



HAL
open science

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

Michel Bergmann, Michele Giuliano Carlino, Angelo Iollo

► To cite this version:

Michel Bergmann, Michele Giuliano Carlino, Angelo Iollo. Second order scheme for unsteady advection-diffusion on moving overset grids with a compact transmission condition. 2021. hal-03119830v1

HAL Id: hal-03119830

<https://inria.hal.science/hal-03119830v1>

Preprint submitted on 25 Jan 2021 (v1), last revised 26 Oct 2021 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

1 SECOND ORDER SCHEME FOR UNSTEADY ADVECTION-DIFFUSION ON MOVING 2 OVERSET GRIDS WITH A COMPACT TRANSMISSION CONDITION*

3 MICHEL BERGMANN^{†‡}, MICHELE GIULIANO CARLINO^{†‡}, AND ANGELO IOLLO^{†‡}

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

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

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

17 **1. Introduction.** One of the main difficulties for the simulation of a phenomenon modeled by a Partial
18 Differential Equation (PDE) is the geometrical modeling of the computational domain with a single mesh
19 block. This problem is especially relevant when the domain is complex or its shape and its topology evolve
20 during the simulation. Classical approaches to tackle this problem include the Arbitrary Lagrangian-Eulerian
21 (ALE) method, fictitious domain approaches and Chimera grids. ALE methods [12] allow a certain degree of
22 mesh deformation and adaptation thanks to an appropriate reformulation of the governing equations and to
23 sophisticated and efficient grid displacement algorithms. However, when the grid deformation leads to exces-
24 sively stretched cells, a delicate (and computationally expensive) global re-meshing step may be necessary.
25 In turn, this operation can introduce approximation irregularities that are caused by the interpolation of the
26 solution from the old grid to the new one. In fictitious domain approaches, including immersed boundary or
27 penalization methods, the original problem is discretised on a simple mesh, usually structured and cartesian,
28 constant in time [9, 18, 1]. The grid hence does not necessarily fit the physical boundaries and special
29 care must be taken to attain a sufficient degree of accuracy at the boundaries. Moreover, the presence of
30 thin boundary layers can significantly reduce the computational advantages deriving from a simple meshing
31 algorithm, because of the uniform aspect ratio of the mesh.

32
33 We focus our investigations on Chimera grids [24, 2, 15, 19]. Chimera grids consist of multiple overlap-
34 ping mesh blocks that together define an overset grid. Usually, one has a background mesh that includes one
35 or more foreground mesh patches that are fitted to the physical domain boundaries. This mesh generation
36 approach considerably simplifies the task of mesh adaptation in the case of boundary layers, changing geom-
37 etry for an unsteady problem (e.g. fluid-structure interaction problems in fluid-dynamics) and for unsteady
38 multiply connected domains. Once the multiple mesh patches are generated, they are collated in order to
39 obtain an appropriate overlapping zone between the neighboring blocks [15]. In the overlap zones, the ex-
40 change of solution information from one grid to another is performed. A compact transmission condition
41 is generally sought in order to limit communications between the grids. However, when a minimal number
42 of cells are locally employed for transferring information among the computational blocks, the order of the
43 scheme can be degraded.

44
45 In this paper, we propose a space-time Finite Volume scheme on Chimera grids. Our objective is
46 to combine some aspects of an ALE approach, notably its flexibility with respect to grid displacement and
47 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 discretization stencil to exchange information between different mesh patches, in the spirit of previous works on cartesian hierarchical grids [20]. 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.

In general, the numerical solution on Chimera grids is obtained in two steps. In a first step one computes the solution in the background mesh and in a second step in the foreground mesh by strongly imposing at the boundaries of the latter mesh the solution found in the previous step. For example, in [10, 25, 14], 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. This oneway approach from the background to the foreground mesh through *donor/receptor* cells was originally proposed in [5].

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 [13] for further details.

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. Hence, special conditions onto fringe cells are no longer needed and iterative methods are not any more necessary to align the solution in the different blocks (as for domain decomposition methods).

The Arbitrary high order DERivatives (ADER) method provides an ideal setting for pursuing our purpose. In [6, 22, 21, 4], the authors presented a method to recover an accurate solution for hyperbolic PDEs with an arbitrary order of accuracy on a single mesh block. 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 $\Omega(t) \subset \mathbb{R}^2$ be the time-dependent computational domain and let T be a positive real. In the following we consider the parabolic problem: *find* $u : \Omega(t) \times [0, T] \rightarrow \mathbb{R}$ *such that*

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

closed with appropriate initial and boundary conditions. Problem (1.1) is a rather general representation of an advection-diffusion model. For the sake of simplicity in the derivation of the numerical method, we focus on a scalar solution $u(\mathbf{x}, t)$. However, the method can be applied to a generic system whose solution is a vector \mathbf{u} of dimension larger than one. In (1.1) the diffusive-convective vector $\mathbf{F}(u, \nabla u) \in \mathbb{R}^2$, eventually nonlinear, and the force term $f(\mathbf{x}, t)$ are defined. In particular, the problem is linear when the diffusive-convective term is written as $\mathbf{F}(u, \nabla u) = \mathbf{a}u - \nu \nabla u$, where $\mathbf{a} : \Omega \times [0, T] \rightarrow \mathbb{R}^2$ is the advective field and $\nu : \Omega \times [0, T] \rightarrow \mathbb{R}_+$ 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) stabi-

101 lization term is introduced and contrasted with the LLF term from the literature. Section 5 is devoted to
 102 the numerical results. In particular, first the second order analysis is conducted on linear 1D and 2D test
 103 cases; successively, we focus on the stability of the method by comparing the performances of the differ-
 104 ent LLF fluxes. At the end of the numerical test cases section, we show results for a nonlinear system of
 105 PDEs, for multiblock grid setting, meshes and time-dependent overset grids for multiply connected domain.
 106 Conclusions are reported in Section 6.

107 **2. The overset grid.** An overset grid or Chimera mesh is a set of mesh blocks covering the compu-
 108 tational domain. Each block may overlap other block(s) in some particular sub-region(s) said *overlapping*
 109 *zone(s)*. Once the multiple mesh patches are generated, they are collated in order to generate an appropriate
 110 topology [15]. Consequently, an overlapping zone between two neighbouring blocks is defined. For the sake
 111 of simplicity with no loss of generality, the whole method is explained by considering a two blocks overset
 112 grid (i.e., the background and the foreground meshes). In Section 5.4.2 of test cases, a multiblock setting is
 113 presented. Figure 1 shows an overset grid; in black there is the *background* mesh and in blue the *foreground*
 114 mesh. In particular, the foreground mesh can move and deform. The overlapping zone is necessary for the
 115 communication and data transfer from one mesh to the other.
 116 In this work, the cell of any block mesh is considered quadrilateral. In particular, when all the cells are
 117 squared, the mesh is uniform. When the cells are either squared or rectangular and the edges are oriented
 118 as the Cartesian axes, the mesh is said to be Cartesian.

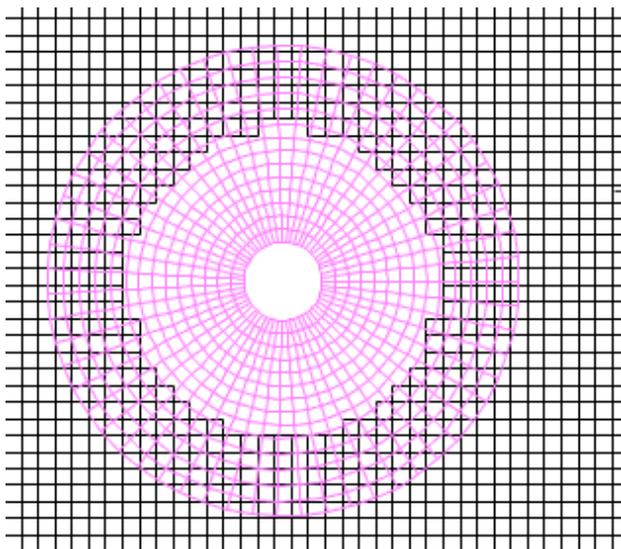


Fig. 1: Example of Chimera grid configuration. In black there is the *background* mesh and in blue the *foreground* mesh.

119 **2.1. The automatic definition of the stencil at the transmission condition.** Let $\mathcal{T}_k = \{\Omega_i^k\}_{i=1}^{N_k}$
 120 be the partition composed of N_k cells referring to the k -th block mesh (in order to simplify the notation, we
 121 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
 122 cell Ω_i . Thus, stencil \mathcal{S}_i is the set collecting the indexes of neighboring cells to Ω_i . By abuse of language,
 123 sometimes we will refer to the physical set $\Omega_i \cup \bigcup_{j \in \mathcal{S}_i} \Omega_j$ as the stencil.
 124 It is possible to distinguish two classes of cells with respect to their proximity to the overlapping interface.
 125 The definition of the stencil depends on the class.
 126 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
 127 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 $j \in \mathcal{S}_i$,
 128 also belong to \mathcal{T}_1 .
 129 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
 130 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
 131 the same partition (see right edge of cell Ω_{16} in Figure 2b). For these cells, we aim in automatically finding

132 the other cells of partition \mathcal{T}_j ($j \neq k$) belonging to the stencil. Let the extremes of the edge be indicated as
 133 \mathbf{v}_1 and \mathbf{v}_2 and its middle point with \mathbf{v}_3 , respectively. Point \mathbf{c}_* is the center of mass of generic cell Ω_* . For
 134 our numerical tests, Algorithm 2.1 is adopted through the two steps:

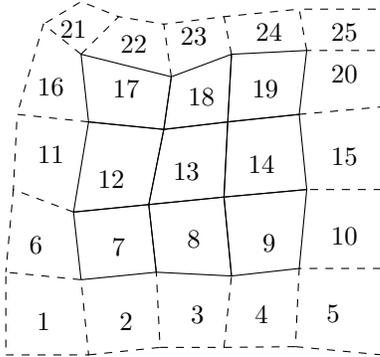
- 135 1. look for the nodes of cells of the other partition \mathcal{T}_j minimizing the Euclidean distance with respect
 136 to points \mathbf{v}_μ , $\mu = 1, 2, 3$, (line 5, see Figure 3a);
- 137 2. compute the symmetric points $\tilde{\mathbf{v}}_\mu$ of center \mathbf{c}_i^k with respect to points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

140 For the edges shared by other cells in the same partition, the cells of the stencil will be those ones sharing
 141 at least one vertex (as cells of indexes 13, 14, 17, 19 and 20 in Figure 2b).

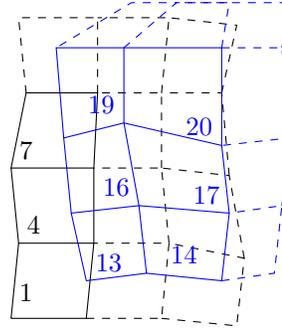
142 The routine presented in this section will be run whenever the foreground mesh configuration as well as the
 143 hole change.

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

Input: $\Omega_i^k, e_{il}^k, \mathcal{T}_j, \mathcal{S}_i^k;$ $\triangleright j \neq k$, i.e. \mathcal{T}_j is the other partition with respect to \mathcal{T}_k
 1: Initialize \mathbf{v}_1 and \mathbf{v}_2 as the two vertexes of edge e_{il}^k ;
 2: $\mathbf{v}_3 \leftarrow (\mathbf{v}_1 + \mathbf{v}_2)/2$; \triangleright Middle point of edge e_{il}^k
 3: $\mathcal{Z}_j \leftarrow \emptyset$; \triangleright Temporary set of indexes of partition \mathcal{T}_j
 4: **for** $\mu = 1, 2, 3$ **do**
 5: $\mathcal{Z}_j \leftarrow \mathcal{Z}_j \cup \{n = 1, \dots, N_j : \|\mathbf{v}_\mu - \mathbf{c}_n^j\| \leq \|\mathbf{v}_\mu - \mathbf{c}_m^j\| \quad \forall m = 1, \dots, N_j\}$;
 6: $\tilde{\mathbf{v}} \leftarrow 2\mathbf{v}_\mu - \mathbf{c}_i^k$; \triangleright Symmetric point of cellcenter \mathbf{c}_i^k of Ω_i^k with respect to \mathbf{v}_μ
 7: $\mathcal{Z}_j \leftarrow \mathcal{Z}_j \cup \{n = 1, \dots, N_j : \|\tilde{\mathbf{v}} - \mathbf{c}_n^j\| \leq \|\tilde{\mathbf{v}} - \mathbf{c}_m^j\| \quad \forall m = 1, \dots, N_j\}$;
 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 $\mathcal{S}_{13} = \{7, 8, 9, 12, 14, 17, 18, 19\}$.

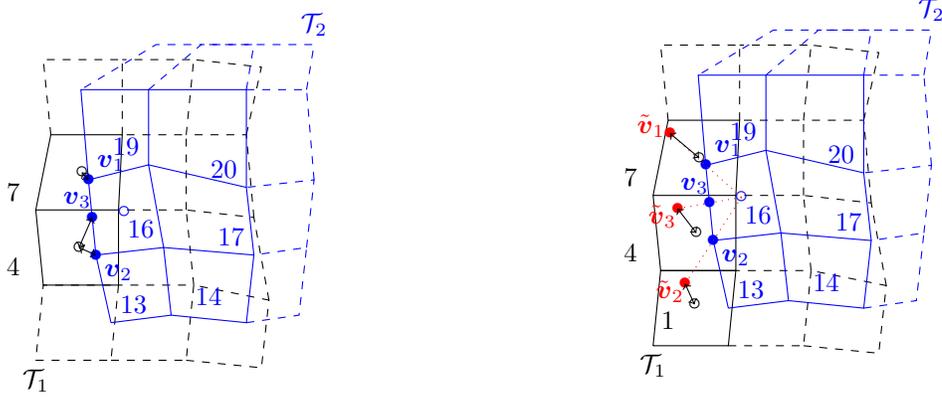


(b) A stencil of cells not belonging to the same partition. Continuous line for the stencil $\mathcal{S}_{16} = \{1, 4, 7, 13, 14, 17, 19, 20\}$.

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

144 **3. The numerical method.** Once the stencil has been defined, the numerical method can both nu-
 145 merically solve problem (1.1) and eventually evolve the overset grid. In this section the scheme is presented.
 146 The method consists in a Finite Volume predictor-corrector scheme stabilised with a Local Lax-Friederichs
 147 approach whose stabilization coefficient is explained in the following section.

148 **3.1. Local polynomial reconstruction.** The first step of the numerical method is to recover a recon-
 149 struction of the solution over any point of the actual cell Ω_i . Since the scheme is cell-centered, at time t^n ,



(a) First step: by identifying the vertexes v_1 and v_2 and the middle point 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{v}_μ , $\mu = 1, 2, 3$, (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 $\Omega_{16} \in \mathcal{T}_2$.

150 we would like to extend (at least locally) the solution to the whole cell by exploiting the information in the
 151 cells of the stencil referring to Ω_i^n . In order to explain the reconstruction, let us consider a generic regular¹
 152 function $\phi : E \rightarrow \mathbb{R}$ 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
 153 zone, the cell composing the subdomain E does not necessary fulfill the non-overlapping condition, i.e., it
 154 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
 155 know the value of function ϕ over the center of mass $(x_k, y_k) = \mathbf{x}_k$, with $k \in \{i\} \cup \mathcal{S}_i$, of any Ω_k composing
 156 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
 157 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
 158 to write the Taylor's polynomial truncated to the quadratic terms with respect to ϕ_i :

$$159 \quad (3.1) \quad \begin{aligned} \phi(x, y) &= \phi_i + \partial_x \phi_i (x - x_i) + \partial_y \phi_i (y - y_i) + \partial_{xy}^2 \phi_i (x - x_i)(y - y_i) \\ &\quad + \frac{1}{2} \partial_{xx}^2 \phi_i (x - x_i)^2 + \frac{1}{2} \partial_{yy}^2 \phi_i (y - y_i)^2 + \mathcal{O}(H^3), \end{aligned}$$

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

$$162 \quad (3.2) \quad p_1 = \partial_x \phi_i \quad p_2 = \partial_y \phi_i \quad p_3 = \partial_{xy} \phi_i \quad p_4 = \partial_{xx} \phi_i \quad p_5 = \partial_{yy} \phi_i,$$

163 the Taylor's expansion (3.1) can be seen as a linear combination of the components of the basis $\{1, x -$
 164 $x_i, y - y_i, (x - x_i)(y - y_i), \frac{1}{2}(x - x_i)^2, \frac{1}{2}(y - y_i)^2\}$ which defines the polynomial space function \mathcal{Q}_2 of quadratic
 165 polynomials centered in \mathbf{x}_i ; thus the polynomial interpolation function $\Pi_i \phi$ reads:

$$166 \quad (3.3) \quad \Pi_i \phi(x, y) = \phi_i + p_1(x - x_i) + p_2(y - y_i) + p_3(x - x_i)(y - y_i) + \frac{1}{2} p_4(x - x_i)^2 + \frac{1}{2} p_5(y - y_i)^2,$$

167 with the polynomial coefficients p_l , $l = 1, \dots, 5$, to be sought. By imposing as constraint that the polynomial
 168 $\Pi_i \phi(x, y)$ 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
 169 in the unknown polynomial coefficients arises:

$$170 \quad (3.4) \quad \begin{bmatrix} h_{ik}^x & h_{ik}^y & h_{ik}^x h_{ik}^y & \frac{1}{2}(h_{ik}^x)^2 & \frac{1}{2}(h_{ik}^y)^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{ij}^x & h_{ij}^y & h_{ij}^x h_{ij}^y & \frac{1}{2}(h_{ij}^x)^2 & \frac{1}{2}(h_{ij}^y)^2 \end{bmatrix} \begin{bmatrix} p_1 \\ \vdots \\ p_5 \end{bmatrix} = \begin{bmatrix} \delta \phi_{ik} \\ \vdots \\ \delta \phi_{ij} \end{bmatrix},$$

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

171 with $h_{ij}^x = x_j - x_i$, $h_{ij}^y = y_j - y_i$ and $\delta\phi_{ij} = \phi_j - \phi_i$, for $j \in \mathcal{S}_i$. The algebraic system (3.4) has to be
 172 solved in least-square sense if $|\mathcal{S}_i| > 5$. Moreover, if the chosen polynomial basis is not reduced, namely if
 173 the Taylor's expansion (3.1) is arrested to the bi-linear or linear terms, the stencil has to contain at least 5
 174 cells in order to ensure a solution for (3.4).

175 This method allows to locally reconstruct all over the stencil a given function. If the function is defined
 176 over the computational domain $\Omega \subset \mathbb{R}^2$ and it is continuous everywhere, then the reconstruction is locally
 177 computed over any stencil and the ensured order of convergence is 3.

178 In the sequel, the local polynomial reconstruction $\Pi_i u^n$ will be referred as w_i^n .

179 **3.2. Local space-time Galerkin predictor.** Let be the time interval $[0, T]$ subdivided in N subin-
 180 tervals $[t^n, t^{n+1}]$, with $n = 0, \dots, N - 1$; thus for a generic time-dependent variable $g(t)$, we define g^n for
 181 $g^n = g(t^n)$. In particular, the domain Ω^n and the solution u^n at time t^n are considered the actual spatial
 182 configuration and the actual time, respectively. Let $\mathcal{C}_i^n = \Omega_i(t) \times [t^n, t^{n+1}]$ be the physical space-time cell
 183 whose lower and upper bases represent the evolution of cell $\Omega_i(t)$ from t^n to t^{n+1} . First, the governing
 184 equation (1.1) is rewritten with respect to a space-time reference system identified by the independent vari-
 185 ables $\boldsymbol{\xi} \equiv (\xi, \eta, \tau)$ in the unit cube $\hat{\mathcal{C}} = [0, 1]^3$. Inspired by [11], the governing equation is discretized using
 186 an efficient nodal formulation of space-time nodes given by a tensor product of Gauss-Legendre quadrature
 187 points along space and time directions. This choice defines an L^2 -orthogonal Lagrange basis used for the
 188 definition of the Galerkin solution. For our purposes, the single direction nodes over the unit interval $[0, 1]$
 189 are $\{(5 - \sqrt{5})/10; 1/2; (5 + \sqrt{5})/10\}$. Consequently, over a space-time cell there will be 27 Gauss-Legendre
 190 nodes $\hat{\boldsymbol{\xi}}_m$ and 27 Lagrange polynomial $\theta_l : \hat{\mathcal{C}} \rightarrow \mathbb{R}$ such that $\theta_l(\hat{\boldsymbol{\xi}}_m) = \delta_{lm}$ and $\int_{\hat{\mathcal{C}}} \theta_l \theta_m \, d\boldsymbol{\xi} = \delta_{lm} \|\theta_l\|_{L^2(\hat{\mathcal{C}})}^2$,
 191 with δ_{lm} the Kronecher's symbol. Let $\mathbf{m} : \{1, 2, 3\}^3 \rightarrow \{1, \dots, 27\}$ be a discrete map from a single direction
 192 index to the global three dimensional index defined as

$$193 \quad \mathbf{m}(i, j, k) = ij + (j - 1)(3 - i) + 9(k - 1),$$

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

$$198 \quad \hat{\boldsymbol{\xi}}_l = (\hat{\xi}_i, \hat{\eta}_j, \hat{\tau}_k); \quad \theta_l(\xi, \eta, \tau) = \theta_i^\xi(\xi) \theta_j^\eta(\eta) \theta_k^\tau(\tau),$$

199 with index $l = \mathbf{m}(i, j, k)$.

200 We want to solve the following problem: *find* $q : \mathcal{C}_i^n \rightarrow \mathbb{R}$ *such that*

$$201 \quad (3.5) \quad \begin{cases} \partial_t q + \nabla \cdot \mathbf{F}(q, \nabla q) = f & \text{in } \mathcal{C}_i^n \\ q|_{t=t^n} = w_i^n & \text{on } \Omega_i^n \end{cases},$$

202 which is problem (1.1) restricted to the space-time cell \mathcal{C}_i^n and redefined as a boundary value problem. We
 203 denote with q_h as the discretized solution of (3.5). In order to refer problem (3.5) to the reference domain
 204 $\hat{\mathcal{C}}$, we use a map $\mathcal{M}_i : \hat{\mathcal{C}} \rightarrow \mathcal{C}_i^n$

$$205 \quad (3.6) \quad \mathcal{M}_i : \begin{cases} x = x(\xi, \eta, \tau) \\ y = y(\xi, \eta, \tau) \\ t = t^n + \Delta t \tau \end{cases},$$

206 such that any space-time point $\mathbf{x} \equiv (x, y, t)$ in the physical space-time cell \mathcal{C}_i^n is a function $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi})$, with
 207 $\boldsymbol{\xi} \in \hat{\mathcal{C}}$ (see Figure 4). Time t is considered as a linear function of τ . From map (3.6), we define the Jacobian
 208 matrix J as

$$209 \quad (3.7) \quad J = \frac{d\mathbf{x}}{d\boldsymbol{\xi}} = \begin{bmatrix} x_\xi & x_\eta & x_\tau \\ y_\xi & y_\eta & y_\tau \\ 0 & 0 & \Delta t \end{bmatrix},$$

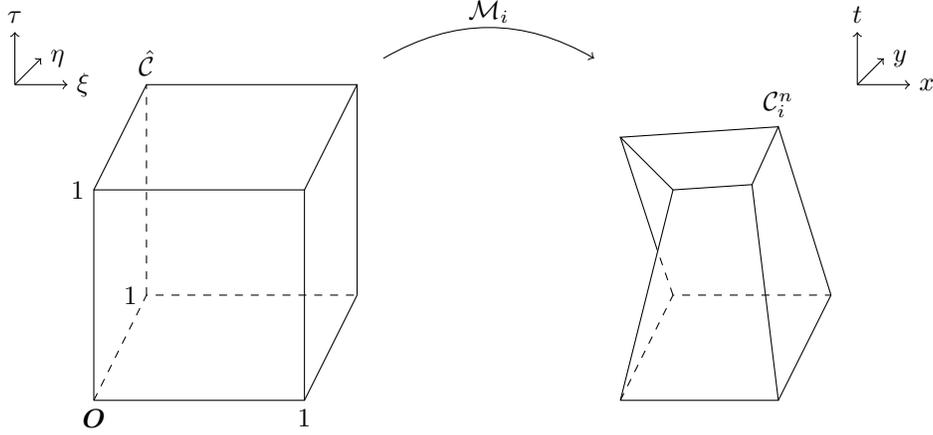


Fig. 4: Representation of the map \mathcal{M}_i from the reference space-time cell $\hat{\mathcal{C}}$ to the physical space-time cell \mathcal{C}_i^n .

210 whose inverse is

$$211 \quad (3.8) \quad J^{-1} = \frac{d\xi}{d\mathbf{x}} = \begin{bmatrix} \xi_x & \xi_y & \xi_t \\ \eta_x & \eta_y & \eta_t \\ 0 & 0 & 1/\Delta t \end{bmatrix}.$$

212 Moreover we call J_s^{-1} the restriction to the spatial coordinates of the inverse of the Jacobian matrix (3.8):

$$213 \quad (3.9) \quad J_s^{-1} = \begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix}.$$

214 Through (3.9), the problem in the reference domain reads

$$215 \quad (3.10) \quad \partial_\tau q + \Delta t \mathcal{F}^*(\hat{\nabla}q) + \Delta t J_s^{-T} \hat{\nabla} \cdot \mathcal{F}^{**}(q, \hat{\nabla}q) = \Delta t f,$$

216 where

$$217 \quad \partial_t u = \frac{\partial_\tau u}{\Delta t} + \mathcal{F}^*(\hat{\nabla}u); \quad \mathcal{F}^*(\hat{\nabla}q) = \begin{bmatrix} \xi_t \\ \eta_t \end{bmatrix} \cdot \hat{\nabla}q; \quad \mathcal{F}^{**}(q, \hat{\nabla}q) = \mathbf{F}(q, J_s^{-T} \hat{\nabla}q) = \begin{bmatrix} \mathcal{F}_\xi^{**} \\ \mathcal{F}_\eta^{**} \end{bmatrix}; \quad \hat{\nabla}q = \begin{bmatrix} \partial_\xi q \\ \partial_\eta q \end{bmatrix}.$$

218 The hat differential operators refer to reference space variables ξ and η in the reference space-time cell $\hat{\mathcal{C}}$. By
 219 abuse of notation and for sake of simplicity, we call all functions involved in both equations (3.5) and (3.10)
 220 with the same symbol (e.g., q and f) even though they take inputs in the physical space-time cell \mathcal{C}_i^n and in
 221 the reference space-time cell $\hat{\mathcal{C}}$, respectively. In order to weaken equation (3.10), for any function f and g in
 222 $L^2(\hat{\mathcal{C}})$ such that they are $L^2((0,1)^2)$ -integrable for any $\tau \in (0,1)$, we introduce the following notation:

$$223 \quad \langle f, g \rangle = \int_{\hat{\mathcal{C}}} fg \, d\xi; \quad [f, g]_\tau = \int_0^1 \int_0^1 f(\xi, \eta, \tau) g(\xi, \eta, \tau) \, d\xi \, d\eta.$$

224 Moreover, for our purposes, we define the following trial and test functional spaces:

$$225 \quad \mathcal{U} = \left\{ v \in H^1(\hat{\mathcal{C}}) : v|_{\tau=0} = w^n(x(\xi, \eta, 0), y(\xi, \eta, 0)) \wedge v \text{ is } L^2((0,1)^2)\text{-integrable } \forall \tau \in [0,1] \right\}$$

226 and

$$227 \quad \mathcal{V} = \left\{ v \in H^1(\hat{\mathcal{C}}) : \partial_\tau v \text{ is } L^2((0,1)^2)\text{-integrable } \forall \tau \in [0,1] \right\},$$

228 respectively. By multiplying left and right side of (3.10) by a generic test function $\theta \in \mathcal{V}$ and by integrating
 229 over the reference space-time cell $\hat{\mathcal{C}}$, the problem reads: *find* $u \in \mathcal{U}$ such that

$$230 \quad (3.11) \quad [\theta, q]_1 - \langle \partial_\tau \theta, q \rangle + \Delta t \langle \theta, \mathcal{F}^*(\hat{\nabla} q) \rangle + \Delta t \langle \theta, J_s^{-T} \hat{\nabla} \cdot \mathcal{F}^{**}(q, \hat{\nabla} q) \rangle = \Delta t \langle \theta, f \rangle + [\theta, w^n]_0 \quad \forall \theta \in \mathcal{V},$$

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

$$236 \quad (3.12) \quad \begin{aligned} &([\theta_k, \theta_l]_1 - \langle \partial_\tau \theta_k, \theta_l \rangle) \hat{q}_l + \Delta t \langle \theta_k, \theta_l \rangle \hat{\mathcal{F}}_l^* + \Delta t \langle \theta_k, (\xi_x \partial_\xi + \eta_x \partial_\eta) \theta_l \rangle \mathcal{F}_{\xi, l}^{**} \\ &+ \Delta t \langle \theta_k, (\xi_y \partial_\xi + \eta_y \partial_\eta) \theta_l \rangle \mathcal{F}_{\eta, l}^{**} = \Delta t \langle \theta_k, f \rangle + [\theta_k, w^n]_0, \end{aligned}$$

237 for any $k = 1, \dots, 27$.

238 In the left hand side of (3.12), we remark that the arising matrices have a sparse pattern due to the L^2 -
 239 orthogonality of the Lagrangian basis (e.g. the mass matrix by $\langle \theta_k, \theta_l \rangle$ is diagonal). Matrices involving the
 240 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
 241 before finding the map itself. On the contrary, the components which do not involve the map, namely
 242 $([\theta_k, \theta_l]_1 - \langle \partial_\tau \theta_k, \theta_l \rangle)$ and $\langle \theta_k, \theta_l \rangle$, can be pre-computed once for all before solving problem (3.12). Equation
 243 (3.12) is nonlinear due to the convective-diffusive terms \mathcal{F}^* and \mathcal{F}^{**} which depend on the solution q_h . For
 244 this reason a fixed point problem is solved: let r be the index of the fixed point iteration, therefore we solve
 245 q_h^{r+1}

$$246 \quad (3.13) \quad \begin{aligned} &([\theta_k, \theta_l]_1 - \langle \partial_\tau \theta_k, \theta_l \rangle) \hat{q}_l^{r+1} + \Delta t \langle \theta_k, \theta_l \rangle \hat{\mathcal{F}}_l^{*,r} + \Delta t \langle \theta_k, (\xi_x \partial_\xi + \eta_x \partial_\eta) \theta_l \rangle \mathcal{F}_{\xi, l}^{**,r} \\ &+ \Delta t \langle \theta_k, (\xi_y \partial_\xi + \eta_y \partial_\eta) \theta_l \rangle \mathcal{F}_{\eta, l}^{**,r} = \Delta t \langle \theta_k, f \rangle + [\theta_k, w^n]_0, \end{aligned}$$

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

249 **3.3. Recovery of the map and foreground mesh motion.** In the previous subsection, the local
 250 map $\mathcal{M}_i : \hat{\mathcal{C}} \rightarrow \mathcal{C}_i^n$ has been involved for the computation of the local weak predictor solution. Moreover,
 251 the foreground mesh of coordinates \mathbf{X} is moving accordingly to the following motion equation:

$$252 \quad (3.14) \quad \frac{d\mathbf{X}}{dt} = \mathbf{V},$$

253 where $\mathbf{V} = \mathbf{V}(\mathbf{x}, t; u)$ is the mesh velocity, eventually dependent on the solution. Equation (3.14) is closed
 254 with a Cauchy condition $\mathbf{X}(0) = \mathbf{X}_0$, which is the initial spatial configuration. Through equation (3.14),
 255 we recover the map \mathcal{M}_i for any cell at least on the foreground mesh. The motion equation (3.14) is solved
 256 through an *isoparametric* approach by locally referring it to the same reference system as done for the
 257 local equation (3.5). This means that the spatial coordinates \mathbf{X} are considered as function of the reference
 258 coordinates, i.e. $\mathbf{X}(\boldsymbol{\xi})$, with $\boldsymbol{\xi} \in \hat{\mathcal{C}}$. Finally, the solution of the referred motion equation is approximated via
 259 a Lagrangian expansion by employing the same Lagrangian basis $\{\theta_k\}_{k=1}^{27}$ built on the tensor combination
 260 of three Gauss-Legendre nodes in $(0, 1)$ along any direction as previously introduced: $\mathbf{X}_h = \theta_l \hat{\mathbf{X}}_l$, with
 261 $\hat{\mathbf{X}}_l = \mathbf{X}(\hat{\xi}_l)$. Thus, from time t^n to t^{n+1} , the motion equation (3.14) is locally re-written as

$$262 \quad (3.15) \quad \frac{d\mathbf{X}}{dt} = \mathbf{V} \text{ in } \mathcal{C}_i^n,$$

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

$$266 \quad (3.16) \quad ([\theta_k, \theta_l]_1 - \langle \partial_\tau \theta_k, \theta_l \rangle) \hat{\mathbf{X}}_l = \Delta t \langle \theta_k, \theta_l \rangle \hat{\mathbf{V}}_l + [\theta_k, \theta_l]_0 \hat{\mathbf{X}}_l^n,$$

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

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

$$272 \quad (3.17) \quad \begin{cases} x = x(\xi) = x_{i-1/2} + h_i^x \xi \\ y = y(\eta) = y_{i-1/2} + h_i^y \eta \end{cases},$$

273 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
 274 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
 275 along x and y of the cell, respectively, i.e. $h_i^x = x_{i+1/2} - x_{i-1/2}$ and $h_i^y = y_{i+1/2} - y_{i-1/2}$.

276 Since the mesh motion equation (3.14) is essentially solved via a sort of Discontinuous Galerkin (DG)
 277 approach, possible numerical (and non physical) discontinuities could arise. As a matter of fact, for a
 278 given vertex $\bar{\mathbf{X}}_k^{n+1}$ 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

279 different values of the vertex, namely $\{\bar{\mathbf{X}}_{k,i}^{n+1}\}_{i \in \mathcal{Z}_k^{n+1}}$, for any map \mathcal{M}_i referring to the cell \mathcal{C}_i^n to which Ω_i^{n+1}

280 belongs. The set \mathcal{Z}_k^{n+1} collects the index(es) of the cells sharing the vertex $\bar{\mathbf{X}}_k^{n+1}$. The cardinality N_k of set
 281 $\{\Omega_i^{n+1}\}_{i \in \mathcal{Z}_k^{n+1}}$, coinciding with the cardinality of the indexes set \mathcal{Z}_k^{n+1} , depends on the position of the vertex

282 $\bar{\mathbf{X}}_k^{n+1}$ on the foreground mesh: it is either 1 or 2 if the vertex is on the boundary of the mesh, otherwise it is

283 4. For this reason we consider a weighted average value for the shared vertex in order to tackle the possible
 284 arising discontinuities. As suggested in [3], we first consider a weighted velocity $\bar{\mathbf{V}}_k^{n+1}$ corresponding to the
 285 vertex $\bar{\mathbf{X}}_k^{n+1}$

$$286 \quad (3.18) \quad \bar{\mathbf{V}}_k^{n+1} = \frac{1}{N_k} \sum_{i \in \mathcal{Z}_k^{n+1}} \bar{\mathbf{V}}_{k,i}^{n+1}, \quad \text{with } \bar{\mathbf{V}}_{k,i}^{n+1} = \int_0^1 \theta_l(\xi^*, \eta^*, \tau) \, d\tau \hat{\mathbf{V}}_{l,i},$$

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

$$291 \quad (3.19) \quad \bar{\mathbf{X}}_k^{n+1} = \bar{\mathbf{X}}_k^n + \Delta t \bar{\mathbf{V}}_k^{n+1}.$$

292 We refer the reader to [6] for another definition of the weighted vertex velocities $\bar{\mathbf{V}}_k^{n+1}$ in (3.18) where the
 293 Voronoi neighborhood parameters of any vertex are exploited.

294 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 \mathcal{F}^{**} in the reference domain;
 - 7: Evolve the local predictor solution through (3.12);
-

295 **3.4. Correction stage: the finite volume scheme over the space-time cell.** Once the local
 296 predictor solution q_h is computed in each space-time cells \mathcal{C}_i^n , we can perform the correction stage. First,
 297 we rewrite the convective-diffusive equation (1.1) in divergence form. Let $\nabla_{\mathbf{x},t} = [\nabla, \partial_t]^T$ be the space-
 298 time differential operator and let $\mathbf{U} = [\mathbf{F}(u, \nabla u), u]^T$ be the space-time solution, thus problem (1.1) can be

299 rewritten as

$$300 \quad (3.20) \quad \nabla_{\mathbf{x},t} \cdot \mathbf{U} = f \quad \text{in } \Omega(t) \times [0, T].$$

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

$$303 \quad (3.21) \quad \partial\mathcal{C}_i^n = \Omega_i^n \cup \Omega_i^{n+1} \cup \bigcup_{j=1}^4 \Gamma_{ij}^n,$$

304 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
305 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
306 theorem to the left side, we obtain

$$307 \quad (3.22) \quad \int_{\partial\mathcal{C}_i^n} \mathbf{U} \cdot \mathbf{n}_{\mathbf{x},t} \, d\Gamma = \int_{\mathcal{C}_i^n} f \, d\mathcal{C},$$

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

$$310 \quad (3.23) \quad U_i^n = \frac{1}{|\Omega_i^n|} \int_{\Omega_i^n} u(x, y, t^n) \, dx \, dy,$$

311 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

$$312 \quad (3.24) \quad -|\Omega_i^n|U_i^n + |\Omega_i^{n+1}|U_i^{n+1} + \sum_{j=1}^4 \int_{\Gamma_{ij}^n} \mathbf{U} \cdot \mathbf{n}_{\mathbf{x},t} \, d\Gamma = \int_{\mathcal{C}_i^n} f \, d\mathcal{C},$$

313 where the unknown is the average solution U_i^{n+1} at time t^{n+1} , while the last term of the left hand side is the
314 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
315 that it is still exact. In order to solve (3.24), we need to approximate the integral function of the space-time
316 flux. Among the several methods proposed in the literature (such as in [6, 7, 8, 22, 11]), we here present a
317 Local Lax-Friederichs (LLF) approach:

$$318 \quad (3.25) \quad [\mathbf{U} \cdot \mathbf{n}_{\mathbf{x},t}]_{\Gamma_{ij}^n} \approx \Phi(q_j^+, q_j^-) = \frac{1}{2}(\mathbf{U}_j^+ + \mathbf{U}_j^-) \cdot \mathbf{n}_{\mathbf{x},t} - \frac{s}{2}(q_j^+ - q_j^-),$$

319 where $\mathbf{U}_j^+ = \mathbf{U}(q_j^+)$ and $\mathbf{U}_j^- = \mathbf{U}(q_j^-)$ are the space-time solution of (3.20) computed by solutions q_j^+ and
320 q_j^- , which represent the local predictor solutions outside and inside the cell, respectively, with respect to the
321 space-time side Γ_{ij}^n . The term s is the stabilization coefficient. Equation (3.24) with the flux approximation
322 (3.25) closes the correction stage of the ADER method. At the end of this stage, a solution u_i^{n+1} is found
323 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
324 Gauss nodes, a second order local polynomial interpolation is performed as explained in Section 3.1.

325 For the computation of the integrals along the space-time manifolds Γ_{ij}^n , we still use the previously computed
326 map \mathcal{M}_i . As a matter of fact, for a generic function $g : \mathcal{C}_i^n \rightarrow \mathbb{R}$ it holds:

$$327 \quad \int_{\Gamma_{ij}^n} g(\mathbf{x}) \, d\Gamma = \int_{\hat{\Gamma}_j} g(\mathbf{x}(\boldsymbol{\xi})) |\text{Cof}(J)\hat{\mathbf{n}}_j| \, d\hat{\Gamma},$$

328 where $\hat{\Gamma}_j$ is the j -th lateral side of the reference cubic domain $\hat{\mathcal{C}}$ of unit outer normal $\hat{\mathbf{n}}_j$, $\Gamma_{ij}^n = \mathcal{M}_i(\hat{\Gamma}_j)$ and
329 $\text{Cof}(J)$ is the cofactor matrix of the Jacobian tensor J of the map.

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

$$333 \quad (3.26) \quad \Delta t = \text{CFL} \frac{h}{\max\{\sup_{\Omega \times [0, T]} |a_x|, \sup_{\Omega \times [0, T]} |a_y|\}},$$

334 where h is the smallest characteristic length among all cells (both of background and foreground meshes)
 335 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
 336 Ω_i^n at discrete time t^n . Coefficient CFL in (3.26) is the Courant-Friedrichs-Lewy number. In this paper, the
 337 CFL coefficient is experimentally sought by conducting an empirical analysis in Section 5.2.

338 **3.5. Dynamics of the overlapping zone.** During the simulation, the foreground mesh moves and,
 339 consequently, the background mesh changes its configuration in the zone of the overlapping as well as in the
 340 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
 341 three possibilities:

- 342 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;
- 343 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;
- 344 3. The overlapping zone does not drastically change its configuration with respect to cell $\Omega_i(t)$, thus
 345 the cell is present at time t^n and it still continues to be present at time t^{n+1} .

346 The third case is trivial. For the first case, the predictor solution is executed in order to compute the fluxes
 347 of the neighboring cells even though the correction stage is not performed. For the second case, information
 348 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
 349 cells (those ones in the hole included). Consequently $i \leq N_1$. By recalling that the order of foreground cells
 350 starts from $N_1 + 1$, we look for an index $j > N_1$ such that

$$351 \quad (3.27) \quad \mathbf{x}_j = \arg \min_{k > N_1} \|\mathbf{x}_i - \mathbf{x}_k\|,$$

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

357 **4. The stabilization of the scheme.** For the definition of the coefficient s in (3.25), there are different
 358 approaches leading to different definitions. Here we analyse two stabilization coefficients, i.e. the advective-
 359 diffusive term s_{AD} and the just advective term s_A .

360 **4.1. The local advective-diffusive stabilization term.** For the definition of the coefficient s_{AD} in
 361 (3.25), we study a relaxed hyperbolic form of the parabolic equation (3.20). Let us consider the following
 362 relaxation by Cattaneo (we refer to [23] and its references for further details): let $0 < \varepsilon \ll 1$ be a relaxed
 363 time and consider variables v and w in $\Omega \times [0, T]$ such that

$$364 \quad (4.1) \quad \partial_t v = \frac{1}{\varepsilon}(\partial_x u - v); \quad \partial_t w = \frac{1}{\varepsilon}(\partial_y u - w).$$

365 Relations (4.1) define the relaxations in the sense that $\partial_x u \rightarrow v$ and $\partial_y u \rightarrow w$ in the limit of a vanishing
 366 ε . Since the flux has to be computed along the manifold Γ_{ij}^n in the space-time continuum, let us consider
 367 solution u and all its first derivatives as *stationary solutions* with respect to a pseudo-time $\mathfrak{t} \in \mathbb{R}_+$. Thus, let
 368 $\mathbf{u}(\mathfrak{t}; x, y, t) = [u, v, w]^T$ be the formal definition of the relaxed hyperbolic system with respect to pseudo-time
 369 \mathfrak{t} . It holds $\partial_{\mathfrak{t}} \mathbf{u} = 0$. The conservative form problem (3.20) in quasi-linear form is

$$370 \quad (4.2) \quad \partial_{\mathfrak{t}} \mathbf{u} + \partial_x(A\mathbf{u}) + \partial_y(B\mathbf{u}) + \partial_t(C\mathbf{u}) = \mathfrak{f} \quad \text{in } \mathbb{R}_+ \times \Omega(t) \times [0, T],$$

371 where A , B and C are 3×3 matrices (eventually involving the solution \mathbf{u} among their components if the
 372 original problem is nonlinear) and the force term $\mathfrak{f} = [f, -v/\varepsilon, -w/\varepsilon]^T$. In particular, A and B always
 373 depend on the relaxation time ε and they are defined by the convection-diffusion term $\mathcal{F}(u, \nabla u)$ and C is
 374 always the identity matrix if the Cattaneo's relaxation (4.1) is employed. In order to study the differential
 375 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
 376 (4.2) helps in treating the real time variable t as any other spatial variable x and y . When the force term in
 377 (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
 378 of real values n_x , n_y and n_t . If the hyperbolicity is ensured, the relaxed hyperbolic system has a planar
 379 wave solution propagating in the space-time continuum $\Omega \times [0, T]$. In particular, if $\mathbf{n}_{\mathbf{x},t} = [n_x, n_y, n_t]^T$ is a
 380 particular direction in the space-time continuum, the eigenvalues of \mathcal{A} define the speeds of propagation of the

381 solution along the principal directions defined by the eigenvectors of \mathcal{A} . For this reason, in the perspective
 382 of an upwind stabilization, the local stabilization term s_{AD} in (3.25) is equal to the maximum speed of
 383 propagation of the wave, as it happens for the LLF flux approximation for a generic hyperbolic problem of
 384 a propagating wave in the space continuum.

385 Here we detail the previous analysis for the convection-diffusion problem with the convective field $\mathbf{a} =$
 386 $[a_x, a_y]^T$ and the diffusive term ν depending on space \mathbf{x} and time t and eventually the solution u itself if a
 387 non-linearity leads the dynamics of the equation. In this case, the matrices of the quasi-linear problem (4.2)
 388 read

$$389 \quad A = \begin{bmatrix} a_x & -\nu & 0 \\ -1/\varepsilon & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} a_y & 0 & -\nu \\ 0 & 0 & 0 \\ -1/\varepsilon & 0 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

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

$$391 \quad (4.3) \quad \rho(\mathcal{A}) = \left\{ n_t; \frac{1}{2} \left[\sigma \pm \sqrt{\left(a_x^2 + \frac{4\nu}{\varepsilon} \right) n_x^2 + 2a_x a_y n_x n_y + \left(a_y^2 + \frac{4\nu}{\varepsilon} \right) n_y^2} \right] \right\},$$

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

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

$$397 \quad (4.4) \quad s_{AD} = \max |\rho(\mathcal{A})| = \frac{1}{2} \left| \sigma + \sqrt{\left(a_x^2 + \frac{4\nu}{\varepsilon} \right) n_x^2 + 2a_x a_y n_x n_y + \left(a_y^2 + \frac{4\nu}{\varepsilon} \right) n_y^2} \right|.$$

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

400 **4.2. The choice of the relaxation time.** For the definition of the advective-diffusive stabilization
 401 term s_{AD} , we considered the relaxed hyperbolic system (4.2) deriving from the parabolic problem (3.20)
 402 through a relaxation time ε . If we were to solve the relaxed problem instead of the original one, the
 403 approximate solution would differ from the exact solution of two errors that are added together: the numerical
 404 error (typical of the scheme) and a relaxation error. For a linear problem, these errors have been investigated
 405 by Montecinos and Toro in [23]. The error $|u_{\text{hip}} - u|$ between the hyperbolized solution u_{hip} and the original
 406 solution u is $\mathcal{O}(\varepsilon)$ [17]. Thus, if $u_{\text{hip},h}$ is a numerical approximation of the exact relaxation solution u_{hip} ,
 407 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
 408 maximum characteristic length of cells $\Omega_i(t)$'s. However, the goal is to choose a relaxation time ε such that
 409 the relaxation error is always dominated by or, at least, comparable to the numerical error, i.e. $\mathcal{O}(\varepsilon) \lesssim \mathcal{O}(h_0^p)$.
 410 The following theoretical result can help in fulfilling our task.

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

$$414 \quad (4.5) \quad C_p \frac{\varepsilon}{h_0^p} = \mathcal{O}(1),$$

415 *with*

$$416 \quad C_p = \frac{1 - 2^{-\frac{1}{2}}}{2^{p-\frac{1}{2}} - 1}.$$

417 For the proof of Proposition 4.2, we refer the reader to Section 2.4.1 of [16]. As a consequence, there is the
 418 following corollary.

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

$$422 \quad (4.6) \quad \varepsilon_p = \frac{\mathcal{O}(1)h_0^p}{C_p}.$$

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

427 **4.3. The local advective stabilization term.** In order to recover a stabilization term s_A by only
 428 considering the first order operator involved in the whole differential operator of the original problem, we
 429 can treat the equation to stabilize as a pure hyperbolic (namely just advective) problem. For this reason,
 430 the advective stabilization term s_A coincides with the maximum eigenvalue of the ALE Jacobian matrix
 431 in a spatial normal direction by excluding the diffusive component which acts on the diffusion from the
 432 advective-diffusive term $\mathbf{F}(u, \nabla u)$ [6]. This matrix reads

$$433 \quad (4.7) \quad A_{\tilde{\mathbf{n}}}^{\mathbf{V}} = \sqrt{n_x^2 + n_y^2} \left[\frac{\partial \mathbf{F}}{\partial u} \tilde{\mathbf{n}} - \mathbf{V} \cdot \tilde{\mathbf{n}} I \right],$$

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

$$437 \quad \tilde{\mathbf{n}} = \frac{[n_x, n_y]^T}{\sqrt{n_x^2 + n_y^2}}.$$

438 By recalling that the recovered map \mathcal{M}_i is defined over $\bar{\mathcal{C}}$ with image in $\bar{\mathcal{C}}_i$, the space-time manifold Γ_{ij}^n ,
 439 $j = 1, \dots, 4$, of the space-time cell \mathcal{C}_i can be described by only two of the three reference space-time variables
 440 (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
 441 to 0 or 1, depending on the specific j -th space-time manifold Γ_{ij}^n . Let χ be the free variable (e.g. $\chi = \xi$)
 442 and $\bar{\kappa}$ be the constrained variable (e.g. $\bar{\kappa} = \bar{\eta}$) for the specific manifold Γ_{ij}^n . Therefore, for a specific point
 443 $\tilde{\mathbf{x}}$ over Γ_{ij}^n it is possible to distinguish two directional vectors provided by the map \mathcal{M}_i

$$444 \quad \mathbf{r}_\chi = \begin{bmatrix} x_\chi \\ y_\chi \\ 0 \end{bmatrix}_{\bar{\kappa}} \quad \text{and} \quad \mathbf{r}_\tau = \begin{bmatrix} x_\tau \\ y_\tau \\ \Delta t \end{bmatrix}_{\bar{\kappa}}.$$

445 The definitions of the directional vectors \mathbf{r}_χ and \mathbf{r}_τ allow to explicitly write the physical normal vector $\mathbf{n}_{\mathbf{x},t}$
 446 on $\tilde{\mathbf{x}}$ as

$$447 \quad \mathbf{n}_{\mathbf{x},t} = \frac{\mathbf{r}_\chi \wedge \mathbf{r}_\tau}{|\mathbf{r}_\chi \wedge \mathbf{r}_\tau|_{\bar{\kappa}}} = \frac{[\Delta t y_\chi, -\Delta t x_\chi, d_{\chi\tau}]^T}{\sqrt{\Delta t^2 y_\chi^2 + \Delta t^2 x_\chi^2 + d_{\chi\tau}^2}_{\bar{\kappa}}},$$

448 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
 449 unit vector $\tilde{\mathbf{n}}$ along the spatial directions and the velocity of the point as

$$450 \quad \tilde{\mathbf{n}} = \frac{[y_\chi, -x_\chi]^T}{\sqrt{y_\chi^2 + x_\chi^2}} \quad \text{and} \quad \mathbf{V} = \frac{d\tilde{\mathbf{x}}}{dt} = \frac{[x_\tau, y_\tau]^T}{\Delta t}.$$

451 Consequently it holds

$$452 \quad (4.8) \quad \mathbf{V} \cdot \tilde{\mathbf{n}} = \frac{-d_{\chi\tau}}{\Delta t \sqrt{y_\chi^2 + x_\chi^2}} = \frac{-n_t}{\sqrt{n_x^2 + n_y^2}}.$$

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

$$454 \quad (4.9) \quad s_A = |a_x n_x + a_y n_y + n_t|.$$

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

457 PROPOSITION 4.4. For linear problem (1.1), let the diffusion parameter ν go to zero, therefore the fol-
458 lowing limit holds

$$459 \quad (4.10) \quad \lim_{\nu \rightarrow 0} s_{AD} = \frac{1}{2} |\sigma + a_x n_x + a_y n_y| = |a_x n_x + a_y n_y + n_t| = s_A.$$

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

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

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

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

478 In *TEST1*, the foreground mesh is subjected to a deformation and rotation around its center of mass. We
479 remark that in this case that the deformation velocity depends on the solution; in *TEST2*, the hyperbolic
480 tangent in the exact solution describes a composed Gaussian bell whose maximum is originally located in
481 the position $\mathbf{x} = (-1, 0)$ and, after a time $T = \pi$, it computes a counterclockwise half rotation up to position
482 $\mathbf{x} = (1, 0)$ along the circumference of unit radius and centered in the origin of the axes. Due to the particular
483 dynamics of the solution, we set a foreground mesh following the movement of the Gaussian bell. At the
484 initial time, the foreground and background meshes in both 2D cases consist of squared cells whose sides
485 have a length equal to h .

486 For all numerical tests, the time step Δt is set accordingly to (3.26) with CFL coefficient equal to 0.4. The
487 reason of this value will be better explained in Section 5.2 where an empirical stability analysis is conducted.

488 **5.1. Order of convergence.** In this section we have a double goal. On one hand we want to nu-
489 merically prove that the presented method is second order when an advective-diffusive LLF stabilization
490 s_{AD} is employed. On the other hand, we want to compare this stabilization term with the local advective
491 stabilization flux s_A . The study of the second order convergence is conducted on all test cases of Table 1.
492 Finally, on the two mentioned 2D test cases the comparison of the performances for the flux approximations
493 is carried out.

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

$$496 \quad (5.1) \quad L^\infty\text{-err} = \|u - u_{ex}\|_{L^\infty(\Omega)} = \operatorname{ess\,sup}_{\mathbf{x} \in \Omega} |u(\mathbf{x}, T) - u_{ex}(\mathbf{x}, T)| \approx e_{L^\infty}^N = \max_{k=1, \dots, N} |u_k^M - u_{ex}(\mathbf{x}_k, T)|$$

497 and

$$498 \quad (5.2) \quad L^2\text{-err} = \|u - u_{ex}\|_{L^2(\Omega)} = \sqrt{\int_{\Omega} (u(\mathbf{x}, T) - u_{ex}(\mathbf{x}, T))^2 \, d\Omega} \approx e_{L^2}^N = \sqrt{\frac{\sum_{k=1}^N (u_k^M - u_{ex}(\mathbf{x}_k, T))^2}{N}},$$

499 respectively, where $N \approx h^{-1/d}$ is the total number of cells of background and foreground partitions at time
500 T (with h the characteristic length of cells and $d = \dim(\Omega)$) and M is the maximum natural such that
501 $T = M\Delta t$, while the convergence rate reads

$$502 \quad (5.3) \quad L^p\text{-rate} = d \frac{\log(e_{L^p}^{N_1}/e_{L^p}^{N_2})}{\log(N_2/N_1)}, \quad \text{for } p = 2, \infty,$$

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 \mathbf{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	
	<i>test1</i>	<i>test2</i>	<i>TEST1</i>	<i>TEST2</i>
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: $u_{ex}(x, y, t) _{\partial\Omega}$	Dirichlet: $u_{ex}(x, y, t) _{\partial\Omega}$
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	$[-0.5, 0.25]$	$[-0.25, 0.25]$	$[-0.5, 0.5]^2$	$[-1.25, -0.75] \times [-0.25, 0.25]$
\mathbf{V}	$2.5 + e^x$	$e^x \chi_{\{t \leq 0.25\}}(t) + \chi_{\{t > 0.25\}}(t)$	$[u(x, y, t) - y, u(x, y, t) + x]^T$	$[-y, x]^T$

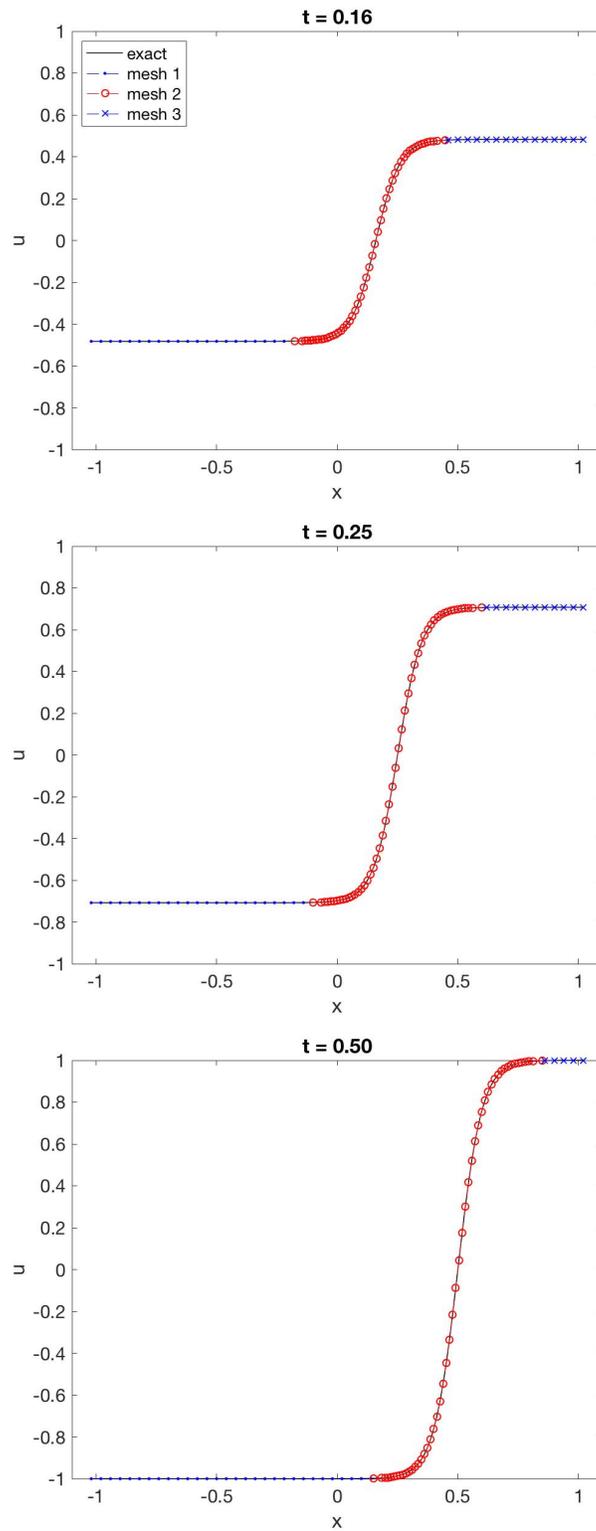


Fig. 5: Three time instants for the 1D test case *test2*. The circle markers define the nodes of the moving foreground mesh. The remaining dot and x markers are the nodes of the two background meshes.

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

	T	h	L^∞ -err	L^2 -err	L^∞ -rate	L^2 -rate
<i>test1</i>	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
<i>test2</i>	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
<i>TEST1</i>	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
<i>TEST2</i>	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

503 for two different partition settings whose number of cells are N_1 and N_2 , respectively, with $N_1 < N_2$.
504 Table 2 sums up the convergence analysis for 1D test cases. In the last two columns there are the rates of
505 convergence of the errors for both L^∞ and L^2 errors. From the analysis, the second order of the method is
506 confirmed.

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

515 **5.2. Empirical analysis of stability condition.** As already mentioned at the end of Section 3.4, the
516 presence of a weak solution, found in the prediction step of the presented method and successively plugged
517 into the flux of the finite volume scheme in the correction stage, makes a classical stability analysis not
518 straightforward to be made. For this reason, we performed an empirical stability analysis by assuming that
519 the right time step Δt allowing a stable computation is defined as in (3.26).

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

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	<i>TEST1</i>				<i>TEST2</i>			
	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

525 The analysis is conducted on the 2D test cases presented in Table 1. In Figure 6 there are three time instants
526 of both test cases.

527 In Table 4 there are the maximum CFL numbers and related maximum time steps Δt such that the method is
528 stable. The time step Δt is computed by formula (3.26). By comparing the performances of a local advective
529 (A) stabilization term against the same ones using a local advective-diffusive (AD) stabilization term, it is
530 evident that an advective LLF flux always needs a smaller CFL with respect to an advective-diffusive LLF
531 flux in order to stabilise the routine.

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

<i>TEST3</i>	
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	$[0.8, 1.2]^2$
\mathbf{V}	$P1, P2, P3$

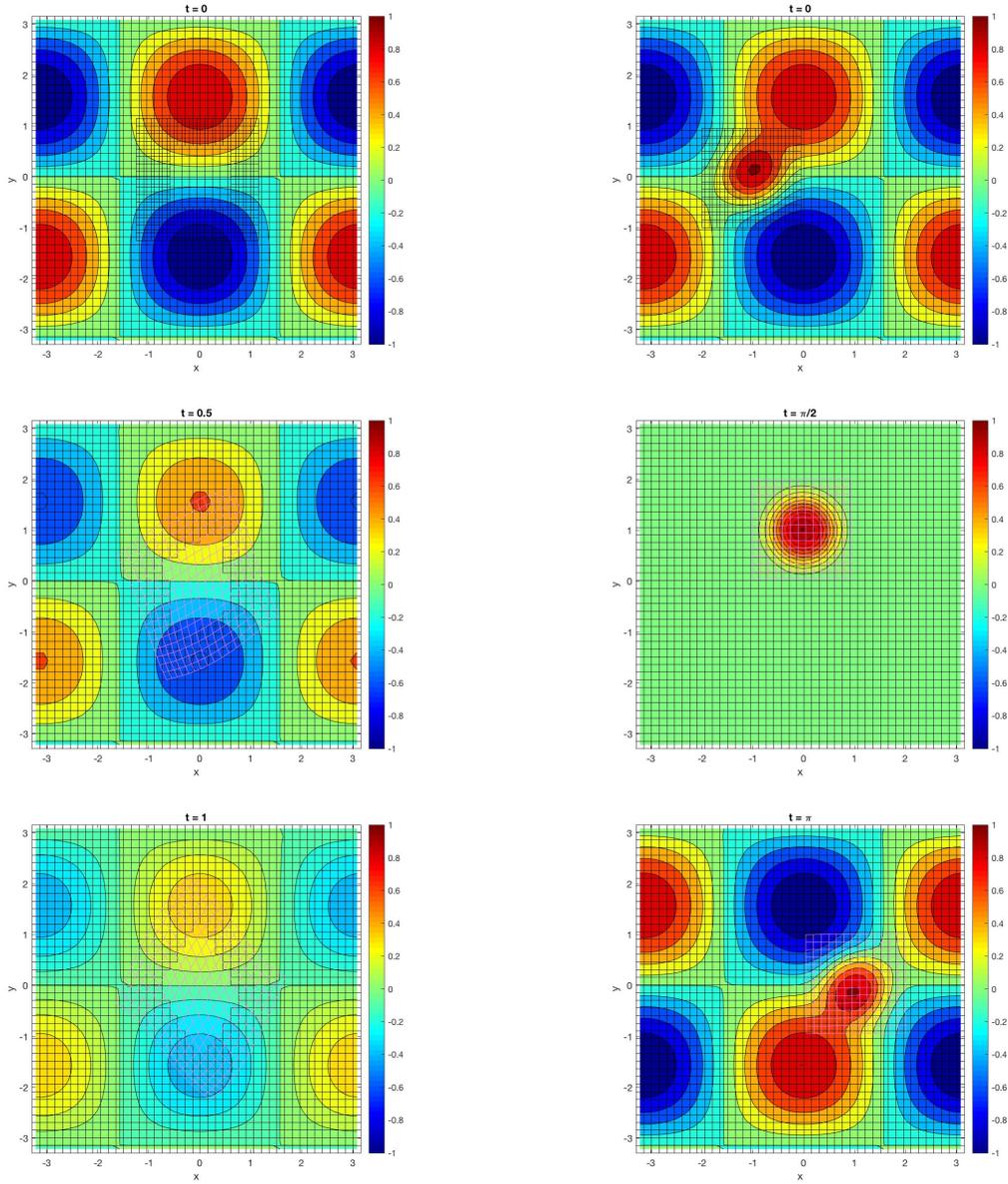
\mathbf{V}

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.

532 **5.3. Relationship between the convective field and the foreground mesh velocity.** From the
533 theoretical explanation of the method, it does not emerge in any way an interaction between the speed of the
534 foreground grid \mathbf{V} and the intrinsic advective field \mathbf{a} of the problem. In other words, there does not seem to
535 be a limitation of the velocity of the mesh that is displaced and deformed in terms of stability of the method.

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

536 The unique limitation of the mesh speed (see section 3.5) is due to the CFL condition with respect to the
 537 dimension of the single cell. In order to allow to the code to perform the automatic information transmission,
 538 the mesh speed is such that it does not allow a given fringe cell Ω_i^n in the foreground mesh to migrate beyond
 539 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
 540 fact, if this process is not ensured, for those new born cells belonging to the background mesh at time t^{n+1}
 541 could not be able to recover the information from the polynomial interpolation. Consequently, the algorithm
 542 would incur a loss of information.

543 In this subsection we test on a numerical case that the stability is not given by a relation between velocity
 544 mesh \mathbf{V} and advective field \mathbf{a} . In particular, on the same linear test case, we will consider different possible

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
<i>P1</i>	2.1554e-2	6.8500e-3
<i>P2</i>	2.1554e-2	4.8809e-3
<i>P3</i>	4.8809e-2	1.0864e-2

545 movements of the foreground mesh by measuring, at final time $t = T$, the L^∞ - and L^2 -errors of the mismatch
 546 between the exact and the numerical solution. The tested case is named *TEST3* and it is summed up in
 547 Table 5 (top).

548 The foreground mesh is either allowed not to move or to rigidly move in the parallel direction with respect to
 549 the abscissae axis. In particular, we consider three possibilities of movements, *P1*, *P2* and *P3*, reported and
 550 explained in Table 5 (bottom). We remark that test *P1* corresponds to a test case with a unique block mesh
 551 due to the position and the uniformity of the foreground mesh with respect to the background mesh. For
 552 this reason, tests *P2* and *P3* are compared with *P1*. In Figure 7 there are both the numerical solutions and
 553 the associated pointwise absolute values of the difference between the exact and numerical solution for the
 554 final time $T = 2$ for the configurations listed above. In particular, the configuration of the foreground mesh
 555 in Figure 7a (left) corresponds to the initial mesh configuration for tests *P2* and *P3* too. By visualising the
 556 different plots of the errors, it is evident the movement of the foreground mesh introduces an error. As a
 557 matter of fact the errors of *P2* and *P3* are neither equal each other nor to the errors of *P1*. The quantitative
 558 differences among the different cases are reported in Table 6. Concerning test *P2*, the L^∞ -error is equal to
 559 the one of *P1*, even though the L^2 -error is the double. This distance between a steady and moving foreground
 560 mesh becomes slightly more evident at increasing of the mesh speed, as the last line of Table 6 shows. In
 561 any case, all the errors are comparable and this confirms that there is no relation between advective field
 562 and mesh velocity in terms of stability. The mesh velocity seems to affect the numerical solution only on the
 563 precision.

564 We conclude this subsection by analysing the loss of information given by a very strong speed of the fore-
 565 ground mesh on the same test case. The foreground mesh is still located in the subset $[0.8, 1.2]^2$ at the
 566 initial time and moves rightwards with a speed equal to 4. This velocity, with the considered time step Δt ,
 567 allows to the cells on the left side of the foreground mesh to overflow from the borders of their stencil from
 568 times t^n to t^{n+1} . In Figure 8 there is a comparison between the recovered numerical solution and the exact
 569 solution for $t = 0.84$ (which corresponds to that time when the right side of the moving mesh is fully aligned
 570 to the right side of the channel). There is no relation between the two solutions because the speed of the
 571 foreground mesh is so fast that it does not allow the algorithm to assign the correct information about the
 572 background cells that arise in the wake of the foreground mesh itself.

573 **5.4. Further topics.** We conclude this section by presenting three test cases that show the potentiality
 574 of the method. Firstly, a *nonlinear advection-diffusion system* is solved; successively a *multimesh* setting of
 575 grids is considered for the already described *TEST2* (see Table 1); finally, we consider a test case with a
 576 *complex domain* in which the foreground mesh is employed in order to adapt its shape to the shape of the
 577 domain.

578 **5.4.1. Nonlinear system.** Let $\Omega = [-\pi, \pi]^2$ and $T = 0.5$ be the computational domain and the final
 579 time, respectively. Thus the problem is: *find* $\mathbf{u} : \Omega \times [0, T] \rightarrow \mathbb{R}^2$ *such that*:

$$580 \quad (5.4) \quad \begin{cases} \partial_t \mathbf{u} + \nabla \cdot (\mathbf{u} \mathbf{u}^T) = \nu \Delta \mathbf{u} + \mathbf{f} & \text{in } \Omega \times [0, T] \\ \mathbf{u} \equiv \mathbf{u}_{ex} & \text{on } \partial\Omega \times [0, T], \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_{ex}(\mathbf{x}, 0) & \text{in } \bar{\Omega} \times \{0\} \end{cases},$$

581 where the force term \mathbf{f} is chosen to have the exact solution

$$582 \quad \mathbf{u}_{ex}(x, y, t) = e^{-t} \begin{bmatrix} \cos(x) \sin(y) \\ -\sin(x) \cos(y) \end{bmatrix}.$$

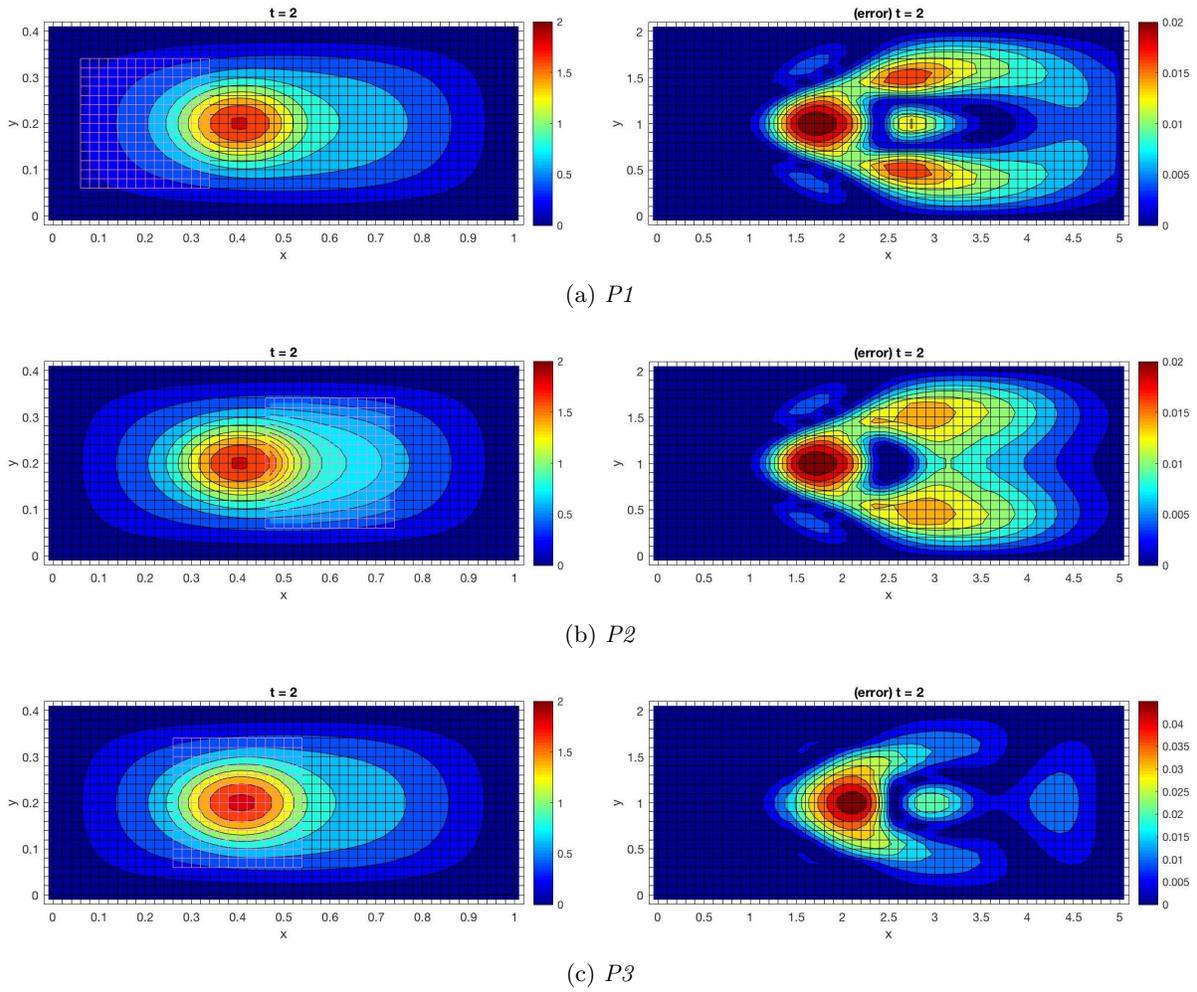


Fig. 7: The numerical solutions, on the left, at final time $t = 2$ of the three possibilities $P1$, $P2$ and $P3$ of foreground mesh movements for the $TEST3$. On the right there are the associated pointwise errors of the mismatch between the exact solution and the numerical solution.

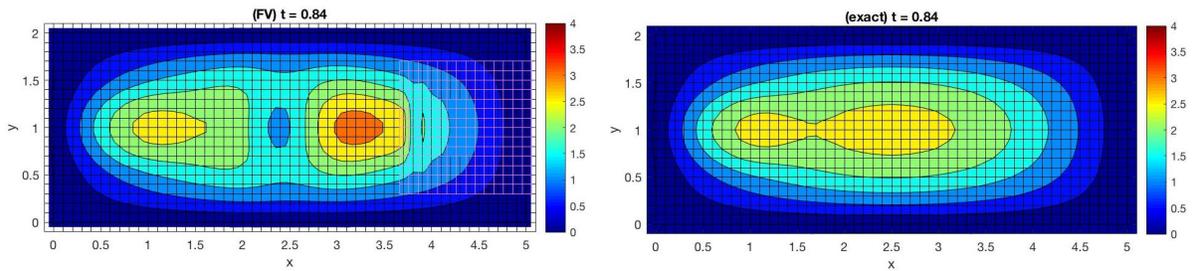
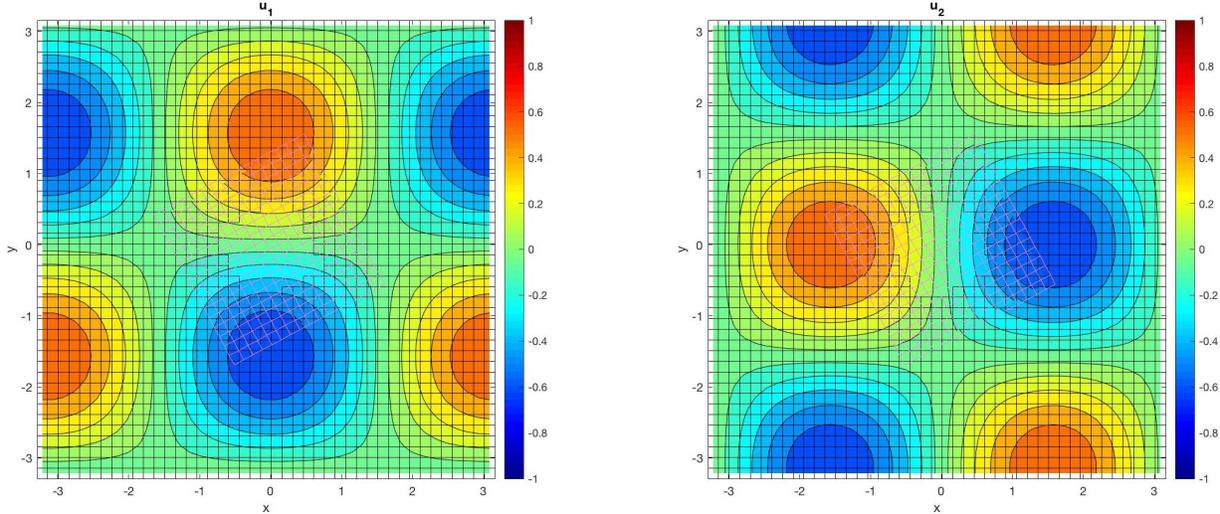


Fig. 8: Comparison between the numerical (left) and exact (right) solution of $TEST3$ at time $t = 0.84$ for a moving foreground mesh travelling with a speed generating a loss of information.

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

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

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

595 **5.4.2. Multimesh setting.** The presented method can be easily extended to more than one foreground
596 mesh. As a matter of fact, different meshes can be set with an independent movement and such that to
597 exchange information with the background grid and with the other moving foreground meshes. Due to the
598 possibility to move, the foreground meshes can overlap each other. Consequently, the hole will be present in
599 the background as well as in some foreground grids by properly applying the same dynamics of the overlap-
600 ping zone of Section 3.5 to the specific intermediate foreground mesh.

601 In order to compare the performances of multimesh setting with two moving foreground meshes, we consid-
602 ered the presented case *TEST2* with a foreground mesh clockwise rotating around the origin (see Table 1) by
603 adding a second foreground mesh. The new grid is originally located to subset $[-0.78, -0.18] \times [-0.62, -0.02]$
604 and horizontally moves on the right with a constant velocity $\mathbf{V}_2 = [-0.8, 0]^T$ (see Figure 10). The new grid

Table 8: Summary scheme of *TEST4*.

<i>TEST4</i>	
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	$B(0.5, 1.5; 0)$
\mathbf{V}	$[0, -2]^T$

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

614 **5.4.3. Complex domains.** An important application of chimera grids is the possibility to use meshes
615 fitting the particular shape of the domain (which eventually evolves in time) by preserving a Cartesian
616 background mesh. Here we present a test case summed up in Table 8. For any positive time t , let the generic
617 moving ball formally be

$$618 \quad B(\rho_{\min}, \rho_{\max}; t) = \{(x, y) \in \mathbb{R}^2 : x = \rho \cos(\theta), y = \rho \sin(\theta) - 2t - \pi; \text{ with } (\rho, \theta) \in [\rho_{\min}, \rho_{\max}] \times [0, 2\pi]\}.$$

619 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
620 radius equal to 0.5 is subtracted at any time $t \in [0, T]$. The circle vertically moves downwards with a
621 constant velocity. In Figure 11a it is reported the numerical solution at the initial and final time instants
622 for the numerical test. In Figure 11b there is a focus on the grid settings. For the foreground mesh, a polar
623 structured grids is employed. It fits the shape of the domain and moves as the domain evolves.

624 **6. Conclusions.** We presented a second order finite volume scheme for unsteady advection-diffusion
625 PDEs on overset grid. The scheme is based on an extension of the ADER method to advection-diffusion
626 equations with compact data transmission conditions from the background to the foreground meshes and
627 *vice versa*. We also introduced a new stabilization term for approximating the fluxes through a Local Lax-
628 Friederichs approach.

629 The numerical illustrations for linear and non-linear systems show that background and foreground
630 moving meshes do not introduce spurious perturbation to the solution, uniformly reaching second order
631 accuracy in space and time. In addition, we showed that the speed of the foreground mesh does not
632 influence the stability of the method. Our results also show that the new LLF stabilization speed improves
633 the precision and robustness of the numerical solution and allows a less restrictive CFL condition. Finally,
634 it is shown that several foreground meshes, possibly overlapping and with independent displacements, can
635 seamlessly be employed thanks to this approach.

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

638 **Acknowledgments.** The authors thank Haysam Telib of Optimad Engineering (Turin, Italy) for his
639 industrial view point in support of the whole work. The authors acknowledge the support by European
640 Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Actions, grant

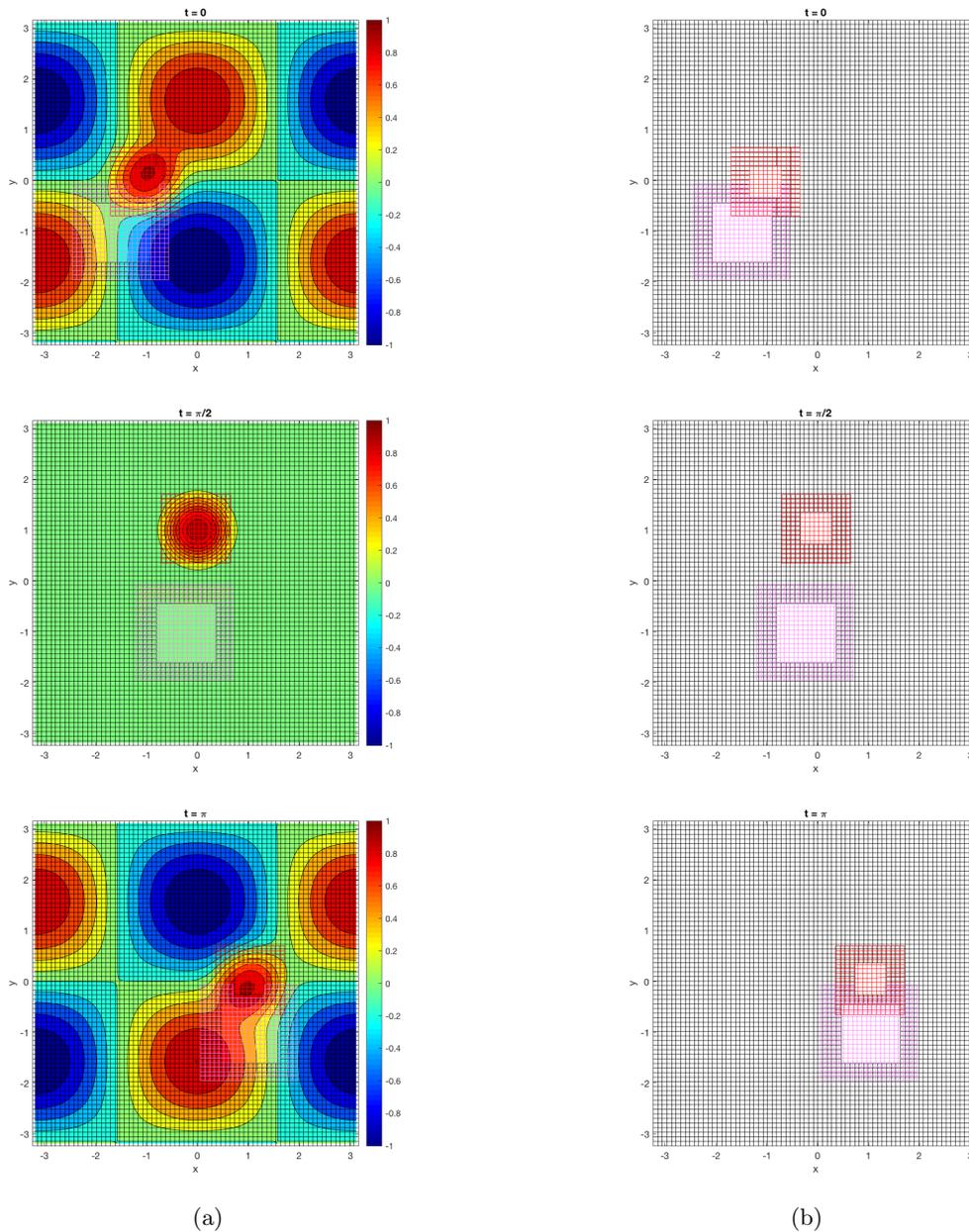


Fig. 10: On the left (a), the solution of *TEST2* for three time instants with a multimesh setting composed of two foreground meshes; on the right (b), for the same time instants, the configuration of the background and foreground grids.

641 agreement 872442 (ARIA). MGC is pleased to thank the company Optimad Engineering for hosting him
 642 during part of the research in Turin.

643

REFERENCES

644 [1] P. ANGOT, C.-H. BRUNEAU, AND P. FABRIE, *A penalization method to take into account obstacles in incompressible*
 645 *viscous flows*, *Numerische Mathematik*, 81 (1999), pp. 497–520.

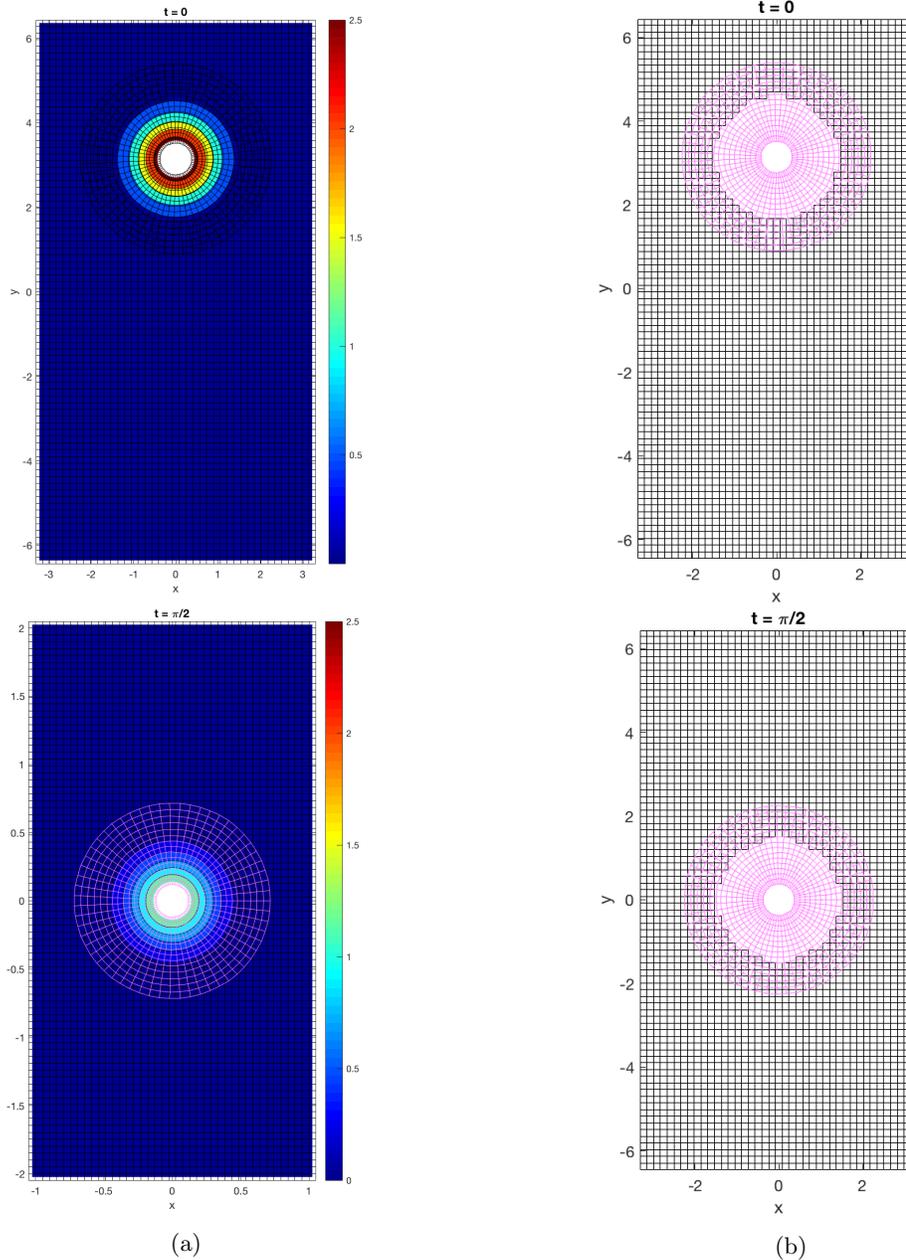


Fig. 11: On the left (a), the solution of $TEST_4$ for the initial and final time instants; on the right (b), for the same time instants, the background and foreground grids setting.

- 646 [2] J. BENEK, P. BUNING, AND J. STEGER, *A 3-d chimera grid embedding technique*, in 7th Computational Physics Conference,
647 1985, p. 1523.
- 648 [3] W. BOSCHERI AND M. DUMBSER, *Arbitrary-lagrangian-eulerian one-step weno finite volume schemes on unstructured*
649 *triangular meshes*, Communications in Computational Physics, 14 (2013), pp. 1174–1206.
- 650 [4] C. E. CASTRO AND E. F. TORO, *Solvers for the high-order riemann problem for hyperbolic balance laws*, Journal of
651 Computational Physics, 227 (2008), pp. 2481–2513.
- 652 [5] I.-T. CHIU AND R. MEAKIN, *On automating domain connectivity for overset grids*, in 33rd Aerospace Sciences Meeting
653 and Exhibit, 1995, p. 854.
- 654 [6] M. DUMBSER, W. BOSCHERI, M. SEMPLICE, AND G. RUSSO, *Central weighted eno schemes for hyperbolic conservation*
655 *laws on fixed and moving unstructured meshes*, SIAM Journal on Scientific Computing, 39 (2017), pp. A2564–A2591.

- 656 [7] M. DUMBSER, C. ENAUX, AND E. F. TORO, *Finite volume schemes of very high order of accuracy for stiff hyperbolic*
657 *balance laws*, Journal of Computational Physics, 227 (2008), pp. 3971–4001.
- 658 [8] M. DUMBSER AND M. KÄSER, *Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for*
659 *linear hyperbolic systems*, Journal of Computational Physics, 221 (2007), pp. 693–723.
- 660 [9] R. GLOWINSKI, T.-W. PAN, AND J. PERIAUX, *A fictitious domain method for dirichlet problem and applications*, Computer
661 *Methods in Applied Mechanics and Engineering*, 111 (1994), pp. 283–303.
- 662 [10] J. GUERRERO, *Overset composite grids for the simulation of complex moving geometries*, DICAT, University of Genoa,
663 Italy, (2006).
- 664 [11] A. HIDALGO AND M. DUMBSER, *Ader schemes for nonlinear systems of stiff advection–diffusion–reaction equations*, Jour-
665 *nal of Scientific Computing*, 48 (2011), pp. 173–189.
- 666 [12] C. W. HIRT, A. A. AMSDEN, AND J. COOK, *An arbitrary lagrangian-eulerian computing method for all flow speeds*, Journal
667 *of computational physics*, 14 (1974), pp. 227–253.
- 668 [13] G. HOUZEAUX, J. CAJAS, M. DISCACCIATI, B. EGUZKITZA, A. GARGALLO-PEIRÓ, M. RIVERO, AND M. VÁZQUEZ, *Domain*
669 *decomposition methods for domain composition purpose: chimera, overset, gluing and sliding mesh methods*, Archives
670 *of Computational Methods in Engineering*, 24 (2017), pp. 1033–1070.
- 671 [14] M.-S. LIOU AND Y. ZHENG, *A novel approach of three-dimensional hybrid grid methodology: Part 2. flow solution*,
672 *Computer methods in applied mechanics and engineering*, 192 (2003), pp. 4173–4193.
- 673 [15] R. MEAKIN, *Chapter 11: Composite overset structured grids*, Handbook of grid generation, JF Thompson, BK Soni and
674 NP Weatherill. CRC, (1999).
- 675 [16] G. I. MONTECINOS, *Numerical methods for advection-diffusion-reaction equations and medical applications*, PhD thesis,
676 University of Trento, 2014.
- 677 [17] G. B. NAGY, O. E. ORTIZ, AND O. A. REULA, *The behavior of hyperbolic heat equations’ solutions near their parabolic*
678 *limits*, Journal of Mathematical Physics, 35 (1994), pp. 4334–4356.
- 679 [18] C. S. PESKIN, *The immersed boundary method*, Acta numerica, 11 (2002), pp. 479–517.
- 680 [19] N. A. PETERSSON, *Hole-cutting for three-dimensional overlapping grids*, SIAM Journal on Scientific Computing, 21 (1999),
681 pp. 646–665.
- 682 [20] A. RAEI, M. BERGMANN, AND A. IOLLO, *A finite-difference method for the variable coefficient poisson equation on*
683 *hierarchical cartesian meshes*, Journal of computational Physics, 355 (2018), pp. 59–77.
- 684 [21] V. A. TITAREV AND E. F. TORO, *Ader: Arbitrary high order godunov approach*, Journal of Scientific Computing, 17
685 (2002), pp. 609–618.
- 686 [22] V. A. TITAREV AND E. F. TORO, *Ader schemes for three-dimensional non-linear hyperbolic systems*, Journal of Compu-
687 *tational Physics*, 204 (2005), pp. 715–736.
- 688 [23] E. F. TORO AND G. I. MONTECINOS, *Advection-diffusion-reaction equations: hyperbolization and high-order ader dis-*
689 *cretizations*, SIAM Journal on Scientific Computing, 36 (2014), pp. A2423–A2457.
- 690 [24] E. VOLKOV, *The method of composite meshes*, Automatic Programming, Numerical Methods and Functional Analysis, 96
691 (1970), p. 145.
- 692 [25] Z. WANG, *A fully conservative interface algorithm for overlapped grids*, Journal of Computational Physics, 122 (1995),
693 pp. 96–106.