{aligned} \tilde{v}_{ij}^e(\mathbf{w}_h) &\doteq \max_{\sigma_h} (\alpha_{\varepsilon_h, i}(\mathbf{w}_h) \lambda_{ij}^{\max}, \alpha_{\varepsilon_h, j}(\mathbf{w}_h) \lambda_{ji}^{\max}) \\ &\quad + \sum_{k \in \mathcal{M}(i)} C_{ki} \max_{\sigma_h} (\alpha_{\varepsilon_h, k}(\mathbf{w}_h) \lambda_{kj}^{\max}, \alpha_{\varepsilon_h, j}(\mathbf{w}_h) \lambda_{jk}^{\max}) \\ &\quad + \sum_{k \in \mathcal{M}(j)} C_{kj} \max_{\sigma_h} (\alpha_{\varepsilon_h, i}(\mathbf{w}_h) \lambda_{ik}^{\max}, \alpha_{\varepsilon_h, k}(\mathbf{w}_h) \lambda_{ki}^{\max}) \\ &\quad + \sum_{k \in \mathcal{M}(i) \cap \mathcal{M}(j)} C_{ki} C_{kj} \alpha_{\varepsilon_h, k}(\mathbf{w}_h) \lambda_{kk}^{\max}, \quad \text{for } j \in \mathcal{N}_h(\Omega_i) \setminus \{i\}, \\ \tilde{v}_{ii}^e(\mathbf{w}_h) &\doteq \sum_{j \in \mathcal{N}_h(\Omega_i) \setminus \{i\}} \tilde{v}_{ij}^e(\mathbf{w}_h). \end{aligned}$$

Let us note that λ_{ij}^{\max} needs to be regularized as $\lambda_{ij}^{\max} = |\mathbf{v}_{ij} \cdot \mathbf{c}_{ij}^e|_{1, \varepsilon_h} + c \|\mathbf{c}_{ij}^e\|$. The shock detector is also regularized as follows:

$$\alpha_{\varepsilon_h, i}(\mathbf{u}_h) \doteq \max_{\sigma_h} \{\alpha_{\varepsilon_h, i}(u_h^\beta)\}_{\beta \in \mathcal{J}}.$$

Notice that the regularized maximum has only been defined for two arguments. However, for $|\mathcal{J}| > 2$, one can chain several times the regularized functions, i.e., $\max_{\sigma_h}(\max_{\sigma_h}(\dots))$. In the case of the component shock detector we recall the definition in [3, Eq. 18]

$$\alpha_{\varepsilon_h, i}(u_h^\beta) \doteq \left[Z \left(\frac{\left| \sum_{j \in \mathcal{N}_h(\Omega_i)} \left[\left| \nabla u_h^\beta \right|_{ij} \right]_{1, \varepsilon_h} + \zeta_h}{\sum_{j \in \mathcal{N}_h(\Omega_i)} 2 \left\{ \left| \nabla u_h^\beta \cdot \hat{\mathbf{r}}_{ij} \right|_{2, \varepsilon_h} \right\}_{ij} + \zeta_h} \right) \right]^q, \quad (14)$$

where ζ_h is a small value for preventing division by zero. Finally, the twice-differentiable stabilized scheme reads: find $\mathbf{u}_h \in \mathbf{V}_h$ such that $u_h^\beta = \bar{u}_h^\beta$ on Γ_{in}^β , $\mathbf{u}_h = \mathbf{u}_{0h}$ at $t = 0$, and

$$\tilde{\mathbf{M}}(\mathbf{u}_h^{n+1}) \delta_t \mathbf{U}^{n+1} + \tilde{\mathbf{K}}_{ij}(\mathbf{u}_h^{n+1}) \mathbf{U}^{n+1} = \mathbf{G} \quad \text{for } n = 1, \dots, n^{ts}, \quad (15)$$

where

$$\begin{aligned} \tilde{\mathbf{M}}_{ij}(\mathbf{u}_h^{n+1}) &\doteq [1 - \max_{\sigma_h} (\alpha_{\varepsilon_h, i}, \alpha_{\varepsilon_h, j})] \mathbf{M}_{ij} + \delta_{ij} \sum_{k \in \mathcal{N}_h} \max_{\sigma_h} (\alpha_i, \alpha_k) \mathbf{M}_{ik}, \\ \tilde{\mathbf{K}}_{ij}(\mathbf{u}_h^{n+1}) &\doteq \mathbf{K}_{ij}(\mathbf{u}_h^{n+1}) + \tilde{\mathbf{B}}_{ij}(\mathbf{u}_h^{n+1}), \end{aligned}$$

and $\tilde{\mathbf{B}}_{ij}$ is the regularized stabilization matrix that takes the form

$$\tilde{\mathbf{B}}_{ij}(\mathbf{w}_h) \doteq \begin{cases} \tilde{v}_{ij}(\mathbf{w}_h) \ell(i, j), & \text{for } m = 1, \\ \sum_{K_e \in \mathcal{T}_h} \tilde{v}_{ij}^e(\mathbf{w}_h) \ell(i, j) I_{m \times m}, & \text{for } m > 1. \end{cases}$$

Corollary 3.2. *The scheme in (2) with $\mathbf{G} = 0$ and the differentiable stabilization in (13) is local bounds preserving, as defined in Def. 2.4, at any region where u_h^β has extreme values for every β in \mathcal{J} .*

Proof. For an extreme value of u_h^β , since $|x|_{2,\varepsilon_h} \leq |x| \leq |x|_{1,\varepsilon_h}$ the quotient of (14) is larger than one. Hence, by definition of $Z(x)$, $\alpha_{\varepsilon_h,i}$ is equal to 1. At this point, it is easy to check that $\tilde{\nu}_{ij}^e \geq \nu_{ij}^e$ in virtue of the definition of $\max \sigma_h$. Therefore, $\rho(\tilde{\mathbf{B}}_{ij}^e(\mathbf{u}_h)) \geq \rho(\mathbf{B}_{ij}^e(\mathbf{u}_h))$, completing the proof. \square

Moreover, it is important to mention that the differentiable shock detector is weakly linearly-preserving as ζ_h tends to zero. This result follows directly from [3]. In order to obtain a differentiable operator, we have added a set of regularizations that rely on different parameters, e.g., σ_h , ε_h , ζ_h . Giving a proper scaling of these parameters is essential to recover theoretic convergence rates. In particular, we use the following relations

$$\sigma_h = \sigma |\lambda^{\max}|^2 L^{2(d-3)} h^4, \quad \varepsilon_h = \varepsilon L^{-4} h^2, \quad \zeta_h = L^{-1} \zeta,$$

where σ , ε , and ζ are small positive parameters, d is the spatial dimension of the problem, L is a characteristic length. The value $|\lambda^{\max}|$ being used is the maximum on the whole domain of the Euclidean norm. In the case of a scalar problem, it is simply the maximum convection velocity, i.e. $\max_{\mathbf{x} \in \Omega} |\mathbf{v}(\mathbf{x})|$.

4. ADAPTIVE MESH REFINEMENT

The motivation of an adaptive FE method is to solve (12) up to a certain tolerance (or resolution) using the *minimum* number of DOFs. To this end, the solution error ($\mathbf{e}_h = \mathbf{u} - \mathbf{u}_h$) is estimated at each element. With this information at hand, it is possible to iteratively adapt the resolution of the mesh at certain regions. This process can be divided into two parts: estimating the error at every cell, and deciding which and how many cells need to be refined or coarsened. This procedure is performed iteratively until a desired tolerance is achieved or, alternatively, a number of elements is reached. In the present work, we start with a rather coarse mesh and perform the following steps till reaching a stopping criterion:

- (1) Compute solution \mathbf{u}_h ;
- (2) Estimate the error \mathbf{e}_h ;
- (3) Select all cells that need to be refinement or coarsened;
- (4) Update the mesh, and project the solution to the new mesh.

In some cases, the refinement might be driven by features of the solution instead of a classical error estimator. For instance, one may decide to refine the regions around discontinuities. In this scenario, one could use an expression that does not estimate the error, but it allows to concentrate the elements around discontinuities.

4.1. Error estimators. One of the keys of AMR is the ability to provide a good estimation of the error. Several error estimators have been proposed to date [1, 22, 30, 31, 59]. These can be classified, at least, in two main types. Some authors [22, 30, 50, 51, 53] try to compute an upper bound of the error for every cell. Then, provided a user defined tolerance, one can decide to refine or coarsen each cell. However, an adjoint problem needs to be solved in order to compute this upper bound [17, 30]. It is possible to approximate the error bounds without solving an adjoint problem only for simple cases, see [30]. Therefore, this kind of error estimators increases the computational cost substantially. Alternatively, one can simply determine the distribution of the error in the mesh and use this information to drive an adaptivity algorithm. In this scenario, some authors [1, 14, 40, 49, 59, 60] drive the adaptivity process with the solution gradient. In this case, explicit expressions of the estimated error are possible, requiring less computational resources than the previous option.

In general, the adaptive procedure can be described as follows. Given a finite element solution \mathbf{u}_h , the error \mathbf{e}_h is approximated as $\mathbf{e}_h \approx \nabla \mathbf{u} - \nabla \mathbf{u}_h$. Then, the reconstruction is used as an approximation of the exact gradient. This strategy is based on *superconvergence* of special recovery techniques (see [60] and refs. in [1, 49]). Kuzmin and co-workers [14, 49] follow [59] to reconstruct an approximation

of the exact gradient. Kelly et al. [31] proposed a well-known estimator based on gradient recovery:

$$\eta_K^2 \doteq \frac{h_K}{24} \int_{\partial K} \left[\left[\frac{\partial \mathbf{u}_h}{\partial n} \right] \right]^2 d\Gamma,$$

where η_K is the estimated error at every element, K . In this case, the jump $[[\cdot]]$ does not correspond with the linear approximation of the gradient jump $[[\nabla(\cdot)]]_{ij}$ defined in (9). Instead, it takes the classical definition, i.e., $[[u]] = u^+ \cdot n^+ + u^- \cdot n^-$. The main advantages of this estimator are its simplicity and its low computational cost. For these reasons, this estimator is used in the present work.

It is worth mentioning that our problems of interest are characterized by exhibiting discontinuities, where the error concentrates. These regions are susceptible to develop instabilities, and thus, these are the regions in which the shock capturing is activated. Therefore, it is natural to use the shock capturing to drive the adaptivity procedure. We propose an indicator based on the graph Laplacian $\ell(i, j)$ present in the stabilization term (4). This way, we reuse available information and reduce the computational overhead associated with the selection of cells to be refined. The indicator reads:

$$(\tilde{\eta}_K^\beta)^2 \doteq h_K^{d-2} \sum_{i \in \mathcal{N}_h(K)} \sum_{j \in \mathcal{N}_h(\Omega_i)} (u_i^\beta - u_j^\beta)^2,$$

where $\beta \in \mathcal{J}$ is the index of the specific component analyzed. This expression is expected to yield high values around shocks and low values in smooth regions.

4.2. Refinement strategy. After the error has been estimated for every element (or the indicator has been computed at every cell), one needs to decide which element needs to be refined and which one coarsened. If an upper bound of the error is computed, then one may use a given tolerance to make this decision. However, in the present case this is not available. A classical alternative is to refine/coarsen a fixed amount of elements at every iteration [9, 11]. In the present study, a 30% of the elements with higher error estimates (or higher indicator values) are refined whereas a 10% of the elements with lower values are coarsened. This percentages are arbitrary and other choices are valid. Notice that using this setting in two dimensions the number of elements is almost doubled at every iteration. We make use of the parallel n th element algorithm [9, 56] to efficiently determine the indicator thresholds for refining or coarsening the elements.

5. NONLINEAR SOLVER

In this section, we describe the method used for solving the nonlinear system of equations arising from the scheme introduced above. In particular, we use a hybrid Picard–Newton approach in order to increase the robustness of the nonlinear solver. Moreover, we also make use of a line–search method to improve the nonlinear convergence.

We define the residual of the equation (15) at the k -th iteration as

$$\mathbf{R}(\mathbf{u}_h^{k,n+1}) \doteq \tilde{\mathbf{M}}(\mathbf{u}_h^{k,n+1}) \delta_t \mathbf{U}^{k,n+1} + \tilde{\mathbf{K}}_{ij}(\mathbf{u}_h^{k,n+1}) \mathbf{U}^{k,n+1} - \mathbf{G}.$$

Hence, the Jacobian is defined as

$$\begin{aligned} \mathbf{J}(\mathbf{u}_h^{k,n+1}) &\doteq \frac{\partial \mathbf{R}(\mathbf{u}_h^{k,n+1})}{\partial \mathbf{U}^{k,n+1}} \\ &= \Delta t_{t+1}^{-1} \tilde{\mathbf{M}}(\mathbf{u}_h^{k,n+1}) + \tilde{\mathbf{K}}_{ij}(\mathbf{u}_h^{k,n+1}) + \Delta t_{t+1}^{-1} \frac{\partial \tilde{\mathbf{M}}(\mathbf{u}_h^{k,n+1})}{\partial \mathbf{U}^{k,n+1}} \delta_t \mathbf{U}^{k,n+1} + \frac{\partial \tilde{\mathbf{K}}_{ij}(\mathbf{u}_h^{k,n+1})}{\partial \mathbf{U}^{k,n+1}} \mathbf{U}^{k,n+1}. \end{aligned} \quad (16)$$

Therefore, Newton method consists in solving $\mathbf{J}(\mathbf{u}_h^{k,n+1}) \Delta \mathbf{U}^{k+1,n+1} = -\mathbf{R}(\mathbf{u}_h^{k,n+1})$. It is well known that Newton method can diverge if the initial guess of the solution $\mathbf{u}_h^{0,n+1}$ is not close enough to the solution. In order to improve robustness, we use a line–search method to update the solution at every time step. The new approximation is computed as $\mathbf{U}^{k+1,n+1} = \mathbf{U}^{k,n+1} + \lambda \Delta \mathbf{U}^{k+1,n+1}$, where λ is obtained using a standard cubic backtracking algorithm.

As introduced at the beginning of the section, we also use a hybrid approach combining Newton method with Picard linearization. Picard nonlinear iterator can be obtained removing the last two terms of (16), i.e.,

$$\left(\Delta t_{t+1}^{-1} \tilde{\mathbf{M}}(\mathbf{u}_h^{k,n+1}) + \tilde{\mathbf{K}}_{ij}(\mathbf{u}_h^{k,n+1})\right) \Delta \mathbf{U}^{k+1,n+1} = -\mathbf{R}(\mathbf{u}_h^{k,n+1}). \quad (17)$$

Clearly, it is equivalent to

$$\left(\Delta t_{t+1}^{-1} \tilde{\mathbf{M}}(\mathbf{u}_h^{k,n+1}) + \tilde{\mathbf{K}}_{ij}(\mathbf{u}_h^{k,n+1})\right) \mathbf{U}^{k+1,n+1} = \Delta t_{t+1}^{-1} \tilde{\mathbf{M}}(\mathbf{u}_h^{k,n+1}) \mathbf{U}^n + \mathbf{G}.$$

Moreover, we modify the left hand side terms in (17); we use $\alpha_i = 1$ for computing these terms while we use the value obtained from (8) for the residual. Using this strategy, the solution remains unaltered but the obtained approximations $\mathbf{u}_h^{k,n+1}$ for intermediate values of k are more diffusive. Even though this modification slows the nonlinear convergence, it is essential at the first iterations. Otherwise, the robustness of the method might be jeopardized.

The resulting iterative nonlinear solver consists in the following steps. We iterate Picard method in (17), with the modification described above, until the Euclidean norm of the residual vector is smaller than a given tolerance. In the present work, we use a tolerance of 10^{-2} . Afterwards, Newton method with the exact Jacobian in (16) is used until the desired nonlinear convergence criteria is satisfied. We summarize the nonlinear solver introduced above in Alg. 1.

Input: $\mathbf{U}^{0,n+1}$, tol_1 , tol_2 , ε
Output: $\mathbf{U}^{k,n+1}$, k
 $k = 1$, $\varepsilon^1 = \varepsilon$
while $\|\mathbf{R}(\mathbf{U}^{k,n+1})\|/\|\mathbf{R}(\mathbf{U}^{0,n+1})\| \geq \text{tol}_1$ **do**
 Compute $\alpha_i(\mathbf{U}^{k,n+1})$ using (8)
 Compute $\Delta \mathbf{U}^{k+1,n+1}$ using (17)
 Minimize $\|\mathbf{R}(\mathbf{U}^{k+1,n+1})\|$, where $\mathbf{U}^{k+1,n+1} = \lambda \Delta \mathbf{U}^{k+1,n+1} + \mathbf{U}^{k,n+1}$, with respect to λ
 Set $\mathbf{U}^{k+1,n+1} = \lambda \Delta \mathbf{U}^{k+1,n+1} + \mathbf{U}^{k,n+1}$
 Update $k = k + 1$
end
while $\|\mathbf{R}(\mathbf{U}^{k,n+1})\|/\|\mathbf{R}(\mathbf{U}^{0,n+1})\| \geq \text{tol}_2$ **do**
 Compute $\alpha_i(\mathbf{U}^{k,n+1})$ using (8)
 Solve $\mathbf{J}(\mathbf{U}^{k,n+1}) \Delta \mathbf{U}^{k+1,n+1} = -\mathbf{R}(\mathbf{U}^{k,n+1})$ with \mathbf{J} in (16)
 Minimize $\|\mathbf{R}(\mathbf{U}^{k+1,n+1})\|$, where $\mathbf{U}^{k+1,n+1} = \lambda \Delta \mathbf{U}^{k,n+1} + \mathbf{U}^{k,n+1}$, with respect to λ
 Set $\mathbf{U}^{k+1,n+1} = \lambda \Delta \mathbf{U}^{k,n+1} + \mathbf{U}^{k,n+1}$
 Update $k = k + 1$
end

Algorithm 1: Hybrid Picard–Newton method.

6. NUMERICAL RESULTS

In this section, we perform several numerical experiments to assess the numerical scheme introduced in the previous sections. First, we perform a convergence analysis to assess its implementation. Then, we use steady benchmark tests to analyze the effectiveness of the high-order scheme in the context of AMR. In particular, we compare the nonlinear scheme in (15) with its linear (first order) counterpart, i.e., using $\alpha_{\varepsilon_n, i}(\mathbf{u}_h) \equiv 1$.

From previous experience [3, 4, 15, 16], we choose the following regularization parameters: $\sigma = 10^{-2}$, $\varepsilon = 10^{-4}$, and $\zeta = 10^{-10}$. In addition, the density is discontinuous at all shocks and contacts for all Euler tests below. Therefore, we use $\mathcal{J} = \{1\}$ in (8), i.e., the shock detector is based on the density behavior in all Euler tests below.

6.1. Convergence. First, the convergence to a discontinuous solution is analyzed. To this end, we solve two different problems. On the one hand, the following scalar problem is solved

$$\begin{aligned} \nabla \cdot (\mathbf{v}u) &= 0 & \text{in } \Omega &= [0, 1] \times [0, 1], \\ u &= u_D & \text{on } \Gamma_{\text{in}}, \end{aligned} \quad (18)$$

where $\mathbf{v}(x, y) \doteq (1/2, \sin^{-\pi/3})$, and inflow boundary conditions $u_D = 1$ on $\{x = 0\} \cap \{y > 0.7\}$ and $y = 1$, while $u_D = 0$ at the rest of the inflow boundary. This problem has the following analytical solution

$$u(x, y) = \begin{cases} 1 & \text{if } y > 0.7 + 2x \sin^{-\pi/3}, \\ 0 & \text{otherwise.} \end{cases}$$

For the Euler equations, the problem is the well known compression corner test [2, 38], also known as oblique shock test [52, 54]. This benchmark consists in a supersonic flow impinging to a wall at an angle. We use a $[0, 1]^2$ domain with a $M = 2$ flow at 10° with respect to the wall. This leads to two flow regions separated by an oblique shock at 29.3° , see Fig. 3.

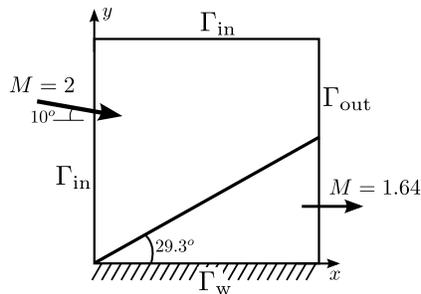


FIGURE 3. Compression corner scheme.

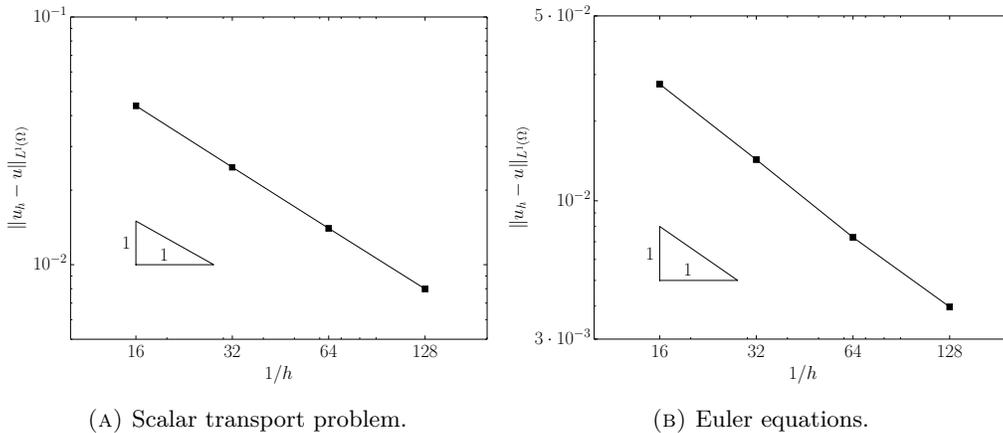


FIGURE 4. Convergence of $\|u - u_h\|_{L^1(\Omega)}$ to a solution with a discontinuity.

Since the solution is not smooth, we expect linear convergence rates in the L^1 -norm. Fig. 4 shows the convergence behavior of both problems with uniform mesh refinements. The experimental convergence rate measured for the scalar transport problem is 0.82, whereas the convergence rate measured is 0.94 for the compression corner test. Therefore, both tests exhibit the expected convergence behavior.

6.2. Linear discontinuity. For this test, we use again the problem in (18). The purpose of this test is twofold. On the one hand, we analyze the effectiveness of the proposed indicator. On the other hand, we compare the effectiveness of the linear and nonlinear stabilization methods. Specifically, this effectiveness is measured as follows. For a given error, we consider a method more effective if it requires less computational time, independently of the number of elements required. In addition, we also solve the problem for successive uniformly refined meshes in order to evaluate the effect of AMR.

For all comparisons, we start with a coarse mesh of 16×16 elements, and proceed adapting the mesh up to a maximum number of elements. For the nonlinear stabilization, we set a maximum of $5 \cdot 10^4$ elements. The maximum number of elements for the low-order method is $2 \cdot 10^6$. The uniform mesh is refined up to a 1024×1024 mesh. We use a nonlinear tolerance of $\|\Delta \mathbf{u}_h\|/\|\mathbf{u}_h\| < 10^{-4}$, and a maximum of 500 iterations.

Fig. 5 shows the evolution of the AMR algorithm for the Kelly error estimator and the proposed refinement strategy based on the graph Laplacian. The results shown in this picture have been obtained using the linear stabilization, and the left-most column using the nonlinear one. It can be observed that both Kelly (η_K) estimator and graph Laplacian ($\tilde{\eta}_K$) indicator refine in the vicinity of the shock. However, the graph Laplacian operator clearly outperforms Kelly estimator.

Figs. 6–8 compare the effectiveness of the low-order and the high-order stabilization schemes. The results are obtained for the stabilization parameter $q = 1$, $q = 2$, and $q = 10$, respectively.

At Fig. 7, the nonlinear stabilization is able to converge the nonlinear problem efficiently and the overhead of solving a nonlinear problem does not strongly affect the overall performance. We note that for the linear scheme the problem is linear. It can be observed that the convergence rate (against time) is much higher for the nonlinear scheme. The linear scheme requires less computational time for coarser meshes but the nonlinear scheme is more effective for tighter accuracies, specially for $q > 1$.

We can observe in Fig. 8 the convergence problems of the nonlinear stabilization at some steps of the refinement procedure. Even though using $q = 10$ improves the accuracy of the method, it also increases the computational cost since the nonlinear problem is harder to solve. As a consequence, the nonlinear stabilization needs a very refined mesh to overcome the performance of the linear stabilization.

6.3. Circular discontinuity. We analyze again the effectiveness of the proposed refinement strategy and the effectiveness of the linear and nonlinear stabilization methods for a slightly more complicated convective field. For this test, we use (18) with $\mathbf{v}(x, y) \doteq (y, -x)$, and inflow boundary conditions

$$\bar{u}(0, y) = \begin{cases} 1 & y \in [0.15, 0.45], \\ \cos^2\left(\frac{10}{3}\pi(y - 0.4)\right) & y \in [0.55, 0.85], \\ 0 & \text{elsewhere.} \end{cases}$$

The analytical solution of this particular configuration consists in the transport of the inflow profile in the direction of the convection. As a result, the solution at the outflow boundary, corresponding to $y = 0$, is $u(x, 0) = \bar{u}(0, x)$. We start with a coarse mesh of 16×16 elements in all cases, and proceed adapting the mesh up to a maximum number of elements. For the nonlinear stabilization, we set a maximum of $5 \cdot 10^4$ elements. The maximum number of elements for the linear stabilization is $2 \cdot 10^6$. We use a nonlinear tolerance of $\|\Delta \mathbf{u}_h\|/\|\mathbf{u}_h\| < 10^{-4}$, and a maximum of 500 iterations.

Figs. 9–11 compare the effectiveness of the linear and nonlinear stabilization. These results use the stabilization parameter $q = 1$, $q = 2$, and $q = 10$, respectively. In Fig. 10, the high-order scheme is able to converge efficiently and the overhead of solving a nonlinear problem does not strongly affect the overall performance. Nevertheless, the low-order scheme usually requires similar computational time for any given error. However, it can be observed that the convergence rate (in time) using the Kelly error estimator is slightly higher for the high-order scheme. Actually, it outperforms the low-order scheme for the finer meshes.

In contrast, we do not observe the significant convergence problems in Fig. 11 even though the linear stabilization is slightly more efficient for coarse meshes. The convergence rate (in time) is higher for the nonlinear stabilization and it is actually more efficient for the finer meshes.

Fig. 12 shows the evolution of the AMR algorithm for both η_K and $\tilde{\eta}_K$ with the linear stabilization and $\tilde{\eta}_K$ with the nonlinear one. It can be observed that both Kelly (η_K) estimator and graph Laplacian

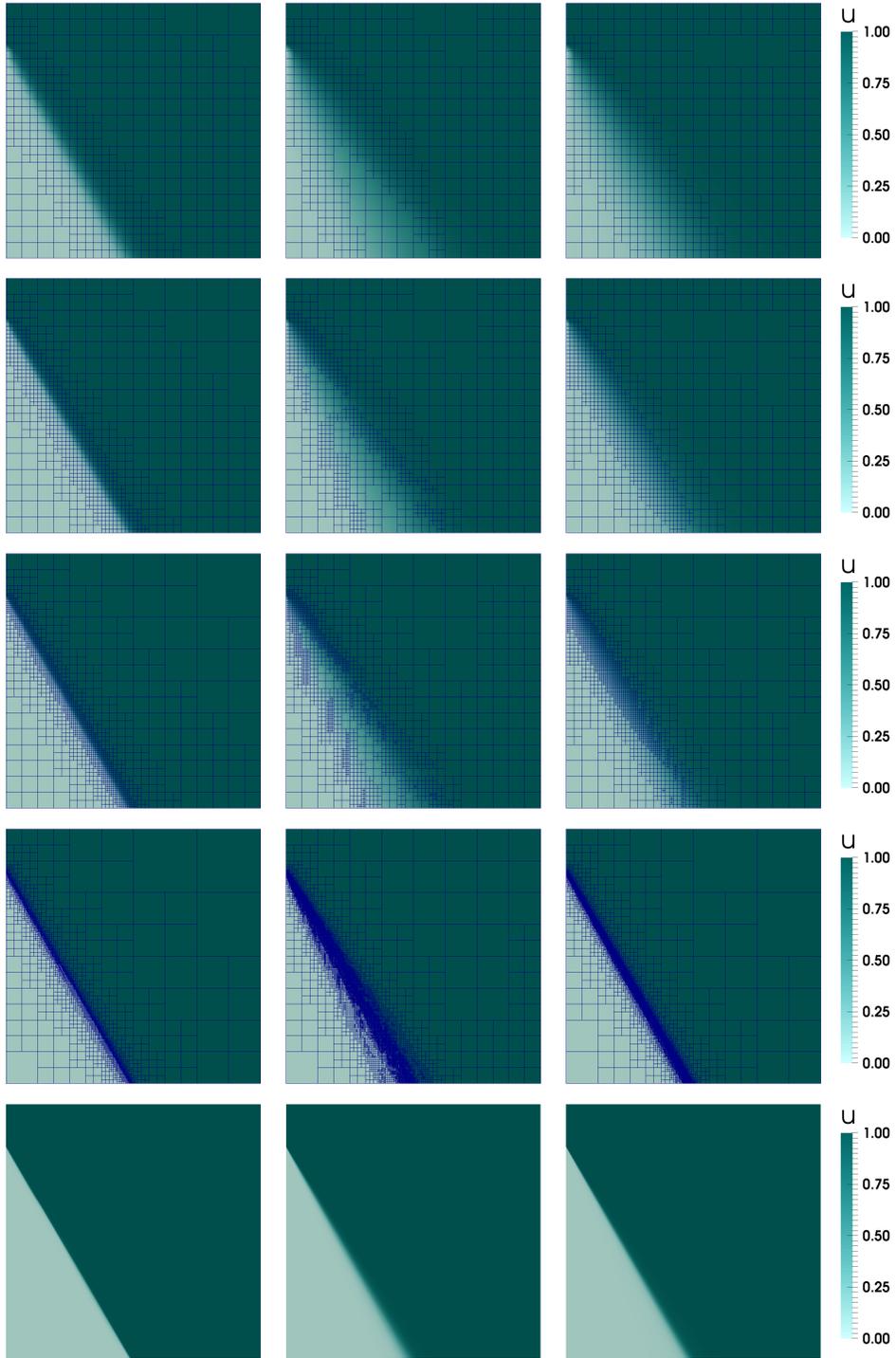


FIGURE 5. Evolution of the mesh refinement process. $\tilde{\eta}_K$ with high-order scheme is used in the left column. Low-order scheme with Kelly estimator is used in the central column. $\tilde{\eta}_K$ with low-order scheme is used in the right column. For the low-order scheme from top to bottom results have been obtained at refinement step 1, 2, 3, 9, and 9. For the high-order refinement steps are 1, 2, 3, 5, and 5.

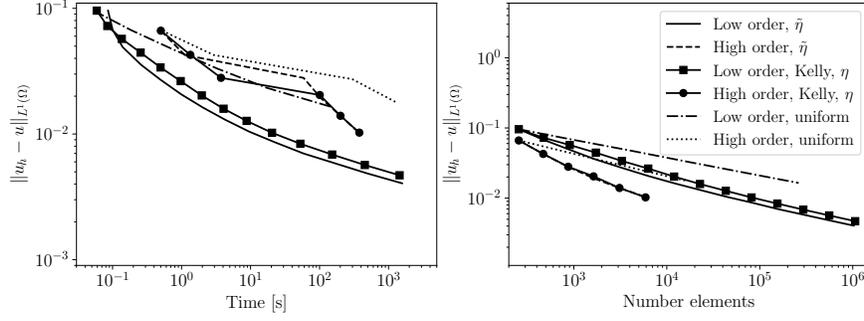


FIGURE 6. Time and elements convergence comparison for the transport problem with a linear discontinuity, $q = 1$.

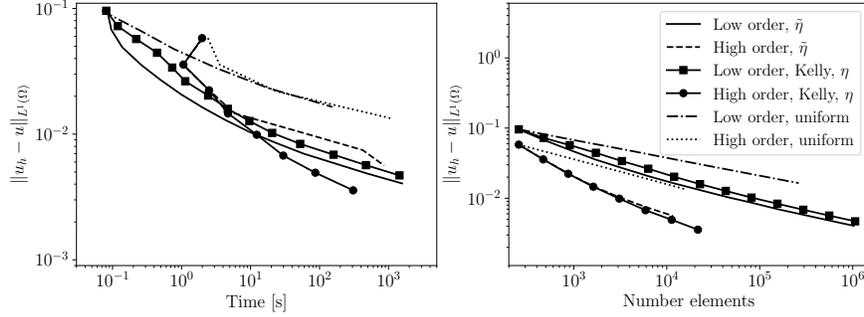


FIGURE 7. Time and elements convergence comparison for the transport problem with a linear discontinuity, $q = 2$.

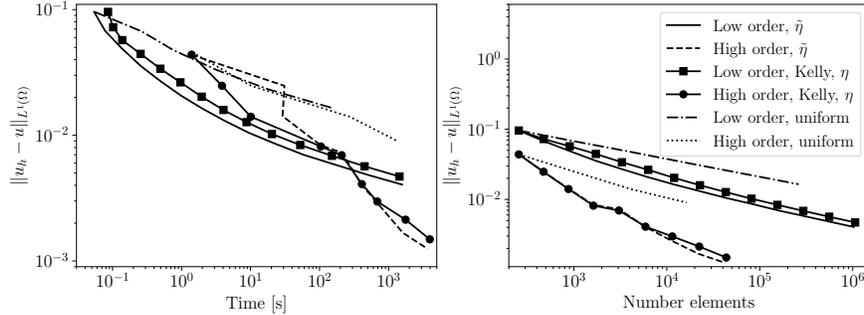


FIGURE 8. Time and elements convergence comparison for the transport problem with a linear discontinuity, $q = 10$.

indicator detect the regions that require more resolution. In any case, as in the previous example, the graph Laplacian operator ($\tilde{\eta}_K$) performs slightly better.

6.4. Compression corner. Let us consider now the Euler equations. We start with the compression corner test (see Fig. 3). We analyze the effectiveness of the high-order scheme, and evaluate the performance of the graph Laplacian indicator. We start with a coarse mesh of 16×16 elements, and adapt it up to a maximum number of elements. For the high-order method, we set a maximum of $5 \cdot 10^3$ elements. The maximum number of elements for the low-order method is $5 \cdot 10^4$. We use a nonlinear tolerance of $\|\Delta \mathbf{u}_h\| / \|\mathbf{u}_h\| < 10^{-4}$ and a maximum of 500 iterations.

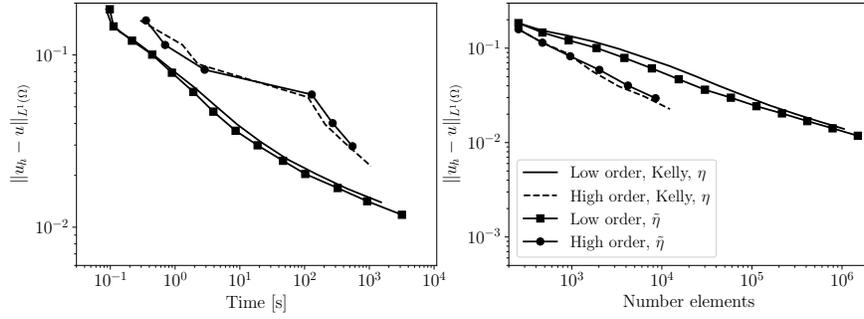


FIGURE 9. Time and elements convergence comparison for the transport problem with a circular convection field, $q = 1$.

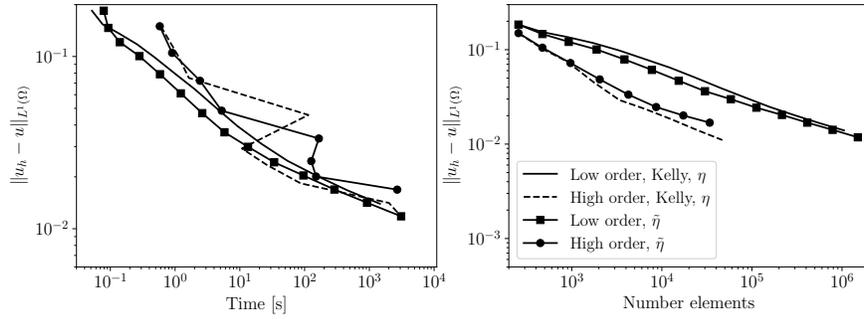


FIGURE 10. Time and elements convergence comparison for the transport problem with a circular convection field, $q = 2$.

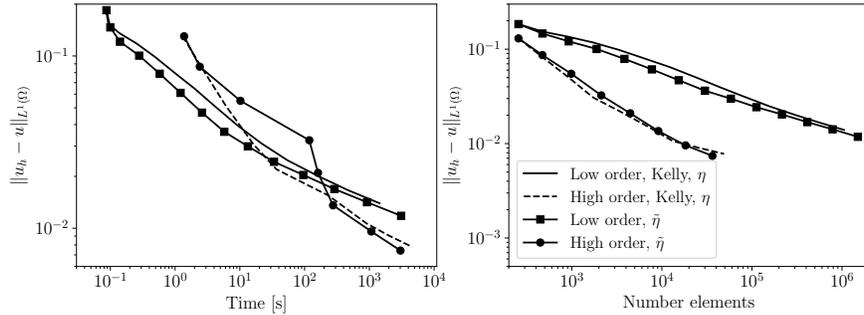


FIGURE 11. Time and elements convergence comparison for the transport problem with a circular convection field, $q = 10$.

In Fig. 13, we depict the refinement evolution for the graph Laplacian indicator ($\tilde{\eta}_K$) for linear and nonlinear stabilization. As expected, we can observe that for the high-order method the scheme is able to resolve the shock with less refinement steps. The linear stabilization is able to provide well-resolved shocks at the final refinement step.

Fig. 14 compares the effectiveness of the low-order and the high-order stabilization schemes for different values of q . The high-order scheme is able to converge efficiently and the overhead of solving a nonlinear problem does not affect the overall performance. In this case, the low-order and the high-order schemes require similar computational time for any given error. Actually, for the finer meshes,

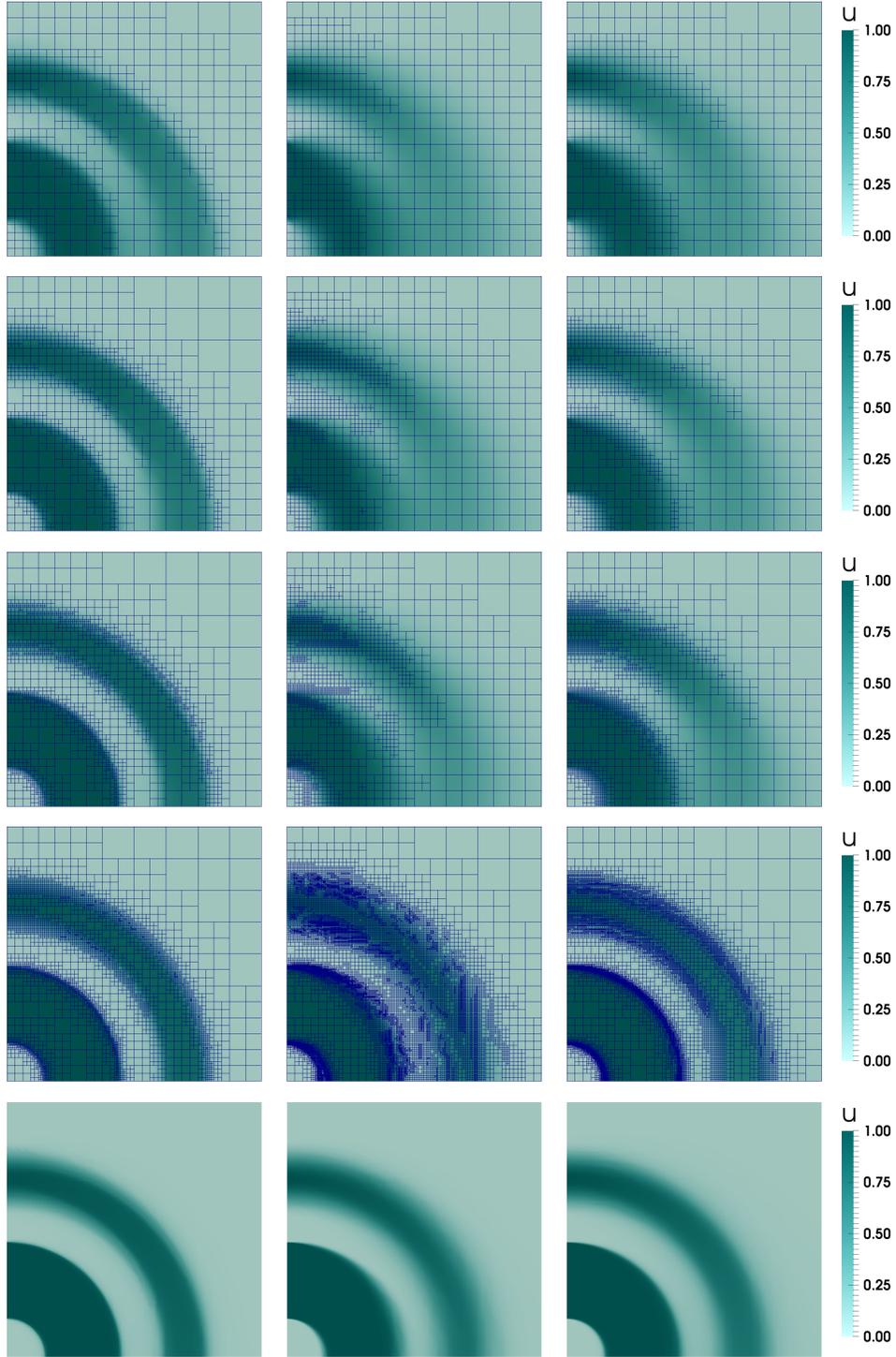


FIGURE 12. Evolution of the mesh refinement process. $\tilde{\eta}_K$ with high-order scheme is used in the left column. Low-order scheme with Kelly estimator is used in the central column. $\tilde{\eta}_K$ with low-order scheme is used in the right column. For the low-order scheme from top to bottom results have been obtained at refinement step 1, 2, 3, 7, and 7. For the high-order refinement steps are 1, 2, 3, 4, and 4.

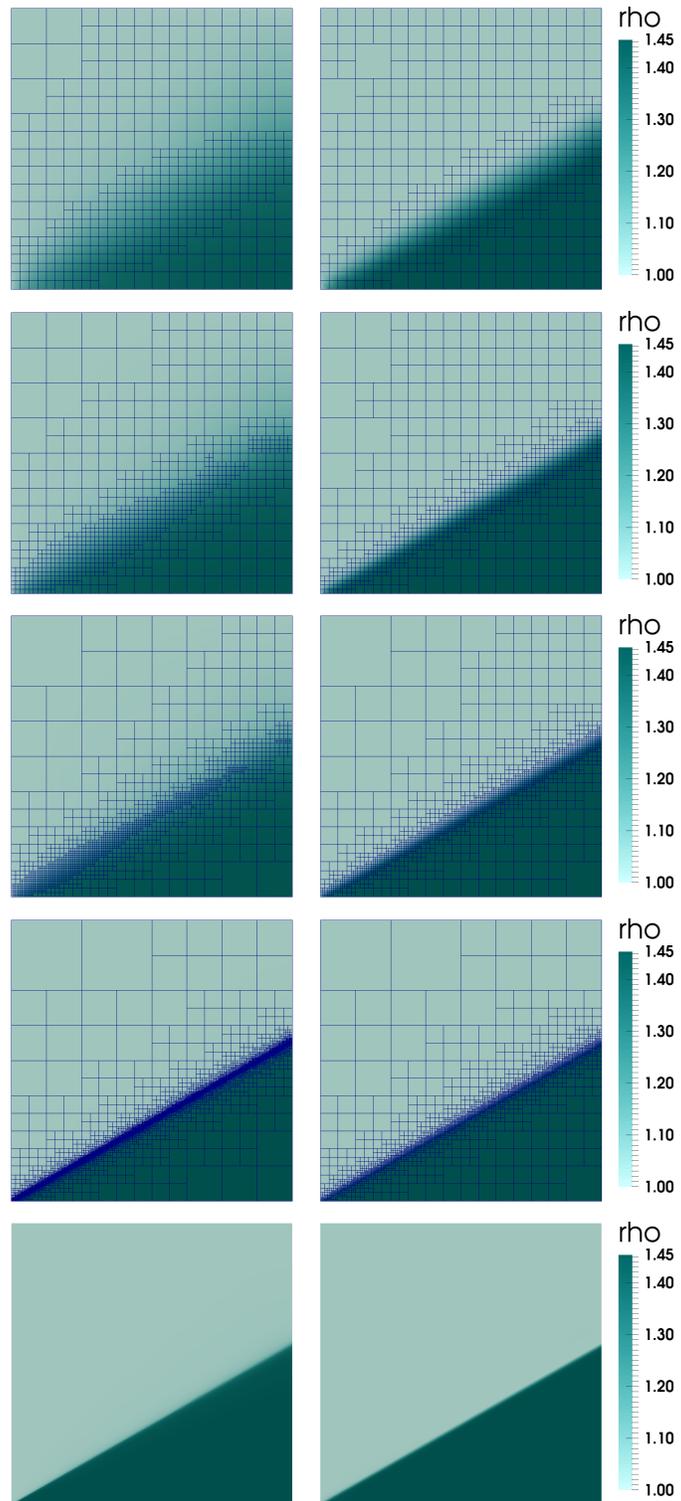


FIGURE 13. Evolution of the mesh refinement process. $\tilde{\eta}_K$ with high-order (right) and low-order (left) schemes are used. For the low-order scheme from top to bottom results have been obtained at refinement step 1, 2, 3, 8, and 8. For the high-order refinement steps are 1, 2, 3, 4, and 4.

the high-order scheme with either $q = 1$ or $q = 2$ already performs better than the low-order scheme. However, for some meshes the high-order scheme exhibits convergence problems. In the case of $q = 10$ the cost of converging the nonlinear problem does not compensate the increase in computational cost.

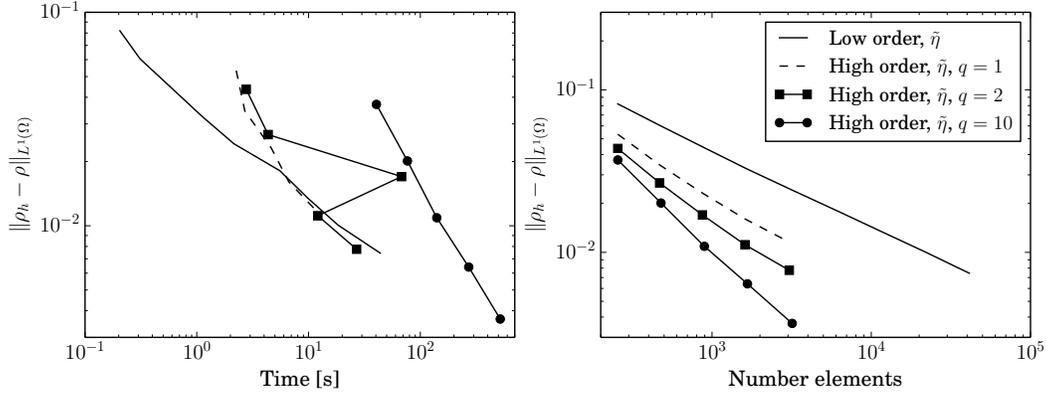


FIGURE 14. Time and elements convergence comparison for the compression corner problem.

6.5. Reflected shock. This benchmark consists in two flow streams colliding at different angles. The domain has dimensions $[0.0, 1.0] \times [0.0, 4.1]$ and a solid wall at its lower boundary. This configuration leads to a steady shock separating both flow regimes that is reflected at the wall producing a third different flow state behind it. A sketch of this benchmark test is given in Fig. 15. The flow states at each region have been collected in Tab. 1.

TABLE 1. Reflected shock solution values at every region.

Region	Density [Kg m^{-3}]	Velocity [m s^{-1}]	Total energy [J]
Ⓐ	1.0	(2.9, 0.0)	5.99075
Ⓑ	1.7	(2.62, -0.506)	5.8046
Ⓒ	2.687	(2.401, 0.0)	5.6122

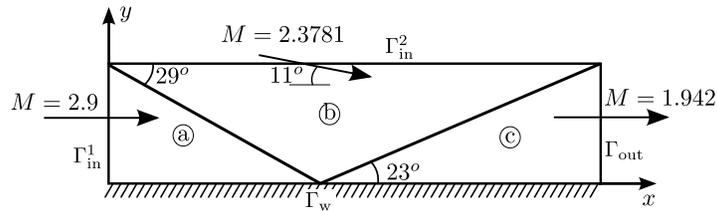


FIGURE 15. Reflected shock scheme.

We analyze the effectiveness of the high-order scheme, and evaluate the performance of the graph Laplacian refinement strategy. We start with a coarse mesh of 16×64 elements and adapt the mesh till a certain number of elements is reached. For the high-order method, we set a maximum of 10^4 elements. The maximum number of elements for the low-order method is $3 \cdot 10^5$. We use a nonlinear tolerance of $\|\Delta \mathbf{u}_h\| / \|\mathbf{u}_h\| < 10^{-4}$ and a maximum of 500 iterations.

Fig. 16 compares the effectiveness of the low-order and the high-order stabilization schemes for different values of q . The high-order scheme converges efficiently and the overhead of solving a nonlinear

problem does not affect the overall performance. Actually, for the most refined meshes the high-order method is more efficient than the low-order one. As for the previous problem, Fig. 16 shows that the high-order scheme can present nonlinear convergence problems at some steps of the refinement process. However, as the mesh becomes more adapted to the problem this issue is reduced.

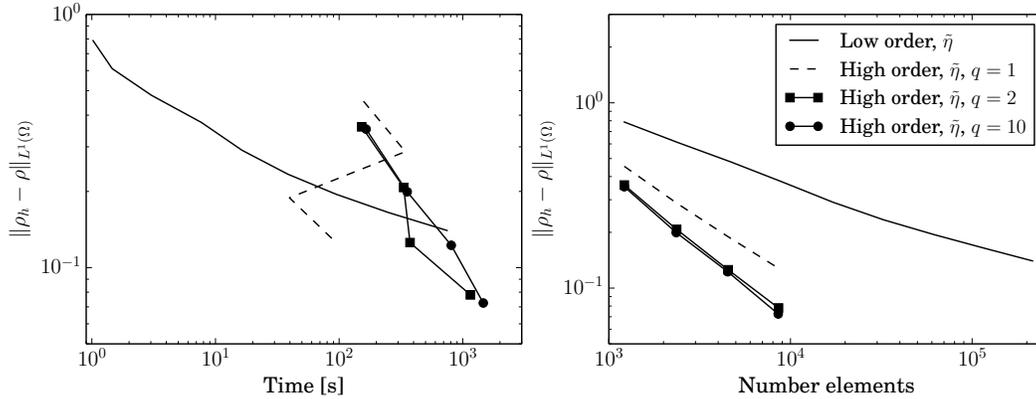


FIGURE 16. Time and elements convergence comparison for the reflected shock problem.

In Fig. 17 we depict the refinement evolution for the graph Laplacian indicator ($\tilde{\eta}_K$) for the low-order scheme. In these figures it can be observed how the graph Laplacian strategy is able to concentrate all the resolution at the shock location. Finally, we can conclude from the lower two figures that both schemes resolve the shocks properly after the mesh has been refined enough.

7. CONCLUSIONS

The stabilization schemes in [3, 5] have been extended and assessed in the AMR context for nonconforming hierarchical octree meshes. The work focuses in assessing the effectiveness of linear (first-order) and nonlinear (higher-order) stabilization. We focus the comparison in terms of accuracy versus computational time.

The results indicate that linear stabilization is more effective for coarse meshes. In this case, the computational cost required to solve the stiff nonlinear problem due to the nonlinear stabilization does not compensate the improvement in the accuracy. This is especially evident for linear systems of partial differential equations (PDEs). On the contrary, as the mesh is refined and properly adapted to the shocks, nonlinear stabilization pays the price. Even though increasing the value of q in the nonlinear stabilization (a parameter that makes shocks sharper but hinders nonlinear convergence) improves accuracy, it turns to be more effective to refine the mesh further for low values of q . Nevertheless, it is worth mentioning that high-order method might exhibit nonlinear convergence problems for some meshes.

In addition, a new refinement strategy have been proposed. The proposed indicator is based on the graph Laplacian used in the definition of the stabilization method. Numerical results show that this shock detector is able to perform better than the well known Kelly estimator for problems with shocks or discontinuities.

ACKNOWLEDGMENTS

J. Bonilla gratefully acknowledges the support received from "la Caixa" Foundation through its PhD scholarship program (LCF/BQ/DE15/10360010). S. Badia gratefully acknowledges the support received from the Catalan Government through the ICREA Acadèmia Research Program. We acknowledge the financial support to CIMNE via the CERCA Programme / Generalitat de Catalunya.

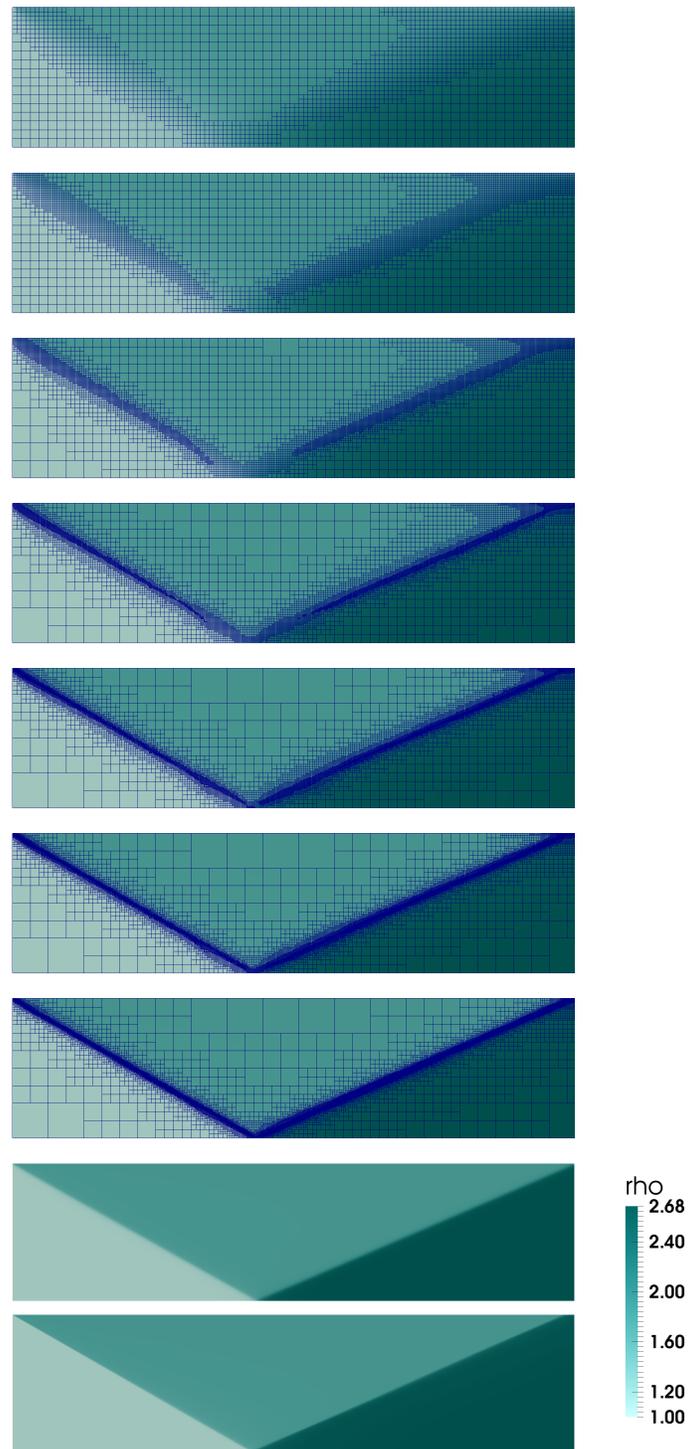


FIGURE 17. Evolution of the mesh refinement process. $\tilde{\eta}_K$ with low-order scheme is used. For the low-order scheme from top to bottom results have been obtained at refinement step 1, 2, 3, 4, 5, 6, and 7. The lower two figures are the high-order (top) and low-order (bottom) results at their last refinement step.

REFERENCES

- [1] M. AINSWORTH AND J. TINSLEY ODEN, *A posteriori error estimation in finite element analysis*, Computer Methods in Applied Mechanics and Engineering, 142 (1997), pp. 1–88.
- [2] J. D. ANDERSON JR., *Modern Compressible Flow*, McGraw-Hill, 2nd ed., 1990.
- [3] S. BADIA AND J. BONILLA, *Monotonicity-preserving finite element schemes based on differentiable nonlinear stabilization*, Computer Methods in Applied Mechanics and Engineering, 313 (2017), pp. 133–158.
- [4] S. BADIA, J. BONILLA, AND A. HIERRO, *Differentiable monotonicity-preserving schemes for discontinuous Galerkin methods on arbitrary meshes*, Computer Methods in Applied Mechanics and Engineering, 320 (2017), pp. 582–605.
- [5] S. BADIA, J. BONILLA, S. MABUZA, AND J. N. SHADID, *Differentiable local bounds preserving stabilization for first order hyperbolic problems*, Submitted, (2019).
- [6] S. BADIA AND A. HIERRO, *On Monotonicity-Preserving Stabilized Finite Element Approximations of Transport Problems*, SIAM Journal on Scientific Computing, 36 (2014), pp. A2673–A2697.
- [7] S. BADIA AND A. HIERRO, *On discrete maximum principles for discontinuous Galerkin methods*, Computer Methods in Applied Mechanics and Engineering, 286 (2015), pp. 107–122.
- [8] S. BADIA AND A. F. MARTÍN, *A tutorial-driven introduction to the parallel finite element library FEMPAR v1.0.0*, (2019).
- [9] S. BADIA, A. F. MARTÍN, E. NEIVA, AND F. VERDUGO, *A generic finite element framework on parallel tree-based adaptive meshes*, Submitted, (2019).
- [10] S. BADIA, A. F. MARTÍN, AND J. PRINCIPE, *FEMPAR: An Object-Oriented Parallel Finite Element Framework*, Archives of Computational Methods in Engineering, 25 (2018), pp. 195–271.
- [11] W. BANGERTH, C. BURSTEDDE, T. HEISTER, AND M. KRONBICHLER, *Algorithms and data structures for massively parallel generic adaptive finite element codes*, ACM Trans. Math. Softw., 38 (2012), pp. 14:1–14:28.
- [12] G. R. BARRENECHEA, E. BURMAN, AND F. KARAKATSANI, *Edge-based nonlinear diffusion for finite element approximations of convection-diffusion equations and its relation to algebraic flux-correction schemes*, Numerische Mathematik, (2016), pp. 1–25.
- [13] G. R. BARRENECHEA, V. JOHN, AND P. KNOBLOCH, *Analysis of Algebraic Flux Correction Schemes*, SIAM Journal on Numerical Analysis, 54 (2016), pp. 2427–2451.
- [14] M. BITTL AND D. KUZMIN, *An hp-adaptive flux-corrected transport algorithm for continuous finite elements*, Computing, 95 (2013), pp. 27–48.
- [15] J. BONILLA AND S. BADIA, *Maximum-principle preserving space-time isogeometric analysis*, Computer Methods in Applied Mechanics and Engineering, 354 (2019), pp. 422–440.
- [16] J. BONILLA, S. MABUZA, J. N. SHADID, AND S. BADIA, *On Differentiable Linearity and Local Bounds Preserving Stabilization Methods for First Order Conservation Law Systems*, in Center for Computing Research Summer Proceedings 2018, A. Cangı and M. L. Parks, eds., Sandia National Laboratories, 2018, pp. 107–119.
- [17] E. BURMAN, *Adaptive finite element methods for compressible flow*, Computer Methods in Applied Mechanics and Engineering, 190 (2000), pp. 1137–1162.
- [18] E. BURMAN AND A. ERN, *Nonlinear diffusion and discrete maximum principle for stabilized Galerkin approximations of the convection-diffusion-reaction equation*, Computer Methods in Applied Mechanics and Engineering, 191 (2002), pp. 3833–3855.
- [19] B. COCKBURN AND C.-W. SHU, *Runge-Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems*, Journal of Scientific Computing, 16 (2001), pp. 173–261.
- [20] R. CODINA, *A discontinuity-capturing crosswind-dissipation for the finite element solution of the convection-diffusion equation*, Computer Methods in Applied Mechanics and Engineering, 110 (1993), pp. 325–342.
- [21] L. DEMKOWICZ, *Computing with hp-ADAPTIVE FINITE ELEMENTS: One and Two Dimensional Elliptic and Maxwell Problems*, vol. 1, CRC Press, oct 2006.
- [22] K. ERIKSSON, D. ESTEP, P. HANSBO, AND C. JOHNSON, *Introduction to Adaptive Methods for*

- Differential Equations*, Acta Numerica, 4 (1995), pp. 105–158.
- [23] M. M. FEISTAUER, J. J. FELCMAN, AND I. I. STRAŠKRABA, *Mathematical and computational methods for compressible flow*, Oxford University Press, 2003.
- [24] C. FLETCHER, *The group finite element formulation*, Computer Methods in Applied Mechanics and Engineering, 37 (1983), pp. 225–244.
- [25] S. GODUNOV, *Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics*, Matematicheskii Sbornik, Steklov Mathematical Institute of Russian Academy of Sciences, 47(89) (1959), pp. 271–306.
- [26] S. GOTTLIEB, C.-W. SHU, AND E. TADMOR, *Strong Stability-Preserving High-Order Time Discretization Methods*, SIAM Review, 43 (2001), pp. 89–112.
- [27] J.-L. GUERMOND, M. NAZAROV, B. POPOV, AND Y. YANG, *A Second-Order Maximum Principle Preserving Lagrange Finite Element Technique for Nonlinear Scalar Conservation Equations*, SIAM Journal on Numerical Analysis, 52 (2014), pp. 2163–2182.
- [28] J.-L. GUERMOND AND B. POPOV, *Invariant domains and first-order continuous finite element approximation for hyperbolic systems*, (2015), pp. 1–22.
- [29] M. GURRIS, *Implicit finite element schemes for compressible gas and particle-laden gas flows*, PhD thesis, Technische Universität Dortmund, 2009.
- [30] C. JOHNSON AND A. SZEPESSY, *Adaptive finite element methods for conservation laws based on a posteriori error estimates*, Communications on Pure and Applied Mathematics, 48 (1995), pp. 199–234.
- [31] D. W. KELLY, J. P. DE S. R. GAGO, O. C. ZIENKIEWICZ, AND I. BABUSKA, *A posteriori error analysis and adaptive processes in the finite element method: Part I-error analysis*, International Journal for Numerical Methods in Engineering, 19 (1983), pp. 1593–1619.
- [32] A. KRITZ AND D. KEYES, *Fusion Simulation Project Workshop Report*, Journal of Fusion Energy, 28 (2009), pp. 1–59.
- [33] D. KUZMIN, *Linearity-preserving flux correction and convergence acceleration for constrained Galerkin schemes*, Journal of Computational and Applied Mathematics, 236 (2012), pp. 2317–2337.
- [34] ———, *Monolithic convex limiting for continuous finite element discretizations of hyperbolic conservation laws*, Computer Methods in Applied Mechanics and Engineering, 361 (2020), p. 112804.
- [35] D. KUZMIN, S. BASTING, AND J. N. SHADID, *Linearity-preserving monotone local projection stabilization schemes for continuous finite elements*, Computer Methods in Applied Mechanics and Engineering, 322 (2017), pp. 23–41.
- [36] D. KUZMIN, R. LÖHNER, AND S. TUREK, *Flux-corrected transport*, Springer, 2005.
- [37] D. KUZMIN AND M. MÖLLER, *Algebraic Flux Correction I. Scalar Conservation Laws*, in Flux-Corrected Transport, D. D. Kuzmin, P. R. Löhner, and P. D. S. Turek, eds., Scientific Computation, Springer Berlin Heidelberg, jan 2005, pp. 155–206.
- [38] D. KUZMIN, M. MÖLLER, AND M. GURRIS, *Algebraic Flux Correction II. Compressible flows*, in Flux-corrected Transport: Principles, Algorithms, and Applications, 2012, pp. 193–238.
- [39] D. KUZMIN, M. MÖLLER, AND S. TUREK, *Multidimensional FEM-FCT schemes for arbitrary time stepping*, International Journal for Numerical Methods in Fluids, 42 (2003), pp. 265–295.
- [40] D. KUZMIN, M. QUEZADA DE LUNA, C. E. KEES, D. KUZMIN, M. QUEZADA DE LUNA, AND C. E. KEES, *A partition of unity approach to adaptivity and limiting in continuous finite element methods*, (2018).
- [41] D. KUZMIN AND S. TUREK, *Flux Correction Tools for Finite Elements*, Journal of Computational Physics, 175 (2002), pp. 525–558.
- [42] R. J. LEVEQUE, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, Cambridge, 2002.
- [43] C. LOHMANN AND D. KUZMIN, *Synchronized flux limiting for gas dynamics variables*, Journal of Computational Physics, 326 (2016), pp. 973–990.

- [44] C. LOHMANN, D. KUZMIN, J. N. SHADID, AND S. MABUZA, *Flux-corrected transport algorithms for continuous Galerkin methods based on high order Bernstein finite elements*, Journal of Computational Physics, 344 (2017), pp. 151–186.
- [45] R. LÖHNER, *An adaptive finite element scheme for transient problems in CFD*, Computer Methods in Applied Mechanics and Engineering, 61 (1987), pp. 323–338.
- [46] R. LOHNER, *Applied Computational Fluid Dynamics Techniques: An Introduction Based on Finite Element Methods*, vol. 508, 2004.
- [47] S. MABUZA, J. N. SHADID, E. C. CYR, R. P. PAWLOWSKI, AND D. KUZMIN, *A linearity preserving nodal variation limiting algorithm for continuous Galerkin discretization of ideal MHD equations*, Journal of Computational Physics, In press (2020).
- [48] S. MABUZA, J. N. SHADID, AND D. KUZMIN, *Local bounds preserving stabilization for continuous Galerkin discretization of hyperbolic systems*, Journal of Computational Physics, 361 (2018), pp. 82–110.
- [49] M. MÖLLER AND D. KUZMIN, *Adaptive mesh refinement for high-resolution finite element schemes*, International Journal for Numerical Methods in Fluids, 52 (2006), pp. 545–569.
- [50] M. NAZAROV, J.-L. GUERMOND, AND B. POPOV, *A posteriori error estimation for the compressible Euler equations using entropy viscosity*, tech. rep., 2011.
- [51] M. NAZAROV AND J. HOFFMAN, *An adaptive finite element method for inviscid compressible flow*, International Journal for Numerical Methods in Fluids, 64 (2010), pp. 1102–1128.
- [52] F. SHAKIB, T. J. R. HUGHES, AND Z. JOHAN, *A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier-Stokes equations*, Computer Methods in Applied Mechanics and Engineering, 89 (1991), pp. 141–219.
- [53] E. SÜLI, *A Posteriori Error Analysis and Adaptivity for Finite Element Approximations of Hyperbolic Problems*, (1999), pp. 123–194.
- [54] T. E. TEZDUYAR AND M. SENGA, *Stabilization and shock-capturing parameters in SUPG formulation of compressible flows*, Computer Methods in Applied Mechanics and Engineering, 195 (2006), pp. 1621–1632.
- [55] T. TIANKAI TU, D. O’HALLARON, AND O. GHATTAS, *Scalable Parallel Octree Meshing for TeraScale Applications*, in ACM/IEEE SC 2005 Conference (SC’05), IEEE, 2005, pp. 4–4.
- [56] A. TIKHONOVA, G. TANASE, O. TKACHYSHYN, N. M. AMATO, AND L. RAUCHWERGER, *Parallel Algorithms in STAPL: Sorting and the Selection Problem*, tech. rep., 2005.
- [57] E. F. TORO, *Riemann Solvers and Numerical Methods for Fluid Dynamics*, 3rd ed., 2009.
- [58] R. VERFURTH, *A Posteriori Error Estimation Techniques for Finite Element Methods*, Oxford University Press, 2013.
- [59] O. C. ZIENKIEWICZ AND J. Z. ZHU, *A simple error estimator and adaptive procedure for practical engineering analysis*, International Journal for Numerical Methods in Engineering, 24 (1987), pp. 337–357.
- [60] O. C. ZIENKIEWICZ AND J. Z. ZHU, *The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique*, International Journal for Numerical Methods in Engineering, 33 (1992), pp. 1331–1364.