



**HAL**  
open science

# Second Order ADER 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 ADER Scheme for Unsteady Advection-Diffusion on Moving Overset Grids with a Compact Transmission Condition. *SIAM Journal on Scientific Computing*, 2022, 44 (1), pp.A524-A553. 10.1137/21M1393911 . hal-03615757

**HAL Id: hal-03615757**

**<https://hal.inria.fr/hal-03615757>**

Submitted on 12 Apr 2022

**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.

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

MICHEL BERGMANN<sup>†‡</sup>, MICHELE GIULIANO CARLINO<sup>†‡</sup>, AND ANGELO IOLLO<sup>†‡</sup>

**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.

<sup>†</sup>INRIA Bordeaux Sud-Ouest, 200 Avenue de la Vieille Tour, 33405 Talence, France. ([michel.bergmann@inria.fr](mailto:michel.bergmann@inria.fr), [michele-giuliano.carlino@inria.fr](mailto:michele-giuliano.carlino@inria.fr), [angelo.iollo@inria.fr](mailto:angelo.iollo@inria.fr))

<sup>‡</sup>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 [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 Discontinuous 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*  $u : \Omega(t) \times [0, T] \rightarrow \mathbb{R}^\delta$  *such that*

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

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

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

109 **2. The overset grid.** An overset grid or Chimera mesh is a set of mesh blocks covering the compu-  
 110 tational domain. Each block may overlap other block(s) in some particular sub-region(s) said *overlapping*  
 111 *zone(s)*. Once the multiple mesh patches are generated, they are collated in order to generate an appropriate  
 112 topology [22]. Consequently, an overlapping zone between two neighbouring blocks is defined. For the sake of  
 113 simplicity with no loss of generality, the whole method is explained by considering a two blocks overset grid  
 114 (i.e., the background and the foreground meshes). For multiple-block meshes (e.g.  $\mathcal{T}_1, \dots, \mathcal{T}_N$ ), a hierarchy  
 115 of meshes from the background to the foreground is defined (e.g.  $\mathcal{T}_1 < \dots < \mathcal{T}_N$ ). Successively the presented  
 116 algorithm for setting the overset grid is performed from one mesh to the union of all other meshes towards  
 117 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  
 118 setting is presented. Figure 1 shows an overset grid; in black there is the *background* mesh and in blue  
 119 the *foreground* mesh. In particular, the foreground mesh can move and deform. The overlapping zone is  
 120 necessary for the communication and data transfer from one mesh to the other.  
 121 In this work, the cell of any block mesh is considered quadrilateral. In particular, when all the cells are  
 122 squared, the mesh is uniform. When the cells are either squared or rectangular and the edges are oriented  
 123 as the Cartesian axes, the mesh is said to be Cartesian.

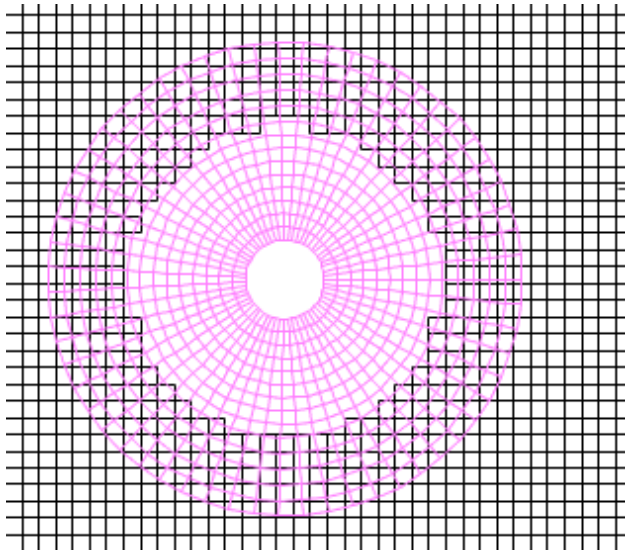


Fig. 1: Example of Chimera grid configuration. In black there is the *background* mesh and in pink the *foreground* mesh.

124 **2.1. The automatic definition of the stencil at the transmission condition.** Let  $\mathcal{T}_k = \{\Omega_i^k\}_{i=1}^{N_k}$   
 125 be the partition composed of  $N_k$  cells referring to the  $k$ -th block mesh (in order to simplify the notation, we  
 126 will omit the superscript  $k$  to the cell  $\Omega_i^k$  by writing  $\Omega_i$ ), moreover, let  $\mathcal{S}_i$  be the stencil centered over the  
 127 cell  $\Omega_i$ . Thus, stencil  $\mathcal{S}_i$  is the set collecting the indexes of neighboring cells to  $\Omega_i$ . By abuse of language,  
 128 sometimes we will refer to the physical set  $\Omega_i \cup \bigcup_{j \in \mathcal{S}_i} \Omega_j$  as the stencil.  
 129 It is possible to distinguish two classes of cells with respect to their proximity to the overlapping interface.  
 130 The definition of the stencil depends on the class.  
 131 If cell  $\Omega_i$  is not at the boundary of the overlapping zone (Figure 2a), the stencil  $\mathcal{S}_i$  is composed of all the

132 cells  $\Omega_j$  sharing at least one vertex with  $\Omega_i$ . Thus, if  $\Omega_i$  belongs to the partition  $\mathcal{T}_1$ , all cells  $\Omega_j$ , with  $j \in \mathcal{S}_i$ ,  
 133 also belong to  $\mathcal{T}_1$ .

134 If the cell  $\Omega_i$  of partition  $\mathcal{T}_k$  is at the boundary of the interface, it is no longer possible to use the criterion of  
 135 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  
 136 the same partition (see right edge of cell  $\Omega_{16}$  in Figure 2b). For these cells, we aim in automatically finding  
 137 the other cells of partition  $\mathcal{T}_j$  ( $j \neq k$ ) belonging to the stencil. Let the extremes of the edge be indicated as  
 138  $\mathbf{v}_1$  and  $\mathbf{v}_2$  and its middle point with  $\mathbf{v}_3$ , respectively. Point  $\mathbf{c}_*$  is the center of mass of generic cell  $\Omega_*$ . For  
 139 our numerical tests, Algorithm 2.1 is adopted through the two steps:

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

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

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

149 Algorithm 2.1 could not define a compact stencil in the case of widely different mesh spacing. In this case,  
 150 more than three points  $\mathbf{v}_\mu$  can be considered for lines 5 and 6. Moreover a weighted symmetry (possibly led  
 151 by the different spacing) can be performed at line 6.

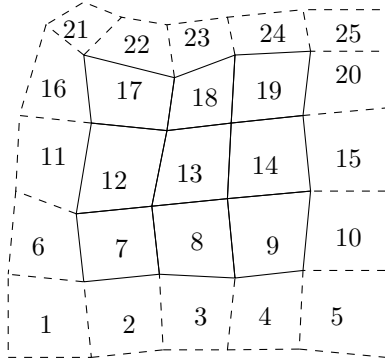
---

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

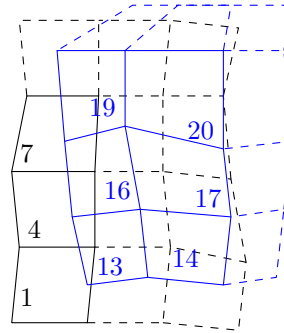
---

**Input:**  $\Omega_i^k, e_{il}^k, \mathcal{T}_j, \mathcal{S}_i^k;$   $\triangleright j \neq k$ , 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  $\Omega_i^k$  with respect to  $\mathbf{v}_\mu$   
 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**  $\mathcal{S}_i^k$

---

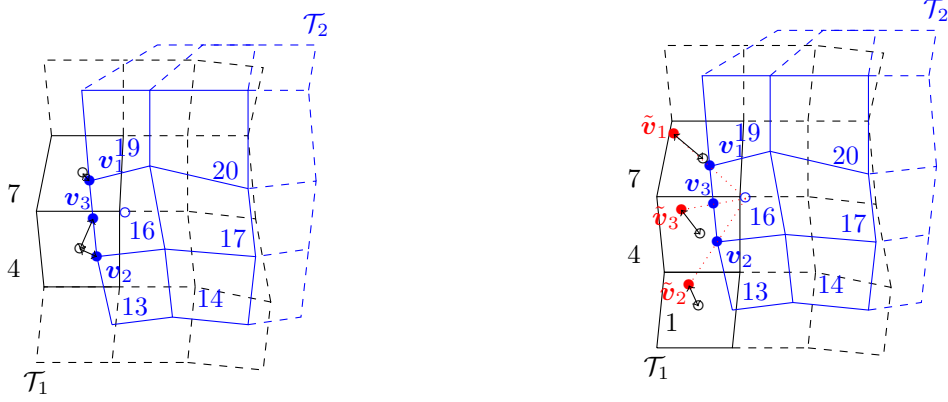


(a) A stencil of cells in the same partition. Continuous line for the stencil  $\mathcal{S}_{13} = \{7, 8, 9, 12, 14, 17, 18, 19\}$ .



(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 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  $v_1$  and  $v_2$  and the middle point  $v_3$  of the edge on the boundary cell  $\Omega_{16}$  (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  $\tilde{v}_\mu$ ,  $\mu = 1, 2, 3$ , (red full dots) of the node of the cell  $\Omega_{16}$  (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  $\Omega_{16} \in \mathcal{T}_2$ .

152 **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.  
 153  
 154 The method consists in a FEM-predictor FVM-corrector scheme stabilised with a Local Lax-Friederichs  
 155 approach whose stabilization coefficient is explained in the following section.

156 **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  $\Omega_i$ . Since the scheme is cell-centered, at time  $t^n$ ,  
 157 we would like to extend (at least locally) the solution to the whole cell by exploiting the information in the  
 158 cells of the stencil referring to  $\Omega_i^n$ . In order to explain the reconstruction, let us consider a generic regular<sup>1</sup>  
 159 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  
 160 zone, the cell composing the subdomain  $E$  does not necessary fulfill the non-overlapping condition, i.e., it  
 161 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  
 162 know the value of function  $\phi$  over the center of mass  $(x_k, y_k) = \mathbf{x}_k$ , with  $k \in \{i\} \cup \mathcal{S}_i$ , of any  $\Omega_k$  composing  
 163  $E$ . 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  
 164 function  $\phi$  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  
 165 to write the Taylor's polynomial truncated to the quadratic terms with respect to  $\phi_i$ :  
 166

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

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

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

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

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

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



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

$$178 \quad (3.4) \quad \begin{bmatrix} h_{ik}^x & h_{ik}^y & h_{ik}^x h_{ik}^y & \frac{1}{2}(h_{ik}^x)^2 & \frac{1}{2}(h_{ik}^y)^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{ij}^x & h_{ij}^y & h_{ij}^x h_{ij}^y & \frac{1}{2}(h_{ij}^x)^2 & \frac{1}{2}(h_{ij}^y)^2 \end{bmatrix} \begin{bmatrix} p_1 \\ \vdots \\ p_5 \end{bmatrix} = \begin{bmatrix} \delta \phi_{ik} \\ \vdots \\ \delta \phi_{ij} \end{bmatrix},$$

179 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 solved in  
 180 least-square sense if  $|\mathcal{S}_i| > 5$ . Moreover, if the chosen polynomial basis is not reduced, namely if the Taylor's  
 181 expansion (3.1) is arrested to the bi-linear or linear terms, the stencil has to contain at least 5 cells in order  
 182 to ensure a solution for (3.4). The proposed  $\mathcal{P}_2$ -interpolation, with the second-order accurate scheme, fulfills  
 183 the condition for the accuracy in the interpolation for overlapping zones whose depth  $d_o$  degrades as the  
 184 characteristic length  $h$  of the chimera mesh (i.e.,  $d_o = \mathcal{O}(h)$ ) [9].

185 This method allows to locally reconstruct all over the stencil a given function. If the function is defined  
 186 over the computational domain  $\Omega \subset \mathbb{R}^2$  and it is (at least locally)  $C^2$ , then the reconstruction is locally  
 187 computed over any stencil and the ensured order of convergence is 3. On the contrary, if the solution  
 188 presents propagating shock waves or discontinuities, this interpolation is no longer adequate because of well-  
 189 known Gibbs' phenomenon, for which spurious oscillation are produced near the discontinuity. For those  
 190 cases, other interpolation could be adopted, such as the central weighted ENO for hyperbolic equations for  
 191 moving meshes in [11].

192 In the sequel, the local polynomial reconstruction  $\Pi_i \mathbf{u}^n$  will be referred as  $\mathbf{w}_i^n$ .

193 **3.2. Local space-time Galerkin predictor.** Let be the time interval  $[0, T]$  subdivided in  $N$  subin-  
 194 tervals  $[t^n, t^{n+1}]$ , with  $n = 0, \dots, N - 1$ ; thus for a generic time-dependent variable  $\mathbf{g}(t)$ , we define  $\mathbf{g}^n$  for  
 195  $\mathbf{g}^n = \mathbf{g}(t^n)$ . In particular, the domain  $\Omega^n$  and the solution  $\mathbf{u}^n$  at time  $t^n$  are considered the actual spatial con-  
 196 figuration 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 whose  
 197 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)  
 198 is rewritten with respect to a space-time reference system identified by the independent variables  $\boldsymbol{\xi} \equiv (\xi, \eta, \tau)$   
 199 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  
 200 equation is discretized using an efficient nodal formulation of space-time nodes given by a tensor product  
 201 of Gauss-Legendre quadrature points along space and time directions. This choice defines an  $L^2$ -orthogonal  
 202 Lagrange basis used for the definition of the Galerkin solution. For our purposes, the single direction nodes  
 203 over the unit interval  $[0, 1]$  are  $\{(5 - \sqrt{15})/10; 1/2; (5 + \sqrt{15})/10\}$ . Consequently, over a space-time cell there  
 204 will be 27 Gauss-Legendre 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  
 205  $\int_{\hat{\mathcal{C}}} \theta_l \theta_m \, d\boldsymbol{\xi} = \delta_{lm} \|\theta_l\|_{L^2(\hat{\mathcal{C}})}^2$ , with  $\delta_{lm}$  the Kronecher's symbol. Let  $\mathbf{m} : \{1, 2, 3\}^3 \rightarrow \{1, \dots, 27\}$  be a discrete  
 206 map from a single direction index to the global three dimensional index defined as

$$207 \quad \mathbf{m}(i, j, k) = ij + (j - 1)(3 - i) + 9(k - 1),$$

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

$$212 \quad \hat{\boldsymbol{\xi}}_l = (\hat{\xi}_i, \hat{\eta}_j, \hat{\tau}_k); \quad \theta_l(\xi, \eta, \tau) = \theta_i^\xi(\xi) \theta_j^\eta(\eta) \theta_k^\tau(\tau),$$

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

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

$$215 \quad (3.5) \quad \begin{cases} \partial_t \mathbf{q} + \nabla \cdot \mathbf{F}(\mathbf{q}, \nabla \mathbf{q}) = \mathbf{f} & \text{in } \mathcal{C}_i^n \\ \mathbf{q}|_{t=t^n} = \mathbf{w}_i^n & \text{on } \Omega_i^n \end{cases},$$

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

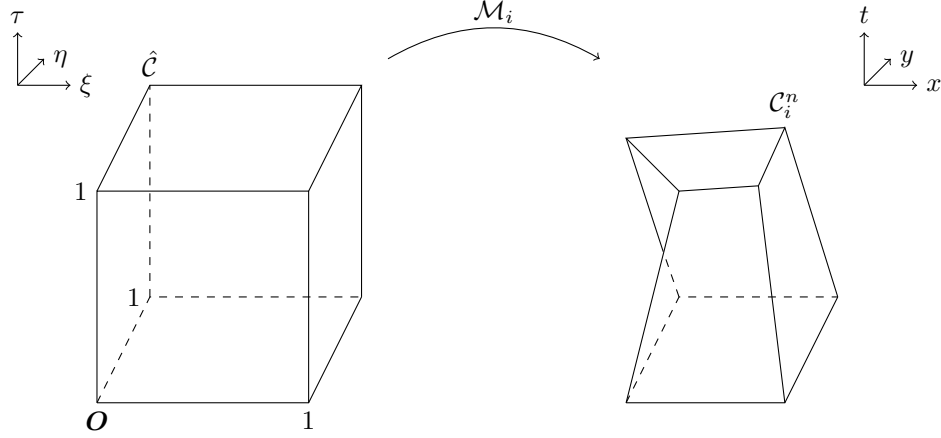


Fig. 4: Representation of the map  $\mathcal{M}_i$  from the reference space-time cell  $\hat{\mathcal{C}}$  to the physical space-time cell  $\mathcal{C}_i^n$ .

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

$$219 \quad (3.6) \quad \mathcal{M}_i : \begin{cases} x = x(\xi, \eta, \tau) \\ y = y(\xi, \eta, \tau) \\ t = t^n + \Delta t \tau \end{cases},$$

220 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  
 221  $\boldsymbol{\xi} \in \hat{\mathcal{C}}$  (see Figure 4). Time  $t$  is considered as a linear function of  $\tau$ . From map (3.6), we define the Jacobian  
 222 matrix  $J$  as

$$223 \quad (3.7) \quad J = \frac{d\mathbf{x}}{d\boldsymbol{\xi}} = \begin{bmatrix} x_\xi & x_\eta & x_\tau \\ y_\xi & y_\eta & y_\tau \\ 0 & 0 & \Delta t \end{bmatrix},$$

224 whose inverse is

$$225 \quad (3.8) \quad J^{-1} = \frac{d\boldsymbol{\xi}}{d\mathbf{x}} = \begin{bmatrix} \xi_x & \xi_y & \xi_t \\ \eta_x & \eta_y & \eta_t \\ 0 & 0 & 1/\Delta t \end{bmatrix} = \begin{bmatrix} J_s^{-1} & \boldsymbol{\Xi}_t \\ \mathbf{0} & 1/\Delta t \end{bmatrix}.$$

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

$$228 \quad (3.9) \quad J_s^{-1} = \begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix}.$$

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

$$231 \quad (3.10) \quad \partial_\tau \mathbf{q} + \Delta t \mathcal{F}^*(\hat{\nabla} \mathbf{q}) + \Delta t J_s^{-T} \hat{\nabla} \cdot \mathcal{F}^{**}(\mathbf{q}, \hat{\nabla} \mathbf{q}) = \Delta t \mathbf{f},$$

232 where

$$233 \quad \partial_t \mathbf{q} = \frac{\partial_\tau \mathbf{q}}{\Delta t} + \mathcal{F}^*(\hat{\nabla} \mathbf{q}); \quad \mathcal{F}^*(\hat{\nabla} \mathbf{q}) = \hat{\nabla} \mathbf{q} \boldsymbol{\Xi}_t; \quad \mathcal{F}^{**}(\mathbf{q}, \hat{\nabla} \mathbf{q}) = \mathbf{F}(\mathbf{q}, J_s^{-T} \hat{\nabla} \mathbf{q}) = (\mathcal{F}_\xi^{**}, \mathcal{F}_\eta^{**}); \quad \hat{\nabla} = \begin{bmatrix} \partial_\xi \\ \partial_\eta \end{bmatrix}.$$

234 The hat differential operators refer to reference space variables  $\xi$  and  $\eta$  in the reference space-time cell  $\hat{\mathcal{C}}$ . By  
 235 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.,  $\mathbf{q}$  and  $\mathbf{f}$ ) 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  $\tau$ . Moreover, the following notation is introduced:

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

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

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

where  $w_k$  is the  $k$ -th component of the interpolated polynomial  $\mathbf{w}^n$ . 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*  $\mathbf{q} \in Q^\delta$  such that

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

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

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

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  $\mathcal{F}^*$  and  $\mathcal{F}^{**}$  which depend on the solution  $\mathbf{q}_h$ . For this reason a fixed point problem is solved: let  $r$  be the index of the fixed point iteration, therefore we solve

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

where terms of fixed point index  $r$  are computed by using the previous solution  $\mathbf{q}_h^r$ . 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  $\mathcal{M}_i : \hat{\mathcal{C}} \rightarrow \mathcal{C}_i^n$  has been involved for the computation of the local weak predictor solution. Moreover, the foreground mesh of coordinates  $\mathbf{X}$  is moving accordingly to the following motion equation:

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

where  $\mathbf{V} = \mathbf{V}(\mathbf{x}, t; u)$  is the mesh velocity, eventually dependent on the solution. Equation (3.14) is closed with a Cauchy condition  $\mathbf{X}(0) = \mathbf{X}_0$ , which is the initial spatial configuration. Through equation (3.14),

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

$$282 \quad (3.15) \quad \frac{d\mathbf{X}}{dt} = \mathbf{V} \text{ in } \mathcal{C}_i^n,$$

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

$$286 \quad (3.16) \quad ([\theta_k, \theta_l]_1 - \langle \partial_\tau \theta_k, \theta_l \rangle) \hat{\mathbf{X}}_l = \Delta t \langle \theta_k, \theta_l \rangle \hat{\mathbf{V}}_l + [\theta_k, \theta_l]_0 \hat{\mathbf{X}}_l^n,$$

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

289 When the mesh is neither moving nor deforming, as for cells in the background, the mesh velocity is thus  
 290 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  
 291 the reference space-time cell  $\hat{\mathcal{C}}$  to the physical space-time cell  $\mathcal{C}_i^n$ :

$$292 \quad (3.17) \quad \begin{cases} x = x(\xi) = x_{i-1/2} + h_i^x \xi \\ y = y(\eta) = y_{i-1/2} + h_i^y \eta \end{cases},$$

293 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  
 294 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  
 295 along  $x$  and  $y$  of the cell, respectively, i.e.  $h_i^x = x_{i+1/2} - x_{i-1/2}$  and  $h_i^y = y_{i+1/2} - y_{i-1/2}$ .

296 Since the mesh motion equation (3.14) is essentially solved via a sort of Discontinuous Galerkin (DG)  
 297 approach, possible numerical (and non physical) discontinuities could arise. As a matter of fact, for a  
 298 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  
 299 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  $\Omega_i^{n+1}$   
 300 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  
 301  $\{\Omega_i^{n+1}\}_{i \in \mathcal{Z}_k^{n+1}}$ , coinciding with the cardinality of the indexes set  $\mathcal{Z}_k^{n+1}$ , depends on the position of the vertex  
 302  $\bar{\mathbf{X}}_k^{n+1}$  on the foreground mesh: it is either 1 or 2 if the vertex is on the boundary of the mesh, otherwise it is  
 303 4. For this reason we consider a weighted average value for the shared vertex in order to tackle the possible  
 304 arising discontinuities. As suggested in [6], we first consider a weighted velocity  $\bar{\mathbf{V}}_k^{n+1}$  corresponding to the  
 305 vertex  $\bar{\mathbf{X}}_k^{n+1}$

$$306 \quad (3.18) \quad \bar{\mathbf{V}}_k^{n+1} = \frac{1}{N_k} \sum_{i \in \mathcal{Z}_k^{n+1}} \bar{\mathbf{V}}_{k,i}^{n+1}, \text{ with } \bar{\mathbf{V}}_{k,i}^{n+1} = \int_0^1 \theta_l(\xi^*, \eta^*, \tau) d\tau \hat{\mathbf{V}}_{l,i},$$

307 where coordinates  $(\xi^*, \eta^*)$  depend on the position of the coordinate  $\bar{\mathbf{X}}_k^{n+1}$  in the cell  $\Omega_i^{n+1}$ ; it can assume four  
 308 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  
 309 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,  
 310 the coordinates  $\bar{\mathbf{X}}_k^{n+1}$  at time  $t^{n+1}$  is

$$311 \quad (3.19) \quad \bar{\mathbf{X}}_k^{n+1} = \bar{\mathbf{X}}_k^n + \Delta t \bar{\mathbf{V}}_k^{n+1}.$$

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

314 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);
- 

315 **3.4. Correction stage: the finite volume scheme over the space-time cell.** Once the local  
 316 predictor solution  $q\mathbf{q}_h$  is computed in each space-time cells  $\mathcal{C}_i^n$ , we can perform the correction stage. First,  
 317 we rewrite the convective-diffusive equation (1.1) in divergence form. Let  $\nabla_{\mathbf{x},t} = [\nabla, \partial_t]^T$  be the space-time  
 318 differential operator and let  $\mathbf{U} = [\mathbf{F}(\mathbf{u}, \nabla \mathbf{u}), \mathbf{u}]^T$  be the space-time solution, thus problem (1.1) can be  
 319 rewritten as

$$320 \quad (3.20) \quad \nabla_{\mathbf{x},t} \cdot \mathbf{U} = \mathbf{f} \quad \text{in } \Omega(t) \times [0, T].$$

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

$$323 \quad (3.21) \quad \partial \mathcal{C}_i^n = \Omega_i^n \cup \Omega_i^{n+1} \cup \bigcup_{j=1}^4 \Gamma_{ij}^n,$$

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

$$327 \quad (3.22) \quad \int_{\partial \mathcal{C}_i^n} \mathbf{U} \cdot \mathbf{n}_{\mathbf{x},t} \, d\Gamma = \int_{\mathcal{C}_i^n} \mathbf{f} \, d\mathcal{C},$$

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

$$330 \quad (3.23) \quad \mathbf{U}_i^n = \frac{1}{|\Omega_i^n|} \int_{\Omega_i^n} \mathbf{u}(x, y, t^n) \, dx \, dy,$$

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

$$332 \quad (3.24) \quad -|\Omega_i^n| \mathbf{U}_i^n + |\Omega_i^{n+1}| \mathbf{U}_i^{n+1} + \sum_{j=1}^4 \int_{\Gamma_{ij}^n} \mathbf{U} \cdot \mathbf{n}_{\mathbf{x},t} \, d\Gamma = \int_{\mathcal{C}_i^n} \mathbf{f} \, d\mathcal{C},$$

333 where the unknown is the average solution  $\mathbf{U}_i^{n+1}$  at time  $t^{n+1}$ , while the last term of the left hand side is the  
 334 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  
 335 that it is still exact. In order to solve (3.24), we need to approximate the integral function of the space-time  
 336 flux. Among the several methods proposed in the literature (such as in [11, 12, 13, 33, 17]), we here present  
 337 a Local Lax-Friederichs (LLF) approach:

$$338 \quad (3.25) \quad [\mathbf{U} \cdot \mathbf{n}_{\mathbf{x},t}]_{\Gamma_{ij}^n} \approx \Phi(\mathbf{q}_j^+, \mathbf{q}_j^-) = \frac{1}{2}(\mathbf{U}_j^+ + \mathbf{U}_j^-) \cdot \mathbf{n}_{\mathbf{x},t} - \frac{s}{2}(\mathbf{q}_j^+ - \mathbf{q}_j^-),$$

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

342 (3.25) closes the correction stage of the ADER method. At the end of this stage, a solution  $u_i^{n+1}$  is found  
 343 over any cell  $\Omega_i^{n+1}$ . Since the predictor solution over space-time cells  $\mathcal{C}_i^{n+1}$  needs to be evaluated over the  
 344 Gauss nodes, a second order local polynomial interpolation is performed as explained in Section 3.1.  
 345 For the computation of the integrals along the space-time manifolds  $\Gamma_{ij}^n$ , we still use the previously computed  
 346 map  $\mathcal{M}_i$ . As a matter of fact, for a generic function  $g : \mathcal{C}_i^n \rightarrow \mathbb{R}$  it holds:

$$347 \quad \int_{\Gamma_{ij}^n} g(\mathbf{x}) \, d\Gamma = \int_{\hat{\Gamma}_j} g(\mathbf{x}(\boldsymbol{\xi})) |\text{Cof}(J)\hat{\mathbf{n}}_j| \, d\hat{\Gamma},$$

348 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  
 349  $\text{Cof}(J)$  is the cofactor matrix of the Jacobian tensor  $J$  of the map.

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

$$353 \quad (3.26) \quad \Delta t = \text{CFL} \frac{h}{\max\{\sup_{\Omega \times [0, T]} |a_x|, \sup_{\Omega \times [0, T]} |a_y|\}},$$

354 where  $h$  is the smallest characteristic length among all cells (both of background and foreground meshes)  
 355 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  
 356  $\Omega_i^n$  at discrete time  $t^n$ . Coefficient CFL in (3.26) is the Courant-Friedrichs-Lewy number. In this paper, the  
 357 CFL coefficient is experimentally sought by conducting an empirical analysis in Section 5.2.

358 **3.5. Dynamics of the overlapping zone.** During the simulation, the foreground mesh moves and,  
 359 consequently, the background mesh changes its configuration in the zone of the overlapping as well as in the  
 360 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  
 361 three possibilities:

- 362 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;
- 363 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;
- 364 3. The overlapping zone does not drastically change its configuration with respect to cell  $\Omega_i(t)$ , thus  
 365 the cell is present at time  $t^n$  and it still continues to be present at time  $t^{n+1}$ .

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

$$371 \quad (3.27) \quad \mathbf{x}_j = \arg \min_{k > N_1} \|\mathbf{x}_i - \mathbf{x}_k\|,$$

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

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

381 **4.1. The local advective-diffusive stabilization term.** For the definition of the coefficient  $s_{AD}$  in  
 382 (3.25), we study a relaxed hyperbolic form of the parabolic equation (3.20). Let us consider the following  
 383 relaxation by Cattaneo (we refer to [34] and its references for further details): let  $0 < \varepsilon \ll 1$  be a relaxed  
 384 time and consider variables  $v$  and  $w$  in  $\Omega \times [0, T]$  such that

$$385 \quad (4.1) \quad \partial_t v = \frac{1}{\varepsilon} (\partial_x u - v); \quad \partial_t w = \frac{1}{\varepsilon} (\partial_y u - w).$$

386 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  
 387  $\varepsilon$ . Since the flux has to be computed along the manifold  $\Gamma_{ij}^n$  in the space-time continuum, let us consider  
 388 solution  $u$  and all its first derivatives as *stationary solutions* with respect to a pseudo-time  $\mathbf{t} \in \mathbb{R}_+$ . Thus, let  
 389  $\mathbf{u}(\mathbf{t}; x, y, t) = [u, v, w]^T$  be the formal definition of the relaxed hyperbolic system with respect to pseudo-time  
 390  $\mathbf{t}$ . It holds  $\partial_{\mathbf{t}} \mathbf{u} = 0$ . The conservative form problem (3.20) in quasi-linear form is

$$391 \quad (4.2) \quad \partial_{\mathbf{t}} \mathbf{u} + \partial_x (A\mathbf{u}) + \partial_y (B\mathbf{u}) + \partial_t (C\mathbf{u}) = \mathbf{f} \quad \text{in } \mathbb{R}_+ \times \Omega(t) \times [0, T],$$

392 where  $A$ ,  $B$  and  $C$  are  $3 \times 3$  matrices (eventually involving the solution  $\mathbf{u}$  among their components if the  
 393 original problem is nonlinear) and the force term  $\mathbf{f} = [f, -v/\varepsilon, -w/\varepsilon]^T$ . In particular,  $A$  and  $B$  always  
 394 depend on the relaxation time  $\varepsilon$  and they are defined by the convection-diffusion term  $\mathcal{F}(u, \nabla u)$  and  $C$  is  
 395 always the identity matrix if the Cattaneo's relaxation (4.1) is employed. In order to study the differential  
 396 operator in (4.2), let us consider a vanishing force term, i.e.  $\mathbf{f} \equiv \mathbf{0}$ . The presence of the pseudo-time  $\mathbf{t}$  in  
 397 (4.2) helps in treating the real time variable  $t$  as any other spatial variable  $x$  and  $y$ . When the force term in  
 398 (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  
 399 of real values  $n_x$ ,  $n_y$  and  $n_t$ . If the hyperbolicity is ensured, the relaxed hyperbolic system has a planar  
 400 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  
 401 particular direction in the space-time continuum, the eigenvalues of  $\mathcal{A}$  define the speeds of propagation of the  
 402 solution along the principal directions defined by the eigenvectors of  $\mathcal{A}$ . For this reason, in the perspective  
 403 of an upwind stabilization, the local stabilization term  $s_{AD}$  in (3.25) is equal to the maximum speed of  
 404 propagation of the wave, as it happens for the LLF flux approximation for a generic hyperbolic problem of  
 405 a propagating wave in the space continuum.

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

$$410 \quad A = \begin{bmatrix} a_x & -\nu & 0 \\ -1/\varepsilon & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} a_y & 0 & -\nu \\ 0 & 0 & 0 \\ -1/\varepsilon & 0 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

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

$$412 \quad (4.3) \quad \rho(\mathcal{A}) = \left\{ n_t; \frac{1}{2} \left[ \sigma \pm \sqrt{\left( a_x^2 + \frac{4\nu}{\varepsilon} \right) n_x^2 + 2a_x a_y n_x n_y + \left( a_y^2 + \frac{4\nu}{\varepsilon} \right) n_y^2} \right] \right\},$$

413 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  
 414 stabilization parameter.

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

$$418 \quad (4.4) \quad s_{AD} = \max |\rho(\mathcal{A})| = \frac{1}{2} \left| \sigma + \sqrt{\left( a_x^2 + \frac{4\nu}{\varepsilon} \right) n_x^2 + 2a_x a_y n_x n_y + \left( a_y^2 + \frac{4\nu}{\varepsilon} \right) n_y^2} \right|.$$

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

421 **4.2. The choice of the relaxation time.** For the definition of the advective-diffusive stabilization  
 422 term  $s_{AD}$ , we considered the relaxed hyperbolic system (4.2) deriving from the parabolic problem (3.20)  
 423 through a relaxation time  $\varepsilon$ . If we were to solve the relaxed problem instead of the original one, the  
 424 approximate solution would differ from the exact solution of two errors that are added together: the numerical  
 425 error (typical of the scheme) and a relaxation error. For a linear problem, these errors have been investigated  
 426 by Montecinos and Toro in [34]. The error  $|u_{\text{hip}} - u|$  between the hyperbolized solution  $u_{\text{hip}}$  and the original  
 427 solution  $u$  is  $\mathcal{O}(\varepsilon)$  [24]. Thus, if  $u_{\text{hip}, h}$  is a numerical approximation of the exact relaxation solution  $u_{\text{hip}}$ ,  
 428 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

429 maximum characteristic length of cells  $\Omega_i(t)$ 's. However, the goal is to choose a relaxation time  $\varepsilon$  such that  
 430 the relaxation error is always dominated by or, at least, comparable to the numerical error, i.e.  $\mathcal{O}(\varepsilon) \lesssim \mathcal{O}(h_0^p)$ .  
 431 The following theoretical result can help in fulfilling our task.

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

$$435 \quad (4.5) \quad C_p \frac{\varepsilon}{h_0^p} = \mathcal{O}(1),$$

436 *with*

$$437 \quad C_p = \frac{1 - 2^{-\frac{1}{2}}}{2^{p-\frac{1}{2}} - 1}.$$

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

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

$$443 \quad (4.6) \quad \varepsilon_p = \frac{\mathcal{O}(1)h_0^p}{C_p}.$$

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

448 **4.3. The local advective stabilization term.** In order to recover a stabilization term  $s_A$  by only  
 449 considering the first order operator involved in the whole differential operator of the original problem, we  
 450 can treat the equation to stabilize as a pure hyperbolic (namely just advective) problem. For this reason,  
 451 the advective stabilization term  $s_A$  coincides with the maximum eigenvalue of the ALE Jacobian matrix  
 452 in a spatial normal direction by excluding the diffusive component which acts on the diffusion from the  
 453 advective-diffusive term  $\mathbf{F}(u, \nabla u)$  [11]. This matrix reads

$$454 \quad (4.7) \quad A_{\tilde{\mathbf{n}}}^{\mathbf{V}} = \sqrt{n_x^2 + n_y^2} \left[ \frac{\partial \mathbf{F}}{\partial u} \tilde{\mathbf{n}} - \mathbf{V} \cdot \tilde{\mathbf{n}} I \right],$$

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

$$458 \quad \tilde{\mathbf{n}} = \frac{[n_x, n_y]^T}{\sqrt{n_x^2 + n_y^2}}.$$

459 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  $\Gamma_{ij}^n$ ,  
 460  $j = 1, \dots, 4$ , of the space-time cell  $\mathcal{C}_i$  can be described by only two of the three reference space-time variables  
 461  $(\xi, \eta, \tau)$ ; i.e., by either couple  $(\xi, \tau)$ , with  $\eta = \bar{\eta}$ , or couple  $(\eta, \tau)$ , with  $\xi = \bar{\xi}$ ; with  $\bar{\xi}$  and  $\bar{\eta}$  alternatively equal  
 462 to 0 or 1, depending on the specific  $j$ -th space-time manifold  $\Gamma_{ij}^n$ . Let  $\chi$  be the free variable (e.g.  $\chi = \xi$ )  
 463 and  $\bar{\kappa}$  be the constrained variable (e.g.  $\bar{\kappa} = \bar{\eta}$ ) for the specific manifold  $\Gamma_{ij}^n$ . Therefore, for a specific point  
 464  $\tilde{\mathbf{x}}$  over  $\Gamma_{ij}^n$  it is possible to distinguish two directional vectors provided by the map  $\mathcal{M}_i$

$$465 \quad \mathbf{r}_\chi = \begin{bmatrix} x_\chi \\ y_\chi \\ 0 \end{bmatrix}_{\bar{\kappa}} \quad \text{and} \quad \mathbf{r}_\tau = \begin{bmatrix} x_\tau \\ y_\tau \\ \Delta t \end{bmatrix}_{\bar{\kappa}}.$$



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

$$468 \quad \mathbf{n}_{\mathbf{x},t} = \frac{\mathbf{r}_\chi \wedge \mathbf{r}_\tau}{|\mathbf{r}_\chi \wedge \mathbf{r}_\tau|_{\bar{\kappa}}} = \frac{[\Delta t y_\chi, -\Delta t x_\chi, d_{\chi\tau}]^T}{\sqrt{\Delta t^2 y_\chi^2 + \Delta t^2 x_\chi^2 + d_{\chi\tau}^2}_{\bar{\kappa}}},$$

469 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  
 470 the unit vector  $\tilde{\mathbf{n}}$  along the spatial directions and the velocity of the point as

$$471 \quad \tilde{\mathbf{n}} = \frac{[y_\chi, -x_\chi]^T}{\sqrt{y_\chi^2 + x_\chi^2}} \quad \text{and} \quad \mathbf{V} = \frac{d\tilde{\mathbf{x}}}{dt} = \frac{[x_\tau, y_\tau]^T}{\Delta t}.$$

472 Consequently it holds

$$473 \quad (4.8) \quad \mathbf{V} \cdot \tilde{\mathbf{n}} = \frac{-d_{\chi\tau}}{\Delta t \sqrt{y_\chi^2 + x_\chi^2}} = \frac{-n_t}{\sqrt{n_x^2 + n_y^2}}.$$

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

$$475 \quad (4.9) \quad s_A = |a_x n_x + a_y n_y + n_t|.$$

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

478 PROPOSITION 4.4. *For linear problem (1.1), let the diffusion parameter  $\nu$  go to zero, therefore the fol-*  
 479 *lowing limit holds*

$$480 \quad (4.10) \quad \lim_{\nu \rightarrow 0} s_{AD} = \frac{1}{2} |\sigma + a_x n_x + a_y n_y| = |a_x n_x + a_y n_y + n_t| = s_A.$$

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

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

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

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

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



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

511 **5.1. Order of convergence.** In this section we have a double goal. On one hand we want to nu-  
 512 merically prove that the presented method is second order when an advective-diffusive LLF stabilization  
 513  $s_{AD}$  is employed. On the other hand, we want to compare this stabilization term with the local advective  
 514 stabilization flux  $s_A$ . The study of the second order convergence is conducted on all test cases of Table 1.  
 515 Finally, on the two mentioned 2D test cases the comparison of the performances for the flux approximations  
 516 is carried out.

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

$$519 \quad (5.1) \quad L^\infty\text{-err} = \|u - u_{ex}\|_{L^\infty(\Omega)} = \operatorname{ess\,sup}_{\mathbf{x} \in \Omega} |u(\mathbf{x}, T) - u_{ex}(\mathbf{x}, T)| \approx e_{L^\infty}^N = \max_{k=1, \dots, N} |u_k^M - u_{ex}(\mathbf{x}_k, T)|$$

520 and

$$521 \quad (5.2) \quad L^2\text{-err} = \|u - u_{ex}\|_{L^2(\Omega)} = \sqrt{\int_{\Omega} (u(\mathbf{x}, T) - u_{ex}(\mathbf{x}, T))^2 \, d\Omega} \approx e_{L^2}^N = \sqrt{\frac{|\Omega| \sum_{k=1}^N (u_k^M - u_{ex}(\mathbf{x}_k, T))^2}{N}},$$

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

$$526 \quad (5.3) \quad L^p\text{-rate} = d \frac{\log(e_{L^p}^{N_1}/e_{L^p}^{N_2})}{\log(N_2/N_1)}, \quad \text{for } p = 2, \infty,$$

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

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

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

541 **5.2. Empirical analysis of stability condition.** As already mentioned at the end of Section 3.4, the  
 542 presence of a weak solution, found in the prediction step of the presented method and successively plugged  
 543 into the flux of the finite volume scheme in the correction stage, makes a classical stability analysis not  
 544 straightforward to be made. For this reason, we performed an empirical stability analysis by assuming that  
 545 the right time step  $\Delta t$  allowing a stable computation is defined as in (3.26).

546 On a given problem, once both background and foreground meshes are set, we considered a time step  $\Delta t$   
 547 starting from a CFL number equal to 0.1 and, by increasing this value of 0.05 each time, we look for the largest  
 548 stable CFL. In particular, this process is executed on the same problem considering an approximated LLF  
 549 flux employing once an advective-diffusive stabilization term  $s_{AD}$  and then with an advective stabilization  
 550 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  $\mathbf{V}$  (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>
$\Omega$	$(-1, 1)$	$(-1, 1)$	$(-\pi, \pi)^2$	$(-\pi, \pi)^2$
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	$\pi$
fg mesh	$[-0.5, 0.25]$	$[-0.25, 0.25]$	$[-0.5, 0.5]^2$	$[-1.25, -0.75] \times [-0.25, 0.25]$
$\mathbf{V}$	$2.5 + e^x$	$e^x \chi_{\{t \leq 0.25\}}(t) + \chi_{\{t > 0.25\}}(t)$	$[u(x, y, t) - y, u(x, y, t) + x]^T$	$[-y, x]^T$

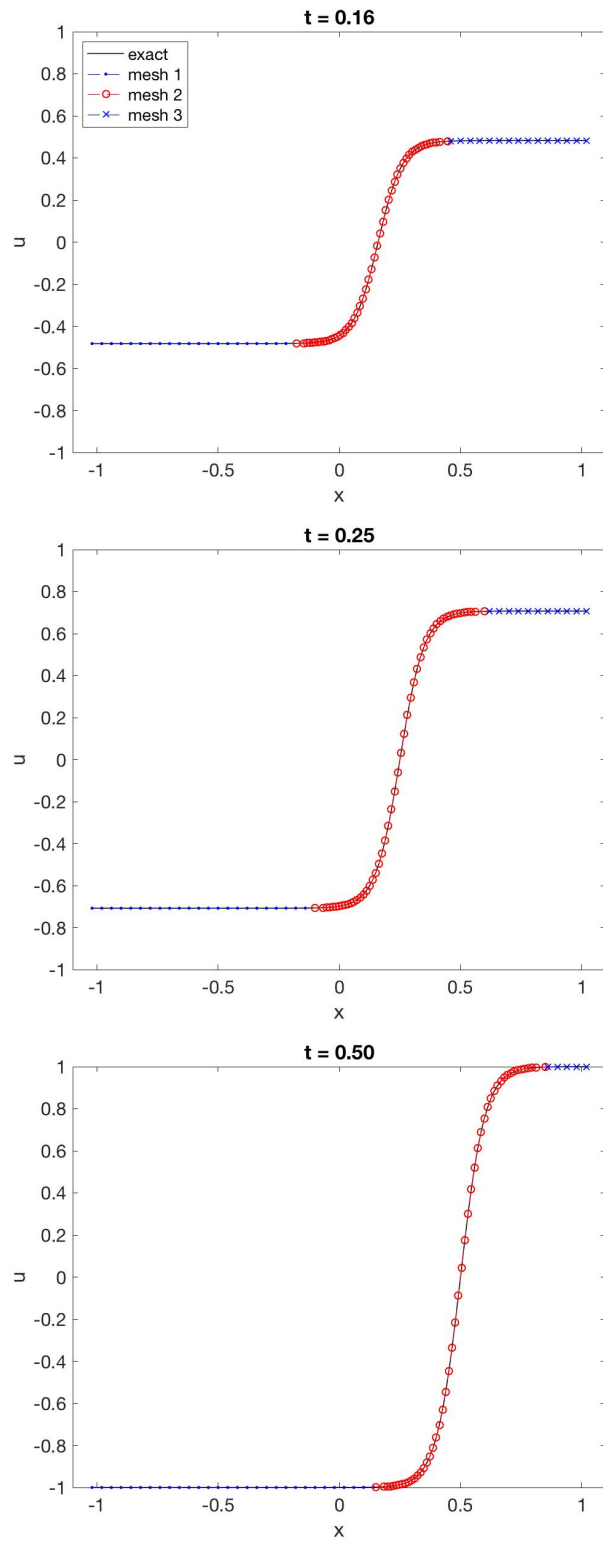


Fig. 5: Three time instants for the 1D test case *test2*. The circle markers define the nodes of the moving foreground mesh. The remaining dot and x markers are the nodes of the two background meshes.

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

	$T$	$h$	$L^\infty$ -err	$L^2$ -err	$L^\infty$ -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.

	$T$	$h$	$L^\infty$ -err		$L^2$ -err		$L^\infty$ -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>	$\pi$	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

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

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

558 **5.3. Relationship between the convective field and the foreground mesh velocity.** From the  
559 theoretical explanation of the method, it does not emerge in any way an interaction between the speed of the  
560 foreground grid  $\mathbf{V}$  and the intrinsic advective field  $\mathbf{a}$  of the problem. In other words, there does not seem to  
561 be a limitation of the velocity of the mesh that is displaced and deformed in terms of stability of the method.  
562 The unique limitation of the mesh speed (see section 3.5) is due to the CFL condition with respect to the  
563 dimension of the single cell. In order to allow to the code to perform the automatic information transmission,  
564 the mesh speed is such that it does not allow a given fringe cell  $\Omega_i^n$  in the foreground mesh to migrate beyond  
565 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  
566 fact, if this process is not ensured, for those new born cells belonging to the background mesh at time  $t^{n+1}$   
567 could not be able to recover the information from the polynomial interpolation. Consequently, the algorithm  
568 would incur a loss of information.

569 In this subsection we test on a numerical case that the stability is only given by the relative advective speed  
570  $\mathbf{a} - \mathbf{V}$  and the mesh velocity  $\mathbf{V}$  does not affect the stability of the method in other ways. In particular, on  
571 the same linear test case, we will consider different possible movements of the foreground mesh by measuring,  
572 at final time  $t = T$ , the  $L^\infty$ - 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  $\Delta t$  consist in the maximum CFL number and the maximum related time step  $\Delta t$  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		$\Delta t$		CFL		$\Delta t$	
	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.

<i>TEST3</i>	
$\Omega$	$(0, 1) \times (0, 5)$
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	$[0.8, 1.2]^2$
$\mathbf{V}$	$P1, P2, P3$

---

**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.

---

573 The tested case is named *TEST3* and it is summed up in Table 5 (top).

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

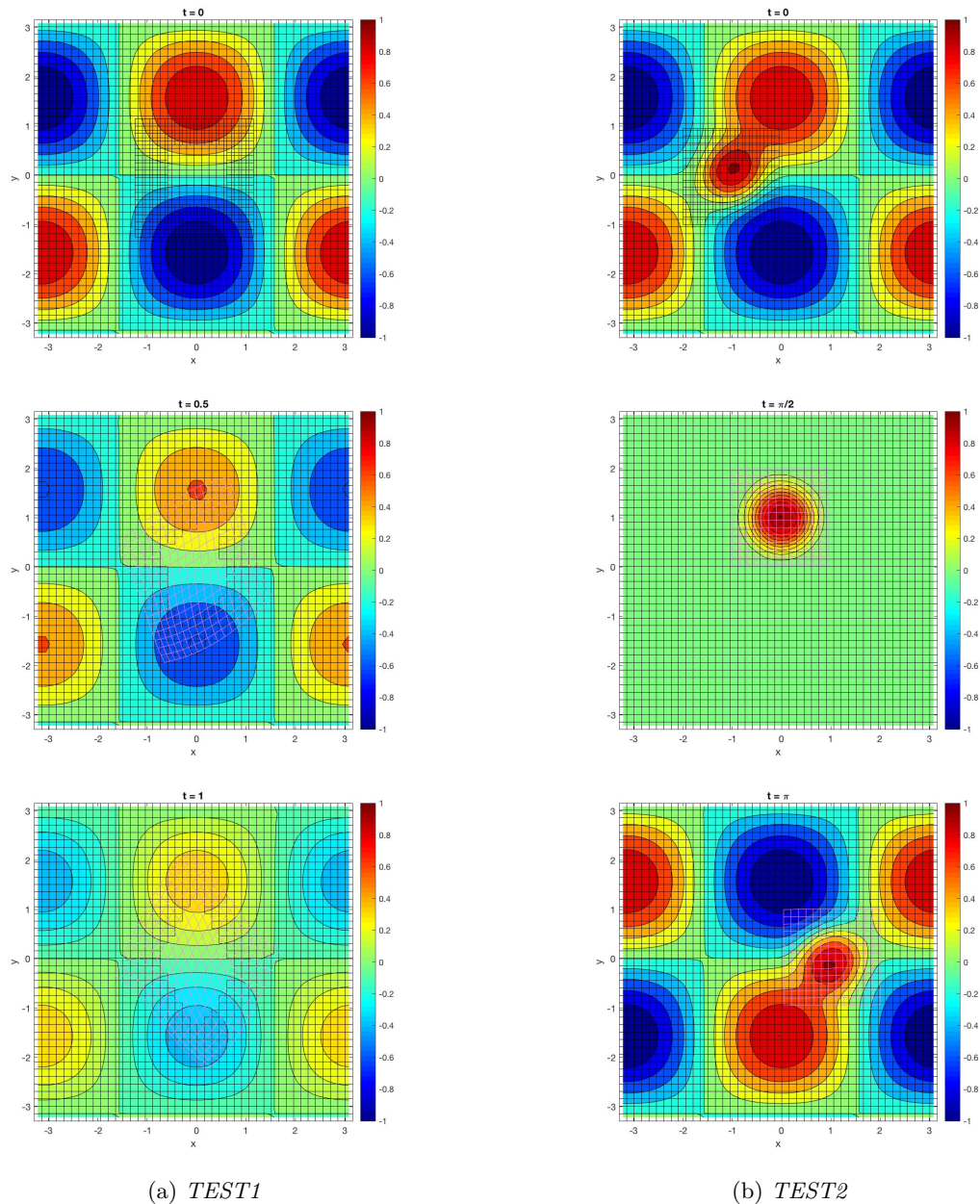


Fig. 6: Three time instants for test cases *TEST1* (a) and *TEST2* (b).

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

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



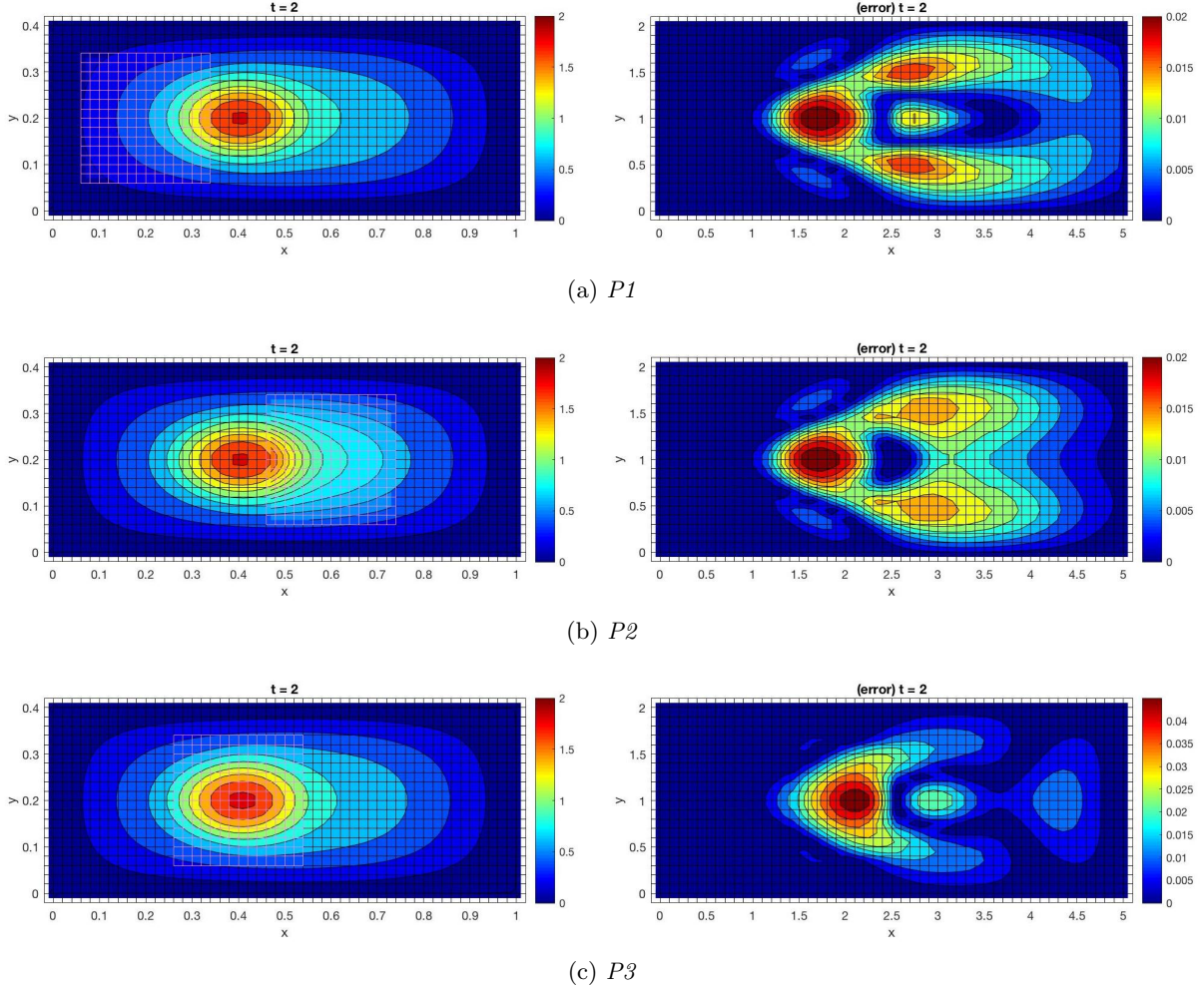


Fig. 7: The numerical solutions, on the left, at final time  $t = 2$  of the three possibilities  $P1$ ,  $P2$  and  $P3$  of foreground mesh movements for the *TEST3*. On the right there are the associated pointwise errors of the mismatch between the exact solution and the numerical solution.

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

	$L^\infty$ -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

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



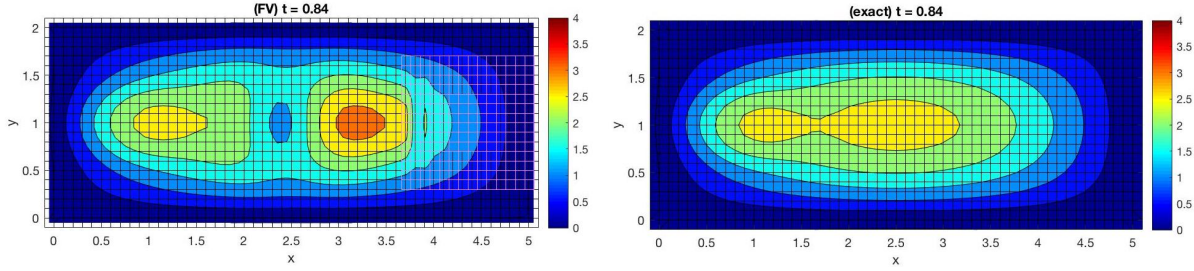


Fig. 8: Comparison between the numerical (left) and exact (right) solution of *TEST3* at time  $t = 0.84$  for a moving foreground mesh travelling with a speed generating a loss of information.

599 **5.4. Further topics.** We conclude this section by presenting three test cases that show the potentiality  
600 of the method. Firstly, a *nonlinear advection-diffusion system* is solved; successively a *multimesh* setting of  
601 grids is considered for the already described *TEST2* (see Table 1); finally, we consider a test case with a  
602 *complex domain* in which the foreground mesh is employed in order to adapt its shape to the shape of the  
603 domain.

604 **5.4.1. Nonlinear system.** Let  $\Omega = [-\pi, \pi]^2$  and  $T = 0.5$  be the computational domain and the final  
605 time, respectively. Thus the problem is: *find*  $\mathbf{u} : \Omega \times [0, T] \rightarrow \mathbb{R}^2$  *such that*:

$$606 \quad (5.4) \quad \begin{cases} \partial_t \mathbf{u} + \nabla \cdot (\mathbf{u}\mathbf{u}^T) = \nu \Delta \mathbf{u} + \mathbf{f} & \text{in } \Omega \times [0, T] \\ \mathbf{u} \equiv \mathbf{u}_{ex} & \text{on } \partial\Omega \times [0, T], \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_{ex}(\mathbf{x}, 0) & \text{in } \bar{\Omega} \times \{0\} \end{cases}$$

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

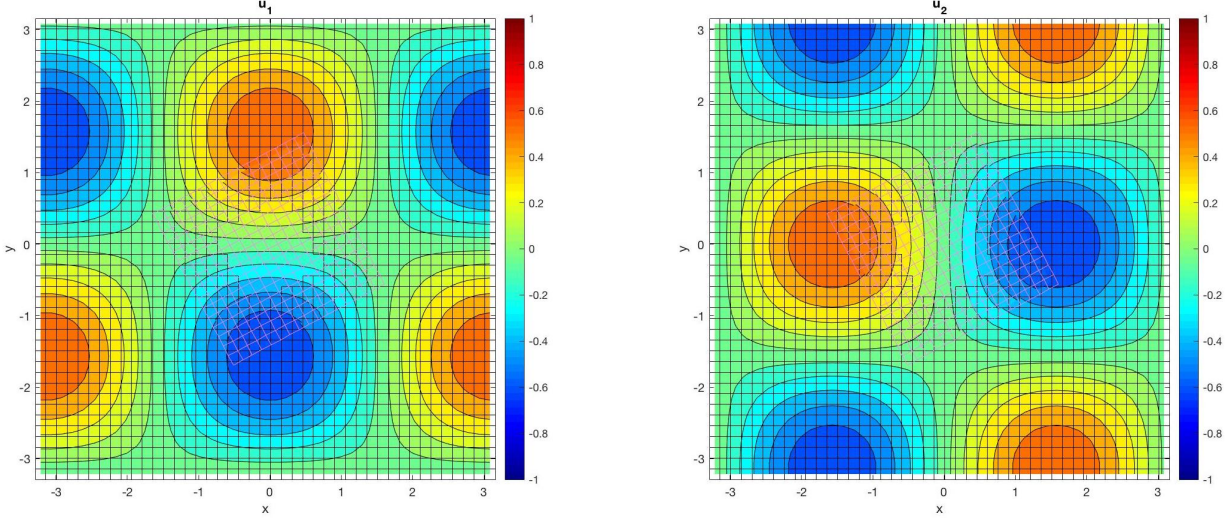
$$608 \quad \mathbf{u}_{ex}(x, y, t) = e^{-t} \begin{bmatrix} \cos(x) \sin(y) \\ -\sin(x) \cos(y) \end{bmatrix}.$$

609 In problem (5.4), the diffusive term  $\nu$  is equal to  $5\pi \times 10^{-3}$  while the convective field is represented by the  
610 solution itself, thus the partial differential equation is nonlinear. For this problem, the convective-diffusive  
611 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  
612 mass of the whole domain and it is allowed to rigidly counterclockwise rotate. In Figure 9 there are the two  
613 components of the numerical solution at final time  $t = T$ .

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

621 **5.4.2. Multimesh setting.** The presented method can be easily extended to more than one foreground  
622 mesh. As a matter of fact, different meshes can be set with an independent movement and such that to  
623 exchange information with the background grid and with the other moving foreground meshes. Due to the  
624 possibility to move, the foreground meshes can overlap each other. Consequently, the hole will be present in  
625 the background as well as in some foreground grids by properly applying the same dynamics of the overlap-  
626 ping zone of Section 3.5 to the specific intermediate foreground mesh.

627 In order to compare the performances of multimesh setting with two moving foreground meshes, we consid-  
628 ered the presented case *TEST2* with a foreground mesh clockwise rotating around the origin (see Table 1) by  
629 adding a second foreground mesh. The new grid is originally located to subset  $[-0.78, -0.18] \times [-0.62, -0.02]$   
630 and horizontally moves on the right with a constant velocity  $\mathbf{V}_2 = [-0.8, 0]^T$  (see Figure 10). The new grid  
631 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  $t = T = 0.5$ .

$h$	$L^\infty$ -err		$L^2$ -err		$L^\infty$ -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

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

640 **5.4.3. Complex domains.** An important application of chimera grids is the possibility to use meshes  
641 fitting the particular shape of the domain (which eventually evolves in time) by preserving a Cartesian  
642 background mesh. Here we present a test case summed up in Table 8. For any positive time  $t$ , let the generic  
643 moving ball formally be

$$644 \quad B(\rho_{\min}, \rho_{\max}; t) = \{(x, y) \in \mathbb{R}^2 : x = \rho \cos(\theta), y = \rho \sin(\theta) - 2t - \pi; \text{ with } (\rho, \theta) \in [\rho_{\min}, \rho_{\max}] \times [0, 2\pi]\}.$$

645 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  
646 radius equal to 0.5 is subtracted at any time  $t \in [0, T]$ . The circle vertically moves downwards with a  
647 constant velocity. In Figure 11a it is reported the numerical solution at the initial and final time instants  
648 for the numerical test. In Figure 11b there is a focus on the grid settings. For the foreground mesh, a polar  
649 structured grids is employed. It fits the shape of the domain and moves as the domain evolves.

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

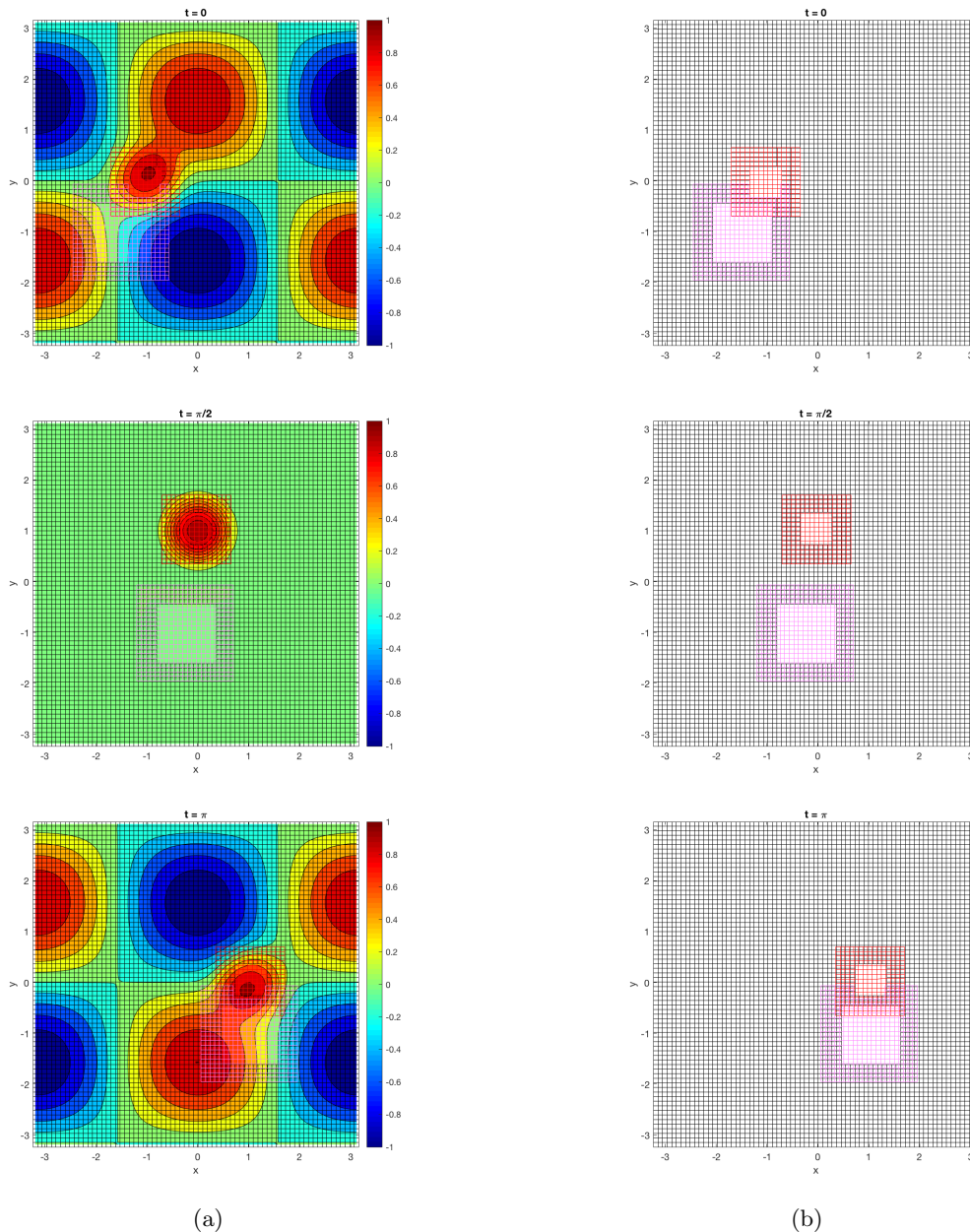


Fig. 10: On the left (a), the solution of *TEST2* for three time instants with a multimesh setting composed of two foreground meshes; on the right (b), for the same time instants, the configuration of the background and foreground grids.

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

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

Table 8: Summary scheme of  $TEST4$ .

$TEST4$	
$\Omega$	$[-\pi, \pi] \times [-2\pi, 2\pi]/B(0, 0.5; t)$
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	$B(0.5, 1.5; 0)$
$\mathbf{V}$	$[0, -2]^T$

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

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

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

669

## REFERENCES

- 670 [1] P. ANGOT, C.-H. BRUNEAU, AND P. FABRIE, *A penalization method to take into account obstacles in incompressible*  
 671 *viscous flows*, Numerische Mathematik, 81 (1999), pp. 497–520.
- 672 [2] J. W. BANKS, W. D. HENSHAW, A. K. KAPILA, AND D. W. SCHWENDEMAN, *An added-mass partition algorithm for fluid-*  
 673 *structure interactions of compressible fluids and nonlinear solids*, Journal of Computational Physics, 305 (2016),  
 674 pp. 1037–1064.
- 675 [3] J. W. BANKS, W. D. HENSHAW, AND B. SJÖGREEN, *A stable fsi algorithm for light rigid bodies in compressible flow*,  
 676 Journal of Computational Physics, 245 (2013), pp. 399–430.
- 677 [4] J. W. BANKS, D. W. SCHWENDEMAN, A. K. KAPILA, AND W. D. HENSHAW, *A high-resolution godunov method for*  
 678 *compressible multi-material flow on overlapping grids*, Journal of Computational Physics, 223 (2007), pp. 262–297.
- 679 [5] J. BENEK, P. BUNING, AND J. STEGER, *A 3-d chimera grid embedding technique*, in 7th Computational Physics Conference,  
 680 1985, p. 1523.
- 681 [6] W. BOSCHERI AND M. DUMBSER, *Arbitrary-lagrangian-eulerian one-step weno finite volume schemes on unstructured*  
 682 *triangular meshes*, Communications in Computational Physics, 14 (2013), pp. 1174–1206.
- 683 [7] S. BUSTO, S. CHIOCCHETTI, M. DUMBSER, E. GABURRO, AND I. PESHKOV, *High order ader schemes for continuum*  
 684 *mechanics*, Frontiers in Physics, 8 (2020), p. 32.
- 685 [8] C. E. CASTRO AND E. F. TORO, *Solvers for the high-order riemann problem for hyperbolic balance laws*, Journal of  
 686 Computational Physics, 227 (2008), pp. 2481–2513.
- 687 [9] G. CHESHIRE AND W. D. HENSHAW, *Composite overlapping meshes for the solution of partial differential equations*,  
 688 Journal of Computational Physics, 90 (1990), pp. 1–64.
- 689 [10] I.-T. CHIU AND R. MEAKIN, *On automating domain connectivity for overset grids*, in 33rd Aerospace Sciences Meeting  
 690 and Exhibit, 1995, p. 854.
- 691 [11] M. DUMBSER, W. BOSCHERI, M. SEMPLICE, AND G. RUSSO, *Central weighted eno schemes for hyperbolic conservation*  
 692 *laws on fixed and moving unstructured meshes*, SIAM Journal on Scientific Computing, 39 (2017), pp. A2564–A2591.
- 693 [12] M. DUMBSER, C. ENAUX, AND E. F. TORO, *Finite volume schemes of very high order of accuracy for stiff hyperbolic*  
 694 *balance laws*, Journal of Computational Physics, 227 (2008), pp. 3971–4001.
- 695 [13] M. DUMBSER AND M. KÄSER, *Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for*  
 696 *linear hyperbolic systems*, Journal of Computational Physics, 221 (2007), pp. 693–723.
- 697 [14] R. GLOWINSKI, T.-W. PAN, AND J. PERIAUX, *A fictitious domain method for dirichlet problem and applications*, Computer  
 698 Methods in Applied Mechanics and Engineering, 111 (1994), pp. 283–303.
- 699 [15] J. GUERRERO, *Overset composite grids for the simulation of complex moving geometries*, DICAT, University of Genoa,  
 700 Italy, (2006).
- 701 [16] W. D. HENSHAW, *On multigrid for overlapping grids*, SIAM Journal on Scientific Computing, 26 (2005), pp. 1547–1572.

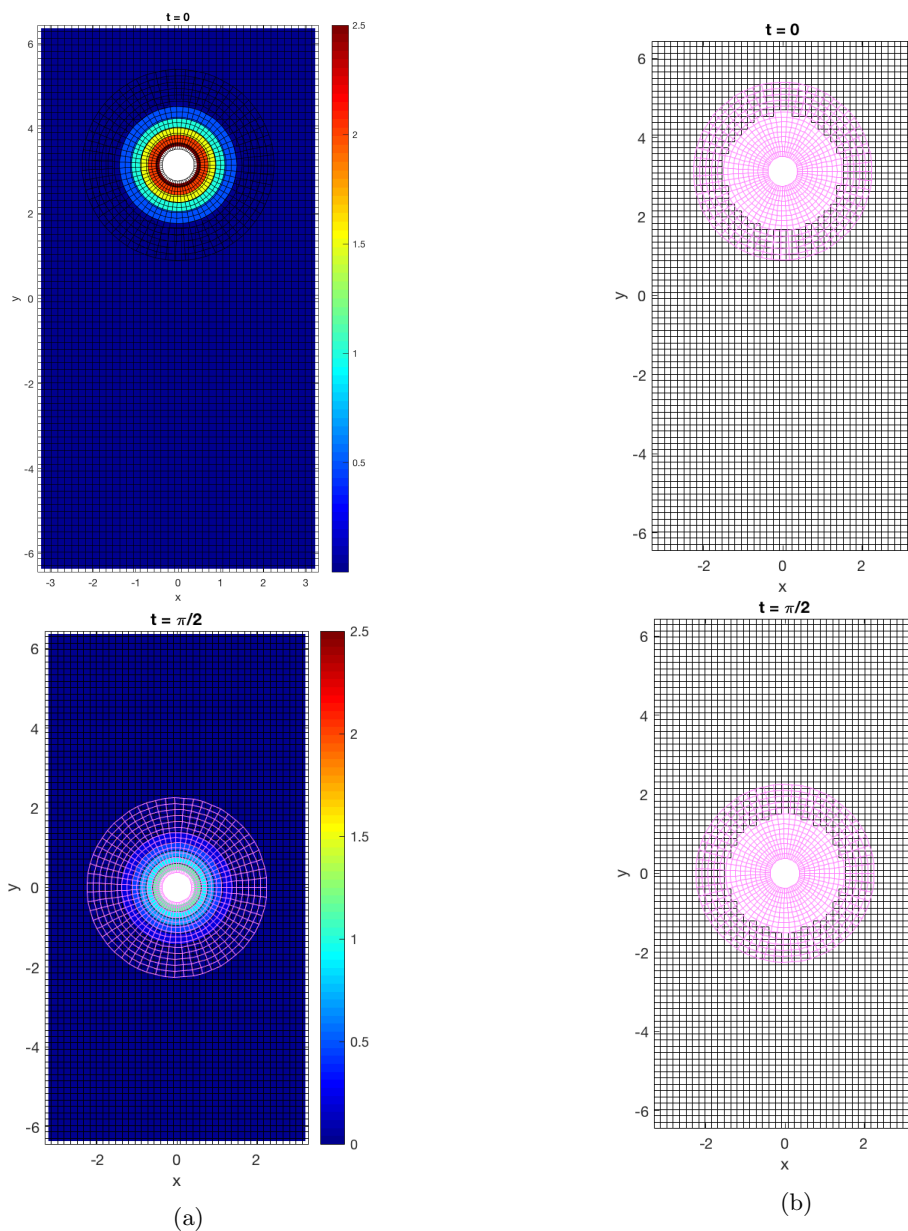


Fig. 11: On the left (a), the solution of  $TEST4$  for the initial and final time instants; on the right (b), for the same time instants, the background and foreground grids setting.

- 702 [17] A. HIDALGO AND M. DUMBSER, *Ader schemes for nonlinear systems of stiff advection–diffusion–reaction equations*, Journal  
703 of Scientific Computing, 48 (2011), pp. 173–189.
- 704 [18] C. W. HIRT, A. A. AMSDEN, AND J. COOK, *An arbitrary lagrangian-eulerian computing method for all flow speeds*, Journal  
705 of computational physics, 14 (1974), pp. 227–253.
- 706 [19] G. HOUZEUX, J. CAJAS, M. DISCACCIATI, B. EGUZKITZA, A. GARGALLO-PEIRÓ, M. RIVERO, AND M. VÁZQUEZ, *Domain  
707 decomposition methods for domain composition purpose: chimera, overset, gluing and sliding mesh methods*, Archives  
708 of Computational Methods in Engineering, 24 (2017), pp. 1033–1070.
- 709 [20] K.-H. KAO AND M.-S. LIOU, *Advance in overset grid schemes—from chimera to dragon grids*, AIAA journal, 33 (1995),  
710 pp. 1809–1815.
- 711 [21] M.-S. LIOU AND Y. ZHENG, *A novel approach of three-dimensional hybrid grid methodology: Part 2. flow solution*,  
712 Computer methods in applied mechanics and engineering, 192 (2003), pp. 4173–4193.
- 713 [22] R. MEAKIN, *Chapter 11: Composite overset structured grids*, Handbook of grid generation, JF Thompson, BK Soni and

- 714 NP Weatherill. CRC, (1999).
- 715 [23] G. I. MONTECINOS, *Numerical methods for advection-diffusion-reaction equations and medical applications*, PhD thesis,  
716 University of Trento, 2014.
- 717 [24] G. B. NAGY, O. E. ORTIZ, AND O. A. REULA, *The behavior of hyperbolic heat equations' solutions near their parabolic*  
718 *limits*, Journal of Mathematical Physics, 35 (1994), pp. 4334–4356.
- 719 [25] C. S. PESKIN, *The immersed boundary method*, Acta numerica, 11 (2002), pp. 479–517.
- 720 [26] N. A. PETERSSON, *Hole-cutting for three-dimensional overlapping grids*, SIAM Journal on Scientific Computing, 21 (1999),  
721 pp. 646–665.
- 722 [27] A. RAELI, M. BERGMANN, AND A. IOLLO, *A finite-difference method for the variable coefficient poisson equation on*  
723 *hierarchical cartesian meshes*, Journal of computational Physics, 355 (2018), pp. 59–77.
- 724 [28] D. SCHWENDEMAN, A. KAPILA, AND W. HENSHAW, *A study of detonation diffraction and failure for a model of compressible*  
725 *two-phase reactive flow*, Combustion Theory and Modelling, 14 (2010), pp. 331–366.
- 726 [29] G. STARIUS, *Composite mesh difference methods for elliptic boundary value problems*, Numerische Mathematik, 28 (1977),  
727 pp. 243–258.
- 728 [30] G. STARIUS, *Constructing orthogonal curvilinear meshes by solving initial value problems*, Numerische Mathematik, 28  
729 (1977), pp. 25–48.
- 730 [31] G. STARIUS, *On composite mesh difference methods for hyperbolic differential equations*, Numerische Mathematik, 35  
731 (1980), pp. 241–255.
- 732 [32] V. A. TITAREV AND E. F. TORO, *Ader: Arbitrary high order godunov approach*, Journal of Scientific Computing, 17  
733 (2002), pp. 609–618.
- 734 [33] V. A. TITAREV AND E. F. TORO, *Ader schemes for three-dimensional non-linear hyperbolic systems*, Journal of Compu-  
735 *tational Physics*, 204 (2005), pp. 715–736.
- 736 [34] E. F. TORO AND G. I. MONTECINOS, *Advection-diffusion-reaction equations: hyperbolization and high-order ader dis-*  
737 *cretizations*, SIAM Journal on Scientific Computing, 36 (2014), pp. A2423–A2457.
- 738 [35] E. VOLKOV, *The method of composite meshes*, Automatic Programming, Numerical Methods and Functional Analysis, 96  
739 (1970), p. 145.
- 740 [36] Z. WANG, *A fully conservative interface algorithm for overlapped grids*, Journal of Computational Physics, 122 (1995),  
741 pp. 96–106.