Second order ADER scheme for advection-diffusion on moving overset grids with a compact transmission condition

Michel Bergmann, Michele Giuliano Carlino, Angelo Iollo

SECOND ORDER ADER SCHEME FOR ADVECTION-DIFFUSION ON MOVING OVERSET GRIDS WITH A COMPACT TRANSMISSION CONDITION*

MICHEL BERGMANN†‡, MICHELE GIULIANO CARLINO†‡, AND ANGELO IOLLO†‡

Abstract. We propose a space-time Finite Volume scheme on moving Chimera grids for a general advection-diffusion problem. Special care is devoted to grid overlapping zones in order to devise a compact and accurate discretization stencil to exchange information between different mesh patches. Like in the ADER method, the equations are discretized on a space-time slab. Thus, instead of time-dependent spatial transmission conditions between relatively moving grid blocks, we define interpolation polynomials on arbitrarily intersecting space-time cells at the block boundaries. Through this scheme, a mesh-free FEM-predictor/FVM-corrector approach is employed for representing the solution. In this discretization framework, a new space-time Local Lax-Friederichs (LLF) stabilization speed is defined by considering both the advective and diffusive nature of the equation. The numerical illustrations for linear and non-linear systems show that background and foreground moving meshes do not introduce spurious perturbation to the solution, uniformly reaching second order accuracy in space and time. Finally, it is shown that several foreground meshes, possibly overlapping and with independent displacements, can be employed thanks to this approach.

Key words. Chimera mesh, overset grid, Finite Volume, second order scheme, ADER, compact transmission condition, unsteady advection-diffusion

AMS subject classifications. 65M08, 65M55, 65Y99

1. Introduction.

One of the main difficulties for the simulation of a phenomenon modeled by a Partial Differential Equation (PDE) is the geometrical modeling of the computational domain with a single mesh block. This problem is especially relevant when the domain is complex or its shape and its topology evolve during the simulation. Classical approaches to tackle this problem include the Arbitrary Lagrangian-Eulerian (ALE) method, fictitious domain approaches and Chimera grids. ALE methods [18] allow a certain degree of mesh deformation and adaptation thanks to an appropriate reformulation of the governing equations and to sophisticated and efficient grid displacement algorithms. However, when the grid deformation leads to excessively stretched cells, a delicate (and computationally expensive) global re-meshing step may be necessary. In turn, this operation can introduce approximation irregularities that are caused by the interpolation of the solution from the old grid to the new one. In fictitious domain approaches, including immersed boundary or penalization methods, the original problem is discretised on a simple mesh, usually structured and cartesian, constant in time [14, 25, 1]. The grid hence does not necessarily fit the physical boundaries and special care must be taken to attain a sufficient degree of accuracy at the boundaries. Moreover, the presence of thin boundary layers can significantly reduce the computational advantages deriving from a simple meshing algorithm, because of the uniform aspect ratio of the mesh.

We focus our investigations on Chimera grids [35, 5, 22, 26]. Chimera grids consist of multiple overlapping mesh blocks that together define an overset grid used for spatially discretise a PDE [30, 31, 29]. Usually, one has a background mesh that includes one or more foreground mesh patches that are fitted to the physical domain boundaries. This mesh generation approach considerably simplifies the task of mesh adaptation in the case of boundary layers, changing geometry for an unsteady problem (e.g. fluid-structure interaction problems in fluid-dynamics) and for unsteady multiply connected domains [2, 3, 28, 4, 9]. Once the multiple mesh patches are generated, they are collated in order to obtain an appropriate overlapping zone between the neighboring blocks [22]. In the overlap zones, the exchange of solution information from one grid to another is performed. A compact transmission condition is generally sought in order to limit communications between the grids. Namely, a compact stencil only composed of the first layer of cells is defined around any cell.

In this paper, we propose a space-time Finite Volume scheme on Chimera grids. Our objective is to combine some aspects of an ALE approach, notably its flexibility with respect to grid displacement and deformation, to the multi-block discretization strategy of overset grids. In particular, we will devote special

*Submitted to the editors 01/22/2021.

†INRIA Bordeaux Sud-Ouest, 200 Avenue de la Vieille Tour, 33405 Talence, France. (michel.bergmann@inria.fr, michele-giuliano.carlino@inria.fr, angelo.iollo@inria.fr)

‡IMB, Institut de Mathématiques de Bordeaux, 351 cours de la Libération, 33405 Talence, France.

care to grid overlapping zones in order to devise a compact and accurate discertization stencil to exchange information between different mesh patches, in the spirit of previous works on cartesian hierarchical grids [27]. We then apply this approach to integrate linear and non-linear Advection-Diffusion partial differential equations and show how the method can exploit the versatility of the Chimera meshes to reach second order accuracy in unsteady multiply connected domains.

The numerical solution on Chimera grids is obtained by exchanging data through the fringe cells at the overlapping zone. For example, in [10, 15, 36, 21], fringe (namely *donor*) cells of a block in proximity of the overlapping zone provide the information to the fringe (i.e., *receptor*) cells of another block by polynomial interpolation. In [16] a coarse grid is automatically generated and a connection of interpolation information at the overlapping zone is presented through a multigrid approach.
Another way of making the different blocks communicate is to use proper Domain Decomposition (DD) methods (e.g., Schwartz, Dirichlet/Neumann or Dirichlet/Robin methods). In particular, each mesh block is considered as a decomposition of the domain and the overlapping zones are the interfaces for coupling the different blocks. Accordingly to these approaches, typically iterative discrete methods are employed. For this two way communication, the reader is referred to [19] for further details.
In the same framework, other approaches connect the background and the foreground meshes, such as the DRAGON grids [20] for which the overlapping zone is replaced by a nonstructured grid during a further stage by preserving the body-fitting advantages of the Chimera meshes.
In contrast, here we derive a second order compact transmission condition by properly defining a set of cells, i.e. the *stencil*, that belong *both* to the back- and foreground meshes, over which the solution is interpolated in space and time by an appropriate polynomial. This hybrid stencil allows a smooth discretization transition from one block to another. In particular, first a mesh-free discontinuous FEM-solution is recovered and then a FVM-correction is performed in any cell by using information provided by near cells. Thus, for fringe cells, the solution is obtained by combining values from different grids.

The Arbitrary high order DERivatives (ADER) method provides an ideal setting for pursuing our purpose. In [11, 33, 32, 8], the authors presented a method to recover an accurate solution for hyperbolic PDEs with an arbitrary order of accuracy on a single mesh block. More recently, in [7] the authors presented an ADER Discontinuos Galerkin scheme with *a posteriori* subcell finite volume limiter on fixed and moving grids such as space-time adaptive Cartesian AMR meshes. The numerical scheme treats the temporal variable indistinctly with respect to the spatial variables by defining the solution on a space-time slab. This discretization approach, therefore, allows us to re-consider the problem of Chimera grids transmission conditions: instead of time-dependent spatial transmission conditions between relatively moving grid blocks, we define interpolation polynomials on arbitrarily intersecting space-time cells at the block boundaries.
In the ADER scheme a local space-time weak solution of the problem from the generic time t to $t + \Delta t$ is computed in every single space-time cell. This solution is defined as the *predictor*. The prediction step is local and hence embarrassingly parallel, because the solution is calculated independently of the information of the neighbouring cells. Then, in the subsequent stage of *correction*, the computation of a space-time numerical flux between neighboring cells provides the appropriate stabilization of the integration scheme. We extend this prediction-correction method to Advection-Diffusion PDEs on overset grids and propose a space-time flux among the space-time cells that provides improved stabilization and precision as it takes into account both the advective and diffusive nature of the equation.

Let $\Omega(t) \subset \mathbb{R}^d$ be the time-dependent computational domain and let T be a positive real. In the following we consider the parabolic problem: *find* $u : \Omega(t) \times [0, T] \to \mathbb{R}^\delta$ *such that*

$$\partial_t \boldsymbol{u} + \nabla \cdot \boldsymbol{F}(\boldsymbol{u}, \nabla \boldsymbol{u}) = \boldsymbol{f}, \quad \boldsymbol{x} \in \Omega(t), \quad t \in [0, T], \tag{1.1}$$

closed with appropriate initial and boundary conditions. Problem (1.1) is a rather general representation of an advection-diffusion model. In (1.1) the diffusive-convective vector $\boldsymbol{F}(\boldsymbol{u}, \nabla \boldsymbol{u})$, eventually nonlinear, and the force term $\boldsymbol{f}(\boldsymbol{x}, t)$ are defined. In particular, the problem is linear when the diffusive-convective term is written as $\boldsymbol{F}(\boldsymbol{u}, \nabla \boldsymbol{u}) = A\boldsymbol{u} - \nu \nabla \boldsymbol{u}$, where $A : \Omega \times [0, T] \to \mathbb{R}^{\delta \times \delta}$ is the advective field and $\nu : \Omega \times [0, T] \to \mathbb{R}_+$ is the diffusion parameter.

In Section 2 the formal definition of the overset (Chimera) grid is given. The predictor-corrector method on a Chimera mesh is then derived in Section 3. In Section 4 the new Local-Lax-Friederichs (LLF) stabilization term is introduced and contrasted with the LLF term from the literature. Section 5 is devoted to the numerical results. In particular, first the second order analysis is conducted on linear 1D and 2D test cases; successively, we focus on the stability of the method by comparing the performances of the different LLF fluxes. At the end of the numerical test cases section, we show results for a nonlinear system of PDEs, for multiblock grid setting, meshes and time-dependent overset grids for multiply connected domain. Conclusions are reported in Section 6.

2. The overset grid. An overset grid or Chimera mesh is a set of mesh blocks covering the computational domain. Each block may overlap other block(s) in some particular sub-region(s) said *overlapping zone(s)*. Once the multiple mesh patches are generated, they are collated in order to generate an appropriate topology [22]. Consequently, an overlapping zone between two neighbouring blocks is defined. For the sake of simplicity with no loss of generality, the whole method is explained by considering a two blocks overset grid (i.e., the background and the foreground meshes). For multiple-block meshes (e.g. $\mathcal{T}_1, \dots, \mathcal{T}_N$), a hierarchy of meshes from the background to the foreground is defined (e.g. $\mathcal{T}_1 < \cdots < \mathcal{T}_N$). Successively the presented algorithm for setting the overset grid is performed from one mesh to the union of all other meshes towards the background (e.g. $\mathcal{T}_i$ for $\bigcup_{j=1}^{i-1} \mathcal{T}_j$ for any $i = 2, \dots, N$). In Section 5.4.2 of test cases, a multiple-block setting is presented. Figure 1 shows an overset grid; in black there is the *background* mesh and in blue the *foreground* mesh. In particular, the foreground mesh can move and deform. The overlapping zone is necessary for the communication and data transfer from one mesh to the other.
In this work, the cell of any block mesh is considered quadrilateral. In particular, when all the cells are squared, the mesh is uniform. When the cells are either squared or rectangular and the edges are oriented as the Cartesian axes, the mesh is said to be Cartesian.

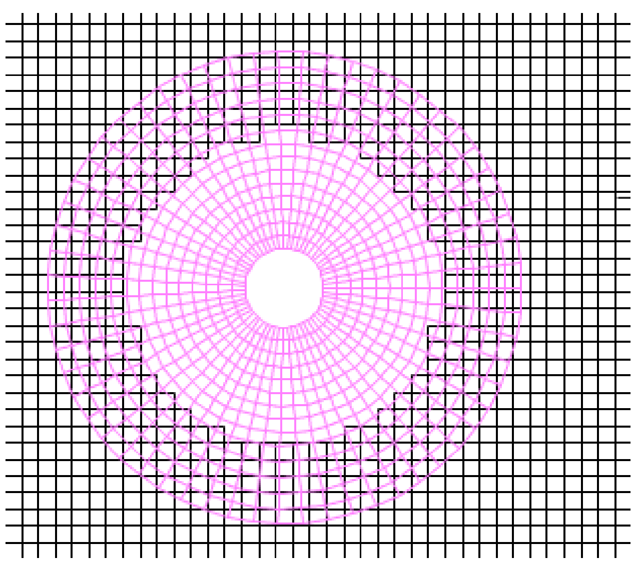


Fig. 1: Example of Chimera grid configuration. In black there is the *background* mesh and in pink the *foreground* mesh.

2.1. The automatic definition of the stencil at the transmission condition. Let $\mathcal{T}_k = \{\Omega_i^k\}_{i=1}^{N_k}$ be the partition composed of N_k cells referring to the k-th block mesh (in order to simplify the notation, we will omit the superscript k to the cell Ω_i^k by writing Ω_i), moreover, let $\mathcal{S}_i$ be the stencil centered over the cell Ω_i. Thus, stencil $\mathcal{S}_i$ is the set collecting the indexes of neighboring cells to Ω_i. By abuse of language, sometimes we will refer to the physical set $\Omega_i \cup \bigcup_{j \in \mathcal{S}_i} \Omega_j$ as the stencil.
It is possible to distinguish two classes of cells with respect to their proximity to the overlapping interface. The definition of the stencil depends on the class.
If cell Ω_i is not at the boundary of the overlapping zone (Figure 2a), the stencil $\mathcal{S}_i$ is composed of all the

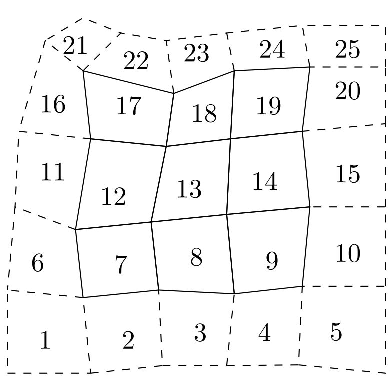
cells Ω_j sharing at least one vertex with Ω_i. Thus, if Ω_i belongs to the partition $\mathcal{T}_1$, all cells Ω_j, with $j \in \mathcal{S}_i$, also belong to $\mathcal{T}_1$.
If the cell Ω_i of partition $\mathcal{T}_k$ is at the boundary of the interface, it is no longer possible to use the criterion of the cells sharing at least a vertex. In fact, there will be at least one edge e_{il} not shared by any other cell of the same partition (see right edge of cell Ω_{16} in Figure 2b). For these cells, we aim in automatically finding the other cells of partition $\mathcal{T}_j$ $(j \neq k)$ belonging to the stencil. Let the extremes of the edge be indicated as $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ and its middle point with $\boldsymbol{v}_3$, respectively. Point $\boldsymbol{c}_\star$ is the center of mass of generic cell $\Omega_\star$. For our numerical tests, Algorithm 2.1 is adopted through the two steps:

1. look for the nodes of cells of the other partition $\mathcal{T}_j$ minimizing the Euclidean distance with respect to points $\boldsymbol{v}_\mu$, $\mu = 1, 2, 3$, (line 5, see Figure 3a);
2. compute the symmetric points $\tilde{\boldsymbol{v}}_\mu$ of center $\boldsymbol{c}_i^k$ with respect to points $\boldsymbol{v}_\mu$ for $\mu = 1, 2, 3$ (line 6), then look for the cells of partition $\mathcal{T}_j$ whose centers minimize the Euclidean distance with the three symmetric points (line 7, see Figure 3b).

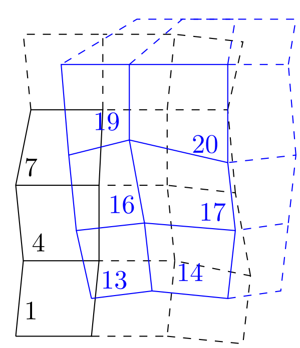
For the edges shared by other cells in the same partition, the cells of the stencil will be those ones sharing at least one vertex (as cells of indexes 13, 14, 17, 19 and 20 in Figure 2b).
The routine presented in this section will be run whenever the foreground mesh configuration as well as the hole change.
Algorithm 2.1 could not define a compact stencil in the case of widely different mesh spacing. In this case, more than three points $\boldsymbol{v}_\mu$ can be considered for lines 5 and 6. Moreover a weighted symmetry (possibly led by the different spacing) can be performed at line 6.

Algorithm 2.1 Compute stencil for cells at the boundary of the overlapping zone.

Input: $\Omega_i^k, e_{il}^k, \mathcal{T}_j, \mathcal{S}_i^k$; ▷ $j \neq k$, i.e. $\mathcal{T}_j$ is the other partition with respect to $\mathcal{T}_k$
1: Initialize $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ as the two vertexes of edge e_{il}^k;
2: $\boldsymbol{v}_3 \leftarrow (\boldsymbol{v}_1 + \boldsymbol{v}_2)/2$; ▷ Middle point of edge e_{il}^k
3: $\mathcal{Z}_j \leftarrow \emptyset$; ▷ Temporary set of indexes of partition $\mathcal{T}_j$
4: **for** $\mu = 1, 2, 3$ **do**
5: $\mathcal{Z}_j \leftarrow \mathcal{Z}_j \cup \{n = 1, \dots, N_j : \|\boldsymbol{v}_\mu - \boldsymbol{c}_n^j\| \leq \|\boldsymbol{v}_\mu - \boldsymbol{c}_m^j\| \quad \forall m = 1, \dots, N_j\}$;
6: $\tilde{\boldsymbol{v}} \leftarrow 2\boldsymbol{v}_\mu - \boldsymbol{c}_i^k$; ▷ Symmetric point of cellcenter $\boldsymbol{c}_i^k$ of Ω_i^k with respect to $\boldsymbol{v}_\mu$
7: $\mathcal{Z}_j \leftarrow \mathcal{Z}_j \cup \{n = 1, \dots, N_j : \|\tilde{\boldsymbol{v}} - \boldsymbol{c}_n^j\| \leq \|\tilde{\boldsymbol{v}} - \boldsymbol{c}_m^j\| \quad \forall m = 1, \dots, N_j\}$;
8: $\mathcal{S}_i^k \leftarrow \mathcal{S}_i \cup \mathcal{Z}_j$;
9: **return** $\mathcal{S}_i^k$

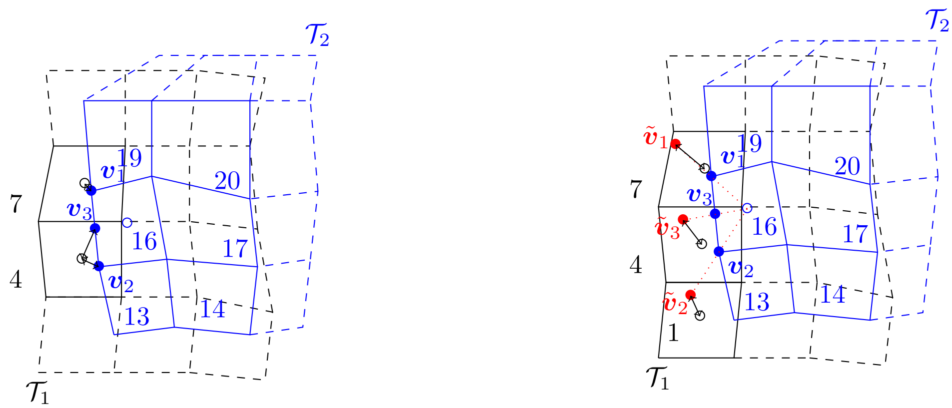


(a) A stencil of cells in the same partition. Continuous line for the stencil $\mathcal{S}_{13} = \{7, 8, 9, 12, 14, 17, 18, 19\}$.



(b) A stencil of cells not belonging to the same partition. Continuous line for the stencil $\mathcal{S}_{16} = \{1, 4, 7, 13, 14, 17, 19, 20\}$.

Fig. 2: Two possible stencils: on the left the stencil is in the same partition; on the right the stencil is composed of cells not belonging to the same partition.



(a) First step: by identifying the vertexes $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ and the middle point $\boldsymbol{v}_3$ of the edge on the boundary cell Ω_{16} (blue full dots), look for the nodes of cells in the partition $\mathcal{T}_1$ (black empty dots) minimizing the Euclidean distance with respect to those points.

(b) Second step: by identifying the symmetric points $\tilde{\boldsymbol{v}}_\mu$, $\mu = 1, 2, 3$, (red full dots) of the node of the cell Ω_{16} (blue empty dot) with respect to the vertexes and the middle point of the not shared edge, look for the nodes of cells in the partition $\mathcal{T}_1$ minimizing the Euclidean distance to those points.

Fig. 3: The two steps for the research of cells in the partition $\mathcal{T}_1$ for the cell $\Omega_{16} \in \mathcal{T}_2$.

3. The numerical method. Once the stencil has been defined, the numerical method can both numerically solve problem (1.1) and eventually evolve the overset grid. In this section the scheme is presented. The method consists in a FEM-predictor FVM-corrector scheme stabilised with a Local Lax-Friederichs approach whose stabilization coefficient is explained in the following section.

3.1. Local polynomial reconstruction. The first step of the numerical method is to recover a reconstruction of the solution over any point of the actual cell Ω_i. Since the scheme is cell-centered, at time t^n, we would like to extend (at least locally) the solution to the whole cell by exploiting the information in the cells of the stencil referring to Ω_i^n. In order to explain the reconstruction, let us consider a generic regular[1] function $\phi : E \to \mathbb{R}$ by identifying the stencil $E = \Omega_i^n \cup \bigcup_{j\in\mathcal{S}_i} \Omega_j^n$. We remark that, due to the overlapping zone, the cell composing the subdomain E does not necessary fulfill the non-overlapping condition, i.e., it could be verified that there is a couple of indexes $k, l \in \{i\} \cup \mathcal{S}_i$ such that $\Omega_k^n \cap \Omega_l^n \neq \emptyset$. Let us suppose to know the value of function ϕ over the center of mass $(x_k, y_k) = \boldsymbol{x}_k$, with $k \in \{i\} \cup \mathcal{S}_i$, of any Ω_k composing E. We would like to have a polynomial function $\Pi_i\phi(x, y)$ for any $(x, y) \in E$ by using the knowledge of the function ϕ only on the centers of mass. Let us define $\phi_k = \phi(x_k, y_k)$. For any $(x, y) \in E$ it is always possible to write the Taylor's polynomial truncated to the quadratic terms with respect to ϕ_i:

$$\begin{aligned} \phi(x,y) = \phi_i &+ (\partial_x\phi)_i\,(x-x_i) + (\partial_y\phi)_i\,(y-y_i) + (\partial^2_{xy}\phi)_i\,(x-x_i)(y-y_i) \\ &+ \frac{1}{2}(\partial^2_{xx}\phi)_i\,(x-x_i)^2 + \frac{1}{2}(\partial^2_{yy}\phi)_i\,(y-y_i)^2 + \mathcal{O}(H^3), \end{aligned} \tag{3.1}$$

with $H = \max\{|x - x_i|, |y - y_i|\}$. In the expansion (3.1) all the derivatives of ϕ_i are unknown. Moreover, by renaming those derivatives as

$$p_1 = (\partial_x\phi)_i \quad p_2 = (\partial_y\phi)_i \quad p_3 = (\partial^2_{xy}\phi)_i \quad p_4 = (\partial^2_{xx}\phi)_i \quad p_5 = (\partial^2_{yy}\phi)_i, \tag{3.2}$$

the Taylor's expansion (3.1) can be seen as a linear combination of the components of the basis $\{1, x - x_i, y - y_i, (x - x_i)(y - y_i), \frac{1}{2}(x - x_i)^2, \frac{1}{2}(y - y_i)^2\}$ which defines the polynomial space function $\mathcal{Q}_2$ of quadratic polynomials centered in $\boldsymbol{x}_i$; thus the polynomial interpolation function $\Pi_i\phi$ reads:

$$\Pi_i\phi(x,y) = \phi_i + p_1(x-x_i) + p_2(y-y_i) + p_3(x-x_i)(y-y_i) + \frac{1}{2}p_4(x-x_i)^2 + \frac{1}{2}p_5(y-y_i)^2, \tag{3.3}$$

[1]We require at least $\phi \in C^2(E)$.

with the polynomial coefficients p_l, $l = 1, \dots, 5$, to be sought. By imposing as constraint that the polynomial $\Pi_i\phi(x, y)$ exactly coincides with the function ϕ on the nodes, i.e. $\Pi_i\phi(x_j, y_j) = \phi_j$ for any $j \in \mathcal{S}_i$, the system in the unknown polynomial coefficients arises:

$$\begin{bmatrix} h^x_{ik} & h^y_{ik} & h^x_{ik}h^y_{ik} & \frac{1}{2}(h^x_{ik})^2 & \frac{1}{2}(h^y_{ik})^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h^x_{ij} & h^y_{ij} & h^x_{ij}h^y_{ij} & \frac{1}{2}(h^x_{ij})^2 & \frac{1}{2}(h^y_{ij})^2 \end{bmatrix} \begin{bmatrix} p_1 \\ \vdots \\ p_5 \end{bmatrix} = \begin{bmatrix} \delta\phi_{ik} \\ \vdots \\ \delta\phi_{ij} \end{bmatrix}, \tag{3.4}$$

with $h^x_{ij} = x_j - x_i$, $h^y_{ij} = y_j - y_i$ and $\delta\phi_{ij} = \phi_j - \phi_i$, for $j \in \mathcal{S}_i$. The algebraic system (3.4) has to be solved in least-square sense if $|\mathcal{S}_i| > 5$. Moreover, if the chosen polynomial basis is not reduced, namely if the Taylor's expansion (3.1) is arrested to the bi-linear or linear terms, the stencil has to contain at least 5 cells in order to ensure a solution for (3.4). The proposed $\mathcal{P}_2$-interpolation, with the second-order accurate scheme, fulfills the condition for the accuracy in the interpolation for overlapping zones whose depth d_o degrades as the characteristic length h of the chimera mesh (i.e., $d_o = \mathcal{O}(h)$) [9].
This method allows to locally reconstruct all over the stencil a given function. If the function is defined over the computational domain $\Omega \subset \mathbb{R}^2$ and it is (at least locally) C^2, then the reconstruction is locally computed over any stencil and the ensured order of convergence is 3. On the contrary, if the solution presents propagating shock waves or discontinuities, this interpolation is no longer adequate because of well-known Gibbs' phenomenon, for which spurious oscillation are produced near the discontinuity. For those cases, other interpolation could be adopted, such as the central weighted ENO for hyperbolic equations for moving meshes in [11].
In the sequel, the local polynomial reconstruction $\Pi_i\boldsymbol{u}^n$ will be referred as $\boldsymbol{w}^n_i$.

3.2. Local space-time Galerkin predictor. Let be the time interval $[0, T]$ subdivided in N subintervals $[t^n, t^{n+1}]$, with $n = 0, \dots, N-1$; thus for a generic time-dependent variable $\boldsymbol{g}(t)$, we define $\boldsymbol{g}^n$ for $\boldsymbol{g}^n = \boldsymbol{g}(t^n)$. In particular, the domain Ω^n and the solution $\boldsymbol{u}^n$ at time t^n are considered the actual spatial configuration and the actual time, respectively. Let $\mathcal{C}^n_i = \Omega_i(t) \times [t^n, t^{n+1}]$ be the physical space-time cell whose lower and upper bases represent the evolution of cell $\Omega_i(t)$ from t^n to t^{n+1}. First, the governing equation (1.1) is rewritten with respect to a space-time reference system identified by the independent variables $\boldsymbol{\xi} \equiv (\xi, \eta, \tau)$ in the unit cube $\hat{\mathcal{C}} = [0, 1]^3$. Let $\boldsymbol{\Xi} = (\xi, \eta)$ be the reference spatial vector. Inspired by [17], the governing equation is discretized using an efficient nodal formulation of space-time nodes given by a tensor product of Gauss-Legendre quadrature points along space and time directions. This choice defines an L^2-orthogonal Lagrange basis used for the definition of the Galerking solution. For our purposes, the single direction nodes over the unit interval $[0, 1]$ are $\{(5 - \sqrt{15})/10; 1/2; (5 + \sqrt{15})/10\}$. Consequently, over a space-time cell there will be 27 Gauss-Legendre nodes $\hat{\boldsymbol{\xi}}_m$ and 27 Lagrange polynomial $\theta_l : \hat{\mathcal{C}} \to \mathbb{R}$ such that $\theta_l(\hat{\boldsymbol{\xi}}_m) = \delta_{lm}$ and $\int_{\hat{\mathcal{C}}} \theta_l\theta_m \,\mathrm{d}\boldsymbol{\xi} = \delta_{lm}\|\theta_l\|^2_{L^2(\hat{\mathcal{C}})}$, with δ_{lm} the Kronecher's symbol. Let $\mathfrak{m} : \{1, 2, 3\}^3 \to \{1, \dots, 27\}$ be a discrete map from a single direction index to the global three dimensional index defined as

$$\mathfrak{m}(i, j, k) = ij + (j-1)(3-i) + 9(k-1),$$

where indexes $i, j, k \in \{1, 2, 3\}$ lead the discretization along ξ, η, τ, respectively. By denoting the Gauss-Legendre nodes with $\hat{\xi}_i$, $\hat{\eta}_j$ and $\hat{\tau}_k$ along ξ, η and τ, respectively, and with $\theta^\xi_i(\xi)$, $\theta^\eta_j(\eta)$ and $\theta^\tau_k(\tau)$ the Lagrange polynomial for ξ-, η- and τ-directions, respectively, the three dimensional Gauss-Legendre node $\hat{\boldsymbol{\xi}}_l$ and its associated Lagrange's polynomial $\theta_l(\boldsymbol{\xi})$ read

$$\hat{\boldsymbol{\xi}}_l = (\hat{\xi}_i, \hat{\eta}_j, \hat{\tau}_k); \quad \theta_l(\xi, \eta, \tau) = \theta^\xi_i(\xi)\theta^\eta_j(\eta)\theta^\tau_k(\tau),$$

with index $l = \mathfrak{m}(i, j, k)$.
We want to solve the following problem: *find* $\boldsymbol{q} : \mathcal{C}^n_i \to \mathbb{R}^\delta$ *such that*

$$\begin{cases} \partial_t\boldsymbol{q} + \nabla \cdot \boldsymbol{F}(\boldsymbol{q}, \nabla\boldsymbol{q}) = \boldsymbol{f} & \text{in } \mathcal{C}^n_i \\ \boldsymbol{q}|_{t=t^n} = \boldsymbol{w}^n_i & \text{on } \Omega^n_i \end{cases}, \tag{3.5}$$

which is problem (1.1) restricted to the space-time cell $\mathcal{C}^n_i$ and redefined as a boundary value problem. We denote with $\boldsymbol{q}_h$ as the discretized solution of (3.5). In order to refer problem (3.5) to the reference domain

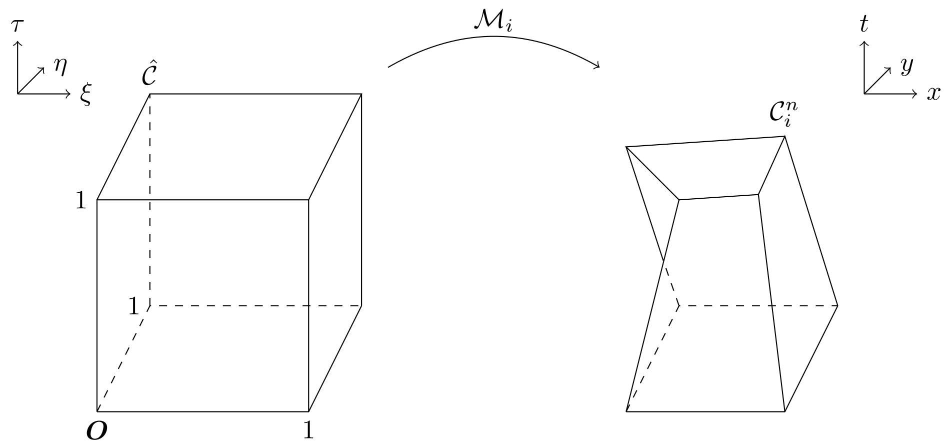


Fig. 4: Representation of the map $\mathcal{M}_i$ from the reference space-time cell $\hat{\mathcal{C}}$ to the physical space-time cell $\mathcal{C}_i^n$.

$\hat{\mathcal{C}}$, we use a map $\mathcal{M}_i : \hat{\mathcal{C}} \to \mathcal{C}_i^n$

$$\mathcal{M}_i : \begin{cases} x = x(\xi, \eta, \tau) \\ y = y(\xi, \eta, \tau) \\ t = t^n + \Delta t\, \tau \end{cases}, \tag{3.6}$$

such that any space-time point $\boldsymbol{x} \equiv (x, y, t)$ in the physical space-time cell $\mathcal{C}_i^n$ is a function $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi})$, with $\boldsymbol{\xi} \in \hat{\mathcal{C}}$ (see Figure 4). Time t is considered as a linear function of τ. From map (3.6), we define the Jacobian matrix J as

$$J = \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\boldsymbol{\xi}} = \begin{bmatrix} x_\xi & x_\eta & x_\tau \\ y_\xi & y_\eta & y_\tau \\ 0 & 0 & \Delta t \end{bmatrix}, \tag{3.7}$$

whose inverse is

$$J^{-1} = \frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}\boldsymbol{x}} = \begin{bmatrix} \xi_x & \xi_y & \xi_t \\ \eta_x & \eta_y & \eta_t \\ 0 & 0 & 1/\Delta t \end{bmatrix} = \begin{bmatrix} J_s^{-1} & \boldsymbol{\Xi}_t \\ \boldsymbol{0} & 1/\Delta t \end{bmatrix}. \tag{3.8}$$

In the above notation, we call J_s^{-1} the restriction to the spatial coordinates of the inverse of the Jacobian matrix

$$J_s^{-1} = \begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix}. \tag{3.9}$$

and $\boldsymbol{\Xi}_t = [\xi_t, \eta_t]^T$ the derivative of the spatial reference vector with respect to time. Through (3.9), the problem in the reference domain reads

$$\partial_\tau \boldsymbol{q} + \Delta t\, \boldsymbol{\mathcal{F}}^*(\hat{\nabla}\boldsymbol{q}) + \Delta t J_s^{-T} \hat{\nabla} \cdot \boldsymbol{\mathcal{F}}^{**}(\boldsymbol{q}, \hat{\nabla}\boldsymbol{q}) = \Delta t\, \boldsymbol{f}, \tag{3.10}$$

where

$$\partial_t \boldsymbol{q} = \frac{\partial_\tau \boldsymbol{q}}{\Delta t} + \mathcal{F}^*(\hat{\nabla}\boldsymbol{q}); \quad \boldsymbol{\mathcal{F}}^*(\hat{\nabla}\boldsymbol{q}) = \hat{\nabla} q\, \boldsymbol{\Xi}_t; \quad \boldsymbol{\mathcal{F}}^{**}(\boldsymbol{q}, \hat{\nabla}\boldsymbol{q}) = \boldsymbol{F}(\boldsymbol{q}, J_s^{-T}\hat{\nabla}\boldsymbol{q}) = (\boldsymbol{\mathcal{F}}_\xi^{**}, \boldsymbol{\mathcal{F}}_\eta^{**}); \quad \hat{\nabla} = \begin{bmatrix} \partial_\xi \\ \partial_\eta \end{bmatrix}.$$

The hat differential operators refer to reference space variables ξ and η in the reference space-time cell $\hat{\mathcal{C}}$. By abuse of notation and for sake of simplicity, we call all functions involved in both equations (3.5) and (3.10) with the same symbol (e.g., $\boldsymbol{q}$ and $\boldsymbol{f}$) even though they take inputs in the physical space-time cell $\mathcal{C}_i^n$ and in the reference space-time cell $\hat{\mathcal{C}}$, respectively. In order to weaken equation (3.10), the following functional space is defined:

$$\Theta = \left\{ v \in H^1(\hat{\mathcal{C}} : [0,1] \ni \tau \mapsto v(\xi,\eta,\tau) \in L^2((0,1)^2) \right\}$$

being the subspace of Sobolev space $H^1(\hat{\mathcal{C}})$ of functions $L^2((0,1)^2)$-integrable at any fixed reference time τ. Moreover, the following notation is introduced:

$$\langle f, \boldsymbol{g} \rangle = \int_{\hat{\mathcal{C}}} f \boldsymbol{g} \; \mathrm{d}\boldsymbol{\xi}; \quad [f, \boldsymbol{g}]_\tau = \int_0^1 \int_0^1 f(\xi,\eta,\tau) \boldsymbol{g}(\xi,\eta,\tau) \; \mathrm{d}\boldsymbol{\Xi}. \quad \forall f \in \Theta, \quad \forall \boldsymbol{g} \in \Theta^D \quad (D = 1,\dots,\delta).$$

For our purposes, functional space Θ is identified as a test space and the following trial functional spaces is defined:

$$Q = \left\{ v \in \Theta : v(\xi,\eta,0) = w_k^n \wedge J^{-1} \begin{bmatrix} \hat{\nabla} v \\ \partial_\tau v \end{bmatrix} \in L^2(\hat{\mathcal{C}}; \mathbb{R}^3) \right\},$$

where w_k is the k-th component of the interpolated polynomial $\boldsymbol{w}^n$. By multiplying left and right side of (3.10) by a generic test function $\theta \in \Theta$ and by integrating over the reference space-time cell $\hat{\mathcal{C}}$, the problem reads: *find* $\boldsymbol{q} \in Q^\delta$ *such that*

$$(3.11) \quad [\theta, \boldsymbol{q}]_1 - \langle \partial_\tau \theta, \boldsymbol{q} \rangle + \Delta t \, \langle \theta, \boldsymbol{\mathcal{F}}^*(\hat{\nabla}\boldsymbol{q}) \rangle + \Delta t \, \langle \theta, J_s^{-T} \hat{\nabla} \cdot \boldsymbol{\mathcal{F}}^{**}(\boldsymbol{q}, \hat{\nabla}\boldsymbol{q}) \rangle = \Delta t \, \langle \theta, \boldsymbol{f} \rangle + [\theta, \boldsymbol{w}^n]_0 \quad \forall \theta \in \Theta,$$

with $[\theta, \boldsymbol{w}^n]_0 = \int_0^1 \int_0^1 \theta(\xi,\eta,0) \boldsymbol{w}^n(\xi,\eta) \; \mathrm{d}\Xi$. For the Galerkin solution $\boldsymbol{q}_h$ and the convective-diffusive terms $\boldsymbol{\mathcal{F}}^*$ and $\boldsymbol{\mathcal{F}}^{**}$ in the reference domain, a Lagrangian polynomial expansion is performed, i.e., by adopting the Einstein's notation, $\boldsymbol{q}_h = \theta_l \hat{\boldsymbol{q}}_l$ and $\boldsymbol{\mathcal{F}}_h^\star = \theta_l \hat{\mathcal{F}}_l^\star$, with $\star = *, **$, where $\hat{\boldsymbol{q}}_l = \boldsymbol{q}(\hat{\boldsymbol{\xi}}_l)$ and $\hat{\boldsymbol{\mathcal{F}}}_l^\star = \boldsymbol{\mathcal{F}}^\star|_{\hat{\boldsymbol{\xi}}_l}$. Considering as the test function the k-th Lagrangian polynomial θ_k and by using the Lagrange expansion, we rewrite equation (3.11) as:

$$(3.12) \quad \begin{aligned} &([\theta_k, \theta_l]_1 - \langle \partial_\tau \theta_k, \theta_l \rangle) \hat{\boldsymbol{q}}_l + \Delta t \langle \theta_k, \theta_l \rangle \hat{\boldsymbol{\mathcal{F}}}_l^* + \Delta t \langle \theta_k, (\xi_x \partial_\xi + \eta_x \partial_\eta) \theta_l \rangle \boldsymbol{\mathcal{F}}_{\xi,l}^{**} \\ &\quad + \Delta t \langle \theta_k, (\xi_y \partial_\xi + \eta_y \partial_\eta) \theta_l \rangle \boldsymbol{\mathcal{F}}_{\eta,l}^{**} = \Delta t \langle \theta_k, \boldsymbol{f} \rangle + [\theta_k, \boldsymbol{w}^n]_0, \end{aligned}$$

for any $k = 1,\dots,27$.
In the left hand side of (3.12), we remark that the arising matrices have a sparse pattern due to the L^2-orthogonality of the Lagrangian basis (e.g. the mass matrix by $\langle \theta_k, \theta_l \rangle$ is diagonal). Matrices involving the derivatives of the map $\mathcal{M}_i$, i.e. $\langle \theta_k, (\xi_x \partial_\xi + \eta_x \partial_\eta) \theta_l \rangle$ and $\langle \theta_k, (\xi_y \partial_\xi + \eta_y \partial_\eta) \theta_l \rangle$, cannot be explicitly computed before finding the map itself. On the contrary, the components which do not involve the map, namely $([\theta_k, \theta_l]_1 - \langle \partial_\tau \theta_k, \theta_l \rangle)$ and $\langle \theta_k, \theta_l \rangle$, can be pre-computed once for all before solving problem (3.12). Equation (3.12) is nonlinear due to the convective-diffusive terms $\boldsymbol{\mathcal{F}}^*$ and $\boldsymbol{\mathcal{F}}^{**}$ which depend on the solution q_h. For this reason a fixed point problem is solved: let r be the index of the fixed point iteration, therefore we solve q_h^{r+1}

$$(3.13) \quad \begin{aligned} &([\theta_k, \theta_l]_1 - \langle \partial_\tau \theta_k, \theta_l \rangle) \hat{\boldsymbol{q}}_l^{r+1} + \Delta t \langle \theta_k, \theta_l \rangle \hat{\boldsymbol{\mathcal{F}}}_l^{*,r} + \Delta t \langle \theta_k, (\xi_x \partial_\xi + \eta_x \partial_\eta) \theta_l \rangle \boldsymbol{\mathcal{F}}_{\xi,l}^{**,r} \\ &\quad + \Delta t \langle \theta_k, (\xi_y \partial_\xi + \eta_y \partial_\eta) \theta_l \rangle \boldsymbol{\mathcal{F}}_{\eta,l}^{**,r} = \Delta t \langle \theta_k, \boldsymbol{f} \rangle + [\theta_k, \boldsymbol{w}^n]_0, \end{aligned}$$

where terms of fixed point index r are computed by using the previous solution $\boldsymbol{q}_h^r$. In our numerical tests, the fixed point iteration stops when the $L^2(\hat{\mathcal{C}})$-norm of residual of equation (3.13) is less than a fixed tolerance.

3.3. Recovery of the map and foreground mesh motion. In the previous subsection, the local map $\mathcal{M}_i : \hat{\mathcal{C}} \to \mathcal{C}_i^n$ has been involved for the computation of the local weak predictor solution. Moreover, the foreground mesh of coordinates $\boldsymbol{X}$ is moving accordingly to the following motion equation:

$$(3.14) \quad \frac{\mathrm{d}\boldsymbol{X}}{\mathrm{d}t} = \boldsymbol{V},$$

where $\boldsymbol{V} = \boldsymbol{V}(\boldsymbol{x}, t; u)$ is the mesh velocity, eventually dependent on the solution. Equation (3.14) is closed with a Cauchy condition $\boldsymbol{X}(0) = \boldsymbol{X}_0$, which is the initial spatial configuration. Through equation (3.14), we recover the map $\mathcal{M}_i$ for any cell at least on the foreground mesh. The motion equation (3.14) is solved through an *isoparametric* or *Lagrangian* approach by locally referring it to the same reference system as done for the local equation (3.5). This means that the spatial coordinates $\boldsymbol{X}$ are considered as function of the reference coordinates, i.e. $\boldsymbol{X}(\boldsymbol{\xi})$, with $\boldsymbol{\xi} \in \hat{\mathcal{C}}$. Finally, the solution of the referred motion equation is approximated via a Lagrangian expansion by employing the same Lagrangian basis $\{\theta_k\}_{k=1}^{27}$ built on the tensor combination of three Gauss-Legendre nodes in $(0,1)$ along any direction as previously introduced: $\boldsymbol{X}_h = \theta_l \hat{\boldsymbol{X}}_l$, with $\hat{\boldsymbol{X}}_l = \boldsymbol{X}(\hat{\boldsymbol{\xi}}_l)$. Thus, from time t^n to t^{n+1}, the motion equation (3.14) is locally re-written as

$$\frac{\mathrm{d}\boldsymbol{X}}{\mathrm{d}t} = \boldsymbol{V} \text{ in } \mathcal{C}_i^n, \tag{3.15}$$

and closed by strongly imposing that the solution $\boldsymbol{X}^n$ at current time is equal to $\boldsymbol{X}(t^n)$ found at the previous physical space-time cell $\mathcal{C}_i^{n-1}$. The local motion equation (3.15) is weaken in a similar way to the local equation (3.5) and in algebraic form it reads

$$([\theta_k, \theta_l]_1 - \langle \partial_\tau \theta_k, \theta_l \rangle) \hat{\boldsymbol{X}}_l = \Delta t \langle \theta_k, \theta_l \rangle \hat{\boldsymbol{V}}_l + [\theta_k, \theta_l]_0 \hat{\boldsymbol{X}}_l^n, \tag{3.16}$$

with $\hat{\boldsymbol{V}}_l = \boldsymbol{V}|_{\hat{\boldsymbol{\xi}}_l}$. The last term $[\theta_k, \theta_l]_0 \hat{\boldsymbol{X}}_l^n$ takes into account the initial given configuration of the space at time t^n.

When the mesh is neither moving nor deforming, as for cells in the background, the mesh velocity is thus coincident with zero, i.e. $\boldsymbol{V} \equiv \boldsymbol{0}$. In that case, the map is known *a priori* and it consists in the rescaling of the reference space-time cell $\hat{\mathcal{C}}$ to the physical space-time cell $\mathcal{C}_i^n$:

$$\begin{cases} x = x(\xi) = x_{i-1/2} + h_i^x \xi \\ y = y(\eta) = y_{i-1/2} + h_i^y \eta \end{cases}, \tag{3.17}$$

where coordinates $x_{i-1/2}$ and $y_{i-1/2}$ and $x_{i+1/2}$ and $y_{i+1/2}$ define the extremes along x- and y-direction of the physical space-time cell $\mathcal{C}_i^n \equiv [x_{i-1/2}, x_{i+1/2}] \times [y_{i-1/2}, y_{i+1/2}] \times [t^n, t^{n+1}]$; and h_i^x and h_i^y are the length along x and y of the cell, respectively, i.e. $h_i^x = x_{i+1/2} - x_{i-1/2}$ and $h_i^y = y_{i+1/2} - y_{i-1/2}$.

Since the mesh motion equation (3.14) is essentially solved via a sort of Discontinuous Galerkin (DG) approach, possible numerical (and non physical) discontinuities could arise. As a matter of fact, for a given vertex $\bar{\boldsymbol{X}}_k^{n+1}$ shared by a set of spatial cells $\{\Omega_i^{n+1}\}_{i \in \mathcal{Z}_k^{n+1}}$ at time t^{n+1}, there could be as many different values of the vertex, namely $\{\bar{\boldsymbol{X}}_{k,i}^{n+1}\}_{i \in \mathcal{Z}_k^{n+1}}$, for any map $\mathcal{M}_i$ referring to the cell $\mathcal{C}_i^n$ to which Ω_i^{n+1} belongs. The set $\mathcal{Z}_k^{n+1}$ collects the index(es) of the cells sharing the vertex $\bar{\boldsymbol{X}}_k^{n+1}$. The cardinality N_k of set $\{\Omega_i^{n+1}\}_{i \in \mathcal{Z}_k^{n+1}}$, coinciding with the cardinality of the indexes set $\mathcal{Z}_k^{n+1}$, depends on the position of the vertex $\bar{\boldsymbol{X}}_k^{n+1}$ on the foreground mesh: it is either 1 or 2 if the vertex is on the boundary of the mesh, otherwise it is 4. For this reason we consider a weighted average value for the shared vertex in order to tackle the possible arising discontinuities. As suggested in [6], we first consider a weighted velocity $\bar{\boldsymbol{V}}_k^{n+1}$ corresponding to the vertex $\bar{\boldsymbol{X}}_k^{n+1}$

$$\bar{\boldsymbol{V}}_k^{n+1} = \frac{1}{N_k} \sum_{i \in \mathcal{Z}_k^{n+1}} \bar{\boldsymbol{V}}_{k,i}^{n+1}, \text{ with } \bar{\boldsymbol{V}}_{k,i}^{n+1} = \int_0^1 \theta_l(\xi^*, \eta^*, \tau) \, \mathrm{d}\tau \hat{\boldsymbol{V}}_{l,i}, \tag{3.18}$$

where coordinates (ξ^*, η^*) depend on the position of the coordinate $\bar{\boldsymbol{X}}_k^{n+1}$ in the cell Ω_i^{n+1}; it can assume four values: $(0,0)$, $(1,0)$, $(1,1)$ and $(0,1)$. Once equation (3.16) is solved, the just found coordinates $\{\hat{\boldsymbol{X}}_l\}_{l=1}^{27}$ are used for computing the velocity components $\hat{\boldsymbol{V}}_{l,i}$ and, thus, the weighted velocities $\bar{\boldsymbol{V}}_k^{n+1}$ in (3.18). Finally, the coordinates $\bar{\boldsymbol{X}}_k^{n+1}$ at time t^{n+1} is

$$\bar{\boldsymbol{X}}_k^{n+1} = \bar{\boldsymbol{X}}_k^n + \Delta t \, \bar{\boldsymbol{V}}_k^{n+1}. \tag{3.19}$$

We refer the reader to [11] for another definition of the weighted vertex velocities $\bar{\boldsymbol{V}}_k^{n+1}$ in (3.18) where the Voronoi neighborhood parameters of any vertex are exploited.

In Algorithm 3.1 we resume the salient stages of the prediction step.

Algorithm 3.1 Prediction step

1: Compute the foreground mesh motion (3.19) from the motion equation (3.14) and through the weighted velocity (3.18);
2: **for** $i = 1, \dots, N$ **do**
3: Find the map $\mathcal{M}_i$ for the space-time cell $\mathcal{C}_i^n$;
4: Compute the Jacobian matrix J associated to $\mathcal{M}_i$;
5: Compute J^{-1} and take the submatrix J_s^{-1} to the spatial coordinates defined in (3.9);
6: Update the convective-diffusive terms $\mathcal{F}^*$ and $\boldsymbol{\mathcal{F}}^{**}$ in the reference domain;
7: Evolve the local predictor solution through (3.12);

3.4. Correction stage: the finite volume scheme over the space-time cell. Once the local predictor solution $q\boldsymbol{q}_h$ is computed in each space-time cells $\mathcal{C}_i^n$, we can perform the correction stage. First, we rewrite the convective-diffusive equation (1.1) in divergence form. Let $\nabla_{\boldsymbol{x},t} = [\nabla, \partial_t]^T$ be the space-time differential operator and let $\boldsymbol{U} = [\boldsymbol{F}(\boldsymbol{u}, \nabla \boldsymbol{u}u), \boldsymbol{u}]^T$ be the space-time solution, thus problem (1.1) can be rewritten as

$$\nabla_{\boldsymbol{x},t} \cdot \boldsymbol{U} = \boldsymbol{f} \quad \text{in } \Omega(t) \times [0, T]. \tag{3.20}$$

We want to find a finite volume solution for the above equation, where the finite volume is the space-time cell $\mathcal{C}_i^n$, whose boundary reads

$$\partial \mathcal{C}_i^n = \Omega_i^n \cup \Omega_i^{n+1} \cup \bigcup_{j=1}^{4} \Gamma_{ij}^n, \tag{3.21}$$

where the boundaries Γ_{ij}^n, $j = 1, \dots, 4$, are the space-time boundaries of $\mathcal{C}_i^n$ linking any edge of Ω_i^n at time t^n to any edge of Ω_i^{n+1} at time t^{n+1}. By integrating equation (3.20) over $\mathcal{C}_i^n$ and by applying the divergence theorem to the left side, we obtain

$$\int_{\partial \mathcal{C}_i^n} \boldsymbol{U} \cdot \boldsymbol{n}_{\boldsymbol{x},t} \, \mathrm{d}\Gamma = \int_{\mathcal{C}_i^n} \boldsymbol{f} \, \mathrm{d}\mathcal{C}, \tag{3.22}$$

with $\boldsymbol{n}_{\boldsymbol{x},t}$ being the normal unit vector to the boundary $\partial \mathcal{C}_i^n$ of the cell. Let $\boldsymbol{U}_i^n$ be the spatial average of the solution u of (1.1) over the spatial cell Ω_i^n and located on its center, i.e.,

$$\boldsymbol{U}_i^n = \frac{1}{|\Omega_i^n|} \int_{\Omega_i^n} \boldsymbol{u}(x, y, t^n) \, \mathrm{d}x \, \mathrm{d}y, \tag{3.23}$$

where $|\Omega_i^n|$ is the measure of the spatial cell Ω_i^n. Though (3.21) and (3.23), equation (3.22) explicitly is

$$- |\Omega_i^n| \boldsymbol{U}_i^n + |\Omega_i^{n+1}| \boldsymbol{U}_i^{n+1} + \sum_{j=1}^{4} \int_{\Gamma_{ij}^n} \boldsymbol{U} \cdot \boldsymbol{n}_{\boldsymbol{x},t} \, \mathrm{d}\Gamma = \int_{\mathcal{C}_i^n} \boldsymbol{f} \, \mathrm{d}\mathcal{C}, \tag{3.24}$$

where the unknown is the average solution $\boldsymbol{U}_i^{n+1}$ at time t^{n+1}, while the last term of the left hand side is the space-time flux along the space-time sides $\bigcup_{j=1}^{4} \Gamma_{ij}^n$. Scheme (3.24) is the Finite Volume scheme; we remark that it is still exact. In order to solve (3.24), we need to approximate the integral function of the space-time flux. Among the several methods proposed in the literature (such as in [11, 12, 13, 33, 17]), we here present a Local Lax-Friederichs (LLF) approach:

$$[\boldsymbol{U} \cdot \boldsymbol{n}_{\boldsymbol{x},t}]_{\Gamma_{ij}^n} \approx \Phi(\boldsymbol{q}_j^+, \boldsymbol{q}_j^-) = \frac{1}{2}(\boldsymbol{U}_j^+ + \boldsymbol{U}_j^-) \cdot \boldsymbol{n}_{\boldsymbol{x},t} - \frac{s}{2}(\boldsymbol{q}_j^+ - \boldsymbol{q}_j^-), \tag{3.25}$$

where $\boldsymbol{U}_j^+ = \boldsymbol{U}(\boldsymbol{q}_j^+)$ and $\boldsymbol{U}_j^- = \boldsymbol{U}(\boldsymbol{q}_j^-)$ are the space-time solution of (3.20) computed by solutions $\boldsymbol{q}_j^+$ and $\boldsymbol{q}_j^-$, which represent the local predictor solutions outside and inside the cell, respectively, with respect to the space-time side Γ_{ij}^n. The term s is the stabilization coefficient. Equation (3.24) with the flux approximation

(3.25) closes the correction stage of the ADER method. At the end of this stage, a solution u_i^{n+1} is found over any cell Ω_i^{n+1}. Since the predictor solution over space-time cells $\mathcal{C}_i^{n+1}$ needs to be evaluated over the Gauss nodes, a second order local polynomial interpolation is performed as explained in Section 3.1.
For the computation of the integrals along the space-time manifolds Γ_{ij}^n, we still use the previously computed map $\mathcal{M}_i$. As a matter of fact, for a generic function $g : \mathcal{C}_i^n \to \mathbb{R}$ it holds:

$$\int_{\Gamma_{ij}^n} g(\boldsymbol{x})\ \mathrm{d}\Gamma = \int_{\hat{\Gamma}_j} g(\boldsymbol{x}(\boldsymbol{\xi}))|\mathrm{Cof}(J)\hat{\boldsymbol{n}}_j|\ \mathrm{d}\hat{\Gamma},$$

where $\hat{\Gamma}_j$ is the j-th lateral side of the reference cubic domain $\hat{\mathcal{C}}$ of unit outer normal $\hat{\boldsymbol{n}}_j$, $\Gamma_{ij}^n = \mathcal{M}_i(\hat{\Gamma}_j)$ and $\mathrm{Cof}(J)$ is the cofactor matrix of the Jacobian tensor J of the map.

Concerning the time step Δt, due to the combination of the weak predictor solution by problem (3.11) and the consequent plug of this solution in the finite volume scheme (3.24) trough the LLF flux (3.25), a classical stability analysis is not evident. We assumed the time step to be

$$\Delta t = \mathrm{CFL}\frac{h}{\max\{\sup_{\Omega\times[0,T]}|a_x|, \sup_{\Omega\times[0,T]}|a_y|\}}, \tag{3.26}$$

where h is the smallest characteristic length among all cells (both of background and foreground meshes) along the whole temporal window $[0,T]$, i.e., $h = \min_{i,n} h_i^n$, with h_i^n the characteristic length of spatial cell Ω_i^n at discrete time t^n. Coefficient CFL in (3.26) is the Courant-Friedrichs-Lewy number. In this paper, the CFL coefficient is experimentally sought by conducting an empirical analysis in Section 5.2.

3.5. Dynamics of the overlapping zone. During the simulation, the foreground mesh moves and, consequently, the background mesh changes its configuration in the zone of the overlapping as well as in the hole. Let $\Omega_i(t)$ be a background cell in a neighborhood of the overlapping. From times t^n to t^{n+1}, there are three possibilities:

1. Cell $\Omega_i(t)$ is present at time t^n and it disappears at time t^{n+1} because the hole completely covers it;
2. Cell $\Omega_i(t)$ is not present at time t^n but it appears at time t^{n+1} because the hole gets away;
3. The overlapping zone does not drastically change its configuration with respect to cell $\Omega_i(t)$, thus the cell is present at time t^n and it still continues to be present at time t^{n+1}.

The third case is trivial. For the first case, the predictor solution is executed in order to compute the fluxes of the neighboring cells even though the correction stage is not performed. For the second case, information u_i^n is missing and it is necessary for computing u_i^{n+1}. For this reason, let N_1 the total number of background cells (those ones in the hole included). Consequently $i \le N_1$. By recalling that the order of foreground cells starts from $N_1 + 1$, we look for an index $j > N_1$ such that

$$\boldsymbol{x}_j = \arg\min_{k>N_1} \|\boldsymbol{x}_i - \boldsymbol{x}_k\|, \tag{3.27}$$

where $\boldsymbol{x}_\mu$ is the center of mass of cell Ω_μ^n, for $\mu = i, j, k$. Then, a local polynomial interpolation $\boldsymbol{w}_j^n$ on the stencil $\mathcal{S}_j$ centered on cell Ω_j^n of the foreground mesh is computed as previously explained in Section 3.1. In particular, since Ω_j^n is chosen to be as the closest foreground cell to background cell Ω_i^n through (3.27), a third order polynomial approximation of solution $\boldsymbol{u}^n$ on $\boldsymbol{x}_i$ is ensured by imposing $\boldsymbol{u}_i^n = \boldsymbol{w}_j^n(\boldsymbol{x}_i)$. Finally the ADER prediction-correction is performed as usual.

4. The stabilization of the scheme. For the definition of the coefficient s in (3.25), there are different approaches leading to different definitions. Here we analyse two stabilization coefficients, i.e. the advective-diffusive term s_{AD} and the just advective term s_A. For the sake of clarity and to lighten the notation, we consider a two-dimensional scalar solution in this section (i.e., $d = 2$ and $\delta = 1$).

4.1. The local advective-diffusive stabilization term. For the definition of the coefficient s_{AD} in (3.25), we study a relaxed hyperbolic form of the parabolic equation (3.20). Let us consider the following relaxation by Cattaneo (we refer to [34] and its references for further details): let $0 < \varepsilon \ll 1$ be a relaxed time and consider variables v and w in $\Omega \times [0,T]$ such that

$$\partial_t v = \frac{1}{\varepsilon}(\partial_x u - v); \quad \partial_t w = \frac{1}{\varepsilon}(\partial_y u - w). \tag{4.1}$$

Relations (4.1) define the relaxations in the sense that $\partial_x u \to v$ and $\partial_y u \to w$ in the limit of a vanishing ε. Since the flux has to be computed along the manifold Γ_{ij}^n in the space-time continuum, let us consider solution u and all its first derivatives as *stationary solutions* with respect to a pseudo-time $\mathfrak{t} \in \mathbb{R}_+$. Thus, let $\mathfrak{u}(\mathfrak{t}; x, y, t) = [u, v, w]^T$ be the formal definition of the relaxed hyperbolic system with respect to pseudo-time $\mathfrak{t}$. It holds $\partial_{\mathfrak{t}}\mathfrak{u} = 0$. The conservative form problem (3.20) in quasi-linear form is

$$\partial_{\mathfrak{t}}\mathfrak{u} + \partial_x(A\mathfrak{u}) + \partial_y(B\mathfrak{u}) + \partial_t(C\mathfrak{u}) = \mathfrak{f} \quad \text{in } \mathbb{R}_+ \times \Omega(t) \times [0, T], \tag{4.2}$$

where A, B and C are 3×3 matrices (eventually involving the solution $\mathfrak{u}$ among their components if the original problem is nonlinear) and the force term $\mathfrak{f} = [f, -v/\varepsilon, -w/\varepsilon]^T$. In particular, A and B always depend on the relaxation time ε and they are defined by the convection-diffusion term $\boldsymbol{\mathcal{F}}(u, \nabla u)$ and C is always the identity matrix if the Cattaneo's relaxation (4.1) is employed. In order to study the differential operator in (4.2), let us consider a vanishing force term, i.e. $\mathfrak{f} \equiv \mathbf{0}$. The presence of the the pseudo-time $\mathfrak{t}$ in (4.2) helps in treating the real time variable t as any other spatial variable x and y. When the force term in (4.2) is null, the problem is hyperbolic if the spectrum of matrix $\mathcal{A} = n_x A + n_y B + n_t C$ is real for any choice of real values n_x, n_y and n_t. If the hyperbolicity is ensured, the relaxed hyperbolic system has a planar wave solution propagating in the space-time continuum $\Omega \times [0, T]$. In particular, if $\boldsymbol{n}_{\boldsymbol{x},t} = [n_x, n_y, n_t]^T$ is a particular direction in the space-time continuum, the eigenvalues of $\mathcal{A}$ define the speeds of propagation of the solution along the principal directions defined by the eigenvectors of $\mathcal{A}$. For this reason, in the perspective of an upwind stabilization, the local stabilization term s_{AD} in (3.25) is equal to the maximum speed of propagation of the wave, as it happens for the LLF flux approximation for a generic hyperbolic problem of a propagating wave in the space continuum.

Here we detail the previous analysis for the convection-diffusion problem with the convective field $\boldsymbol{a} = [a_x, a_y]^T$ and the diffusive term ν depending on space $\boldsymbol{x}$ and time t and eventually the solution u itself if a non-linearity leads the dynamics of the equation. In this case, the matrices of the quasi-linear problem (4.2) read

$$A = \begin{bmatrix} a_x & -\nu & 0 \\ -1/\varepsilon & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} a_y & 0 & -\nu \\ 0 & 0 & 0 \\ -1/\varepsilon & 0 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Consequently, the spectrum $\rho(\mathcal{A})$ of matrix $\mathcal{A}$ is

$$\rho(\mathcal{A}) = \left\{ n_t; \frac{1}{2}\left[\sigma \pm \sqrt{\left(a_x^2 + \frac{4\nu}{\varepsilon}\right) n_x^2 + 2a_x a_y n_x n_y + \left(a_y^2 + \frac{4\nu}{\varepsilon}\right) n_y^2} \right] \right\}, \tag{4.3}$$

where $\sigma = \boldsymbol{a} \cdot \boldsymbol{n} + 2n_t$ and $\boldsymbol{n} = [n_x, n_y]^T$. The following proposition finally defines the advective-diffusive stabilization parameter.

PROPOSITION 4.1. *For the advection-diffusion problem* (1.1) *with the convective field* $\boldsymbol{a} = [a_x, a_y]^T$ *and the diffusive term* ν*, the advection-diffusion stabilization coefficient* s_{AD} *is chosen to be the absolute value of the maximum of spectrum* (4.3)*, i.e.,*

$$s_{AD} = \max |\rho(\mathcal{A})| = \frac{1}{2}\left| \sigma + \sqrt{\left(a_x^2 + \frac{4\nu}{\varepsilon}\right) n_x^2 + 2a_x a_y n_x n_y + \left(a_y^2 + \frac{4\nu}{\varepsilon}\right) n_y^2} \right|. \tag{4.4}$$

Since the spectrum $\rho(\mathcal{A}) \subset \mathbb{R}$ for any nonnegative ε, it yields the relaxed system (4.2) is always hyperbolic for any nonnegative ε.

4.2. The choice of the relaxation time. For the definition of the advective-diffusive stabilization term s_{AD}, we considered the relaxed hyperbolic system (4.2) deriving from the parabolic problem (3.20) through a relaxation time ε. If we were to solve the relaxed problem instead of the original one, the approximate solution would differ from the exact solution of two errors that are added together: the numerical error (typical of the scheme) and a relaxation error. For a linear problem, these errors have been investigated by Montecinos and Toro in [34]. The error $|u_{\text{hip}} - u|$ between the hyperbolized solution u_{hip} and the original solution u is $\mathcal{O}(\varepsilon)$ [24]. Thus, if $u_{\text{hip},h}$ is a numerical approximation of the exact relaxation solution u_{hip}, the error $|u_{\text{hip},h} - u_{\text{hip}}|$ is $\mathcal{O}(h_0^p)$, with p the order of the method (i.e., $p = 2$ in this paper), and h_0 the maximum characteristic length of cells $\Omega_i(t)$'s. However, the goal is to choose a relaxation time ε such that the relaxation error is always dominated by or, at least, comparable to the numerical error, i.e. $\mathcal{O}(\varepsilon) \lessapprox \mathcal{O}(h_0^p)$. The following theoretical result can help in fulfilling our task.

PROPOSITION 4.2. *The solution u of the original parabolic problem (3.20) is approximated by a relaxed solution u_{hip} solving the relaxed problem (4.2) with accuracy p for all relaxation time ε and characteristic length cell h_0 satisfying*

$$C_p \frac{\varepsilon}{h_0^p} = \mathcal{O}(1), \tag{4.5}$$

with

$$C_p = \frac{1 - 2^{-\frac{1}{2}}}{2^{p-\frac{1}{2}} - 1}.$$

For the proof of Proposition 4.2, we refer the reader to Section 2.4.1 of [23]. As a consequence, there is the following corollary.

PROPOSITION 4.3. *For a given mesh whose characteristic length is h_0 and a numerical method of order p for solving the hyperbolized problem (4.2) derived by the original parabolic problem (3.20), the optimal relaxation time ε_p is*

$$\varepsilon_p = \frac{\mathcal{O}(1) h_0^p}{C_p}. \tag{4.6}$$

We remark that, if a relaxation time ε is chosen to be less than or equal to ε_p, the numerical error dominates the relaxation error; on the contrary, if a relaxation time ε is chosen to be greater than the optimal value, the relaxation error dominates the numerical error. For this reason, in our simulation relaxation time $\varepsilon = \varepsilon_2/2$ is chosen.

4.3. The local advective stabilization term. In order to recover a stabilization term s_A by only considering the first order operator involved in the whole differential operator of the original problem, we can treat the equation to stabilize as a pure hyperbolic (namely just advective) problem. For this reason, the advective stabilization term s_A coincides with the maximum eigenvalue of the ALE Jacobian matrix in a spatial normal direction by excluding the diffusive component which acts on the diffusion from the advective-diffusive term $\boldsymbol{F}(u, \nabla u)$ [11]. This matrix reads

$$A_{\tilde{\boldsymbol{n}}}^{\boldsymbol{V}} = \sqrt{n_x^2 + n_y^2} \left[\frac{\partial \boldsymbol{F}}{\partial u} \, \tilde{\boldsymbol{n}} - \boldsymbol{V} \cdot \tilde{\boldsymbol{n}} \, I \right], \tag{4.7}$$

where I is the identity tensor whose dimension is that one of the image space of the solution u and the unit vector $\tilde{\boldsymbol{n}}$ is the normalized projection of the space-time unit vector $\boldsymbol{n}_{\boldsymbol{x},t}$ along the spatial directions given by vector $[n_x, n_y]^T$, i.e.

$$\tilde{\boldsymbol{n}} = \frac{[n_x, n_y]^T}{\sqrt{n_x^2 + n_y^2}}.$$

By recalling that the recovered map $\mathcal{M}_i$ is defined over $\overline{\hat{\mathcal{C}}}$ with image in $\overline{\mathcal{C}}_i$, the space-time manifold $\Gamma_{ij}^{\boldsymbol{n}}$, $j = 1, \ldots, 4$, of the space-time cell $\mathcal{C}_i$ can be described by only two of the three reference space-time variables (ξ, η, τ); i.e., by either couple (ξ, τ), with $\eta = \bar{\eta}$, or couple (η, τ), with $\xi = \bar{\xi}$; with $\bar{\xi}$ and $\bar{\eta}$ alternatively equal to 0 or 1, depending on the specific j-th space-time manifold $\Gamma_{ij}^{\boldsymbol{n}}$. Let χ be the free variable (e.g. $\chi = \xi$) and $\bar{\kappa}$ be the constrained variable (e.g. $\bar{\kappa} = \bar{\eta}$) for the specific manifold $\Gamma_{ij}^{\boldsymbol{n}}$. Therefore, for a specific point $\tilde{\boldsymbol{x}}$ over $\Gamma_{ij}^{\boldsymbol{n}}$ it is possible to distinguish two directional vectors provided by the map $\mathcal{M}_i$

$$\boldsymbol{r}_\chi = \begin{bmatrix} x_\chi \\ y_\chi \\ 0 \end{bmatrix}_{\bar{\kappa}} \quad \text{and} \quad \boldsymbol{r}_\tau = \begin{bmatrix} x_\tau \\ y_\tau \\ \Delta t \end{bmatrix}_{\bar{\kappa}}.$$

The definitions of the directional vectors $\boldsymbol{r}_\chi$ and $\boldsymbol{r}_\tau$ allow to explicitly write the physical normal vector $\boldsymbol{n}_{\boldsymbol{x},t}$ on $\tilde{\boldsymbol{x}}$ as

$$\boldsymbol{n}_{\boldsymbol{x},t} = \left.\frac{\boldsymbol{r}_\chi \wedge \boldsymbol{r}_\tau}{|\boldsymbol{r}_\chi \wedge \boldsymbol{r}_\tau|}\right|_{\bar{\kappa}} = \left.\frac{[\Delta t\, y_\chi,\, -\Delta t\, x_\chi,\, d_{\chi\tau}]^T}{\sqrt{\Delta t^2\, y_\chi^2 + \Delta t^2\, x_\chi^2 + d_{\chi\tau}^2}}\right|_{\bar{\kappa}},$$

with $d_{\chi\tau} = x_\chi y_\tau - x_\tau y_\chi$. From now on we will omit the constraint variable $\bar{\kappa}$. It is now possible to write the unit vector $\tilde{\boldsymbol{n}}$ along the spatial directions and the velocity of the point as

$$\tilde{\boldsymbol{n}} = \frac{[y_\chi, -x_\chi]^T}{\sqrt{y_\chi^2 + x_\chi^2}} \quad \text{and} \quad \boldsymbol{V} = \frac{\mathrm{d}\tilde{\boldsymbol{x}}}{\mathrm{d}t} = \frac{[x_\tau, y_\tau]^T}{\Delta t}.$$

Consequently it holds

$$\boldsymbol{V} \cdot \tilde{\boldsymbol{n}} = \frac{-d_{\chi\tau}}{\Delta t\sqrt{y_\chi^2 + x_\chi^2}} = \frac{-n_t}{\sqrt{n_x^2 + n_y^2}}. \tag{4.8}$$

In the case of a linear problem the advective stabilization term reads

$$s_A = |a_x n_x + a_y n_y + n_t|. \tag{4.9}$$

The next proposition, through 4.8, allows to connect the advective-diffusive parameter s_{AD} with the advective parameter s_A in the limit of a vanishing diffusion parameter ν.

PROPOSITION 4.4. *For linear problem* (1.1)*, let the diffusion parameter ν go to zero, therefore the following limit holds*

$$\lim_{\nu\to 0} s_{AD} = \frac{1}{2}|\sigma + a_x n_x + a_y n_y| = |a_x n_x + a_y n_y + n_t| = s_A. \tag{4.10}$$

The above Proposition confirms that, in the limit of small diffusion in the dynamics of linear problem (1.1), the two stabilization techniques coincide.

5. Numerical results. In this section we are going to present some numerical test cases in order to analyse the method.

Table 1 synthetically sums up the test cases that will be used for the different analyses. In particular, *test1* and *test2* (in lowercase letters) are the 1D tests and *TEST1* and *TEST2* (in capital letters) are the 2D test cases.

In the 1D tests, the foreground mesh is put in the middle between other two meshes composing the background mesh, and it deforms according to the deformation laws specified in the last row of Table 1. In the following, for *test1* we are not presenting a figure but only the rate of convergence. In Figure 5 three instants for *test2* simulation are showed; in particular, the red circle markers define the nodes of the moving foreground mesh which is in the middle between the other two meshes (in the background) whose nodes are marked by blue dots and x-symbols. The background meshes are always uniform while the foreground mesh is allowed to be displaced and deformed. The solution of *test2* is flat towards the boundaries of the computational domain and develops a moving front affected by a large spatial derivative; for this reason, the foreground mesh is set in order to follow the front. Finally we remark that, if h is the characteristic length of the cells in the background mesh, at the initial time $t = 0$ the foreground mesh is uniform with a characteristic length equal to $h/2$ in *test1* and $h/4$ for *test2*.

In *TEST1*, the foreground mesh is subjected to a deformation and rotation around its center of mass. We remark that in this case that the deformation velocity depends on the solution; in *TEST2*, the hyperbolic tangent in the exact solution describes a composed Gaussian bell whose maximum is originally located in the position $\boldsymbol{x} = (-1, 0)$ and, after a time $T = \pi$, it computes a counterclockwise half rotation up to position $\boldsymbol{x} = (1, 0)$ along the circumference of unit radius and centered in the origin of the axes. Due to the particular dynamics of the solution, we set a foreground mesh following the movement of the Gaussian bell. At the initial time, the foreground and background meshes in both 2D cases consist of squared cells whose sides have a length equal to h.

For all numerical tests, the time step Δt is set accordingly to (3.26) with CFL coefficient equal to 0.4. The reason of this value will be better explained in Section 5.2 where an empirical stability analysis is conducted. Without reporting numerical evidences, we checked the scheme is free-stream preserving, i.e. it exactly solves a constant but nonzero solution.

5.1. Order of convergence. In this section we have a double goal. On one hand we want to numerically prove that the presented method is second order when an advective-diffusive LLF stabilization s_{AD} is employed. On the other hand, we want to compare this stabilization term with the local advective stabilization flux s_A. The study of the second order convergence is conducted on all test cases of Table 1. Finally, on the two mentioned 2D test cases the comparison of the performances for the flux approximations is carried out.

For quantifying the convergence rate, we considered the L^∞- and L^2-norms of the mismatch between the exact solution and the numerical solution at final time $t = T$. The errors are defined and approximated as

$$L^\infty\text{-err} = \|u - u_{ex}\|_{L^\infty(\Omega)} = \operatorname{ess\,sup}_{\boldsymbol{x}\in\Omega} |u(\boldsymbol{x}, T) - u_{ex}(\boldsymbol{x}, T)| \approx e^N_{L^\infty} = \max_{k=1,\dots,N} |u^M_k - u_{ex}(\boldsymbol{x}_k, T)| \tag{5.1}$$

and

$$L^2\text{-err} = \|u - u_{ex}\|_{L^2(\Omega)} = \sqrt{\int_\Omega \Big(u(\boldsymbol{x}, T) - u_{ex}(\boldsymbol{x}, T)\Big)^2 \mathrm{d}\Omega} \approx e^N_{L^2} = \sqrt{\frac{|\Omega| \sum_{k=1}^N \Big(u^M_k - u_{ex}(\boldsymbol{x}_k, T)\Big)^2}{N}}, \tag{5.2}$$

respectively, where $N \approx |\Omega| h^{-1/d}$ is the number of cells such that any part of the of the domain is covered by one and only one cell at time T (with h the characteristic length of cells and $d = \dim(\Omega)$) and M is the maximum natural such that $T = M\Delta t$. Approximation (5.2) is valid only in the case of cells having approximatively or exactly the same spacing. The convergence rate reads

$$L^p\text{-rate} = d\frac{\log\big(e^{N_1}_{L^p}/e^{N_2}_{L^p}\big)}{\log(N_2/N_1)}, \quad \text{for } p = 2, \infty, \tag{5.3}$$

for two different partition settings whose number of cells are N_1 and N_2, respectively, with $N_1 < N_2$. The mesh refinement is performed by reducing the spacing (kept constant for any cell) and by preserving a layer of 4 cells both in background and foreground for the overlapping zone.

Table 2 sums up the convergence analysis for 1D test cases. In the last two columns there are the rates of convergence of the errors for both L^∞ and L^2 errors. From the analysis, the second order of the method is confirmed.

In Table 3 we report the L^∞- and L^2-errors with their respective rate of convergence with respect to a local advective-diffusive (AD, white cells) and advective (A, grey cells) stabilization. We first remark that, for both cases, the errors relative to AD stabilization are slightly smaller with respect to the same errors with an A stabilization. The rate of convergence of the errors for an AD stabilization is at least 2. On the other hand, even though a second order of accuracy is also reached by employing an A stabilization, the convergence rate shows an irregular trend (especially for *TEST2*). For this reason we can state that an AD flux approximation allows to reach a more precise solution with a monotone trend for the rate of convergence with respect to the same solution with an A flux stabilization.

5.2. Empirical analysis of stability condition. As already mentioned at the end of Section 3.4, the presence of a weak solution, found in the prediction step of the presented method and successively plugged into the flux of the finite volume scheme in the correction stage, makes a classical stability analysis not straightforward to be made. For this reason, we performed an empirical stability analysis by assuming that the right time step Δt allowing a stable computation is defined as in (3.26).

On a given problem, once both background and foreground meshes are set, we considered a time step Δt starting from a CFL number equal to 0.1 and, by increasing this value of 0.05 each time, we look for the largest stable CFL. In particular, this process is executed on the same problem considering an approximated LLF flux employing once an advective-diffusive stabilization term s_{AD} and then with an advective stabilization term s_A.

Table 1: Summary scheme of test cases used in Sections 5.1 and 5.2. Rows *Diffusion* and *Advection* report the diffusion parameter and the advection field value(s), respectively. Rows *B.C.* and *I.C.* describe the boundary conditions and the initial conditions, respectively. The subsets defined in *fg mesh* row are those ones covered at initial time by the foreground mesh subjected to a motion equation with velocity $\boldsymbol{V}$ (last row). Finally, function $\chi_E(z)$, with $z \in \mathbb{R}$ and $E \subset \mathbb{R}$, is the indicator function, i.e., $\chi_E(z) = 1$ if $z \in E$ and $\chi_E(z) = 0$ otherwise.

	1D		2D	
	test1	*test2*	*TEST1*	*TEST2*
Ω	$(-1,1)$	$(-1,1)$	$(-\pi,\pi)^2$	$(-\pi,\pi)^2$
Diffusion	0.5	1	6.37e-3	6.37e-3
Advection	3	1	$[0.6, 0.8]^T$	$[0.6, 0.8]^T$
u_{ex}	$e^{-t}\sin(\pi(x-t))$	$\cos(\pi(t-1/2))\tanh(10(x-t))$	$e^{-t}\sin(x)\cos(y)$	$-\tanh(2(x+\cos(t))^2+2(y-\sin(t))^2)+$ $-\cos(t)\cos(x)\sin(y)$
B.C.	periodic	Dirichlet: $u_{ex}(\pm 1, t)$	Dirichlet: $u_{ex}(x,y,t)\big\vert_{\partial\Omega}$	Dirichlet: $u_{ex}(x,y,t)\big\vert_{\partial\Omega}$
I.C.	$\sin(\pi x)$	0	$\sin(x)\cos(y)$	$-\tanh(2x^2+2(y-1)^2)$
T	0.25	0.5	1	π
fg mesh	$[-0.5, 0.25]$	$[-0.25, 0.25]$	$[-0.5, 0.5]^2$	$[-1.25, -0.75] \times [-0.25, 0.25]$
$\boldsymbol{V}$	$2.5 + e^x$	$e^x \chi_{\{t \le 0.25\}}(t) + \chi_{\{t>0.25\}}(t)$	$[u(x,y,t)-y, u(x,y,t)+x]^T$	$[-y, x]^T$

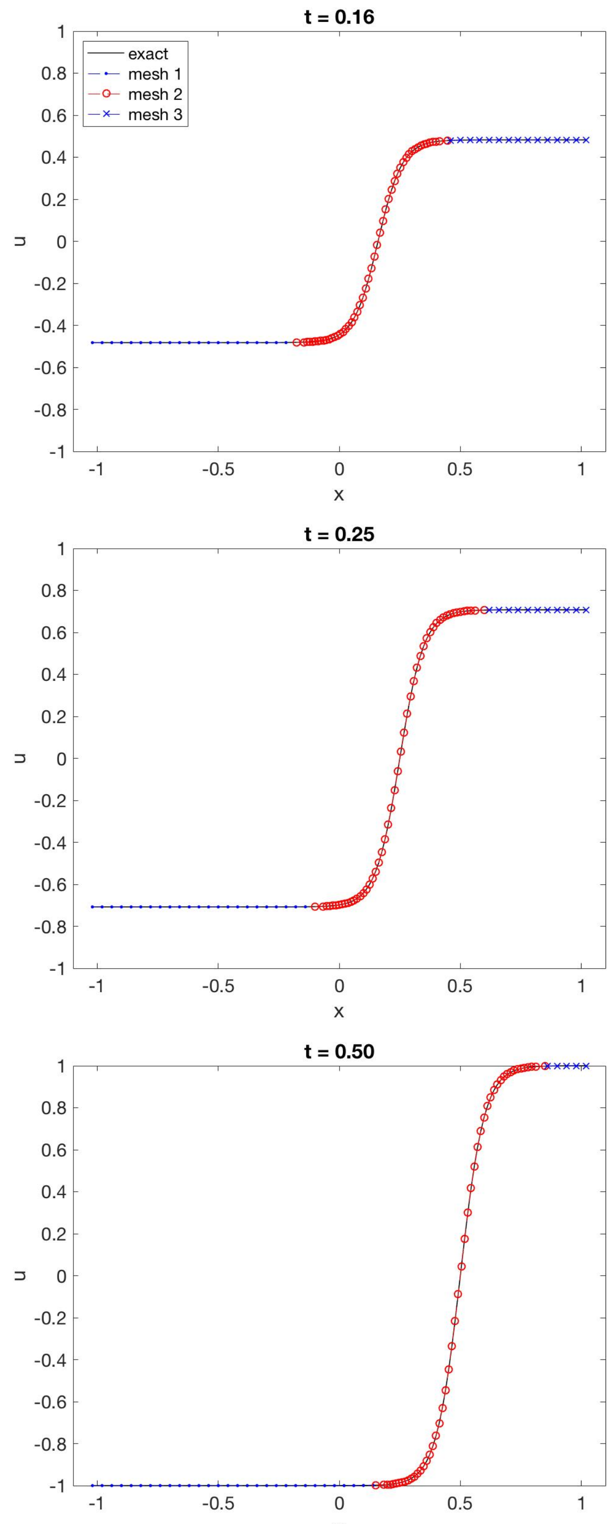


Fig. 5: Three time instants for the 1D test case *test2*. The circle markers define the nodes of the moving foreground mesh. The remaining dot and x markers are the nodes of the two background meshes.

Table 2: Convergence analysis for 1D test cases *test1* and *test2*.

	T	h	L^∞-err	L^2-err	L^∞-rate	L^2-rate
test1	0.25	2.00e-2	1.2740e-3	1.3903e-3	0	0
		1.00e-2	2.5042e-4	2.9250e-4	2.37	2.79
		5.00e-3	5.6957e-5	6.6934e-5	2.15	2.14
		2.50e-3	1.3675e-5	1.6068e-5	2.06	2.06
test2	0.5	1.00e-2	9.2733e-4	6.3960e-4	0	0
		5.00e-3	1.1948e-4	1.0081e-4	2.88	2.60
		2.50e-3	2.1898e-5	1.6359e-5	2.49	2.67
		1.25e-3	5.6504e-6	2.8547e-6	1.96	2.44

Table 3: Convergence analysis for 2D test cases *TEST1* and *TEST2*. Column labeled with h reports the smallest characteristic length among all cells.

	T	h	L^∞-err		L^2-err		L^∞-rate		L^2-rate	
			AD	A	AD	A	AD	A	AD	A
TEST1	1	3.00e-1	1.9012e-2	2.1887e-2	4.6211e-3	9.1724e-3	0	0	0	0
		1.50e-1	4.3829e-3	5.8280e-3	1.0854e-3	2.4464e-3	2.28	2.06	2.25	2.05
		7.50e-2	9.5837e-4	1.2096e-3	2.1323e-4	4.8789e-4	2.25	2.32	2.41	2.38
		3.75e-2	3.0646e-4	2.7571e-4	2.9265e-5	5.5269e-5	1.95	2.16	2.65	3.18
TEST2	π	3.00e-1	6.5375e-2	6.5375e-2	1.0682e-2	1.0682e-2	0	0	0	0
		2.25e-1	3.1934e-2	3.1598e-2	5.5980e-3	1.0043e-2	2.66	2.70	2.40	0.23
		1.50e-1	1.1276e-2	1.1276e-2	2.0116e-3	2.0116e-3	2.71	2.70	2.66	4.18
		1.13e-1	5.2093e-3	8.8807e-3	9.3905e-4	2.2073e-3	2.78	0.86	2.74	-0.33
		7.50e-2	2.4154e-3	3.6814e-3	3.9534e-4	8.6362e-4	1.94	2.22	2.19	2.37

The analysis is conducted on the 2D test cases presented in Table 1. In Figure 6 there are three time instants of both test cases.
In Table 4 there are the maximum CFL numbers and related maximum time steps Δt such that the method is stable. The time step Δt is computed by formula (3.26). By comparing the performances of a local advective (A) stabilization term against the same ones using a local advective-diffusive (AD) stabilization term, it is evident that an advective LLF flux always needs a smaller CFL with respect to an advective-diffusive LLF flux in order to stabilise the routine.

5.3. Relationship between the convective field and the foreground mesh velovity. From the theoretical explanation of the method, it does not emerge in any way an interaction between the speed of the foreground grid $\boldsymbol{V}$ and the intrinsic advective field $\boldsymbol{a}$ of the problem. In other words, there does not seem to be a limitation of the velocity of the mesh that is displaced and deformed in terms of stability of the method. The unique limitation of the mesh speed (see section 3.5) is due to the CFL condition with respect to the dimension of the single cell. In order to allow to the code to perform the automatic information transmission, the mesh speed is such that it does not allow a given fringe cell Ω_i^n in the foreground mesh to migrate beyond the boundaries of the stencil $\mathcal{S}_i$ centered on the cell itself in any time interval from t^n to t^{n+1}. As a matter of fact, if this process is not ensured, for those new born cells belonging to the background mesh at time t^{n+1} could not be able to recover the information from the polynomial interpolation. Consequently, the algorithm would incur a loss of information.
In this subsection we test on a numerical case that the stability is only given by the relative advective speed $\boldsymbol{a} - \boldsymbol{V}$ and the mesh velocity $\boldsymbol{V}$ does not affect the stability of the method in other ways. In particular, on the same linear test case, we will consider different possible movements of the foreground mesh by measuring, at final time $t = T$, the L^∞- and L^2-errors of the mismatch between the exact and the numerical solution.

Table 4: Experimental stability analysis. For both tests, the reported CFL and Δt consist in the maximum CFL number and the maximum related time step Δt such that the method is stable. Labels A and AD underline the usage of an advective and advective-diffusive stabilization term for the LLF flux, respectively. The first column reports the space steps h used for the different simulations.

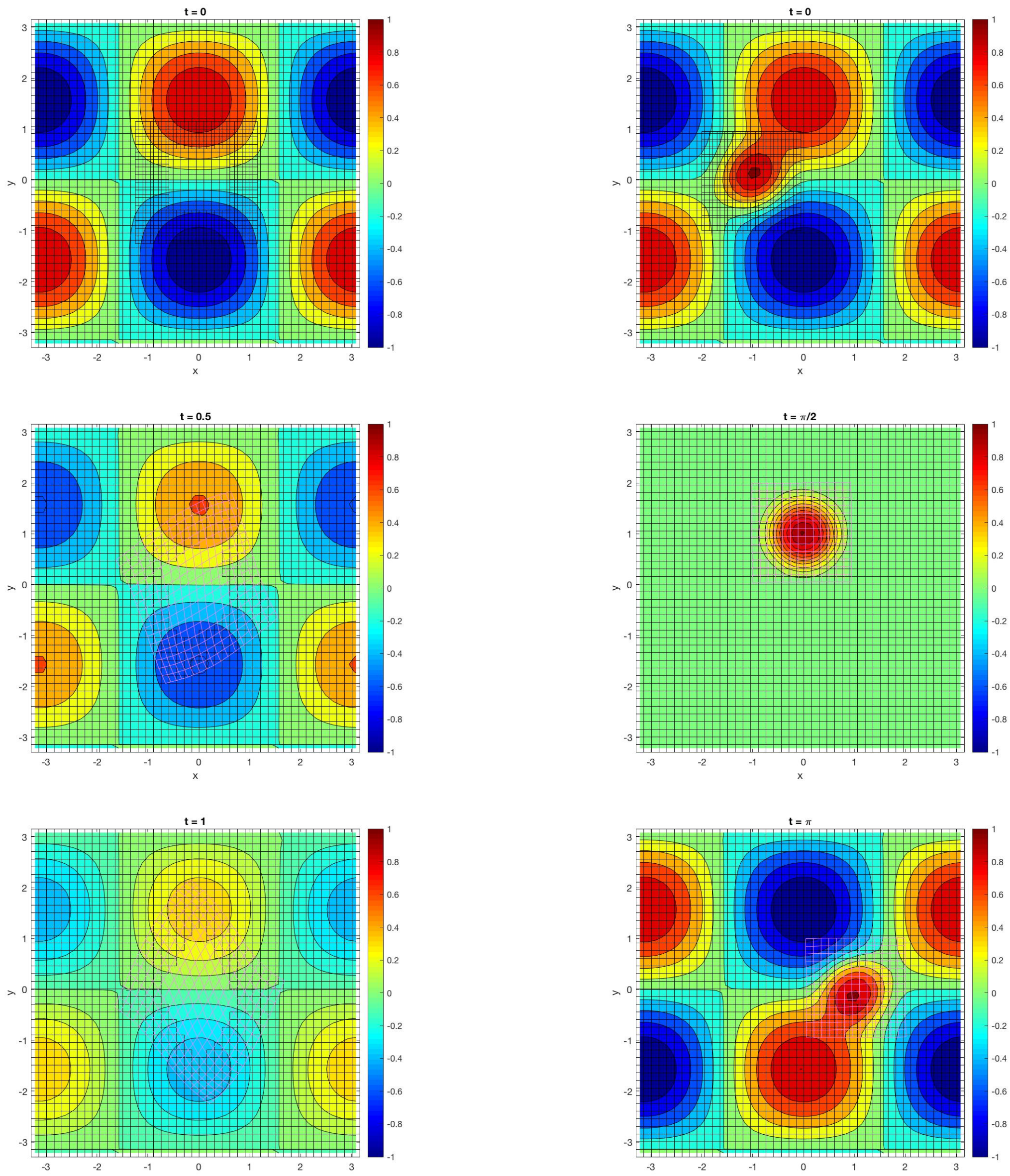
h	*TEST1*				*TEST2*			
	CFL		Δt		CFL		Δt	
	A	AD	A	AD	A	AD	A	AD
3.00e-1	0.55	0.95	2.06e-1	3.56e-1	0.75	0.95	2.81e-1	3.56e-1
1.50e-1	0.75	1.15	1.41e-1	2.16e-1	0.65	0.85	1.22e-1	1.59e-1
7.50e-2	0.75	0.95	7.03e-2	8.91e-2	0.55	0.75	5.16e-2	7.03e-2

Table 5: On the top, features of *TEST3* are reported. On the bottom, there are the three considered movements of the foreground mesh.

	TEST3
Ω	$(0,1)\times(0,5)$
Diffusion	2e-3
Advection	$[1,0]^T$
u_{ex}	$-\tanh(2(x-t)^2+5(y-1)^2)+$ $+e^{-t}(5x-x^2)(2y-y^2)+1$
B.C.	Dirichlet: $u\|_{\partial\Omega}\equiv 0$
I.C.	$-\tanh(2x^2+5(y-1)^2)$ $+(5x-x^2)(2y-y^2)+1$
T	2
fg mesh	$[0.8,1.2]^2$
$\boldsymbol{V}$	*P1*, *P2*, *P3*

$\boldsymbol{V}$	
P1	The foreground mesh is not moving for the whole period of the simulation.
P2	The foreground has a constant velocity equal to the advective velocity for any time.
P3	For half of the time the mesh moves with double the speed compared to the advective field and for the remaining half of the time the mesh moves with the same speed in modulus but in the opposite direction compared to the advective field.

The tested case is named *TEST3* and it is summed up in Table 5 (top).
The foreground mesh is either allowed not to move or to rigidly move in the parallel direction with respect to the abscissae axis. In particular, we consider three possibilities of movements, *P1*, *P2* and *P3*, reported and explained in Table 5 (bottom). We remark that test *P1* corresponds to a test case with a unique block mesh due to the position and the uniformity of the foreground mesh with respect to the background mesh. For this reason, tests *P2* and *P3* are compared with *P1*. In Figure 7 there are both the numerical solutions and the associated pointwise absolute values of the difference between the exact and numerical solution for the final time $T=2$ for the configurations listed above. In particular, the configuration of the foreground mesh in Figure 7a (left) corresponds to the initial mesh configuration for tests *P2* and *P3* too. By visualising the different plots of the errors, it is evident the movement of the foreground mesh introduces an error. As a matter of fact the errors of *P2* and *P3* are neither equal each other nor to the errors of *P1*. The quantitative differences among the different cases are reported in Table 6. Concerning test *P2*, the L^∞-error is equal to

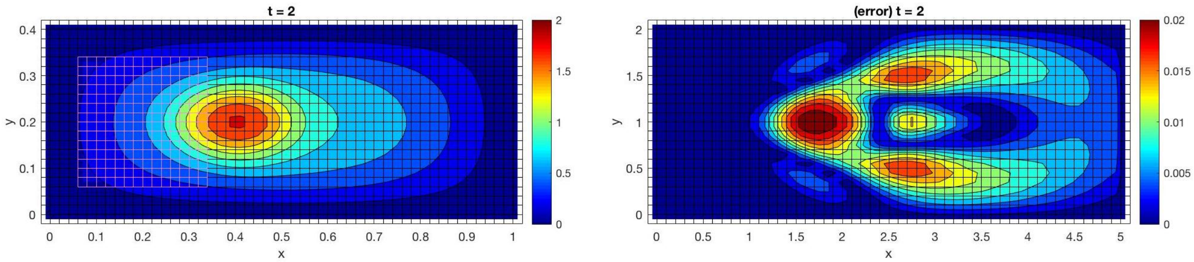


(a) *TEST1* (b) *TEST2*

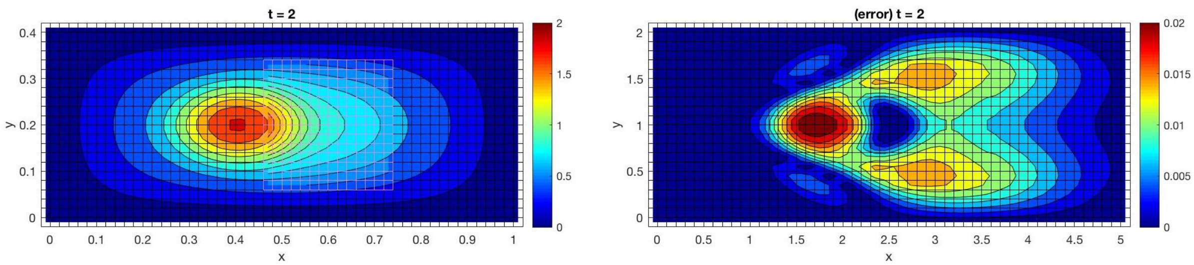
Fig. 6: Three time instants for test cases *TEST1* (a) and *TEST2* (b).

the one of *P1*, even though the L^2-error is the double. This distance between a steady and moving foreground mesh becomes slightly more evident at increasing of the mesh speed, as the last line of Table 6 shows. In any case, all the errors are comparable and this confirms that there is no relation between advective field and mesh velocity in terms of stability. The mesh velocity seems to affect the numerical solution only on the precision.

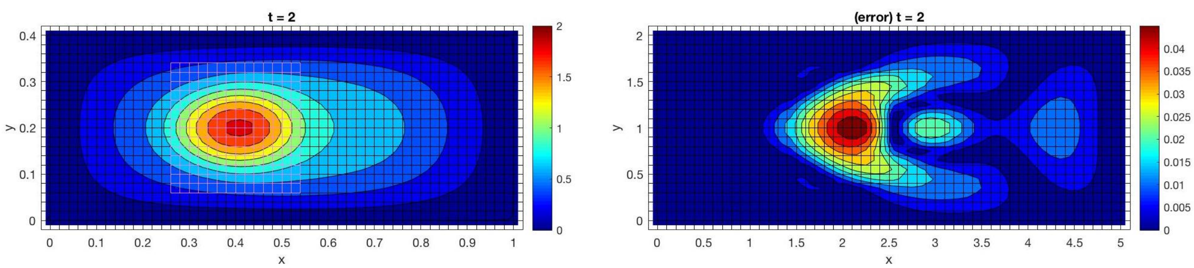
We conclude this subsection by analysing the loss of information given by a very strong speed of the foreground mesh on the same test case. The foreground mesh is still located in the subset $[0.8, 1.2]^2$ at the initial time and moves rightwards with a speed equal to 4. This velocity, with the considered time step Δt, allows to the cells on the left side of the foreground mesh to overflow from the borders of their stencil from



(a) *P1*



(b) *P2*



(c) *P3*

Fig. 7: The numerical solutions, on the left, at final time $t = 2$ of the three possibilities *P1*, *P2* and *P3* of foreground mesh movements for the *TEST3*. On the right there there are the associated pointwise errors of the mismatch between the exact solution and the numerical solution.

Table 6: Errors for *TEST3*. The errors refer to a characteristic length h equal to the cell of 2e-2 and a time $t = T = 2$.

	L^∞-err	L^2-err
P1	2.1554e-2	6.8500e-3
P2	2.1554e-2	4.8809e-3
P3	4.8809e-2	1.0864e-2

times t^n to t^{n+1}. In Figure 8 there is a comparison between the recovered numerical solution and the exact solution for $t = 0.84$ (which corresponds to that time when the right side of the moving mesh is fully aligned to the right side of the channel). There is no relation between the two solutions because the speed of the foreground mesh is so fast that it does not allow the algorithm to assign the correct information about the background cells that arise in the wake of the foreground mesh itself.

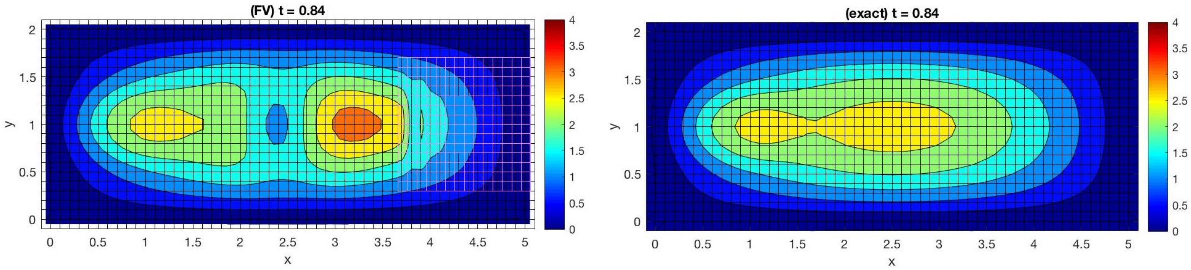


Fig. 8: Comparison between the numerical (left) and exact (rigth) solution of *TEST3* at time $t = 0.84$ for a moving foreground mesh travelling with a speed generating a loss of information.

5.4. Further topics. We conclude this section by presenting three test cases that show the potentiality of the method. Firstly, a *nonlinear advection-diffusion system* is solved; successively a *multimesh* setting of grids is considered for the already described *TEST2* (see Table 1); finally, we consider a test case with a *complex domain* in which the foreground mesh is employed in order to adapt its shape to the shape of the domain.

5.4.1. Nonlinear system. Let $\Omega = [-\pi, \pi]^2$ and $T = 0.5$ be the computational domain and the final time, respectively. Thus the problem is: *find* $\boldsymbol{u} : \Omega \times [0, T] \to \mathbb{R}^2$ *such that*:

$$\begin{cases} \partial_t \boldsymbol{u} + \nabla \cdot (\boldsymbol{u}\boldsymbol{u}^T) = \nu \Delta \boldsymbol{u} + \boldsymbol{f} & \text{in } \Omega \times [0, T] \\ \boldsymbol{u} \equiv \boldsymbol{u}_{ex} & \text{on } \partial\Omega \times [0, T] \\ \boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_{ex}(\boldsymbol{x}, 0) & \text{in } \overline{\Omega} \times \{0\} \end{cases}, \tag{5.4}$$

where the force term $\boldsymbol{f}$ is chosen to have the exact solution

$$\boldsymbol{u}_{ex}(x, y, t) = e^{-t} \begin{bmatrix} \cos(x)\sin(y) \\ -\sin(x)\cos(y) \end{bmatrix}.$$

In problem (5.4), the diffusive term ν is equal to $5\pi \times 10^{-3}$ while the convective field is represented by the solution itself, thus the partial differential equation is nonlinear. For this problem, the convective-diffusive component $\boldsymbol{F}$ is the matrix $\boldsymbol{u}\boldsymbol{u}^T - \nu \nabla \boldsymbol{u}$. The foreground mesh is originally located around the center of mass of the whole domain and it is allowed to rigidly counterclockwise rotate. In Figure 9 there are the two components of the numerical solution at final time $t = T$.

The error and convergence analysis is conducted as for the already presented linear test cases by comparing the performances of the flux discretization either with local advective-diffusive or just advective stabilization term. For this reason, Table 7 reports the L^∞ and L^2 errors and convergence rates by decreasing four times the characteristic length h of the cells. As already observed for the linear tests, also in this specific nonlinear case the errors of AD and A fluxes are similar even though an AD discretization is almost always more precise. Finally, we remark that both flux approximations have a second order discretization rate, as we expected *a priori*.

5.4.2. Multimesh setting. The presented method can be easily extended to more than one foreground mesh. As a matter of fact, different meshes can be set with an independent movement and such that to exchange information with the background grid and with the other moving foreground meshes. Due to the possibility to move, the foreground meshes can overlap each other. Consequently, the hole will be present in the background as well as in some foreground grids by properly applying the same dynamics of the overlapping zone of Section 3.5 to the specific intermediate foreground mesh.

In order to compare the performances of multimesh setting with two moving foreground meshes, we considered the presented case *TEST2* with a foreground mesh clockwise rotating around the origin (see Table 1) by adding a second foreground mesh. The new grid is originally located to subset $[-0.78, -0.18] \times [-0.62, -0.02]$ and horizontally moves on the right with a constant velocity $\boldsymbol{V}_2 = [-0.8, 0]^T$ (see Figure 10). The new grid intercepts the original foreground mesh at the beginning and at the end of the simulation. For this reason,

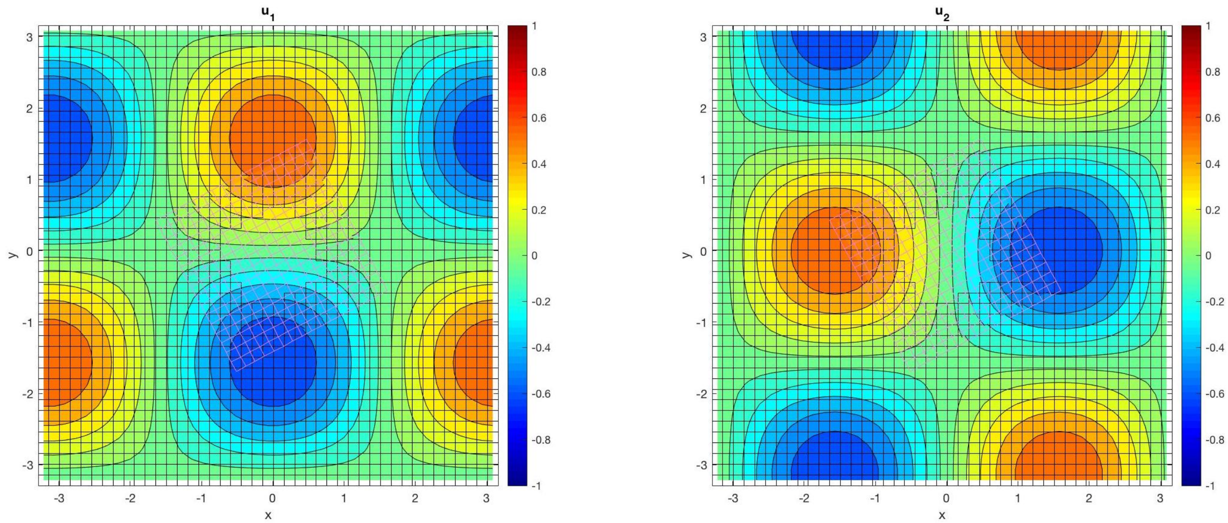


Fig. 9: Components of the solution of nonlinear test at time $t = T = 0.5$.

Table 7: Convergence analysis of the nonlinear test case. The errors refer to time $t = T = 0.5$.

h	L^∞-err		L^2-err		L^∞-rate		L^2-rate	
	AD	A	AD	A	AD	A	AD	A
3.00e-1	2.3700e-2	2.01643e-2	5.2187e-3	4.9065e-3	0	0	0	0
1.50e-1	5.2138e-3	5.8552e-3	1.1061e-3	1.5086e-3	2.36	1.93	2.42	1.84
7.50e-2	2.4113e-3	2.4344e-3	2.4506e-4	5.7129e-4	1.15	1.33	1.30	1.44
3.75e-2	6.1828e-4	6.4658e-4	1.0332e-4	1.4322e-4	1.99	1.94	2.16	2.02

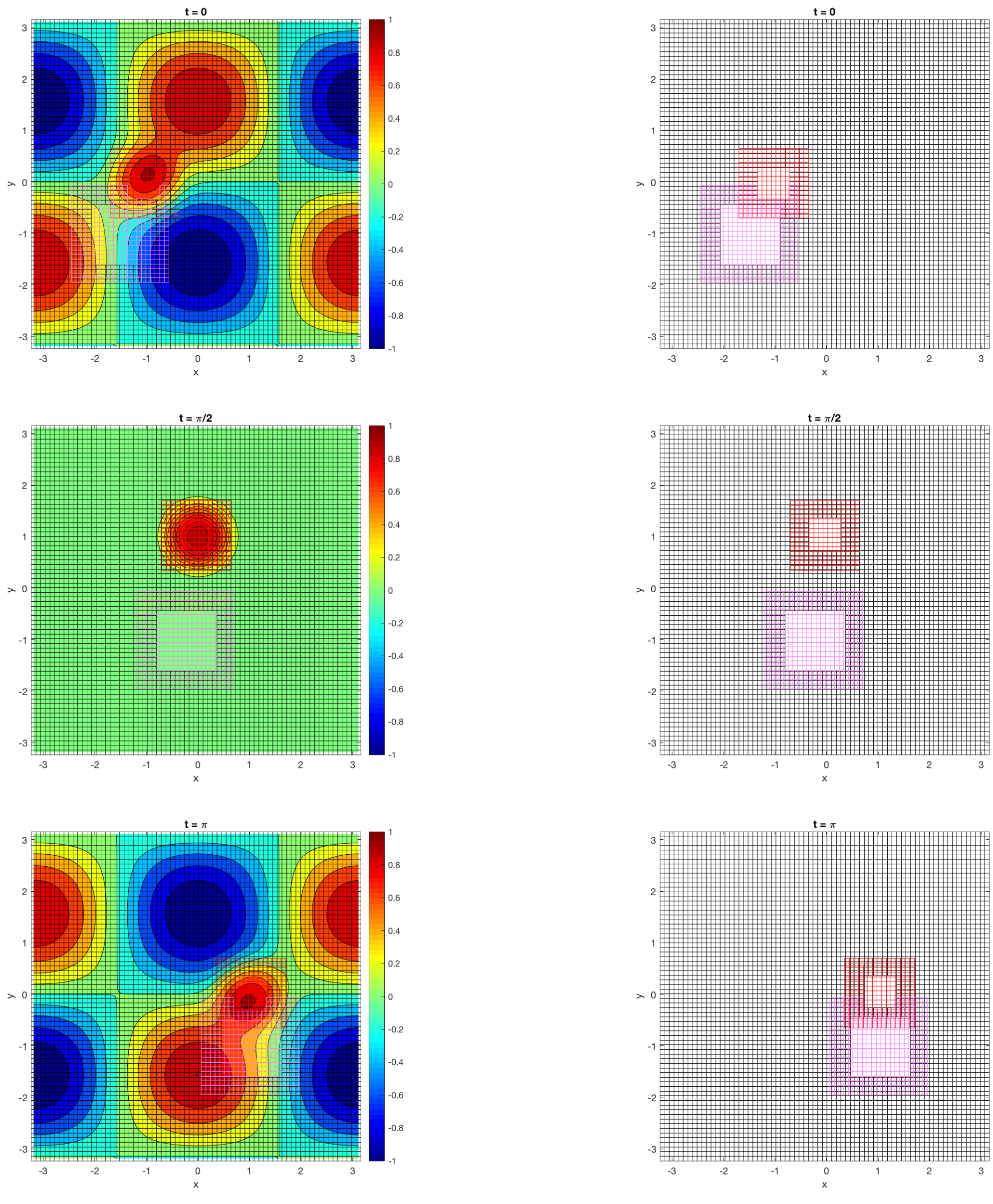
the original foreground mesh partially covers the new mesh by creating a new partial hole on it (see first and last rows in Figure 10b). Moreover a new hole is generated in the background. Since each foreground mesh is independent from the other, the holes in the background can be ether connected (if the foreground grids overlap each other) or unconnected (if the foreground meshes are far enough to not overlap each other). Figure 10a refers to the solution where each grid is defined by squared grids whose cells have a characteristic length $h = 7.50e{-}2$. The L^∞- and L^2-errors with respect to the exact solution are exactly the same reported in Table 3 (last row). This means that the new grid does not influence the performance of the method with respect to the previous grid setting.

5.4.3. Complex domains. An important application of chimera grids is the possibility to use meshes fitting the particular shape of the domain (which eventually evolves in time) by preserving a Cartesian background mesh. Here we present a test case summed up in Table 8. For any positive time t, let the generic moving ball formally be

$$B(\rho_{\min}, \rho_{\max}; t) = \big\{(x, y) \in \mathbb{R}^2 : x = \rho\cos(\theta),\ y = \rho\sin(\theta) - 2t - \pi;\ \text{with } (\rho, \theta) \in [\rho_{\min}, \rho_{\max}] \times [0, 2\pi]\big\}.$$

The domain is the channel of dimensions $[-\pi, \pi] \times [-2\pi, 2\pi]$ from which the moving circle $B(0, 0.5; t)$ of radius equal to 0.5 is subtracted at any time $t \in [0, T]$. The circle vertically moves downwards with a constant velocity. In Figure 11a it is reported the numerical solution at the initial and final time instants for the numerical test. In Figure 11b there is a focus on the grid settings. For the foreground mesh, a polar structured grids is employed. It fits the shape of the domain and moves as the domain evolves.

6. Conclusions. We presented a second order finite volume scheme for unsteady advection-diffusion PDEs on overset grid. The scheme is based on an extension of the ADER method to advection-diffusion



(a) (b)

Fig. 10: On the left (a), the solution of *TEST2* for three time instants with a multimesh setting composed of two foreground meshes; on the right (b), for the same time instants, the configuration of the background and foreground grids.

equations with compact data transmission conditions from the background to the foreground meshes and *vice versa*. We also introduced a new stabilization term for approximating the fluxes through a Local Lax-Friederichs approach.

The numerical illustrations for linear and non-linear systems show that background and foreground moving meshes do not introduce spurious perturbation to the solution, uniformly reaching second order accuracy in space and time. In addition, we showed that the speed of the foreground mesh does not influence the stability of the method. Our results also show that the new LLF stabilization speed improves

Table 8: Summary scheme of *TEST4*.

	TEST4
Ω	$[-\pi,\pi]\times[-2\pi,2\pi]/B(0,0.5;t)$
Diffusion	0.05
Advection	$[0,-2]^T$
$u_{ex}(x,y,t)$	$\exp[-x^2-(y-2t-\pi)^2+0.5](\cos(t)+1)$
B.C.	Dirichlet: $u_{ex}(x,y,t)\|_{\partial\Omega}$
I.C.	$u_{ex}(x,y,0)$
T	$\pi/2$
fg mesh	$B(0.5,1.5;0)$
$\boldsymbol{V}$	$[0,-2]^T$

the precision and robusteness of the numerical solution and allows a less restrictive CFL condition. Finally, it is shown that several foreground meshes, possibly overlapping and with independent displacements, can seamlessly be employed thanks to this approach.

Future investigations will extend this integration scheme to the compressible and incompressible Navier-Stokes equations.

Acknowledgments. The authors thank Haysam Telib of Optimad Engineering (Turin, Italy) for his industrial view point in support of the whole work. The authors acknowledge the support by European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Actions, grant agreement 872442 (ARIA). MGC is pleased to thank the company Optimad Engineering for hosting him during part of the research in Turin.

REFERENCES

[1] P. Angot, C.-H. Bruneau, and P. Fabrie, *A penalization method to take into account obstacles in incompressible viscous flows*, Numerische Mathematik, 81 (1999), pp. 497–520.

[2] J. W. Banks, W. D. Henshaw, A. K. Kapila, and D. W. Schwendeman, *An added-mass partition algorithm for fluid–structure interactions of compressible fluids and nonlinear solids*, Journal of Computational Physics, 305 (2016), pp. 1037–1064.

[3] J. W. Banks, W. D. Henshaw, and B. Sjögreen, *A stable fsi algorithm for light rigid bodies in compressible flow*, Journal of Computational Physics, 245 (2013), pp. 399–430.

[4] J. W. Banks, D. W. Schwendeman, A. K. Kapila, and W. D. Henshaw, *A high-resolution godunov method for compressible multi-material flow on overlapping grids*, Journal of Computational Physics, 223 (2007), pp. 262–297.

[5] J. Benek, P. Buning, and J. Steger, *A 3-d chimera grid embedding technique*, in 7th Computational Physics Conference, 1985, p. 1523.

[6] W. Boscheri and M. Dumbser, *Arbitrary-lagrangian-eulerian one-step weno finite volume schemes on unstructured triangular meshes*, Communications in Computational Physics, 14 (2013), pp. 1174–1206.

[7] S. Busto, S. Chiocchetti, M. Dumbser, E. Gaburro, and I. Peshkov, *High order ader schemes for continuum mechanics*, Frontiers in Physics, 8 (2020), p. 32.

[8] C. E. Castro and E. F. Toro, *Solvers for the high-order riemann problem for hyperbolic balance laws*, Journal of Computational Physics, 227 (2008), pp. 2481–2513.

[9] G. Chesshire and W. D. Henshaw, *Composite overlapping meshes for the solution of partial differential equations*, Journal of Computational Physics, 90 (1990), pp. 1–64.

[10] I.-T. Chiu and R. Meakin, *On automating domain connectivity for overset grids*, in 33rd Aerospace Sciences Meeting and Exhibit, 1995, p. 854.

[11] M. Dumbser, W. Boscheri, M. Semplice, and G. Russo, *Central weighted eno schemes for hyperbolic conservation laws on fixed and moving unstructured meshes*, SIAM Journal on Scientific Computing, 39 (2017), pp. A2564–A2591.

[12] M. Dumbser, C. Enaux, and E. F. Toro, *Finite volume schemes of very high order of accuracy for stiff hyperbolic balance laws*, Journal of Computational Physics, 227 (2008), pp. 3971–4001.

[13] M. Dumbser and M. Käser, *Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems*, Journal of Computational Physics, 221 (2007), pp. 693–723.

[14] R. Glowinski, T.-W. Pan, and J. Periaux, *A fictitious domain method for dirichlet problem and applications*, Computer Methods in Applied Mechanics and Engineering, 111 (1994), pp. 283–303.

[15] J. Guerrero, *Overset composite grids for the simulation of complex moving geometries*, DICAT, University of Genoa, Italy, (2006).

[16] W. D. Henshaw, *On multigrid for overlapping grids*, SIAM Journal on Scientific Computing, 26 (2005), pp. 1547–1572.

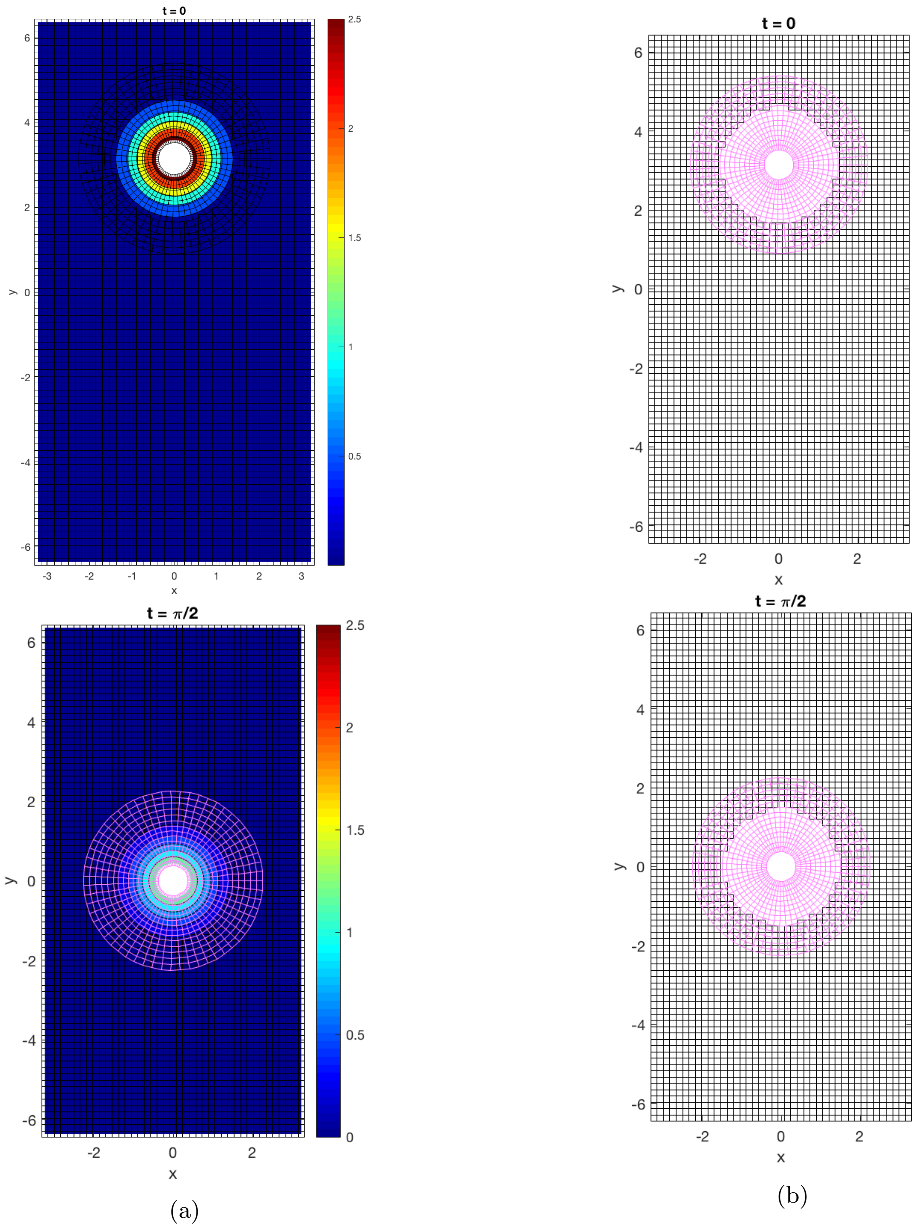


Fig. 11: On the left (a), the solution of *TEST4* for the initial and final time instants; on the right (b), for the same time instants, the background and foreground grids setting.

[17] A. Hidalgo and M. Dumbser, *Ader schemes for nonlinear systems of stiff advection–diffusion–reaction equations*, Journal of Scientific Computing, 48 (2011), pp. 173–189.

[18] C. W. Hirt, A. A. Amsden, and J. Cook, *An arbitrary lagrangian-eulerian computing method for all flow speeds*, Journal of computational physics, 14 (1974), pp. 227–253.

[19] G. Houzeaux, J. Cajas, M. Discacciati, B. Eguzkitza, A. Gargallo-Peiró, M. Rivero, and M. Vázquez, *Domain decomposition methods for domain composition purpose: chimera, overset, gluing and sliding mesh methods*, Archives of Computational Methods in Engineering, 24 (2017), pp. 1033–1070.

[20] K.-H. Kao and M.-S. Liou, *Advance in overset grid schemes-from chimera to dragon grids*, AIAA journal, 33 (1995), pp. 1809–1815.

[21] M.-S. Liou and Y. Zheng, *A novel approach of three-dimensional hybrid grid methodology: Part 2. flow solution*, Computer methods in applied mechanics and engineering, 192 (2003), pp. 4173–4193.

[22] R. Meakin, *Chapter 11: Composite overset structured grids*, Handbook of grid generation, JF Thompson, BK Soni and

NP Weatherill. CRC, (1999).
[23] G. I. Montecinos, *Numerical methods for advection-diffusion-reaction equations and medical applications*, PhD thesis, University of Trento, 2014.
[24] G. B. Nagy, O. E. Ortiz, and O. A. Reula, *The behavior of hyperbolic heat equations' solutions near their parabolic limits*, Journal of Mathematical Physics, 35 (1994), pp. 4334–4356.
[25] C. S. Peskin, *The immersed boundary method*, Acta numerica, 11 (2002), pp. 479–517.
[26] N. A. Petersson, *Hole-cutting for three-dimensional overlapping grids*, SIAM Journal on Scientific Computing, 21 (1999), pp. 646–665.
[27] A. Raeli, M. Bergmann, and A. Iollo, *A finite-difference method for the variable coefficient poisson equation on hierarchical cartesian meshes*, Journal of computational Physics, 355 (2018), pp. 59–77.
[28] D. Schwendeman, A. Kapila, and W. Henshaw, *A study of detonation diffraction and failure for a model of compressible two-phase reactive flow*, Combustion Theory and Modelling, 14 (2010), pp. 331–366.
[29] G. Starius, *Composite mesh difference methods for elliptic boundary value problems*, Numerische Mathematik, 28 (1977), pp. 243–258.
[30] G. Starius, *Constructing orthogonal curvilinear meshes by solving initial value problems*, Numerische Mathematik, 28 (1977), pp. 25–48.
[31] G. Starius, *On composite mesh difference methods for hyperbolic differential equations*, Numerische Mathematik, 35 (1980), pp. 241–255.
[32] V. A. Titarev and E. F. Toro, *Ader: Arbitrary high order godunov approach*, Journal of Scientific Computing, 17 (2002), pp. 609–618.
[33] V. A. Titarev and E. F. Toro, *Ader schemes for three-dimensional non-linear hyperbolic systems*, Journal of Computational Physics, 204 (2005), pp. 715–736.
[34] E. F. Toro and G. I. Montecinos, *Advection-diffusion-reaction equations: hyperbolization and high-order ader discretizations*, SIAM Journal on Scientific Computing, 36 (2014), pp. A2423–A2457.
[35] E. Volkov, *The method of composite meshes*, Automatic Programming, Numerical Methods and Functional Analysis, 96 (1970), p. 145.
[36] Z. Wang, *A fully conservative interface algorithm for overlapped grids*, Journal of Computational Physics, 122 (1995), pp. 96–106.