

# INVOLUTION-PRESERVING SCHEMES THROUGH CONSTRAINED MESH MOTION: A NOVEL ARBITRARY-LAGRANGIAN-EULERIAN FRAMEWORK

SACHA CARDONNA\*, ELENA BERNARDELLI†, AND WALTER BOSCHERI‡

**Abstract.** We introduce a novel structure-preserving finite volume method that exploits mesh motion to enforce differential involutions arising in hyperbolic systems of conservation laws. The key idea is to leverage the geometric degrees of freedom of an Arbitrary-Lagrangian–Eulerian (ALE) framework to construct a constrained mesh motion that is intrinsically compatible with the involution constraints embedded in the governing equations. The method is developed within a cell-centered formulation on moving quadrilateral meshes, where discrete divergence and curl operators are constructed from vertex-based differential operators. Particular care is devoted to the consistent evolution of the mesh geometry and the satisfaction of the Geometric Conservation Law (GCL).

The proposed framework is applied to two representative hyperbolic systems featuring distinct involution constraints: the ideal magnetohydrodynamics equations, endowed with the divergence-free condition on the magnetic field, and the hyperbolic Euler–Heat equations, subject to a curl-free constraint on the thermal impulse. Second-order accuracy in time is achieved through a Runge–Kutta discretization, with the geometric constraints consistently enforced at each stage. A comprehensive set of classical benchmark problems is presented to assess the accuracy and effectiveness of the method. The numerical results demonstrate that appropriately constrained mesh motion provides an effective mechanism for preserving differential involutions, establishing a novel connection between mesh geometry and the structure-preserving discretization of hyperbolic systems.

**Key words.** structure-preserving schemes; involution preservation; continuum mechanics; moving unstructured meshes; constrained mesh motion

**AMS subject classifications.** 35L40, 65M08.

**1. Introduction.** Many systems arising in continuum mechanics are endowed with additional geometric and structural constraints, commonly referred to as involutions. Among these, divergence-free and curl-free conditions play a particularly important role. Such constraints arise in several physical models, and their preservation at the discrete level is particularly important, since violations may deteriorate the quality of the numerical solution, lead to numerical instabilities or finite-time blow-up [8, 9], and result in the loss of relevant physical properties [1, 6, 7, 3, 19]. Here, we consider two representative hyperbolic systems: (i) the equations of magnetohydrodynamics (MHD), which describe the dynamics of electrically conducting fluids, such as plasma, through the coupling of fluid motion and electromagnetic fields; and (ii) the first-order hyperbolic Euler–Heat model for heat-conducting compressible fluids introduced in [10, 4], which extends the compressible Euler equations to incorporate heat conduction through an additional vector field, the thermal impulse  $\mathbf{j}$ . For the MHD equations, the relevant involution is the solenoidal constraint on the magnetic field,  $\nabla \cdot \mathbf{B} = 0$ , which follows from Maxwell equations and expresses the absence of magnetic monopoles. For the hyperbolic Euler–Heat model, the evolution equation for the thermal impulse has a gradient structure, which implies the curl-free constraint  $\nabla \times \mathbf{j} = \mathbf{0}$ . If satisfied at the initial time, these conditions are preserved by the con-

---

\*Institut Montpellierain Alexander Grothendieck UMR 5149, 34090 Montpellier, France ([sacha.cardonna@umontpellier.fr](mailto:sacha.cardonna@umontpellier.fr))

†Department of Computer Science, University of Verona, 37134 Verona, Italy ([elena.bernardelli@univr.it](mailto:elena.bernardelli@univr.it))

‡Laboratoire de Mathématiques UMR 5127 CNRS, Université Savoie Mont Blanc, 73376 Le Bourget du Lac, France ([walter.boscheri@univ-smb.fr](mailto:walter.boscheri@univ-smb.fr)) and Department of Mathematics and Computer Science, University of Ferrara, via Machiavelli 30, 44121 Ferrara, Italy

tinuous evolution of the systems. Satisfying these structures at the discrete level of the numerical method remains an open challenge.

The numerical approximation of hyperbolic systems on moving computational meshes is naturally formulated within the Arbitrary Lagrangian–Eulerian (ALE) framework [12, 13, 18, 16, 2]. In contrast to the Eulerian description, where the computational mesh remains fixed, and the Lagrangian description, where the mesh moves with the local fluid velocity, ALE methods allow the grid nodes to move with an arbitrary prescribed velocity. This additional degree of freedom combines the advantages of Eulerian and Lagrangian approaches while providing greater control over the evolution and quality of the computational mesh. In particular, by adapting the mesh motion to the underlying flow, ALE schemes can reduce the numerical dissipation associated with convective transport and improve the resolution of moving discontinuities, contact waves, and material interfaces. At the same time, the freedom in the choice of the mesh velocity can be exploited to limit excessive mesh deformation, which represents one of the main difficulties of purely Lagrangian approaches. A fundamental requirement of numerical schemes on moving meshes is the satisfaction of the Geometric Conservation Law (GCL), which ensures that the evolution of the computational geometry is consistently coupled with the discrete governing equations. In particular, satisfaction of the GCL is closely related to the preservation of uniform flow under arbitrary mesh motion and is essential to avoid errors generated solely by the deformation of the computational grid. While the GCL guarantees consistency between the numerical solution and the evolving geometry, additional difficulties arise when the governing PDE system is subject to differential involutions.

In this work, we introduce a novel involution-preserving ALE finite volume approach based on constrained mesh motion. The key idea is to exploit the freedom in the choice of the ALE mesh velocity and determine it through a constrained problem that accounts for the differential structure of the underlying physical system. The proposed method is formulated in a cell-centered finite volume framework on moving structured quadrilateral meshes. The discrete differential operators are constructed from corner-based geometrical quantities, providing a natural discrete representation of the divergence and curl operators on the evolving mesh. The computational mesh is consistently updated during the mesh evolution, while a second-order Runge–Kutta method is employed for the time discretization. The mesh motion is constrained with the aim of preserving the involution over time, according to the corresponding differential structure, either the divergence or the curl operator.

The outline of the paper as follows. In Section 2 we introduce the two systems of partial differential equations treated in this paper as representative examples. In Section 3 we present the geometrical and discrete ingredients used by the novel approach, such as the corner vectors and the discrete operators to compute the involutions. Section 4 describes the fully discrete structure-preserving mesh motion. In Section 5, a validation through classical benchmark test cases is shown. Conclusions and an outlook to further work are given in Section 6.

**2. Governing equations and continuous involutions.** Let  $\Omega \subset \mathbb{R}^2$  be a bounded domain with boundary  $\partial\Omega$ , let  $\mathbf{x} = (x, y) \in \Omega$  denote the spatial coordinates, and let  $t \in \mathbb{R}_+$  be the time coordinate. The governing equations can be written in the general form

$$(2.1) \quad \frac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{q}) = \mathbf{0},$$

where  $\mathbf{q} \in \Omega_{\mathbf{q}}$  is the vector of conserved variables defined in the set of admissible states  $\Omega_{\mathbf{q}}$ , and  $\mathbb{F}(\mathbf{q})$  is the nonlinear flux tensor. Vectors are denoted by bold symbols and double-stroke symbols are used for tensors.

**Ideal magnetohydrodynamics (MHD).** The MHD equations can be cast into the formulation (2.1) by defining the state vector and the flux tensor

$$(2.2) \quad \mathbf{q} = \begin{pmatrix} \rho \\ \rho \mathbf{u} \\ \rho E \\ \mathbf{B} \end{pmatrix}, \quad \mathbb{F}(\mathbf{q}) = \begin{pmatrix} \rho \mathbf{u} \\ \rho \mathbf{u} \otimes \mathbf{u} + \left(p + \frac{\|\mathbf{B}\|^2}{8\pi}\right) \mathbb{I}_2 - \frac{1}{4\pi} \mathbf{B} \otimes \mathbf{B} \\ \left(\rho E + p + \frac{\|\mathbf{B}\|^2}{8\pi}\right) \mathbf{u} - \frac{1}{4\pi} \mathbf{B}(\mathbf{u} \cdot \mathbf{B}) \\ \mathbf{B} \otimes \mathbf{u} - \mathbf{u} \otimes \mathbf{B} \end{pmatrix}.$$

Here,  $\rho$  is the fluid density,  $\mathbf{u} = (u, v)$  is the velocity,  $p$  is the pressure,  $\rho E$  is the total energy density, and  $\mathbf{B} = (B_x, B_y)$  is the magnetic field. The system is closed with the ideal gas equation of state, thus the the total energy density is given by

$$(2.3) \quad \rho E = \rho e + \frac{1}{2} \rho \|\mathbf{u}\|^2 + \frac{\|\mathbf{B}\|^2}{8\pi}, \quad \rho e = \frac{p}{\gamma - 1},$$

with  $\gamma > 1$  representing the ratio of specific heats of the gas. The magnetic field is subject to the solenoidal constraint  $\nabla \cdot \mathbf{B} = 0$ , which must be preserved over time.

**Hyperbolic Euler–Heat model (EH).** The EH model is given by defining

$$(2.4) \quad \mathbf{q} = \begin{pmatrix} \rho \\ \rho \mathbf{u} \\ \rho E \\ \mathbf{j} \end{pmatrix}, \quad \mathbb{F}(\mathbf{q}) = \begin{pmatrix} \rho \mathbf{u} \\ \rho \mathbf{u} \otimes \mathbf{u} + (p - \alpha(\rho) \|\mathbf{j}\|^2) \mathbb{I}_2 + \alpha(\rho) \mathbf{j} \otimes \mathbf{j} \\ (\rho E + p - \alpha(\rho) \|\mathbf{j}\|^2) \mathbf{u} + \alpha(\rho) \phi \mathbf{j} \\ \phi \mathbb{I}_2 \end{pmatrix},$$

with  $\phi = \mathbf{j} \cdot \mathbf{u} + \theta$  and the thermal impulse  $\mathbf{j} = (j_x, j_y)$ . The system is closed by the constitutive relations

$$(2.5) \quad \alpha(\rho) = \frac{\kappa^2}{\rho}, \quad \rho E = \frac{1}{2} \rho \|\mathbf{u}\|^2 + \frac{p}{\gamma - 1} + \frac{1}{2} \alpha(\rho) \|\mathbf{j}\|^2, \quad \theta = \frac{p}{\rho c_v (\gamma - 1)},$$

with  $\kappa \geq 0$ ,  $c_v > 0$ , and  $\gamma > 1$ . For this choice of  $\alpha$ , the general isotropic impulse contribution  $\frac{1}{2}(\rho \alpha'(\rho) - \alpha(\rho)) \|\mathbf{j}\|^2$  reduces to  $-\alpha(\rho) \|\mathbf{j}\|^2$ . The thermal impulse obeys the involution  $\nabla \times \mathbf{j} = \mathbf{0}$ .

**REMARK 2.1.** *The complete EH model also contains the rotational term  $[\nabla \mathbf{j} - (\nabla \mathbf{j})^\top] \mathbf{u}$ , which vanishes on the curl-free manifold, which is considered here. Furthermore, a source term of the form  $-\mathbf{j}/\tau$  might be added to the EH system with a constant relaxation parameter  $\tau$ , which still implies a curl-free manifold.*

**3. Arbitrary-Eulerian-Lagrangian (ALE) finite volume scheme.** At time  $t$ , the domain is covered by a conforming tessellation  $\mathcal{T}_h(t) = \{\omega_c(t)\}_{c=1}^{N_c}$  made of  $N_c$  non-overlapping time-dependent control volumes  $\omega_c(t)$  with boundary  $\partial\omega_c(t)$  and outward pointing normal vector  $\mathbf{n}$ . Let  $N_p$  be the total number of vertices. We define the complete, boundary, and interior nodal index sets by

$$(3.1) \quad n_h := \llbracket 1, N_p \rrbracket, \quad n_h^\partial := \{p \in n_h \mid \mathbf{x}_p(t) \in \partial\Omega\}, \quad n_h^\circ := n_h \setminus n_h^\partial.$$

The set of vertices of  $\omega_c$  is denoted by  $\mathcal{P}(c) \subset \mathcal{N}_h$ , while  $\mathcal{C}(p)$  is the set of cells sharing vertex  $p$ ;  $\sharp\mathcal{P}(c)$  denotes the cardinality of  $\mathcal{P}(c)$ . The position and grid velocity of vertex  $p \in \mathcal{N}_h$  are denoted by  $\mathbf{x}_p(t)$  and  $\mathbf{w}_p(t)$ , respectively. Let us introduce  $\mathbf{X} := (\mathbf{x}_p)_{p \in \mathcal{N}_h}$  to represent the complete vector of nodal coordinates. The mesh topology is kept fixed during the computation; only the vertex coordinates evolve.

**DEFINITION 3.1** (Corner geometry). *Let  $p^-$  and  $p^+$  be the previous and next vertices of  $p$  along the counter-clockwise oriented boundary of  $\omega_c$ . The two edges adjacent to the corner  $(p, c)$  have lengths  $\ell_{pc}^-$  and  $\ell_{pc}^+$  and outward unit normals  $\mathbf{n}_{pc}^-$  and  $\mathbf{n}_{pc}^+$ . The associated corner vector is*

$$(3.2) \quad \boldsymbol{\ell}_{pc} = \frac{1}{2} (\ell_{pc}^- \mathbf{n}_{pc}^- + \ell_{pc}^+ \mathbf{n}_{pc}^+).$$

*It contains the two half-edge contributions attached to vertex  $p$  and satisfies the discrete Gauss theorem*

$$(3.3) \quad \sum_{p \in \mathcal{P}(c)} \boldsymbol{\ell}_{pc} = \mathbf{0}.$$

*We further introduce the tangent corner vector obtained by a counter-clockwise rotation, that is*

$$(3.4) \quad \mathbf{t}_{pc} := \boldsymbol{\ell}_{pc}^\perp = \begin{pmatrix} -(\boldsymbol{\ell}_{pc})_2 \\ (\boldsymbol{\ell}_{pc})_1 \end{pmatrix}.$$

**DEFINITION 3.2** (Dual cell and subcells). *The dual cell  $\omega_p$  associated with a mesh vertex  $p$  is constructed by connecting the barycenters of the cells sharing  $p$  with the midpoints of the edges incident to  $p$ , thereby defining a vertex-centered polygonal control volume surrounding  $p$ . Each primal cell is further subdivided into subcells  $\omega_{pc}$ , each bounded by the cell barycenter, a vertex  $p$ , and the midpoints of the two cell edges incident to  $p$ . The volume of the dual cell is then given by*

$$(3.5) \quad |\omega_p| = \sum_{c \in \mathcal{C}(p)} |\omega_{pc}|,$$

*and the fundamental geometry identity (3.3) holds true on the dual cell, that is*

$$(3.6) \quad \sum_{c \in \mathcal{C}(p)} \boldsymbol{\ell}_{cp} = \mathbf{0},$$

*with  $\boldsymbol{\ell}_{cp} = -\boldsymbol{\ell}_{pc}$  being the corner normal vector of the dual cell.*

We design an ALE finite volume method closely related to the framework used in cell-centered Lagrangian schemes [17, 5]. Within each cell  $\omega_c$ , the solution is stored at time  $t$  as cell-averages given by

$$(3.7) \quad \mathbf{q}_c = \frac{1}{|\omega_c|} \int_{\omega_c} \mathbf{q}(\mathbf{x}) \, d\mathbf{x}.$$

The cell averages are transferred to the dual cells through the volume-weighted average

$$(3.8) \quad \mathbf{q}_p = \frac{1}{|\omega_p|} \sum_{c \in \mathcal{C}(p)} |\omega_{pc}| \mathbf{q}_c.$$

The semi-discrete ALE finite volume scheme writes

$$(3.9) \quad \frac{d}{dt} (|\omega_c| \mathbf{q}_c) + \sum_{p \in \mathcal{P}(c)} \widehat{\mathbb{F}}_p \boldsymbol{\ell}_{pc} = \mathbf{0},$$

with the nodal ALE flux given by the following multidimensional Riemann solver

$$(3.10) \quad \widehat{\mathbb{F}}_p = \mathbb{F}_p - \frac{1}{2} \sqrt{|\omega_p|} a_p \mathbb{G}_p, \quad \mathbb{F}_p = \mathbb{F}(\mathbf{q}_p) - \mathbf{q}_p \otimes \boldsymbol{\omega}_p.$$

The nodal gradient is computed as

$$(3.11) \quad \mathbb{G}_p := -\frac{1}{|\omega_p|} \sum_{c \in \mathcal{C}(p)} \mathbf{q}_c \otimes \boldsymbol{\ell}_{pc}.$$

The dissipation coefficient  $a_p$  is chosen such that

$$(3.12) \quad a_p \geq \max_{\substack{c \in \mathcal{C}(p) \\ k \in \{1,2\}}} \rho_{\text{sp}}(\mathbb{J}_{F,k}(\mathbf{q}_c) - \boldsymbol{\omega}_{p,k} \mathbb{I}), \quad k \in \{1,2\},$$

where  $\rho_{\text{sp}}(\cdot)$  denotes the spectral radius of the flux Jacobian

$$(3.13) \quad \mathbb{J}_{F,k}(\mathbf{q}) = \frac{\partial(\mathbb{F}(\mathbf{q}) \mathbf{e}_k)}{\partial \mathbf{q}}, \quad \mathbf{e}_1 = (1, 0)^\top, \quad \mathbf{e}_2 = (0, 1)^\top.$$

For later use, we denote the integrated spatial residual by

$$(3.14) \quad \mathbf{H}_c := - \sum_{p \in \mathcal{P}(c)} \widehat{\mathbb{F}}_p \boldsymbol{\ell}_{pc}, \quad \text{so that} \quad \frac{d}{dt} (|\omega_c| \mathbf{q}_c) = \mathbf{H}_c.$$

The evolution of the cell volume is linked to the nodal grid velocities through the geometric conservation law (GCL):

$$(3.15) \quad \frac{d}{dt} |\omega_c| = \sum_{p \in \mathcal{P}(c)} \boldsymbol{\omega}_p \cdot \boldsymbol{\ell}_{pc}.$$

**PROPERTY 3.3** (Conservation and preservation of constant states). *On a periodic mesh, the semi-discrete ALE scheme (3.9) is globally conservative. Furthermore, if (3.15) is satisfied, it preserves every spatially constant admissible state.*

*Proof.* Summing (3.9) over all cells, exchanging the cell and node sums, and using (3.6), we obtain

$$\frac{d}{dt} \sum_c |\omega_c| \mathbf{q}_c = - \sum_{p \in \mathcal{N}_h} \widehat{\mathbb{F}}_p \sum_{c \in \mathcal{C}(p)} \boldsymbol{\ell}_{pc} = \mathbf{0}.$$

For a constant state  $\bar{\mathbf{q}}$ , the gradient (3.11) vanishes by (3.6). The physical flux contribution vanishes cellwise by (3.3), while the grid transport contribution is  $-\bar{\mathbf{q}} d|\omega_c|/dt$  according to (3.15). It exactly balances the time derivative of  $|\omega_c| \bar{\mathbf{q}}$ .  $\square$

The divergence and curl constraints are measured on the dual cells by the following discrete operators

$$(3.16) \quad \mathfrak{D}_p(\mathbf{B}_h) = \sum_{c \in \mathcal{C}(p)} \mathbf{B}_c \cdot \boldsymbol{\ell}_{pc}, \quad \mathfrak{C}_p(\mathbf{j}_h) = \sum_{c \in \mathcal{C}(p)} \mathbf{j}_c \cdot \mathbf{t}_{pc}.$$

Their normalized counterparts are obtained by division by  $|\omega_p|$ . Up to the orientation convention induced by (3.11),  $\mathfrak{D}_p/|\omega_p|$  and  $\mathfrak{C}_p/|\omega_p|$  are finite volume approximations of  $\nabla \cdot \mathbf{B}$  and  $\nabla \times \mathbf{j}$ , respectively. The purpose of the constrained mesh motion is to determine the interior grid velocities so that the appropriate quantity in (3.16) vanishes on the current geometry configuration.

**3.1. Mesh motion and admissibility.** The nodal coordinates satisfy the kinematic equation

$$(3.17) \quad \frac{d\mathbf{x}_p}{dt} = \mathbf{w}_p, \quad p \in \mathcal{N}_h.$$

For the sake of clarity, we assume a fixed-boundary setting with  $\mathbf{w}_p = \mathbf{0}$  for  $p \in \mathcal{N}_h^\partial$ . Only the interior grid velocities are thus unknown, and the connectivity is kept fixed throughout the motion.

**DEFINITION 3.4** (Admissible moving mesh). *For a vertex  $(p, c)$  of cell  $\omega_c$ , let  $p^+$  and  $p^-$  be the next and previous vertices in the counter-clockwise ordering, and define*

$$(3.18) \quad \mathbb{E}_{pc}(\mathbf{X}) := [\mathbf{x}_{p^+} - \mathbf{x}_p, \mathbf{x}_{p^-} - \mathbf{x}_p].$$

*A configuration  $\mathbf{X}$  is admissible if every cell is simple, has nonzero edges, and is strictly convex with the prescribed counter-clockwise orientation, namely*

$$(3.19) \quad |\omega_c(\mathbf{X})| > 0, \quad \det \mathbb{E}_{pc}(\mathbf{X}) > 0, \quad \|\mathbf{x}_q - \mathbf{x}_p\|_2 > 0, \quad q \in \{p^-, p^+\},$$

*and non-adjacent edges of the same polygon do not intersect.*

The characteristic mesh size  $h(t)$  is given by the area-weighted quantity

$$(3.20) \quad h(t) = \left( \frac{\sum_{c=1}^{N_c} |\omega_c(t)|^2}{\sum_{c=1}^{N_c} |\omega_c(t)|} \right)^{1/2}.$$

**4. Involution-preserving ALE finite volume method.** We now present the fully discrete involution-preserving scheme. We first introduce the Runge–Kutta ALE update, then derive the corresponding nonlinear constraint, its linearization, and the Lagrange-multiplier formulation. Finally, we prove that the discrete involution is preserved.

**4.1. Fully discrete Runge–Kutta ALE update.** Let  $t^n$  and  $\Delta t = t^{n+1} - t^n$  be the current time level and the current time step, respectively. The time step is subject to a classical CFL-type stability condition with  $\text{CFL} = 1/2$ . The time discretization relies on the SSP–RK2 method with  $N_{\text{RK}} = 2$  stages and Butcher matrix  $(\mathbf{a}_{sj})$  and weights  $(\mathbf{b}_s)$  for  $s, j = 1, \dots, N_{\text{RK}}$  (see [14]).

Let  $\mathbf{H}_c^{(s)} = \mathbf{H}_c(\mathbf{q}^{(s)}, \mathbf{w}^{(s)})$  denote the integrated finite volume residual (3.14) evaluated on the geometry  $\mathbf{X}^{(s)}$  of stage  $s$ . Geometry and numerical solution are combined with exactly the same Runge–Kutta coefficients at each stage, namely

$$(4.1) \quad \mathbf{x}_p^{(s)} = \mathbf{x}_p^n + \Delta t \sum_{j=1}^{s-1} \mathbf{a}_{sj} \mathbf{w}_p^{(j)}, \quad |\omega_c^{(s)}| \mathbf{q}_c^{(s)} = |\omega_c^n| \mathbf{q}_c^n + \Delta t \sum_{j=1}^{s-1} \mathbf{a}_{sj} \mathbf{H}_c^{(j)},$$

and the final update is

$$(4.2) \quad \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \sum_{s=1}^{N_{\text{RK}}} \mathbf{b}_s \mathbf{w}_p^{(s)}, \quad |\omega_c^{n+1}| \mathbf{q}_c^{n+1} = |\omega_c^n| \mathbf{q}_c^n + \Delta t \sum_{s=1}^{N_{\text{RK}}} \mathbf{b}_s \mathbf{H}_c^{(s)}.$$

REMARK 4.1 (Geometric Conservation Law). *The SSP-RK2 method is not enough to guarantee a fully discrete GCL because the polygonal area is quadratic in the vertex coordinates. Let  $\mathbf{X}_0$  be the initial geometry and let  $\alpha_j$  be the Runge–Kutta coefficient multiplying the  $j^{\text{th}}$  grid velocity in the targeted combination:  $\alpha_j = \mathbf{a}_{sj}$  for a stage path and  $\alpha_j = \mathbf{b}_j$  for the final path. Define successively*

$$(4.3) \quad \Delta \mathbf{X}_j := \Delta t \alpha_j \mathbf{w}^{(j)}, \quad \mathbf{X}_j := \mathbf{X}_{j-1} + \Delta \mathbf{X}_j.$$

*Each nonzero displacement  $\Delta \mathbf{X}_j$  is interpreted as a straight segment, and its geometric flux must be evaluated at the segment midpoint*

$$(4.4) \quad \mathbf{X}_{j-1/2} = \mathbf{X}_{j-1} + \frac{1}{2} \Delta \mathbf{X}_j, \quad G_{c,j}^{\text{mid}} = \sum_{p \in \mathcal{P}(c)} \mathbf{w}_p^{(j)} \cdot \boldsymbol{\ell}_{pc}(\mathbf{X}_{j-1/2}).$$

*Since the corner vector is affine along a straight mesh trajectory, the midpoint quadrature is exact and gives*

$$(4.5) \quad |\omega_c(\mathbf{X}_j)| - |\omega_c(\mathbf{X}_{j-1})| = \Delta t \alpha_j G_{c,j}^{\text{mid}}.$$

*Summing (4.5) along the actual Runge–Kutta path yields the fully discrete GCL. Cell volumes are then always recomputed from the resulting coordinates. Note that  $G_{c,j}^{\text{mid}}$  is a swept-geometry flux and does not necessarily coincide with the physical residual evaluated on the Runge–Kutta stage geometry.*

**4.2. Nonlinear constraint: involutions through mesh displacement.** To provide a general description of the new methodology, let us define

$$(4.6) \quad \mathbf{z} = \begin{cases} \mathbf{B} & \text{for MHD} \\ \mathbf{j} & \text{for EH} \end{cases}, \quad \mathbf{g}_{pc} = \begin{cases} \boldsymbol{\ell}_{pc} & \text{for MHD} \\ \mathbf{t}_{pc} & \text{for EH} \end{cases}.$$

Here  $\mathbf{z} \in \mathbb{R}^2$  is the constrained physical field, whereas  $\mathbf{g}_{pc} \in \mathbb{R}^2$  is the geometrical vector defining its discrete involution. In addition,  $\mathbb{F}_z$  denotes the contributions of the physical flux tensor  $\mathbb{F}$  associated with  $\mathbf{z}$ , and  $\mathbb{G}_{z,p}$  denotes the corresponding nodal gradient  $\mathbb{G}_p$ . Therefore, both discrete involutions  $\mathfrak{D}$  and  $\mathfrak{C}$  in (3.16) can be written as

$$(4.7) \quad \mathfrak{L}_p(\mathbf{z}_h, \mathbf{X}) = \sum_{c \in \mathcal{C}(p)} \mathbf{z}_c \cdot \mathbf{g}_{pc}(\mathbf{X}), \quad p \in \mathcal{N}_h^\circ.$$

The grid velocity of stage  $s$  is chosen to constrain the actual Runge–Kutta combination to which this velocity contributes. To express this point, define

$$(4.8) \quad \beta_j^{(s)} = \begin{cases} \mathbf{a}_{s+1,j} & s < N_{\text{RK}} \\ \mathbf{b}_j & s = N_{\text{RK}} \end{cases}, \quad \sigma_s := \beta_s^{(s)}.$$

The sequential construction is restricted to Runge–Kutta schemes satisfying  $\sigma_s > 0$  for every  $s$ . For an intermediate stage, the target is therefore the geometry and state of the next Runge–Kutta stage; for the last stage, it is the final Runge–Kutta combination.

DEFINITION 4.2 (Runge–Kutta target of stage  $s$ ). *The contributions already known before solving for  $\mathbf{w}^{(s)}$  define the target base geometry and solution, that is*

$$(4.9) \quad \mathbf{x}_{p,\text{base}}^{(s)} = \mathbf{x}_p^n + \Delta t \sum_{j=1}^{s-1} \beta_j^{(s)} \mathbf{w}_p^{(j)}, \quad |\omega_c^{(s)}| \mathbf{z}_{c,\text{base}}^{(s)} = |\omega_c^n| \mathbf{z}_c^n + \Delta t \sum_{j=1}^{s-1} \beta_j^{(s)} \mathbf{H}_{z,c}^{(j)}.$$

The displacement unknown and the corresponding stage grid velocity are

$$(4.10) \quad \mathbf{d}_p^{(s)} = \sigma_s \Delta t \boldsymbol{\omega}_p^{(s)}, \quad \mathbf{x}_p^{(s)}(\mathbf{d}) = \mathbf{x}_{p,\text{base}}^{(s)} + \mathbf{d}_p, \quad p \in n_h,$$

with  $\mathbf{d}_p = \mathbf{0}$  for  $p \in n_h^\partial$ . The corresponding global vectors are denoted by  $\mathbf{d} := (\mathbf{d}_p)_{p \in n_h}$  and  $\mathbf{X}^{(s)}(\mathbf{d}) := (\mathbf{x}_p^{(s)}(\mathbf{d}))_{p \in n_h}$ .

The flux of stage  $s$  is always evaluated on the stage geometry  $\mathbf{X}^{(s)}$  in (4.1), thus in (4.9) we have

$$(4.11) \quad \mathbf{H}_{z,c}^{(s)}(\mathbf{d}) = - \sum_{p \in \mathcal{P}(c)} \widehat{\mathbb{F}}_{z,p}^{(s)} \left( \frac{\mathbf{d}}{\sigma_s \Delta t} \right) \boldsymbol{\ell}_{pc}^{(s)}.$$

The stage state, its discrete gradient, the stage corner vectors, and the numerical viscosity coefficient are fixed during the nonlinear solver. However, the ALE transport term is not frozen:

$$(4.12) \quad \widehat{\mathbb{F}}_{z,p}^{(s)}(\mathbf{d}) = \mathbb{F}_z(\mathbf{q}_p^{(s)}) - \mathbf{z}_p^{(s)} \otimes \frac{\mathbf{d}_p}{\sigma_s \Delta t} - \frac{1}{2} h_p^{(s)} a_p^{(s)} \mathbf{G}_{z,p}^{(s)},$$

where the nodal value  $\mathbf{z}_p^{(s)}$  is obtained from the stage cell averages by the volume-weighted transfer (3.8). It follows that the target numerical solution is

$$(4.13) \quad \mathbf{Z}_c^{(s)}(\mathbf{d}) = |\omega_{c,\text{base}}^{(s)}| \mathbf{z}_{c,\text{base}}^{(s)} + \sigma_s \Delta t \mathbf{H}_{z,c}^{(s)}(\mathbf{d}),$$

with the definition

$$(4.14) \quad \mathbf{Z}_c^{(j)}(\mathbf{d}) := |\omega_c^{(j)}(\mathbf{d})| \mathbf{z}_c^{(j)}(\mathbf{d}), \quad j \in [1, N_{\text{RK}}].$$

Notice that (4.13) is recomputed at every nonlinear iteration. Freezing it would neglect the dependence of the ALE flux on the unknown grid velocity. Notice also that we directly solve for the quantity  $\mathbf{Z}$  and not for  $\mathbf{z}$ , which can be ultimately retrieved from the definition (4.14).

**DEFINITION 4.3** (Fully coupled involution constraint). *For every  $p \in n_h^\circ$ , the displacement of stage  $s$  is determined by the nonlinear equations*

$$(4.15) \quad \mathcal{R}_p^{(s)}(\mathbf{d}) := \sum_{c \in \mathcal{C}(p)} \mathbf{z}_c^{(s)}(\mathbf{d}) \cdot \mathbf{g}_{pc} \left( \mathbf{X}^{(s)}(\mathbf{d}) \right) = 0.$$

*Both the solution in (4.13) and all geometrical quantities entering (4.15) are evaluated on the current trial displacement, and solved with a Newton method.*

**4.3. Analytical linearization.** Let  $N_I$  denote the number of internal nodes, which are the free vertices after the boundary conditions have been imposed. Their  $2N_I$  scalar displacement degrees of freedom are collected in  $\mathbf{d} = (\mathbf{d}_x, \mathbf{d}_y) \in \mathbb{R}^{2N_I}$ . Thus  $\mathbf{d}_x, \mathbf{d}_y \in \mathbb{R}^{N_I}$  are vectors, while  $d_{k,r}$  is the scalar component in direction  $k \in \{x, y\}$  associated with  $r \in n_h^\circ$ . At each nonlinear iteration  $m$  of the Newton solver, the residual vector and its Jacobian are denoted by

$$(4.16) \quad \mathcal{R}^{(m)} = \mathcal{R}(\mathbf{d}^{(m)}), \quad \mathbf{J}^{(m)} = \frac{\partial \mathcal{R}}{\partial \mathbf{d}} \left( \mathbf{d}^{(m)} \right) = \begin{pmatrix} \mathbf{J}_x^{(m)} & \mathbf{J}_y^{(m)} \end{pmatrix}.$$

There are therefore  $2N_I$  scalar unknowns and, before removal of dependencies,  $N_I$  scalar nodal constraints. If the constraint Jacobian has row rank  $r \leq N_I$ , the linearized

feasible space has dimension  $2N_I - r$ ; in the full-row-rank case its expected dimension is  $N_I$ . Periodic identification, prescribed boundary motion, and global patch identities may reduce  $r$ , whereas incompatible boundary data may make the affine linearized equations inconsistent. In practice, redundant constraint rows must be removed or handled by a rank-revealing factorization. The smallest retained singular value of the scaled Jacobian should be monitored as the mesh deforms, since its decay signals loss of regularity before cell inversion, i.e. negative volume, necessarily occurs. The Jacobian contains three coupled effects: the variation of the corner vector, the variation of the trial cell volume, and the variation of the ALE flux. Indeed, omitting the stage index for readability, for  $k \in \{x, y\}$  one has

$$(4.17a) \quad \frac{\partial \mathbf{z}_c}{\partial d_{k,r}} = \frac{1}{|\omega_c|} \left( \frac{\partial \mathbf{Z}_c}{\partial d_{k,r}} - \mathbf{z}_c \frac{\partial |\omega_c|}{\partial d_{k,r}} \right),$$

$$(4.17b) \quad \frac{\partial \mathbf{Z}_c}{\partial d_{k,r}} = \sigma_s \Delta t \frac{\partial \mathbf{H}_{z,c}}{\partial d_{k,r}},$$

$$(4.17c) \quad \frac{\partial \mathcal{R}_p}{\partial d_{k,r}} = \sum_{c \in \mathcal{C}(p)} \left[ \frac{\partial \mathbf{z}_c}{\partial d_{k,r}} \cdot \mathbf{g}_{pc} + \mathbf{z}_c \cdot \frac{\partial \mathbf{g}_{pc}}{\partial d_{k,r}} \right].$$

For a polygonal cell with counter-clockwise local node numbering, the corner vector has the geometrical representation

$$(4.18) \quad \boldsymbol{\ell}_{pc} = \frac{1}{2} \begin{pmatrix} y_{p^+} - y_{p^-} \\ x_{p^-} - x_{p^+} \end{pmatrix}, \quad \frac{\partial |\omega_c|}{\partial \mathbf{x}_p} = \boldsymbol{\ell}_{pc}.$$

Therefore its derivatives are local and nonzero only with respect to the two neighbouring vertices. The derivatives of  $\mathbf{t}_{pc}$  follow by applying the same fixed counter-clockwise rotation. Since  $\mathbf{q}^{(s)}$ ,  $\mathbf{G}^{(s)}$ , and  $\mathbf{a}_p^{(s)}$  are frozen in (4.12), only the grid-transport part depends explicitly on  $\mathbf{d}$  in the Newton method. Thus

$$(4.19) \quad \frac{\partial \boldsymbol{\omega}_q^{(s)}}{\partial d_{x,r}} = \frac{\delta_{qr}}{\sigma_s \Delta t} \mathbf{e}_x, \quad \frac{\partial \boldsymbol{\omega}_q^{(s)}}{\partial d_{y,r}} = \frac{\delta_{qr}}{\sigma_s \Delta t} \mathbf{e}_y.$$

Here  $\delta_{qr}$  is the Kronecker symbol, and  $\mathbf{e}_x, \mathbf{e}_y$  are the Cartesian basis vectors. The factor  $(\sigma_s \Delta t)^{-1}$  in (4.19) cancels the factor  $\sigma_s \Delta t$  in (4.17b). The resulting contribution can be written explicitly as

$$(4.20) \quad \left. \frac{\partial \mathbf{H}_{z,c}^{(s)}}{\partial d_{k,r}} \right|_{\text{ALE}} = \frac{\mathbf{z}_r^{(s)} (\boldsymbol{\ell}_{rc}^{(s)})_k}{\sigma_s \Delta t}, \quad k \in \{x, y\}, \quad r \in \mathcal{P}(c).$$

It is therefore local to the cells adjacent to the displaced node, and the resulting constraint Jacobian has thus a compact node-based stencil.

**4.4. Mesh optimization.** The nonlinear involution constraint determines which mesh displacements are feasible, but it does not determine a unique displacement. We remove this indeterminacy with a secondary elliptic criterion. Its first part distributes the motion through the whole mesh and attracts the nodes toward the initial configuration; its second part penalizes local changes of shape and volume. Importantly, this criterion only selects the mesh: the involution itself is imposed by the fully coupled ALE constraint (4.15), and no correction is applied to the physical field after the update. The construction combines linear-elastic mesh motion with the target-matrix optimization paradigm (TMOP) [20, 15, 11].

DEFINITION 4.4 (Corner distortion measure). *For every corner  $(p, c)$  of cell  $\omega_c$ , let  $p^+$  and  $p^-$  be the next and previous vertices in the counter-clockwise ordering, and write  $\mathbf{x}_p(\mathbf{d}) = \mathbf{x}_{p, \text{base}}^{(s)} + \mathbf{d}_p$ . The reference and trial corner matrices are*

$$(4.21) \quad \mathbf{W}_{pc} := [\mathbf{x}_{p^+}^{\text{ref}} - \mathbf{x}_p^{\text{ref}}, \mathbf{x}_{p^-}^{\text{ref}} - \mathbf{x}_p^{\text{ref}}],$$

$$(4.22) \quad \mathbb{A}_{pc}(\mathbf{d}) := [\mathbf{x}_{p^+}(\mathbf{d}) - \mathbf{x}_p(\mathbf{d}), \mathbf{x}_{p^-}(\mathbf{d}) - \mathbf{x}_p(\mathbf{d})].$$

The reference mesh is assumed admissible, so that  $\det \mathbf{W}_{pc} > 0$  at every corner. Their relative deformation and its Jacobian are

$$(4.23) \quad \mathbb{T}_{pc}(\mathbf{d}) := \mathbb{A}_{pc}(\mathbf{d}) \mathbf{W}_{pc}^{-1}, \quad J_{pc}(\mathbf{d}) := \det \mathbb{T}_{pc}(\mathbf{d}).$$

For  $J = \det \mathbb{T} > 0$ , define

$$(4.24) \quad \varrho_q(\mathbb{T}) := \frac{(T_{11} - T_{22})^2 + (T_{12} + T_{21})^2}{2J} + \frac{3}{4} \left( \sqrt{J} - \frac{1}{\sqrt{J}} \right)^2, \quad \Psi(\mathbb{T}) := \varrho_q(\mathbb{T}) + \varrho_q(\mathbb{T})^2,$$

and

$$(4.25) \quad \mathcal{E}_q(\mathbf{d}) := \sum_{c=1}^{N_c} \frac{1}{|\mathcal{P}(c)|} \sum_{p \in \mathcal{P}(c)} \det(\mathbf{W}_{pc}) \Psi(\mathbb{T}_{pc}(\mathbf{d})).$$

The first contribution in (4.24) measures shear and anisotropy, while the second measures the local change of area. Hence  $\Psi$  in (4.25) vanishes at the reference configuration and diverges when a corner approaches degeneracy.

Let  $\mathbf{X}^{\text{ref}}$  be the initial mesh and let  $a_{\max}^{(s)}$  be the largest stage wave-speed bound. We introduce the dimensionless stage Courant number and the distortion of the base mesh, that is

$$(4.26) \quad \chi_s := \frac{\sigma_s \Delta t a_{\max}^{(s)}}{h_{\min}^{\text{base}}}, \quad h_{\min}^{\text{base}} := \min_c \sqrt{|\omega_c^{\text{base}}|},$$

$$(4.27) \quad \delta_s := \max_{(p,c)} \sqrt{2\Psi(\mathbb{T}_{pc}(\mathbf{0}))},$$

which measures the largest corner distortion of the base mesh. We further define the bounded function

$$(4.28) \quad \zeta(\chi) := \begin{cases} \frac{\chi}{1-\chi}, & 0 \leq \chi < \frac{1}{2}, \\ 1, & \chi \geq \frac{1}{2} \end{cases},$$

and set

$$(4.29) \quad \mathbf{d}_p^{\text{ref},(s)} := \vartheta_s \left( \mathbf{x}_p^{\text{ref}} - \mathbf{x}_{p, \text{base}}^{(s)} \right), \quad \mathbf{e}_p := \mathbf{d}_p - \mathbf{d}_p^{\text{ref},(s)},$$

with the definition

$$(4.30) \quad \vartheta_s := \frac{\chi_s}{1 + \chi_s} + \frac{\zeta(\chi_s)}{1 + \chi_s} \frac{\delta_s}{\chi_s + \delta_s},$$

and we take  $\vartheta_s = 0$  when  $\chi_s = \delta_s = 0$ . The first term in  $\vartheta_s$  supplies a bounded return toward the reference mesh on the stage time scale; the second term activates a stronger recovery only when distortion has accumulated. All quantities are computed from the current stage. For a fixed mesh, bounded wave speed, and bounded distance between the base and reference coordinates,  $\mathbf{d}^{\text{ref},(s)} = O(\Delta t)$  as  $\Delta t \rightarrow 0$ .

For a nodal vector field  $\mathbf{v}_h = (\mathbf{v}_p)_{p \in n_h}$  satisfying  $\mathbf{v}_p = \mathbf{0}$  for  $p \in n_h^\partial$ , define the cellwise gradient and symmetric strain

$$(4.31) \quad \nabla_c^h \mathbf{v}_h := \frac{1}{|\omega_c^{\text{base}}|} \sum_{r \in \mathcal{P}(c)} \mathbf{v}_r \otimes \boldsymbol{\ell}_{rc}^{\text{base}}, \quad \boldsymbol{\varepsilon}_c(\mathbf{v}_h) := \frac{1}{2} (\nabla_c^h \mathbf{v}_h + (\nabla_c^h \mathbf{v}_h)^\top).$$

With  $L_\Omega$  the largest Cartesian extent of the domain, the elliptic bilinear form is

$$(4.32) \quad a_{\text{el}}(\mathbf{u}_h, \mathbf{v}_h) := \sum_{c=1}^{N_c} \left[ \frac{|\omega_c^{\text{base}}|}{|\mathcal{P}(c)|} \sum_{p \in \mathcal{P}(c)} \mathbf{u}_p \cdot \mathbf{v}_p + 2L_\Omega^2 |\omega_c^{\text{base}}| \boldsymbol{\varepsilon}_c(\mathbf{u}_h) : \boldsymbol{\varepsilon}_c(\mathbf{v}_h) \right].$$

It discretizes the screened elasticity operator

$$(4.33) \quad \mathbf{e} - L_\Omega^2 \nabla \cdot (2\boldsymbol{\varepsilon}(\mathbf{e})), \quad \mathbf{e} = \mathbf{d} - \mathbf{d}^{\text{ref}}.$$

The zeroth-order term removes rigid modes and anchors the displacement, whereas the strain term couples its two components and distributes a local correction over neighbouring cells. The factor  $L_\Omega^2$  normalizes the elastic coupling with the domain scale; no mesh-dependent tuning parameter is here introduced.

**DEFINITION 4.5** (Constrained mesh problem). *At stage  $s$ , the mesh-selection problem is*

$$(4.34a) \quad \min_{\mathbf{d}} \quad \mathcal{E}_h^{(s)}(\mathbf{d}) := \frac{1}{2} a_{\text{el}}(\mathbf{e}_h, \mathbf{e}_h) + \chi_s \mathcal{E}_q(\mathbf{d}),$$

$$(4.34b) \quad \text{subject to } \mathcal{R}^{(s)}(\mathbf{d}) = \mathbf{0}, \quad \text{with } \mathbf{X}^{(s)}(\mathbf{d}) \text{ admissible.}$$

A feasible Sequential Quadratic Programming (SQP) correction is obtained as follows. At iteration  $m$ , let  $\mathbb{H}^{(m)}$  be the Gauss–Newton model of (4.34a), made positive definite by the elliptic part. We denote the objective gradient and the Jacobian of the involution constraint by

$$(4.35) \quad \boldsymbol{\gamma}^{(m)} := (\boldsymbol{\gamma}_x^{(m)}, \boldsymbol{\gamma}_y^{(m)}) = \nabla_{\mathbf{d}} \mathcal{E}_h^{(s)}(\mathbf{d}^{(m)}), \quad \mathbb{J}^{(m)} := (\mathbb{J}_x^{(m)}, \mathbb{J}_y^{(m)}).$$

The displacement correction and Lagrange multiplier solve

$$(4.36) \quad \begin{pmatrix} \mathbb{H}_{xx} & \mathbb{H}_{xy} & -\mathbb{J}_x^\top \\ \mathbb{H}_{yx} & \mathbb{H}_{yy} & -\mathbb{J}_y^\top \\ \mathbb{J}_x & \mathbb{J}_y & \mathbf{0} \end{pmatrix} \begin{pmatrix} \delta \mathbf{d}_x \\ \delta \mathbf{d}_y \\ \boldsymbol{\lambda} \end{pmatrix} = - \begin{pmatrix} \boldsymbol{\gamma}_x \\ \boldsymbol{\gamma}_y \\ \mathcal{R} \end{pmatrix}.$$

All blocks and right-hand sides in (4.36) are evaluated at iteration  $m$ ; the superscript is omitted there for readability. The vector  $\boldsymbol{\lambda} \in \mathbb{R}^{N_I}$  contains the Lagrange multipliers associated with the nodal involution constraints. Note that the elliptic metric and the constraint Jacobian have compact node-based stencils; consequently, (4.36) is a sparse saddle-point system. During feasibility restoration,  $\boldsymbol{\gamma}_x = \boldsymbol{\gamma}_y = \mathbf{0}$ ; after feasibility has been reached, the last right-hand side is set to zero and the same system produces a tangent mesh-quality step.

**PROPOSITION 4.6** (Elliptic selection of a feasible correction). *Assume that  $\mathbb{H}^{(m)}$  is positive definite on the free displacement space and that  $\mathbb{J}^{(m)}$  has full row rank. Then (4.36) has a unique displacement component. During restoration, this component is the minimum- $\mathbb{H}^{(m)}$ -norm correction satisfying the linearized involution constraint.*

*Proof.* The first two block rows are the stationarity equations of a strictly convex quadratic functional under the affine constraint given by the last block row. The KKT conditions are therefore necessary and sufficient, and strict convexity gives a unique displacement correction.  $\square$

The SQP iteration is globalized by a backtracking line search restricted to admissible meshes. Restoration corrections first recover the nonlinear involution constraint; tangent corrections then decrease (4.34a) to first order along the constraint manifold, with feasibility restored after each accepted correction. Therefore the procedure seeks an admissible local solution of (4.34), while the accepted state satisfies the complete non-frozen constraint to the prescribed nonlinear tolerance.

**DEFINITION 4.7** (Predictor–corrector algorithm). *An inexpensive initial displacement is first obtained by solving a frozen-flux problem, in which the numerical flux is evaluated only once while the trial geometry is allowed to vary. The subsequent fully coupled corrector updates the geometry, ALE flux, conservative state, and involution residual at each nonlinear iteration. Only solutions satisfying the complete, non-frozen nonlinear problem are considered admissible.*

---

**Algorithm 4.1** Involution-preserving mesh displacement at Runge–Kutta stage  $s$ .

---

- 1: Construct the Runge–Kutta target base geometry and conservative state from (4.9).
  - 2: Compute an initial displacement by solving a frozen-flux version of (4.11)–(4.15).
  - 3: Solve the constrained mesh problem (4.34) by successive KKT corrections (4.36).  
At every iterate, recompute the target geometry, ALE flux (4.12), conservative state (4.13), and constraint Jacobian (4.16).
  - 4: Globalize each correction by a line search restricted to meshes satisfying (3.19).
  - 5: Accept the displacement when the complete non-frozen residual (4.15) reaches the prescribed tolerance.
  - 6: Recover the grid velocity from (4.10) and use the same constrained flux balance in the Runge–Kutta update (4.1)–(4.2).
- 

#### 4.5. Discrete involution preservation.

**THEOREM 4.8** (Preservation of the targeted Runge–Kutta state). *Assume that the state at time  $t^n$  satisfies the appropriate discrete involution. At every Runge–Kutta stage, suppose that (4.15) is solved exactly and that the flux balance used in (4.13) is exactly the flux balance entering the finite volume residual. Then every targeted intermediate state and the final state (4.2) satisfy the discrete involution exactly. With an iterative nonlinear solver, the constraint is preserved up to its stopping tolerance and floating-point roundoff.*

*Proof.* At stage  $s$ , combine (4.9) with the current contribution  $\sigma_s \Delta t \mathbf{H}_{z,c}^{(s)}$ . Because the same flux balance is used by the constrained solver and the finite volume update, the resulting numerator is precisely (4.13). Likewise, (4.9) and (4.10) produce exactly the geometry of the targeted Runge–Kutta combination. Therefore the cell average entering that combination is (4.13), not an auxiliary state. Equation (4.15)

is consequently identical to the discrete involution (4.7) evaluated on the targeted state. For  $s < N_{\text{RK}}$  this establishes the property for the next Runge–Kutta stage. At  $s = N_{\text{RK}}$ ,  $\beta_j^{(s)} = \mathbf{b}_j$ , so the target is exactly (4.2). The conclusion follows stage by stage.  $\square$

PROPOSITION 4.9 (Inexact nonlinear solves). *Suppose the accepted displacement at target  $s$  satisfies*

$$(4.37) \quad \|\mathcal{R}^{(s)}\|_{\star} \leq \tau_s, \quad \tau_s > 0,$$

*in a fixed norm  $\|\cdot\|_{\star}$ , and that the state actually updated by the finite volume code is the same state used to evaluate this residual. Then the targeted discrete involution satisfies the same bound. In particular,*

$$(4.38) \quad \|\mathcal{L}^{n+1}\|_{\star} \leq \tau_{N_{\text{RK}}} + \varepsilon_{\text{alg}},$$

*where  $\varepsilon_{\text{alg}}$  collects floating-point contributions. Hence one may take  $C_{\text{RK}} = 1$  for the RK-target formulation; errors from earlier targets do not accumulate in (4.38) because the last nonlinear residual is evaluated directly on the final RK combination.*

For a normalized diagnostic, the same conclusion holds after division by the smallest interior nodal-patch area. Thus the raw tolerance should scale like the desired normalized accuracy times  $\min_{p \in n_h^{\circ}} |\omega_p|$  (and, in convergence studies, should be chosen below the targeted truncation error). This provides a mesh-dependent stopping rule instead of a fixed raw residual threshold.

REMARK 4.10 (Scope and modularity of this construction). *Theorem 4.8 applies equally to the divergence constraint of ideal MHD and to the curl constraint of EH. The two cases differ only through the constrained variable, the geometrical vector in (4.6), and the corresponding physical flux. The Runge–Kutta target, nonlinear coupling, admissibility checks, and linear algebra are common to both.*

*The proof does not rely on the particular Rusanov-type flux in (3.10). The construction can be coupled to another conservative ALE spatial operator provided that its discrete involution is available on the updated state and geometry, the identical constrained residual is used in both (4.15) and the conservative update, and every mesh-dependent contribution is included in the nonlinear residual and its linearization. Under these conditions, the constrained mesh solve is a modular structure-preserving layer rather than a model-specific field correction.*

**5. Numerical results.** In order to validate the scheme presented in this paper, we solve in the following a set of classical benchmark test cases. A Cartesian computational mesh is initially assigned, which then evolves in order to respect the prescribed involutions.

**5.1. MHD numerical convergence studies.** The numerical convergence studies for the new involution-preserving scheme are carried out using a smooth manufactured vortex solution defined over the computational domain  $\Omega = [-5, 5]^2$ , starting from [1]. The exact solution is given by

$$(5.1) \quad \rho = 1 + 0.1 \psi, \quad \mathbf{u} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \frac{\varepsilon}{2\pi} \psi \begin{pmatrix} -y \\ x \end{pmatrix}, \quad \mathbf{B} = \frac{\mu}{2\pi} \psi \begin{pmatrix} -y \\ x \end{pmatrix},$$

with

$$(5.2) \quad \psi = \exp\left(\frac{1-r^2}{2}\right), \quad r = \sqrt{x^2 + y^2},$$

while the pressure is defined as

$$(5.3) \quad p = \varepsilon + \frac{\mu}{\sqrt{4\pi}} + \frac{1}{8\pi} \left( \frac{\mu}{2\pi} \right)^2 (1 - r^2) e^{1-r^2} - \frac{1}{2} \left( \frac{\varepsilon}{2\pi} \right)^2 e^{1-r^2} + 0.1 \psi.$$

The parameters are set to  $\varepsilon = 1$  and  $\mu = 4\pi$ .

The simulations are run on a sequence of successively refined meshes until the final time  $t_f = 0.1$ . In Table 1, we report the errors in  $L^1$  and  $L^2$  norm. We observe that the expected order of one is achieved.

TABLE 1  
MHD convergence rates on moving meshes at  $t_f = 0.1$ .  $L^1$  and  $L^2$  errors and experimental orders of convergence for the primitive variables.

| $N_x = N_y$ | $\rho$       |       | $u$      |       | $v$      |       | $p$      |       | $B_x$    |       | $B_y$    |       |
|-------------|--------------|-------|----------|-------|----------|-------|----------|-------|----------|-------|----------|-------|
|             | $L^1$ errors |       |          |       |          |       |          |       |          |       |          |       |
| 30          | 2.017e-3     | -     | 3.379e-3 | -     | 3.369e-3 | -     | 4.863e-3 | -     | 4.234e-2 | -     | 4.241e-2 | -     |
| 40          | 1.532e-3     | 0.964 | 2.557e-3 | 0.978 | 2.564e-3 | 0.958 | 3.684e-3 | 0.973 | 3.206e-2 | 0.975 | 3.222e-2 | 0.963 |
| 50          | 1.232e-3     | 0.989 | 2.059e-3 | 0.984 | 2.071e-3 | 0.969 | 2.997e-3 | 0.938 | 2.580e-2 | 0.988 | 2.599e-2 | 0.976 |
| 60          | 1.058e-3     | 0.840 | 1.757e-3 | 0.876 | 1.768e-3 | 0.873 | 2.607e-3 | 0.768 | 2.169e-2 | 0.957 | 2.186e-2 | 0.955 |
|             | $L^2$ errors |       |          |       |          |       |          |       |          |       |          |       |
| 30          | 5.006e-3     | -     | 8.167e-3 | -     | 8.123e-3 | -     | 1.805e-2 | -     | 1.013e-1 | -     | 1.017e-1 | -     |
| 40          | 3.818e-3     | 0.950 | 6.207e-3 | 0.962 | 6.184e-3 | 0.956 | 1.398e-2 | 0.895 | 7.718e-2 | 0.955 | 7.754e-2 | 0.951 |
| 50          | 3.070e-3     | 0.991 | 5.011e-3 | 0.973 | 5.008e-3 | 0.958 | 1.139e-2 | 0.932 | 6.236e-2 | 0.968 | 6.278e-2 | 0.959 |
| 60          | 2.589e-3     | 0.938 | 4.230e-3 | 0.934 | 4.226e-3 | 0.937 | 9.628e-3 | 0.926 | 5.258e-2 | 0.941 | 5.288e-2 | 0.947 |

**5.2. MHD Rotor Problem.** The so-called rotor problem proposed in [1] is characterized by a large density difference inside and outside the rotor. The computational domain is the square  $\Omega = [-0.5, 0.5]^2$ , and  $r := \sqrt{x^2 + y^2}$  denotes the distance from its centre. The background pressure and magnetic field are constant and set to  $p = 1$  and  $\mathbf{B} = (2.5, 0)^\top$ , respectively. The density and the velocity field are defined differently inside and outside the rotor:

$$(5.4) \quad \rho = \begin{cases} 10 & 0 \leq r \leq 0.1, \\ 1 & \text{otherwise,} \end{cases} \quad (u, v) = \begin{cases} (-10y, 10x) & 0 \leq r \leq 0.1, \\ (0, 0) & \text{otherwise.} \end{cases}$$

The ratio of specific heats is set to  $\gamma = 1.4$  and the simulation is run until the final time  $t_f = 0.25$ . The numerical solution is displayed in Figure 1, while Figure 2 compares the initial and final meshes in the central region. Over the complete computation, the maximum discrete divergence is  $9.82 \times 10^{-12}$  and its final value is  $2.06 \times 10^{-14}$ .

**5.3. MHD blast wave.** The MHD blast wave problem introduced in [1] represents a challenging test since it contains a strong magnetic field and a jump in the pressure of four orders of magnitude. The initial density is constant  $\rho = 1$ , the initial velocity field is set to zero  $(u, v) = (0, 0)$  and the magnetic field is  $\mathbf{B} = (70, 0)$ . The computational domain is  $\Omega = [-0.5, 0.5]^2$ . Inside the circular region  $r < 0.1$ , with  $r = \sqrt{x^2 + y^2}$ , the pressure is set to  $p = 1000$  whereas outside is assigned with  $p = 0.1$ . Thus initially, the Alfvén waves are faster than the sound speed inside the circular domain. Here we set  $\gamma = 1.4$  and the final time is chosen to be  $t_f = 0.01$ . The numerical solution at the final time is reported in Figure 3. The maximum discrete divergence over time is  $9.71 \times 10^{-12}$  and its final value is  $2.79 \times 10^{-12}$ .

**5.3.1. Euler-Heat numerical convergence studies.** To assess the accuracy, we consider a stationary manufactured solution for the Euler-Heat model. The computational domain is the square  $\Omega = [0, 10]^2$ . We use the centred coordinates

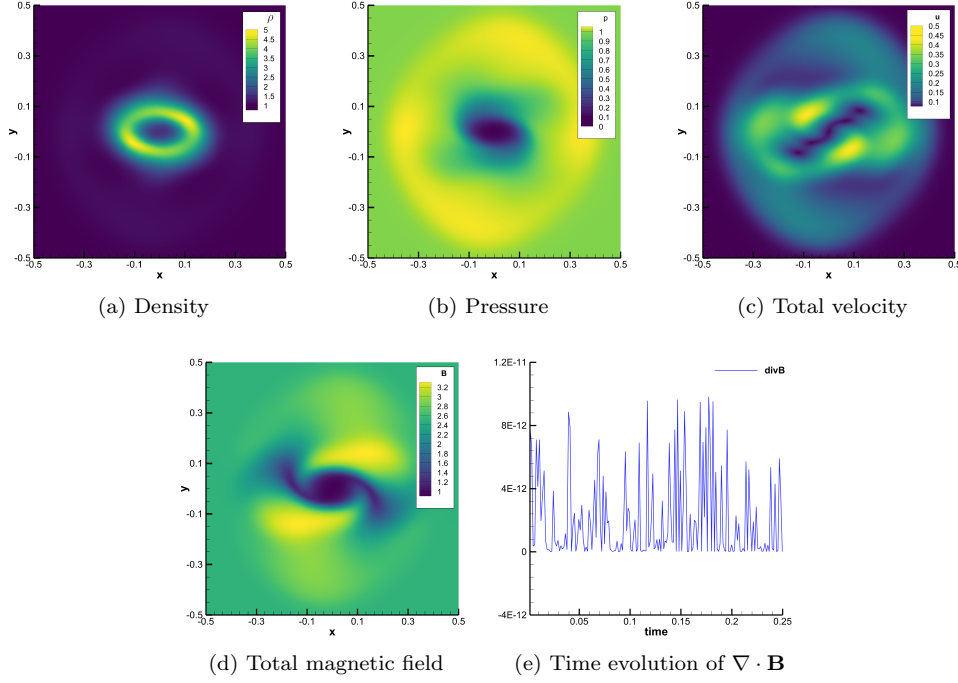


FIG. 1. Numerical solution of the MHD Rotor problem at time  $t_f = 0.25$  obtained using the new involution-preserving scheme on a  $N_x = N_y = 200$  mesh.

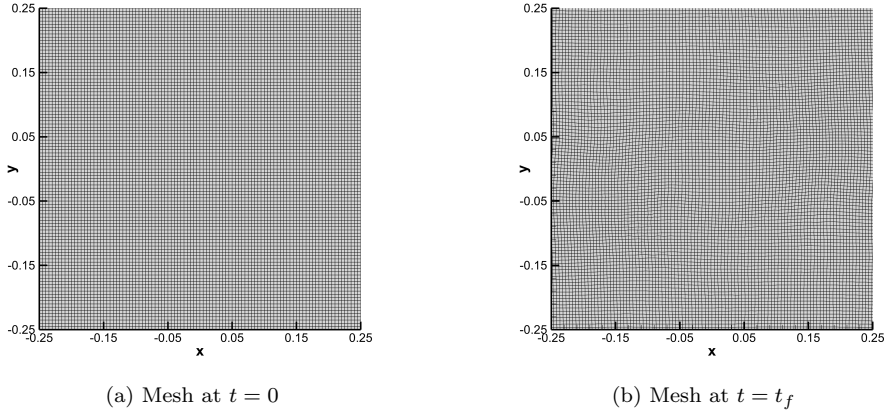


FIG. 2. Central zoom  $[-0.25, 0.25]^2$  of the rotor mesh at the initial and final time. The comparison highlights the smooth redistribution of the ALE displacement without obscuring the cell structure of the  $N_x = N_y = 200$  mesh.

$\tilde{x} := x - 5$ ,  $\tilde{y} := y - 5$ , and  $r := \sqrt{\tilde{x}^2 + \tilde{y}^2}$ . The fluid is assigned a unit density and pressure with the following velocity and scalar potential  $\varphi$ :

$$(5.5) \quad \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -\frac{\beta e^{\frac{1}{2}(1-r^2)}}{2\pi} \tilde{y} \\ \frac{\beta e^{\frac{1}{2}(1-r^2)}}{2\pi} \tilde{x} \end{pmatrix}, \quad \varphi = -\frac{\lambda e^{\frac{1}{2}(1-r^2)}}{2\pi}, \quad \beta, \lambda \in \mathbb{R}.$$

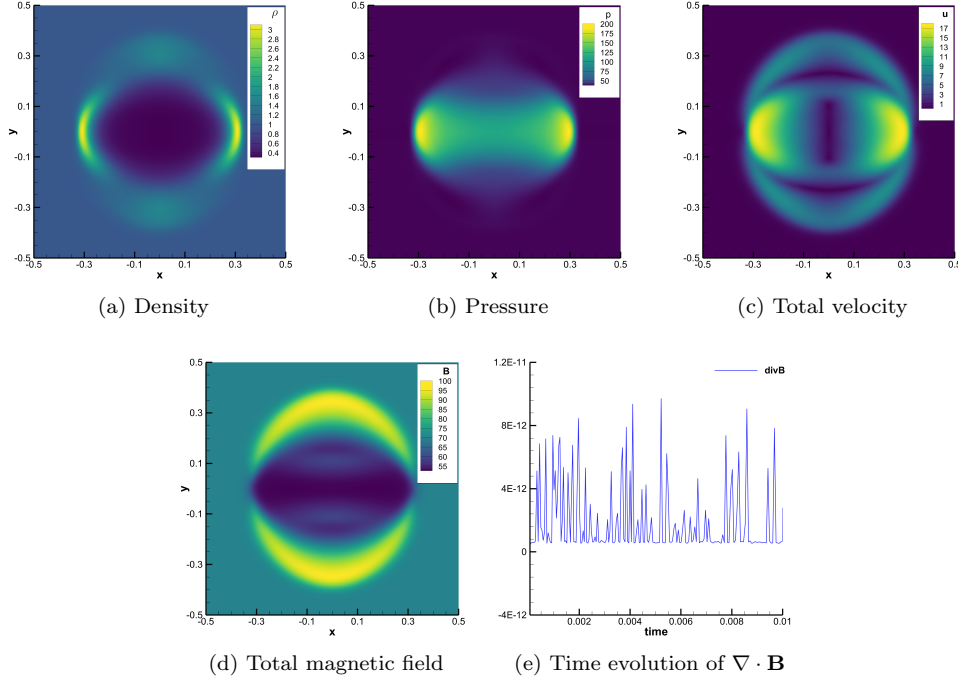


FIG. 3. Numerical solution of the MHD Blast wave at the final time  $t_f = 0.01$  obtained using our involution preserving scheme on a  $N_x = N_y = 200$  mesh.

The thermal impulse is initialized by  $\mathbf{j} = \nabla\varphi$ , which is curl-free in the continuous sense. We choose  $\beta = 1$  and  $\lambda = 1$ . The stationary exact state  $\mathbf{q}^{\text{ex}}$  is maintained by the manufactured source vector

$$(5.6) \quad \mathbf{S}^{\text{ex}}(\mathbf{x}) := \nabla \cdot \mathbf{F}^{\text{EH}}(\mathbf{q}^{\text{ex}}(\mathbf{x})),$$

which is evaluated consistently for all six equations of the model.

The simulation is run on a sequence of successively refined meshes until the final time  $t_f = 0.1$ . Both the  $L^1$  and the  $L^2$  errors are reported in Table 2, showing first-order accuracy, as expected.

TABLE 2  
Euler–Heat convergence on moving meshes at  $t_f = 0.1$ .  $L^1$  and  $L^2$  errors and experimental orders of convergence for the primitive variables.

| $N_x = N_y$ | $\rho$       |       | $u$      |       | $v$      |       | $p$      |       | $j_x$    |       | $j_y$    |       |
|-------------|--------------|-------|----------|-------|----------|-------|----------|-------|----------|-------|----------|-------|
|             | $L^1$ errors |       |          |       |          |       |          |       |          |       |          |       |
| 20          | 3.126e-3     | -     | 5.034e-3 | -     | 5.034e-3 | -     | 3.113e-3 | -     | 5.027e-3 | -     | 5.027e-3 | -     |
| 30          | 2.126e-3     | 0.957 | 3.447e-3 | 0.940 | 3.447e-3 | 0.940 | 2.102e-3 | 0.974 | 3.436e-3 | 0.945 | 3.436e-3 | 0.945 |
| 40          | 1.657e-3     | 0.875 | 2.668e-3 | 0.901 | 2.668e-3 | 0.901 | 1.633e-3 | 0.887 | 2.631e-3 | 0.938 | 2.631e-3 | 0.938 |
| 50          | 1.352e-3     | 0.926 | 2.186e-3 | 0.905 | 2.186e-3 | 0.905 | 1.331e-3 | 0.930 | 2.145e-3 | 0.930 | 2.145e-3 | 0.930 |
| 60          | 1.153e-3     | 0.888 | 1.859e-3 | 0.904 | 1.859e-3 | 0.904 | 1.128e-3 | 0.921 | 1.810e-3 | 0.943 | 1.810e-3 | 0.943 |
|             | $L^2$ errors |       |          |       |          |       |          |       |          |       |          |       |
| 20          | 7.628e-3     | -     | 1.190e-2 | -     | 1.190e-2 | -     | 7.356e-3 | -     | 1.181e-2 | -     | 1.181e-2 | -     |
| 30          | 5.302e-3     | 0.903 | 8.353e-3 | 0.879 | 8.353e-3 | 0.879 | 5.025e-3 | 0.946 | 8.259e-3 | 0.888 | 8.259e-3 | 0.888 |
| 40          | 4.197e-3     | 0.821 | 6.492e-3 | 0.886 | 6.492e-3 | 0.886 | 3.949e-3 | 0.847 | 6.369e-3 | 0.914 | 6.369e-3 | 0.914 |
| 50          | 3.454e-3     | 0.887 | 5.375e-3 | 0.858 | 5.375e-3 | 0.858 | 3.241e-3 | 0.899 | 5.240e-3 | 0.887 | 5.240e-3 | 0.887 |
| 60          | 2.895e-3     | 0.982 | 4.590e-3 | 0.880 | 4.590e-3 | 0.880 | 2.715e-3 | 0.985 | 4.450e-3 | 0.910 | 4.450e-3 | 0.910 |

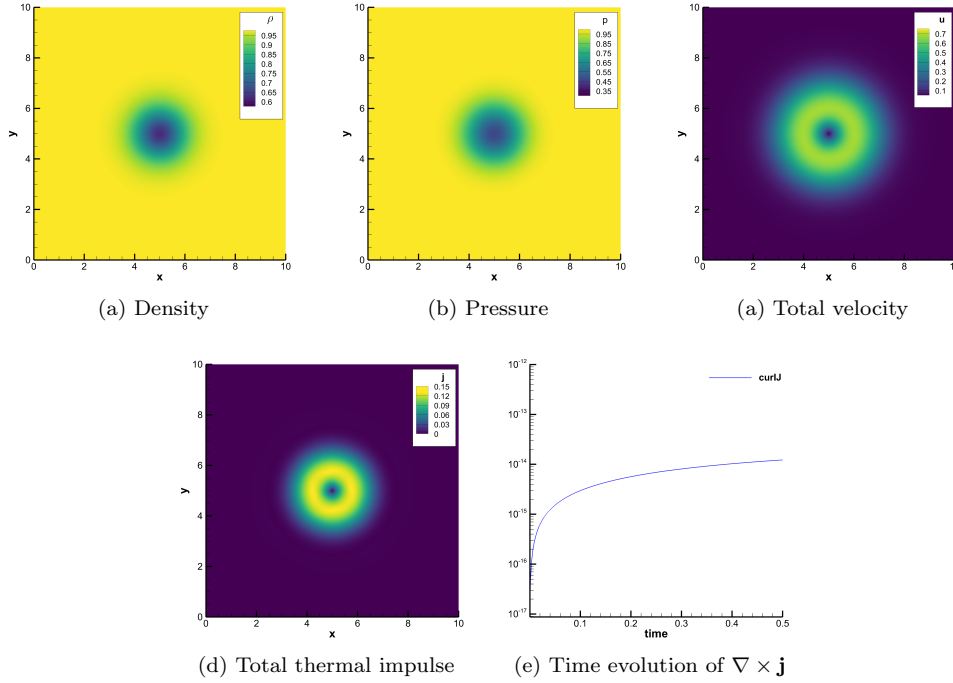


FIG. 4. Numerical solution of the Euler-Heat vortex with dissipation at the final time  $t_f = 0.5$  obtained using our involution preserving scheme on a  $N_x = N_y = 200$  mesh.

**5.3.2. Euler-Heat vortex problem with dissipation.** The computational domain is  $\Omega = [0, 10]^2$ . The initial condition reads

$$(5.7) \quad \rho = (1 + \delta\theta)^{\frac{1}{\gamma-1}}, \quad \begin{pmatrix} u \\ v \end{pmatrix} = \frac{5}{2\pi} e^{\frac{1-r^2}{2}} \begin{pmatrix} 5-y \\ x-5 \end{pmatrix}, \quad \delta\theta = -\frac{25(\gamma-1)}{8\gamma\pi^2} e^{1-r^2},$$

with  $p = \rho^\gamma$  and  $r = \sqrt{(x-5)^2 + (y-5)^2}$ . The thermal impulse is initially  $\mathbf{j} = \mathbf{0}$ . We set the hyperbolic heat parameter to  $\kappa = 10^{-2}$  and the target Fourier conductivity to  $K = 10^{-3}$ . The constant relaxation time is then  $\tau = K/\kappa^2$ , hence preserving a zero curl under the relaxation source  $-\mathbf{j}/\tau$  in the evolution equations for  $\mathbf{j}$ . The simulation is run until the final time  $t_f = 0.5$ . The resulting vortex and the discrete curl are shown in Figure 4.

This test is advanced by a symmetric source splitting: one half of the exact relaxation flow, the homogeneous constrained ALE step, and a second half relaxation flow. For the two components of  $\mathbf{j}$ , the centered nodal flux is used, whereas the remaining equations retain the stabilized numerical flux. On the initially Cartesian mesh, both this compatible flux and the scalar relaxation map preserve the discrete curl. Consequently, the constrained mesh problem is still posed at every Runge–Kutta stage, but its admissible minimum is the zero displacement and the computed grid velocity remains zero up to roundoff. This behavior is noteworthy: the method does not introduce mesh motion when the spatial discretization already satisfies the involution. The experiment is therefore complementary to the preceding convergence test, in which a nonzero ALE motion is required to enforce the curl constraint. The maximum and final raw discrete curl are both  $1.23 \times 10^{-14}$ .

**5.4. Geometric Conservation Law.** The proposed ALE discretization is also assessed in terms of the discrete Geometric Conservation Law (GCL), which guarantees that the numerical evolution of the cell volumes is exactly consistent with the prescribed mesh motion. This property is essential for the preservation of uniform solutions on moving meshes.

For each control volume  $\omega_c$ , the continuous GCL reads

$$(5.8) \quad \frac{d}{dt}|\omega_c| = \sum_{p \in \mathcal{P}(c)} \mathbf{w}_p \cdot \boldsymbol{\ell}_{pc},$$

which is the geometric counterpart of the ALE finite-volume formulation.

To verify the discrete GCL, we compare the cell volume obtained after updating the mesh coordinates through the Runge–Kutta scheme with the volume predicted along the same piecewise-linear displacement path. The corner vectors are evaluated at the midpoint of each segment, as in (4.4), rather than simply on the physical Runge–Kutta stage geometry. Denoting by  $|\omega_c|_{\text{geom}}^{n+1}$  the cell volume computed from the updated mesh, the GCL residual is defined as

$$(5.9) \quad R_{\text{GCL},c} = |\omega_c|_{\text{geom}}^{n+1} - |\omega_c|^n - \Delta t \sum_{s=1}^{N_{\text{RK}}} \mathbf{b}_s G_{c,s}^{\text{mid}},$$

where

$$(5.10) \quad G_{c,s}^{\text{mid}} = \sum_{p \in \mathcal{P}(c)} \mathbf{w}_p^{(s)} \cdot \boldsymbol{\ell}_{pc}(\mathbf{X}_{s-1/2})$$

is the exact swept-volume flux for the  $s$ -th displacement segment. This distinction is essential because polygonal area is quadratic in the coordinates.

The reported diagnostic is the normalized residual

$$(5.11) \quad \varepsilon_{\text{GCL}} = \max_c \frac{|R_{\text{GCL},c}|}{\max(|\omega_c^n|, |\omega_c^{n+1}|, \delta_m)},$$

with  $\delta_m$  denoting the machine precision of the floating-point arithmetic.

Furthermore, the same verification is performed at every Runge–Kutta stage, ensuring that the geometry evolution is consistent throughout the multistage integration. In all numerical experiments, both the stage-wise and the final GCL residuals remain at the level of machine precision, confirming the exact satisfaction of the discrete GCL, as shown in Figure 5. As discussed previously, for the Euler–Heat dissipation vortex test, the constrained mesh problem admits the zero displacement as its solution and the resulting grid velocity remains zero up to roundoff. Consequently, the mesh remains stationary and both the stage-wise and final GCL residuals are identically zero.

**6. Conclusions.** In this work, we have introduced an involution-preserving ALE finite volume framework for the numerical approximation of first-order hyperbolic systems on moving meshes. The main idea consists in exploiting the additional freedom provided by the ALE formulation to construct the mesh motion consistently with the differential involutions associated to the governing equations.

The method has been formulated in a cell-centered setting on initially quadrilateral meshes. Corner-based geometrical quantities are devised to define the discrete

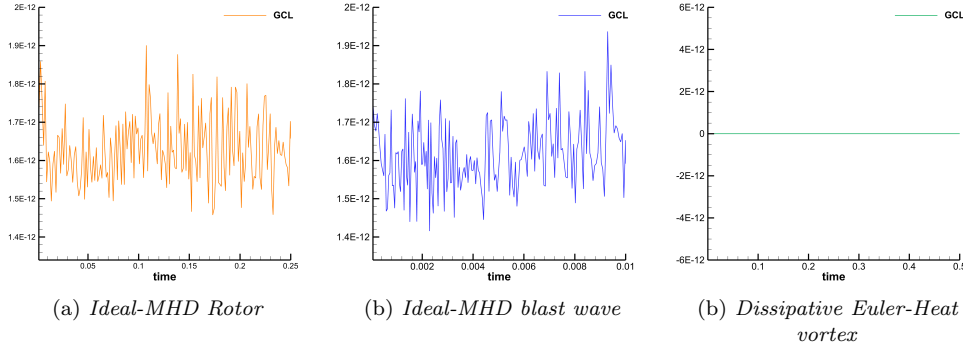


FIG. 5. Time evolution of the maximum normalized GCL residual  $\varepsilon_{\text{GCL}}$  for the different test cases.

differential operators, while all geometrical contributions are consistently updated along the mesh evolution. The resulting constrained mesh motion has been incorporated into a second-order Runge–Kutta time discretization, ensuring consistency between the intermediate stages, the evolving geometry, and the imposed differential constraint. The formulation also satisfies the Geometric Conservation Law, preventing the introduction of spurious errors emerging from the mesh motion.

The proposed approach has been investigated for two different hyperbolic systems. For the ideal MHD equations, the mesh motion is constructed in connection with the divergence-free involution of the magnetic field, whereas for the hyperbolic Euler–Heat system the corresponding construction is associated with the curl-free involution of the thermal impulse vector. The numerical results confirm the expected accuracy of the scheme and show that the relevant differential constraints remain satisfied throughout the evolution on moving meshes at the fully discrete level up to machine precision. Classical benchmark problems further demonstrate the capability of the method to handle non-trivial flow configurations while maintaining the prescribed structural properties.

Future developments will concern the extension of the proposed framework to higher-order spatial and temporal discretizations and to polygonal moving meshes. The satisfaction of multiple involutions at the same time will also be the subject of investigations, as well as the application of our novel strategy to other hyperbolic systems like hyperelasticity.

**Acknowledgments.** The authors would like to thank the framework offered by the SUN HYPE 2026 event organized in Crete in June 2026, where the work presented in this paper was conceived and its initial development took place. In particular, we thank the Technical University of Crete for hosting the event and the financial support received from the European Union through the ERC Starting Grant ALcHyMiA (grant agreement No. 101114995). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

## REFERENCES

- [1] D. S. BALSARA AND D. S. SPICER, *A staggered mesh algorithm using high order godunov fluxes to ensure solenoidal magnetic fields in magnetohydrodynamic simulations*, Journal of Computational Physics, 149 (1999), pp. 270–292.
- [2] A. J. BARLOW, P.-H. MAIRE, W. J. RIDER, R. N. RIEBEN, AND M. J. SHASHKOV, *Arbitrary Lagrangian–Eulerian methods for modeling high-speed compressible multimaterial flows*, Journal of Computational Physics, 322 (2016), pp. 603–665, <https://doi.org/10.1016/j.jcp.2016.07.001>.
- [3] E. BERNARDELLI, E. GABURRO, AND M. DUMBSER, *A structure-preserving semi-implicit finite volume scheme on vertex-staggered unstructured meshes*, arXiv preprint arXiv:2604.21906, (2026).
- [4] W. BOSCHERI AND F. DHAOUADI, *Structure preserving finite volume schemes on Voronoi grids: curl involution, asymptotic limit and thermodynamics*, J. Sci. Comput., 107 (2026), p. 80, <https://doi.org/10.1007/s10915-026-03251-6>.
- [5] W. BOSCHERI, R. LOUBÈRE, J. BRAEUNIG, AND P.-H. MAIRE, *A geometrically and thermodynamically compatible finite volume scheme for continuum mechanics on unstructured polygonal meshes*, J. Comput. Phys., 507 (2024), p. 112957.
- [6] J. BRACKBILL, *Fluid modeling of magnetized plasmas*, Space Science Reviews, 42 (1985), pp. 153–167.
- [7] J. U. BRACKBILL AND D. C. BARNES, *The effect of nonzero  $\nabla \cdot B$  on the numerical solution of the magnetohydrodynamic equations*, Journal of Computational Physics, 35 (1980), pp. 426–430.
- [8] S. CHIOCCETTI AND M. DUMBSER, *An exactly curl-free staggered semi-implicit finite volume scheme for a first order hyperbolic model of viscous two-phase flows with surface tension*, Journal of Scientific Computing, 94 (2023), p. 24.
- [9] F. DHAOUADI AND M. DUMBSER, *A structure-preserving finite volume scheme for a hyperbolic reformulation of the navier–stokes–korteweg equations*, Mathematics, 11 (2023), p. 876.
- [10] F. DHAOUADI AND S. GAVRILYUK, *An Eulerian hyperbolic model for heat transfer derived via Hamilton's principle: analytical and numerical study*, Proc. R. Soc. A, 480 (2024), p. 20230440.
- [11] V. A. DOBREV, P. KNUPP, T. KOLEV, K. MITTAL, AND V. Z. TOMOV, *The target-matrix optimization paradigm for high-order meshes*, SIAM J. Sci. Comput., 41 (2019), pp. B50–B68, <https://doi.org/10.1137/18M1167206>.
- [12] M. DUMBSER, A. UURIHNTSETSEG, AND O. ZANOTTI, *On arbitrary-lagrangian-eulerian one-step weno schemes for stiff hyperbolic balance laws*, Communications in Computational Physics, 14 (2013), pp. 301–327.
- [13] E. GABURRO, M. KLÍMA, M. BONAFINI, AND M. TAVELLI, *On the treatment of topology changes on 3d polyhedral moving meshes via 4d space-time hole-like elements in direct ale ader-dg methods*, Computer Methods in Applied Mechanics and Engineering, 463 (2027), p. 119434.
- [14] S. GOTTLIEB, C.-W. SHU, AND E. TADMOR, *Strong stability-preserving high-order time discretization methods*, SIAM Review, 43 (2001), pp. 89–112.
- [15] P. KNUPP, *Achieving finite element mesh quality via optimization of the jacobian matrix norm and associated quantities. part ii – a framework for volume mesh optimization and the condition number of the jacobian matrix.*, Int. J. Numer. Meth. Engng., 48 (2000), pp. 1165 – 1185.
- [16] R. LOUBÈRE, P.-H. MAIRE, M. SHASHKOV, J. BREIL, AND S. GALERA, *ReALE: A reconnection-based arbitrary-Lagrangian–Eulerian method*, Journal of Computational Physics, 229 (2010), pp. 4724–4761, <https://doi.org/10.1016/j.jcp.2010.03.011>.
- [17] P. MAIRE, *A high-order cell-centered lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes.*, J. Comput. Phys., 228 (2009), pp. 2391–2425.
- [18] M. SHASHKOV AND P. KNUPP, *Optimization-based reference-matrix rezone strategies for arbitrary Lagrangian-Eulerian methods on unstructured meshes*, Selçuk Journal of Applied Mathematics, 3 (2002), pp. 81–99.
- [19] D. S. SIDILKOVER, *Spurious vorticity in eulerian and lagrangian methods*, Journal of Computational Physics, 520 (2025), p. 113510, <https://doi.org/10.1016/j.jcp.2024.113510>.
- [20] P. TONON, R. SANCHES, K. TAKIZAWA, AND T. TEZDUYAR, *A linear-elasticity-based mesh moving method with no cycle-to-cycle accumulated distortion*, Comput. Mech., 67 (2021), pp. 413–434, <https://doi.org/10.1007/s00466-020-01941-y>.