

Interpolation with restrictions between finite element meshes for flow problems in an ALE setting

Arnau Pont, Ramon Codina and Joan Baiges^{*,†}

Centre Internacional de Metodes Numerics en Enginyeria, Universitat Politecnica de Catalunya, C/ Gran Capita s/n, Campus Nord UPC, Barcelona 08034, Spain

SUMMARY

The need for remeshing when computing flow problems in domains suffering large deformations has motivated the implementation of a tool that allows the proper transmission of information between finite element meshes. Because the Lagrangian projection of results from one mesh to another is a dissipative method, a new conservative interpolation method has been developed. A series of constraints, such as the conservation of mass or energy, are applied to the interpolated arrays through Lagrange multipliers in an error minimization problem, so that the resulting array satisfies these physical properties while staying as close as possible to the original interpolated values in the L^2 norm. Unlike other conservative interpolation methods that require a considerable effort in mesh generation and modification, the proposed formulation is mesh independent and is only based on the physical properties of the field being interpolated. Moreover, the performed corrections are neither coupled with the main calculation nor with the interpolation itself, for which reason the computational cost is very low. Copyright © 2016 John Wiley & Sons, Ltd.

Received 23 May 2016; Revised 5 September 2016; Accepted 2 October 2016

KEY WORDS: conservation; finite element method; interpolation; numerical methods; restrictions

1. INTRODUCTION

The interpolation of numerical solutions between computational meshes is a well-known procedure with multiple applications, for example, the transmission of arrays in case of problems with evolving domains, coupling between different physical problems or computational codes, initialization or update of boundary conditions or visualization of results. In some of these cases, a simple Lagrangian interpolation within the elements is acceptable, but in other scenarios, the non-conservative character of the interpolation operation can cause considerable dissipation and other kinds of numerical error. This has led to the development of multiple methods for the transmission of information between meshes in finite element computations.

The accurate projection of transmission conditions between two subdomains was the main motivation for the development of an interpolation method based on the conservation of relevant magnitudes [1], which has already been applied to the fixed mesh ALE method by [2]. In this paper, the projection of vector fields corresponding to the transmission conditions between subdomains in a domain decomposition problem is constrained by the conservation of mass in a weak sense and by the conservation of the L^2 norm of the solution along the boundary. The application of these restrictions is performed by solving an optimization problem with constraints controlled by Lagrange multipliers, which enforce that the distance between the interpolated and the corrected solutions is

^{*}Correspondence to: Joan Baiges, Departament d'Enginyeria Civil i Ambiental Campus Nord – Universitat Politecnica de Catalunya Edifici C1Jordi Girona, Barcelona 1-308034, Spain.

[†]E-mail: jbaiges@cimne.upc.edu

minimum in the L^2 norm and that the restrictions are fulfilled. This concept is the background of the formulation presented in this work.

The pioneer works on conservative interpolation between computational meshes were an answer to the needs regarding the domain deformation in Arbitrary Lagrangian Eulerian (ALE) calculations, where the computational nodes of the mesh may either remain fixed as in the Eulerian configuration or can be prescribed an arbitrary velocity [3]. The techniques for achieving the conservation of certain magnitudes during the interpolation process consisted in either building a new mesh by locally modifying the original one, or interpolating the results to a new random grid after computing the intersections between both meshes. The first case corresponds to the local rezoning, which normally takes place at every time step but at a low computational cost. An example of such an algorithm is presented by [4], a second-order accurate scheme that divides the new cells taking into account the intersections with the old ones and guarantees the conservation of the mass fluxes along the edges of the finite volumes in two dimensions. This method was extended to three-dimensional (3D) domains by [5]. On the other hand, global remapping does not assume any relation between the original and the target mesh, for which reason these algorithms deal with the whole domain and increase the computational cost, although they might be used less often during the simulation. In this sense, [6] and [7] present a remapping method that computes the intersection volume between meshes with a surface integral and builds the interpolated field performing an area-weight averaging. On the other hand, some disciplines such as weather forecast use conservative numerical methods in a semi-Lagrangian frame of reference that require the tracking of the trajectory and the interpolation of the convection velocity to an Eulerian frame of reference. An area-weighted interpolation is also used to prevent loss of accuracy by [8]. The concept of interpolation with constraints is presented by [9] for finite volumes and finite differences, where the integration weights and the interpolation coefficients are forced to fulfill certain conditions. [10] propose a P1 exact, mass conservative interpolation scheme, which also fulfills the maximum principle by reconstructing the mass field and its gradient with the elemental intersections between both meshes. Another popular technique developed recently is the common-refinement or the supermesh [11], [12], and [13]. It consists in the construction of an intermediate mesh as a union of the original and the target meshes in order to allow an efficient Galerkin projection of the results, which minimizes the L^2 norm of the interpolation error. An important method regarding the formulation that will be presented in this work is that of [14]. It is not based on grid operations or algorithms but on the physics of the problem, in this case, the shallow water equations. The unknown velocity on the target mesh is expressed as an L^2 projection of the old one plus a correction factor defined in terms of the gradient of a scalar function (pseudo-pressure), which leads to an elliptic problem of the projected velocity correction. Additionally, this method yields local mass conservation without changing the vorticity of the velocity field.

As stated before, one of the main fields of application of conservative interpolation methods is the transmission of loads between interfaces belonging to non-coincident meshes. This scenario happens in coupled problems such as fluid–structure interaction (FSI) or in domain decomposition methods. In the first case, [15] propose the conservation of the load along the interface with a node-projection scheme. On the other hand, [16] use a quadrature projection of the loads, which involves the transmission of the integration points to the target mesh, and finally, the local version of the common refinement [11], being this last one the only whose error is independent of the degree of mismatching between meshes. A comparison between them can be found in [17], where the common-refinement method is recommended because of its general good performance with a reasonable computational cost. In the case of non-conforming domain decomposition methods, one of the most extended methods of conservative interpolation is the mortar element [18].

An alternative approach to the remeshing problem is that presented by [19], which presents a method for a homogeneous distortion of moving computational meshes by preventing those elements located near the boundaries to collapse. This is achieved by solving a virtual structural problem on the mesh and giving the problematic elements a higher Young modulus in order to reach a uniform mesh displacement field. For problems involving moderate deformations leading to high local mesh distortion, this method avoids remeshing, whereas for large deformations, it

would reduce the remeshing frequency. For this reason, it is foreseen as a complementary tool to the present work.

The presented methodology overcomes the typical difficulties related to the described conservative schemes. First of all, the method is valid for all discretizations and is independent of the degree of coincidence between them. Moreover, neither algorithms related to the mesh, such as rezoning, nor computation of local integrals are performed, for which reason the computational cost will be low and only dependent on the size of the meshes. Another important property is that, unlike many of the aforementioned methods, the conservation is achieved interpolating directly the solution values instead of fluxes at overlapping boundaries. Although the conservation of magnitudes will be global and not satisfied locally, the procedure presented in this work allows to enforce conservation conditions of a general form that may not be restricted to mass and momentum conservation.

2. FORMULATION

2.1. The concept

Let us consider a finite element partition $\mathcal{T}_{h,1}$ of a domain $\Omega \subset \mathbb{R}^d$, $d \geq 1$, and a discrete solution $u_{h,1}$ of an arbitrary problem defined in the subspace of functions $\mathcal{V}_{h,1} \subset \mathcal{V}$, where \mathcal{V} is the space of functions where the continuous solution lives. For the sake of conciseness, we will consider scalar functions in this section. Given another finite element partition $\mathcal{T}_{h,2}$ of the same domain, the objective is to construct $u_{h,2}$ from $u_{h,1}$ with the same approximation properties to the continuous solution as $u_{h,1}$. We will call $u_{h,2}$ the interpolation of $u_{h,1}$ on $\mathcal{T}_{h,2}$ (Figure 1) and $\mathcal{V}_{h,2}$ the corresponding finite element space.

In order to simplify the notation for the rest of the paper, the finite element functions will be expressed in terms of the shape functions of each finite element partition, which will be denoted as $\{N_n^a\}$, where $n = 1, 2$ refers to the mesh and the superscript to the nodes, so that a will run from 1 to the number of nodes of each mesh. Denoting by U_n^a the corresponding nodal values and assuming a standard Lagrangian interpolation, we may write

$$u_{h,n} = \sum_a N_n^a U_n^a, \quad n = 1, 2,$$

where summation is understood to run over the set of nodal values.

Let us describe two simple possibilities to compute $u_{h,2}$.

Option 1: L^2 projection

The interpolation $u_{h,2}$ might consist in an L^2 projection of $u_{h,1}$ onto $\mathcal{V}_{h,2}$. This is equivalent to minimize the functional $J(v_{h,2}) = \frac{1}{2} \|v_{h,2} - u_{h,1}\|_{L^2(\Omega)}^2$, and leads to the problem: find $u_{h,2} \in \mathcal{V}_{h,2}$ such that

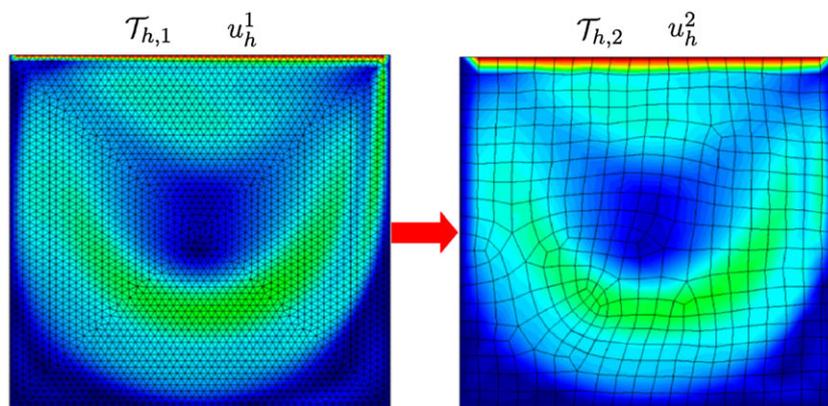


Figure 1. Interpolation of a finite element solution between two computational meshes.

$$\int_{\Omega} u_{h,2} v_{h,2} = \int_{\Omega} u_{h,1} v_{h,2},$$

for all test functions $v_{h,2} \in \mathcal{V}_{h,2}$. Using the shape functions and nodal values, the problem to be solved becomes

$$\sum_b \int_{\Omega} N_2^a N_2^b U_2^b = \sum_c \int_{\Omega} N_2^a N_1^c U_1^c, \tag{1}$$

for all nodes a of $\mathcal{T}_{h,2}$. Let us call \mathbf{U}_n the array of nodal values of $u_{h,n}$, M_2 the standard mass matrix of $\mathcal{T}_{h,2}$ and \mathbf{P}_{21} the array whose components appear in the right-hand-side of Equation (1). The solution to this equation can thus be written as

$$\mathbf{U}_2 = M_2^{-1} \mathbf{P}_{21}.$$

Apart from the cost of solving this algebraic system, the algorithmic difficulty of this method is the evaluation of the components of \mathbf{P}_{21} . Suppose that numerical integration associated to $\mathcal{T}_{h,2}$ is used, with integration points $\{\mathbf{x}_2^g\}$ and associated weights $\{w_2^g\}$. Then

$$\sum_c \int_{\Omega} N_2^a N_1^c U_1^c \approx \sum_c \sum_g w_2^g N_2^a(\mathbf{x}_2^g) N_1^c(\mathbf{x}_2^g) U_1^c$$

The difficulty is the evaluation of $N_1^c(\mathbf{x}_2^g)$. This requires finding the element domain $K_1 \in \mathcal{T}_{h,1}$ where \mathbf{x}_2^g belongs. If node c belongs to this element K_1 , then $N_1^c(\mathbf{x}_2^g)$ can be computed from interpolation. Otherwise, it is 0.

Option 2: Standard Lagrangian interpolation

The method described earlier requires the solution of a system of equations, which may be inconvenient from a computational point of view, even if it is easy to invert the mass matrix. A simpler alternative is to obtain the nodal values of $u_{h,2}$ simply by interpolation, that is,

$$U_2^a = \sum_b N_1^b(\mathbf{x}_2^a) U_1^b,$$

where \mathbf{x}_2^a are the coordinates of node a . This is a node of $\mathcal{T}_{h,2}$, whereas $\{N_1^b\}$ are the shape functions of $\mathcal{T}_{h,1}$. Therefore, one needs to find the element $K_1 \in \mathcal{T}_{h,1}$ where \mathbf{x}_2^a belongs, and then proceed to interpolate as in the previous approach.

Interpolation with restrictions

The approach described earlier has the drawback of being non-conservative and very diffusive, which is precisely what the application of certain restrictions aims to compensate. However, it is more flexible and easier to parallelize, whereas the L^2 projection, in spite of yielding a smaller interpolation error, is not completely conservative either and, in some cases, the mass and other quantities should be corrected as well. Therefore, the idea we propose consists in obtaining $\tilde{u}_{h,2} \in \mathcal{V}_{h,2}$ that satisfies two important features:

- It must remain the nearest solution to $u_{h,2}$ in the L^2 norm, $u_{h,2}$ being obtained from a standard Lagrangian interpolation, for example.
- It must fulfill some physical properties of $u_{h,1}$, which could involve for example the conservation of a set of magnitudes.

In order to describe the second point, let us define the restriction operators (forms). When applied to the finite element functions, these operators give a scalar result corresponding to a relevant magnitude of the physical problem being calculated. We write them as

$$R_{n,i} : \mathcal{V}_{h,n} \longrightarrow \mathbb{R}, \tag{2}$$

where the subindex $n = 1, 2$ refers either to the donor 1 or to the target mesh 2, and $i = 1, \dots, m$ is the restriction counter. Operators $R_{n,i}$ can be written in terms of their nodal values, so that

$$R_{n,i}(u_{h,n}) = \sum_a R_{n,i}^a U_n^a,$$

where $R_{n,i}^a = R_{n,i}(N_n^a)$ for linear restrictions and $R_{n,i}^a$ may depend on \mathbf{U}_n in the case of nonlinear ones.

Because it has been stated that the restriction consists in conserving the value of these selected magnitudes, the following equalities must hold:

$$\sum_a R_{2,i}^a \tilde{U}_2^a = \sum_b R_{1,i}^b U_1^b, \quad i = 1, \dots, m, \tag{3}$$

where \tilde{U}_2 denotes the nodal values associated to $\tilde{u}_{h,2}$. The computational domain Ω remains the same for all stages of the process, because the interpolation process is completely static although the simulation considers a dynamic domain. In some cases, only a part of the domain needs to be interpolated, for example, in the case of FSI problems, where the values at the interface must be exchanged between both subdomains. This scenario will not be considered in the present work.

Although this method is only presented for some particular flow problems, it can be extended to other physical problems. For example, the formulation could be extended to flows with free surfaces in a straightforward way by restricting the constraints to the flow subdomain, which would require subelemental integration [2]. Problems in which the restrictions are inequalities rather than equalities could also be treated by embedding the scheme in a iterative loop that requires to solve a sequence of problems with equality constraints, as it is performed in optimization; these inequality restrictions are found for example in plasticity problems, where the stresses are required to be admissible [20]. This last example requires a further extension consisting in posing the restriction in terms of variables defined at the numerical integration points, such as internal variables like plastic multipliers or damage; it is only required to be able to express these variables in terms of the nodal unknowns of the problem and proceed as we propose here.

At this point, it is important to distinguish the current method from the mortar method [18]. Unlike the present method, the mortar method consists in a domain decomposition technique that performs a L^2 projection along an interface between non-matching grids satisfying adequate matching conditions. The restriction of these fluxes might be performed with Lagrange multipliers. In our case, the interpolation is performed point by point and the restrictions affect the whole domain.

2.2. Application to linear restrictions

Let us develop the concept introduced earlier in the case in which the restrictions are linear functionals. For example, bearing in mind that one of the main issues in the numerical computations of incompressible flows is the conservation of mass, it is reasonable to formulate a problem with this linear restriction.

The restrictions can be enforced through Lagrange multipliers $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_m) \in \mathbb{R}^m$. Given $u_{h,2} \in \mathcal{V}_{h,2}$, the interpolation of $u_{h,1} \in \mathcal{V}_{h,1}$ on $\mathcal{T}_{h,2}$, let us define the following functional $L : \mathcal{V}_{h,2} \times \mathbb{R}^m \rightarrow \mathbb{R}$:

$$L(v_{h,2}, \boldsymbol{\mu}) = \frac{1}{2} \left\| \sum_a N_2^a (V_2^a - U_2^a) \right\|_{L^2(\Omega)}^2 - \sum_{i=1}^m \mu_i \left(\sum_a R_{2,i}^a V_2^a - \sum_b R_{1,i}^b U_1^b \right), \tag{4}$$

where V_2^a are the nodal values of $v_{h,2}$, $\boldsymbol{\mu} \in \mathbb{R}^m$ is an admissible set of Lagrange multipliers, and the subscript in the norm indicates that it is the $L^2(\Omega)$ one. Then the solution that minimizes the distance to the interpolated values and imposes the restrictions will be

$$[\tilde{u}_{h,2}, \boldsymbol{\lambda}] = \arg \inf_{v_{h,2} \in \mathcal{V}_{h,2}} \sup_{\boldsymbol{\mu} \in \mathbb{R}^m} [L(v_{h,2}, \boldsymbol{\mu})]. \tag{5}$$

This is a saddle point problem. A necessary and sufficient condition for it to be well posed is that the finite element space $\mathcal{V}_{h,2}$ and \mathbb{R}^m satisfy the appropriate inf-sup condition. This in particular restricts the number of Lagrange multipliers. However, because m is usually very small, we have not encountered any stability problem.

The equations to be solved are obtained by differentiation of the functional L with respect to the unknowns:

$$\frac{\partial L}{\partial V_2^b} = 0 \implies \sum_a \int_{\Omega} N_2^b N_2^a \tilde{U}_2^a - \sum_{i=1}^m \lambda_i R_{2,i}^b = \sum_a \int_{\Omega} N_2^b N_2^a U_2^a, \quad \text{for all } b, \tag{6}$$

$$\frac{\partial L}{\partial \mu_i} = 0 \implies \sum_a R_{2,i}^a \tilde{U}_2^a = \sum_b R_{1,i}^b U_1^b, \quad \text{for all } i. \tag{7}$$

These two equations can be arranged into the following algebraic system:

$$\begin{bmatrix} M_2 & -R_2^T \\ R_2 & 0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{U}}_2 \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} M_2 \mathbf{U}_2 \\ R_1 \mathbf{U}_1 \end{bmatrix},$$

where R_n are the matrices of the linear restriction operators, $n = 1, 2$.

In order to avoid solving the full linear system, which can be expensive and requires to add the Lagrange multiplier unknowns to the solver, we can solve first the Schur complement problem for the Lagrange multipliers and later compute the nodal values $\tilde{\mathbf{U}}_2$:

$$R_2 M_2^{-1} R_2^T \boldsymbol{\lambda} = R_1 \mathbf{U}_1 - R_2 \mathbf{U}_2, \tag{8}$$

$$\tilde{\mathbf{U}}_2 = \mathbf{U}_2 + M_2^{-1} R_2^T \boldsymbol{\lambda}. \tag{9}$$

Note that system (8) is of small size $m \times m$. The only difficulty for solving this problem is the calculation of M_2^{-1} . This is trivial if the mass matrix M_2 is approximated by a diagonal matrix. If not, the problem can be solved using iterative methods, each iteration of which requiring the solution of a system of equations with matrix M_2 ; it is known that these systems can be solved with a small computational cost.

2.3. Nonlinear restrictions

The same idea can be applied in the case in which the restrictions are nonlinear, that is, mappings $R_{n,i}$ in (2) are nonlinear. Let us assume that we write them as

$$R_{n,i}(v_{h,n}) = \sum_a R_{n,i}^a(\mathbf{V}_n) V_n^a,$$

with the components $R_{n,i}^a(\mathbf{V}_n)$ depending of the nodal values \mathbf{V}_n . We consider now the functional

$$\begin{aligned} L(v_{h,2}, \boldsymbol{\mu}) &= \frac{1}{2} \left\| \sum_a N_2^a (V_2^a - U_2^a) \right\|_{L^2(\Omega)}^2 \\ &\quad - \sum_{i=1}^m \mu_i \left(\sum_a R_{2,i}^a(\mathbf{V}_2) V_2^a - \sum_b R_{1,i}^b(\mathbf{U}_1) U_1^b \right). \end{aligned} \tag{10}$$

The optimization of this functional leads to the equations:

$$\begin{aligned} \frac{\partial L}{\partial V_2^b} = 0 &\implies \sum_a \int_{\Omega} N_2^b N_2^a \tilde{U}_2^a - \sum_{i=1}^m \lambda_i \left(R_{2,i}^b(\tilde{\mathbf{U}}_2) + \sum_a \frac{\partial R_{2,i}^a(\tilde{\mathbf{U}}_2)}{\partial \tilde{U}_2^b} \tilde{U}_2^a \right) \\ &= \sum_a \int_{\Omega} N_2^b N_2^a U_2^a, \quad \text{for all } b, \end{aligned} \tag{11}$$

$$\frac{\partial L}{\partial \mu_i} = 0 \implies \sum_a R_{2,i}^a(\tilde{\mathbf{U}}_2) \tilde{U}_2^a = \sum_b R_{1,i}^b(\mathbf{U}_1) U_1^b, \quad \text{for all } i. \tag{12}$$

This is a nonlinear system that needs to be linearized. Let us use a superscript between parenthesis to denote the iteration counter. Using a fixed point method for (11) and a fixed point or Newton–Raphson linearization for (12), we have that

$$\sum_a \int_{\Omega} N_2^b N_2^a \tilde{U}_2^{a,(k)} - \sum_{i=1}^m \lambda_i^{(k)} \left(R_{2,i}^b \left(\tilde{\mathbf{U}}_2^{(k-1)} \right) + \sum_a \frac{\partial R_{2,i}^a \left(\tilde{\mathbf{U}}_2^{(k-1)} \right)}{\partial \tilde{U}_2^b} \tilde{U}_2^{a,(k-1)} \right) \tag{13}$$

$$= \sum_a \int_{\Omega} N_2^b N_2^a U_2^a, \quad \text{for all } b,$$

$$\sum_b \left(R_{2,i}^b \left(\tilde{\mathbf{U}}_2^{(k-1)} \right) + \gamma \sum_a \frac{\partial R_{2,i}^a \left(\tilde{\mathbf{U}}_2^{(k-1)} \right)}{\partial \tilde{U}_2^b} \tilde{U}_2^{a,(k-1)} \right) \tilde{U}_2^{b,(k)} \tag{14}$$

$$= \sum_b R_{1,i}^b(\mathbf{U}_1) U_1^b + \gamma \sum_a \frac{\partial R_{2,i}^a \left(\tilde{\mathbf{U}}_2^{(k-1)} \right)}{\partial \tilde{U}_2^b} \tilde{U}_2^{a,(k-1)} \tilde{U}_2^{b,(k-1)}, \quad \text{for all } i.$$

For $k = 1$, the algorithm can be initialized taking $\tilde{\mathbf{U}}_2^{(0)} = \mathbf{U}_2$. The Newton–Raphson linearization for (12) corresponds to $\gamma = 1$, whereas a fixed-point (Picard’s) method is obtained for $\gamma = 0$. Note that a Newton–Raphson linearization of (11) would imply a modification of the term multiplying $\tilde{U}_2^{a,(k)}$ in(13).

Let us introduce matrix $\hat{R}_2 \left(\tilde{\mathbf{U}}_2 \right)$ of components

$$\hat{R}_{2,i}^b \left(\tilde{\mathbf{U}}_2 \right) = \sum_a \frac{\partial R_{2,i}^a \left(\tilde{\mathbf{U}}_2 \right)}{\partial \tilde{U}_2^b} \tilde{U}_2^a.$$

We can now write (13)–(14) as

$$\begin{bmatrix} M_2 & -R_2^T - \hat{R}_2^T \left(\tilde{\mathbf{U}}_2^{(k-1)} \right) \\ R_2 + \gamma \hat{R}^T \left(\tilde{\mathbf{U}}_2^{(k-1)} \right) & 0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{U}}_2^{(k)} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} M_2 \mathbf{U}_2 \\ R_1 \mathbf{U}_1 + \gamma \hat{R}^T \left(\tilde{\mathbf{U}}_2^{(k-1)} \right) \tilde{\mathbf{U}}_2^{(k-1)} \end{bmatrix}.$$

The matrix of this system can be made symmetric only if $\gamma = 1$.

As for the linear problem, we could proceed to compute first $\boldsymbol{\lambda}$ by solving the corresponding Schur complement system and then compute $\tilde{\mathbf{U}}_2$.

3. IMPLEMENTATION

The implemented interpolator is based on a search octree method that has been extended for Message Passing Interface (MPI) parallel computing. This algorithm builds cubic boxes around the selected entities (points, elements, or subdomains) and organizes them hierarchically in a way that the search can proceed forwards and backwards recursively without having to apply brute force every time [21]. First of all, an octree of subdomains links each node of the target mesh with the processor containing this coordinate at the old mesh. Each subdomain has its own octree of elements that finds the element of the old mesh that corresponds to a node of the new one. Once all connections have been made and communicated between processors, all arrays of the physical problem can be projected to the target mesh by performing a Lagrangian interpolation within the found element. Finally, the obtained array can go through the optimization problem with restrictions in a completely independent subroutine (Figure 2).

4. APPLICATION TO THE INCOMPRESSIBLE NAVIER–STOKES EQUATIONS

The presented formulation has been applied to the incompressible Navier–Stokes equations solved in an ALE frame of reference. Our motivation to develop the formulation described here is to use it

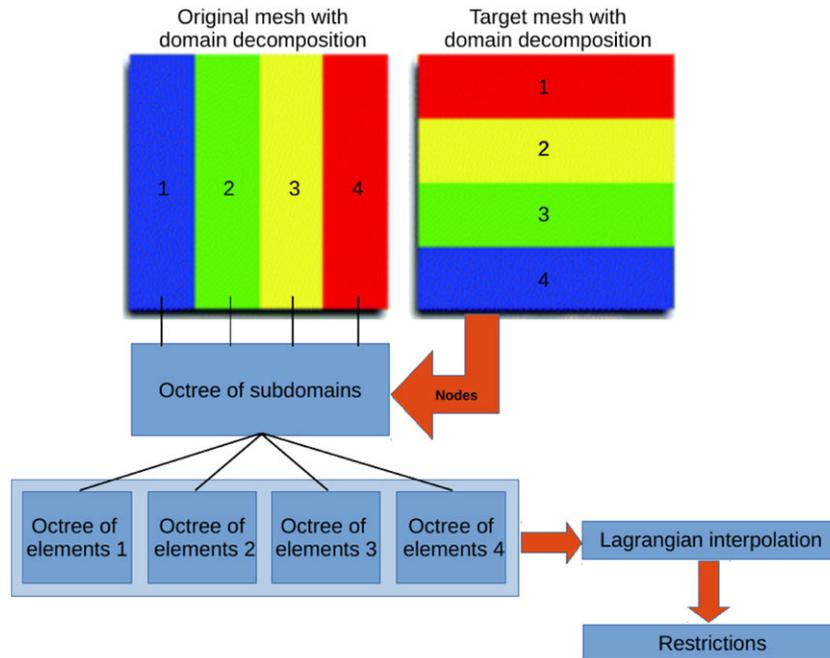


Figure 2. Scheme showing the computational implementation of the interpolation method. Example with four subdomains.

as a first step towards a complete ALE aeroacoustics calculation using Lighthill's acoustic analogy [22], with incompressible flow and 3D dynamic realistic vocal tract geometries. The simulation of syllables with a CFD code represents a challenge in regard to the accuracy of the interpolation of velocity and pressure fields during the remeshing stages. If we consider the numerical calculation of the syllable [as], sharp gradients are expected to appear when the upper incisors approach the lower teeth, which combined with the large deformation of the domain from the initial open position to the final closed configuration, makes a conservative interpolation method extremely necessary in order to prevent the propagation of important numerical error sources.

4.1. Continuous problem

The incompressible Navier–Stokes equations model the motion of isentropic, low-speed viscous flows that do not involve relevant compressibility effects. These are to be solved in a computational domain $\Omega \subset \mathbb{R}^d$ (where $d = 2, 3$ is the number of space dimensions) with boundary $\partial\Omega$ and prescribed initial and boundary conditions. Splitting $\partial\Omega$ into two disjoint sets $\partial\Omega = \Gamma_D \cup \Gamma_N$, the mathematical problem to be faced reads

$$\begin{aligned}
 \partial_t \mathbf{u} - \nu \Delta \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p &= \mathbf{f} && \text{in } \Omega, t > 0, \\
 \nabla \cdot \mathbf{u} &= 0 && \text{in } \Omega, t > 0, \\
 \mathbf{u}(\mathbf{x}, 0) &= \mathbf{u}_0(\mathbf{x}) && \text{in } \Omega, t = 0, \\
 \mathbf{u}(\mathbf{x}, t) &= \mathbf{u}_D(\mathbf{x}, t) && \text{on } \Gamma_D, t > 0, \\
 \mathbf{n} \cdot \boldsymbol{\sigma}(\mathbf{x}, t) &= \mathbf{t}_N(\mathbf{x}, t) && \text{on } \Gamma_N, t > 0,
 \end{aligned}$$

where \mathbf{u} is the velocity, p the pressure, \mathbf{f} the vector of body forces, ν the kinematic viscosity, \mathbf{u}_0 the velocity initial condition, \mathbf{u}_D the velocity Dirichlet condition, \mathbf{n} the exterior unit normal to $\partial\Omega$, $\boldsymbol{\sigma} = -p\mathbf{I} + \nu \mathbf{n} \cdot \nabla \mathbf{u}$, and \mathbf{t}_N the prescribed (pseudo) traction.

Let \mathcal{V}_D the space of vector functions with components in $H^1(\Omega)$ and equal to \mathbf{u}_D on Γ_D and \mathcal{V}_0 the analogous space of functions vanishing on Γ_D . Let also $\mathcal{Q} = L^2(\Omega)$, being the mean of these functions 0 if Γ_N is empty. Let $\langle \cdot, \cdot \rangle_\omega$ be the integral of the product of two functions in the domain ω (with the subscript omitted when $\omega = \Omega$) and (\cdot, \cdot) the $L^2(\Omega)$ -inner product. The weak form of the Navier–Stokes equations can be written as follows: for all time $t > 0$, find $\mathbf{u}(t) \in \mathcal{V}_D$ and $p(t) \in \mathcal{Q}$, with appropriate regularity in time, such that

$$\begin{aligned}
 (\partial_t \mathbf{u}, \mathbf{v}) + \langle \mathbf{u} \cdot \nabla \mathbf{u}, \mathbf{v} \rangle + \nu (\nabla \mathbf{u}, \nabla \mathbf{v}) - (p, \nabla \cdot \mathbf{v}) &= \langle \mathbf{f}, \mathbf{v} \rangle + \langle \mathbf{t}_N, \mathbf{v} \rangle_{\Gamma_N}, \\
 (q, \nabla \cdot \mathbf{u}) &= 0,
 \end{aligned}$$

for all $\mathbf{v} \in \mathcal{V}_0$ and $q \in \mathcal{Q}$ and supplemented with the weak form of the initial conditions.

When the domain Ω is time dependent, this dependency can be taken into account using an ALE formulation. If \mathbf{u}_{dom} is the velocity of the points in Ω , the only modification that needs to be introduced to the previous equations is the replacement of the advection velocity in the convective term by $\mathbf{u} - \mathbf{u}_{\text{dom}}$.

4.2. Finite element approximation and time integration using a monolithic scheme

Let us consider a finite element partition of the domain Ω of size h and use this letter as subscript to denote finite element functions and spaces. Only conforming finite element approximations will be considered in what follows.

Concerning the time integration, the monolithic approach for solving the incompressible Navier–Stokes equations consists in building a linearized system with both velocity and pressure degrees of freedom, which leads to the coupled calculation of the momentum and mass equations in one single step. To approximate the first-order time derivative, a third-order backward finite difference scheme (BDF3) has been used. Let us partition the time interval $[0, T]$ into N equal time steps of size $\delta t := t^{n+1} - t^n$ so that $0 \equiv t^0 < t^1 < \dots < t^n < \dots < t^N \equiv T$. Given a generic time dependent function $g(t)$, the following notation will be used for the BDF3 approximation to the first time derivative:

$$\partial_t g|_{t^{n+1}} \approx \delta_t g^{n+1} := \frac{1}{\delta t} \left(\frac{11}{6} g^{n+1} - 3g^n + \frac{3}{2} g^{n-1} - \frac{1}{3} g^{n-2} \right),$$

where g^n denotes evaluation of g at time step t^n . At an arbitrary time step of the numerical simulation, the final fully discretized implicit scheme in space and time reads as follows. From known \mathbf{u}_h^{n-2} , \mathbf{u}_h^{n-1} , and \mathbf{u}_h^n , compute the incompressible velocity and pressure at time step t^{n+1} , $[\mathbf{u}_h^{n+1}, p_h^{n+1}] \in \mathcal{V}_{D,h} \times \mathcal{Q}_h$, such that

$$\begin{aligned}
 (\delta_t \mathbf{u}_h^{n+1}, \mathbf{v}_h) + \langle \mathbf{u}_h^{n+1} \cdot \nabla \mathbf{u}_h^{n+1}, \mathbf{v}_h \rangle + \nu (\nabla \mathbf{u}_h^{n+1}, \nabla \mathbf{v}_h) - (p_h^{n+1}, \nabla \cdot \mathbf{v}_h) \\
 + (q_h, \nabla \cdot \mathbf{u}_h^{n+1}) = \langle \mathbf{f}^{n+1}, \mathbf{v}_h \rangle + \langle \mathbf{t}_N^{n+1}, \mathbf{v}_h \rangle_{\Gamma_N},
 \end{aligned} \tag{15}$$

for all test functions $[\mathbf{v}_h, q_h] \in \mathcal{V}_{0,h} \times \mathcal{Q}_h$. This is the standard Galerkin approximation of the problem. Stabilization is required to deal with convection-dominated flows. Likewise, the pressure and velocity spaces have to satisfy an adequate inf-sup condition or, otherwise, a stabilization technique is also required. In our calculations, we use the approach described in [23].

4.3. Examples of restrictions for incompressible flows

Let $\mathbf{u}_{h,1}$ be the velocity computed from (15) at a certain time step and with a mesh $\mathcal{T}_{h,1}$. Suppose now that Ω is remeshed and the new finite element partition is $\mathcal{T}_{h,2}$. Using the notation of the previous section, if $\mathbf{u}_{h,1} = \sum_a N_1^a \mathbf{U}_1^a$, the straightforward interpolation will produce $\mathbf{u}_{h,2} = \sum_b N_2^b \mathbf{U}_2^b$. From this, we wish to compute now $\tilde{\mathbf{u}}_{h,2}$, an approximation to $\mathbf{u}_{h,2}$ satisfying certain restrictions.

The imposition of restrictions is obviously problem dependent. In the case of the incompressible Navier–Stokes equations, we will consider *conservation of mass*, *conservation of kinetic energy*, and *conservation of linear momentum*. The first and last restrictions are linear, whereas the second is nonlinear (quadratic). They can be written as

$$\int_{\Omega} \nabla \cdot \tilde{\mathbf{u}}_{h,2} = \int_{\Omega} \nabla \cdot \mathbf{u}_{h,1}, \tag{16}$$

$$\int_{\Omega} |\tilde{\mathbf{u}}_{h,2}|^2 = \int_{\Omega} |\mathbf{u}_{h,1}|^2, \tag{17}$$

$$\int_{\Omega} \tilde{\mathbf{u}}_{h,2} = \int_{\Omega} \mathbf{u}_{h,1}. \tag{18}$$

Note that density can be canceled in all these conditions, as well as the factor 1/2 of the kinetic energy in (17). The mass conservation Equation (16) is crucial; even if we may assume $\mathbf{u}_{h,1}$ to be mass conserving, that is, the right-hand-side of (16) to be 0, this property cannot be guaranteed for $\mathbf{u}_{h,2}$, and therefore, it may be very important to explicitly impose it.

Let λ_1, λ_2 and $\lambda_{2+j}, j = 1, \dots, d$, be the Lagrange multipliers to enforce (16)–(18), respectively. In this case, the Lagrangian functional to be minimized, whose generic expression is (10), is given by

$$\begin{aligned}
 L(\mathbf{v}_{h,2}, \boldsymbol{\mu}) = & \frac{1}{2} \int_{\Omega} \left(\sum_a N_2^a (\mathbf{v}_2^a - \mathbf{U}_2^a) \right)^2 \\
 & - \mu_1 \int_{\Omega} \left(\sum_a \sum_{j=1}^d \partial_j N_2^a V_{2,j}^a - \sum_b \sum_{j=1}^d \partial_j N_1^b U_{1,j}^b \right) \\
 & - \mu_2 \int_{\Omega} \left[\left(\sum_a N_2^a \mathbf{v}_2^a \right)^2 - \left(\sum_b N_1^b \mathbf{U}_1^b \right)^2 \right] \\
 & - \sum_{j=1}^d \mu_{2+j} \int_{\Omega} \left(\sum_a N_2^a V_{2,j}^a - \sum_b N_1^b U_{1,j}^b \right).
 \end{aligned} \tag{19}$$

The restrictions have been expressed in terms of the velocity nodal values that now are vectors whose components have been identified with subscript j .

Now, we can proceed as in the general case, taking the derivatives of L with respect to the nodal unknowns and the Lagrange multipliers to obtain the optimality conditions and then linearizing the resulting nonlinear problem, the nonlinearity being due to the imposition of conserving the kinetic energy.

The optimality conditions read

$$\begin{aligned}
 \frac{\partial L}{\partial V_{2,j}^b} = 0 & \implies \sum_a \int_{\Omega} N_2^b N_2^a \tilde{U}_{2,j}^a - \lambda_1 \int_{\Omega} \partial_j N_2^b - 2\lambda_2 \sum_a \int_{\Omega} N_2^b N_2^a \tilde{U}_{2,j}^a - \lambda_{2+j} \int_{\Omega} N_2^b \\
 & = \sum_a \int_{\Omega} N_2^b N_2^a U_{2,j}^a, \quad \text{for all } b, j = 1, \dots, d, \\
 \frac{\partial L}{\partial \mu_1} = 0 & \implies \int_{\Omega} \sum_a \sum_{j=1}^d \partial_j N_2^a \tilde{U}_{2,j}^a = \int_{\Omega} \sum_b \sum_{j=1}^d \partial_j N_1^b U_{1,j}^b, \\
 \frac{\partial L}{\partial \mu_2} = 0 & \implies \int_{\Omega} \left(\sum_a N_2^a \tilde{\mathbf{U}}_2^a \right)^2 = \int_{\Omega} \left(\sum_b N_1^b \mathbf{U}_1^b \right)^2, \\
 \frac{\partial L}{\partial \mu_{2+j}} = 0 & \implies \int_{\Omega} \sum_a N_2^a \tilde{U}_{2,j}^a = \int_{\Omega} \sum_b N_1^b U_{1,j}^b, \quad j = 1, \dots, d.
 \end{aligned}$$

Using the notation introduced for the general case, a Newton–Raphson linearization of the problem yields

$$\begin{aligned}
 & \sum_a \int_{\Omega} N_2^b N_2^a \tilde{U}_{2,j}^{a,(k)} - \lambda_1^{(k)} \int_{\Omega} \partial_j N_2^b - 2\lambda_2^{(k)} \sum_a \int_{\Omega} N_2^b N_2^a \tilde{U}_{2,j}^{a,(k-1)} - \lambda_{2+j}^{(k)} \int_{\Omega} N_2^b \\
 & = \sum_a \int_{\Omega} N_2^b N_2^a U_{2,j}^a, \quad \text{for all } b, j = 1, \dots, d, \\
 & \int_{\Omega} \sum_a \sum_{j=1}^d \partial_j N_2^a \tilde{U}_{2,j}^{a,(k)} = \int_{\Omega} \sum_b \sum_{j=1}^d \partial_j N_1^b U_{1,j}^b,
 \end{aligned}$$

$$2 \sum_{a,b} \sum_{j=1}^d \int_{\Omega} N_2^b N_2^a \tilde{U}_{2,j}^{a,(k-1)} \tilde{U}_{2,j}^{b,(k)} = \int_{\Omega} \left(\sum_b N_1^b \mathbf{U}_1^b \right)^2 + \int_{\Omega} \left(\sum_a N_2^a \tilde{\mathbf{U}}_2^{a,(k-1)} \right)^2,$$

$$\int_{\Omega} \sum_a N_2^a \tilde{U}_{2,j}^{a,(k)} = \int_{\Omega} \sum_b N_1^b U_{1,j}^b, \quad j = 1, \dots, d.$$

In the case of a Picard’s scheme for the third equation, it would read

$$\sum_{a,b} \sum_{j=1}^d \int_{\Omega} N_2^b N_2^a \tilde{U}_{2,j}^{a,(k-1)} \tilde{U}_{2,j}^{b,(k)} = \int_{\Omega} \left(\sum_b N_1^b \mathbf{U}_1^b \right)^2.$$

Let $R_{n,1}$ and $R_{n,3}$ be the matrices that arise from the imposition of the conservation of mass and linear momentum, respectively, on mesh n , $n = 1, 2$. Let also $\lambda_3 = (\lambda_3, \dots, \lambda_{2+d})$. Using Picard’s linearization for the conservation of kinetic energy, the system to be solved at each iteration is

$$\begin{bmatrix} M_2 & -R_{2,1}^T & -2M_2 \tilde{\mathbf{U}}_2^{(k-1)} & -R_{2,3}^T \\ R_{2,1} & 0 & 0 & 0 \\ (\tilde{\mathbf{U}}_2^{(k-1)})^T & M_2 & 0 & 0 \\ R_{2,3} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{U}}_2^{(k)} \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} M_2 \mathbf{U}_2 \\ R_{1,1} \mathbf{U}_1 \\ \mathbf{U}_1^T M_1 \mathbf{U}_1 \\ R_{1,3} \mathbf{U}_1 \end{bmatrix}.$$

As a measure of convergence of this iterative scheme, we have used precisely the kinetic energy of $\tilde{\mathbf{U}}_2$. In general, convergence is quite fast, but it can deteriorate when dealing with fields with sharp gradients and highly non-coincident meshes. Convergence can be improved using under relaxation.

As in the general case, the Lagrange multipliers can be solved first from a Schur complement system, which is trivially constructed if M_2 is approximated by a diagonal matrix.

4.4. Time integration using a fractional step scheme and pressure interpolation

Instead of using the monolithic approach to integrate in time the Navier–Stokes equations, one can use a fractional step scheme, in which the calculation of the velocity and the pressure are segregated. The interest of this approach is well known; in particular, it is known that the computational cost is drastically reduced, particularly for problems with many degrees of freedom. In general, within each time step fractional step schemes consist in the computation of a non divergence-free velocity in a first stage and a Poisson problem for the pressure in a second step; finally, the velocity is corrected in order to fulfill the incompressibility condition and consistency (see [24] for a review).

Let us consider the simplest backward Euler discretization of the Navier–Stokes equations combined with a second-order fractional step method. Supposing that there are only Dirichlet conditions, the equations to be solved are

$$\frac{1}{\delta t} (\hat{\mathbf{u}}_h^{n+1} - \mathbf{u}_h^n, \mathbf{v}_h) + \langle \hat{\mathbf{u}}_h^{n+1} \cdot \nabla \hat{\mathbf{u}}_h^{n+1}, \mathbf{v}_h \rangle + \nu (\nabla \hat{\mathbf{u}}_h^{n+1}, \nabla \mathbf{v}_h) - (p_h^n, \nabla \cdot \mathbf{v}_h) = \langle \mathbf{f}^{n+1}, \mathbf{v}_h \rangle,$$

$$(\nabla p_h^{n+1}, \nabla q_h) = (\nabla p_h^n, \nabla q_h) + \frac{1}{\delta t} (\nabla \cdot \hat{\mathbf{u}}_h^{n+1}, q_h),$$

$$\frac{1}{\delta t} (\mathbf{u}_h^{n+1} - \hat{\mathbf{u}}_h^{n+1}, \mathbf{v}_h) + (\nabla p_h^{n+1}, \mathbf{v}_h) - (\nabla p_h^n, \mathbf{v}_h) = 0,$$

where $\hat{\mathbf{u}}$ is the non-solenoidal intermediate velocity. Once again, the stabilization terms needed for using equal interpolation for the velocity and pressure fields have been omitted for the sake of simplicity.

An important particularity of the second order fractional step scheme presented is that the momentum equation uses the pressure field of the previous time step. If the computational domain is remeshed at a certain time step, not only the velocity but also the pressure must be interpolated to the target mesh when the calculation restarts. In order to ensure an accurate interpolation, the conservation of the L^2 norm of the pressure can be imposed as a restriction. Proceeding as described

for the general case, the calculation of the pressure nodal values $\tilde{\mathbf{P}}_2$ from \mathbf{P}_2 , computed from the standard interpolation of \mathbf{P}_1 , consists in the optimization of the functional:

$$L(q_{h,2}, \boldsymbol{\mu}) = \frac{1}{2} \left\| \sum_a N_2^a (Q_2^a - P_2^a) \right\|_{L^2(\Omega)}^2 - \mu \int_{\Omega} \left(\sum_{a,b} Q_2^b N_2^b N_2^a Q_2^a - \sum_{c,d} P_1^d N_1^d N_1^c P_1^c \right), \quad (20)$$

where we have assumed that functions in the pressure spaces are interpolated also with the shape functions $\{N_i^a\}$, $i = 1, 2$.

Now, we can proceed as for the velocity interpolation with restrictions, obtaining the optimality conditions for (20), linearizing them, solving for the optimal Lagrange multiplier λ from a Schur complement problem and, finally, obtaining $\tilde{\mathbf{P}}_2$ as a correction of \mathbf{P}_2 with λ known. Details are omitted.

5. NUMERICAL RESULTS

5.1. Method validation: interpolation between identical translated meshes

Before proceeding to analyze the correction performed by the restrictions on the interpolated field in common application cases, such ALE calculations, a first case consisting in an analytical solution is presented to assess the dissipation caused by the non-conservative interpolation and the efficiency of the compensation introduced by the present method. This function is interpolated from an original structured mesh of bilinear elements with dimensionless size $h = 0.025$ covering the domain $\Omega = [0, 1] \times [0, 1]$ to an identical mesh that has been shifted to the right with elements of size $h/2$, so the old and the new nodes do not coincide at all [25]. Afterwards, the function is interpolated back to the original mesh, repeating this sequence up to 20 times. The analytical solution is given by $\mathbf{u} = (u_x, u_y)$, with

$$\begin{aligned} u_x &= 2x^2y(x-1)^2(y-1)(2y-1), \\ u_y &= -2y^2x(y-1)^2(x-1)(2x-1). \end{aligned}$$

Because $\mathbf{u}|_{\Gamma} = \mathbf{0}$, where $\Gamma = \partial\Omega$, only the conservation of momentum and L^2 norm are applied. Table I shows the evolution of the maximum of $|\mathbf{u}|$ during the first interpolation process: after a 2% loss caused by the point-by-point interpolation, the correction brings the maximum back to its original value with a small overflow. In spite of the benefit, this scenario shows that this global method does not respect the local features of the solution, such as peak values. However, these deviations are very small and new spurious peaks are not possible because the method enforces that the final solution remains the nearest possible to the interpolated one. The same happens with the sign preservation and the monotonicity of the solution, because the method does not enforce them explicitly and some unwanted sign changes might appear at points where the function or its derivative are very close to 0. Nonetheless, these errors will be always negligible for the same reason.

Next, the accumulation of the interpolation error along the whole sequence and its compensation are analyzed. After the 20th interpolation the dissipation is considerable as it can be seen in the following series of plots of $|\mathbf{u}|$, see Figure 3, which use the same scale in order to reflect the degradation

Table I. Maximum values of $|\mathbf{u}|$ at the first interpolation stage.

Original value	0.013650
After interpolation without restrictions	0.013621
Interpolation with restrictions	0.013653

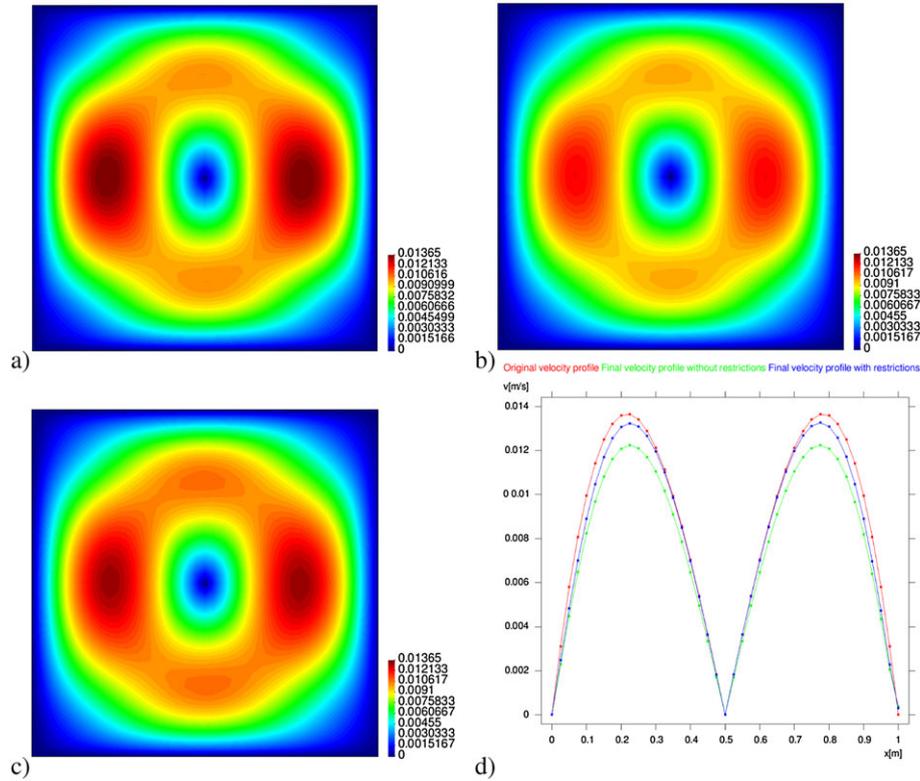


Figure 3. (a) Original profile of $|u|$, (b) final profile of $|u|$ after interpolating without restrictions, (c) final profile of $|u|$ after interpolating with restrictions, and (d) superposition of the previous profiles at $y = 0.5$.

Table II. Maximum values of the final norm of $|u|$.

Original value	0.013650
After interpolation without restrictions	0.012244
Interpolation with restrictions	0.013261

of the solution. Figure 3(d) and Table II show how the imposition of momentum and L^2 norm conservation compensate most of the dissipation.

5.2. Flow around a cylinder

In order to show the benefit of the presented formulation in a straightforward way, a simple $Re = 500$ flow around a cylinder benchmark case has been computed with the monolithic approach of the incompressible Navier–Stokes equations in a static domain and a fine finite element mesh (see [26] for more details). During 2000 time steps, five equispaced interpolations to a coarse mesh have been performed in two different ways: without applying any restrictions and enforcing the conservation of mass, momentum, and kinetic energy to the velocity field. After each interpolation, the results have been projected back to the original mesh after only one time step, in order to avoid further distortion of the solution than that caused by the interpolation error. This procedure allows the comparison of the results to a reference case that undergoes the same interpolation stages, but projecting the velocity field always to the same, slightly moved, fine mesh. Therefore, in this case, no accuracy losses are expected other than those because of the interpolation between meshes and the results can be considered as ideal for our purpose. The mesh parameters of the simulation are listed in Table III.

In spite of the small differences expected for only five bidirectional interpolations in an overall run of 2000 time steps, a considerable phase error between the reference curve and the solution computed without applying restrictions can be clearly observed in Figure 4, which shows the last

Table III. Parameters for the flow over a cylinder example.

Mesh	Fine	Coarse
No. of elements	63,413	13,860
No. of nodes	32,445	7079
Element type	Stabilized P1/P1	Stabilized P1/P1

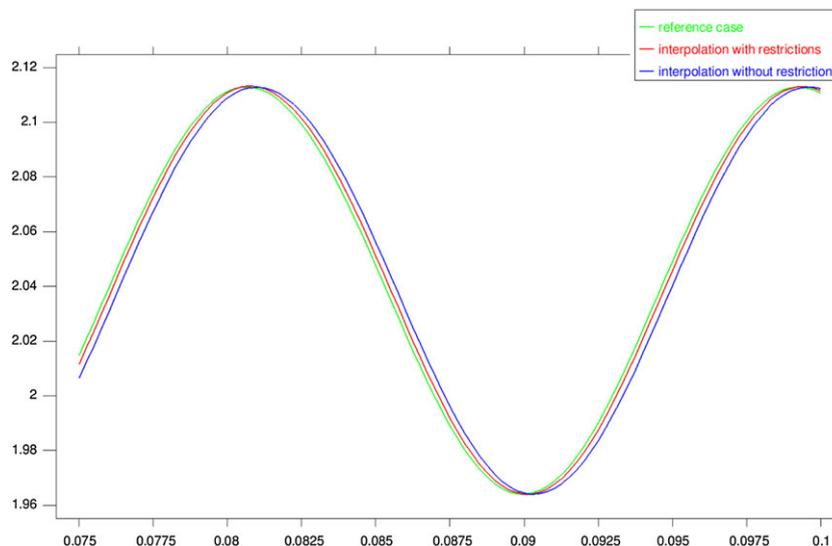


Figure 4. Time history of the last time steps of the velocity field.

steps of the velocity time history. On the other hand, the red curve, corresponding to the velocity interpolated with restrictions, nearly fits the reference solution. This proves that the imposition of the restrictions leads to a reduction of the interpolation error and its propagation.

5.3. A problem in aeroacoustics using an ALE description and remeshing

Following the practical motivation of this work, a preliminary case representing the syllable [as] is presented next. It consists in an air flux (density 1.2 kg/m^3 and viscosity $1.5 \cdot 10^{-5} \text{ Pa s}$) flowing through a constriction in a simplified two-dimensional geometry based on the 3D simplified model of a vocal tract [27]. The computation starts from a developed flow computed with the incompressible Navier–Stokes equations on the initial open configuration corresponding to the phoneme [a] (Figure 5). The upper part of the vocal tract has been prescribed an initial velocity ramp followed by a constant vertical velocity of -5 m/s until the final position corresponding to the fricative sound is reached, which happens after a total closure path of 5.15 cm at $t = 0.012 \text{ s}$. During this dynamic phase, an ALE frame of reference is used and a total of seven remeshing stages are performed because of the fast distortion of the elements. Moreover, the domain velocity and the closure itself change the flow characteristics, for which reason the time step has been changed progressively from the initial value $\delta t = 10^{-5} \text{ s}$ to a final size of $\delta t = 2.5 \cdot 10^{-6} \text{ s}$. The computational domain, from now on referred as Ω , with an inlet velocity of $(2.4, 0) \text{ m/s}$, rigid walls with a non-slip condition except the vertical outlet boundary, has been discretized with two different finite element meshes (Figure 6), whereas the time integration scheme is BDF3 in all cases. Therefore, three time components of the velocity are interpolated in order to guarantee a smooth and accurate restart of the computation on the new mesh.

It will be shown how the accuracy of the interpolation, and hence, the impact of the restrictions, depend, as expected, on the degree of coincidence between the donor and the target mesh. Therefore, the same case will be run twice: first, interpolating always to a similar mesh as usual and then going from the original fine mesh to a coarse mesh and vice versa. This scenario might not be usual in ALE calculations, but is indeed common in the computation of coupled problems with two or more

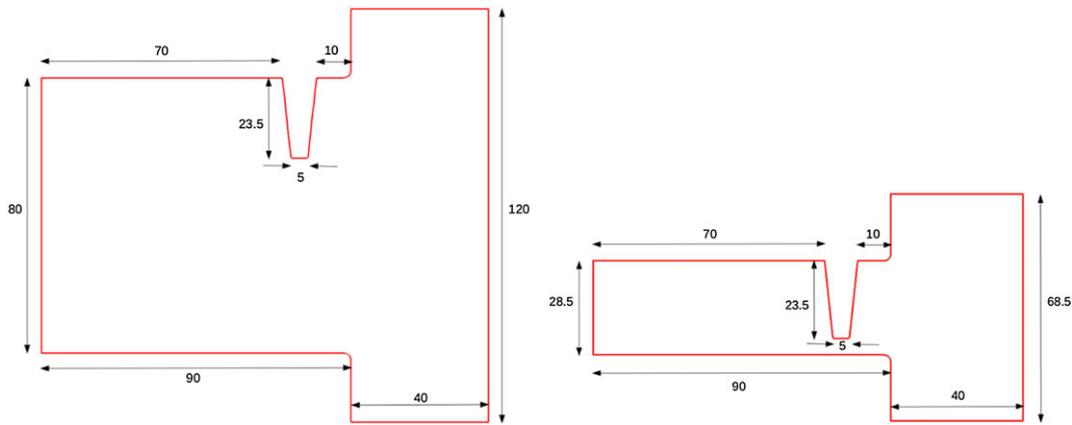


Figure 5. Initial geometry corresponding to phoneme [a] (left), final geometry corresponding to phoneme [s] (right).

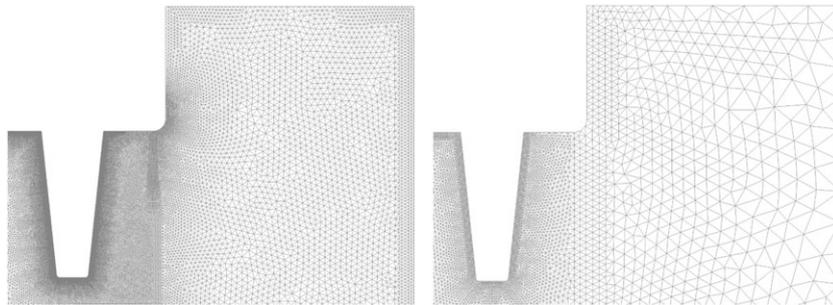


Figure 6. Original fine mesh (left) and coarse mesh used at the first interpolation (right). Dimensions in millimeters.

Table IV. Mesh parameters for the aeroacoustics example.

Initial mesh	Fine	Coarse
No. of elements	254,907	16,784
No. of nodes	131,058	8706
Element type	Stabilized P1/P1	Stabilized P1/P1
Maximum size	0.001	0.005
Minimum size	10^{-5}	$2 \cdot 10^{-4}$

different meshes, such as FSI and aeroacoustics, where the acoustic source term arising from the CFD calculation can be projected to a coarser mesh for the less demanding wave equation [28]. The results will be also analyzed from both a static and a dynamic point of view. In the first one, the effect of the restrictions within the interpolation will be assessed, whereas the latter will focus on the propagation of the interpolation error along a complete simulation (Table IV).

5.3.1. Monolithic scheme. As stated earlier, the presented case has been run twice for analyzing the dependence of the formulation on the degree of coincidence between meshes. In both cases the following restrictions for the velocity field of the incompressible Navier–Stokes equations have been applied together: conservation of mass, conservation of momentum in x and y , and conservation of kinetic energy. On the other hand, when they are applied separately the compensation performed by each one turns to be nearly the same, for which reason the decoupled case will not be presented. Moreover, this fact reveals that the non-diagonal terms of the system matrix, which force the simultaneous fulfillment of all restrictions, scarcely modify the values of the Lagrange multipliers and only compensate the linearization. A direct solver has been used in order to better evaluate the

interpolation error at the remeshing stages and its propagation, because it provides the exact discrete solution of the problem.

5.3.1.1. Static analysis. This subsection intends to illustrate the performance of the restrictions in a single interpolation. The procedure is called static because the projection of results takes place in a fixed domain. Table V summarizes the corrections performed by each restriction in all the remeshing stages from the initial geometry [a] to the final configuration [s] in the aforementioned two different cases: projection to a similar fine mesh ($F \rightarrow F$) and intercalation of coarse meshes ($F \rightarrow C$ or $C \rightarrow F$). It shows the amount of mass as well as the deviation percentage of momentum and kinetic energy that needs to be compensated at each interpolation when all restrictions are applied together. First of all, the first interpolation to the coarse mesh with restrictions will be compared with that without them and to the equivalent ideal projection to a similar fine mesh, in order to check how far this method can compensate a non-accurate interpolation. Later on, the most critical interpolation will be carefully analyzed by evaluating the contribution of the restrictions, not only as a whole, but seeing how they perform individually.

It can be clearly seen, as expected, that the interpolations going from a fine mesh to a coarse one ($F \rightarrow C$) are far more inaccurate than the others because of the low degree of coincidence between nodes. If we consider the first interpolation, which departs from the same solution in both cases, an interesting analysis regarding the peak value can be derived. The current formulation does not conserve explicitly the peak value, because this would require local restrictions, which are not part of the scope of this global method. However, the injection of mass, energy, and momentum by the imposition of restrictions performs indeed a correction on the peak values that have been shaved by the interpolation. Table VI shows the maximum values of the velocity norm, which appear around the obstacle, at the first remeshing stage.

Although the resulting peak value after applying the restrictions will mostly depend on the capability of the mesh of capturing local sharp gradients, which is not always possible given a low degree of coincidence between meshes and a coarse resolution of the target mesh, the method manages to restore partially the lost peaks, above all in the coarse solution, which needs a greater correction than the fine one.

Table V. Corrections performed by each restriction at all interpolation stages.

	No. of interpolation							
	1 $F \rightarrow F$	1 $F \rightarrow C$	2 $F \rightarrow F$	2 $C \rightarrow F$	3 $F \rightarrow F$	3 $F \rightarrow C$	4 $F \rightarrow F$	4 $C \rightarrow F$
Mass	8.4e-5	0.029	1.0e-5	1.7e-4	3.4e-6	0.014	5.0e-5	5.7e-4
MomX (%)	0.029	0.542	0.034	0.022	0.027	0.174	0.031	0.016
MomY (%)	4.3e-4	0.004	0.002	1.8e-4	4.9e-3	0.113	9.3e-3	4.3e-3
Kinet. (%)	0.027	0.328	0.049	0.031	0.079	0.514	0.101	0.062
	5 $F \rightarrow F$	5 $F \rightarrow C$	6 $F \rightarrow F$	6 $C \rightarrow F$	7 $F \rightarrow F$	7 $F \rightarrow C$		
Mass	1.8e-4	9.2e-3	2.5e-4	1.3e-3	2.9e-4	0.052		
MomX (%)	0.040	0.332	0.052	0.022	0.026	1.263		
MomY (%)	3.7e-3	0.145	4.9e-3	8.0e-3	0.013	0.069		
Kinet. (%)	0.143	1.208	0.201	0.116	0.267	1.908		

Table VI. Maximum values of the velocity norm at the first remeshing stage.

Original value	29.637	
Target mesh	Fine	Coarse
Interpolation without restrictions	29.591	29.397
Interpolation with restrictions	29.600	29.435

The last interpolation to the coarse mesh is clearly the most critical one. The losses of mass and momentum are considerable, but the most remarkable fact is the loss of kinetic energy, which nearly reaches the 2%. The next series of plots show the correction performed by each restriction separately and all together to the velocity field in Figure 7, which corresponds to the last remeshing stage at the final geometry (Figure 5).

Figure 8 shows the non-conservative interpolation of the velocity field to the coarse mesh. For a better tracking of the whole process, the same scaling has been used. The conservation of mass is achieved by injecting fluid through the only open boundary, the outlet, as shown in (Figure 9), taking

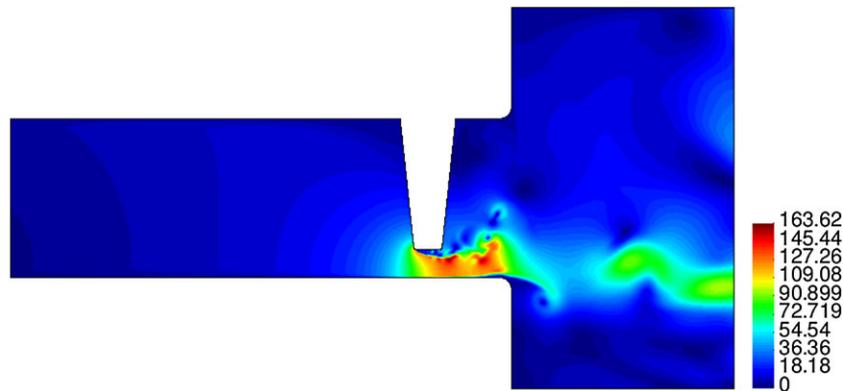


Figure 7. Velocity field before the last interpolation to the coarse mesh.

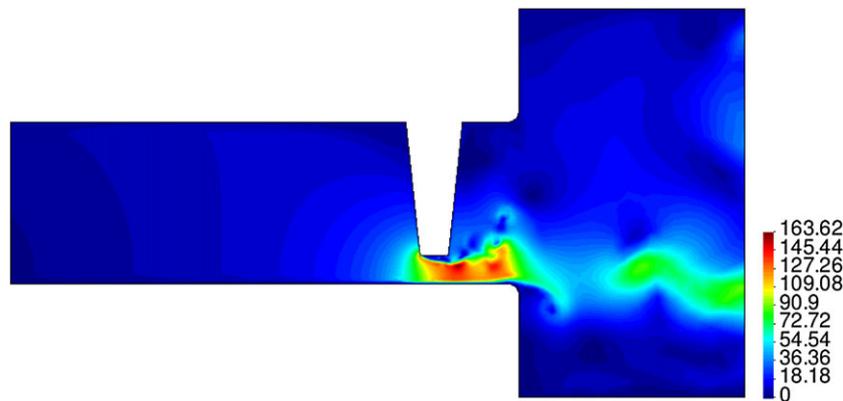


Figure 8. Velocity field after the last interpolation to the coarse mesh before being corrected by the restrictions.

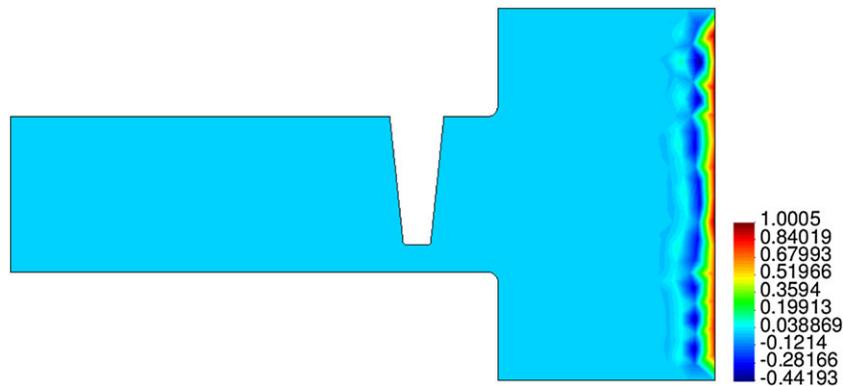


Figure 9. Correction of the velocity field performed by the conservation of mass.

into account the divergence theorem. Then, the mass is diffused into the bulk by a consistent mass matrix. However, the correction is essentially restricted to the boundary nodes because it must also satisfy that the final solution remains the nearest possible to the interpolated field. The correction of momentum follows a mesh dependent discontinuous pattern, because the corresponding restriction operator is the integral of the shape functions at the nodes (Figure 10). On the other hand, the correction of the kinetic energy follows perfectly the motion of the flow because it can be understood as the conservation of the velocity L^2 norm. Bearing in mind that this correction has a quadratic effect, the injection of kinetic energy will be proportional to the velocity field in order to minimize the relative variation of the solution (Figure 11). The total correction scale (Figure 12) shows the key

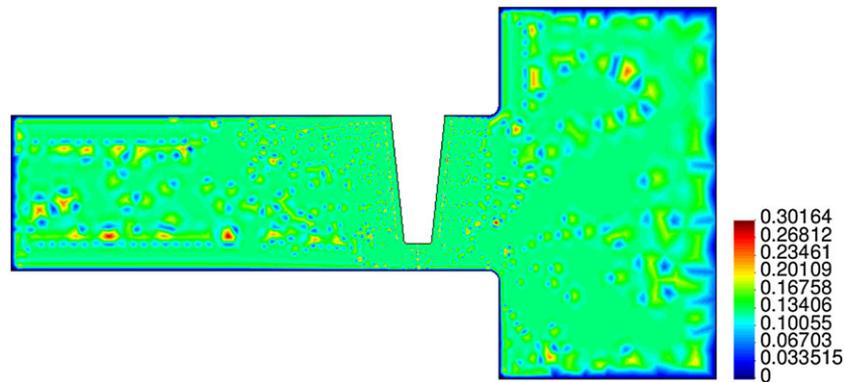


Figure 10. Correction of the velocity field performed by the conservation of momentum.

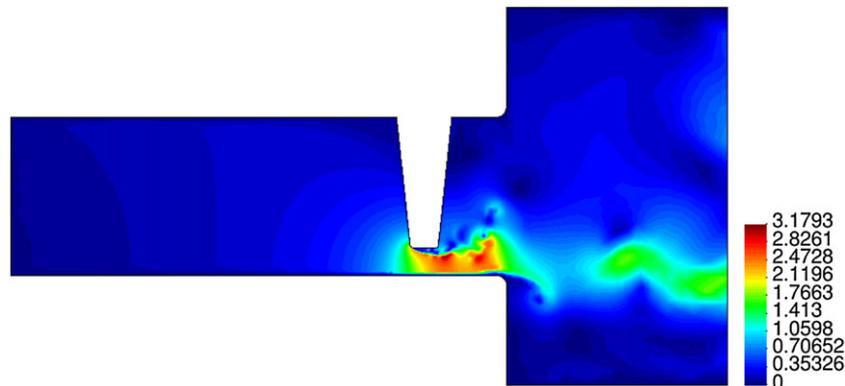


Figure 11. Correction of the velocity field performed by the conservation of kinetic energy.

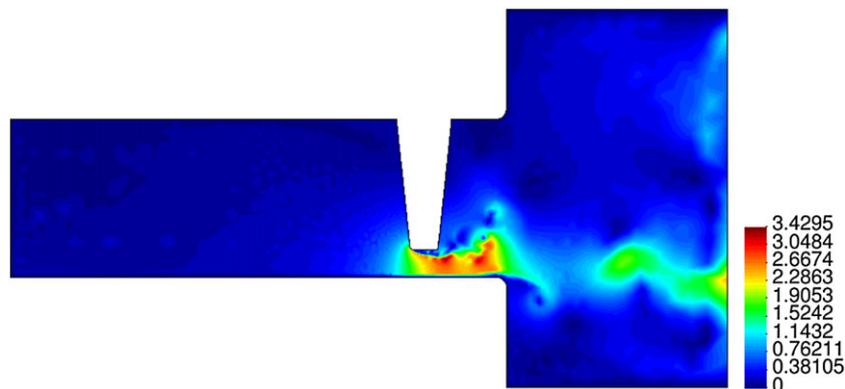


Figure 12. Total correction of the velocity when applying all restrictions together.

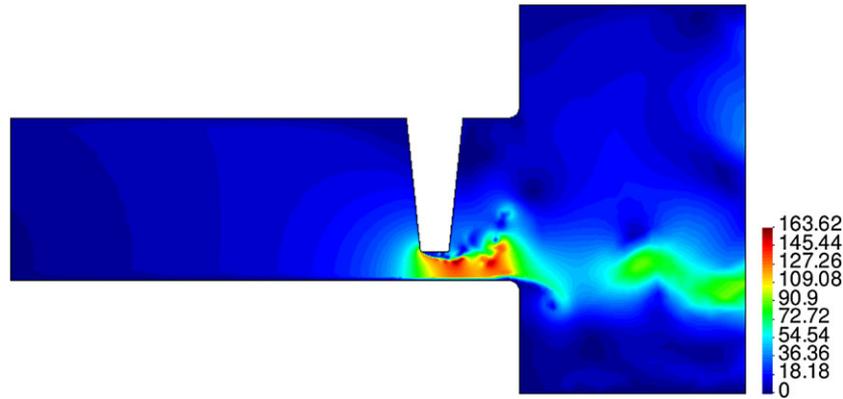


Figure 13. Final velocity field after applying all restrictions.

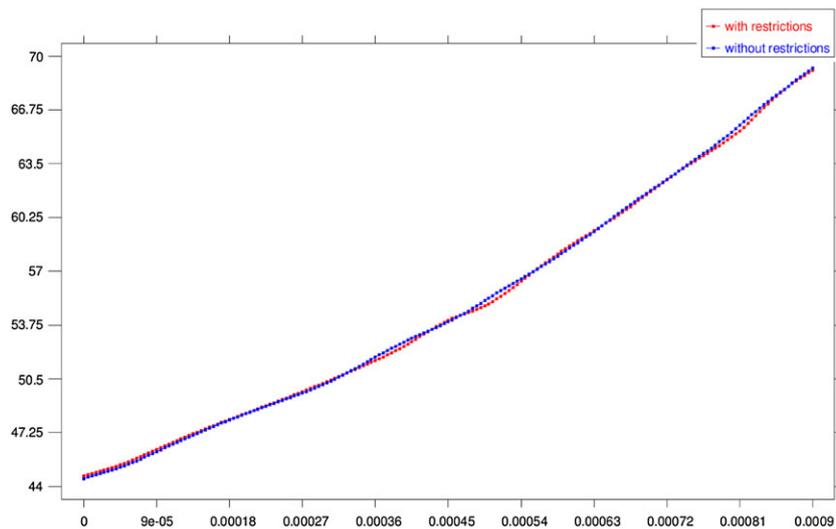


Figure 14. Time history of the velocity between interpolations 5 and 6.

importance of this method when dealing with highly non-coincident grids, a typical scenario that can be found in some coupled problems, otherwise the solution could get distorted in just one time step. Finally, Figure 13 shows the final velocity field after applying the previous correction, which recovers most of the flow features of the original field that had been lost during the non-conservative interpolation (Figures 7 and 8).

5.3.1.2. Dynamic analysis. Up to now, only the static transmission of information between two meshes has been analyzed. However, the most important features are the degradation of the results and the accumulation of error during a dynamic calculation when no restrictions are applied, as well as the global effect of the restrictions along the simulation. It is not easy to quantify the benefit just by adding the effect of the successive application of the restrictions because after losing mass, momentum, and energy at one interpolation, the numerical solution of the Navier–Stokes equations is computed again, and this interpolation error is partially compensated by the imposition of the incompressibility condition as well as by the injection of momentum and kinetic energy coming from the inlet boundary condition. For this reason, if we did not apply the restrictions and we measured the losses at the last interpolation, these values would not reveal the accumulated error, but just a frozen picture of the accuracy of a single interpolation. Instead, the accumulation of error arises from an interpolated value that does not fulfill the physical problem features and that is later propagated when the computation restarts.

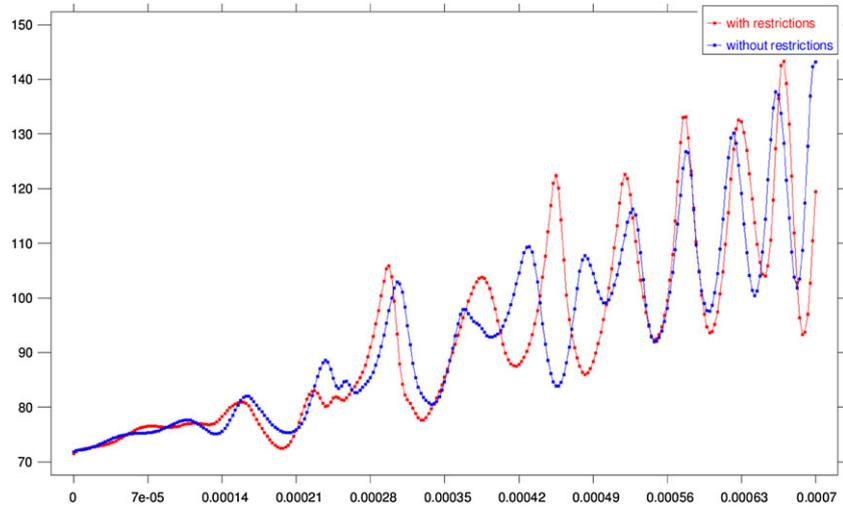


Figure 15. Time history of the velocity between interpolations 6 and 7.

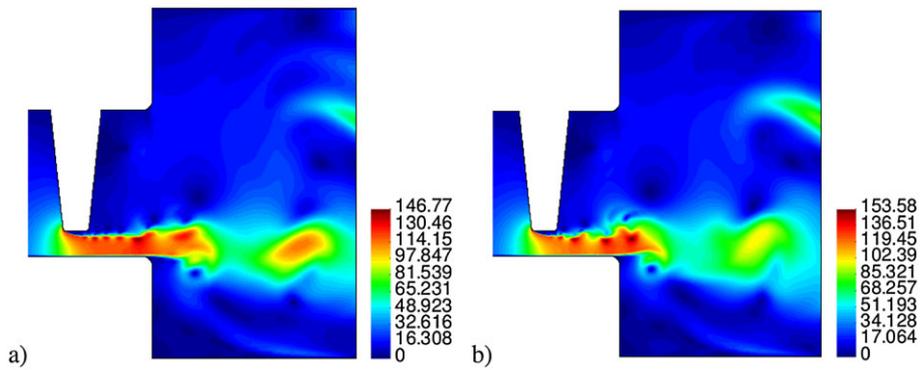


Figure 16. Final velocity profile applying no restrictions (a) and final velocity profile applying all restrictions (b).

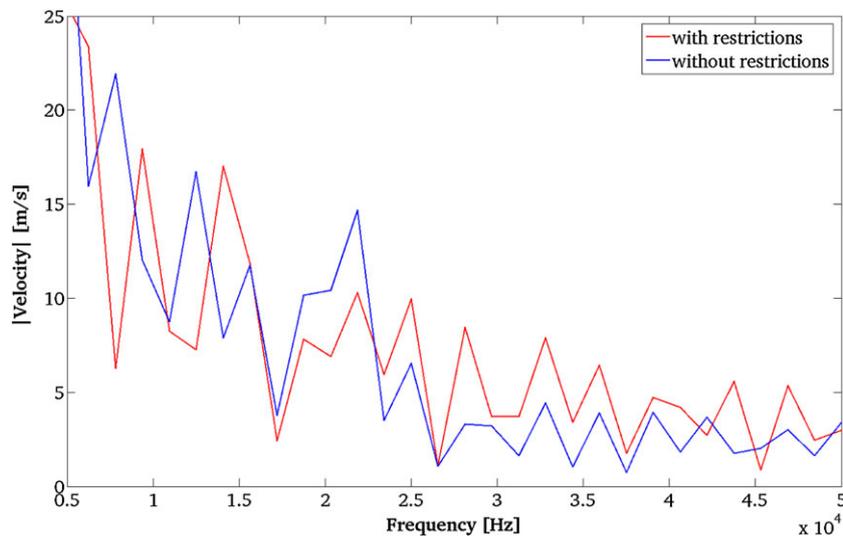


Figure 17. FFT of the velocity between interpolations 6 and 7.

In this case, only interpolations to similar fine meshes have been considered. Unlike the previous case, the numerical error cannot be assessed comparing the velocity before and after the interpolation stage, because the high degree of coincidence between meshes avoids a visible dissipation. The matter of study here is the accumulation of this error; hence, the time histories of the velocity at a point under the tooth (78.7, 5.33) mm during the last two stages of the simulation (Figures 14 and 15), and the final velocity fields at the configuration [s] have been compared (Figure 16). In spite of keeping the same mean value, the solutions start diverging when the first vortices go across the analyzed point. Between interpolations 5 and 6, the tooth is approximating its final position but the

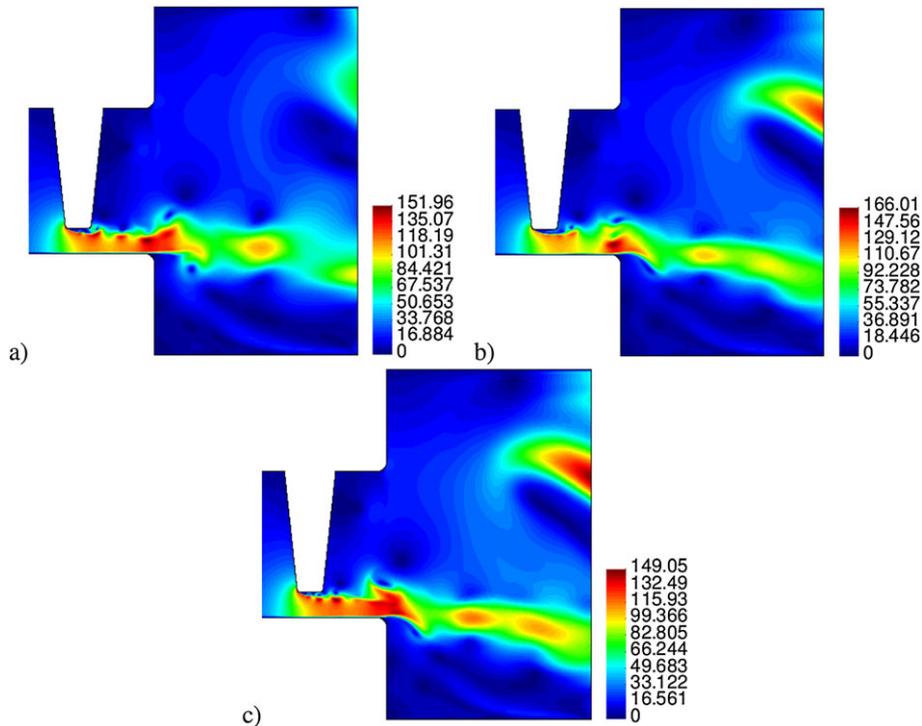


Figure 18. Final velocity profile applying no restrictions (a), final velocity profile applying only velocity restrictions (b), and final velocity profile applying all restrictions (c).

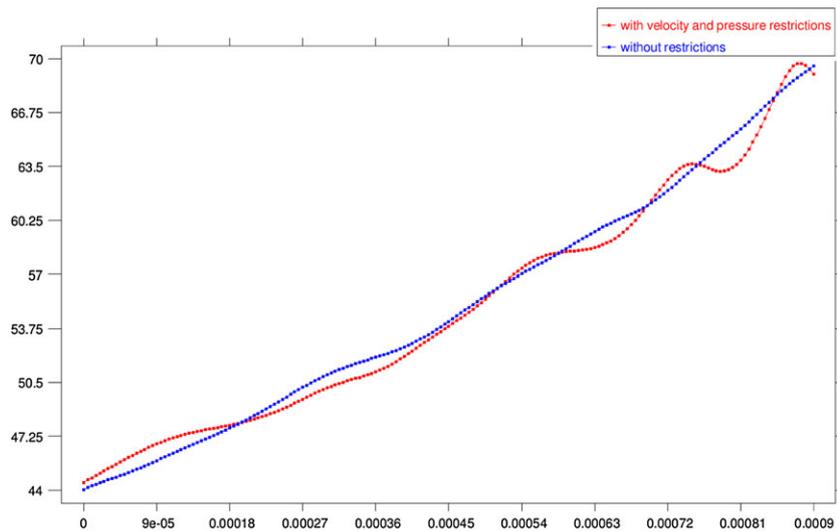


Figure 19. Time history of the velocity between interpolations 5 and 6.

flow under the obstacle has not yet experimented important oscillations and the two solutions show a good agreement. However, at the following stage between interpolations 6 and 7, just before reaching the final position, the sudden presence of sharp velocity gradients amplifies the propagation of the initial interpolation error, leading to different peak values and to a considerable phase change. The impact of the application of the restrictions in the oscillating solution can be clearly observed in Figure 17, where the fast Fourier transform of both velocity time histories has been computed.

5.3.2. Second-order fractional step scheme. The same case as for the monolithic scheme is now solved using a fractional step scheme. The same computational meshes have also been used. The problem has been computed again three times: without restrictions, with only velocity restrictions, and with velocity and pressure restrictions as presented in Section 4, in order to evaluate the benefits of the interpolation procedure applied to the pressure field (Figure 18).

Once again, in spite of showing some similar patterns, the solutions yield significant discrepancies regarding the peak value and the phase of oscillation. Unlike the monolithic case, the solutions with and without restrictions diverge at the stage between interpolations 5 and 6. Figure 19 shows that the

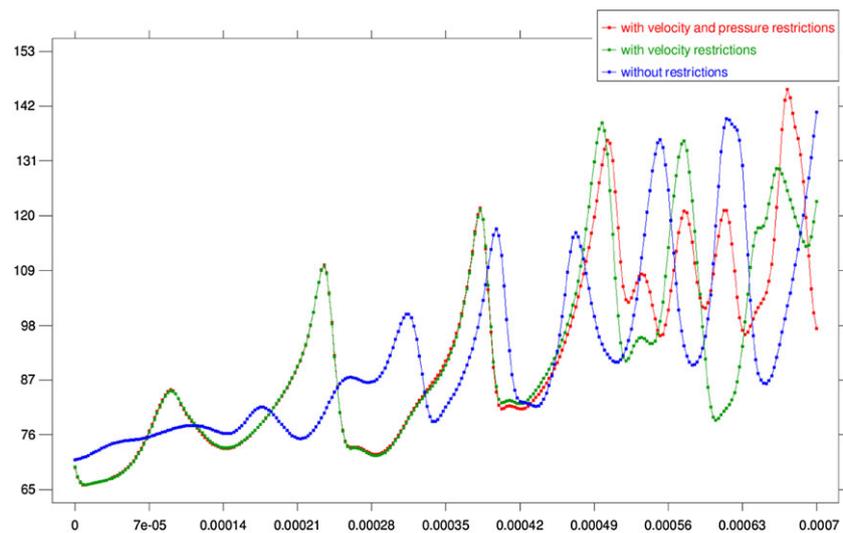


Figure 20. Time history of the velocity between interpolations 6 and 7.

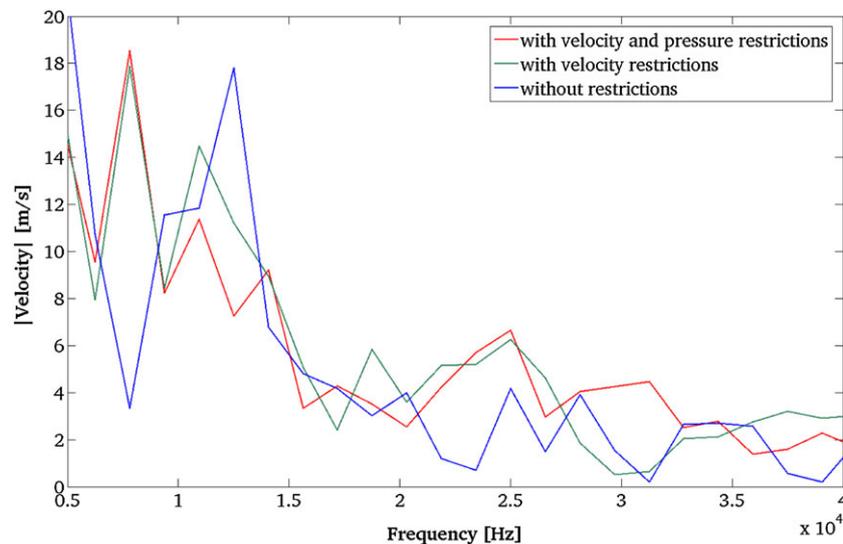


Figure 21. FFT of the velocity between interpolations 6 and 7.

red solution starts oscillating earlier than the blue one. In this case, the solution without enforcing the pressure constraint has not been plotted because it coincides perfectly with the solution applying all restrictions. On the other hand, at the following stage (Figure 20), these two solutions finally diverge once sharp gradients appear at the analyzed point below the tooth, which shows that the enforcement of the pressure L^2 norm conservation is as important as the velocity restrictions to prevent phase error. This can also be observed in Figure 21, where the Fourier transform of the velocity between to interpolation steps has been depicted.

6. CONCLUSIONS

The presented formulation has proved to be very effective in compensating the interpolation error when using two highly non-coincident computational grids, where it was already expected that the loss of mass, momentum, and kinetic energy would be relevant in flow problems. It has also turned to be of crucial importance in scenarios, such as in ALE calculations, where the interpolations take place between similar fine meshes. In this case, the effect of the restrictions on the interpolated array is not so relevant, but the accumulation and propagation of this theoretically small error because of the non-conservative interpolation turns to be relevant indeed after several thousands of time steps. One of the most important factors regarding the accuracy of the interpolation between meshes is the presence of sharp gradients as well as the mesh resolution and coincidence to capture them. Although this is a global method, it also manages to partially restore the lost peaks in the interpolated field.

The method has been applied successfully to the solution of the incompressible Navier–Stokes equations. In this case, the most relevant restriction is the conservation of kinetic energy. It has also been proved that the propagation of interpolation error when using a non-conservative interpolation becomes considerable after several hundreds of time steps. Although the mean value is conserved, the oscillation pattern becomes clearly divergent when the vorticity increases. In a similar way, for second-order fractional schemes the conservation of the pressure L^2 norm has turned to be necessary to prevent phase error.

From a computational point of view, the method yields a fast convergence and a very low computational cost and is flexible enough for being easily extended to other physical problems with other restrictions to be imposed.

ACKNOWLEDGEMENTS

This work is partially supported by the EU-FET grant EUNISON 308874 and the Spanish Government Elastic-Flow project DPI2015-67857-R. The authors thankfully acknowledge the computer resources, technical expertise, and assistance provided by the Red Espanola de Supercomputacion (RES-BSC). Moreover, the first author would also like to acknowledge the Agencia de Gestio d'Ajuts Universitaris i de Recerca for the predoctoral FI grant no. 2015 FI-B 00227. The second author gratefully acknowledges the support received from the Catalan Government through the ICREA Acadèmia Research Program.

REFERENCES

1. Houzeaux G, Codina R. Transmission conditions with constraints in finite element domain decomposition methods for flow problems. *Communications in Numerical Methods in Engineering* 2001; **17**(3):179–190.
2. Codina R, Houzeaux G, Coppola-Owen H, Baiges J. The fixed-mesh ALE approach for the numerical approximation of flows in moving domains. *Journal of Computational Physics* 2009; **228**(5):1591–1611.
3. Donea J, Huerta A, Ponthot JP, Rodriguez-Ferran A. Arbitrary Lagrangian–Eulerian methods. In *Encyclopedia of Computational Mechanics*, Vol. 1: Fundamentals, Stein E, de Borst R, Hugues TJR (eds.) Wiley: New York, 2004.
4. Margolin LG, Shashkov M. Second-order sign-preserving conservative interpolation (remapping) on general grids. *Journal of Computational Physics* 2003; **184**(1):266–298.
5. Garimella R, Kucharik M, Shashkov M. An efficient linearity and bound preserving conservative interpolation (remapping) on polyhedral meshes. *Computers & Fluids* 2007; **36**(2):224–237.
6. Ramshaw JD. Conservative rezoning algorithm for generalized two-dimensional meshes. *Journal of Computational Physics* 1985; **59**(2):193–199.
7. Dukowicz JK, Kodis JW. Accurate conservative remapping (rezoning) for arbitrary Lagrangian–Eulerian computations. *SIAM Journal on Scientific and Statistical Computing* 1987; **8**(3):305–321.
8. Scroggs JS, Semazzi FHM. A conservative semi-Lagrangian method for multidimensional fluid dynamics applications. *Numerical Methods for Partial Differential Equations* 1995; **11**(5):445–452.

9. Chesshire G, Henshaw WD. A scheme for conservative interpolation on overlapping grids. *SIAM Journal on Scientific Computing* 1994; **15**(4):819–845.
10. Alauzet F, Mehrenberger M. P1-conservative solution interpolation on unstructured triangular meshes. *International Journal for Numerical Methods in Engineering* 2010; **84**(13):1552–1588.
11. Jiao X, Heath MT. Common-refinement-based data transfer between non-matching meshes in multiphysics simulations. *International Journal for Numerical Methods in Engineering* 2004; **61**(14):2402–2427.
12. Farrell PE, Piggott MD, Pain CC, Gorman GJ, Wilson CR. Conservative interpolation between unstructured meshes via supermesh construction. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**(33):2632–2642.
13. Farrell PE, Maddison JR. Conservative interpolation between volume meshes by local Galerkin projection. *Computer Methods in Applied Mechanics and Engineering* 2011; **200**(1):89–100.
14. Chippada S, Dawson CN, Martinez ML, Wheeler MF. A projection method for constructing a mass conservative velocity field. *Computer Methods in Applied Mechanics and Engineering* 1998; **157**(1):1–10.
15. Cebal JR, Lohner R. Conservative load projection and tracking for fluid–structure problems. *AIAA Journal* 1997; **35**(4):687–692.
16. Farhat C, Lesoinne M, Le Tallec P. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: momentum and energy conservation, optimal discretization and application to aeroelasticity. *Computer Methods in Applied Mechanics and Engineering* 1998; **157**(1):95–114.
17. Jaiman RK, Jiao X, Geubelle PH, Loth E. Assessment of conservative load transfer for fluid–solid interface with non-matching meshes. *International Journal for Numerical Methods in Engineering* 2005; **64**(15):2014–2038.
18. Bernardi C. A New Nonconforming approach to domain decomposition: the mortar element method. In *Nonlinear Partial Differential Equations and Their Applications*: Amsterdam, 1994.
19. Chiandussi G, Bugada G, Oñate E. A simple method for automatic update of finite element meshes. *Communications in Numerical Methods in Engineering* 2000; **16**(1):1–19.
20. Brancherie D, Villon P, Ibrahimbegović A, Rassineux A, Breitkopf P. Transfert de champs par approximation diffuse avec conservation de l'énergie. *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique* 2006; **15**(1–3):107–118.
21. Löhner R. Robust, vectorized search algorithms for interpolation on unstructured grids. *Journal of computational Physics* 1995; **118**(2):380–387.
22. Lighthill MJ. On sound generated aerodynamically. I. general theory. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 1952; **211**(1107):564–587.
23. Codina R. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**(39):4295–4321.
24. Badia S, Codina R. Algebraic pressure segregation methods for the incompressible Navier–Stokes equations. *Archives of Computational Methods in Engineering* 2007; **15**:1–52.
25. Bussetta P, Boman R, Ponthot J-P. Efficient 3D data transfer operators based on numerical integration. *International Journal for Numerical Methods in Engineering* 2015; **102**(3–4):892–929.
26. Guasch O, Codina R. Computational aeroacoustics of viscous low speed flows using subgrid scale finite element methods. *Journal of Computational Acoustics* 2009; **17**(03):309–330.
27. Cisonni J, Nozaki K, Van Hirtum A, Grandchamp X, Wada S. Numerical simulation of the influence of the orifice aperture on the flow around a teeth-shaped obstacle. *Fluid Dynamics Research* 2013; **45**(2):025505.
28. Kaltenbacher M, Zörner S, Hüppe A, Sidlof P. 3D numerical simulations of human phonation. *Proceedings of the 11th world congress on computational mechanics-WCCM XI*, IACM, Barcelona, 2014.