Flux-conserving treatment of non-conformal interfaces for finite-volume discretization of conservation laws

Enrico Rinaldi^a, Piero Colonna^b, Rene Pecnik^{a,*}

^a*Process & Energy Department, Delft University of Technology,
Leeghwaterstraat 39, 2628 CB, Delft, The Netherlands*

^b*Propulsion & Power, Delft University of Technology,
Kluyverweg 1, 2629 HS, Delft, The Netherlands*

Abstract

We present a new flux-conserving treatment of non-conformal mesh block interfaces for the numerical solution of conservation laws by high order finite volumes schemes. An auxiliary mesh is used at the interface to establish a connectivity between non-conformal blocks. The method does not involve any flux interpolation and conservation is therefore guaranteed by construction, without enforcing additional constraints. **Additionally, several gradient reconstructions across the interface have been adapted and their order of accuracy is studied analytically and numerically.** Applications to two and three dimensional fluid dynamic problems are considered, and the verification of the method is provided by comparing solutions obtained on single-block grids with no interfaces. Also the accuracy and numerical stability of the interface treatment is demonstrated empirically for a variety of test cases, such as a cylinder in supersonic cross-flow, low Mach-number vortex shedding, and a transonic turbine stage. The exemplary flow simulation problems are solved using explicit and implicit time integration schemes. The obtained results successfully demonstrate the effectiveness of the proposed method for stationary non-conformal blocks domains, as well as for mesh blocks in relative motion.

Keywords: Non-conformal interface; sliding interface; supermesh; flux conservation; finite-volume; conservation laws.

*Corresponding author

Email address: r.pecnik@tudelft.nl (Rene Pecnik)

1. Introduction

The numerical discretization of complex geometries is a challenging task which often requires the generation of unstructured or multi-block grids in order to guarantee a high quality of the resulting computational domain. Node-to-node connectivity among the discrete elements is required by standard numerical techniques and is guaranteed by most algorithms for mesh generation. However, there are applications for which the grid must be split into subregions, or patches, with non-matching interface boundaries. Typical examples are found in aeroelasticity simulations, where the aerodynamic and structural interface nodes often do not coincide, or multiphysics simulations. A similar situation occurs when parts of the domain are in relative motions with respect to each other, as it is the case for turbomachinery computations.

The appropriate treatment of the artificially introduced internal boundaries is crucial. The most critical aspects such a treatment has to comply with are: numerical conservation, stability, accuracy and efficiency. In general, the methods can be categorized based on their numerical conservation properties, namely they can be subdivided in conservative and non-conservative schemes.

Berger [1] was among the first to propose a method to conserve variables at grid interfaces. Rai [2–4] developed a conservative zonal-boundary scheme for grids having a common cell-centered line at the interface, which was later applied to study the two-dimensional stator/rotor interaction of an axial turbine stage [5]. Lerat and Wu [6] proposed a patched grid technique based on a common cell-side line instead. Interface cell faces are sub-divided into smaller segments on which the fluxes are calculated and then summed together in order to obtain the total numerical flux, thus avoiding any interpolation. Lerat and Wu’s method was later successfully applied by Benkenida *et al.* [7] to calculate transport of vortices. Zhang *et al.* [8] proposed instead a polynomial remapping of the variables on the non-matching faces, enforcing conservation in the definition of the polynomials coefficients. An alternative approach which ensures conservation is obtained by introducing a local mesh at the boundary between domains, whose purpose is to correctly “attach” the non-conformal patches, documented in Refs. [9–11], where CFD applications are treated, and in Refs. [12, 13], which in turn are concerned with multiphysics simulations.

Non-conservative schemes suffer from possible loss of accuracy and stability issues, especially

when discontinuities, e.g., shock waves, occur across the interface. One of the most widely used non-conservative scheme is the CHIMERA interpolation technique on overlapping grids in its original formulation [14, 15]. As acknowledged by the authors, problems were encountered in case of shock/interface interaction. Wang [16] extended the scheme to a conservative formulation which replaces the overlapped zones by patched zones. Another modification to the original CHIMERA method is the one proposed by Kao and Liou [17]: they replaced the grid overlapping with a newly defined unstructured grid. The conservation error for the CHIMERA scheme was analyzed by Wang *et al.* [18], who showed that it is a first order term of the mesh elements size if a second order conservative scheme is used, and if discontinuities are not in the proximity of the overlapping area interface. In addition, they also showed that the non-conservative formulation might cause inaccurate solutions for smooth flows too, and pointed to the benefits of using a conservative formulation. Overlapping grids were also considered by Liu [19], who extended the central scheme of Kurganov and Tadmor [20] to account for the new mesh topology. Other alternative non-conservative interface methods were successfully applied to CFD calculations on complex geometries such as a complete aircraft configuration [21, 22]. Tang and Zhou [23] rigorously investigated the convergence and accuracy of non-conservative schemes showing that the conservation error is bounded if the solution itself is bounded.

This paper presents a new flux-conserving treatment of non-conformal interfaces for the numerical solution of conservation laws by high order finite volumes schemes. The method is based on the concept of a supermesh introduced by Farrell *et al.* [24] and later applied to the conservative interpolation of cell-centered finite volume variables by Menon and Schmidt [25]. The supermesh acts as an auxiliary interface grid that establishes connectivity between the mesh blocks. The numerical fluxes are evaluated on the supermesh and then summed to the flux balance of the control volumes of their respective parent meshes.

Section 2 introduces the supermesh and describes its construction. The flux assembly technique for the internal control volumes and at the interface is described in Sec. 3. The method automatically guarantees flux conservation by construction, without any interpolation or the need to enforce additional constraints. The challenge of an accurate gradient reconstruction at the interface control volumes, which is needed for high order spatial accuracy, is addressed in Sec. 4.

58 Modifications to the least-squares and Green–Gauss methods are proposed, and their order of ac-
59 curacy is derived analytically and verified numerically on two and three dimensional domains.
60 The parallelization of the supermesh generation and of the interface treatment is discussed in
61 Sec. 5. The verification of the proposed interface treatment is provided by solving the Navier–
62 Stokes equations. Several two and three dimensional numerical tests are presented and discussed
63 in Sec. 6. Numerical stability is shown empirically using explicit and implicit time integration
64 schemes. Finally, conclusions are drawn in Sec. 7.

2. Non-conformal mesh interface treatment using a supermesh

2.1. Supermesh definition

The auxiliary grid between two non-conformal mesh blocks, as displayed in Fig. 1, is constructed using the supermesh methodology. The most important properties of the supermesh, as described in Refs. [24, 25], are reported in the following for the sake of completeness. Let $\mathcal{M}_A$ and $\mathcal{M}_B$ be two arbitrary surface meshes of the same geometrical domain $\Omega \subset \mathbb{R}^3$. They will be referred to as *parent* meshes. $\mathcal{N}_A$ and $\mathcal{N}_B$ are the nodes of $\mathcal{M}_A$ and $\mathcal{M}_B$, respectively, and $\mathcal{E}_A$ and $\mathcal{E}_B$ the elements (faces in this case) of the two meshes.

Definition 1. A supermesh $\mathcal{M}_S$ of $\{\mathcal{M}_A, \mathcal{M}_B\}$ is a mesh of Ω which satisfies the following properties:

- $\mathcal{N}_S \supseteq \mathcal{N}_A \cup \mathcal{N}_B$;
- $A(\mathcal{E}_S \cap \mathcal{E}) \in \{0, A(\mathcal{E}_S)\} \quad \forall \mathcal{E}_S \in \mathcal{M}_S, \mathcal{E} \in \mathcal{M}, \mathcal{M} \in \{\mathcal{M}_A, \mathcal{M}_B\}$;

where A is the surface area function. ■

The first property asserts that the supermesh contains every node of the parent meshes. The second asserts that the intersection area of each element of the supermesh $\mathcal{E}_S$ with any element of a parent mesh must either be null or measure the whole supermesh element area $A(\mathcal{E}_S)$. In other words, it is possible to reconstruct every element of a parent mesh using an appropriate number of supermesh elements.

2.2. Supermesh construction

The supermesh construction consists of an intersection problem between $n - 1$ dimensional elements, where n is the number of dimensions of the grid blocks. The task is trivial in case of 2D meshes, whereby the supermesh is made of segments demarcated by the set of nodes $\mathcal{N}_S$, which coincides with $\mathcal{N}_A \cup \mathcal{N}_B$. This is valid for every 1D intersection problem, as depicted in Fig. 1(a).

In case of 3D mesh blocks, the supermesh definition consists of a 2D face intersection problem between two interface surface meshes. Figure 1(b) shows a three-dimensional mesh consisting of two blocks with non-conformal interface elements. $\mathcal{M}_A$ and $\mathcal{M}_B$ are the surface meshes on the

91 patched boundary faces. Unstructured quadrilateral and triangular elements are used on $\mathcal{M}_A$ and
 92 $\mathcal{M}_B$, respectively. $\mathcal{M}_S$ denotes the interface supermesh, which is an hybrid unstructured grid.

93 In order to satisfy the properties of Definition 1 (discussed in Sec. 2.1) and guarantee flux-
 94 conservation across the interface, it is crucial that the intersection scheme used to calculate $\mathcal{M}_S$
 95 is accurate and robust. Exact numerical conservation can only be achieved if the intersection is
 96 computed to machine precision. The algorithm used in this study to calculate the intersection
 97 between two generic parent mesh elements, $\mathcal{E}_{A,i}$ and $\mathcal{E}_{B,j}$, is described graphically in Fig. 2 and
 98 consists of four steps:

- 99 1. identify the vertices of $\mathcal{E}_{A,i}$ contained within $\mathcal{E}_{B,j}$, and vice versa;
- 100 2. calculate the intersection between the edges of $\mathcal{E}_{A,i}$ and $\mathcal{E}_{B,j}$;
- 101 3. remove double points from step 1 and 2;
- 102 4. if the number of calculated nodes ($\mathcal{N}_{S,ij}$) after step 3 is larger than 2, define new supermesh
 103 elements.

104 At step 3, two nodes N_1 and N_2 are considered to coincide if $|\mathbf{x}_{N_1} - \mathbf{x}_{N_2}| < 10^{-14}$. Step 3 addresses
 105 all the special situations stemming from coincidental nodes or edges. In general, the intersection
 106 procedure may lead to small elements of size comparable to machine precision, which however
 107 does not pose any limitations in the flux calculation and in the choice of the time-step, the latter
 108 being computed using the volume of the mesh cells of the original blocks.

109 Several exemplary face intersections are shown in Fig. 3 in order to characterize common
 110 situations usually encountered. If $\mathcal{N}_{S,ij} = 3$ the supermesh element $\mathcal{E}_{S,k}$ is a triangle, Fig. 3(a),
 111 while if $\mathcal{N}_{S,ij} = 4$, $\mathcal{E}_{S,k}$ is a quadrilateral element, Fig. 3(b). The situation in which the calculated
 112 supermesh element $\mathcal{E}_{S,k}$ coincides either with $\mathcal{E}_{B,j}$ or with $\mathcal{E}_{A,i}$ is depicted in Fig. 3(c) and 3(d),
 113 respectively. Finally, if $\mathcal{N}_{S,ij} > 4$, multiple elements are defined, Fig. 3(e) and 3(f). In order
 114 to keep the number of supermesh elements to a minimum, it is convenient to define as many
 115 quadrilateral elements as possible at each face intersection, to limit the use of triangular elements
 116 when an odd number of nodes is available.

117 The intersection procedure discussed can be generalized to an arbitrary surface mesh, not nec-
 118 essarily planar as in Fig. 1(b), by parametrizing the parent meshes using a common local coordi-

119 nate system, and by computing the intersections in the local frame of reference.

(a) 2D mesh blocks, 1D parent meshes interpolation. Empty circles indicate the cell centers of the elements, while the black circles denote N_S .

(b) 3D mesh blocks, 2D parent meshes and interpolation.

Figure 1: Interface supermesh definition for 2D and 3D parent grids. In both cases the mesh blocks are displaced for the sake of clarity and the interface supermesh is displayed in the middle.

Figure 2: Necessary steps to calculate supermesh elements by intersecting two generic elements $\mathcal{E}_{A,i}$ and $\mathcal{E}_{B,j}$.

Figure 3: Exemplary intersections of 2D elements.

120 3. Conservative flux assembly at the interface

121 In this work we consider finite volume integration scheme for conservation laws written as

$$\frac{\partial}{\partial t} \int_{\Omega} \mathbf{U} d\Omega + \int_{\partial\Omega} \mathbf{F}(\mathbf{U}) dA = \mathbf{0}, \quad (1)$$

122 where $\mathbf{U}$ is the vector of the conservative variables and $\mathbf{F}(\mathbf{U})$ the flux vector, which is in general a
 123 non-linear function of $\mathbf{U}$. The physical domain of interest is denoted by Ω and its external boundary
 124 by $\partial\Omega$. In the finite-volume framework, Ω is discretized numerically into control volumes whose
 125 total number is indicated by N_{cv} . Equation (1) can be re-written in the semi-discrete form in the
 126 cell center of each control volume cv as

$$\frac{\partial \mathbf{U}_{cv}}{\partial t} = -\mathbf{RHS}(\mathbf{U}_{cv}) \quad (2)$$

127 where the discretized flux balance is casted to the right-hand-side (**RHS**) of the equations as

$$\mathbf{RHS}(\mathbf{U}_{cv}) = -\frac{1}{V_{cv}} \sum_{fa=1}^{N_{fa}} \mathbf{F}(\mathbf{U}_L, \mathbf{U}_R) A_{fa} - \frac{1}{V_{cv}} \sum_{fa=1}^{N_{bfa}} \mathbf{F}(\mathbf{U}_L, \mathbf{U}_{bc}) A_{fa}. \quad (3)$$

128 The nomenclature used in Eq. (3) is displayed in Fig. 4. Here, V_{cv} indicates the volume of cv and
 129 N_{fa} and N_{bfa} the number of discrete faces defining the control volume, which can be either internal
 130 or on the boundary of the domain (N_{bfa}). A_{fa} denotes the area of the face fa . The value of the
 131 face flux depends on the conservative variable states on the left and on the right side of the face,
 132 $\mathbf{U}_L$ and $\mathbf{U}_R$, and is considered positive if it points outwards of the control volume. A first order
 133 accurate scheme is obtained if $\mathbf{U}_L$ and $\mathbf{U}_R$ are assumed equal to the values in the cell centers of the
 134 respective control volumes, namely if $\mathbf{U}_L = \mathbf{U}_{cv}$ and $\mathbf{U}_R = \mathbf{U}_{nb}$, where nb indicates the neighbor of
 135 cv sharing the face fa on which the flux is calculated. The total number of internal neighbors for
 136 each cv is denoted by N_{nb} and satisfies $N_{nb} = N_{fa} - N_{bfa}$. In case one of the faces is a boundary
 137 face, the boundary condition $\mathbf{U}_{bc}$ is used in the flux calculation.

138 The proposed interface treatment uses the supermesh to replace the parent meshes $\mathcal{M}_A$ and
 139 $\mathcal{M}_B$ in the flux balance calculation of the interface control volumes. At each $\mathcal{E}_S$, the left and right
 140 states used to evaluate the supermesh face flux are taken from the corresponding $\mathcal{E}_A$ and $\mathcal{E}_B$. To
 141 this end, it is necessary to map the supermesh on the parent meshes. For every $\mathcal{E}_S$, $\mathcal{T}_A$ and $\mathcal{T}_B$

Figure 4: Schematic of the finite volume discretization.

142 contain the index of the parent mesh control volume, such that

$$\mathcal{T}_{A,i} = \mathcal{P}(k) \quad \text{if} \quad A(\mathcal{E}_{S,i} \cap \mathcal{E}_{A,k}) = A(\mathcal{E}_{S,i}), \quad (4)$$

$$\mathcal{T}_{B,i} = \mathcal{P}(k) \quad \text{if} \quad A(\mathcal{E}_{S,i} \cap \mathcal{E}_{B,k}) = A(\mathcal{E}_{S,i}), \quad (5)$$

143 where i and k indicate the index of a generic supermesh and parent mesh element, respectively.

144 $\mathcal{P}(k)$ returns the index of the control volume cv , which contains the boundary face k . The flux

145 balance for an interface control volume belonging to $\mathcal{M}_A$ on a two-dimensional structured grid is

146 illustrated in Fig. 5. The fluxes computed on the supermesh faces contribute to the flux balance as

$$\mathbf{RHS}_{int}(\mathbf{U}_{cv}) = \mathbf{RHS}(\mathbf{U}_{cv}) - \frac{1}{V_{cv}} \sum_{sf=1}^{N_{sf}} \mathbf{F}(\mathbf{U}_{cv}, \mathbf{U}_{\mathcal{T}_{B,sf}}) A_{sf}, \quad (6)$$

147 where N_{sf} is the number of supermesh elements that are mapped on cv (for $cv = 6$ they are two,

148 indicated by the indices 9 and 10), and A_{sf} is the area of the single supermesh face. The same

149 formula is applied to a control volume contained in $\mathcal{M}_B$ by replacing $\mathcal{T}_{B,sf}$ with $\mathcal{T}_{A,sf}$. The flux

150 integral conservation over the whole interface is guaranteed by the second property of Definition

151 1, provided that the interpolation method is accurate to machine precision.

Figure 5: Schematic showing the flux balance calculation for the interface control volume $cv = 6$. For the case displayed $\mathcal{T}_A = (4, 5, 5, 6, 6, 7)$ and $\mathcal{T}_B = (8, 8, 9, 9, 10, 10)$, thus the neighbors of cv that contribute to the flux balance calculation are $nb = (7, 2, 5, 9, 10)$.

4. Interface gradient reconstruction

In order to achieve second or higher order spatial accuracy, the states are extrapolated from the cell centers to the faces using an appropriate gradient reconstruction scheme. Let us consider the exemplary unstructured mesh depicted in Fig. 4; a generic scalar quantity ϕ is calculated at a cell face as

$$\phi_{fa} = \phi_{cv} + \psi_{cv} \nabla \phi_{cv} \cdot (\mathbf{x}_{fa} - \mathbf{x}_{cv}), \quad (7)$$

whereby $\nabla \phi_{cv} = (\partial_x \phi_{cv}, \partial_y \phi_{cv}, \partial_z \phi_{cv})$ is the gradient of the scalar in the cell center, and the partial derivatives are written in the compact form $\partial_x = \partial/\partial x$, $\partial_y = \partial/\partial y$, $\partial_z = \partial/\partial z$. The location of the face and control volume centroids are denoted by $\mathbf{x}_{fa}$ and $\mathbf{x}_{cv}$, respectively, and ψ_{cv} is a slope limiter function which prevents spurious oscillations in the regions where ϕ varies sharply. The number of control volumes used to approximate the gradient at each grid cell center is referred to as *stencil* and depends on the order of accuracy of the scheme and on the scheme itself. In this paper, we consider second order methods only and we limit our discussion to the least-squares (LS) and Green–Gauss (GG) gradient reconstruction schemes.

In the following, we analyze the LS and GG methods as formulated on the internal cells of the computational domain, and we discuss their modification to be adapted to the interface problem. We make use of structured mesh blocks to describe the schemes and to study their order of accuracy. The reasons for choosing two structured blocks are to simplify the analytical analysis of the order of accuracy (presented in Sec. 4.3), and to prevent the numerical results of Sec. 4.4 from being polluted by the quality of the internal control volumes. Hereafter, we will refer to the structured grid blocks depicted in Fig. 6. In order to be consistent with the equations introduced in Sec. 3, the naming of the elements is the same as the one adopted for the unstructured grids.

4.1. Least-squares gradient

The LS method approximates the gradient in the center of a control volume by minimizing an objective function in the form (see e.g. Mavriplis [26])

$$J = \sum_{nb=1}^{N_{nb}} w_{nb}^2 E_{nb}^2, \quad (8)$$

(a) Internal control volume.

(b) Interface: extended stencil.

(c) Interface: face value.

Figure 6: Stencil for the least-squares and Green–Gauss gradient reconstruction in the cell centroid $cv = 0$.

176 where $w_{nb}^2 = 1/|\mathbf{x}_{nb} - \mathbf{x}_{cv}|^2$ and E_{nb} denotes the error

$$E_{nb}^2 = \left(-(\phi_{nb} - \phi_{cv}) + \partial_x \phi_{cv}(x_{nb} - x_{cv}) + \partial_y \phi_{cv}(y_{nb} - y_{cv}) + \partial_z \phi_{cv}(z_{nb} - z_{cv}) \right)^2. \quad (9)$$

177 The minimization of Eq. (8) leads to

$$\begin{pmatrix} \partial_x \phi_{cv} \\ \partial_y \phi_{cv} \\ \partial_z \phi_{cv} \end{pmatrix} = \begin{pmatrix} S_{w^2 dx^2} & S_{w^2 dx dy} & S_{w^2 dx dz} \\ S_{w^2 dx dy} & S_{w^2 dy^2} & S_{w^2 dy dz} \\ S_{w^2 dx dz} & S_{w^2 dy dz} & S_{w^2 dz^2} \end{pmatrix}^{-1} \begin{pmatrix} \sum_{nb=1}^{Nnb} w_{nb}^2 (x_{nb} - x_{cv})(\phi_{nb} - \phi_{cv}) \\ \sum_{nb=1}^{Nnb} w_{nb}^2 (y_{nb} - y_{cv})(\phi_{nb} - \phi_{cv}) \\ \sum_{nb=1}^{Nnb} w_{nb}^2 (z_{nb} - z_{cv})(\phi_{nb} - \phi_{cv}) \end{pmatrix}. \quad (10)$$

178 The coefficients of the matrix depend on the geometry of the mesh only, and can be efficiently
179 calculated and stored in a pre-processing step. They are defined as

$$S_{w^2 dx dy} = \sum_{nb=1}^{Nnb} w_{nb}^2 (x_{nb} - x_{cv})(y_{nb} - y_{cv}), \quad (11)$$

180 and similarly for $S_{w^2 dx^2}$, $S_{w^2 dy^2}$, and $S_{w^2 dz^2}$, $S_{w^2 dx dz}$, and $S_{w^2 dy dz}$.

181 In case at least one of the faces enclosing cv is part of $\partial\Omega$, the contribution of the boundary
182 faces must be included in the objective function, as

$$J = \sum_{nb=1}^{Nnb} w_{nb}^2 E_{nb}^2 + \sum_{fa=1}^{Nbfa} w_{fa}^2 E_{fa}^2, \quad (12)$$

183 with $w_{fa}^2 = 1/|\mathbf{x}_{fa} - \mathbf{x}_{cv}|^2$, and

$$E_{fa}^2 = \left(-(\phi_{fa} - \phi_{cv}) + \partial_x \phi_{cv}(x_{fa} - x_{cv}) + \partial_y \phi_{cv}(y_{fa} - y_{cv}) + \partial_z \phi_{cv}(z_{fa} - z_{cv}) \right)^2. \quad (13)$$

184 The values of the scalar on the external faces are specified as boundary conditions. The minimiza-
185 tion of Eq. (12) results in an expression for $\nabla \phi_{cv}$ similar to Eq. (10).

186 4.1.1. Extended stencil

187 An intuitive possibility is to extend the list of neighbors of a generic control volume cv by
188 adding neighbors across the interface that share a face in the supermesh, and to make use of
189 Eq. (10) with the updated connectivity **(LSI-F)**. A generic cv in the mesh patch containing $\mathcal{M}_A$ has
190 a number of new neighbors which corresponds to the number of supermesh elements satisfying
191 $\mathcal{T}_{A,i} = cv$. The index which identifies the new neighbor belonging to the mesh patch containing
192 $\mathcal{M}_B$ is given by $\mathcal{T}_{B,i}$.

Figure 6(b) shows the extended stencil for the interface control volume contained in $\mathcal{M}_A$ on a structured mesh; the cells 4 and 5 are added to the list $nb = 1, 2, 3$. Although, the two mesh blocks are both structured cartesian grids, the interface stencil is skewed due to the non-conformity of the grids, resembling the typical stencil used on an unstructured mesh. In case the elements of $\mathcal{M}_B$ are much smaller than the ones of $\mathcal{M}_A$, the stencil is strongly biased toward $\mathcal{M}_B$. This leads to a deterioration of the gradient accuracy not only along the direction which crosses the interface, but also along the direction parallel to it. To avoid this biasing, only the closest neighbor can be considered (LSI-C).

4.1.2. Approximate face value

An alternative approach consists of treating the faces of the control volume on the interface as external boundary faces whose value is computed as

$$\phi_{fa} = \frac{1}{A(\mathcal{E}_{A,k})} \sum_{i: \mathcal{T}_{A,i}=cv} A(\mathcal{E}_{S,i}) \frac{\phi_{cv} d_{\mathcal{T}_{B,i}} + \phi_{\mathcal{T}_{B,i}} d_{cv,i}}{d_{cv,i} + d_{\mathcal{T}_{B,i}}}, \quad (14)$$

where $\mathcal{E}_{A,k}$ is the k -th element of the parent mesh $\mathcal{M}_A$, which corresponds to the boundary face fa of cv on the interface. This approach will be referred to as LSI-Fa in the following. Note that $\sum_{i: \mathcal{T}_{A,i}=cv} A(\mathcal{E}_{S,i}) = A(\mathcal{E}_{A,k})$ by definition. $d_{cv,i}$ and $d_{\mathcal{T}_{B,i}}$ denote the distances between the face centers of the supermesh elements i and the cell centers of cv and $\mathcal{T}_{B,i}$, respectively. Equation (14) assumes that cv belongs to the grid block containing $\mathcal{M}_A$. If cv is on the grid block containing $\mathcal{M}_B$, the indices A and B must be interchanged. Considering the two-dimensional case depicted in Fig. 6(c), Eq. (14) reduces to

$$\phi_{fa} = \frac{a(\phi_4 d_{0,1} + \phi_0 d_4)}{(d_{0,1} + d_4)} + \frac{(1-a)(\phi_5 d_{0,2} + \phi_0 d_5)}{(d_{0,2} + d_5)}. \quad (15)$$

4.2. Green–Gauss gradient

The GG method numerically approximates the Green–Gauss theorem as

$$\nabla \phi_{cv} = \frac{1}{V_{cv}} \sum_{fa=1}^{N_{fa}} \phi_{fa} A_{fa} \mathbf{n}_{fa}, \quad (16)$$

with $\mathbf{n}_{fa}$ indicating the unit vector normal to the face. In second order cell-centered numerical schemes ϕ_{fa} is commonly linearly interpolated as

$$\phi_{fa} = \frac{\phi_{cv}d_{nb} + \phi_{nb}d_{cv}}{d_{nb} + d_{cv}}, \quad (17)$$

where $d_{cv} = |\mathbf{x}_{cv} - \mathbf{x}_{fa}|$ and $d_{nb} = |\mathbf{x}_{nb} - \mathbf{x}_{fa}|$. The GG gradient only requires the face value on the interface boundary for the non-conformal interface problem (GGI). This is done by using Eq. (14). No additional modification to the standard scheme is required.

4.3. Analytical order of accuracy

The analytical derivation of the order of accuracy for the schemes introduced above is given in this section. The procedure is based on the Taylor series expansion of the solution in the neighboring cells, which is substituted into the schemes described above. The order of accuracy is quantified by the order of magnitude of the truncation error.

In case of internal control volumes (no interface), LS and GG are equivalent on a structured grid

$$\begin{pmatrix} \partial_x^{LS} \phi_0 \\ \partial_y^{LS} \phi_0 \end{pmatrix} \equiv \begin{pmatrix} \partial_x^{GG} \phi_0 \\ \partial_y^{GG} \phi_0 \end{pmatrix} = \begin{pmatrix} \frac{\phi_4 - \phi_0}{2\Delta x_4} + \frac{\phi_2 - \phi_0}{2\Delta x_2} \\ \frac{\phi_3 - \phi_0}{2\Delta y_3} + \frac{\phi_1 - \phi_0}{2\Delta y_1} \end{pmatrix}, \quad (18)$$

with $\Delta x_{nb} = x_{nb} - x_0$ and $\Delta y_{nb} = y_{nb} - y_0$. In Eq. (18) and in the following it is assumed that the unit normal vectors to the faces of $cv = 0$ point outwards from the control volume. The length of the faces A_h and A_v is calculated as $A_h = x_{fa,4} - x_{fa,2}$ and $A_v = y_{fa,3} - y_{fa,1}$, where fa, nb is the face shared by $cv = 0$ and the neighbor nb . It follows that $x_0 = (x_{fa,2} + x_{fa,4})/2$ and $y_0 = (y_{fa,1} + y_{fa,3})/2$.

The function in the neighbors cell centers is approximated by the Taylor series expansion as

$$\phi_{nb} = \phi_0 + \partial_x \phi_0 \Delta x_{nb} + \partial_y \phi_0 \Delta y_{nb} + \partial_{xx}^2 \phi_0 \frac{\Delta x_{nb}^2}{2} + \partial_{yy}^2 \phi_0 \frac{\Delta y_{nb}^2}{2} + \partial_{xy}^2 \phi_0 \Delta x_{nb} \Delta y_{nb} + O(\Delta x^3) + O(\Delta y^3). \quad (19)$$

Substituting the Taylor expansions into Eq. (18) leads to

$$\begin{pmatrix} \partial_x^{LS} \phi_0 \\ \partial_y^{LS} \phi_0 \end{pmatrix} \equiv \begin{pmatrix} \partial_x^{GG} \phi_0 \\ \partial_y^{GG} \phi_0 \end{pmatrix} = \begin{pmatrix} \partial_x \phi_0 + \partial_{xx}^2 \phi_0 \frac{\Delta x_4 + \Delta x_2}{4} + O(\Delta x^2) \\ \partial_y \phi_0 + \partial_{yy}^2 \phi_0 \frac{\Delta y_3 + \Delta y_1}{4} + O(\Delta y^2) \end{pmatrix}, \quad (20)$$

which shows that the methods are in general first order accurate on non-uniform grids and second order accurate in case either $\partial_{xx}^2 \phi_0 = \partial_{yy}^2 \phi_0 = 0$ or the grid is uniform ($|\Delta x_2| = |\Delta x_4|$ and $|\Delta y_1| = |\Delta y_3|$).

4.3.1. LSI-C

The order of accuracy for LSI-C is derived next. Note, the order of accuracy for LSI-F is the same as for LSI-C and is not shown here for the sake of brevity. Let us consider the stencil depicted in Fig. 6(b). The weights for the least squares gradient are

$$w_1^2 = \frac{1}{\Delta y_1^2}, \quad w_2^2 = \frac{1}{\Delta x_2^2}, \quad w_3^2 = \frac{1}{\Delta y_3^2}, \quad w_4^2 = \frac{1}{\Delta x_4^2 + \Delta y_4^2}, \quad (21)$$

Equation (10) reduces to

$$\begin{pmatrix} \partial_x^{LSI-C} \phi_0 \\ \partial_y^{LSI-C} \phi_0 \end{pmatrix} = (\Delta x_4^2 + \Delta y_4^2) \begin{pmatrix} 2\Delta x_4^2 + \Delta y_4^2 & \Delta x_4 \Delta y_4 \\ \Delta x_4 \Delta y_4 & 2\Delta x_4^2 + 3\Delta y_4^2 \end{pmatrix}^{-1} \begin{pmatrix} \frac{\phi_2 - \phi_0}{\Delta x_2} + \frac{(\phi_4 - \phi_0)\Delta x_4}{\Delta x_4^2 + \Delta y_4^2} \\ \frac{\phi_1 - \phi_0}{\Delta y_1} + \frac{\phi_3 - \phi_0}{\Delta y_3} + \frac{(\phi_4 - \phi_0)\Delta y_4}{\Delta x_4^2 + \Delta y_4^2} \end{pmatrix}. \quad (22)$$

The least squares approximation of the gradient can be thus written as

$$\begin{pmatrix} \partial_x^{LSI-C} \phi_0 \\ \partial_y^{LSI-C} \phi_0 \end{pmatrix} = \frac{1}{4\Delta x_4^2 + 3\Delta y_4^2} \begin{pmatrix} (2\Delta x_4^2 + 3\Delta y_4^2) \frac{\phi_2 - \phi_0}{\Delta x_2} - \Delta x_4 \Delta y_4 \left(\frac{\phi_1 - \phi_0}{\Delta y_1} + \frac{\phi_3 - \phi_0}{\Delta y_3} \right) + 2\Delta x_4 (\phi_4 - \phi_0) \\ (2\Delta x_4^2 + \Delta y_4^2) \left(\frac{\phi_1 - \phi_0}{\Delta y_1} + \frac{\phi_3 - \phi_0}{\Delta y_3} \right) - \Delta x_4 \Delta y_4 \frac{\phi_2 - \phi_0}{\Delta x_2} + \Delta y_4 (\phi_4 - \phi_0) \end{pmatrix}. \quad (23)$$

The order of accuracy of Eq. (23) is calculated by replacing ϕ_1 , ϕ_2 , ϕ_3 , and ϕ_4 with their Taylor expansions

$$\begin{pmatrix} \partial_x^{LSI-C} \phi_0 \\ \partial_y^{LSI-C} \phi_0 \end{pmatrix} = \begin{pmatrix} \partial_x \phi_0 + \frac{\partial_{xx}^2 \phi_0}{2} \frac{\Delta x_2 (2\Delta x_4^2 + 3\Delta y_4^2) + 2\Delta x_4^3}{4\Delta x_4^2 + 3\Delta y_4^2} + \frac{\partial_{yy}^2 \phi_0}{2} \frac{2\Delta x_4 \Delta y_4^2 - \Delta x_4 \Delta y_4 (\Delta y_1 + \Delta y_3)}{4\Delta x_4^2 + 3\Delta y_4^2} \\ \partial_y \phi_0 + \frac{\partial_{xx}^2 \phi_0}{2} \frac{\Delta y_4 \Delta x_4^2 - \Delta x_2 \Delta x_4 \Delta y_4}{4\Delta x_4^2 + 3\Delta y_4^2} + \frac{\partial_{yy}^2 \phi_0}{2} \frac{(2\Delta x_4^2 + \Delta y_4^2)(\Delta y_1 + \Delta y_3) + \Delta y_4^3}{4\Delta x_4^2 + 3\Delta y_4^2} \\ + \partial_{xy}^2 \phi_0 \frac{\Delta x_4^2 \Delta y_4}{4\Delta x_4^2 + 3\Delta y_4^2} + O(\Delta x^2) + O\left(\frac{\Delta y^3 \Delta x}{\Delta x^2 + \Delta y^2}\right) + O\left(\frac{\Delta x^4}{\Delta x^2 + \Delta y^2}\right) \\ + \partial_{xy}^2 \phi_0 \frac{\Delta y_4^2 \Delta x_4}{4\Delta x_4^2 + 3\Delta y_4^2} + O(\Delta y^2) + O\left(\frac{\Delta x^3 \Delta y}{\Delta x^2 + \Delta y^2}\right) + O\left(\frac{\Delta y^4}{\Delta x^2 + \Delta y^2}\right) \end{pmatrix}. \quad (24)$$

Equation (24) shows that LSI-C is in general first order accurate even if the two non-conformal mesh blocks are regular structured grids of squared elements.

243 4.3.2. LSI-Fa and GGI

244 In case the face which is located on the interface is treated as an external boundary face, see
 245 Fig. 6(c), LS-Fa and GGI reduce to the same expression

$$\begin{pmatrix} \partial_x^{LSI-Fa} \phi_0 \\ \partial_y^{LSI-Fa} \phi_0 \end{pmatrix} \equiv \begin{pmatrix} \partial_x^{GGI} \phi_0 \\ \partial_y^{GGI} \phi_0 \end{pmatrix} = \begin{pmatrix} \frac{\phi_{fa}-\phi_0}{\Delta x_{fa}} + \frac{\phi_2-\phi_0}{2\Delta x_2} \\ \frac{\phi_3-\phi_0}{2\Delta y_3} + \frac{\phi_1-\phi_0}{2\Delta y_1} \end{pmatrix}. \quad (25)$$

246 The approximation of the gradient along the direction parallel to the interface is not affected by
 247 the interface treatment, i.e. $\partial_y^{LSI-Fa} \phi_0 \equiv \partial_y^{GGI} \phi_0 \equiv \partial_y^{GG} \phi_0 \equiv \partial_y^{LS} \phi_0$. Replacing ϕ_{fa} with Eq. (15)

$$\partial_x^{GGI} \phi_0 = \frac{1}{2\Delta x_{fa}} \left[a \frac{\phi_4 d_{0,1} + \phi_0 d_{4,1}}{d_{0,1} + d_{4,1}} + (1-a) \frac{\phi_5 d_{0,2} + \phi_0 d_{5,2}}{d_{0,2} + d_{5,2}} - \phi_0 \right] + \frac{\phi_2 - \phi_0}{2\Delta x_2}. \quad (26)$$

Expanding ϕ_2 , ϕ_4 , and ϕ_5 in Taylor series we obtain

$$\begin{aligned} \partial_x^{GGI} \phi_0 &= \partial_x \phi_0 - \frac{\partial_x \phi_0}{2} \frac{1}{\Delta x_{fa}} \left[\Delta x_{fa} - \frac{a d_{0,1} \Delta x_4}{d_{0,1} + d_{4,1}} + \frac{(1-a) d_{0,2} \Delta x_5}{d_{0,2} + d_{5,2}} \right] \\ &+ \frac{\partial_y \phi_0}{2} \frac{1}{\Delta x_{fa}} \left[\frac{a d_{0,1} \Delta y_4}{d_{0,1} + d_{4,1}} + \frac{(1-a) d_{0,2} \Delta y_5}{d_{0,2} + d_{5,2}} \right] + \frac{\partial_{xx}^2 \phi_0}{4} \frac{1}{\Delta x_{fa}} \left[\frac{a \Delta x_4^2 d_{0,1}}{d_{0,1} + d_{4,1}} + \frac{(1-a) \Delta x_5^2 d_{0,2}}{d_{0,2} + d_{5,2}} + \Delta x_2 \Delta x_{fa} \right] \\ &+ \frac{\partial_{yy}^2 \phi_0}{4} \frac{1}{\Delta x_{fa}} \left[\frac{a \Delta y_4^2 d_{0,1}}{d_{0,1} + d_{4,1}} + \frac{(1-a) \Delta y_5^2 d_{0,2}}{d_{0,2} + d_{5,2}} \right] + \frac{\partial_{xy}^2 \phi_0}{2} \frac{1}{\Delta x_{fa}} \left[\frac{a d_{0,1} \Delta x_4 \Delta y_4}{d_{0,1} + d_{4,1}} + \frac{(1-a) d_{0,2} \Delta x_5 \Delta y_5}{d_{0,2} + d_{5,2}} \right] + \dots \end{aligned} \quad (27)$$

248 which can be recasted as

$$\partial_x^{GGI} \phi_0 = \partial_x \phi_0 + \partial_x \phi_0 O(\Delta x^0) + \partial_y \phi_0 O\left(\frac{\Delta y}{\Delta x}\right) + \dots, \quad (28)$$

249 showing the presence of two terms of order-0 which prevent the approximated gradient to converge
 250 as the computational grid is refined.

251 4.4. Numerical convergence

252 The accuracy of the gradient reconstruction is verified numerically in two and three dimensions
 253 for the following test functions

$$f_1^{2D} = xy, \quad (29)$$

$$f_2^{2D} = \sin(2\pi\bar{x}) \sin(2\pi\bar{y}), \quad (30)$$

$$f_3^{3D} = \sin(4\pi x) + \sin(4\pi y) + \sin(4\pi z), \quad (31)$$

$$f_4^{3D} = \sin(2\pi\tilde{x}) \sin(2\pi\tilde{y}) \sin(2\pi\tilde{z}). \quad (32)$$

where the bar ($\bar{\cdot}$) and tilde ($\tilde{\cdot}$) indicate a transformed coordinate system, which is rotated by $\pi/4$ about the z and y axis, respectively. The test domains are $\Omega_{2D} = [0, 1] \times [0, 1] \subset \mathbb{R}^2$ and $\Omega_{3D} = [0, 1] \times [0, 1] \times [0, 1] \subset \mathbb{R}^3$. The interface is placed at $x = 0.5$ in the 2D case and at $z = 0.5$ in the 3D case. The computational domain is made of a coarse structured grid on the left side of the interface and of a finer grid on the right side of the interface. The elements of both grid blocks are regular quadrilaterals in 2D and cubes in 3D. The edge ratio of the fine grid elements to the coarse grid is 0.7, and it is kept constant during the mesh refinements. To assess the calculated order of accuracy of the LS, GG, LSI-C, LSI-F, LSI-Fa and GGI methods, the L_1 norm of the difference between the reconstructed and analytical value of the gradient is used, namely

$$\|\varepsilon_j\|_{L_1} = \left(\sum_{i=1}^N |\partial_j^{NUM} \phi_i - \partial_j \phi_i| / |\partial_j \phi_i| \right) / N, \quad (33)$$

where N indicates the total number of control volumes used to calculate the error, and $j = x, y, z$ is the component of the gradient. No slope limiter is used. Results for the particular case of matching mesh blocks are not included since equations (24) and (27) reduce to Eq. (20) in this specific situation, therefore recovering the same convergence property as for the internal cells.

Figure 7 displays the convergence of the error for the test functions in the 2D domain. The calculated convergence rates for LSI-F, LSI-C, LSI-Fa, and GGI are consistent with the analytical derivation of the order of accuracy of Sec. 4.3. LSI-F and LSI-C both converge first order, and LSI-Fa and GGI show an error of order zero along the direction which crosses the interface, while the convergence along the direction parallel to the interface is not affected and is the same as for the internal cells, see Eq. (25). In Fig. 7(a) (f_1^{2D}), the results for the LS and GG methods in the internal cells are not shown since both methods exactly reconstruct a function that has zero second and higher order derivatives on a regular grid. The same property holds for the y derivative in case of GGI and LSI-Fa.

The calculated order of accuracy in 3D is shown in Fig. 8. The rate of convergence agrees with the derivation of Sec. 4.3 obtained in 2D case. In case of f_3^{3D} , LSI-C shows an order of accuracy close to two, while LSI-F remains first order as predicted by the analytical expression. For f_4^{3D} , the calculated converge of the y derivative of both LSI-F and LSI-C is below first order.

Overall, the results do not lead to conclusive evidence about which of the methods is best.

(a) $f_1^{2D} = x y$

(b) $f_2^{2D} = \sin(2\pi\bar{x}) \sin(2\pi\bar{y})$

Figure 7: Average L_1 norm of the error on the gradient reconstruction in the internal and interface control volumes on a two dimensional domain. Symbols indicate the components of the derivatives, namely $\square$ is $\partial_x\phi$ and $\triangle$ is $\partial_y\phi$.

LSI-C appears to offer a better reconstruction of the gradient as compared to LSI-F, and shows second order convergence in case of f_3^{3D} while LSI-F remains first order for all the test functions considered. GGI and LS-Fa are a trade-off between accuracy and simplicity of implementation. On the one hand a constant error in the direction perpendicular to the interface is introduced, which is quantified to about 10% at most for all the numerical tests performed. On the other hand, second order accuracy is maintained along the directions parallel to the interface.

(a) $f_3^{3D} = \sin(4\pi x) + \sin(4\pi y) + \sin(4\pi z)$

(b) $f_4^{3D} = \sin(2\pi\tilde{x}) \sin(2\pi y) \sin(2\pi\tilde{z})$

Figure 8: Average L_1 norm of the error on the gradient reconstruction in the internal and interface control volumes on a three dimensional domain. Symbols indicate the components of the derivatives, namely $\square$ is $\partial_x\phi$, $\triangle$ is $\partial_y\phi$, and $\triangleright$ is $\partial_z\phi$.

5. Parallelization

The parallel distribution of the supermesh generation procedure and of the interface flux calculation are discussed here. The former is independent from the topology of the mesh after the partitioning step. A possible strategy to compute the supermesh in parallel is detailed in Sec. 5.1. The interface flux calculation requires an adaptation in case the interface is splitted among several cores. The choice of whether to distribute the interface control volumes among different cores or not depends on several factors that are summarized in Tab. 1. Load balancing benefits from a sub-division of the interface. However, having the interface distributed on several cores, requires additional data exchange between them in order to update the interface supermesh properties. Moreover, memory usage is penalized by the need to store these additional variables. Given the usually small ratio of interface-to-internal faces, the requirements in terms of additional memory usage and data exchange between the cores are not large enough to justify the loss in load balance. Therefore, only the case where the interface is split among multiple cores is treated here.

5.1. Distributed supermesh construction

The schematic graphical representation of an exemplary parallel supermesh construction is given in Fig. 9. It shows a 2D grid partitioning on three cores that also splits the non-conformal interface. First, the interface control volumes are pre-processed in order to define the complete parent meshes $\mathcal{M}_A$ and $\mathcal{M}_B$, see Fig. 9(b), and this information is shared among all the cores. Subsequently, each core checks only a limited number of $\mathcal{E}_A$ for intersections with $\mathcal{E}_B$ in order to define a temporary local supermesh $\overline{\mathcal{M}}_{S,i}$, see Fig. 9(c), with $i = 1, 2, 3$ in this case. All the

Table 1: Advantages and disadvantages of partitioning the non-conformal interface.

	Single core	Multiple cores
Load balance	–	+
Memory	+	–
Data exchange	+	–
Supermesh generation	+	+

307 $\overline{\mathcal{M}}_{S,i}$ are then grouped together to define the complete supermesh $\mathcal{M}_S$. The last step consists in
 308 distributing to each core only the $\mathcal{E}_S$ mapped on the local $\mathcal{M}_A$ and $\mathcal{M}_B$, see Fig. 9(d) and 9(e). As
 309 a consequence, every core has a new local supermesh $\mathcal{M}_{S,i} \subseteq \mathcal{M}_S$. If no parent mesh elements are
 310 present locally, $\mathcal{M}_{S,i} = \emptyset$, as it is the case for CORE 3 in Fig. 9(a). As expected, the scaling of the
 311 proposed parallelization is linear, see Fig. 10.

(a) Grid partition.

(b) Parent meshes identification.

(c) Parallel supermesh generation.

(d) Local supermesh: CORE

1.

(e) Local supermesh: CORE

2.

Figure 9: Schematic representation of the procedural steps involved in the generation of a supermesh. Filled symbols indicate the cell centers of the supermesh elements.

Figure 10: Calculated scaling of the parallel supermesh construction (dashed line). Results are obtained for two fixed $\mathcal{M}_{A,B}$ counting 5×10^4 elements each.

5.2. Flux assembly across the interface

If information about the control volumes, identified by $\mathcal{T}_{A,i}$ and $\mathcal{T}_{B,i}$, is stored locally on the core, no additional information is needed in order to calculate the supermesh face flux and sum it to the corresponding control volume flux balances. Every variable can be retrieved by using the control volume index only. Conversely, if either $\mathcal{T}_{A,i}$ or $\mathcal{T}_{B,i}$ is not stored locally for that core, communication is required to retrieve the necessary flow variables. Thus, we propose to store on each supermesh face the flow properties and to update them at each time step. This comes at a computational cost that depends on the number of the needed flow variables, on the number of supermesh elements, and on the number of cores used for the partition. Memory usage also increases with the number of variables and supermesh elements. However, given the relatively small interface to internal mesh face ratio, the penalty in terms of memory and computational efficiency is not expected to significantly influence the code performance.

6. Numerical verification for the Navier–Stokes equations

The Navier–Stokes equations are used for the verification of the proposed mesh blocks interface treatment. Five exemplary test cases provide the numerical verification. Stationary non-conformal grid blocks are considered in Sec. 6.2, 6.3 and 6.4, while mesh blocks in relative motion are considered in Sec. 6.5 and 6.6. A wide range of flow regimes is investigated, i.e., subsonic, transonic and supersonic flows, under steady state and unsteady conditions, including flow discontinuities (shock waves) crossing the mesh interface.

For the stationary test cases, the validation is performed by comparing the flow field calculated on a two-block mesh to the one obtained on a single-block mesh, i.e., no interface, with comparable elements size. The numerical flux conservation is verified for the continuity and energy equations as the difference between the integrated values at the interface and the ones calculated at the outlet and inlet boundaries. Relative differences were found to be of the order of machine precision and are not included here. Convergence histories are also shown and the numerical stability is demonstrated empirically for explicit and implicit time integration schemes.

6.1. Navier–Stokes solver

The conservative variables vector for the Navier–Stokes equations is

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho \mathbf{v} \\ \rho e^t \end{pmatrix}, \quad (34)$$

whereby ρ is the fluid density, $\mathbf{v}$ is the velocity vector, and $e^t = e + \frac{1}{2}|\mathbf{v}|^2$ is the specific total energy.

The fluxes are conveniently splitted into a convective and diffusive part, namely $\mathbf{F} = \mathbf{F}_c + \mathbf{F}_d$ and

$$\mathbf{F}_c = \begin{pmatrix} \mathbf{n} \cdot \rho \mathbf{v} \\ \rho \mathbf{v} (\mathbf{n} \cdot \mathbf{v}) + P \mathbf{n} \\ (\rho e^t + P) (\mathbf{v} \cdot \mathbf{n}) \end{pmatrix} \quad \text{and} \quad \mathbf{F}_d = \begin{pmatrix} 0 \\ \mathbf{n} \cdot \Pi \\ \mathbf{v} \cdot (\mathbf{n} \cdot \Pi) + \mathbf{n} \cdot (\lambda \nabla T) \end{pmatrix}, \quad (35)$$

in which P denotes the pressure, T the temperature and λ the thermal conductivity. The stress tensor is expressed as $\Pi = \mu (\nabla \otimes \mathbf{v} + (\nabla \otimes \mathbf{v})^T - \frac{2}{3} (\nabla \cdot \mathbf{v}) \mathbb{I})$, with μ being the dynamic viscosity

of the fluid, T indicating the transpose operator, $\mathbb{I}$ the identity matrix, and $\otimes$ the outer (or tensor) product.

The convective fluxes are evaluated numerically at each face of the domain using an approximate Riemann solver, e.g., AUSM⁺ [27] or HLLC [28]. The evaluation of the diffusive fluxes needs an accurate evaluation of the temperature and velocity gradients on the faces, which are calculated as

$$\nabla \phi_{fa} \cdot \mathbf{n}_{fa} = \frac{\phi_{nb} - \phi_{cv}}{|\mathbf{x}_{nb} - \mathbf{x}_{cv}|} \alpha_{fa} + \frac{1}{2} (\nabla \phi_{nb} + \nabla \phi_{cv}) \cdot (\mathbf{n}_{fa} - \alpha_{fa} \mathbf{s}_{fa}), \quad (36)$$

with $\alpha_{fa} = \mathbf{s}_{fa} \cdot \mathbf{n}_{fa}$, and $\mathbf{s}$ the normalized vector connecting the nb and cv cell centroids. Second order spatial accuracy is achieved using the GG method for the internal control volumes and GGI at the interface. The gradient slope limiter proposed by Venkatakrishnan [29] is adopted. More details on the flow solver used in this study are given by Pecnik *et al.* [30].

Time integration can be performed using explicit three-steps Runge–Kutta (RK3) scheme [31], which is second order accurate, or implicit schemes, namely backward Euler and backward-differentiation-formula (BDF2), in order to avoid restrictions related to the choice of the time step. In case implicit schemes are used, the numerical scheme requires the definition of the flux Jacobians. A thorough derivation of the Jacobians employed in the CFD code is given by Rinaldi *et al.* [32].

6.2. Cylinder in supersonic cross flow

An incoming inviscid supersonic flow at Mach number $M = 4$ is considered. First order accuracy was used for the convective flux discretization and the Euler equations were integrated in time with the explicit RK3 method and the implicit backward Euler scheme. For the latter, the Courant number was gradually increased after 100 time steps from $CFL = 1$ to $CFL = 1000$.

Fig. 11 shows the calculated Mach number contour lines on a single-block and two-block domain. The shock waves cross the interface without generating any spurious oscillation and no discrepancies are observed between the two solutions. The only differences in the Mach distribution downstream of the cylinder are due to a coarser spacing of the elements of the right-hand side mesh block for the two-block grid. The convergence history is not affected by the presence of the interface for both, explicit and implicit simulations, see Fig. 12.

Figure 11: Mach number contour lines obtained from the simulation results of a cylinder in a $M = 4$ supersonic cross flow. The solution obtained on a single-block mesh (solid line) is compared to the one obtained on a two-block mesh (dashed lines). Detailed views of the wake and bow shock at the interface are given in the square boxes.

Figure 12: Convergence history of the energy equation related to the simulation whose results are displayed in Fig. 11. The Euler equations were integrated using the implicit backward Euler (top) and explicit three-steps Runge–Kutta (bottom) methods.

371 6.3. *Laminar vortex shedding*

372 A second test case is provided by the simulation of the laminar vortex shedding from a two-
373 dimensional cylinder at Reynolds number $Re = 1300$ (based on the inlet properties and the cylinder
374 diameter). Second order spatial and temporal accuracy is used. The Navier-Stokes equations are
375 integrated with the explicit RK3 scheme using a constant $\Delta t = 2.5 \times 10^{-6}$ s.

376 The calculated flow fields for the single-block and two-block meshes are superimposed in
377 Fig. 13 in terms of density, vorticity and pressure, at the same normalized time. A coarser mesh is
378 used for the right block of the two-block computational domain, which introduces more diffusion
379 in the vortices downstream of the cylinder and is the reason for slight differences in the contour
380 lines. The coarse mesh block also influences the solution on the fine mesh block (left-hand side)
381 since the information can propagate upstream in a subsonic flow.

Figure 13: Results of the simulation of the laminar vortex shedding at $Re = 1300$. The solution obtained by means of a single-block mesh (solid line) is compared to the one obtained with a two-block mesh (dashed lines). **Note that in case of a two-block mesh, the blocks are considered as disconnected by the visualization tool. Thus, the contour lines might display a discontinuity at the interface, which is not physical but merely an artifact of the visualization.**

382 6.4. *Inviscid supersonic flow in a three-dimensional double bump channel*

383 The simulation of an inviscid, $M = 2.5$ supersonic flow in a double bump channel is used to
384 test the interface treatment in case the flow is three-dimensional. The two bumps are located on
385 the bottom and side wall. This configuration guarantees that the flow is fully three-dimensional
386 due to the interaction between shock waves and expansion fans. A structured mesh is used for
387 the single-block domain. The interface for the two-block mesh is located downstream of the
388 bumps. The mesh upstream of the interface is the same as for the single block configuration, while
389 downstream of the interface the mesh consists of wedge elements. In both cases the computational
390 domain counted approximately 650,000 control volumes. Second order spatial accuracy was used,
391 and the Euler equations were integrated using the implicit backward Euler scheme with an initial
392 $CFL = 50$, which was gradually increased after 10 time steps up to $CFL = 1000$.

393 The calculated flow fields are displayed in Fig. 14 on two planes, at half of the channel width
394 and height. The small but appreciable differences seen in the Mach number contour lines down-
395 stream of the interface are due to the different volume distribution of mesh elements between the
396 single-block and two-block cases. The convergence history of the energy equation is reported in
397 Fig. 15 and shows a slower convergence rate for the two-block case after the residual has decreased
398 by six orders of magnitude.

(a) Half channel height plane.

(b) Half channel width plane.

Figure 14: Mach number contour lines for the simulation results of an inviscid $M = 2.5$ supersonic flow in a three-dimensional double bump channel. The solution obtained by means of a single-block mesh (solid line) is compared to the one obtained with a two-block mesh (dashed lines).

Figure 15: Convergence history of the energy equation related to the simulation whose results are displayed in Fig. 14. Equations were integrated using the backward Euler scheme.

6.5. *Cylinders in relative motion in inviscid supersonic cross flow*

This and the following section describe the results of two test cases in which the interface treatment is applied to mesh blocks in relative motion. For these cases the supermesh construction as described in Sec. 2 has to be repeated at every time step after the relative position of the two mesh blocks is updated.

The right patch of the computational domain moves vertically at a constant speed v_R such that $v_R/c_{in} = 0.2$, where c_{in} is the speed of sound at the inlet. The test case is the simulation of a uniform normalized ideal gas flow entering from the inlet boundary at $M = 4$ and crossing two cylinders. Periodic boundary conditions are set on the upper and lower boundaries. Second order spatial accuracy is used and Euler equations are integrated using the implicit two-steps backward-differentiation-formula (BDF2) method with a constant time step defined such that one translating period is discretized in 28,000 steps.

Fig. 16 displays the evolution of the Mach distribution over one period. Downstream of the first cylinder the flow is still supersonic. This results in a second bow shock in front of the moving cylinder, which interacts with the first one. The shock crosses the interface smoothly, as seen by the contour lines and the same conclusion of Sec. 6.2 also apply here. The convergence history shows that a periodic flow state is achieved after two complete cylinder passes, and that the present interface treatment does not have any negative effect on the periodicity, see Fig. 17. The peak of the residual at $t = 0$ corresponds to the downstream cylinder being in the wake of the upstream cylinder. The two peaks at $t = 0.25T$ and $t = 0.75T$, where T indicates the period, correspond to the impingement of the first bow shock tails on the front part of the second bow shock, as can be observed in Fig. 16.

Figure 16: Mach number contour lines for the simulation results of two cylinders in relative motion in a $M = 4$ supersonic cross flow. The external edge of the mesh blocks is shown in order to locate the interface and the periodic boundaries for the moving patch.

Figure 17: Convergence history of the energy equation related to the simulation whose results are displayed in Fig. 16. Equations were integrated using the BDF2 method and a constant time step chosen such that one period corresponds to 28,000 steps.

6.6. Transonic axial turbine stage

The final test case chosen for the verification of the interface treatment is the simulation of the transonic flow through the blades of an axial turbine stage. The considered geometry was designed by Erhard and Gehrler [33] and numerically investigated by Schennach *et al.* [34]. The purpose of this test case is to show the effectiveness of the interface treatment on a problem of engineering interest, such as an unsteady turbomachinery simulation, which features the complexity of combined vortex shedding crossing the interface and shock waves propagating upstream and downstream of the interface. Due to the periodicity of the geometry, a reduced computational domain was used to perform the CFD simulation. It includes two stator vanes and three rotor blades, counting approximately 50,000 cells in total. A two-dimensional turbulent simulation of the mid-span plane was performed using the Spalart and Allmaras (SA) turbulence model [35], whereby a non-uniform distribution of the cross-section area was included. Second order spatial accuracy was used. Total pressure $P_{tot} = 3.429$ bar and total temperature $T_{tot} = 454.4$ K were set at the inlet, while a constant static pressure $P = 1.102$ bar was imposed at the outlet. The rotor blades move at a vertical velocity of $v_R = 268.166$ m/s. The Navier–Stokes equations and the transport equation for the SA turbulence model were integrated with the BDF2 method using 40,000 time steps to discretize one period.

Contours for the magnitude of the vorticity and lines for the magnitude of the pressure gradient are shown in Fig. 18, which displays the evolution of the flow field over a complete period. The fluid enters the domain from the left-hand-side boundary and is accelerated in the stator to a supersonic state, which leads to the formation of a normal shock wave near the trailing edge of the stator vanes. The shock propagates to the rotor and impinges on the leading edge of the rotor blades. A second shock is then reflected back to the stator interacting with the vortex shedding from the vanes trailing edge, and eventually impinges on the vane suction side. Supersonic conditions are met also further downstream in the rotor channel, as demonstrated by the presence of a trailing edge shock wave on the rotor blades.

Figure 18: Results of the simulation of the flow in the transonic turbine stage described in Refs. [33, 34]. Colors represent the vorticity field (blue indicates negative values, red positive values, the range is from -35,000 1/s to 35,000 1/s). Lines indicate the pressure gradient (100 lines are used on the range 5 MPa/m to 1,100 MPa/m). The images cover a complete time period starting from $t = 0$ in the top-left sub-figure, whereby the computational domain is shown (black line).

7. Conclusions

This paper presents a new flux-conserving treatment of the interface between non-conformal mesh blocks. The method is based on the definition of an auxiliary interface mesh, called supermesh, which establishes a connection between the interface control volumes and is defined by the intersection of the elements of the non-conformal interface grid blocks. The method can be applied to the finite-volume discretization of any system of conservation laws, however the present study is applied to the Euler and Navier–Stokes equations only.

Numerical flux conservation is ensured by calculating the fluxes on the supermesh elements and summing their contribution to the respective control volumes of the parent meshes, thus avoiding any flux interpolation. Details of the procedure to generate the supermesh are discussed in 2D and 3D cases. The accuracy of the intersection scheme used to define the supermesh elements is of crucial importance to satisfy the definition of the supermesh and to ensure the flux conservation property of the method. The parallelization of the supermesh generation and interface treatment is also described for simulations on multi-core computing architectures.

Modifications to the least-squares and Green–Gauss gradient reconstruction methods are proposed at the non-conformal mesh interface. The order of accuracy of the new schemes is derived analytically and a numerical verification of the results is presented for 2D and 3D domains. It is found that, in case of least-squares reconstruction, extending the stencil to the control volumes of the mesh block opposite of the interface leads to a first order convergence of the gradient along each spatial direction. Alternatively, the interface faces can be treated as external boundaries and their values can be set as a linear combination of the neighbors. In this case, the accuracy of the gradient reconstruction along the direction which crosses the interface is reduced to 0-order, thus preventing convergence as the mesh is refined.

The supermesh interface treatment is verified on five test cases featuring shock waves crossing the interface and the transport of viscous vortices. Numerical fluxes conservation is achieved to machine precision. The calculated flow fields show no spurious oscillations at the interface and agree with the results obtained on a single-block mesh, i.e., no interface. Numerical stability is demonstrated empirically by solving the equations with three different time integration schemes,

475 namely explicit three-steps Runge–Kutta, implicit backward Euler and BDF2 methods, and by
476 studying subsonic, transonic and supersonic flow regimes. The convergence rates were not found
477 to be affected by the presence of the interface.

478 **Acknowledgments**

479 The development of the Navier–Stokes solver has been supported by the (U.S.) Department of
480 Energy (DOE) [National Nuclear Security Administration] under Award No. NA28614. Antonio
481 Buonocore and Gustavo Otero Rodriguez are acknowledged for their help in the implementation
482 of a preliminary version of the method.

References

- [1] M. J. Berger, On conservation at grid interfaces, *SIAM J. Numer. Anal.* 24 (5) (1987) 967–984.
- [2] M. M. Rai, A conservative treatment of zonal boundaries for Euler equation calculations, *J. Comput. Phys.* 62 (2) (1986) 472–503.
- [3] K. A. Hessenius, M. M. Rai, Applications of a conservative zonal scheme to transient and geometrically complex problems, *Computers and Fluids* 14 (1) (1986) 43–58.
- [4] M. M. Rai, An implicit, conservative, zonal-boundary scheme for Euler equation calculations, *Comput. Fluids* 14 (3) (1986) 295–319.
- [5] M. M. Rai, Navier-Stokes simulations of rotor/stator interaction using patched and overlaid grids, *J. Propul. Power* 3 (5) (1987) 387–396.
- [6] A. Lerat, Z. N. Wu, Stable conservative multidomain treatments for implicit Euler solvers, *J. Comput. Phys.* 123 (1) (1996) 45–64.
- [7] A. Benkenida, J. Bohbot, J. C. Jouhaud, Patched grid and adaptive mesh refinement strategies for the calculation of the transport of vortices, *Int. J. Numer. Meth. Fl.* 40 (7) (2002) 855–873.
- [8] Y. Zhang, H. Chen, S. Fu, Improvement to patched grid technique with high-order conservative remapping method, *J. Aircraft* 48 (3) (2011) 884–893.
- [9] M. Behr, T. Tezduyar, The shear-slip mesh update method, *Comput. Method. Appl. M.* 174 (3-4) (1999) 261–274.
- [10] M. Behr, T. Tezduyar, Shear-slip mesh update in 3D computation of complex flow problems with rotating mechanical components, *Comput. Method. Appl. M.* 190 (24-25) (2001) 3189–3200.
- [11] Y. Wang, N. Qin, G. Carnie, S. Shahpar, Zipper layer method for linking two dissimilar structured meshes, *J. Comput. Phys.* 255 (2013) 130–148.
- [12] X. Jiao, M. T. Heath, Common-refinement-based data transfer between non-matching meshes in multiphysics simulations, *Int. J. Numer. Meth. Eng.* 61 (14) (2004) 2402–2427.
- [13] D. K. Gartling, Multipoint constraint methods for moving body and non-contiguous mesh simulations, *Int. J. Numer. Meth. Fl.* 47 (6-7) (2005) 471–489.
- [14] J. A. Benek, J. L. Steger, F. C. Dougherty, A flexible grid embedding technique with application to the Euler equations, in: *Computational Fluid Dynamics Conference 6th*, no. AIAA paper 83-1944, Danvers, MA, 1983.
- [15] J. A. Benek, P. G. Buning, J. L. Steger, A 3-D CHIMERA grid embedding technique, in: *Computational Fluid Dynamics Conference 7th*, no. AIAA paper 85-1523, Cincinnati, OH, 1985.
- [16] Z. J. Wang, A fully conservative interface algorithm for overlapped grids, *J. Comput. Phys.* 122 (1) (1995) 96–106.
- [17] K.-H. Kao, M.-S. Liou, Advance in overset grid schemes: From Chimera to DRAGON grids, *AIAA J.* 33 (10) (1995) 1809–1815.

- [18] Z. J. Wang, N. Hariharan, R. Chen, Recent development on the conservation property of Chimera, *Int. J. Comput. Fluid D.* 15 (4) (2001) 265–278.
- [19] Y. Liu, Central schemes on overlapping cells, *J. Comput. Phys.* 209 (1) (2005) 82–104.
- [20] A. Kurganov, E. Tadmor, New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations, *J. Comput. Phys.* 160 (1) (2000) 241–282.
- [21] G. Chesshire, Composite overlapping meshes for the solution of partial differential equations, *J. Comput. Phys.* 90 (1) (1990) 1–64.
- [22] J. Flores, N. M. Chaderjian, Zonal Navier-Stokes methodology for flow simulation about a complete aircraft, *Journal of Aircraft* 27 (7) (1990) 583–590.
- [23] H. S. Tang, T. Zhou, On nonconservative algorithms for grid interfaces, *SIAM J. Numer. Anal.* 37 (1) (2000) 173–193.
- [24] P. E. Farrell, M. D. Piggott, C. C. Pain, G. J. Gorman, C. R. Wilson, Conservative interpolation between unstructured meshes via supermesh construction, *Comput. Method. Appl. M.* 198 (33–36) (2009) 2632–2642.
- [25] S. Menon, D. P. Schmidt, Conservative interpolation on unstructured polyhedral meshes: An extension of the supermesh approach to cell-centered finite-volume variables, *Comput. Method. Appl. M.* 200 (41–44) (2011) 2797–2804.
- [26] D. Mavriplis, Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes, in: *AIAA paper*, Vol. 3986, 2003, pp. 1–13.
- [27] M.-S. Liou, A sequel to AUSM: AUSM⁺, *J. Comput. Phys.* 129 (2) (1996) 364–382.
- [28] E. F. Toro, M. Spruce, W. Speares, Restoration of the contact surface in the HLL-Riemann solver, *Shock Waves* 4 (1) (1994) 25–34.
- [29] V. Venkatakrishnan, Convergence to steady state solutions of the Euler equations on unstructured grids with limiters, *J. Comput. Phys.* 118 (1) (1995) 120–130.
- [30] R. Pecnik, V. E. Terrapon, F. Ham, G. Iaccarino, H. Pitsch, Reynolds-averaged Navier-Stokes simulations of the HyShot II scramjet, *AIAA J.* 50 (8) (2012) 1717–1732.
- [31] C. W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, *J. Comput. Phys.* 77 (2) (1988) 439–471.
- [32] E. Rinaldi, R. Pecnik, P. Colonna, Exact Jacobians for implicit Navier–Stokes simulations of equilibrium real gas flows, *J. Comput. Phys.* 270 (2014) 459–477.
- [33] J. Erhard, A. Gehrler, Design and construction of a transonic test-turbine facility, in: *Proceedings of the ASME Turbo Expo*, no. 2000-GT-480, 2000, pp. 1–8.
- [34] O. Schennach, R. Pecnik, B. Paradiso, E. Göttlich, A. Marn, J. Woisetschlager, The effect of vane clocking on the unsteady flow field in a one-and-a-half stage transonic turbine, *J. Turbomach.* 130 (3) (2008) 031022.
- [35] P. R. Spalart, S. R. Allmaras, One-equation turbulence model for aerodynamic flows, *Rech. Aerospatale* (1)