

## MAST-RT0 solution of the incompressible Navier–Stokes equations in 3D complex domains

Costanza Aricò, Marco Sinagra, Calogero Picone & Tullio Tucciarelli

To cite this article: Costanza Aricò, Marco Sinagra, Calogero Picone & Tullio Tucciarelli (2021) MAST-RT0 solution of the incompressible Navier–Stokes equations in 3D complex domains, *Engineering Applications of Computational Fluid Mechanics*, 15:1, 53-93, DOI: [10.1080/19942060.2020.1860830](https://doi.org/10.1080/19942060.2020.1860830)

To link to this article: <https://doi.org/10.1080/19942060.2020.1860830>



© 2020 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group



Published online: 29 Dec 2020.



Submit your article to this journal [↗](#)



Article views: 1211







View related articles [↗](#)



View Crossmark data [↗](#)

# MAST-RT0 solution of the incompressible Navier–Stokes equations in 3D complex domains

Costanza Aricò , Marco Sinagra , Calogero Picone  and Tullio Tucciarelli 

Department of Engineering, University of Palermo, Palermo, Italy

## ABSTRACT

A new numerical methodology to solve the 3D Navier–Stokes equations for incompressible fluids within complex boundaries and unstructured body-fitted tetrahedral mesh is presented and validated with three literature and one real-case tests. We apply a fractional time step procedure where a predictor and a corrector problem are sequentially solved. The predictor step is solved applying the MAST (Marching in Space and Time) procedure, which explicitly handles the non-linear terms in the momentum equations, allowing numerical stability for Courant number greater than one. Correction steps are solved by a Mixed Hybrid Finite Elements discretization that assumes positive distances among tetrahedrons circumcentres. In 3D problems, non-Delaunay meshes are provided by most of the mesh generators. To maintain good matrix properties for non-Delaunay meshes, a continuity equation is integrated over each tetrahedron, but the momentum equations are integrated over clusters of tetrahedrons, such that each external face shared by two clusters belongs to two tetrahedrons whose circumcentres have positive distance. A numerical procedure is proposed to compute the velocities inside clusters with more than one tetrahedron. Model preserves mass balance at the machine error and there is no need to compute pressure at each time iteration, but only at target simulation times.

## ARTICLE HISTORY

Received 3 August 2020  
Accepted 1 December 2020

## KEYWORDS

Navier–Stokes equations; 3D numerical model; Eulerian method; unstructured tetrahedral mesh; predictor–corrector scheme; Mixed Hybrid Finite element

## 1. Introduction

The Navier–Stokes Equations (NSEs) govern external and internal flows in many real-life industrial, environmental and biological problems (e.g. aircraft and ship problems, rotomachinery blade applications, hemodynamic and biological flows). The challenging research topics involved in the numerical solution of such mathematical problems mainly concern the choice of the velocity–pressure coupling algorithm and the design of the computational mesh discretizing the physical domain.

In compressible flows, pressure and density are linked by the state algebraic equation. In contrast, in incompressible flows, density is constant and pressure has to be solved along with velocities from all the momentum and continuity equations. With this motivation, the numerical algorithms used to solve the NSEs can generally be divided into density-based and pressure-based solvers (Shyy & Mittal, 1998; Tao, 2001). Density-based solvers are commonly applied to high-Ma compressible flows, while pressure-based solvers were originally proposed for incompressible flows, and then successfully extended to compressible flows (Tao, 2001).

Two approaches are generally used in pressure-based solvers, namely the direct (or coupled) approach and the segregated approach (e.g. Mazhar, 2016 and cited references). In the first case, the whole set of momentum and continuity equations is solved simultaneously, resulting in a strong coupling between pressure and velocity. The main drawback is the large amount of required computational effort and computer memory, which makes this approach unsuitable to facing many practical engineering applications (Darwish et al., 2015). In the segregated approach, pressure and velocity are solved separately and sequentially, using previously computed values of the other dependent variable. The core of the problem is how to update the pressure field so that the divergence-free velocity field condition is satisfied.

Several numerical procedures can be distinguished in the segregated approach, e.g. the projection method (Chorin, 1968; Kim & Moin, 1985), the penalty method (Braaten & Shyy, 1986; Hughes et al., 1979), the artificial compressibility method (Harlow & Welch, 1965; Malan et al., 2002; Vrahliotis et al., 2012), and the pressure-correction method (Ozoe & Tao, 2001; Patankar, 1980).

**CONTACT** Costanza Aricò  [costanza.arico@unipa.it](mailto:costanza.arico@unipa.it)

The fractional step projection method (Chorin, 1968; Kim & Moin, 1985) and the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) pressure-correction method (Patankar, 1980, 1981) have become very popular. In both methodologies, a pressure correction is introduced to improve the velocities computed from the solution of the momentum equations, and to satisfy the continuity equation. Projection methods, commonly regarded as fractional-step methods, can be classified into pressure-correction methods, velocity-correction methods and consistent splitting methods (Guermond et al., 2006). A sequence of decoupled elliptic equations for velocity and pressure has to be solved at each time step, and this represents one of the most attractive features of such procedures, especially for simulations of large-scale problems (Guermond et al., 2006). In the projection method, the pressure correction Poisson equation is solved once per time step, while for the SIMPLE method, the momentum and pressure correction equations are solved several times in each time step. For these reasons, projection methods could handle unsteady flow problems more easily than SIMPLE.

The SIMPLE-family algorithms (e.g. SIMPLER, SIMPLEC, SIMPLEX) are extensions of and improvements to the SIMPLE algorithm. Other pressure correction methods can be regarded as further extensions of the SIMPLE-family models, for example PISO and CLEAR (e.g. Aguerre et al., 2020 and cited works). Segregated solvers like the SIMPLE-family algorithms have shown poor convergence, especially when used for swirling flow fields (Hanby & Silvester, 1996). In these problems, where the coupling between radial and tangential momentum equations is strong, the linearization of the momentum equations leads to a sequential solution of these equations, without accounting for the coupling between the momentum equations and the velocity components (Hanby & Silvester, 1996). Even though several ‘ad hoc’ procedures have been presented in order to overcome the above-mentioned convergence problems (e.g. Gosman et al., 1976), the need of parameter calibration for convergence of the iterative procedure requires a high computational effort, which limits the application of these algorithms. These reasons, along with the increase of computer performance and memory, have motivated several authors towards new coupled approaches (e.g. Darwish et al., 2009 and cited references).

Other options have been proposed in the literature, e.g. the vorticity-stream function methods, where pressure is eliminated from the governing equations and velocity and pressure are replaced by the vorticity and the stream function (e.g. Calhoun, 2002 and cited references). Two main reasons have limited their use: the difficulty of handling wall boundary conditions and the

difficulties arising in the extension from 2D to 3D problems (Calhoun, 2002).

It is widely recognized that spatial discretization of the incompressible flow equations on collocated grids leads to unphysical odd–even coupling of the pressure (i.e. the so-called spurious checkerboard modes) (e.g. Dalal et al., 2008; Perron et al., 2004). Staggered grids have been one of the most common ways to overcome these problems, where different grid points for velocity and pressure are used (e.g. Harlow & Welch, 1965; Perot, 2000).

In the last decade, the use of unstructured grids has become popular due to their capacity to discretize real arbitrary domains and to easily get local refinement. Geometric complexity is a drawback for a straightforward extension of the staggered mesh to unstructured grids and most solvers require the addition of terms which could generate unphysical solutions and loss of mass conservations (e.g. Perot, 2000). Moreover, the time step limitation required by the Courant condition can become very severe, due to the existence of few computational elements much smaller than the average ones.

In the past few decades, Finite Volume Methods (FVMs) (e.g. Kim & Choi, 2000; Mathur & Murthy, 1997; Perron et al., 2004; Plana Fattori et al., 2013; Vidovic et al., 2004) and Finite Element Methods (FEMs) (e.g. Bazilevs et al., 2013; Fortin, 1981; Pai et al., 2013; Zienkiewicz et al., 2013) have been preferred to other methods (e.g. finite difference methods) in handling irregular boundaries, because they can use unstructured triangular or tetrahedral meshes, which can easily discretize arbitrary geometries. A large multitude of FVMs has been proposed in the last few decades, with different choices of control volumes, both for collocated and staggered grids. Many of these proposed techniques suffer from a ‘non-orthogonality condition’, as pointed out in (Gao et al., 2012), in discretization of the second-order partial derivative terms in the momentum equations and in the pressure-correction Poisson equation. Hybrid FV/FEMs (e.g. Busto et al., 2018; Gao et al., 2012) take advantage of both FVMs and FEMs, discretizing the momentum equations with the FVMs and the Poisson equation with the FEMs. The Discontinuous Galerkin Finite Element Methods (DG-FEMs) (e.g. Bassi & Rebay, 1997; Lehrenfeld & Schöberl, 2016 and cited references) combine the performing features of the FEMs and FVMs. Like the classical FEMs, the DG-FEMs achieve high-order accuracy using high-order polynomial approximation within an element rather than using wide stencils, as in the case of finite volume methods. The physics of wave propagation is however correctly accounted for by the solution of a Riemann problem (e.g. Toro et al., 2020 and cited references) arising from the discontinuous solution at element interfaces, which makes them similar to FVMs (e.g. Toro

& Vazquez-Cendon, 2012). The main drawback of DG-FEMs, compared to the classical FEMs and FVMs, is that they require solutions of systems of equations with more unknowns for the same grids, and have been recognized to be very demanding in terms of both computational costs and storage requirements.

Recently, mesh-free methods have received increasing attention from researchers, due to their ability to solve flow problems with complex geometries or involving multi-phases fluids. The Smoothed Particle Hydrodynamics (SPH) method is a pure Lagrangian, mesh-free procedure, originally proposed for the solution of astrophysical problems (Gingold & Monaghan, 1977; Lucy, 1977). The fluid domain is discretized by a set of moving particles, and the governing equations are solved for each of them. Each particle has a support domain with a characteristic spatial distance over which their physical properties (namely pressure and velocity) are 'smoothed' by means of a kernel function, generally with a Gaussian-type shape. Thanks to its quite easy implementation, this procedure has largely been applied for the solution of compressible and incompressible NSEs (e.g. Oger et al., 2007 and cited references). One of the main drawbacks of the Lagrangian schemes is that the relations among particles have to be updated at each time step, and the topology setting of the matrix system of the Poisson pressure-correction equations has to be performed at each time iteration, as well as its factorization operations.

During the last years, Virtual Elements Methods (VEMs) (e.g. Beirão da Veiga et al., 2018, and cited references) have been proposed as a new FEM paradigm to solve Partial Differential Equations. VEMs construct a conforming Galerkin FE scheme, dealing with general polygonal/polyhedral mesh elements, also with non-convex shape. VEMs have been recently applied for the solution of the NSEs in 2D problems (e.g. Beirão da Veiga et al., 2018, and cited references), but their extension to 3D problems is still limited to very simple domain geometries and boundary shapes (Liu & Chen, 2019).

In the present paper, we propose a new finite volume solver for the solution of 3D incompressible NSEs over unstructured tetrahedral meshes. We apply a fractional time step procedure where a predictor and a corrector problem are sequentially solved. The procedure presents substantial differences compared to the fractional step procedures presented in the literature, based upon the common projection procedures (e.g. Guermond et al., 2006; Perron et al., 2004). The predictor step (PS) is carried out by applying the Eulerian Finite Volume MAST (Marching in Space and Time) numerical procedure, recently proposed for the solution of shallow waters and groundwater problems (Aricò et al., 2007; Aricò et al.,

2011, 2012, 2013a, 2013b; Aricò & Tucciarelli, 2007a, 2007b, 2009). In this step, all the terms are retained in the momentum equations. The major advantages of MAST are the following: (1) it explicitly handles the non-linear momentum terms in the momentum equations, by means of the sequential solution of a three-variable Ordinary Differential Equations (ODEs) system for each computational cell, with a computational effort which is simply proportional to the number of tetrahedral elements, (2) it provides numerical stability with respect to large Courant–Friedrichs-Lewy (CFL) numbers, that can be much greater than one, at a cost of local accuracy reduction (Aricò et al., 2007; Aricò & Tucciarelli, 2007a, 2007b). The correction step is split into two parts, named CS1 step and CS2 step. In CS1 step, three linear systems are solved for the three velocity components (one system for each velocity component), which update the viscous terms in the momentum equations. In the second corrector step (CS2), a single linear system is solved for the pressure correction unknown. The matrices of the systems of CS1 and CS2 steps are well conditioned, as they are sparse, symmetric and diagonally dominant, and lead to a very fast solution of the associated systems by the use of a preconditioned conjugated gradient solver. The matrix coefficients are constant in time, computed and factorized only once at the beginning of the numerical simulation. This makes it possible to save a lot of computational time, compared to other numerical schemes (e.g. Lagrangian schemes).

Both CS1 and CS2 steps are solved assuming a mass lumping Mixed Hybrid Finite Element (MHFE) (e.g. Auricchio et al., 2017, and cited references) discretization inside each tetrahedron, similar to the one proposed by Younes et al. (2006). The mass lumping option has been chosen because it is easy to be used together with tetrahedral elements. To maintain good convergence and accuracy properties, our MHFE scheme assumes the distance among circumcenters to be positive, a condition which is always satisfied in the Delaunay meshes. Unfortunately, in 3D problems either bad-quality Delaunay or non-Delaunay meshes are provided by most of the available mesh generators (Li & Teng, 2001). To cope with this problem, and to use non-Delaunay meshes still saving the previously mentioned good matrix properties, in the present procedure, a continuity equation is integrated over each single tetrahedron, but the momentum equations are integrated over clusters of tetrahedrons, such that each external face shared by two different clusters is part of two tetrahedrons whose circumcenters have positive distance. To define the velocities inside each cluster with more than one tetrahedron correctly, the minimum energy condition is finally enforced for velocities inside the clusters.

In the proposed numerical solver, mass balance is always preserved at the error machine precision, and there is no need to update the pressure values at each time iteration, because only the pressure gradient appear in the NSEs. We compute the spatial pressure distribution only at target simulation times.

The paper is organized as follows. In section 2.1, we present the governing equations and the projection-correction formulation of the problem and in section 2.2 the spatial discretization. In sections 3.1, 3.2, and 3.3, we present the numerical details of the solution of respectively the PS, CS1 and CS2 steps as defined in section 2.1. In section 4, we present the extension of the previous algorithms to the real case of 3D non-Delaunay meshes and in section 5, we show the model application to three well-known literature tests and one real-life case, as well as an analysis of the associated computational costs.

## 2. RT0 spatial discretization of the governing equations

### 2.1. Governing equations and fractional time step discretization

We solve the 3D Navier–Stokes Equations (NSEs) for a real and incompressible fluid,

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla \Psi - \nu \nabla^2 \mathbf{u} = 0 \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (2)$$

where Equations (1) are the momentum conservation equations, Equation (2) is the mass conservation equation,  $t$  is time,  $\nu$  kinematic viscosity,  $\mathbf{u}$  the velocity vector whose components are  $u$ ,  $v$ , and  $w$ , along the  $x$ ,  $y$  and  $z$  directions respectively, and  $\psi$  is the kinematic pressure  $p/\rho$ , where  $p$  is the fluid pressure and  $\rho$  is the fluid density, constant in space and time for an incompressible fluid. The governing equations are solved for the  $\mathbf{u}$  and  $\psi$  unknowns.

The problem is well-posed if we correctly assign the initial and boundary conditions (ICs and BCs, respectively). With respect to the BCs, we assign either (a) all the velocity components (essential BCs) or (b) the stress vector (natural BCs) or (c) a combination of the previous ones called free-slip BC. Let  $\Omega$  be the computational domain and  $\Gamma$  its boundary surface, and let us call  $\Gamma_u$ ,  $\Gamma_\sigma$  and  $\Gamma_m$  the three non-overlapping portions, where BCs (a), (b) and (c) respectively apply, such that  $\Gamma = \Gamma_u + \Gamma_\sigma + \Gamma_m$ . We formulate the BCs as

$$\mathbf{u}(\mathbf{x}) = \mathbf{g}(\mathbf{x}), \mathbf{x} \in \Gamma_u, t \geq 0 \quad (3a)$$

$$\boldsymbol{\sigma}(\mathbf{x}) = \left( \Psi - 2\nu \frac{\partial u_n}{\partial n} \right) \mathbf{n} - \nu \left( \frac{\partial u_n}{\partial \tau_1} + \frac{\partial u_{\tau_1}}{\partial n} \right) \mathbf{t}_1$$

$$- \nu \left( \frac{\partial u_n}{\partial \tau_2} + \frac{\partial u_{\tau_2}}{\partial n} \right) \mathbf{t}_2, \mathbf{x} \in \Gamma_\sigma, t \geq 0 \quad (3b)$$

$$u_n(\mathbf{x}) = 0 \text{ and } \boldsymbol{\tau}(\mathbf{x}) = 0, \mathbf{x} \in \Gamma_m, t \geq 0 \quad (3c)$$

where  $\mathbf{x}$  is the coordinate position vector,  $\mathbf{g}$  is the velocity vector assigned at the boundary,  $\mathbf{n}$  is the unit outward vector normal to the boundary, and  $\mathbf{t}_1$  and  $\mathbf{t}_2$  are two other orthogonal unit vectors such that  $\mathbf{n}$ ,  $\mathbf{t}_1$  and  $\mathbf{t}_2$  give the reference frame attached to the tetrahedron face,  $\boldsymbol{\tau}$  is the stress tangent vector component in the  $\mathbf{t}_1$ - $\mathbf{t}_2$  plane,  $u_n$ ,  $u_{\tau_1}$  and  $u_{\tau_2}$  are the corresponding  $\mathbf{u}$  components. In the following sections, for simplicity's sake we assume only hydrostatic stress occurring along  $\Gamma_\sigma$ , which is equivalent to assuming the stress normal to the boundary plane and to neglecting all the viscous terms in Equation (3b).

The assigned ICs on the system, in the  $\bar{\Omega}$  ( $\bar{\Omega} = \Omega \cap \Gamma$ ) domain are

$$\mathbf{u} = \mathbf{u}_0 \quad \text{with} \quad \nabla \cdot \mathbf{u}_0 = 0 \quad \text{and} \quad \Psi = \Psi_0 \text{ at } t = 0 \quad (4)$$

As mentioned in section 1, we apply a predictor–corrector projection procedure, sequentially solving one predictor and two corrector problems. The predictor and the first corrector steps deal with the momentum equations, while in the second corrector step we combine the mass and momentum conservation equations to enforce the divergence-free condition.

In the next sections, time levels  $t^k$ ,  $t^{k+1/3}$ ,  $t^{k+2/3}$ ,  $t^{k+1}$  represent the beginning of the generic time iteration, the end of the prediction step, as well as the end of the first and second corrector steps, respectively, and  $t^{k+1}$  also marks the end of the time iteration. Superscripts  $k$ ,  $k+1/3$ ,  $k+2/3$  and  $k+1$  mark the values of the variables (i.e.  $\mathbf{u}$  and  $\psi$ ) at the corresponding time levels.

In vector-matrix form, we write the momentum Equations (1) as

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = \mathbf{B}(\mathbf{U}) + \nabla \cdot \mathbf{E}(\mathbf{U}) \quad (5a)$$

$$\mathbf{U} = (u \quad v \quad w)^T, \mathbf{F}(\mathbf{U}) = (\mathbf{F}_1 \quad \mathbf{F}_2 \quad \mathbf{F}_3)^T,$$

$$\mathbf{E}(\mathbf{U}) = (\mathbf{E}_1 \quad \mathbf{E}_2 \quad \mathbf{E}_3)^T$$

$$\mathbf{B}(\mathbf{U}) = -(\nabla_x \Psi \quad \nabla_y \Psi \quad \nabla_z \Psi)^T \quad (5b)$$

$$\mathbf{F}_1 = \begin{pmatrix} uu \\ uv \\ uw \end{pmatrix}^T, \mathbf{F}_2 = \begin{pmatrix} vu \\ vv \\ vw \end{pmatrix}^T, \mathbf{F}_3 = \begin{pmatrix} wu \\ wv \\ ww \end{pmatrix}^T \quad (5c)$$

$$\mathbf{E}_j = \begin{pmatrix} v \frac{\partial u}{\partial x_j} & v \frac{\partial v}{\partial x_j} & v \frac{\partial w}{\partial x_j} \end{pmatrix}^T \quad j = 1, 2, 3 \quad x_1 = x, x_2 = y, x_3 = z \quad (5d)$$



In the framework of a fractional time step procedure, we set

$$\mathbf{E} = \mathbf{E}^{k-1/3} + (\mathbf{E} - \mathbf{E}^{k-1/3}) \quad (6a)$$

$$\mathbf{B} = \mathbf{B}^k + (\mathbf{B} - \mathbf{B}^k) \quad (6b)$$

and we split Equation (5a) into

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} = \nabla \cdot \mathbf{E}^{k-1/3} + \mathbf{B}^k \quad (7a)$$

$$\frac{\partial \mathbf{U}}{\partial t} = \nabla \cdot \mathbf{E} - \nabla \cdot \mathbf{E}^{k-1/3} \quad (7b)$$

$$\frac{\partial \mathbf{U}}{\partial t} = \mathbf{B} - \mathbf{B}^k \quad (7c)$$

where  $\mathbf{E}^{k-1/3}$  marks the matrix  $\mathbf{E}$  computed at the end of the first correction system of the previous time iteration and  $\mathbf{B}^k$  marks the pressure gradient term at the beginning of the new time step. Functional analysis easily shows that, due to the stationarity of the pressure gradient term, Equation (7a) form a fully convective system, with only one characteristic line passing through the  $(\mathbf{x}, t)$  point, while the systems in Equation (7b) and (7c) are fully parabolic (Aricò et al., 2007; Aricò & Tucciarelli, 2009). Integrating in time, from Equation (7) we get

$$\mathbf{U}^{k+1/3} - \mathbf{U}^k + \nabla \cdot \int_0^{\Delta t} \mathbf{F} dt = \nabla \cdot \mathbf{E}^{k-1/3} \Delta t + \mathbf{B}^k \Delta t \quad (8a)$$

$$\mathbf{U}^{k+2/3} - \mathbf{U}^{k+1/3} = \nabla \cdot \mathbf{E}^{k+2/3} \Delta t - \nabla \cdot \mathbf{E}^{k-1/3} \Delta t \quad (8b)$$

$$\mathbf{U}^{k+1} - \mathbf{U}^{k+2/3} = (\mathbf{B}^{k+1} - \mathbf{B}^k) \Delta t \quad (8c)$$

where system (8a) is the prediction step (PS), (8b) and (8c) are the first and second correction systems (CS1 and CS2), respectively.

Summing systems (8), the integral of the original system (1) is formally obtained. Further details on time discretization will be given below in section 3.

## 2.2. RT0 spatial tetrahedral discretization of pressure and velocity

We discretize the computational domain by means of  $N_T$  non-overlapping tetrahedrons (named also elements) and assume the velocity field in each tetrahedron  $e$ ,  $\mathbf{u}_e(\mathbf{x}) \in \mathbf{X}_e$ , where  $\mathbf{X}_e$  is the lowest-order Raviart-Thomas (RT0) space (Raviart & Thomas, 1977), such that

$$\mathbf{u}_e(\mathbf{x}) = \sum_{j=1,4} \omega_j^e Q_j^e \quad \text{with } \omega_j^e = \frac{(\mathbf{x} - \mathbf{x}_j^e)}{3W_e} \quad (9)$$

where  $\omega_j^e$  are the space basis functions of  $\mathbf{X}_e$ ,  $W_e$  is the volume of tetrahedron  $e$ ,  $\mathbf{x}_j^e$  is the coordinate vector of the

$j$ th node of  $e$ , and  $Q_j^e$  is the volumetric flux crossing the face of  $e$  opposite to the  $j$ th node. Important properties of the space  $\mathbf{X}_e$  are that  $\nabla \cdot \mathbf{u}_e$  is constant over  $e$ ,  $\mathbf{u}_e \cdot \mathbf{n}_j$  is constant over each face  $j$  of  $e$ , and  $\mathbf{n}_j$  represents the unit outward vector orthogonal to face  $j$  of tetrahedron  $e$  (Raviart & Thomas, 1977). As a result of these properties, each velocity component of the RT0 discretization is piece-wise linear inside each element, and a constant velocity can occur only if the sum of the four fluxes is equal to zero. In this case the divergence is zero inside each element and mass continuity is both locally and globally conserved if the fluxes of two neighboring elements are always opposite one another in the common face.

In PS and CS1, the pressure gradient  $\mathbf{B}$  is kept constant. In CS2, its correction is computed as the solution of a conservative problem, where the velocity is opposite to the pressure gradient. We will show in the following that, at the end of each step PS, CS1 and CS2, the computed velocity is piece-wise constant in each element, but the fluxes  $Q_j^e$  in Equation (9) are opposite at the common face of two neighboring elements only at the end of CS2. Assuming a piece-wise linear pressure as initial condition, because the CS2 velocity correction is conservative with respect to the pressure correction, the pressure gradient remains piece-wise constant in all the steps. It can also be shown that the pressure correction computed in CS2 step by the MHFE is continuous along the element faces only in their circumcenters. Because pressure changes only in CS2 step, the same condition holds also for the pressure at the end of each time step. See in Figure 1 the pressure contour lines inside the face common to two neighboring tetrahedrons.

The presentation of the solution of each step in Equations (8) is restricted first to the hypothesis of a mesh with an Extended Delaunay Property (EDP), as defined in Aricò et al. (2011), and then generalized in section 4 to the case of totally irregular meshes. In an EDP mesh, the circumsphere of any tetrahedron does not include any other node inside it (Forsyth, 1991; Joe, 1986; Letniowski, 1992) and, with reference to Figure 2, the following conditions hold for two neighboring tetrahedrons  $e$  and  $ep$ ,

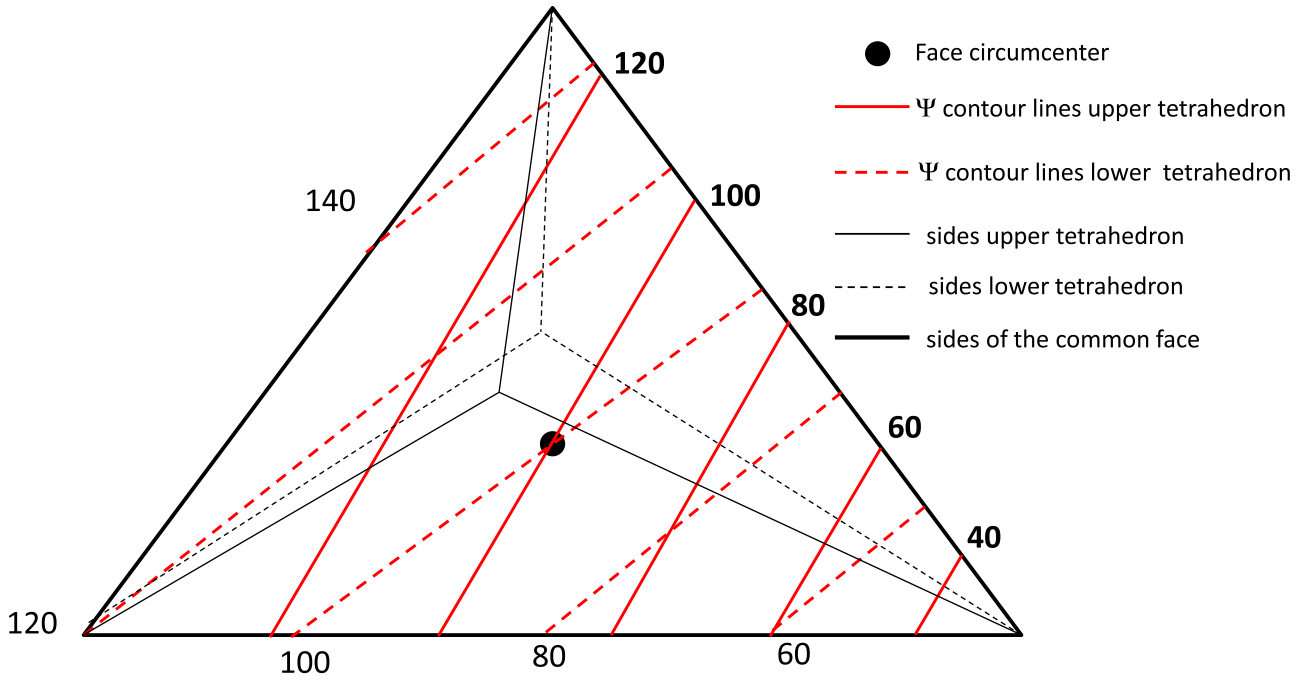
$$(\mathbf{v}_2 - \mathbf{v}_1) \cdot \mathbf{v}_1 < 0 \quad \text{if } |\mathbf{v}_1| > |\mathbf{v}_2| \quad (10a)$$

$$(\mathbf{v}_1 - \mathbf{v}_2) \cdot \mathbf{v}_2 < 0 \quad \text{if } |\mathbf{v}_2| > |\mathbf{v}_1| \quad (10b)$$

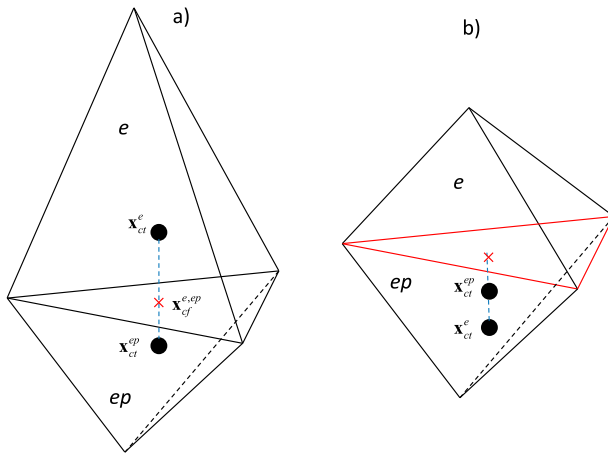
where vectors  $\mathbf{v}_1$  and  $\mathbf{v}_2$  are defined as

$$\mathbf{v}_1 = (\mathbf{x}_{ct}^e - \mathbf{x}_{cf}^{e,ep}), \quad \mathbf{v}_2 = (\mathbf{x}_{ct}^{ep} - \mathbf{x}_{cf}^{e,ep}) \quad (10c)$$

$\mathbf{x}_{ct}^{e(ep)}$  is the coordinate vector of the circumcenter of tetrahedron  $e$  ( $ep$ ),  $\mathbf{x}_{cf}^{e,ep}$  is the coordinate vector of the



**Figure 1.** RT0 kinematic pressure contour lines in the face common to two tetrahedrons.



**Figure 2.** (a) The two tetrahedrons  $e$  and  $ep$  satisfy the EDP. (b) The two tetrahedrons do not satisfy the EDP.

circumcenter of the face shared by tetrahedrons  $e$  and  $ep$ . Moreover, the following condition holds for the boundary elements  $e$  and the corresponding boundary face

$$\mathbf{v} \cdot \mathbf{n} < 0 \quad (11)$$

where  $\mathbf{n}$  is the unit vector normal to the boundary face and oriented in the outward direction,  $\mathbf{v}$  is defined as

$$\mathbf{v} = (\mathbf{x}_{ct}^e - \mathbf{x}_{cf}^e) \quad (12)$$

and  $\mathbf{x}_{cf}^e$  is the coordinate vector of the circumcenter of the boundary face. In Figure 2(a,b), tetrahedrons  $e$  and  $ep$  satisfy and do not satisfy the EDP, respectively.

We will prove in the next sections that in an EDP mesh the system matrix, associated with the solution of the steps (8b) and (8c), is an  $M$ -matrix (Letniowski, 1992), and this avoids nonphysical local extrema in the computed solution (Forsyth, 1991; Letniowski, 1992). The same matrix property will be saved in the more general case of non-Delaunay meshes by changing the control volumes of the momentum equations, as will be shown in section 4.

### 3. MAST-RT0 solution in the case of Delaunay meshes

#### 3.1. Prediction step

The step (8a) is solved by assuming, in each tetrahedral element, constant viscous and pressure gradient forces and by integrating convective inertial terms according to the MARCHING in Space and Time (MAST) procedure. MAST computes the solution of convective problems, with only one characteristic line passing through each  $(\mathbf{x}, t)$  point of the domain, in the framework of a fractional time step procedure.

In MAST the three discretized momentum equations are solved in each element, uncoupled from the other ones. To this end, at the beginning of each  $k^{\text{th}}$  time step, all the elements are marked with an index  $R_e^k \in \mathbf{R}^k$ , called 'rank'. The rank of an element  $e$  is an integer which is a unit greater than the ranks of all the neighboring  $ep$  elements with a common face crossed by a flux entering

element  $e$  from element  $ep$ . All the elements are initialized at the beginning of each new time iteration with rank zero, and vector  $\mathbf{R}^k$  is computed starting from the boundary elements with all interior fluxes oriented outward, which are set with rank 1. The rank is then computed for the elements that are neighbors to elements with rank greater than zero and have no interior faces crossed by fluxes oriented inward. The procedure is continued until all the elements of the computational domain have rank greater than zero. After computation of  $\mathbf{R}^k$ , all the elements are sorted according to increasing rank values and solved one after the other.

During the solution of element  $e$ , the integral of the velocity momentum is computed and the mean momentum fluxes are added as external forces to the neighboring elements  $ep$  with entering flux and higher rank. The MAST solution of the prediction problem along time step  $k$  can be viewed as a reduction of the computational domain carried out after the solution of each boundary element, through extraction of the latter. This reduction makes it possible to solve originally internal elements according to the information carried on by characteristic lines rooted at the boundary of internal elements that behave like domain boundaries. In the example of Figure 3, cell 1 is solved using the information at  $x_0$  carried on by the first characteristic line. After interpolation in time of the solution at the cell boundary in  $x_1$ , the second cell is solved using information carried on by the second characteristic line rooted between  $t^k$  and  $t^{k+1}$  at  $x_1$ . In this way, the sought after solution is not subject to the Courant restriction on the maximum size of the time step.

In Aricò et al. (2013b) it has been shown that the basic requirement for the application of the MAST algorithm is the existence of a continuous ‘anisotropic scalar potential’  $P$  of the flow field, such that the velocity can be computed as

$$\mathbf{u} = -\mathbf{K}_0 \nabla P \quad (13)$$

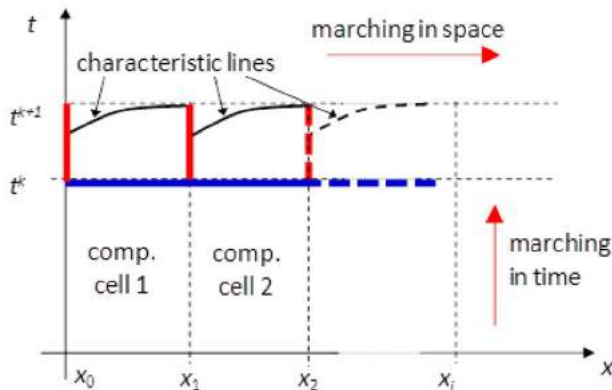


Figure 3. Sketch of MAST algorithm in the 1D case.

where  $\mathbf{K}_0$  is a  $(3 \times 3)$  positive-definite tensor. In the same paper it is shown that for any incompressible and viscous fluid this potential always exists and streamlines are always open but, due to numerical discretization, it is possible in the computed velocity field to get one or more loops where a single element without a flux entering from other elements of the same loop does not exist. In this condition the rank vector cannot be computed and, to apply MAST, we need to select a ‘cut’ face between two elements of the same loop with a small flux, which has to be assumed constant during the time step and equal to its initial value. See in Appendix 1 a very fast procedure, named Order, aimed to compute the  $\mathbf{R}^k$  vector, where the flux entering from the ‘cut’ face of a loop is treated as a known boundary flux. After application of the Order algorithm, an open source subroutine ‘QUICKSORT’ in the package KB07<sup>1</sup> can be used to order the elements according to their rank values.

After integration in space, the prediction step of Equation (7a) can be written, for any element  $e$ , as

$$\int_{W_e} \frac{\partial \mathbf{u}}{\partial t} dw + \int_{W_e} \mathbf{u} \cdot \nabla \mathbf{u} dw + \int_{W_e} \nabla \Psi^k dw - \nu \int_{W_e} \nabla_2 \mathbf{u}^{k-1/3} dw = 0 \quad e = 1, \dots, N_T \quad (14)$$

with the symbols specified as above. The first and second terms on the l.h.s. of Equation (14) represent the local and convective inertial terms, respectively, the third term is the force over the element due to the gradient of the kinematic pressure and the last term accounts for the effect of the viscous forces.

From now on we will call  $j$  (in the local element reference,  $j = 1, \dots, 4$ ) the face that tetrahedron  $e$  shares with its neighboring  $ep$ , and  $jp$  (in a local reference too) the face that  $ep$  shares with  $e$ . The symbols  $\sigma_j^e$  and  $\mathbf{n}_j$  represent the area and the unit outward orthogonal vector of the  $j$ th face of tetrahedron  $e$ , respectively.

Due to the cell sorting operation at the beginning of each time iteration, the solution of the momentum Equation (14) for element  $e$  with rank  $R_e^k$ , depends (a) on the incoming momentum fluxes from neighboring  $ep$  elements with  $R_{ep}^k < R_e^k$ , (b) on the kinematic pressure gradients and viscous terms, which are known from the solution of the previous time step, and (c) on the initial solution inside the element at the beginning of the new time step. For these reasons, the PDEs system in Equation (14) can be regarded as a small  $(3 \times 3)$  Ordinary Differential Equations (ODEs) system to be solved for the velocity components in element  $e$ .

Starting from a piecewise constant in space ( $P_0$ ) and divergence-free velocity distribution inside each tetrahedron  $e$ ,  $\mathbf{u}_e^k \in \mathbf{X}_e$  and  $\mathbf{u}_e^k \in P_{0,e}$ , we assume that at the



generic time  $\tau$  inside the time step ( $0 \leq \tau \leq \Delta t$ ) the velocity value is

$$\mathbf{u}_e(\tau) = \mathbf{u}_e^k + \Delta \mathbf{u}_e^f(\tau) \quad (15)$$

where  $\Delta \mathbf{u}_e^f(\tau) \in P_{0,e}$  and  $\Delta \mathbf{u}_e^f(0) = 0$ . Applying the Green lemma to the second integral on the l.h.s. of Equation (14), the equilibrium ODEs system for tetrahedron  $e$  can be written as

$$\begin{aligned} \frac{d(\Delta \mathbf{u}_e^f)}{d\tau} W_e + \sum_{j=1,4} \varphi_j^e \mathbf{M}_j^{e,out}(\tau) + \sum_{j=1,4} (1 - \varphi_j^e) \overline{\mathbf{M}}_j^{e,in} \\ + \mathbf{S}_{\Psi,e}^k + \mathbf{V}\mathbf{F}_e^{k-1/3} = 0 \quad e = 1, \dots, N_T \end{aligned} \quad (16)$$

where  $\mathbf{M}_j^{e,out}(\tau)$  is the leaving momentum flux from tetrahedron  $e$  to the neighboring tetrahedron  $ep$ ,  $\overline{\mathbf{M}}_j^{e,in}$  is the mean incoming momentum flux entering tetrahedron  $e$  crossing the  $j$ th face, computed along with the solution of the previous elements,  $\varphi_j^e = 1$  for faces shared by elements with higher rank or boundary faces with positive flux, otherwise  $\varphi_j^e = 0$ .  $\mathbf{S}_{\Psi,e}^k$  and  $\mathbf{V}\mathbf{F}_e^{k-1/3}$  are the sum of the kinematic pressure and the viscous forces over the four faces of tetrahedron  $e$ , respectively, computed in the previous time step as specified in sections 3.2 and 3.3. ODEs systems (16) are sequentially solved, one for each element, starting from the tetrahedrons with the smallest  $R_e^k$  value, and proceeding to the tetrahedrons with higher  $R_e^k$  values. We call this step MAST forward step (MAST-fs). To solve system (16) from time 0 to time  $\Delta t$ , we use an explicit Runge–Kutta (RK) code (Brankin et al., 1993). This code adopts an internal time sub-grid, selected within the interval  $[0 - \Delta t]$ , on which the approximate ODEs solution is computed. The position of the nodes of the grid is automatically selected by the RK code according to a local error estimation (Brankin et al., 1993).

Due to the change of the velocity vector during the ODEs system solution we could obtain, in intermediate times between 0 and  $\Delta t$ , momentum fluxes moving from the element  $e$  into other  $ep$  elements with a lower rank ( $R_{ep}^k < R_e^k$ ). To avoid this, we approximate the leaving momentum flux in the MAST-fs forward step as

$$\mathbf{M}_j^{e,out}(\tau) = \sigma_j^e \mathbf{u}_e(\tau) \max[0, \mathbf{u}_e(\tau) \cdot \mathbf{n}_j] \quad (17)$$

with the symbols specified above. To restore the force integral neglected in the integration of Equation (14) due to the limit of the momentum flux assigned in Equation (17), after the end of the forward step we again perform the solution of the 3-ODEs system in sequential way, assuming as initial  $\mathbf{u}_e$  value the one computed at the end of the MAST-fs forward step. In MAST backward solution (MAST-bs) we start from the tetrahedrons

with the highest  $R_e^k$  value and proceed to the tetrahedrons with smaller  $R_e^k$  values, saving only the inertial terms of Equation (14), that is:

$$\begin{aligned} \frac{d(\Delta \mathbf{u}_e^b)}{d\tau} W_e + \sum_{j=1,4} (1 - \varphi_j^e) \mathbf{M}_j^{e,out}(\tau) \\ + \sum_{j=1,4} \varphi_j^e \overline{\mathbf{M}}_j^{e,in} = 0 \end{aligned} \quad (18a)$$

$$\mathbf{u}_e(\tau) = \mathbf{u}_e^k + \Delta \mathbf{u}_e^f(\Delta t) + \Delta \mathbf{u}_e^b(\tau) \quad (18b)$$

where  $\Delta \mathbf{u}_e^b(\tau) \in P_{0,e}$ ,  $\Delta \mathbf{u}_e^b(0) = 0$ , and the momentum fluxes  $\mathbf{M}_j^{e,out}(\tau)$  are computed as for the forward step.

During the solution of the ODEs system (16) or (18), we compute the solution at  $n_G$  selected number of Gauss integration points chosen in the interval  $[0 - \Delta t]$ .

At the end of the solution of each ODEs system (16) or (18), in the MAST-fs and MAST-bs problems we compute the momentum fluxes coming into element  $e$  from the neighboring  $ep$  tetrahedrons as

$$\overline{\mathbf{M}}_j^{e,in} = -\overline{\mathbf{M}}_{jp}^{ep,out} \quad (19)$$

$$\begin{aligned} \overline{\mathbf{M}}_{jp}^{ep,out} = \sum_{l=1, n_G} \frac{\varphi_{jp}^{ep} \sigma_{jp}^{ep} \max[0, \mathbf{u}_{ep}(\tau_l) \cdot \mathbf{n}_j]}{\sum_{m=1,4} \{\varphi_m^{ep} \sigma_m^{ep} \max[0, \mathbf{u}_{ep}(\tau_l) \cdot \mathbf{n}_m]\}} \\ \times \mathbf{M}_{jp}^{ep,out}(\tau_l) w_l \text{ in MAST - fs} \end{aligned} \quad (20a)$$

$$\begin{aligned} \overline{\mathbf{M}}_{jp}^{ep,out} = \sum_{l=1, n_G} \frac{(1 - \varphi_{jp}^{ep}) \sigma_{jp}^{ep} \max[0, \mathbf{u}_{ep}(\tau_l) \cdot \mathbf{n}_{jp}]}{\sum_{m=1,4} \{(1 - \varphi_m^{ep}) \sigma_m^{ep} \max[0, \mathbf{u}_{ep}(\tau_l) \cdot \mathbf{n}_m]\}} \\ \times \mathbf{M}_{jp}^{ep,out}(\tau_l) w_l \text{ in MAST - bs} \end{aligned} \quad (20b)$$

where  $\tau_l$  and  $w_l$  are the time and the weight associated with the  $l^{\text{th}}$  Gauss point, with  $1 \leq l \leq n_G$ .

The final velocity  $\mathbf{u}^{k+1/3}$  is computed by Equation (18b) for  $\tau = \Delta t$ .  $\mathbf{u}^{k+1/3}$  is piecewise constant and divergence-free inside each tetrahedron ( $\mathbf{u}_e^{k+1/3} \in P_{0,e}$ ), but the fluxes crossing the same face of two neighboring elements will not be opposite for the two elements in the computed solution and this disrupts mass conservation at time level  $t^{k+1/3}$ .

In the MAST-fs problem, we compute the boundary momentum fluxes of tetrahedron  $e$ , different from zero, as

$$\begin{aligned} \mathbf{M}_j^{e,in}(\tau) = \sigma_j^e \mathbf{g}_{j,e}(\tau) (\mathbf{g}_{j,e}(\tau) \cdot \mathbf{n}_j) \\ \text{if } (\mathbf{g}_{j,e}(\tau) \cdot \mathbf{n}_j \leq 0 \text{ and } j \in \Gamma_u) \end{aligned} \quad (21a)$$

$$\begin{aligned} \mathbf{M}_j^{e,in}(\tau) = \sigma_j^e \mathbf{u}_e(\tau) \min(0, \mathbf{u}_e(\tau) \cdot \mathbf{n}_j) \\ \text{if } (\mathbf{u}_e^k \cdot \mathbf{n}_j \leq 0 \text{ and } j \in \Gamma_\sigma) \end{aligned} \quad (21b)$$

$$\mathbf{M}_j^{e,out}(\tau) = \sigma_j^e \mathbf{u}_e(\tau) \max(0, \mathbf{u}_e(\tau) \cdot \mathbf{n}_j) \text{ if } (\mathbf{u}_e^k \cdot \mathbf{n}_j \geq 0 \quad (21c)$$

where  $\mathbf{g}_{j,e}$  is the velocity vector assigned on the face  $j \in \Gamma_u$  of element  $e$ .

In the MAST-bs problem, we compute the same fluxes as

$$\mathbf{M}_j^{e,in}(\tau) = \sigma_j^e \mathbf{u}_e(\tau) \min(0, \mathbf{u}_e(\tau) \cdot \mathbf{n}_j) \text{ if } (\mathbf{u}_e^k \cdot \mathbf{n}_j > 0 \text{ and } j \in \Gamma_\sigma) \quad (22a)$$

$$\mathbf{M}_j^{e,out}(\tau) = \sigma_j^e \mathbf{u}_e(\tau) \max(0, \mathbf{u}_e(\tau) \cdot \mathbf{n}_j) \text{ if } \mathbf{u}_e^k \cdot \mathbf{n}_j < 0 \quad (22b)$$

### 3.1.1. Parallel solution of the MAST prediction step

The solution of the MAST PS in 1D problems is inherently serial and cannot be achieved with parallel computing. On the opposite, in 2D and 3D problems it could be possible to carry out simultaneously the solution of all the elements with the same rank using several CPU physical processors, just saving the average entering momentum fluxes computed for each element in both the forward and backward steps. See in Figure 4(a) the scheme of the MAST-fs solution, for a computer with five processors, of a single time step of a model with 13 elements, ordered in two groups of rank 1 (7 elements) and rank 2 (6 elements). The white circles represent the initial value of the variables and the black circles the final one.  $T_T$  is the computational time required to each processor for the solution of one element. Observe that, due to the need of solving all the elements with rank 1, before proceeding to the other ones, some processors have to solve 4 elements before moving to the next time step, where in any traditional marching in time method (MTM) only a maximum of 3 elements would be required, as shown in Figure 4(b). An additional time  $\epsilon$  is also required, for

each time step, for the rank computation and the element ordering in the MAST-fs (see Figure 4(a)). These last operations, as better specified in the tests run in section 5 with millions of elements, require a computational time two or three order of magnitude smaller than  $T_T$ .

For meshes with at least a few hundred thousand elements, the maximum number  $N_{r,k}$  of elements within a single rank  $R_e^k$  at time step  $k$  is usually much larger than the available physical processors in standard computers. For test 3 (in section 5.3.3), we estimate  $N_{r,k}$  and predict the corresponding computational time of parallelization of the MAST algorithm for different number of processors.

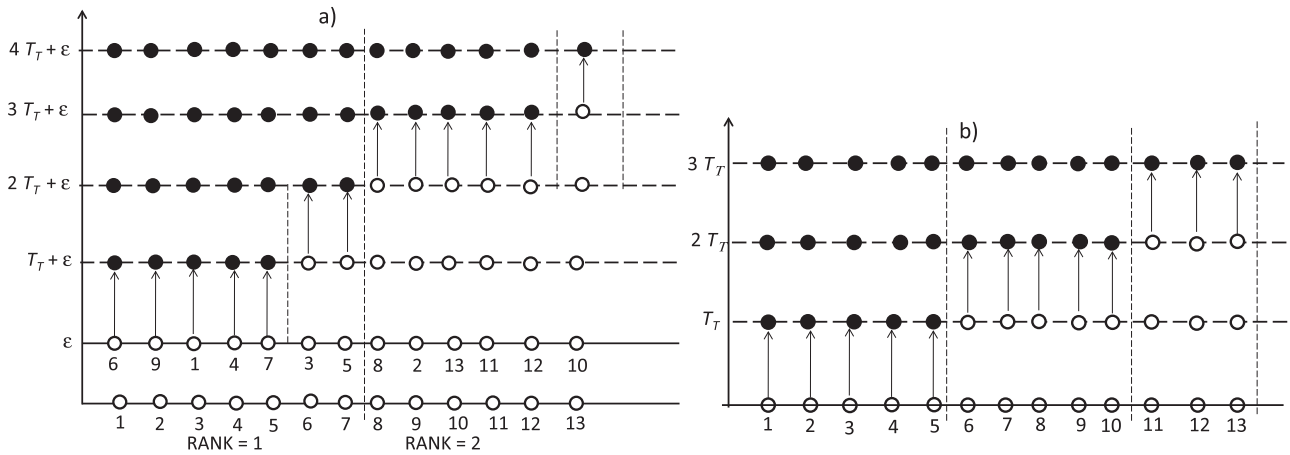
### 3.2. The CS1 correction step

By integrating Equation (8b) in space, we obtain the following system, to be solved for the three unknown velocity components  $\mathbf{u}_e^{k+2/3} \in P_{0,e}$ , from time level  $t^{k+1/3}$  to time level  $t^{k+2/3}$ ,

$$\begin{aligned} & \frac{\mathbf{u}_e^{k+2/3} - \mathbf{u}_e^{k+1/3}}{\Delta t} W_e - \int_{W_e} \nu \Delta_2 \mathbf{u}_e^{k+2/3} dw \\ & = - \int_{W_e} \nu \Delta_2 \mathbf{u}_e^{k-1/3} dw \quad e = 1, \dots, N_T \end{aligned} \quad (23)$$

Application of the Green lemma to the volume integral on the l.h.s. of Equation (23) leads to

$$\begin{aligned} -\nu \int_{W_e} \Delta_2 \mathbf{u}_e^{k+2/3} dw & = -\nu \sum_{j=1,4} \sigma_j^e \frac{\partial \mathbf{u}_e^{k+2/3}}{\partial n} \\ & = \nu \sum_{j=1,4} \sigma_j^e \frac{\mathbf{u}_e^{k+2/3} - \mathbf{u}_{ep}^{k+2/3}}{d_{e,ep}} \end{aligned} \quad (24)$$



**Figure 4.** Solution of a single time step. (a) Scheme of the parallel solution of the MAST-fs (or MAST-bs). (b) Scheme of the parallel solution of traditional MArching in Time method (MTM).

where  $\frac{\partial \mathbf{u}}{\partial n}$  is the derivative of the velocity vector along the direction orthogonal to face  $\sigma_j^e$ ,  $d_{e,ep}$  is the distance of the circumcenters of the two neighboring tetrahedrons  $e$  and  $ep$ , computed as specified in Equation (25), and the other symbols have already been defined. We compute the distance as

$$d_{e,ep} = (|\mathbf{v}_1 - \mathbf{v}_2|) \text{sign}$$

where  $\text{sign}$

$$= 1 \text{ if } \begin{cases} ((\mathbf{v}_2 - \mathbf{v}_1) \cdot \mathbf{v}_1 < 0 \text{ and } |\mathbf{v}_1| > |\mathbf{v}_2|) \\ \text{.or.} \\ ((\mathbf{v}_1 - \mathbf{v}_2) \cdot \mathbf{v}_2 < 0 \text{ and } |\mathbf{v}_2| > |\mathbf{v}_1|) \end{cases} \begin{matrix} \text{sign} = -1 \\ \\ \text{otherwise} \end{matrix} \quad (25)$$

where vectors  $\mathbf{v}_1$  and  $\mathbf{v}_2$  have been defined in Equation (10c). For each tetrahedron  $e$  we set

$$\Delta \widehat{\mathbf{u}}_e = \mathbf{u}_e^{k+2/3} - \mathbf{u}_e^{k+1/3} \text{ with } \Delta \widehat{\mathbf{u}}_e \in P_{0,e} \quad (26)$$

such that the r.h.s. of Equation (24) becomes,

$$\begin{aligned} & \nu \sigma_j^e \frac{\mathbf{u}_e^{k+2/3} - \mathbf{u}_{ep}^{k+2/3}}{d_{e,ep}} \\ &= \nu \sigma_j^e \left( \frac{\mathbf{u}_e^{k+1/3} - \mathbf{u}_{ep}^{k+1/3}}{d_{e,ep}} + \frac{\Delta \widehat{\mathbf{u}}_e - \Delta \widehat{\mathbf{u}}_{ep}}{d_{e,ep}} \right) \end{aligned} \quad (27)$$

On the r.h.s. of Equation (23) we assume

$$\begin{aligned} -\nu \int_{W_e} \Delta_2 \mathbf{u}_e^{k-1/3} dw &\simeq -\nu \int_{W_e} \Delta_2 \mathbf{u}_e^k dw \\ &= \nu \sum_{j=1,4} \sigma_j^e \frac{\mathbf{u}_e^k - \mathbf{u}_{ep}^k}{d_{e,ep}} \end{aligned} \quad (28)$$

We set

$$\Delta \widetilde{\mathbf{u}}_e = \mathbf{u}_e^{k+1/3} - \mathbf{u}_e^k \text{ with } \Delta \widetilde{\mathbf{u}}_e \in P_{0,e} \quad (29)$$

and merging Equation (29) with Equation (27) we get

$$\begin{aligned} & \nu \sigma_j^e \frac{\mathbf{u}_e^{k+1/3} - \mathbf{u}_{ep}^{k+1/3}}{d_{e,ep}} \\ &= \nu \sigma_j^e \left( \frac{\mathbf{u}_e^k - \mathbf{u}_{ep}^k}{d_{e,ep}} + \frac{\Delta \widetilde{\mathbf{u}}_e - \Delta \widetilde{\mathbf{u}}_{ep}}{d_{e,ep}} \right) \end{aligned} \quad (30)$$

According to Equations (26)-(30), system (30) can be rewritten as

$$\frac{\Delta \widehat{\mathbf{u}}_e}{\Delta t} W_e + \nu \sum_{j=1,4} \sigma_j^e \frac{\Delta \widehat{\mathbf{u}}_e - \Delta \widehat{\mathbf{u}}_{ep}}{d_{e,ep}}$$

$$= \nu \sum_{j=1,4} \sigma_j^e \frac{\Delta \widetilde{\mathbf{u}}_e - \Delta \widetilde{\mathbf{u}}_{ep}}{d_{e,ep}} \quad (31)$$

and solved for the components of the velocity correction  $\Delta \widehat{\mathbf{u}}_e$ .

The matrix of system (31) is sparse, symmetric and positive definite, with diagonal and off-diagonal coefficients  $M_{e,e}^{CS_1}$  and  $M_{e,ep}^{CS_1}$  equal to

$$M_{e,e}^{CS_1} = \frac{W_e}{\Delta t} + \nu \sum_{j=1,4} \sigma_j^e \frac{1}{d_{e,ep}}, M_{e,ep}^{CS_1} = -\nu \sigma_j^e \frac{1}{d_{e,ep}} \quad (32)$$

and the  $e$ -th element of the source term vector is the r.h.s. of Equation (31).

For the boundary face  $j \in \Gamma_u$  of tetrahedron  $e$ , the velocity at the circumcenter of the neighboring tetrahedron is replaced in the system of Equation (31) by the boundary velocity at the circumcenter of the boundary face and the distance  $d_{e,ep}$  with the distance  $d_j^e$  between the tetrahedron and the face circumcenter, defined as

$$d_j^e = -\mathbf{v} \cdot \mathbf{n} \quad (33)$$

where  $\mathbf{v}$  and  $\mathbf{n}$  have been already defined in Equations (11) and (12). If the boundary face belongs to  $\Gamma_u$  and condition  $\mathbf{g}_{j,e}(t^{k+1/3}) \cdot \mathbf{n}_j \leq 0$  holds, the velocity correction  $\Delta \widehat{\mathbf{u}}_j$  at the boundary face, from  $t^{k+1/3}$  to  $t^{k+2/3}$ , is

$$\begin{aligned} \Delta \widehat{\mathbf{u}}_j &= \mathbf{g}_{j,e}(t^{k+2/3}) - \mathbf{g}_{j,e}(t^{k+1/3}) \\ &= \mathbf{g}_{j,e}(t^k + \Delta t) - \mathbf{g}_{j,e}(t^k) \end{aligned} \quad (34a)$$

equal to zero along with the corresponding off-diagonal matrix coefficient in Equation (32), and the contribution to the source term becomes

$$\Delta \widetilde{\mathbf{u}}_j = \mathbf{g}_{j,e}(t^{k+1/3}) - \mathbf{g}_{j,e}(t^k) = \mathbf{g}_{j,e}(t^k + \Delta t) - \mathbf{g}_{j,e}(t^k) \quad (34b)$$

If the boundary face belongs to  $\Gamma_u$  and condition  $\mathbf{g}_{j,e}(t^{k+1/3}) \cdot \mathbf{n}_j > 0$  holds, the boundary velocity at time  $t^{k+1/3}$  is assumed to be the tetrahedral element velocity  $\mathbf{u}_e^{k+1/3}$ , that implies  $\Delta \widehat{\mathbf{u}}_j = \mathbf{g}_{j,e}(t^k + \Delta t) - \mathbf{u}_e^{k+1/3}$  and

$$\Delta \widetilde{\mathbf{u}}_j = \mathbf{u}_e^{k+1/3} - \mathbf{u}_e^k \quad (35)$$

If the boundary face belongs to  $\Gamma_\sigma$  or to  $\Gamma_m$ , for simplicity's sake we neglect the viscous tangent stress components, along with the terms in Equation (31) proportional to the viscosity.

If the EDP is satisfied (see section 2), due to Equations (11)-(12), the diagonal and off-diagonal coefficients in Equation (32) are positive and non-positive, respectively, such that the  $M$ -matrix property of the matrix is always guaranteed (Forsyth, 1991; Letniowski, 1992).

We adopt a preconditioned conjugate gradient method to solve system (31), using incomplete Cholesky factorization<sup>2</sup> and a compressed row storage (CRS) format, which makes it possible to save a lot of computational memory allocation (we store the main diagonal and the upper off-diagonal matrix coefficients). Since the matrix coefficients only depend on the geometrical variables, as well as the value of the kinematic viscosity, the matrix is only factorized once, before the time iterations loop starts, and this makes it possible to save a lot of computational effort.

After system (31) is solved, the velocity in each tetrahedron is updated according to Equation (26). At time level  $t^{k+2/3}$  for each tetrahedron  $e$  we compute the sum  $\mathbf{FV}_e^{k+2/3}$  of the viscous forces over its four faces as

$$\mathbf{FV}_e^{k+2/3} = \sum_{j=1,4} v \sigma_j^e \frac{\mathbf{u}_e^{k+2/3} - \mathbf{u}_{ep}^{k+2/3}}{d_{e,ep}} \quad (36)$$

and we neglect the change in it occurring along the next CS2 correction step.

At the end of the CS1 problem, the continuity of the fluxes crossing the same face of two neighboring elements  $e$  and  $ep$  is not yet restored, and, similarly to  $\mathbf{u}_e^{k+1/3}$  at the end of the PS problem (section 3.1),  $\mathbf{u}_e^{k+2/3} \in P_{0,e}$ , but mass conservation is not satisfied.

The spatial discretization of the derivative  $\frac{\partial \mathbf{u}}{\partial n}$  proposed in Equation (24) is similar to the one presented by Younes et al. (2004, 2006) for the 2D Mixed Hybrid Finite Element method lumped in the circumcenter of triangles. Unlike the formulation of the present work, Younes et al. (2004) proved that their corresponding 3D discretization exists only for regular tetrahedrons, and cannot be extended to a general 3D tetrahedral discretization.

### 3.3. The CS2 correction step

Substituting Equations (5b) in Equations (8c), we get

$$\frac{\mathbf{u}^{k+1} - \mathbf{u}^{k+2/3}}{\Delta t} + \nabla \cdot (\Psi^{k+1} - \Psi^k) = 0 \quad (37)$$

which has to be solved to restore the flux continuity disrupted in the prediction and in the first correction steps. To this end we set

$$\mathbf{u}^{k+1} - \mathbf{u}_{RT0}^{k+2/3} = \nabla \eta \Delta t \quad (38)$$

where  $\eta$  is an unknown function with the same dimensions as  $\psi$ , and  $\mathbf{u}_{RT0}^{k+2/3} \in \mathbf{X}_e$  is the RT0 velocity at time level  $t^{k+2/3}$  computed by Equation (9) in each tetrahedron  $e$  as function of the weight mean flux  $\overline{Fl}_{e,j}^{k+2/3}$

through face  $j$  of  $e$ , given by

$$\overline{Fl}_{j,e}^{k+2/3} = \frac{Fl_{j,e}^{k+2/3} W_{ep} - Fl_{jp,ep}^{k+2/3} W_e}{W_e + W_{ep}} \quad j, jp = 1, \dots, 4 \quad (39)$$

where  $Fl_{j,e}^{k+2/3}$  is the flux due to velocity  $\mathbf{u}_e^{k+2/3}$  crossing face  $j$  of  $e$ , and  $Fl_{jp,ep}^{k+2/3}$  is the flux due to velocity  $\mathbf{u}_{ep}^{k+2/3}$  crossing the same face  $jp$  of  $ep$ . From Equation (39), flux  $\overline{Fl}_{j,e}^{k+2/3}$  is opposite for the two neighboring tetrahedrons  $e$  and  $ep$  and the weight of each flux is inversely proportional to the volume of the corresponding element. Because the common face can be thought of as the basis of the two tetrahedrons, this is equivalent to assuming an inverse proportionality of the flux with respect to the corresponding height of the tetrahedron. According to Equation (9), we get

$$\mathbf{u}_{e,RT0}^{k+2/3} = \sum_{j=1,4} \omega_j^e \overline{Fl}_{j,e}^{k+2/3} \quad (40)$$

with the symbols already specified. Since the sum of the four fluxes  $Fl_{j,e}^{k+2/3}$   $j = 1, \dots, 4$  is not zero, mass conservation is not satisfied by the  $\mathbf{u}_{RT0}^{k+2/3}$  velocity field in tetrahedron  $e$ ,  $\nabla \cdot \mathbf{u}_{RT0}^{k+2/3} \neq 0$  and  $\mathbf{u}_{RT0}^{k+2/3} \in P_{1,e}$  (piecewise linear inside the tetrahedrons), as explained in section 2.2. By subtracting  $\mathbf{u}^k$  from both members in Equation (38), we get

$$\nabla \eta \Delta t + \mathbf{u}_{RT0}^{k+2/3} - \mathbf{u}^k = \mathbf{u}^{k+1} - \mathbf{u}^k \quad (41)$$

Taking the divergence of Equation (41), we get

$$\Delta t \nabla^2 \eta + \nabla \cdot (\mathbf{u}_{RT0}^{k+2/3} - \mathbf{u}^k) = \nabla \cdot (\mathbf{u}^{k+1} - \mathbf{u}^k) \quad (42)$$

Integration of Equation (42) and application of the Green lemma leads to

$$\Delta t \sum_{j=4} \sigma_j^e \frac{\eta_e - \eta_{ep}}{d_{e,ep}} + \sum_{j=4} (\overline{Fl}_{j,e}^{k+2/3} - Fl_{j,e}^k) = Fl_e \quad (43)$$

where the first term is the flux of the vector  $\mathbf{u}^{k+1} - \mathbf{u}_{RT0}^{k+2/3}$ ,  $\overline{Fl}_{j,e}^{k+2/3}$  is defined in Equation (39),  $Fl_{j,e}^k$  is the flux crossing face  $j$  of element  $e$  due to velocity  $\mathbf{u}^k \in \mathbf{X}_e$  and  $Fl_e$  is the total flux of the vector  $(\mathbf{u}^{k+1} - \mathbf{u}^k)$  crossing the surface of element  $e$ . For the gradient of  $\eta$  we adopt the spatial discretization already applied in section 3.2 for the velocity gradient (see Equation (24)).

To compute  $\eta$ , we impose the condition that both  $\mathbf{u}^{k+1}$  and  $\mathbf{u}^k$  are divergence-free, with  $\mathbf{u}^{k+1} \in \mathbf{X}_e$  and  $\mathbf{u}^{k+1} \in P_{0,e}$ , which implies that in Equation (43) the flux  $Fl_e$  of the velocity  $(\mathbf{u}^{k+1} - \mathbf{u}^k)$  crossing the total surface of the

tetrahedron is zero (see in Figure 5 a 1D sketch of velocity vectors inside tetrahedron  $e$ ). The resulting system is

$$\Delta t \sum_{j=4} \sigma_j^e \frac{\eta_e - \eta_{ep}}{d_{e,ep}} + \sum_{j=4} (\overline{Fl}_{j,e}^{k+2/3} - Fl_{j,e}^k) = 0 \quad (44)$$

Equations (44) form a well-conditioned linear system to be solved for the  $\eta$  unknowns. The matrix of the system is sparse, symmetric and positive-definite. Diagonal and off-diagonal coefficients  $M_{e,e}^{CS_2}$  and  $M_{e,ep}^{CS_2}$  are, respectively,

$$M_{e,e}^{CS_2} = \sum_{j=1,4} \sigma_j^e \frac{\Delta t}{d_{e,ep}}, M_{e,ep}^{CS_2} = -\sigma_j^{ep} \frac{\Delta t}{d_{e,ep}} \quad (45a)$$

and the same  $M$ -property of the coefficients of the matrix of the system of CS1 holds (see section 3.2). The  $e$ -th coefficient of the source term vector is

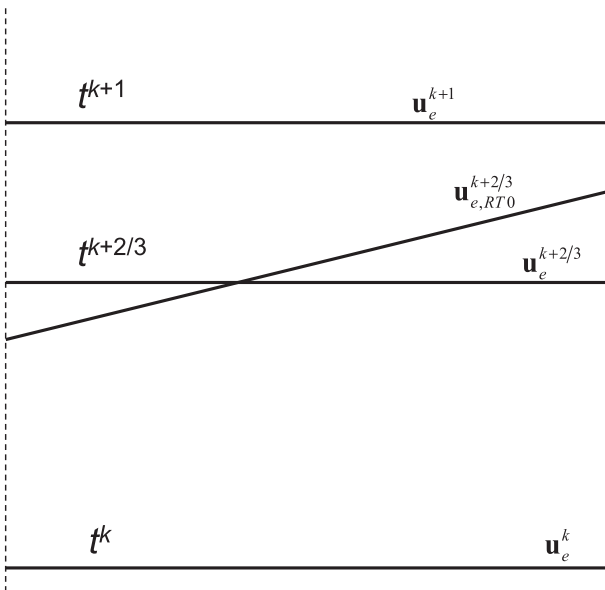
$$St_e^{CS_2} = -\sum_{j=4} (\overline{Fl}_{j,e}^{k+2/3} - Fl_{j,e}^k) \quad (45b)$$

The solution of system (44)-(45) is performed in the same way as for system (31)-(32). In the present case too, matrix coefficients only depend on the geometrical variables and time step size, so that the system matrix is factorized only once before the time iteration loop starts, saving a lot of computational time.

We call

$$Fl_{j,e}^\eta = \Delta t \sigma_j^e \frac{\eta_e - \eta_{ep}}{d_{e,ep}} \quad (46)$$

the flux crossing face  $j$  of  $e$  due to the gradient of  $\eta$ . This flux is continuous for the two neighboring elements  $e$  and



**Figure 5.** 1D sketch of velocity vectors inside tetrahedron  $e$ .

$ep$ , with  $Fl_{j,e}^\eta = -Fl_{jp,ep}^\eta$ . For the tetrahedron  $e$  we compute the final velocity at time level  $t^{k+1}$  from the fluxes on the l.h.s. of Equations (44), coupled with Equation (46), as

$$\begin{aligned} \mathbf{u}_e^{k+1} &= \sum_{j=1,4} \omega_j^e (\overline{Fl}_{j,e}^{k+2/3} + Fl_{j,e}^\eta - Fl_{j,e}^k) + \mathbf{u}_e^k \\ &= \sum_{j=1,4} \omega_j^e (\overline{Fl}_{j,e}^{k+2/3} + Fl_{j,e}^\eta) \end{aligned} \quad (47)$$

Because the fluxes in the brackets of Equation (47) are continuous, mass conservation is satisfied along all the faces of each tetrahedron and inside each tetrahedron.

The pressure gradient at time level  $t^{k+1}$  is finally computed from Equation (37) as

$$\nabla \Psi_e^{k+1} = \frac{\mathbf{u}_e^{k+2/3} - \mathbf{u}_e^{k+1}}{\Delta t} + \nabla \Psi_e^k \quad (48)$$

Observe, from Equation (38) and from the definition of  $\mathbf{u}_{e,RT0}^{k+2/3} \in P_{1,e}$  given in Equation (40), that the gradient  $\nabla \eta$  and  $\eta$  have respectively a linear and a quadratic variation inside each element, while the pressure gradient and the pressure correction in Equation (48) have respectively a linear and a constant variation. From Equations (48) and (38) we get

$$\nabla \Psi_e^{k+1} - \nabla \Psi_e^k = \frac{\mathbf{u}_e^{k+2/3} - \mathbf{u}_{RT0}^{k+2/3}}{\Delta t} - \nabla \eta \quad (49)$$

Equation (49) says that, inside the computational domain, the gradient of the function  $\eta$  is different from the gradient of the kinematic pressure correction  $\Psi_e^{k+1} - \Psi_e^k$ . Integration of the divergence of both members of Equation (49) would allow us to compute the pressure at the element circumcenters. On the other hand, the kinematic pressure does not need to be known inside the domain in order to proceed to the solution of the next time steps. In section 5.2 we will show how to estimate the kinematic pressure at the computational nodes only at a given number of simulation times. If the velocity is known at the circumcenter of a boundary face belonging to  $\Gamma_u$ , we can set  $\overline{Fl}_{j,e}^{k+2/3}$  equal to the corresponding flux, include it in the r.h.s. of Equation (44) and set the corresponding flux  $Fl_{j,e}^\eta$  equal to zero.

Observe that the tangent components of the velocity computed by Equation (47) are not the same as those used along the  $\Gamma_u$  boundary surface to compute the viscous boundary forces in the previous CS1 correction step. This implies a small difference between the computed boundary face and boundary element velocities.

At the circumcenter of the faces belonging to  $\Gamma_\sigma$ , where the hydrostatic stress boundary condition is set,



the following Dirichlet type condition is finally assigned to the corresponding equations

$$\eta_c = \Psi_c^{k+1} - \Psi_c^k \quad (50)$$

and the boundary flux  $\overline{Fl}_{j,e}^{k+2/3} - Fl_{j,e}^k$  is computed 'a posteriori' from the corresponding equation of system (44). In all the other boundary faces the corrective flux  $Fl_{j,e}^\eta$  is set equal to zero.

After computation of  $\nabla \Psi_e^{k+1}$ , the sum  $\mathbf{S}_{\Psi,e}^{k+1}$  of the kinematic pressure forces over the four faces of tetrahedron  $e$ , can be computed applying the Green lemma as

$$\mathbf{S}_{\Psi,e}^{k+1} = \int_{\Omega_e} \mathbf{n} \Psi_e^{k+1} d\Omega = \int_{W_e} \nabla \Psi_e^{k+1} dW = \nabla \Psi_e^{k+1} W_e \quad (51)$$

Vector  $\mathbf{S}_{\Psi,e}^{k+1}$  is assumed equal to  $\mathbf{S}_{\Psi,e}^k$  in the MAST PS in the next time iteration (see Equation (16) in section 3.1).

Unfortunately, even if in the 2D case it is always possible to get a mesh satisfying the EDP as defined in Equation (11), also for very irregular domain geometries (Aricò et al., 2011; Aricò & Tucciarelli, 2013; Forsyth, 1991, and cited references), in 3D space it is almost impossible to obtain a mesh satisfying the EDP and a 3D Delaunay mesh always has very irregular elements inside it (e.g. slivers, caps, skinny tetrahedrons, ...), which could affect the stability of the numerical solution (Joe, 1986). This implies the need to extend the methodology presented in all section 3 to the more general case of non-Delaunay meshes.

### 3.4. The MAST-RT0 pseudo code in the case of Delaunay meshes

- (1) Compute model constants, including CS1 and CS2 matrix coefficients by means of Equations (32) and (45a). Perform matrix factorization and set  $k = 1$
- (2) Given  $\mathbf{u}^k$  and  $\nabla \Psi^k$  at time  $t^k$ , apply the MAST prediction step (PS)
  - Compute velocity variation  $\Delta \mathbf{u}_e^f(\Delta t)$  and  $\Delta \mathbf{u}_e^b(\Delta t)$  by solving Equations (16) and (18).
  - Update  $\mathbf{u}_e^{k+1/3} = \mathbf{u}_e(\Delta t)$  for each tetrahedron  $e$  by means of Equation (18b)
- (3) Apply the 1st corrective step (CS1)
  - Solve system (31) for the  $\Delta \widehat{\mathbf{u}}_e$  unknowns
  - update velocity  $\mathbf{u}_e^{k+2/3}$  for each tetrahedron  $e$  by means of Equation (26)
- (4) Update the viscous forces  $\mathbf{FV}_e^{k+2/3}$  for each tetrahedron  $e$  by means of Equation (36)
- (5) Compute fluxes  $\overline{Fl}_{e,j}^{k+2/3}(\mathbf{u}^{k+2/3})$  for each face  $j$  ( $j = 1, \dots, 4$ ) of tetrahedron  $e$  by means of Equation (39)

- (6) Apply the 2nd corrective step (CS2)
  - Compute  $\eta_e$  for each tetrahedron  $e$  by solving system (44)
  - Compute fluxes  $Fl_{j,e}^\eta$  for each tetrahedron  $e$  by means of Equation (46)
  - Update the final velocity  $\mathbf{u}_e^{k+1}$  for each tetrahedron  $e$  according to Equation (47)
- (7) Update the gradient of the kinematic pressure  $\nabla \Psi^{k+1}$  according to Equation (48)
- (8) Compute the kinematic pressure forces  $\mathbf{S}_{\Psi,e}^{k+1}$  for each tetrahedron  $e$  by means of Equation (51)
- (9) Update  $k$  with  $k+1$  and go back to point two for the next time step

## 4. The numerical procedure for non-Delaunay meshes

### 4.1. Tetrahedron clusters

The discretization of the second-order derivative terms in CS1 and CS2 problems (i.e. Equations (31) and (44)) along a face of the computational mesh shared by two tetrahedrons with negative distance, as defined in Equations (25) and (33), can lead to positive off-diagonal matrix coefficients  $M_{e,ep}^{CS1}$  and  $M_{e,ep}^{CS2}$  and a negative contribution to the corresponding diagonal coefficients  $M_{e,e}^{CS1}$  and  $M_{e,e}^{CS2}$  (see Equations (31)-(32) and (44)-(45)), so that the diagonal dominance and the  $M$ -property of the matrix system could be lost (Forsyth, 1991; Joe, 1986; Letniowski, 1992). Moreover, unphysical numerical solutions could arise, corresponding to poorly oriented viscous forces and pressure gradients. To avoid this problem, we propose the procedure described in the present section to handle the PS, CS1 and CS2 steps in the case of non-Delaunay meshes.

Let us consider irregular a face shared by two tetrahedrons with a negative distance  $d_{e,ep}$  between the corresponding circumcenters, as defined in Equation (25). If one or more irregular faces are present in the mesh, the EDP condition is no longer satisfied (see section 2.2). We group all the tetrahedrons in clusters. A cluster is to be seen as a small non-empty group of neighboring tetrahedrons not sharing any irregular face with other clusters. For example, the two tetrahedrons in Figure 2(b) form a cluster. Each tetrahedron belongs to a single cluster. A single tetrahedron  $e$  forms a cluster by itself if it has no irregular faces. In the cluster, we distinguish the external faces, shared by two tetrahedrons of different clusters, and the other internal ones. According to the previous assumptions, all external faces are regular, and in a cluster composed of a single tetrahedron we do not have internal faces.

Let  $N_C$  be the number of clusters, with  $N_C \leq N_T$ . The general strategy is to write, instead of the dynamic equilibrium of each single tetrahedron, the dynamic equilibrium of each cluster as a function of a single velocity variation and finally to correct all the tetrahedron velocities inside the clusters, after the CS2 correction step, in order to guarantee flux continuity through all the faces.

From now on, indices  $m$  and  $mp$  refer to two neighboring clusters,  $N_{T,m}$  is the number of tetrahedrons  $e$  belonging to the  $m$ -th cluster,  $N_{f,m}^{ext}$  and  $N_{f,m}^{int}$  are the number of external and internal faces of the cluster  $m$ ,  $l$  and  $r$  are the local counters of the external and internal faces of the cluster, respectively ( $l = 1, \dots, N_{f,m}^{ext}$ ,  $r = 1, \dots, N_{f,m}^{int}$ ), and  $W_m$  is the volume of the cluster,  $W_m = \sum_{e=1, N_{T,m}} W_e$  (see also Figure 6).

#### 4.2. The PS and CS1 problems for non-Delaunay meshes

Solution of the MAST prediction step, as explained in section 3.1, is not affected by the existence of irregular faces. On the other hand, after solution of all tetrahedrons, we need to evaluate, for each cluster, a single velocity variation corresponding to the cluster dynamic equilibrium. Call  $\mathbf{u}_e^{k+1/3} \in P_{0,e}$  the predicted tetrahedron velocity computed at time  $t^{k+1/3}$  as described in section 3.1 for  $\mathbf{u}_e^{k+1/3}$  in the case of Delaunay meshes, and  $\Delta \tilde{\mathbf{u}}_m$  the unknown velocity variation between time levels  $t^{k+1/3}$  and  $t^k$  to be assigned to cluster  $m$ .

For each tetrahedron  $e$  of cluster  $m$ , we can write the correction velocity  $\mathbf{u}_e^{k+1/3} - \mathbf{u}_e^k \in P_{0,e}$  by summing the time integrals of the ODEs system (16) and (18), to get

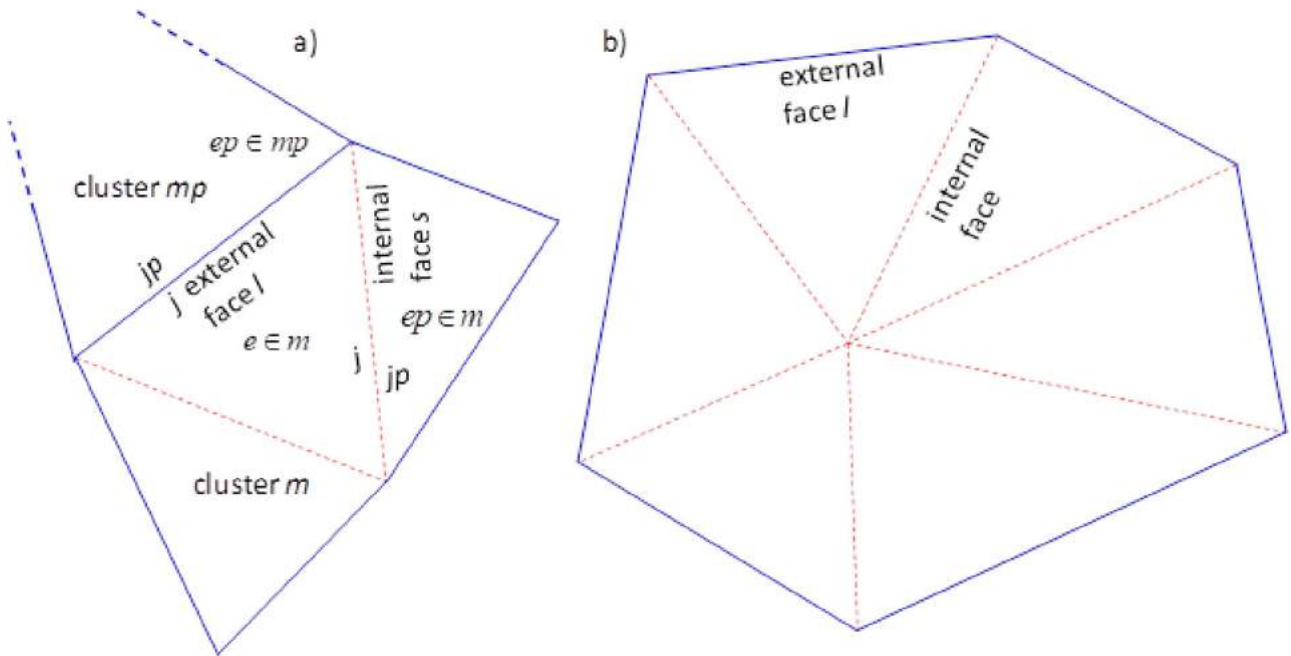
$$\begin{aligned} & (\mathbf{u}_e^{k+1/3} - \mathbf{u}_e^k) W_e \\ &= - \left( \sum_{j=1,4} \bar{\mathbf{M}}_j^{e,out} + \sum_{j=1,4} \bar{\mathbf{M}}_j^{e,in} + \mathbf{S}_{\Psi,e}^k + \mathbf{FV}_e^{k-1/3} \right) \\ & \quad \times \Delta t \end{aligned} \quad (52)$$

where the symbols have been defined in section 3.1.  $\bar{\mathbf{M}}_j^{e,out}$  has been computed in the forward step if  $R_e^k < R_{ep}^k$  and in the backward step otherwise,  $\bar{\mathbf{M}}_j^{e,in}$  has been computed in the forward step if  $R_e^k > R_{ep}^k$  and in the backward step otherwise. Observe that, in all internal faces of the cluster, the following condition holds

$$\bar{\mathbf{M}}_j^{e,out} = -\bar{\mathbf{M}}_{jp}^{ep,in} \quad (53)$$

This implies that, summing the Equation (52) of all the tetrahedrons of the same cluster, we get

$$\begin{aligned} & \sum_e (\mathbf{u}_e^{k+1/3} - \mathbf{u}_e^k) W_e \\ &= - \sum_e \left( \sum_{j=1,4} \bar{\mathbf{M}}_j^{e,out} \right. \\ & \quad \left. + \sum_{j=1,4} \bar{\mathbf{M}}_j^{e,in} + \mathbf{S}_{\Psi,e}^k + \mathbf{FV}_e^{k-1/3} \right) \Delta t \end{aligned}$$



**Figure 6.** (a) 2D sketch of case  $N_{f,m}^{int} = N_{T,m} - 1$ . (b) 2D sketch case of  $N_{f,m}^{int} = N_{T,m}$ . Blue solid lines are traces of the external faces of the cluster, and red dashed lines are traces of the internal faces of the cluster.

$$e = 1, \dots, N_{T,m} \quad (54)$$

where all the momentum fluxes belonging to internal faces as well as the corresponding viscous forces, sum zero and can be deleted. Equation (54) can be seen as the equilibrium equation of the cluster  $m$ , which can be approximated by setting

$$\Delta \tilde{\mathbf{u}}_m = \frac{\sum_e (\mathbf{u}_e^{k+1/3} - \mathbf{u}_e^k) W_e}{W_m} \quad \text{with} \quad \Delta \tilde{\mathbf{u}}_m \in P_{0,m} \quad (55)$$

and by replacing in each tetrahedron the previously computed  $\mathbf{u}_e^{k+1/3}$  velocity with

$$\mathbf{u}_e^{k+1/3} = \mathbf{u}_e^k + \Delta \tilde{\mathbf{u}}_m \quad e = 1, \dots, N_{T,m} \quad (56)$$

In the CS1 step, solution of Equation (31) in not EDP meshes is hindered by the negative distances  $d_{e,ep}$  holding between the two circumcenters of tetrahedrons sharing an irregular face. To circumvent the problem, we write the viscous forces equilibrium of the cluster, where the external forces act always on regular external faces. Following the same procedure applied in section 3.2, integrating in space and time Equation (7b) and applying the Green lemma, we get the following system,

$$\begin{aligned} & \sum_{e=1, N_{T,m}} \frac{\mathbf{u}_e^{k+2/3} - \mathbf{u}_e^{k+1/3}}{\Delta t} W_e \\ & + \nu \sum_{l=1, N_{f,m}^{ext}} \sigma_l^m \frac{\mathbf{u}_e^{k+2/3} - \mathbf{u}_{ep}^{k+2/3}}{d_{e,ep}} \\ & = \nu \sum_{l=1, N_{f,m}^{ext}} \sigma_l^m \frac{\mathbf{u}_e^{k+1/3} - \mathbf{u}_{ep}^{k+1/3}}{d_{e,ep}} \quad m = 1, \dots, N_C \end{aligned} \quad (57)$$

where  $ep$  is the tetrahedron of cluster  $r$ , sharing with tetrahedron  $e$  the  $l$ -th external face of cluster  $m$ , with area  $\sigma_l^m$ , distance  $d_{e,ep}$  is defined in Equation (25), and the other symbols have been previously specified.

We set, for the tetrahedrons of cluster  $m$

$$\begin{aligned} \mathbf{u}_e^{k+2/3} &= \mathbf{u}_e^{k+1/3} + \Delta \hat{\mathbf{u}}_m \\ e &= 1, \dots, N_{T,m} \quad \text{with} \quad \Delta \hat{\mathbf{u}}_m \in P_{0,m} \end{aligned} \quad (58)$$

and, substituting Equations (56) and (58) in Equation (57), we obtain 3 systems, one for each of the unknown components of  $\Delta \hat{\mathbf{u}}_m$ ,

$$\frac{\Delta \hat{\mathbf{u}}_m}{\Delta t} W_m + \nu \sum_{l=1, N_{f,m}^{ext}} \sigma_l^m \frac{\Delta \hat{\mathbf{u}}_m - \Delta \hat{\mathbf{u}}_r}{d_{e,ep}}$$

$$= \nu \sum_{l=1, N_{f,m}^{ext}} \sigma_l^m \frac{\Delta \tilde{\mathbf{u}}_m - \Delta \tilde{\mathbf{u}}_r}{d_{e,ep}} \quad m = 1, \dots, N_C \quad (59)$$

Matrix of system (59) has the same properties of matrix of system (31), is symmetric and positive definite, its diagonal and off-diagonal coefficients  $M_{e,e}^{CS1}$  and  $M_{e,ep}^{CS1}$  are, respectively,

$$M_{m,m}^{CS1} = \frac{W_m}{\Delta t} + \nu \sum_{l=1, N_{f,m}^{ext}} \sigma_l^m \frac{1}{d_{e,ep}}, M_{m,r}^{CS1} = -\nu \sigma_l^m \frac{1}{d_{e,ep}} \quad (60)$$

and the  $m$ -th coefficient of the source term vector is the r.h.s. of Equation (57). Diagonal and off-diagonal matrix coefficients in Equation (60) are positive and non-positive, respectively, since all the distances  $d_{e,ep}$  are positive.

We deal with the BCs of the CS1 problem as described in section 3.2. Observe that the previously described procedure fails to guarantee in the clusters a positive distance from a boundary element circumcenter and the circumcenter of its boundary face. On the other hand, to guarantee a positive distance from the boundary we can simply avoid any internal node with a distance from the circumcenter of each triangular boundary face smaller than the radius of the circle passing through the three nodes of the same boundary triangle. This can be easily done with a methodology that will be shown in section 5.1.

After solution of system (59), the velocity in the tetrahedrons  $e$  of each cluster  $m$  is updated according to Equation (58). At time level  $t^{k+2/3}$  we need to compute for each tetrahedron  $e$  the sum  $\mathbf{FV}_e^{k+2/3}$  of the viscous forces over its four faces, including the irregular ones, needed for the next computational time step. To do that we assume, coherently with the approximation of Equation (58), the local inertia per unit volume in each tetrahedron, to get

$$\begin{aligned} \mathbf{FV}_e^{k+2/3} &= -\frac{(\mathbf{u}_e^{k+1/3} - \mathbf{u}_e^k)}{\Delta t} W_e - \mathbf{S}_{\Psi,e}^k \\ &\quad - \sum_{j=1,4} \bar{\mathbf{M}}_j^{e,out} + \sum_{j=1,4} \bar{\mathbf{M}}_j^{e,in} \end{aligned} \quad (61)$$

where the symbols of the momentum fluxes are the same used for the r.h.s of Equation (52).

As it happens for the CS1 problem in the EDP meshes (section 3.2), at time level  $t^{k+2/3}$ , the continuity of the fluxes crossing the same face of two neighboring elements  $e$  and  $ep$  belonging to two different clusters, is not yet restored,  $\mathbf{u}_e^{k+2/3} \in P_{0,e}$ , and mass conservation is not satisfied.

### 4.3. The CS2 problem for non-Delaunay meshes

In non-Delaunay meshes the solution of the CS2 problem is split into two sub-steps. In the first sub-step we apply the procedure described in section 3.3, including in the mass balance the fluxes crossing the external faces of the cluster instead of the fluxes of the four faces of the single tetrahedron (as in Equations (43) and (44)), and assuming a single  $\eta$  value for the circumcenters of all the tetrahedrons inside the cluster. System (44) becomes

$$\begin{aligned} \Delta t \sum_{l=1, N_{f,m}^{ext}} \sigma_l^e \frac{\eta_m - \eta_{mp}}{d_{e,ep}} + \sum_{l=1, N_{f,m}^{ext}} (\overline{F}_{l,e}^{k+2/3} - F_{l,e}^k) \\ = 0 \quad m = 1, \dots, N_C \end{aligned} \quad (62)$$

with the symbols already specified and distance  $d_{e,ep}$  is always positive according to section 4.1. We call  $\overline{F}_{l,e}^{k+2/3}$  the flux crossing face  $l$  as defined in Equation (39) over face  $j$  ( $j = 1, \dots, 4$ ) of  $e$ , and  $F_{l,e}^k$  the flux due to velocity  $\mathbf{u}_e^k \in \mathbf{X}_e$  crossing the same face.

The diagonal and off-diagonal matrix coefficients for system (62a)  $M_{m,m}^{CS_2}$  and  $M_{m,mp}^{CS_2}$  are, respectively,

$$M_{m,m}^{CS_2} = \sum_{l=1, N_{f,m}^{ext}} \sigma_l^e \frac{\Delta t}{d_{e,ep}}, M_{m,mp}^{CS_2} = -\sigma_l^m \frac{\Delta t}{d_{e,ep}} \quad (63a)$$

and the same  $M$ -property of the coefficients of the matrix of the system of CS1 holds (see section 3.2). The  $m$ -th coefficient of the source term vector is

$$St_m^{CS_2} = - \sum_{l=1, N_{f,m}^{ext}} (\overline{F}_{l,e}^{k+2/3} - F_{l,e}^k) \quad (63b)$$

The solution of the system formed by Equations (62)–(63) guarantees flux continuity on the external faces and global mass conservation inside the cluster, but because the velocities  $\mathbf{u}_e^{k+1}$  inside each cluster do not generally belong to a single RT0 space, it does not guarantee zero divergence condition inside the cluster, unless the cluster includes only one tetrahedron.

For clusters composed of more than one tetrahedron, we apply the following procedure. We change, in Equation (47), the fluxes  $(\overline{F}_{j,e}^{k+2/3} + F_{j,e}^\eta)$  crossing the internal faces of the cluster, with the unknown flux  $F_{j,e}^{int}$ , to get

$$\mathbf{u}_e^{k+1} = \sum_{j=1,4} \omega_j^e [\tilde{\varphi}_j F_{j,e}^{int} + (1 - \tilde{\varphi}_j)(F_{j,e}^\eta + \overline{F}_{j,e}^{k+2/3})] \quad (64)$$

where  $F_{j,e}^{int}$  is the flux crossing face  $j$  of element  $e$ , which is internal to the cluster, and  $\tilde{\varphi}_j$  is 1 if face  $j$  is internal to

the cluster, and 0 if it is external. We also assume

$$F_{j,e}^{int} = -F_{jp,ep}^{int} \quad (65)$$

and we look for the optimal set of internal fluxes that: (1) guarantees the condition  $\mathbf{u}_e^{k+1} \in P_{0,e}$  for all the elements of the cluster and, as a consequence of constraint (65), mass conservation too, (2) minimizes the kinetic energy inside the cluster.

To get the required  $F_{j,e}^{int}$  fluxes, in the second sub-step of the CS2 problem, for each cluster we solve the following linearly constrained quadratic minimization problem,

$$\begin{aligned} \text{Minimize } \mathfrak{S} \\ = \frac{1}{2} \sum_{e=1, N_{T,m}} \left\{ \left[ \sum_{j=1,4} \omega_j^e (\tilde{\varphi}_j F_{j,e}^{int} \right. \right. \\ \left. \left. + (1 - \tilde{\varphi}_j)(F_{j,e}^\eta + \overline{F}_{j,e}^{k+2/3}) \right)^2 W_e \right\} \end{aligned} \quad (66)$$

subject to Equation (65) and

$$\begin{aligned} \sum_{j=1,4} (\tilde{\varphi}_j F_{j,e}^{int} + (1 - \tilde{\varphi}_j)(F_{j,e}^\eta + \overline{F}_{j,e}^{k+2/3})) \\ = 0e = 1, \dots, N_{T,m-1} \end{aligned} \quad (67)$$

where functional  $\mathfrak{S}$  is the total kinetic energy of the cluster, given by the sum of the kinetic energy of the single tetrahedrons inside it, and the term in the square brackets in Equation (66) is the velocity  $\mathbf{u}_e^{k+1}$ , according to Equation (64). The last mass conservation Equation (67) in tetrahedron  $N_{T,m}$  has been skipped because solving Equation (62) guarantees global mass conservation inside the cluster and this implies that any Equation (67) can be written as a linear combination of all the other ones. A very efficient way to solve problem (66)–(67) is to compute the Lagrangian multipliers, along with the required unknowns, as the solution of the following unconstrained quadratic minimization problem,

$$\begin{aligned} \text{Minimize}_{F_l, \lambda} \mathfrak{S} \\ = \frac{1}{2} \sum_{e=1, N_{T,m}} \left\{ \sum_{j=1,4} \left[ \omega_j^e \left( \tilde{\varphi}_j \sum_{s=1, N_{f,m}^{int}} \delta_{e,j}^s F_{l,s} \right. \right. \right. \\ \left. \left. \left. + (1 - \tilde{\varphi}_j)(F_{j,e}^\eta + \overline{F}_{j,e}^{k+2/3}) \right)^2 W_e \right] \right\} \\ + \sum_{e=1, N_{T,m-1}} \left\{ \lambda_e \sum_{j=1,4} \left( \tilde{\varphi}_j \sum_{s=1, N_{f,m}^{int}} \delta_{e,j}^s F_{l,s} \right) \right\} \end{aligned}$$

$$\left. + (1 - \tilde{\varphi}_j)(F_{j,e}^{\eta} + \overline{F}_{j,e}^{k+2/3}) \right\} \quad (68a)$$

where  $FI_s$  is the  $s^{\text{th}}$  internal flux, occurring between tetrahedrons  $e$  and  $ep$  of the same cluster  $m$ , assumed positive if going from the element with the smallest index to the element with the highest index, and

$$\delta_{e,j}^s = 1 \text{ if face } s \text{ belongs to element } e \text{ and } e < ep \quad (68b)$$

$$\delta_{e,j}^s = -1 \text{ if faces belongs to element } e \text{ and } e > ep \quad (68c)$$

$$\delta_{e,j}^s = 0 \text{ if face } s \text{ does not belong to element } e \quad (68d)$$

and  $F_{j,e}^{\text{int}} = \delta_{j,e}^s FI_s$ , if  $s$  the index of the face between elements  $e$  and  $ep$ . The unknowns  $\lambda_e$  are the so-called Lagrangian multipliers. Setting at zero the derivatives of functional  $\mathfrak{S}$  with respect to  $FI_s$  and  $\lambda_e$ , we get the linear system

$$\begin{aligned} & \sum_{e=1, N_{T,m}} \left\{ \sum_{j=1,4} \left[ \sum_{q=1,3} \omega_{j,q}^e \left( \tilde{\varphi}_j \sum_{s=1, N_{f,m}^{\text{int}}} \delta_{e,j}^s FI_s \right. \right. \right. \\ & \quad \left. \left. + (1 - \tilde{\varphi}_j)(F_{j,e}^{\eta} + \overline{F}_{j,e}^{k+2/3}) \right) \omega_{j,q}^e \right] \tilde{\varphi}_j \delta_{e,j}^r W_e \right\} \\ & + \sum_{e=1, N_{T,m}-1} \left\{ \lambda_e \sum_{j=1,4} (\tilde{\varphi}_j \delta_{e,j}^r) \right\} \\ & = 0 \text{ with } r = 1, \dots, N_{f,m}^{\text{int}} \quad (69a) \end{aligned}$$

$$\begin{aligned} & \sum_{j=1,4} \left( \tilde{\varphi}_j \sum_{s=1, \dots, N_{S,m}} \delta_{e,j}^s FI_s + (1 - \tilde{\varphi}_j)(F_{j,e}^{\eta} + \overline{F}_{j,e}^{k+2/3}) \right) \\ & = 0 \text{ for } e = 1, \dots, N_{T,m}-1 \quad (69b) \end{aligned}$$

Observe that Equation (69b) represent the mass conservation equations for all the tetrahedrons of cluster  $m$ . Moreover, if  $N_{f,m}^{\text{int}} = N_{T,m}-1$  (see the 2D sketch in Figure 6(a)), system (69b) can be solved independently of Equation (69a) and there is only one set of internal fluxes that satisfy the mass conservation equations. If  $N_{f,m}^{\text{int}} \geq N_{T,m}$  (see the 2D sketch in Figure 6(b)), we also have to compute Equation (69a) along with Equation (69b), because we need to select, among all the sets of internal fluxes that satisfy mass conservation,

the one that minimizes the kinetic energy within the cluster.

Equation system (69) has no special structure and has to be solved with direct solvers, but it is small and only includes a few tetrahedrons of the mesh, so its solution requires a negligible computational burden.

After Equations (64)-(69) are solved, in non-Delaunay meshes we cannot compute the gradient  $\nabla \Psi_e^{k+1}$  of each tetrahedron by applying Equation (48) with the computed  $\mathbf{u}_e^{k+1}$  velocity. The reason is that, after the first sub-step of the CS2 problem, the computed solution satisfies the momentum and continuity equations of the clusters (not of the single tetrahedrons) and only the fluxes crossing the external faces of the clusters are computed under these constraints. The fluxes crossing the internal faces of each cluster are computed, during the second sub-step of the CS2 problem, by satisfying the continuity equations only. For these reasons, we compute  $\nabla \Psi_e^{k+1}$  by applying the following general procedure.

We look for a new velocity  $\mathbf{u}_m^c$  common to all the tetrahedrons of the cluster. Let  $\mathfrak{S}'$  be a scalar functional, defined as

$$\mathfrak{S}' = \sum_{l=1, N_{f,m}^{\text{ext}}} (\mathbf{u}_m^c \cdot \mathbf{n}_l - fl_{u,l})^2 \quad (70a)$$

$$fl_{u,l} = \frac{(\overline{F}_{l,e}^{k+2/3} + F_{l,e}^{\eta})}{\sigma_l^m} \quad (70b)$$

where  $fl_{u,l}$  is the flux per unitary area crossing the  $l$ -th external face of the cluster,  $\mathbf{n}_l$  is the unit outward vector normal to face  $l$ , and all the other symbols have already been defined. In the case of a cluster made up of a single tetrahedron, according to Equation (47)  $fl_{u,l}$  is the flux per unit area of the velocity  $\mathbf{u}_e^{k+1}$  crossing the external  $l$  face.

The relationship of  $\mathbf{u}_m^c$  with the pressure gradient  $\nabla \Psi_e^{k+1}$  is given by Equation (48), where  $\mathbf{u}_e^{k+1}$  is replaced by  $\mathbf{u}_m^c$  in the case of  $N_{T,m} > 1$ . Observe that the second step of the CS2 problem does not change the fluxes crossing the external faces and that the pressure gradient  $\nabla \Psi_e^{k+1}$  is not affected by the corrected fluxes of the internal faces. To compute  $\mathbf{u}_m^c$  we minimize  $\mathfrak{S}'$ , which is equivalent to setting at zero the partial derivatives of the functional with respect to the three components of  $\mathbf{u}_m^c$ . The size of the resulting system is  $(3 \times 3)$ , it does not have a special structure, and, as for system (69), is solved using direct solvers.

Once  $\nabla \Psi_e^{k+1}$  is computed, we obtain the kinematic pressure force on each element of the cluster by setting in all the elements

$$\mathbf{S}_{\Psi_e}^{k+1} = \nabla \Psi_e^{k+1} W_e \quad (71)$$



Vector  $\mathbf{S}_{\Psi,e}^{k+1}$  is assumed equal to  $\mathbf{S}_{\Psi,e}^k$  in the MAST PS of the next time iteration (see Equation (16) in section 3.1).

#### 4.4. The MAST-RT0 pseudo code in the case of non-Delaunay meshes

- (1) Compute cluster geometry and model constants, including CS1 and CS2 matrix coefficients by means of Equations (60) and (63a). Perform matrix factorization and set  $k = 1$
- (2) Given  $\mathbf{u}^k$  and  $\nabla\Psi^k$  at time  $t^k$ , apply the MAST prediction step (PS)
  - (i) Compute velocity variation  $\Delta\mathbf{u}_e^f(\Delta t)$  and  $\Delta\mathbf{u}_e^b(\Delta t)$  by solving Equations (16) and (18) for each tetrahedron  $e$
  - (ii) update  $\mathbf{u}_e^{k+1/3} = \mathbf{u}_e(\Delta t)$  for each tetrahedron  $e$  by means of Equation (18b)
  - (iii) compute one velocity variation  $\Delta\tilde{\mathbf{u}}_m$  for each cluster  $m$  according to Equation (55)
  - (iv) update velocity  $\mathbf{u}_e^{k+1/3}$  for each tetrahedron  $e$  of each cluster  $m$  by means of Equation (56)
- (3) Apply the 1st corrective step (CS1)
  - (i) Solve system (59) for the  $\Delta\hat{\mathbf{u}}_m$  unknown for each cluster  $m$
  - (ii) update velocity  $\mathbf{u}_e^{k+2/3}$  for each tetrahedron  $e$  of each cluster  $m$  by means of Equation (58)
- (4) Update the viscous forces  $\mathbf{FV}_e^{k+2/3}$  for each tetrahedron  $e$  by means of Equation (61)
- (5) Compute fluxes  $\overline{F}_{l,e}^{k+2/3}(\mathbf{u}^{k+2/3})$  for each external face  $l$  of tetrahedron  $e$  of each cluster  $m$ , by means of Equation (39)
- (6) Apply the 2<sup>nd</sup> corrective step (CS2)
  - (i) Compute  $\eta_m$  for each cluster  $m$  by solving system (62)
  - (ii) Compute the fluxes  $F_{j,e}^{int}$  of the internal faces  $j$  of tetrahedrons  $e$  of the cluster  $m$  according to Equations (66)-(69)
  - (iii) Compute fluxes  $F_{j,e}^m$  of the cluster external faces by means of Equation (46)
  - (iv) Update the final velocity  $\mathbf{u}_e^{k+1}$  for each tetrahedron  $e$  according to Equation (64)
  - (v) Compute  $\mathbf{u}_m^c$  by solving the minimization problem in Equations (70)
- (7) Update the gradient of the kinematic pressure  $\nabla\Psi^{k+1}$  according to Equation (48), where  $\mathbf{u}_e^{k+1}$  is replaced by  $\mathbf{u}_m^c$
- (8) Compute the kinematic pressure forces  $\mathbf{S}_{\Psi,e}^{k+1}$  for each tetrahedron  $e$  by means of Equations (70)-(71)
- (9) Set  $k = k+1$  and go to point 2 for the solution of the next time step.

## 5. Model applications

### 5.1. Construction of the computational mesh and preliminary model operations

As mentioned in section 3.3, in the 3D space it is almost impossible to obtain a mesh satisfying the EDP given in Equations (10)–(11) (Forsyth, 1991; Joe, 1986 and cited references). Some algorithms exist (e.g. Qhull in Matlab<sup>3</sup>), which generate a 3D tetrahedral mesh forming a convex hull with arbitrary node location. Unfortunately, the generated mesh has very irregular elements inside it (e.g. slivers, caps, skinny tetrahedrons, ...), along with several distances  $d_{e,ep}$  (see Equation (25)) strictly equal or close to zero. These irregularities always lead to instability and poor accuracy of the numerical solution (Letniowski, 1992).

The computational mesh used by the present solver is created by an off-line procedure, using two open source mesh generators, Netgen (Schöberl, 1997) and Tetgen (Hang, 2015).

We discretize first the computational domain with tetrahedrons using the mesh generator Netgen. The tetrahedral elements of the output mesh are quite regular in shape and size, even if they do not satisfy the EDP. Netgen also allows us to change the size of the tetrahedrons to discretize the internal subdomains properly, with quite smooth transitions of element size. Netgen also allows ‘user-friendly’ handling of the boundary domain.

Let  $N_{NET}$  be the number of nodes of the Netgen mesh. For each boundary triangle  $bt$  we compute the corresponding circumsphere  $\Sigma_{bt}$  whose diametral plane contains  $bt$ , and we check if any internal node (i.e. not belonging to boundary faces) is internal to  $\Sigma_{bt}$ . At the end of this operation, we remove all the internal nodes inside the circumsphere of the boundary triangles (usually 0.02-0.03% of  $N_{NET}$ ). We call the number of the removed nodes  $N_r$ . The ensemble of the  $(N_{NET} - N_r)$  nodes is the input for the mesh generator TETGEN, which regenerates the mesh domain starting from the nodal positions of the  $(N_{NET} - N_r)$  nodes, still preserving the input domain boundaries. In order to optimize some aspect-ratio and shape tetrahedrons conditions, Tetgen can also insert additional Steiner nodes (Hang, 2015). The present numerical solver uses as its input the Tetgen output mesh.

Preliminary model operations concern (1) generation of the topology of the tetrahedrons and clusters of tetrahedrons, saved in separate arrays, (2) calculation of the matrix coefficients of systems (59)–(60) and (62), and (3) their factorization before the time iterations loop start.

## 5.2. Computation of the kinematic pressure $\psi$ and of the body forces

Let  $N$  be the number of nodes of the computational mesh. We approximate the function  $\psi$  according to a Galerkin Finite Element approach,

$$\Psi = \sum_{i=1,N} w_i \tilde{\Psi}_i \quad (72)$$

where  $\tilde{\Psi}_i$  is the unknown nodal pressure value and  $w_i$  is the Galerkin shape function in node  $i$ . Once  $\nabla \Psi_e^{k+1}$  has been obtained inside each tetrahedron  $e$  as specified in sections 3.3 and 4.3, we minimize the following scalar functional,

$$\mathfrak{S}_\Psi = \sum_{e=1,N_T} (\nabla \tilde{\Psi}_e^{k+1} - \nabla \Psi_e^{k+1})^2 \quad (73a)$$

rewritten as

$$\mathfrak{S}_\Psi = \sum_{e=1,N_T} \left[ \sum_{q=1,3} \left\{ \sum_{i=1,N} \left( \frac{\partial w_i}{\partial x_q} \tilde{\Psi}_i - (\nabla_{x_q} \Psi^{k+1})_e \right)^2 \right\} \right] \quad (73b)$$

$$q = \begin{cases} 1 \\ 2 \\ 3 \end{cases} \Rightarrow x_q = \begin{cases} x \\ y \\ z \end{cases}$$

where  $(\nabla_{x_q} \Psi^{k+1})_e$  is the  $q$ -th components of  $\nabla \Psi^{k+1}$  in tetrahedron  $e$ , previously computed.  $\mathfrak{S}_\Psi$  is a convex function and its minimum is obtained by setting at zero the partial derivatives of Equation (73b) with respect to the nodal  $\tilde{\Psi}_i^{k+1}$  values,

$$\frac{\partial \mathfrak{S}_\Psi}{\partial \tilde{\Psi}_i^{k+1}} = 0 \quad i = 1, \dots, N \quad (74a)$$

Equation (74a) can be written, according to Equation (72), as

$$\frac{\partial \mathfrak{S}_\Psi}{\partial \tilde{\Psi}_i^{k+1}} = \sum_{e=1,N_T} \left[ \sum_{q=1,3} \left\{ \sum_{l=1,N} \left( -\frac{\partial w_l}{\partial x_q} \tilde{\Psi}_l - (\nabla_{x_q} \Psi^{k+1})_e \right) \frac{\partial w_i}{\partial x_q} \right\} \right] = 0 \quad (74b)$$

Equation (74b) represents a linear system solved for the  $\tilde{\Psi}_l$  unknowns, with an  $(N \times N)$  system matrix that is sparse, symmetric and positive-definite. Over the boundary faces of  $\Gamma_\sigma$ , we assign Dirichlet BCs for  $\tilde{\Psi}_l$  according to the prescribed boundary values. The diagonal and

off-diagonal coefficients are

$$M_{i,i}^\Psi = \sum_{q=1,3} \frac{\partial w_i}{\partial x_q} \frac{\partial w_i}{\partial x_q}, M_{i,l}^\Psi = \sum_{q=1,3} \frac{\partial w_i}{\partial x_q} \frac{\partial w_l}{\partial x_q} \quad (75a)$$

and the  $i$ -th coefficient of the source term vector is

$$S_i^\Psi = -(\nabla_{x_q} \Psi^{k+1})_e \frac{\partial w_i}{\partial x_q} \quad (75b)$$

We solve system (74)–(75) as the previous ones in sections 3.2, 3.3, or 4.2 and 4.3.

The forces over the bodies are easily computed as,

$$\mathbf{F} = \mathbf{F}_\Psi + \mathbf{F}_\nu \text{ with } \mathbf{F}_\Psi = \sum_{b=1,N_{bf}} \left( \sigma_b \frac{(\sum_{f_b=1,2,3} \tilde{\Psi}_{f_b}) \mathbf{n}_b}{3} \right) \quad (76a)$$

and

$$\mathbf{F}_\nu = \sum_{b=1,N_{bf}} \left( \nu \frac{(\mathbf{u}_e^{k+1} - \mathbf{g}_b)}{d_j^e} \sigma_b \right) \text{ if } b \in \Gamma_u \quad (76b)$$

$$\mathbf{F}_\nu = 0 \text{ if } b \in \Gamma_\sigma \vee \Gamma_m \quad (76c)$$

where  $N_{bf}$  is the number of triangular faces of the body,  $\sigma_b$  and  $\mathbf{n}_b$  are the area of the  $b$ -th body face and the unit normal vector coming into the body, respectively,  $\tilde{\Psi}_{f_b}$  is the value of the kinematic pressure at  $f_b$ -th node of the body face,  $\mathbf{g}_b$  is the velocity vector assigned to the face,  $\mathbf{u}_e^{k+1}$  is the velocity vector in the tetrahedron  $e$  with the  $b$ -th body face, and  $d_j^e$  is the distance between the circumcenter of tetrahedron  $e$  and the circumcenter of the body face, computed as in Equation (33).

## 5.3. Test cases

The proposed numerical solver was applied to four different numerical tests. In the first test we analyzed the spatial and temporal accuracy of the model, as well as the required computational costs. In tests 2 and 3 we present two well-known literature applications, the lid driven cavity and the flow past a fixed sphere, according to different values of the Reynolds number. In the last test the real case of hemodynamic blood flow inside an abdominal aorta affected by an aneurysm is solved by the model inside a very irregular boundary.

For post-processing of the model outputs, we used the open source program Paraview.<sup>4</sup>

### 5.3.1. Taylor-Green vortices test

We first analyzed the accuracy of the proposed solver by comparison of the computed results with the known analytical solution of the Taylor–Green vortex test (Taylor &

Green, 1937). The velocity vector components and the pressure field are (Taylor & Green, 1937),

$$\begin{aligned} u &= \cos(\alpha x) \sin(\alpha y) e^{-\beta t} \\ v &= -\sin(\alpha x) \cos(\alpha y) e^{-\beta t} \quad w = 0 \\ p &= -\frac{\cos(2\alpha x) + \cos(2\alpha y)}{4} e^{-2\beta t} \\ a &= \frac{\pi}{2}, \beta = 2\nu a^2, \quad \nu = 0.01 \frac{m^2}{s} \end{aligned} \quad (77)$$

such that the r.h.s. of Equation (1) is always zero. The initial conditions are found by setting  $t = 0$  in Equation (77). We solve this problem up to 2 s in the domain  $[-3, 3]^2 \times [0, 0.25]$ , by setting time-dependent essential BCs for the velocity, according to Equation (77), in the circumcenters of the boundary triangular faces of the lateral walls of the domain ( $\Gamma_u$  in section 2.1), as well as the hydrostatic stress in the nodes of the upper and lower wall ( $\Gamma_\sigma$  in section 2.1). The initial velocity vectors are set in the circumcenters of each tetrahedron, and assumed piecewise constant, while the initial kinematic pressures are set in the nodes. The initial values of the forces  $\mathbf{S}_{\Psi,e}^0$  and  $\mathbf{VF}_e^0$  of the momentum equations of the MAST PS of the first time iteration are analytically found as

$$\mathbf{S}_{\Psi,e}^0 = \nabla \Psi_e^0 W_e \quad (78a)$$

$$\mathbf{VF}_e^0 = \nu \Delta_2 \mathbf{u}_e^0 W_e \quad (78b)$$

where  $\nabla \Psi_e^0$  is computed from Equation (77) in the center of mass of tetrahedron  $e$ .

We discretize the domain using 5 meshes with mean element characteristic size  $h_l$  ( $l = 1, \dots, 5$ ) ranging from 0.00996 m to 0.066 m.  $h_l$  is the mean value of the length of the sides of the tetrahedrons.

The discretized ICs  $\mathbf{u}_e^k = \mathbf{u}_{e,0}$  do not satisfy the momentum and continuity equations of each element at time  $t = 0$ , as  $\mathbf{u}_{e,0} \in P_{0,e}$ , but the flux continuity through

each face is missing. In order to circumvent this problem, before the beginning of the transient flow simulation with the BCs given by Equation (77), we compute the steady-state asymptotic solution corresponding to the BCs at  $t = 0$  and we use it as ICs of the transient problem. We assume that the steady-state solution is attained when the  $L_2$  norm of the relative scatters of the computed  $u$ ,  $v$ ,  $w$ , and  $\psi$ , compared with the ones of the previous iteration, are small and the following tolerance holds

$$\begin{aligned} & \sqrt{\sum_{e=1, N_T} \left( \frac{u(v, w)_e^{k+1} - u(v, w)_e^k}{u(v, w)_e^k} \right)^2} \\ & < 1e - 04 \text{ .and. } \sqrt{\sum_{i=1, N} \left( \frac{\Psi_i^{k+1} - \Psi_i^k}{\Psi_i^k} \right)^2} < 1e - 04 \end{aligned} \quad (79)$$

See in Tables 1 and 2 the  $L_2$  and  $L_\infty$  norms of the relative errors of the steady-state  $x$  and  $y$  velocity components and kinematic pressure with respect to the values computed at the face circumcenters according to Equation (77), for meshes with different size. The error of the  $z$  velocity component is negligible compared to the  $x$  and  $y$  errors.

We also assume that the relative error  $err_l$ , computed for the mesh with mean element size  $h_l$ , is proportional to a power of  $h_l$ ,

$$err_l = (h_l)^{r_{c,s}} \quad (80a)$$

where  $r_{c,s}$  is the spatial rate of convergence, obtained by comparing the relative errors of two sequential sizes  $h_l$  and  $h_{l+1}$  as

$$r_{c,s} = \frac{\log\left(\frac{L_2 err_l}{L_2 err_{l+1}}\right)}{\log\left(\frac{h_l}{h_{l+1}}\right)} \quad (80b)$$

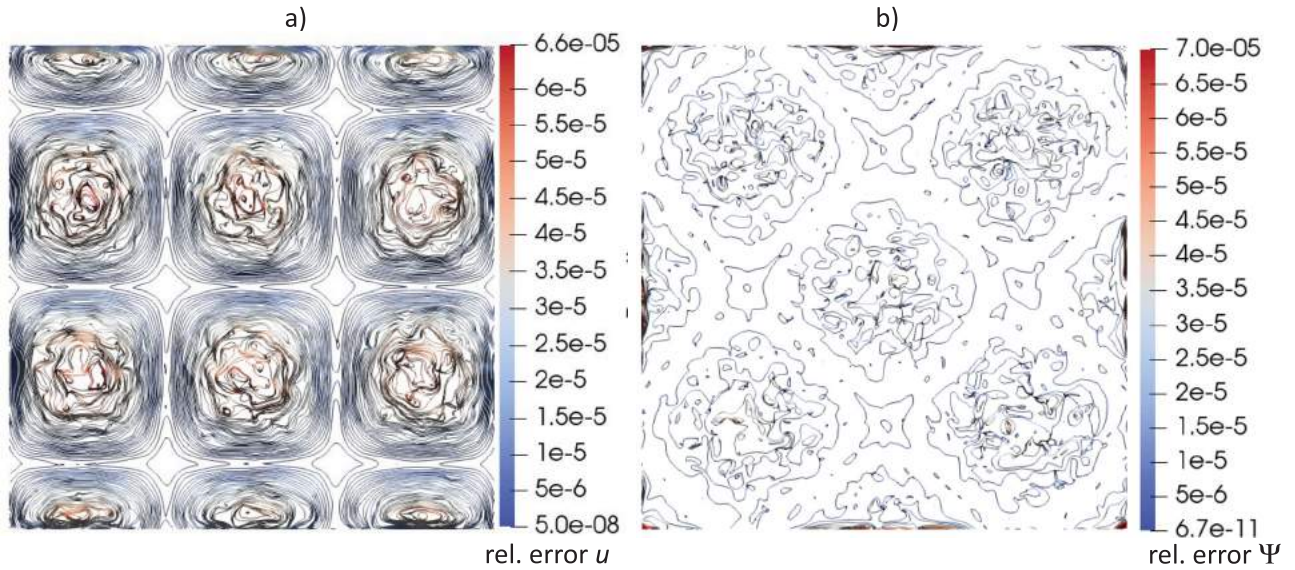
The rate of convergence is shown in Tables 1 and 2. The computed  $r_{c,s}$  of the velocity components are slightly

**Table 1.** Test 1.  $L_2$  norms of relative errors and spatial rate of convergence.

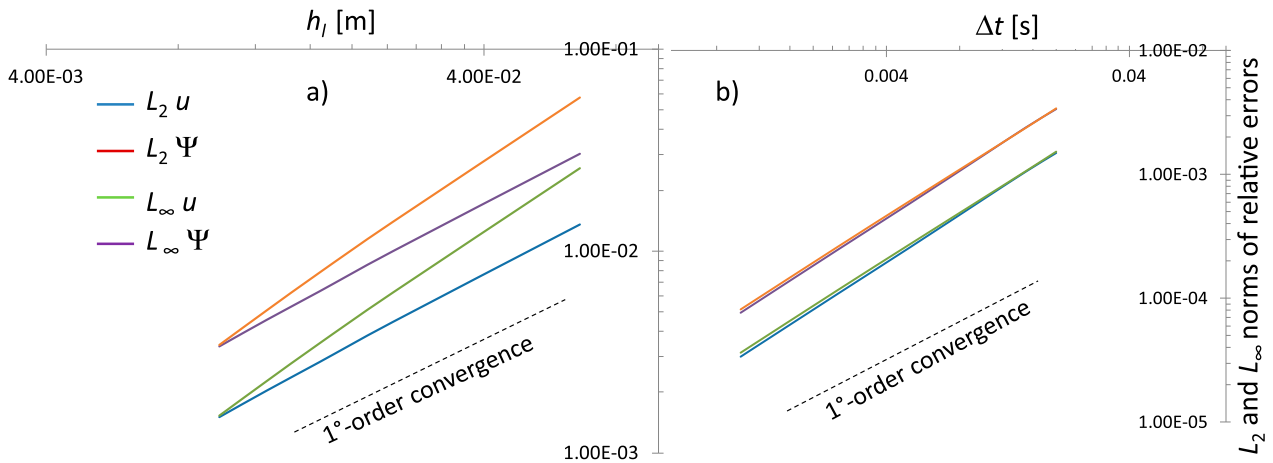
$h_l$ [m]	$L_2$ err $u$	$L_2$ err $v$	$L_2$ err $\psi$	$r_{c,s} u$	$r_{c,s} v$	$r_{c,s} \psi$
6.60E-02	1.36E-02	1.37E-02	2.58E-02	-	-	-
2.47E-02	4.45E-03	4.47E-03	6.24E-03	1.14E+00	1.14E+00	1.45E+00
1.69E-02	2.82E-03	2.84E-03	3.50E-03	1.19E+00	1.18E+00	1.50E+00
1.28E-02	2.03E-03	2.05E-03	2.28E-03	1.18E+00	1.17E+00	1.55E+00
9.96E-03	1.50E-03	1.51E-03	1.53E-03	1.22E+00	1.21E+00	1.59E+00

**Table 2.** Test 1.  $L_\infty$  norms of relative errors and spatial rate of convergence.

$h_l$ [m]	$L_\infty$ err $u$	$L_\infty$ err $v$	$L_\infty$ err $\psi$	$r_{c,s} u$	$r_{c,s} v$	$r_{c,s} \psi$
6.60E-02	3.04E-02	3.07E-02	5.77E-02	-	-	-
2.47E-02	9.98E-03	1.00E-02	1.40E-02	1.13E+00	1.14E+00	1.44E+00
1.69E-02	6.32E-03	6.35E-03	7.85E-03	1.19E+00	1.18E+00	1.50E+00
1.28E-02	4.56E-03	4.59E-03	5.10E-03	1.18E+00	1.17E+00	1.55E+00
9.96E-03	3.37E-03	3.39E-03	3.44E-03	1.22E+00	1.21E+00	1.59E+00



**Figure 7.** Test 1. Iso-contours of the relative error of the norm of (a)  $u$ , (b)  $\psi$ .



**Figure 8.** Test 1. Investigation of the spatial and temporal accuracy. (a) Norms of relative errors vs. mean mesh size, (b) norms of relative errors vs. time step size.

greater than 1 (ranging from 1.14–1.22) due to the piecewise approximation of  $u$  and  $v$  inside each tetrahedron. The convergence rate obtained for the kinematic pressure is greater (ranging from 1.45–1.59) and the reason could be the nodal pressure distribution inside each tetrahedron, as described in section 5.2. In Figure 7 we plot the iso-contour lines, over a horizontal plane with  $z = 0.125$  m, of the relative errors of the norm of  $u$  and of  $\psi$  obtained for the coarsest mesh. Close to the lateral boundary walls, the errors of the velocity decrease, due to the imposed BCs. On the opposite, the highest values of the relative errors of the kinematic pressure are close to the boundary lateral walls, since the Dirichlet BCs of  $\psi$  have been imposed over the upper and lower horizontal walls.

In Figure 8(a) we plot the norms of the relative errors, along with the 1st order convergence line.

With this test we also analyzed the time convergence rate of the algorithm. In order to cancel out the error due to the spatial discretization, we assumed as the reference solution the one obtained over the finest mesh (with mean size  $h_l = 0.00996$  m), and a time step size  $\Delta t = 0.001$  s. At simulation time 2 s, we compared with the reference solution the numerical solutions obtained over the same mesh, assuming five different  $\Delta t$  values, ranging from 0.0015 s to 0.02 s. The  $L_2$  and  $L_\infty$  norms of the relative errors are shown in Tables 3 and 4, along with the time rate of convergence,  $r_{c,t}$ , computed by comparing the relative errors of two sequential time step sizes  $\Delta t_l$  and  $\Delta t_{l+1}$



**Table 3.** Test 1.  $L_2$  norms of relative errors and time rate of convergence.

$\Delta t$ [s]	$L_2$ err $u$	$L_2$ err $v$	$L_2$ err $\psi$	$r_{c,t}u$	$r_{c,t}v$	$r_{c,t}\psi$
0.02	1.50E-03	1.51E-03	1.53E-03	–	–	–
0.015	1.06E-03	1.06E-03	1.07E-03	1.20E+00	1.22E+00	1.26E+00
0.01	6.33E-04	6.33E-04	6.47E-04	1.28E+00	1.28E+00	1.24E+00
0.005	2.60E-04	2.61E-04	2.74E-04	1.28E+00	1.28E+00	1.24E+00
0.0025	1.08E-04	1.08E-04	1.16E-04	1.26E+00	1.27E+00	1.25E+00
0.001	3.39E-05	3.45E-05	3.63E-05	1.27E+00	1.25E+00	1.27E+00

**Table 4.** Test 1.  $L_\infty$  norms of relative errors and time rate of convergence.

$\Delta t$ [s]	$L_\infty$ err $u$	$L_\infty$ err $v$	$L_\infty$ err $\psi$	$r_{c,t}u$	$r_{c,t}v$	$r_{c,t}\psi$
0.02	3.37E-03	3.39E-03	3.44E-03	–	–	–
0.015	2.39E-03	2.38E-03	2.39E-03	1.19E+00	1.22E+00	1.26E+00
0.01	1.42E-03	1.42E-03	1.45E-03	1.29E+00	1.28E+00	1.24E+00
0.005	5.83E-04	5.84E-04	6.15E-04	1.28E+00	1.28E+00	1.24E+00
0.0025	2.44E-04	2.43E-04	2.59E-04	1.26E+00	1.27E+00	1.25E+00
0.001	7.62E-05	7.73E-05	8.12E-05	1.27E+00	1.25E+00	1.27E+00

$$r_{c,t} = \frac{\log\left(\frac{L_2 \text{err}_l}{L_2 \text{err}_{l+1}}\right)}{\log\left(\frac{\Delta t_l}{\Delta t_{l+1}}\right)} \quad (80c)$$

The rate of convergence  $r_{c,t}$  is always greater than 1, ranging from 1.21–1.28, even if the model is 1st order accurate in time. This could be due to a twofold reason related to the MAST-PS, (1) the use of the internal time sub-grid during the ODEs solution, and (2) a polynomial time approximation order of the leaving momentum fluxes, using  $n_G = 3$  Gauss points. In Figure 8(b) we plot the norms of the relative errors along with the 1st-order convergence line.

For the finest mesh, for each time step adopted for the time convergence rate analysis, we computed the maximum CFL number, as

$$\text{CFL}_{\max} = \max\left(\frac{\Delta t}{\sqrt[3]{W_e}} \|\mathbf{u}_e\|\right) \quad e = 1, \dots, N_T \quad (81)$$

$\text{CFL}_{\max}$  ranges from 2.56 (for  $\Delta t = 0.02$ ) to 0.0128 (for the reference solution).

We also investigated the computational (CPU) times required by the different model steps, using a single Intel Core i7 at 3.49 GHz. Because computational times strongly depends on the adopted computer and on the specific algorithm coding, we focused on the correlation existing between the the computational time of the single step and the number of elements. We set the average CPU time per iteration and per model step equal to

$$\overline{\text{CPU}}_{\text{step}} = \exp(c) N_T^\beta \quad (82a)$$

and we assumed that a single step is efficiently solved as much as the  $\beta$  power exponent in the correlation 82(a) is small and close to 1. Equation (82a) in logarithmic space

becomes

$$\ln(\overline{\text{CPU}}_{\text{step}}) = c + \beta \ln(N_T) \quad (82b)$$

See in Figure 9 the  $\overline{\text{CPU}}_{\text{step}}$  time required for the solution of the single model steps, i.e. cell sorting ( $\overline{\text{CPU}}_S$ ), solution of the MAST-PS step ( $\overline{\text{CPU}}_{\text{MPS}}$ ), solution of the CS1 and CS2 steps ( $\overline{\text{CPU}}_{\text{CS1}}$  and  $\overline{\text{CPU}}_{\text{CS2}}$ , respectively), as well as the kinematic pressure computation ( $\overline{\text{CPU}}_\psi$ ). MAST-PS is the most demanding one, but in this case the CPU is simply proportional to  $N_T$ , and in Equation (82a) power  $\beta$  is equal to one. The CPU required by the other model steps grows in the logarithm space more than linearly with the number of tetrahedrons due to their ‘non-explicit nature’, since solution of large linear systems is involved, but  $\beta$  is smaller than 1.20 for the CS2 step, and smaller than 1.12 in all the other ones. The sorting cell operation is the least demanding algorithm step and its CPU time is 2–3 magnitude orders smaller than the MAST-PS one.

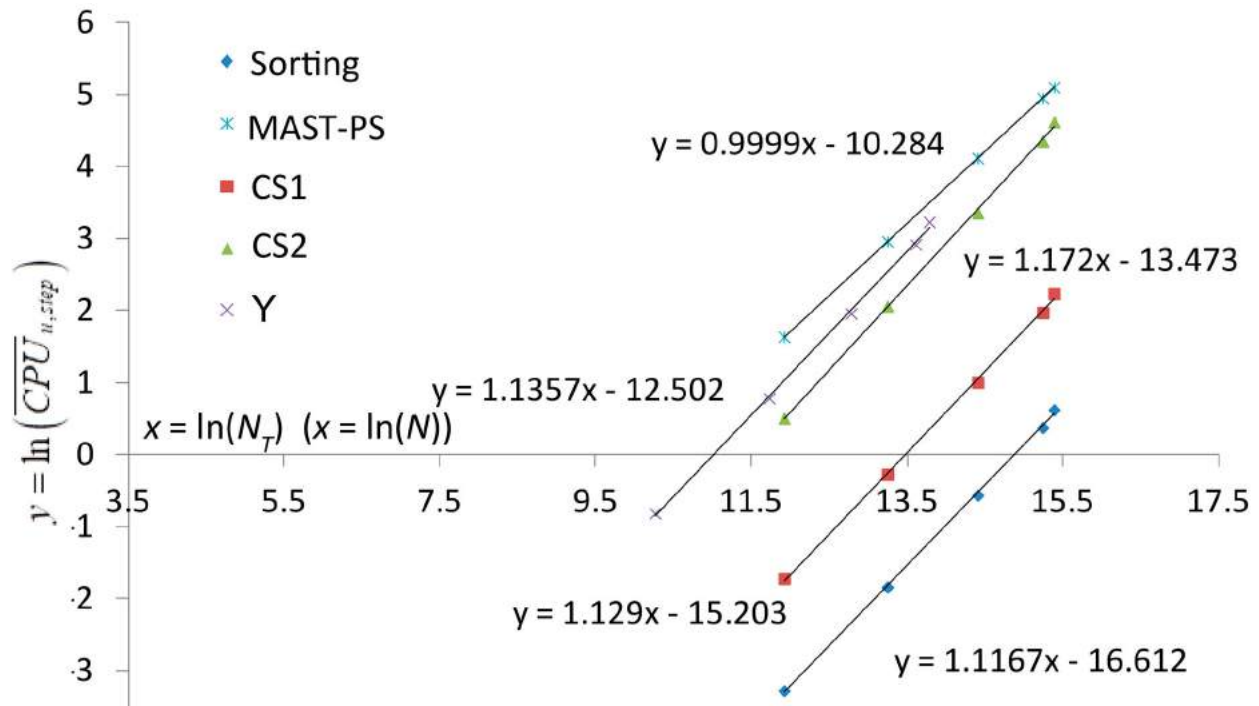
### 5.3.2. Lid driven cavity flow at different Reynolds numbers

In this test the flow is confined inside a square cavity, and it is driven by the upper wall displacement in horizontal direction. The set-up of the test is shown in Figure 10(a). Cavity is  $[0, L] \times [0, 0.2] \times [0, L]$  with  $L = 1$  m. At the east, west and bottom walls we set zero velocity with no slip BCs ( $\Gamma_u$ ), while at the front and rear walls we set free slip BCs ( $\Gamma_m$ ), and we impose a constant in time horizontal velocity equal to 1 m/s over the top wall with no slip BCs ( $\Gamma_u$ ). We set zero kinematic pressure in the lowest left corner with coordinates (0, 0, 0), as shown in Figure 10(a). ICs are zero velocity and pressure inside domain.

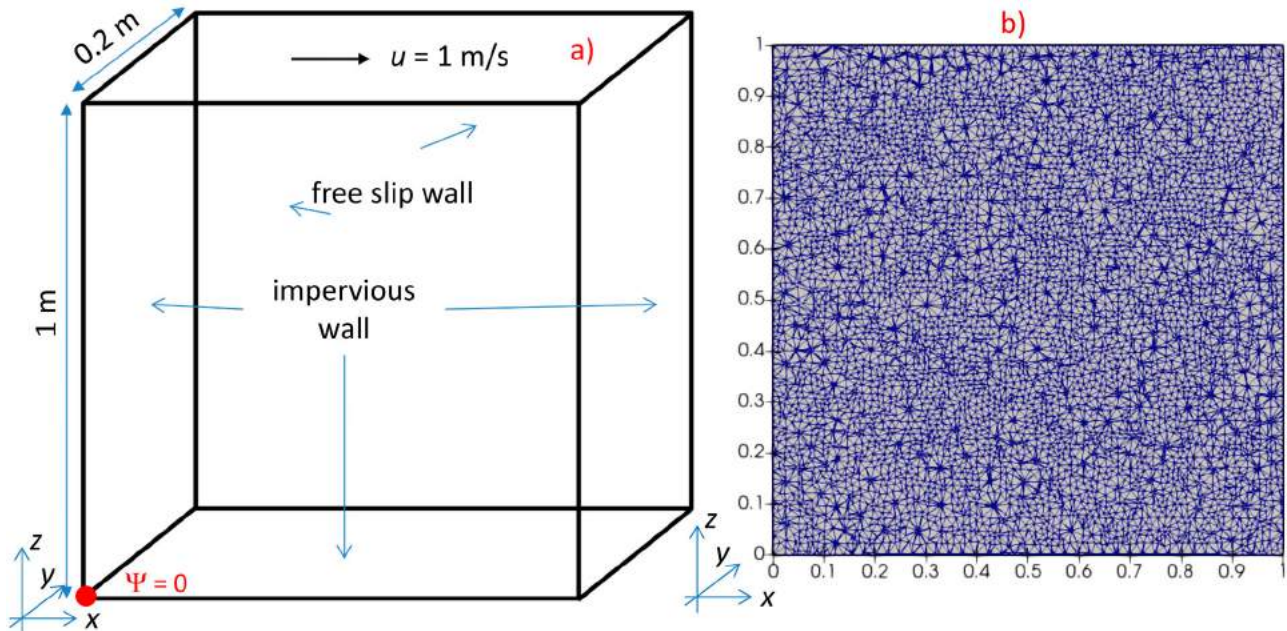
Let the Reynolds number be

$$Re = \frac{v_{\max} L}{\nu} \quad (83)$$





**Figure 9.** Test 1. Computational time of model steps.

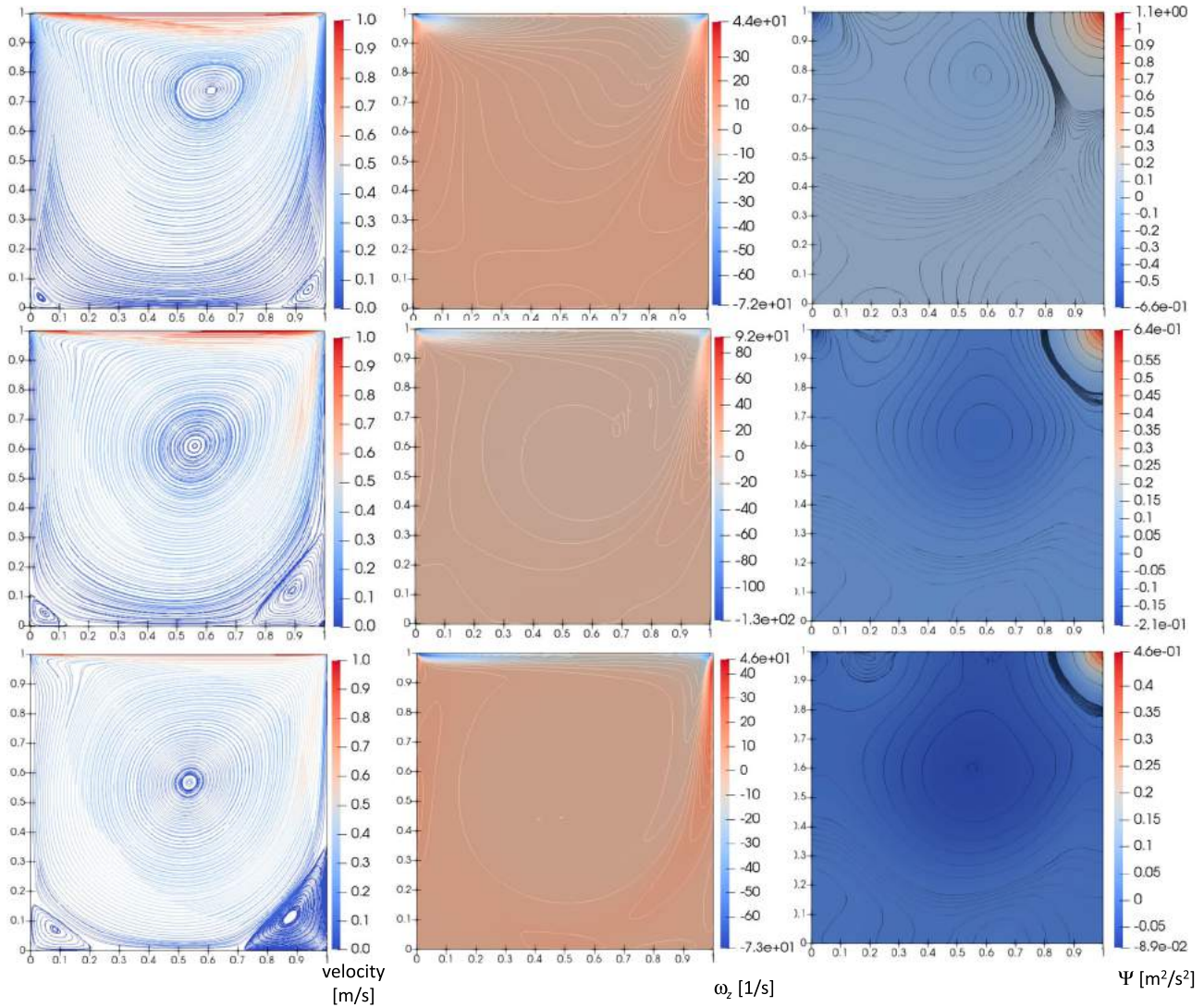


**Figure 10.** Test 2. (a) set-up of the numerical test and BCs. (b) section of the mesh with a cutting plane.

where  $v_{max} = 1$  m/s. We run simulations for  $Re = 100$ , 400 and 1000.

The imposed horizontal velocity of the upper lid drives the fluid inside the cavity into a vortical flow. The resulting complex flow structure shows a large central vortex and small recirculating zones close to the cavity corners, whose shape depends on the value of  $Re$ .

We discretize the domain with 87,740 tetrahedrons and 17,388 nodes (see in Figure 10(b) an intersection of the mesh with a cutting plane), resulting in 84,385 clusters. The time step size  $\Delta t$  is set to 0.025 s, and the maximum computed CFL values are 2.18, 2.08 and 1.96 for  $Re = 100$ , 400 and 1000, respectively.

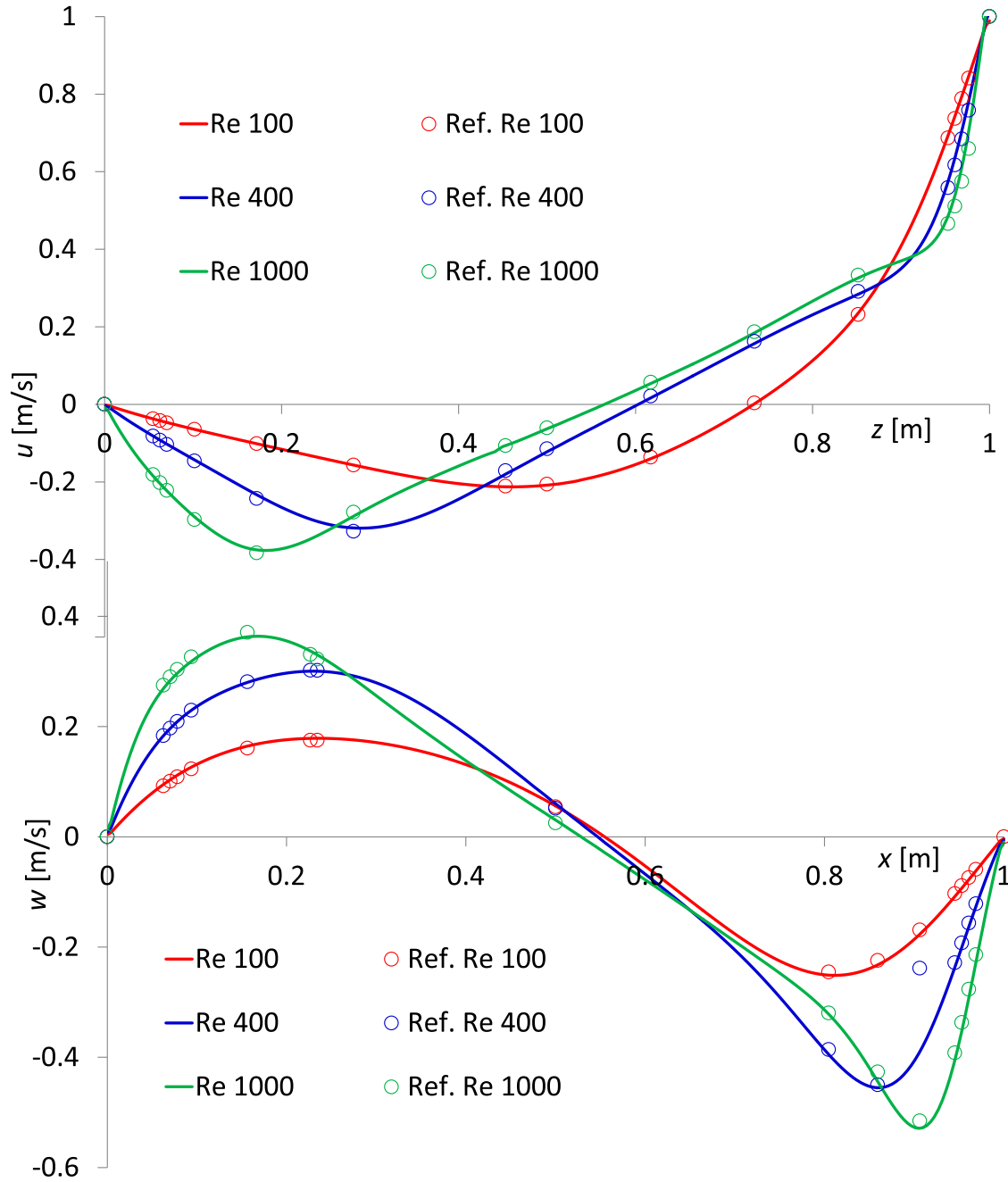


**Figure 11.** Test 2. Velocity streamlines (left panels), vorticity ( $\omega_z$ ) (central panels), iso (kinematic) pressure (right panels). Top  $Re = 100$ , middle  $Re = 400$ , bottom  $Re = 1000$ .

In Figure 11 we plot the streamlines, as well as the vorticity and the iso-pressure fields. The minimum and maximum pressures are computed respectively at the upper-left and upper-right corners, where a discontinuity in the boundary condition occurs. Observe that, because of this singularity, no *ad-hoc* handling is required in the model, unlike what is found in Botella and Peyret (1998), Boppana and Gajjar (2010), Kuhlmann and Romanò (2019), and cited references. The minimum pressure values are associated with the foci of the vortices, due to the high centrifugal acceleration occurring around them. We obtain good agreement with the literature results (e.g. Dalal et al., 2008 and cited references), and the results provided by the present solver match very well the ones of the 2D benchmark solutions given by Ghia et al. (1982) and shown in Figure 12. The results in Figures 11 and 12

refer to the cutting plane ( $x$ - $z$ ) with  $y = 0.1$  m (i.e. the diametral plane of the domain). The results obtained for other cutting planes ( $x$ - $z$ ) are very similar to the previous ones, and for brevity are not shown here.

Observe that the value of the vertical velocity component computed for  $Re = 400$  and  $x = 0.9063$  provided by Ghia et al. (1982) ( $w_{Ghia} = -0.23827$ ) (see table 2 of the referred paper) is quite different from the result obtained by the present solver. On the other hand, the  $w_{Ghia}$  result is missing in most of the papers where the lid-driven test is used as benchmark, including the papers reporting the solution by Ghia et al. (1982) for many other points (e.g. Xue & Burton, 2013 and many others), whereas in other papers the mentioned  $w_{Ghia}$  vertical velocity component does not match the result, like in Dalal et al. (2008).



**Figure 12.** Test 2.  $x$  velocity component (top panel),  $z$  velocity component (bottom panel) (Nomenclature 'Ref.' are the results by Ghia et al. 1982).

### 5.3.3. Flow past a stationary sphere at different Reynolds numbers

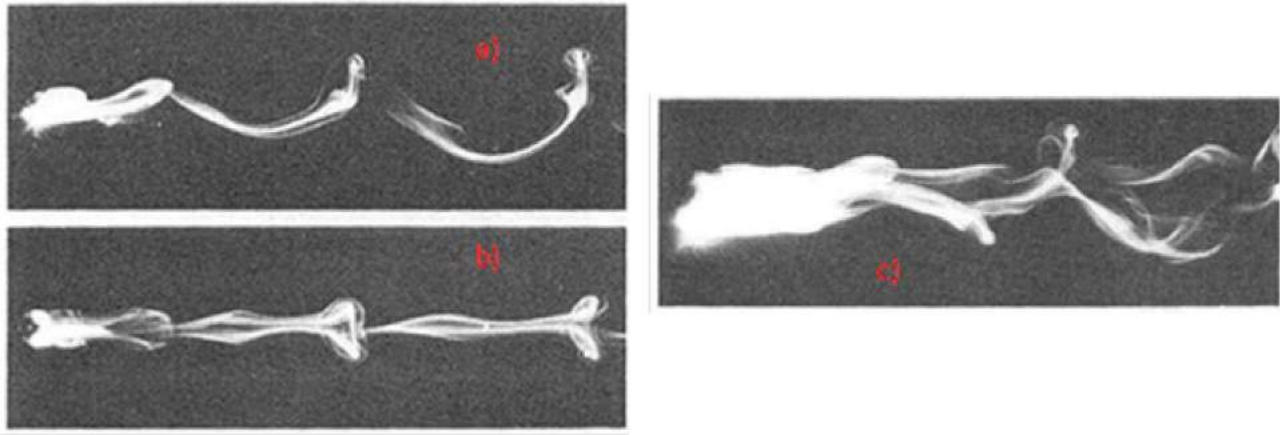
In the past few decades, a plethora of experimental, theoretical and numerical studies of viscous flow past a stationary sphere  $S$  have been presented to investigate wake structures. The Reynolds number, defined as

$$Re = \frac{U_0 D_s}{\nu} \quad (84)$$

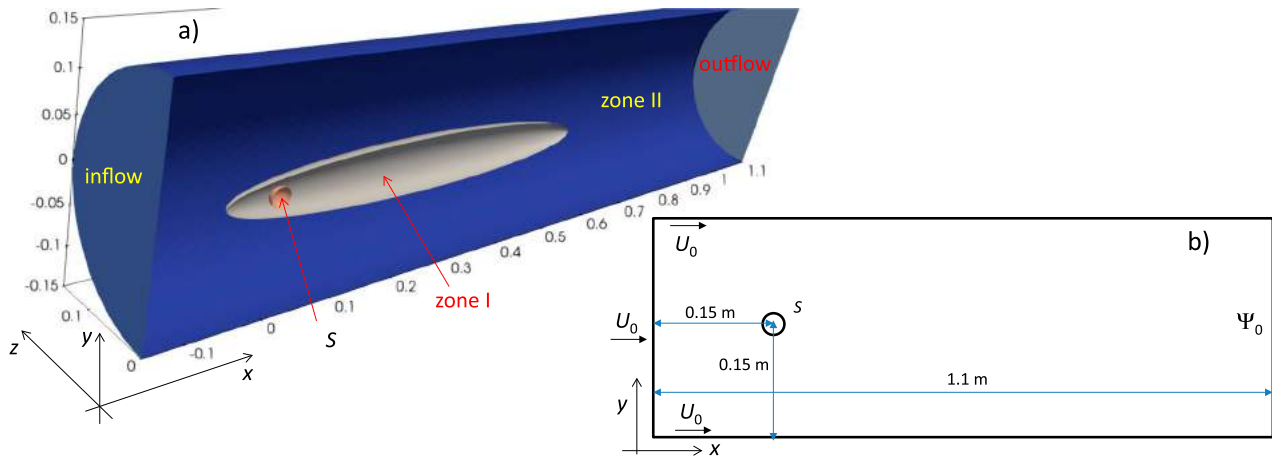
based on the uniform undisturbed flow velocity  $U_0$ , on the diameter of the sphere  $D_s$ , and on the surrounding

fluid viscosity  $\nu$ , is used as a parameter to classify the wake structure (e.g. Johnson & Patel, 1999; Ploumhans et al., 2002; Sakamoto & Haniu, 1990, and cited references). The wake structure has been a strongly debated topic, and several controversial findings have been obtained by authors in the literature studies (e.g. Johnson & Patel, 1999; Ploumhans et al., 2002; Sakamoto & Haniu, 1990, and cited references). According to experimental and numerical studies, above  $270 < Re < 290$ , the flow becomes unsteady but periodic, and vortex shedding starts around  $Re = 300$  (e.g. Johnson & Patel, 1999;





**Figure 13.** Test 3. Pattern of vortex shedding. Left panels  $Re = 300$ , (a) side view, (b) upper view. Right panel (c)  $480 < Re < 800$  (from Sakamoto & Haniu 1990).



**Figure 14.** Test 3. (a) 3D view of the domain. (b) setup and BCs of the numerical runs.

Ploumhans et al., 2002; Sakamoto & Haniu, 1990, and cited references), with formation of hairpin vortices (as shown in Figure 13, from the experimental studies by Sakamoto and Haniu (1990)). By progressively increasing  $Re$ , the vortices start to intertwine with each other, and above  $Re = 500$  periodicity is lost (e.g. Johnson & Patel, 1999; Ploumhans et al., 2002; Sakamoto & Haniu, 1990, and cited references). For a more comprehensive review, we refer the readers to the cited works.

In the research referred to here we investigated the flow structures around a stationary sphere for  $Re = 300$  and  $600$ . The fluid is assumed to be water. In Figure 14 we plot the 3D view of the domain and the setup of the numerical test. We assume the sphere  $S$ , with  $D_s = 0.0254$  m, symmetrically placed inside a large cylinder  $C$  with diameter  $D_C = 0.3$  m ( $\simeq 12 D_s$ ) and length  $1.1$  m ( $\simeq 43.5 D_s$ ). The center of  $S$  is located  $0.15$  m ( $\simeq 6 D_s$ ) downstream of the inflow upstream face of  $C$ . At the upstream inflow section of  $C$  we set  $\mathbf{u}$  uniform and constant in time  $(U_0, 0, 0)$ , and the same velocity is

imposed at the lateral walls of  $C$  in such a way that the flow around  $S$  is only weakly affected by the walls. Zero pressure is assumed at the downstream outflow section. Over the surface of  $S$  we assume no-slip BC. We assume flow at rest and zero pressure inside  $C$  at  $t = 0$ .

The mesh size is refined in zone I (around  $S$  and downstream of it, as shown in Figure 14(a)) in order to reproduce the strong velocity gradients close to the surface of  $S$  and the fluid vortices in the wake. A larger mesh size is adopted for the rest of the domain (zone II in Figure 14(a)). The mesh size (defined as in test 1) used for zones I and II is  $0.00055$  and  $0.0128$  m, respectively, with a smooth transition between the two zones. The total number of tetrahedrons in the mesh is  $2,309,771$ , with  $2,243,199$  clusters and  $392,850$  nodes, and the surface of  $S$  is discretized with  $4332$  triangles.

Unfortunately, description of initiation of vortex shedding is most often ignored in numerical studies. In physical experiments, initiation of vortex shedding is generated by flow instabilities amplifying small flow

disturbances (Sungsu, 2000). Such disturbances include, among others, asymmetric domain geometry, vibrations of the pipe, non-uniformity and turbulence of the inflow velocity, non-uniform roughness of the pipe walls and sphere surface, ... (Sungsu, 2000). All these sources are missing in numerical experiments, and, for a stationary sphere inside a symmetric domain, symmetric steady solution are attained even for Reynolds numbers at which experimental unsteady vortex shedding has been detected (Sungsu, 2000). In numerical experiments, vortex shedding could be generated by (1) computational truncation and round-off errors, strongly dependent on the characteristics of the numerical solver and the computer, or (2) specifically introduced numerical perturbations (Sungsu, 2000). Examples of such numerical perturbations can be found, for example, in Ploumhans et al. (2002), Sungsu (2000) and cited references.

We performed a first series of simulations at  $Re = 300$  without numerical perturbations, with an impulsive start of the inflow velocity. After rapid changes during the early stages of the process, the flow characteristics converted to a stationary solution. In Figure 15 we plot the streamlines of the stationary flow field in the  $(x-y)$  plane. After the early transient process due to the impulsive flow start, we computed the stationary values of drag and lift coefficients  $C_D$  and  $C_L$  listed in Table 5. These were obtained as  $C_D = F_x/(1/2\rho U_0\pi D_S^2/4)$  and

$C_L = F_y/(1/2\rho U_0\pi D_S^2/4)$ , where  $F_x$  and  $F_y$  are the  $x$  and  $y$  components of the total force, sum of the pressure and the viscous forces. The streamlines in the  $(x-z)$  plane are almost symmetrical, and the mean-in-time value of the side coefficient  $C_S$ , obtained as  $C_S = F_z/(1/2\rho U_0\pi D_S^2/4)$ , is  $1.34e-05$ .

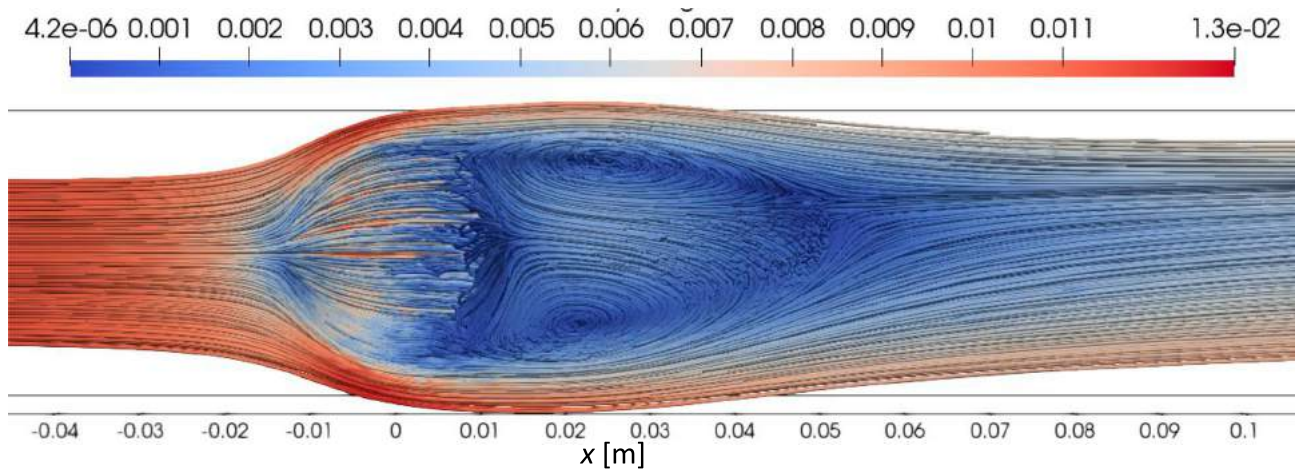
It is important to underline that  $C_D$  and  $C_L$  are in very good agreement with the mean in time values provided by literature studies, which are reported in Table 5.

Similarly to Ploumhans et al. (2002), the most efficient method for the present solver to trigger vortices has been, after the impulsive flow start, to set the  $y$  velocity component

$$v = \sin(\pi(\tau^* - 3)) \frac{U_0}{4} \text{ with } \tau^* = tU_0/D_S \quad (85)$$

in the interval  $3 \leq \tau^* \leq 4$ , along the inflow section and the lateral walls of  $C$ . The time step size used for the simulations is 0.05 s and the maximum computed value of the CFL number is 3.5.

In Figure 16 we plot the 3D view of the vorticity structures identified by the  $Q$ -criterion (Hunt et al., 1988). In Figures 17 and 18 we show the velocity streamlines and the iso-contour lines of the kinematic pressure in the  $(x-y)$  and  $(x-z)$  planes, respectively. The time difference among the panels is one quarter of a period, and after the fourth panel (3/4 of period), the cycle repeats again.

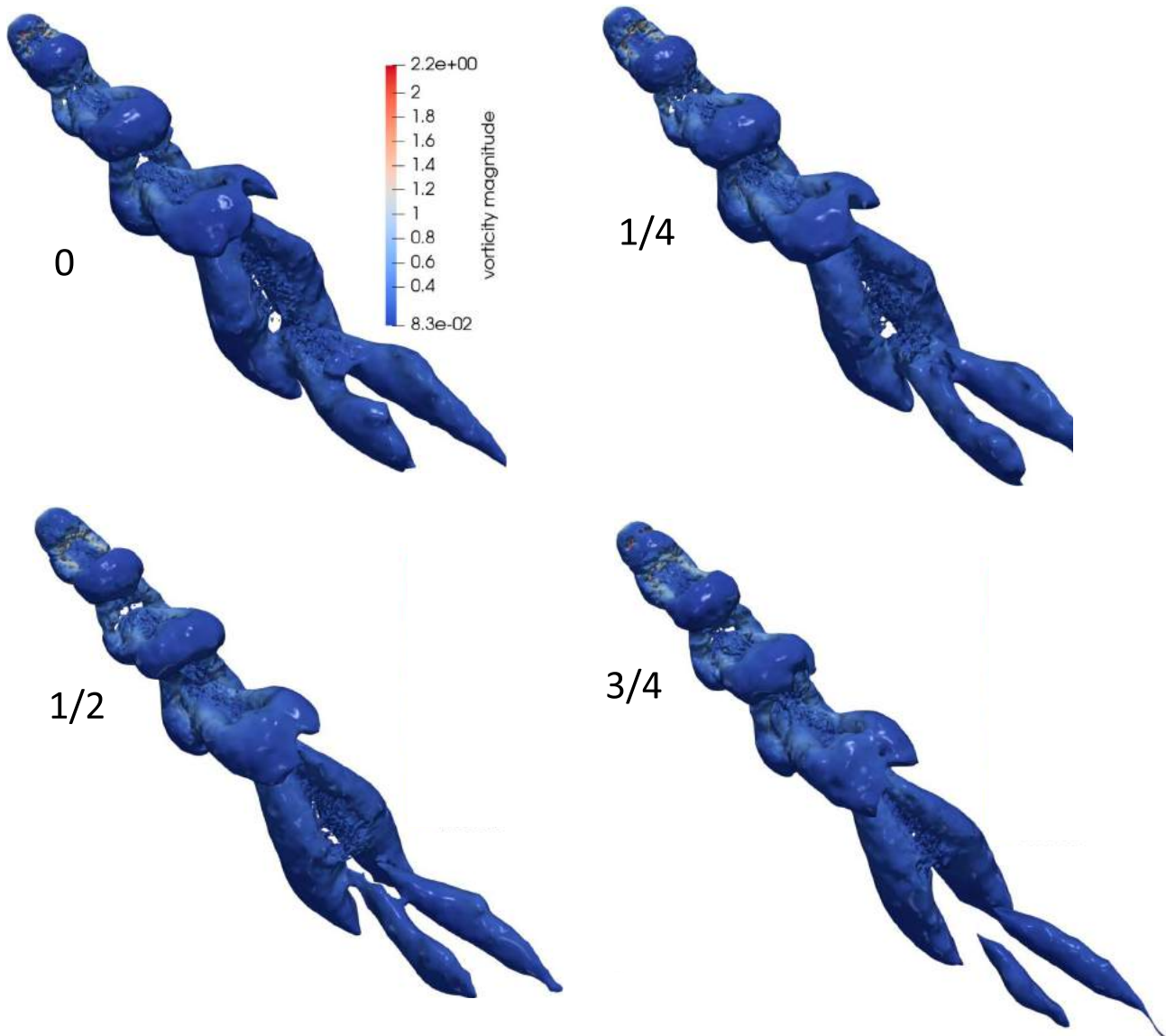


**Figure 15.** Test 3,  $Re = 300$ . Velocity streamlines in the  $(x-y)$  plane, stationary case.

**Table 5.** Test 3. Values of the drag and lift coefficients.

	$C_D$ [-]		$C_L$ [-]	
	Mean value	Amplitude value	Mean value	Amplitude value
Present solver without perturbation	0.67043	–	–0.0675	–
Present solver with perturbation (Equation (85))	0.6678	2.91E-03	–0.06061	1.47E-02
Ploumhans et al. (2002)	0.683	2.50E-03	–0.061	1.40E-02
Johnson and Patel (1999)	0.656	3.50E-03	–0.069	1.60E-02
Tomboulides (1993)	0.671	2.80E-03	not reported	not reported
Roos and Willmarth (1971)	0.629	not reported	not reported	not reported





**Figure 16.** Test 3,  $Re = 300$ . 3D periodic time evolution of the vortical structures.

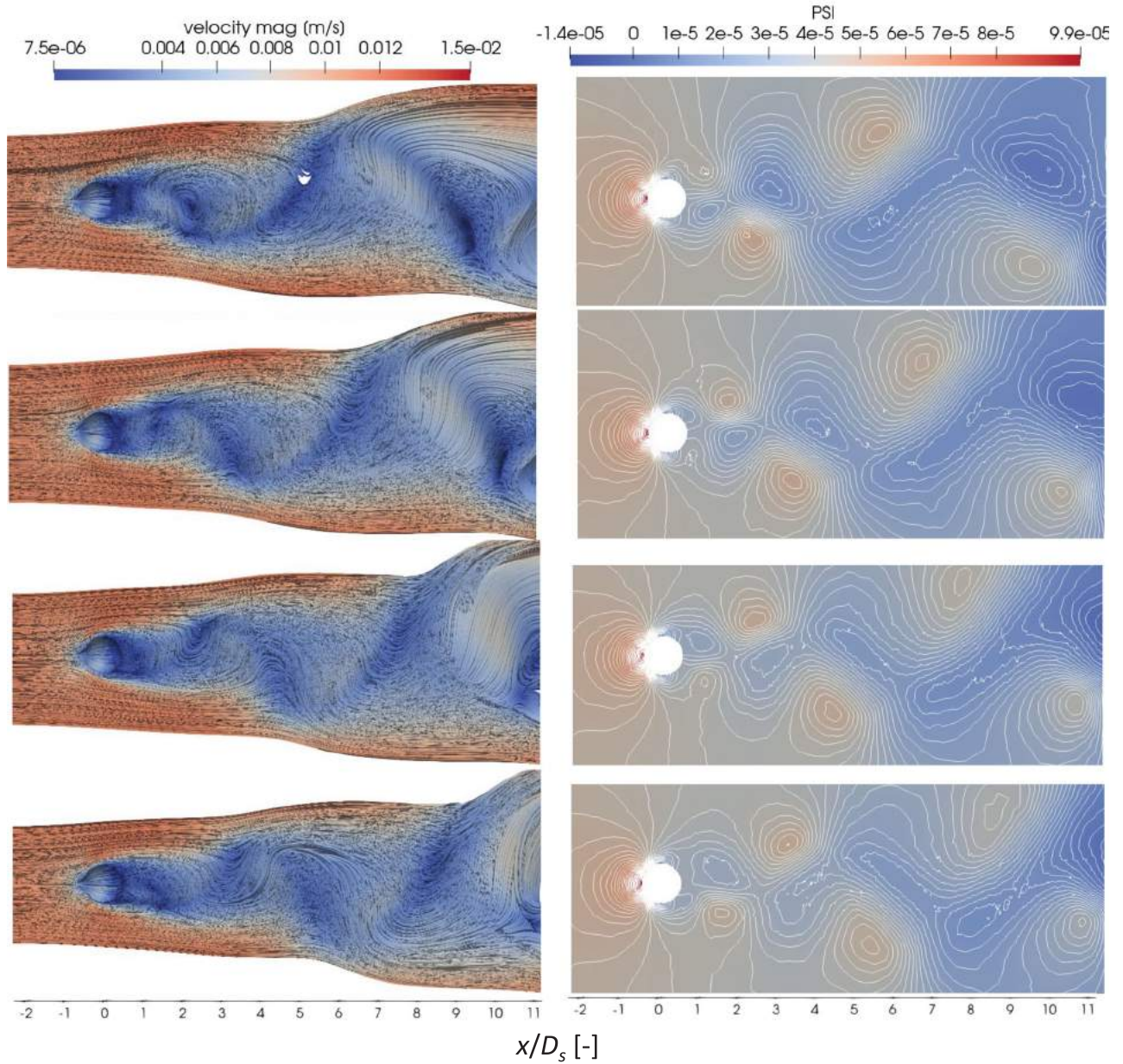
The present model satisfactorily reproduces the hairpin shapes of the vortical structure. Observe that, due to the strong pressure gradients feeding the movement of the vortices, the pressure minima in this case do not match their foci.

After a rapid transient phase, due to the impulsive start and to the imposed numerical perturbation, the time evolution of the  $C_D$  and  $C_L$  coefficients becomes periodic in time. The mean and amplitude values are listed in Table 5, and compared with the results provided by other literature works.

We also simulate the case with  $Re = 600$ . According to experimental observations, the shedding vortices become irregular for  $Re > 480$  (Sakamoto & Haniu, 1990). In Figure 13(c) we have previously shown the pattern of vortex shedding for  $480 < Re < 800$  (from

Sakamoto & Haniu, 1990). The setting of the ICs and BCs is the same as for  $Re = 300$ , as is the impulsive flow start. In this case, perturbation for generating vortex shedding was not necessary. The time step size used for the numerical runs was 0.025 s and the maximum value of the CFL number attained during the simulations was 3.12.

In Figure 19(a) we show the 3D vortical structure at  $\tau^* = 35$ , where  $\tau^*$  is defined in Equation (85) and the irregularity of the hairpin vortices and their intertwining with each other is evident, as experimentally observed by Sakamoto and Haniu (1990). In Figure 19(b) we plot the time histories of the drag lift and side coefficients. As expected, we lost the periodic trend of  $C_D$  and  $C_L$  observed for  $Re = 300$ , and the value of the side coefficient is of the same order as  $C_L$ .



**Figure 17.** Test 3,  $Re = 300$ . Periodic time evolution of the velocity streamlines (left panels) and kinematic pressure (right panels), ( $x$ - $z$ ) plane.

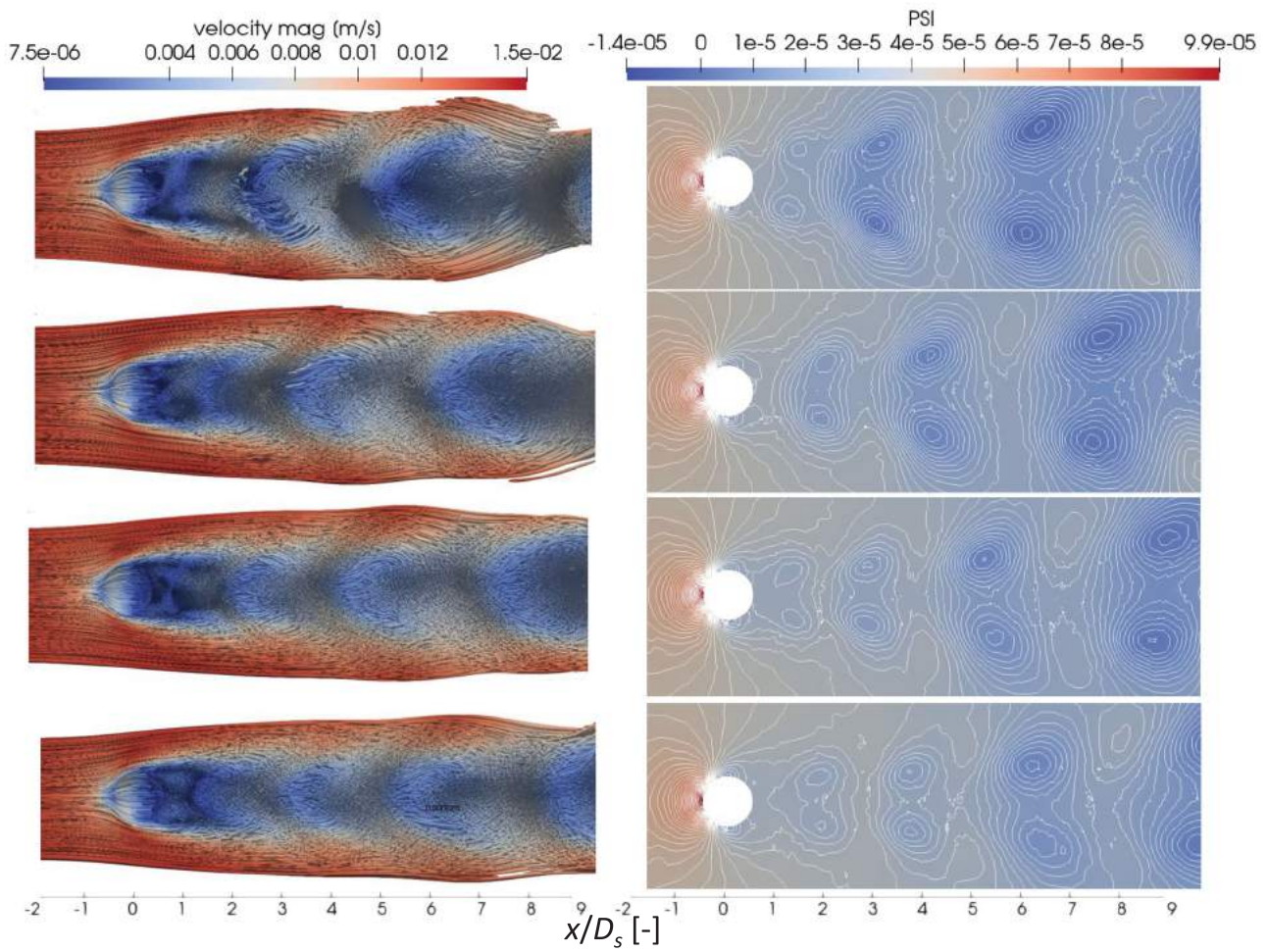
As mentioned in section 3.1.1, for the case of  $Re = 300$ , we estimate  $N_{r,k}$  and predict the corresponding computational time of parallelization of the MAST algorithm assuming different  $N_p$  number of processors. The maximum  $N_{r,k}$  is 3847. Neglecting the time  $\epsilon$  and the other parallelization costs, the MAST solution time  $T_{MAST}$  would be equal to

$$T_{MAST} = \sum_{i=1, N_{iter}} \left( \sum_{r=1, N_{R,k}} \left( \text{Int} \left( \frac{N_{r,k}}{N_p} \right) + 1 \right) \right) 2T_T \quad (86)$$

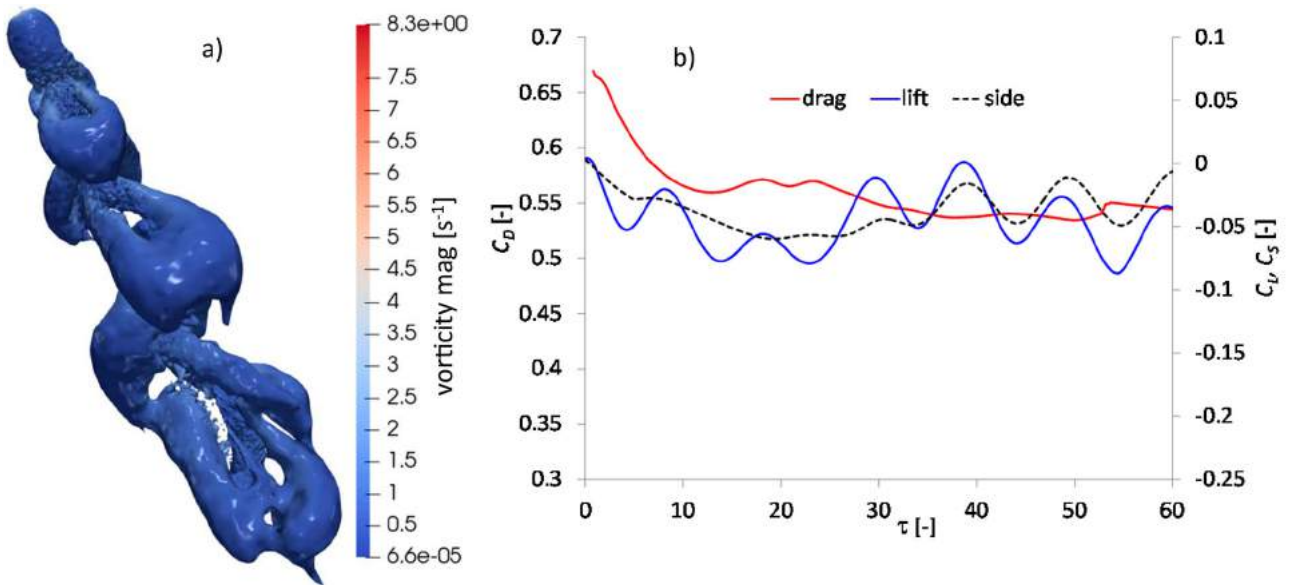
where  $N_{iter}$  is the number of time steps and  $N_{R,k}$  is the number of ranks at iteration  $k$ . This time is of course larger than the time  $T_{min}$  computed assuming all available processors working together, because  $N_{r,i}$  can be smaller than  $N_p$  and the ratio between  $T_{MAST}$  and  $T_{min}$  grows along with  $N_p$ .

See in Table 6 the ratios  $T_{MAST} / T_T$  and  $T_{MAST} / T_{min}$  computed in the simulation of test 3, with a mesh of 2,309,771 tetrahedrons, assuming a number of processors in the range 4-250, at the present time corresponding to small-medium workstations. You can observe that the ratio  $T_{MAST} / T_{min}$  attains a maximum value of 1.4 with





**Figure 18.** Test 3,  $Re = 300$ . Periodic time evolution of the velocity streamlines (left panels) and kinematic pressure (right panels),  $(x-y)$  plane.



**Figure 19.** Test 3,  $Re = 600$ . (a) 3D vortical structure at  $\tau^* = 35$ . (b) Time evolution of the drag, lift and side coefficients.

**Table 6.** MAST solution time vs. no. of processors.

$N_p$	$T_{\min}/T_T$	$T_{\text{MAST}}/T_T$	$T_{\text{MAST}}/T_{\min}$
4	676,184,582	680,355,106	1.006
12	225,394,080	229,004,379	1.016
40	67,618,224	71,041,472	1.051
50	54,094,345	57,521,651	1.063
100	27,046,587	30,304,361	1.120
150	18,031,058	20,938,102	1.161
200	13,522,708	16,347,121	1.209
250	10,818,869	15,173,010	1.402

the maximum number of 250 processors. This means that parallelization should work very well with this type of very popular computers even with large size problems.

#### 5.3.4. Simulation of blood flow inside aneurism

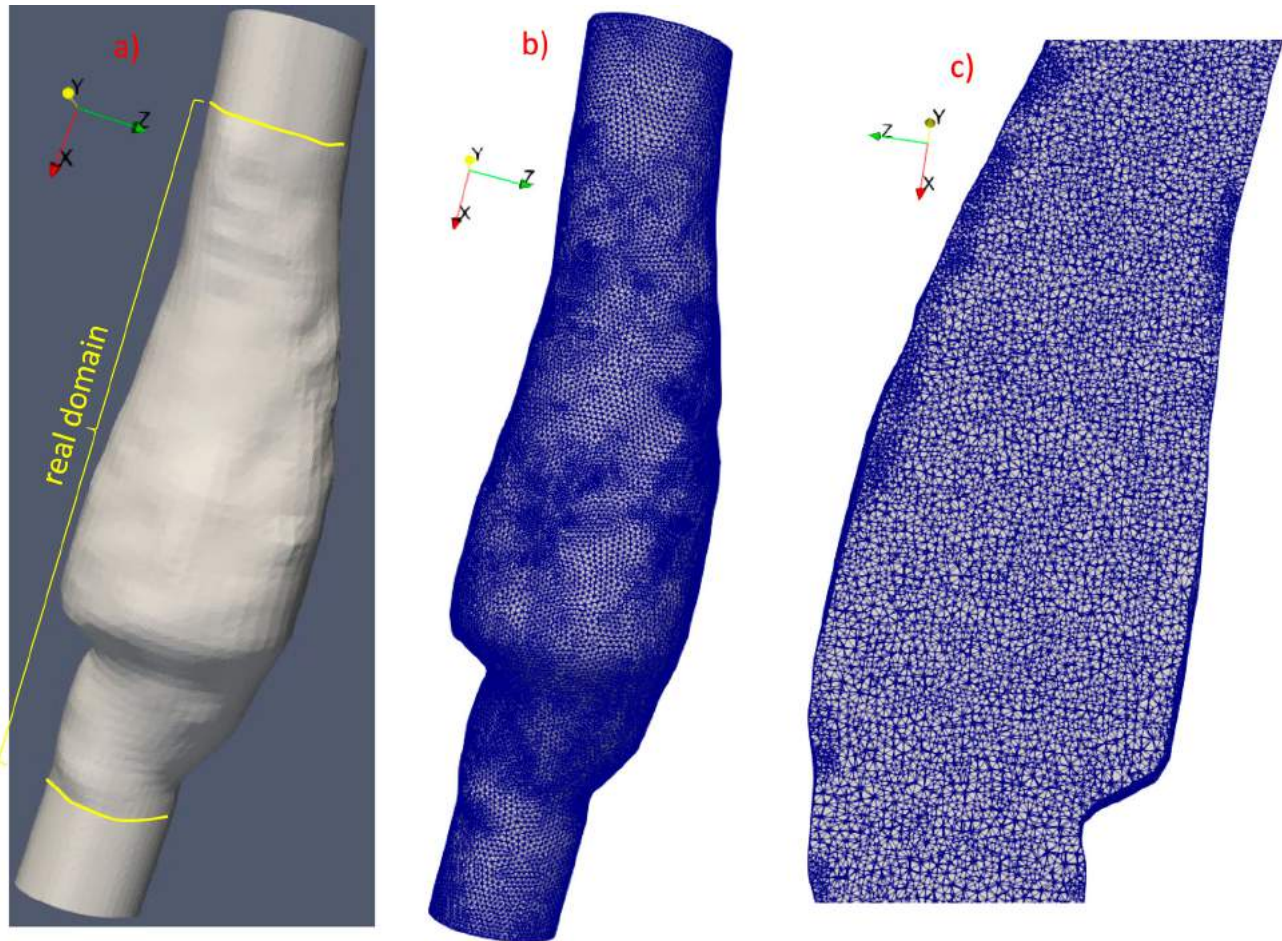
In this test, we simulated the hemodynamic flow conditions inside a real abdominal aorta affected by a large aneurysm without thrombus in the lumen. The computational domain was computed starting from the kinematic field of a real (44-year-old female) patient-specific aortic wall, obtained from the data recorded by an electrocardiogram-gated computer tomography

angiography (CTA) during a stabilized cardiac cycle, as described in (Aricò et al., 2020 and cited references).

Besides the CTA images, additional input data were the measurements, in a resting condition, of pressure (on the left arm) and aorta volumetric flow rate (in the carotid artery), during a stabilized cardiac cycle (Aricò et al., 2020). The cardiac cycle  $T_c$  of the patient was 0.83333 s.

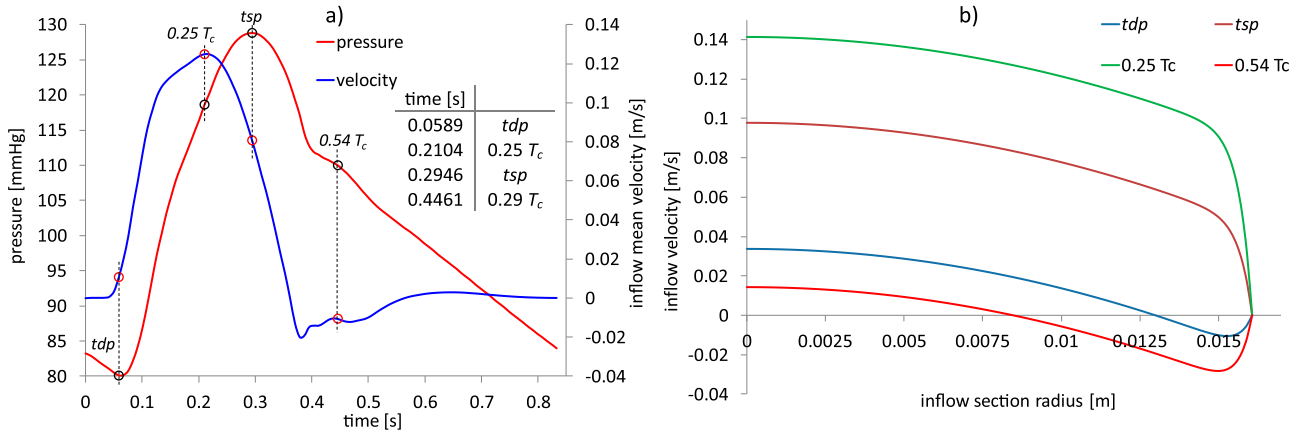
The real aortic segment was approximately 0.16 m long (see Figure 20(a)). The computational domain was extended with respect to the real one by means of two transition stretches, and the real cross-sections were linearly morphed into circles of equivalent radii  $r = \sqrt{A/\pi}$  (Figure 20(a)). The diameters of the inflow and outflow artificial sections,  $D_i$  and  $D_o$ , are 0.03213 and 0.0256 m, respectively, and the two stretches were approximately  $D_i$  long.

For the numerical model simulations, we set an inflow velocity profile and a uniform spatial pressure distribution along the upstream and downstream sections of the computational domain, respectively. The BCs of the present model were obtained as described in Aricò et al. (2020) and cited references.

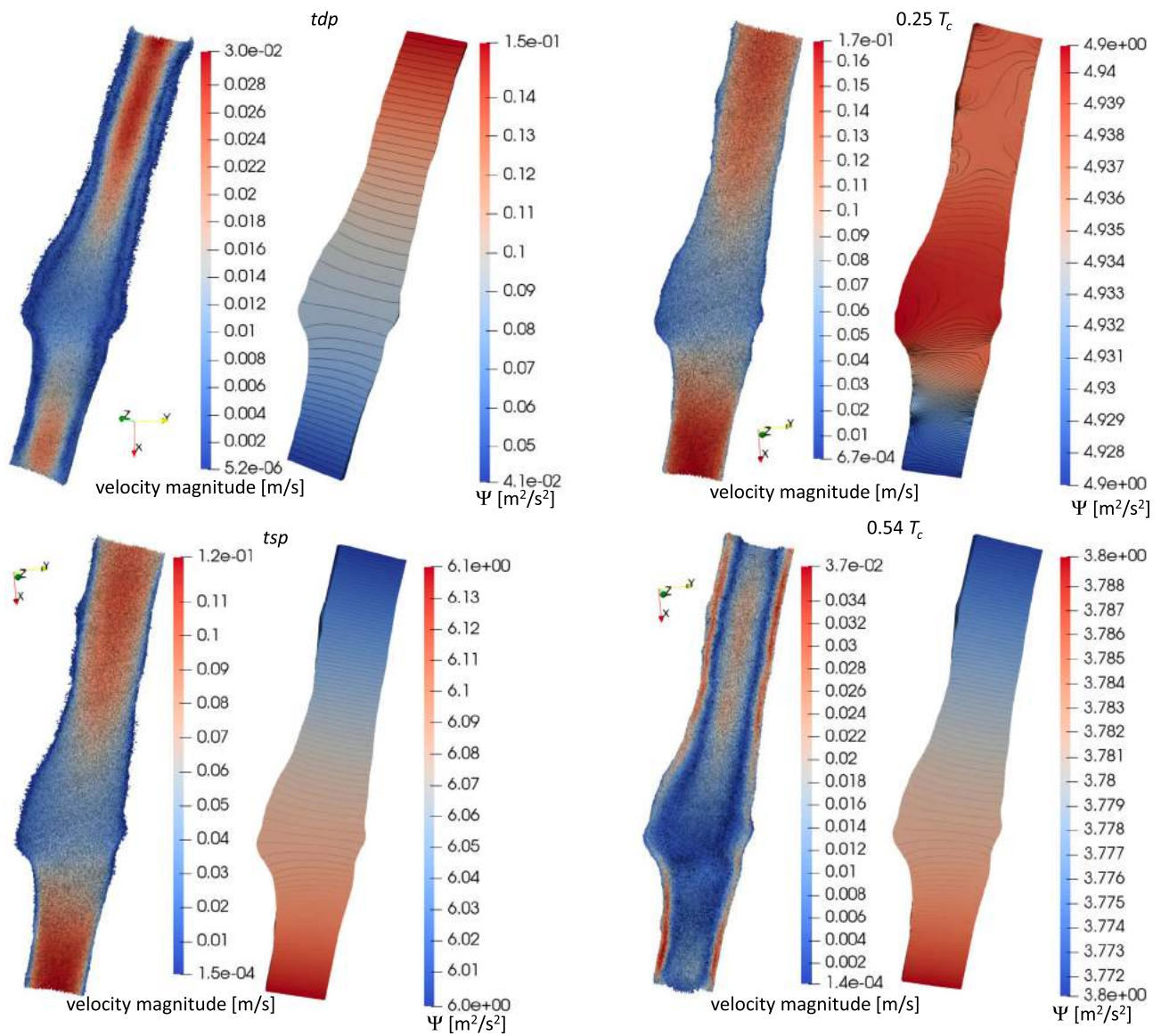


**Figure 20.** (a) Test 4. Real and computational domain. (b) computational mesh. (c) Section of the mesh with a generic plane.





**Figure 21.** Test 4. (a) waveforms of mean-in-time inflow velocity and outlet pressure. (b) Womersley inflow velocity profiles at four significant times (from Aricò et al., 2020).



**Figure 22.** Test 4. Computed velocity vectors and kinematic pressure at the significant times in Figure 21.



In Figure 21(a) we plot the waveforms of the inflow velocity (obtained by dividing the waveform of the flow rate by the area of the inflow cross-section), and the outlet pressure. The ‘time of the diastolic (systolic) pressure’ is the time corresponding to the minimal (maximal) aortic pressure –  $tdp$  ( $tsp$ ) in Figure 21.  $tdp$  and  $tsp$  were computed to be 0.0589 and 0.2946 s after the start of the cycle (Nagy et al., 2015).

A Womersley velocity profile of the pulsatile flow (Womersley, 1955) was analytically computed for the diameter  $D_i$ , as described in Aricò et al. (2020), and assigned to the upstream boundary of the computational domain. The Womersley number  $\alpha$  can be regarded as the ratio between the unsteady inertial forces and the viscous forces, and it is defined as (Womersley, 1955)

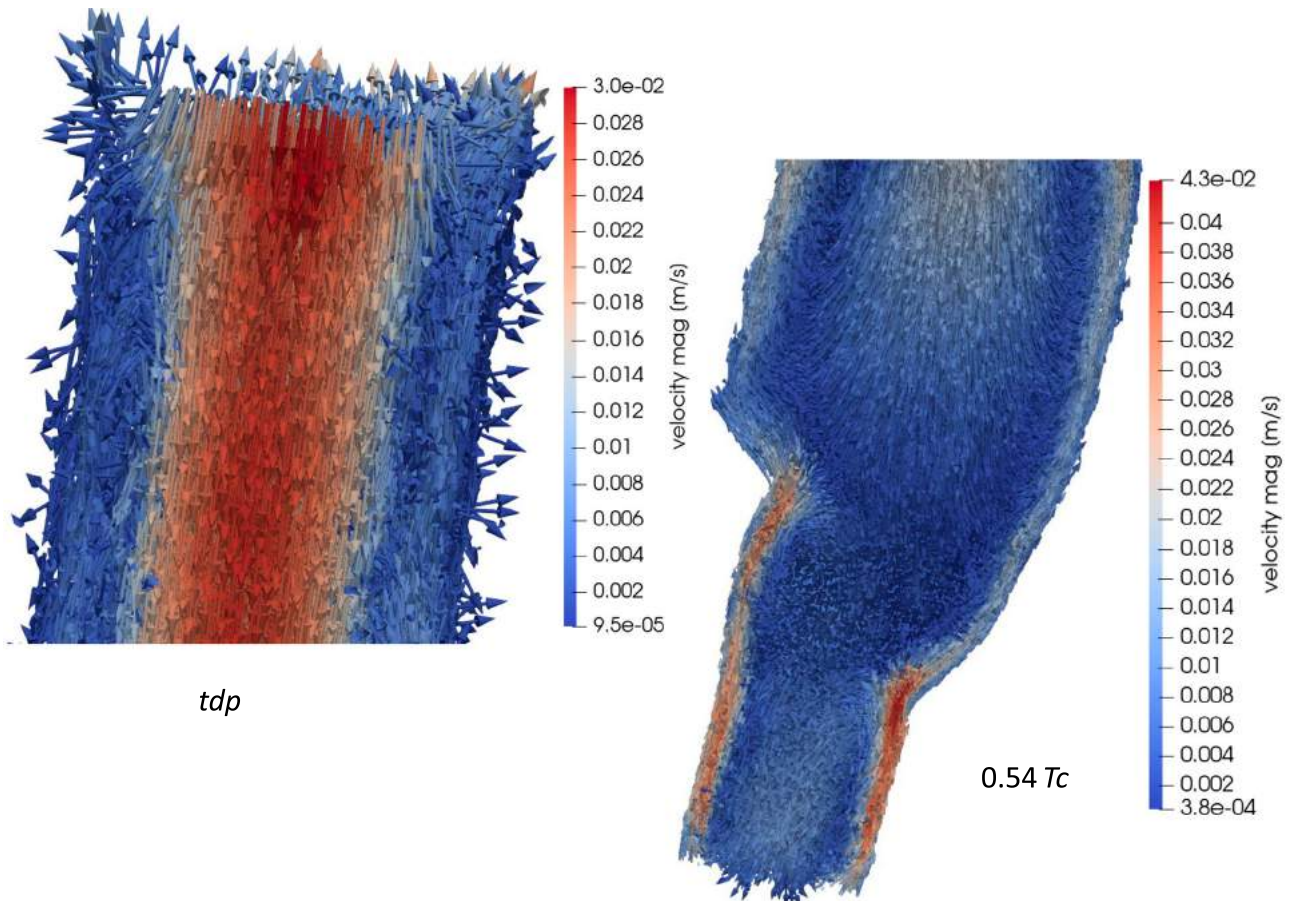
$$\alpha = R\sqrt{\frac{\omega}{\nu}} \text{ with } \omega = 1/T_c \quad (87)$$

where  $R$  is the radius of the vessel at the boundary section. We set  $\nu = 3.77 \times 10^{-6} \text{ m}^2/\text{s}$  (blood kinematic viscosity). The original Poiseuille velocity profile is flattened proportionally to the  $\alpha$  number. In the present case  $\alpha$  is around 24, and Figure 21(b) shows the profiles computed for the significant times listed in the table of Figure 21(a).

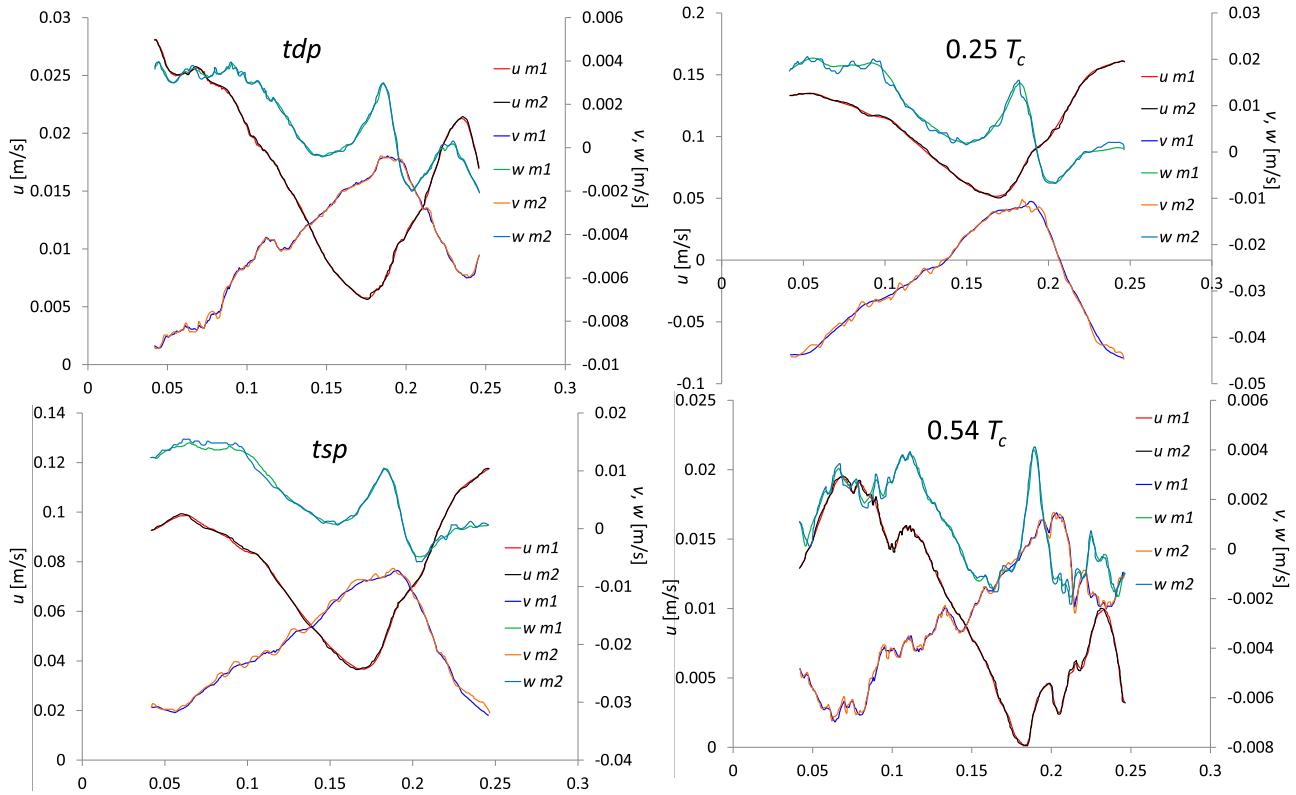
The maximum value of the Reynolds number attained during the simulations was approximately 1200, which implies a fully laminar flow.

The computational domain corresponds to the fixed-in-time geometry computed, at the  $tsp$  time, applying the procedure proposed in Nagy et al. (2015). The computational mesh had 790,346 tetrahedrons and 146,995 nodes, and the number of cluster is 729,038. The mesh size ranged from  $1.e-04 \text{ m}$  to  $1e-03 \text{ m}$ . In Figure 20(b,c) we show the mesh and a zoom of a cutting generic plane. The time step size was 0.01 s and the maximum value of the CFL number computed during the simulations was 3.45.

In Figure 22, for the significant times listed in Figure 21(b), we show the computed velocity and the kinematic pressure fields. The black arrow indicates the main upstream-downstream flow direction. At  $tdp$ , the pressure gradient is oriented according to the main upstream-downstream direction. The velocity profile in the upstream portion of the studied reach is almost uniform along the radial vessel direction, and, close to the walls, recirculating flow zones arise (see the zoom in Figure 23). These recirculating flows could be generated by the inflow velocity computed in the most lateral part



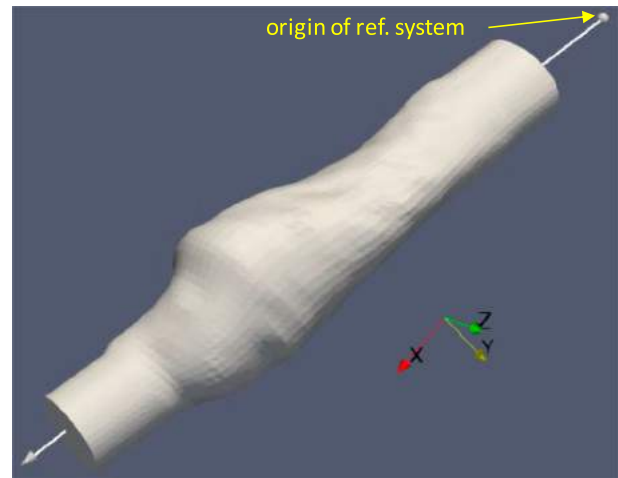
**Figure 23.** Test 4. Zoom of the velocity fields for  $tdp$  and  $0.54 T_c$ .



**Figure 24.** Test 4. Velocity components computed along the axis shown in Figure 25, for two meshes.

of the Womersley profile (see Figure 21(b)). The blood flow decelerates in the central region of the aorta, due to the enlargement of the vessel because of the aneurysm, and accelerates downstream of the aneurysm, due to the reduction of the section of the vessel. Due to the reduction of the inflow velocity assigned at the inflow section, from time  $0.25 T_c$  to  $tsp$ , and the corresponding increase of the assigned outlet pressure in the same time interval (see Figure 21), the pressure gradient along the principal flow direction changes sign, becoming downstream-upstream oriented. The recirculating flow velocity zones close to the lateral walls then disappear, since the boundary Womersley velocities assigned at the inflow section are inward oriented. At time  $0.54 T_c$ , the flow recirculation close to the vessel walls is stronger than at  $tdp$ . The size of the portion of the inflow section where the leaving Womersley velocities are set is similar to the one where the incoming velocities are assigned, and their norm is comparable to (or greater than) the values of the inflow velocity norm (see Figure 21(b)). Vorticities also arise in the central and downstream portions of the aortic vessel, as shown in the zoom of the velocity vector (Figure 22).

In Figure 24 we plot the values of the velocity components computed along the axis of the aneurism, shown in Figure 25, at four different times. The origin of the coordinates along the axis is also shown in Figure 25. These results are marked as ‘ $u (v, w) m1$ ’. In Figure 24,



**Figure 25.** Test 4. Initial and final axis position.

we also superimpose the results obtained using a refined mesh, with 3,361,925 tetrahedrons, 599,988 nodes, and 3,116,998 clusters, marked as ‘ $u (v, w) m2$ ’. The scatters between the two solutions are almost negligible.

## 6. Conclusions

A new algorithm for the numerical solution of the 3D Navier–Stokes equations for incompressible fluids has been presented and validated with synthetic tests. The algorithm is radically new and is based on the

Raviart-Thomas first order spatial discretization of pressure and velocity. The convective terms were solved using the Marching in Space and Time (MAST) technique, previously applied only to groundwater transport and shallow water problems. The algorithm has the following merits with respect to the many other competitors: (1) It can be applied to any tetrahedral, unstructured and non-Delaunay mesh, generated inside irregular boundaries; (2) it fully preserves mass conservation, (3) the time step size is not constrained by the CFL limit, (4) the CPU time required for the solution of a single time step, with a single physical processor, grows with a  $\beta$  power almost equal to 1.1, due to the solution of linear systems with matrices that are always positive-definite, holding the  $M$ -property, factorized only once at the beginning of the first time step.

Code parallelization and introduction of a turbulence model to solve the turbulence structures within the element scale would make MAST-RT0 model ready to solve also many other problems of great interest, like Fluid-Structure Interactions (FSI), flows around moving boundaries (e.g. swimming fishes) (e.g. Salih et al., 2019), sloshing tanks (e.g. Ghalandari et al., 2019) and many others.

## Notes

1. [www.hsl.rl.ac.uk/catalogue/kb07.html](http://www.hsl.rl.ac.uk/catalogue/kb07.html), 1962.
2. [hsl.rl.ac.uk/catalogue/mi21.html](http://hsl.rl.ac.uk/catalogue/mi21.html), 2013.
3. [www.qhull.org/html/index.htm](http://www.qhull.org/html/index.htm), 2020, and cited references.
4. [www.paraview.org](http://www.paraview.org), 2020.

## Disclosure statement

No potential conflict of interest was reported by the author(s).

## ORCID

Costanza Aricò  <http://orcid.org/0000-0001-7173-0981>

Marco Sinagra  <http://orcid.org/0000-0002-6160-6941>

Calogero Picone  <http://orcid.org/0000-0003-4256-6689>

Tullio Tucciarelli  <http://orcid.org/0000-0002-2131-6659>

## References

- Aguerre, H. J., Venier, C. M., Pairetti, C. I., Márquez Damián, S., & Nigro, N. M. (2020). A SIMPLE-based algorithm with enhanced velocity corrections: The COMPLEX method. *Computers & Fluids*, 198(15), Article 104396. <https://doi.org/10.1016/j.compfluid.2019.104396>
- Aricò, C., Nasello, C., & Tucciarelli, T. (2007). A marching in space and time (MAST) solver of the shallow water equations. Part II: The 2D model. *Advances in Water Resources*, 30(5), 1253–1271. <https://doi.org/10.1016/j.advwatres.2006.11.004>
- Aricò, C., Sinagra, M., Begnudelli, L., & Tucciarelli, T. (2011). MAST-2D diffusive model for flood prediction on domains with triangular Delaunay unstructured meshes. *Advances in Water Resources*, 34(11), 1427–1449. <https://doi.org/10.1016/j.advwatres.2011.08.002>
- Aricò, C., Sinagra, M., Nagy, R., Napoli, E., & Tucciarelli, T. (2020). Investigation of the hemodynamic flow conditions and blood-induced stresses inside an abdominal aortic aneurysm by means of a SPH numerical model. *International Journal for Numerical Methods in Biomedical Engineering*, 36, 3263. <https://doi.org/10.1002/cnm.3263>
- Aricò, C., Sinagra, M., & Tucciarelli, T. (2012). The MAST-edge centred lumped scheme for the flow simulation in variably saturated heterogeneous porous media. *Journal of Computational Physics*, 231(4), 1387–1425. <https://doi.org/10.1016/j.jcp.2011.10.012>
- Aricò, C., Sinagra, M., & Tucciarelli, T. (2013a). Monotonic solution of flow and transport problems in heterogeneous media using Delaunay unstructured triangular meshes. *Advances in Water Resources*, 52(4), 132–150. <https://doi.org/10.1016/j.advwatres.2012.09.006>
- Aricò, C., Sinagra, M., & Tucciarelli, T. (2013b). Anisotropic potential of velocity fields in real fluids: Application to the MAST solution of shallow water equations. *Advances in Water Resources*, 62, 13–36. <https://doi.org/10.1016/j.advwatres.2013.09.010>
- Aricò, C., & Tucciarelli, T. (2007a). MAST solution of advection problems in irrotational flow fields. *Advances in Water Resources*, 30(3), 665–685. <https://doi.org/10.1016/j.advwatres.2006.03.007>
- Aricò, C., & Tucciarelli, T. (2007b). A marching in space and time (MAST) solver of the shallow water equations. Part I: The 1D model. *Advances in Water Resources*, 30(5), 1236–1252. <https://doi.org/10.1016/j.advwatres.2006.11.003>
- Aricò, C., & Tucciarelli, T. (2009). The MAST-FV/FE scheme for the simulation of thermohaline processes in variable density saturated porous media. *Journal of Computational Physics*, 228(4), 1234–1274. <https://doi.org/10.1016/j.jcp.2008.10.015>
- Aricò, C., & Tucciarelli, T. (2013). Monotonic solution of heterogeneous anisotropic diffusion problems. *Journal of Computational Physics*, 252, 219–249. <https://doi.org/10.1016/j.jcp.2013.06.017>
- Auricchio, F., Beirão da Veiga, L., Brezzi, F., & Lovadina, C. (2017). Mixed finite element methods. In *Encyclopedia of computational mechanics*. 2nd ed. (pp. 1–53). <https://doi.org/10.1002/9781119176817.ecm2004>
- Bassi, F., & Rebay, S. (1997). A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 131, 267–279. <https://doi.org/10.1006/jcph.1996.5572>
- Bazilevs, Y., Takizawa, K., & Tezduyar, T. E. (2013). *Computational fluid-structure interaction: Methods and applications*. Wiley. <https://doi.org/10.1002/9781118483565>
- Beirão da Veiga, L., Lovadina, C., & Vacca, G. (2018). Virtual elements for the Navier-Stokes problems on polygonal meshes. *SIAM Journal on Numerical Analysis*, 56(3), 1210–1242. <https://doi.org/10.1137/17M1132811>
- Boppana, V. B. L., & Gajjar, J. S. B. (2010). Global flow instability in a lid-driven cavity. *International Journal for Numerical Methods in Fluids*, 62, 827–853. <https://doi.org/10.1002/flid.2040>
- Botella, O., & Peyret, R. (1998). Benchmark spectral results on the lid driven cavity flow. *Computers & Fluids*, 27, 421–433. [https://doi.org/10.1016/S0045-7930\(98\)00002-4](https://doi.org/10.1016/S0045-7930(98)00002-4)



- Braaten, M. E., & Shyy, W. (1986). Comparison of iterative and direct method for viscous flow, calculations in body-fitted coordinates. *International Journal for Numerical Methods in Fluids*, 6, 325–349. <https://doi.org/10.1002/flid.1650060603>
- Brankin, R. W., Gladwell, I., & Shampine, L. F. (1993). RKSUITE: A suite of explicit Runge-Kutta codes. *Computer Science*. [https://doi.org/10.1142/9789812798886\\_0004](https://doi.org/10.1142/9789812798886_0004)
- Busto, F., Ferrin, J. L., Toro, E. F., & Vazquez-Cendon, M. E. (2018). A projection hybrid high order finite volume/finite element method for incompressible turbulent flows. *Journal of Computational Physics*, 353, 169–192. <https://doi.org/10.1016/j.jcp.2017.10.004>
- Calhoun, D. (2002). A Cartesian grid method for solving the two-dimensional stream function-vorticity equations in irregular regions. *Journal of Computational Physics*, 176, 231–275. <https://doi.org/10.1006/jcph.2001.6970>
- Chorin, A. J. (1968). Numerical solution of the Navier-Stokes equations. *Mathematics of Computation*, 22, 745–762. <https://doi.org/10.2307/2004575>
- Dalal, A., Eswaran, V., & Biswas, G. (2008). A finite-volume method for Navier-Stokes equations on unstructured meshes. *Numerical Heat Transfer, Part B: Fundamentals*, 54(3), 238–259. <https://doi.org/10.1080/10407790802182653>
- Darwish, M., Abdel Aziz, A., & Moukalled, F. (2015). A coupled pressure-based finite-volume solver for incompressible two-phase flow. *Numerical Heat Transfer, Part B: Fundamentals*, 67, 47–74. <https://doi.org/10.1080/10407790.2014.949500>
- Darwish, M., Sraj, I., & Moukalled, F. (2009). A coupled finite volume solver for the solution of incompressible flows on unstructured grids. *Journal of Computational Physics*, 228, 180–201. <https://doi.org/10.1016/j.jcp.2008.08.027>
- Forsyth, P. A. (1991). A control volume finite element approach to NAPL groundwater contamination. *SIAM Journal on Scientific and Statistical Computing*, 12, 1029–1057. <https://doi.org/10.1137/0912055>
- Fortin, M. (1981). Old and new finite elements for incompressible flows. *International Journal for Numerical Methods in Fluids*, 1, 347–364. <https://doi.org/10.1002/flid.1650010406>
- Gao, W., Liu, R. X., & Li, H. (2012). A hybrid vertex-centered finite volume/element method for viscous incompressible flows on non-staggered unstructured meshes. *Acta Mechanica Sinica*, 28(2), 324–334. doi:10.1007/s10409-012-0038-2
- Ghalandari, M., Bornassi, S., Shamshirband, S., Mosavi, A., & Chau, K. W. (2019). Investigation of submerged structures' flexibility on sloshing frequency using a boundary element method and finite element analysis. *Engineering Applications of Computational Fluid Mechanics*, 13(1), 519–528. <https://doi.org/10.1080/19942060.2019.1619197>
- Ghia, U., Ghia, K. N., & Shin, C. T. (1982). High-resolutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics*, 48, 387–411. [https://doi.org/10.1016/0021-9991\(82\)90058-4](https://doi.org/10.1016/0021-9991(82)90058-4)
- Gingold, R. A., & Monaghan, J. J. (1977). Smoothed particle hydrodynamics-theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181, 375–389. <https://doi.org/10.1093/mnras/181.3.375>
- Gosman, A. D., Koosinlin, M. L., Lockwad, F. C., & Spalding, D. B. (1976). Transfer of heat in rotating systems. ASME M. 76-GT 25. <https://doi.org/10.1115/76-GT-25>
- Guermond, J. L., Mineev, P., & Shen, J. (2006). An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 195, 6011–6045. <https://doi.org/10.1016/j.cma.2005.10.010>
- Hanby, R. F., & Silvester, D. J. (1996). A comparison of coupled and segregated iterative solution techniques for incompressible swirling flow. *International Journal for Numerical Methods in Fluids*, 22, 353–373. [https://doi.org/10.1002/\(SICI\)1097-0363\(19960315\)22:5 < 353::AID-FLD327 > 3.0.CO; 2-Z](https://doi.org/10.1002/(SICI)1097-0363(19960315)22:5 < 353::AID-FLD327 > 3.0.CO; 2-Z)
- Hang, S. (2015). Tetgen, a Delaunay-based quality tetrahedral mesh generator. *ACM Transactions on Mathematical Software*, 41(2), 1. <https://doi.org/10.1145/2629697>
- Harlow, F. H., & Welch, J. E. (1965). Numerical calculation of three-dimensional time dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, 8(12), 2182–2189. <https://doi.org/10.1063/1.1761178>
- Hughes, T. J. R., Liu, W. K., & Brooks, A. (1979). Finite element analysis of incompressible viscous flows by the penalty function formulation. *Journal of Computational Physics*, 30(1), 1–60. [https://doi.org/10.1016/0021-9991\(79\)90086-X](https://doi.org/10.1016/0021-9991(79)90086-X)
- Hunt, J. C. R., Wray, A., & Moin, P. (1988). *Eddies, stream, and convergence zones in turbulent flows*. Center for Turbulence Research, Report CTR-S88.
- Joe, B. (1986). Delaunay triangular meshes in convex polygons. *SIAM Journal on Scientific and Statistical Computing*, 7, 514–539. <https://doi.org/10.1137/0907035>
- Johnson, T. A., & Patel, V. C. (1999). Flow past a sphere up to a Reynolds number of 300. *Journal of Fluid Mechanics*, 378, 19–70. <https://doi.org/10.1017/S0022112098003206>
- Kim, D., & Choi, H. (2000). A second-order time-accurate finite volume method for unsteady incompressible flow on hybrid unstructured grids. *Journal of Computational Physics*, 162, 411–428. <https://doi.org/10.1006/jcph.2000.6546>
- Kim, J., & Moin, P. (1985). Application of a fractional-step method to incompressible Navier-Stokes equations. *Journal of Computational Physics*, 59, 308–323. [https://doi.org/10.1016/0021-9991\(85\)90148-2](https://doi.org/10.1016/0021-9991(85)90148-2)
- Kuhlmann, H. C., & Romanò, F. (2019). The lid-driven cavity. In *Computational methods in applied sciences*, Vol. 50 (pp. 233–309). [https://doi.org/10.1007/978-3-319-91494-7\\_8](https://doi.org/10.1007/978-3-319-91494-7_8)
- Lehrenfeld, C., & Schöberl, J. (2016). High order exactly divergence-free hybrid discontinuous Galerkin methods for unsteady incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 307, 339–361. <https://doi.org/10.1016/j.cma.2016.04.025>
- Letniowski, F. W. (1992). Three-dimensional Delaunay triangulations for finite element approximations to a second-order diffusion operator. *SIAM Journal on Scientific and Statistical Computing*, 13, 765–770. <https://doi.org/10.1137/0913045>
- Li, X. Y., & Teng, S. H. (2001, January 7–9). Generating well-shaped Delaunay meshes in 3D. Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, Washington, DC, USA. Association for Computing Machinery. <https://doi.org/10.1145/365411.365416>
- Liu, X., & Chen, Z. (2019). The nonconforming virtual element method for the Navier-Stokes equations. *Advances in Computational Mathematics*, 45, 51–74. <https://doi.org/10.1007/s10444-018-9602-z>
- Lucy, L. B. (1977). A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, 82(12), 1013–1024. <https://doi.org/10.1086/112164>

- Malan, A. G., Lewis, R. W., & Nithiarasu, P. (2002). An improved unsteady, unstructured, artificial compressibility, finite volume scheme for viscous incompressible flows: Part I. Theory and implementation. *International Journal for Numerical Methods in Engineering*, 54, 695–714. <https://doi.org/10.1002/nme.447>
- Mathur, S. R., & Murthy, J. Y. (1997). A pressure-based method for unstructured meshes. *Numerical Heat Transfer, Part B: Fundamentals*, 31(2), 195–215. <https://doi.org/10.1080/10407799708915105>
- Mazhar, Z. (2016). A novel fully implicit block coupled solution strategy for the ultimate treatment of the velocity–pressure coupling problem in incompressible fluid flow. *Numerical Heat Transfer, Part B: Fundamentals*, 69(2), 130–149. <https://doi.org/10.1080/10407790.2015.1093787>
- Nagy, R., Csobay-Novák, C., Lovas, A., Sótonyi, P., & Bojtár, I. (2015). Non-invasive in vivo time-dependent strain measurement method in human abdominal aortic aneurysms: Towards a novel approach to rupture risk estimation. *Journal of Biomechanics*, 48(10), 1876–1886. <https://doi.org/10.1016/j.jbiomech.2015.04.030>
- Oger, G., Doring, M., Alessandrini, B., & Ferrant, P. (2007). An improved SPH method: Towards higher order convergence. *Journal of Computational Physics*, 225(2), 1472–1492. <https://doi.org/10.1016/j.jcp.2007.01.039>
- Ozoe, H., & Tao, W. Q. (2001). A modified pressure-correction scheme for the SIMPLER method, MSIMPLER. *Numerical Heat Transfer, Part B: Fundamentals*, 39(5), 435–449. <https://doi.org/10.1080/104077901750188831>
- Pai, S. A., Prakash, P., & Patnaik, B. S. V. (2013). Numerical simulation of chaotic mixing in lid drive cavity: Effect of passive plug. *Engineering Applications of Computational Fluid Mechanics*, 7(3), 406–418. <https://doi.org/10.1080/19942060.2013.11015481>
- Patankar, S. V. (1980). *Numerical heat transfer and fluid flow*. <https://doi.org/10.1201/9781482234213>
- Patankar, S. V. (1981). A calculation procedure for two-dimensional elliptic situations. *Numerical Heat Transfer*, 4(4), 409–425. <https://doi.org/10.1080/01495728108961801>
- Perot, J. B. (2000). Conservation properties of unstructured staggered mesh schemes. *Journal of Computational Physics*, 159, 58–89. <https://doi.org/10.1006/jcph.2000.6424>
- Perron, S., Boivin, S., & Herard, V. (2004). A finite volume method to solve the 3D Navier-Stokes equations on unstructured collocated meshes. *Computers & Fluids*, 33, 1305–1333. <https://doi.org/10.1016/j.compfluid.2003.10.006>
- Plana Fattori, A., Chantoiseau, E., Doursat, C., & Flick, D. (2013). Two-way coupling of fluid-flow, heat transfer and product transformation during heat treatment of starch suspension inside tubular exchanger. *Engineering Applications of Computational Fluid Mechanics*, 7(3), 334–345. <https://doi.org/10.1080/19942060.2013.11015475>
- Ploumhans, P., Winckelmans, G. S., Salmon, J. K., Leonard, A., & Warren, M. S. (2002). Vortex methods for direct numerical simulation of three-dimensional bluff body flows: Application to the sphere at  $Re = 300, 500,$  and  $1000$ . *Journal of Computational Physics*, 178, 427–463. <https://doi.org/10.1006/jcph.2002.7035>
- Raviart, P. A., & Thomas, J. M. (1977). A mixed finite element method for 2-nd order elliptic problems. In I. Galligani & E. Magenes (Eds.), *Mathematical aspects of finite element methods*. Lecture Notes in Mathematics, Vol. 606. Springer. <https://doi.org/10.1007/BFb0064470>
- Roos, F. W., & Willmarth, W. W. (1971). Some experimental results on sphere and disk drag. *AIAA Journal*, 9, 285–291. <https://doi.org/10.2514/3.6164>
- Sakamoto, H., & Haniu, H. (1990). A study of vortex shedding from spheres in a uniform flow. *Journal of Fluids Engineering*, 112, 386. <https://doi.org/10.1115/1.2909415>
- Salih, S. Q., Aldlemy, M. S., Rasani, M. R., Ariffin, A. K., Ya, T. M. Y. S. T., Al-Ansari, N., Yaseen, Z. M. M., & Chau, K. W. (2019). Thin and sharp edges bodies-fluid interaction simulation using cut-cell immersed boundary method. *Engineering Applications of Computational Fluid Mechanics*, 13(1), 860–877. <https://doi.org/10.1080/19942060.2019.1652209>
- Schöberl, J. (1997). NETGEN – an advancing front 2D/3D-mesh generator based on abstract rules. *Computing and Visualization in Science*, 1(1), 41–52. <https://doi.org/10.1007/s007910050004>
- Shyy, W., & Mittal, R. (1998). Solution methods for the incompressible Navier-Stokes equations. In R. W. Johnson (Ed.), *Handbook of fluid dynamics* (pp. 31.1–31.33). CRC Press. ISBN 9780849325090.
- Sungsu, L. (2000). A numerical study of the unsteady wake behind a sphere in a uniform flow at moderate Reynolds numbers. *Computers & Fluids*, 29, 639–667. [https://doi.org/10.1016/S0045-7930\(99\)00023-7](https://doi.org/10.1016/S0045-7930(99)00023-7)
- Tao, W. Q. (2001). *Numerical heat transfer* (2nd ed.). Xi’an Jiaotong University Press. ISBN 10: 7560514367/ISBN 13: 9787560514369.
- Taylor, G. I., & Green, A. E. (1937). Mechanism of the production of small eddies from large ones. *Proceedings of the Royal Society of London. Series A – Mathematical and Physical Sciences*, 158, 499–521. <https://doi.org/10.1098/rspa.1937.0036>
- Tomboulides, A. G. (1993). *Direct and Large-Eddy simulation of wake flows: Flow past a sphere*, PhD thesis, Princeton University.
- Toro, E. F., Müller, L. O., & Siviglia, A. (2020). Bounds for wave speeds in the Riemann problem: Direct theoretical estimates. *Computers & Fluids*, 209. <https://doi.org/10.1016/j.compfluid.2020.104640>
- Toro, E. F., & Vazquez-Cendon, M. E. (2012). Flux splitting schemes for the Euler equations. *Computers & Fluids*, 70, 1–12. <https://doi.org/10.1016/j.compfluid.2012.08.023>
- Vidovic, D., Segal, A., & Wesseling, P. (2004). A superlinearly convergent finite volume method for the incompressible Navier-Stokes equations on staggered unstructured grids. *Journal of Computational Physics*, 198, 159–177. <https://doi.org/10.1016/j.jcp.2004.01.005>
- Vrahliotis, S., Pappou, T., & Tsangaris, S. (2012). Artificial compressibility 3-D Navier-Stokes solver for unsteady incompressible flows with hybrid grids. *Engineering Applications of Computational Fluid Mechanics*, 6(2), 248–270. <https://doi.org/10.1080/19942060.2012.11015419>
- Womersley, J. R. (1955). Method for the calculation of velocity, rate of flow and viscous drag in arteries when the pressure gradient is known. *The Journal of Physiology*, 127(3), 553–563. <https://doi.org/10.1113/jphysiol.1955.sp005276>
- Xue, S. C., & Burton, G. W. (2013). A finite volume formulation for transient convection and diffusion equations with unstructured distorted grids and its applications in fluid flow simulations with a collocated variable arrangement. *Computer Methods in Applied Mechanics and Engineering*, 253, 146–159. <https://doi.org/10.1016/j.cma.2012.09.016>
- Younes, A., Ackerer, P., & Chavent, G. (2004). From mixed finite elements to finite volumes for elliptic PDEs in two and three



dimensions. *International Journal for Numerical Methods in Engineering*, 59, 365–388. <https://doi.org/10.1002/nme.874>  
 Younes, A., Ackerer, P., & Lehmann, F. (2006). A new mass lumping scheme for the mixed hybrid finite element method. *International Journal for Numerical Methods in Engineering*, 67(1), 89–107. <https://doi.org/10.1002/nme.1628>  
 Zienkiewicz, O., Taylor, R., & Nithiarasu, P. (2013). *The finite element method for fluid dynamics*. Elsevier. <https://doi.org/10.1016/C2009-0-26328-8>

**Appendix 1. The Order subroutine**

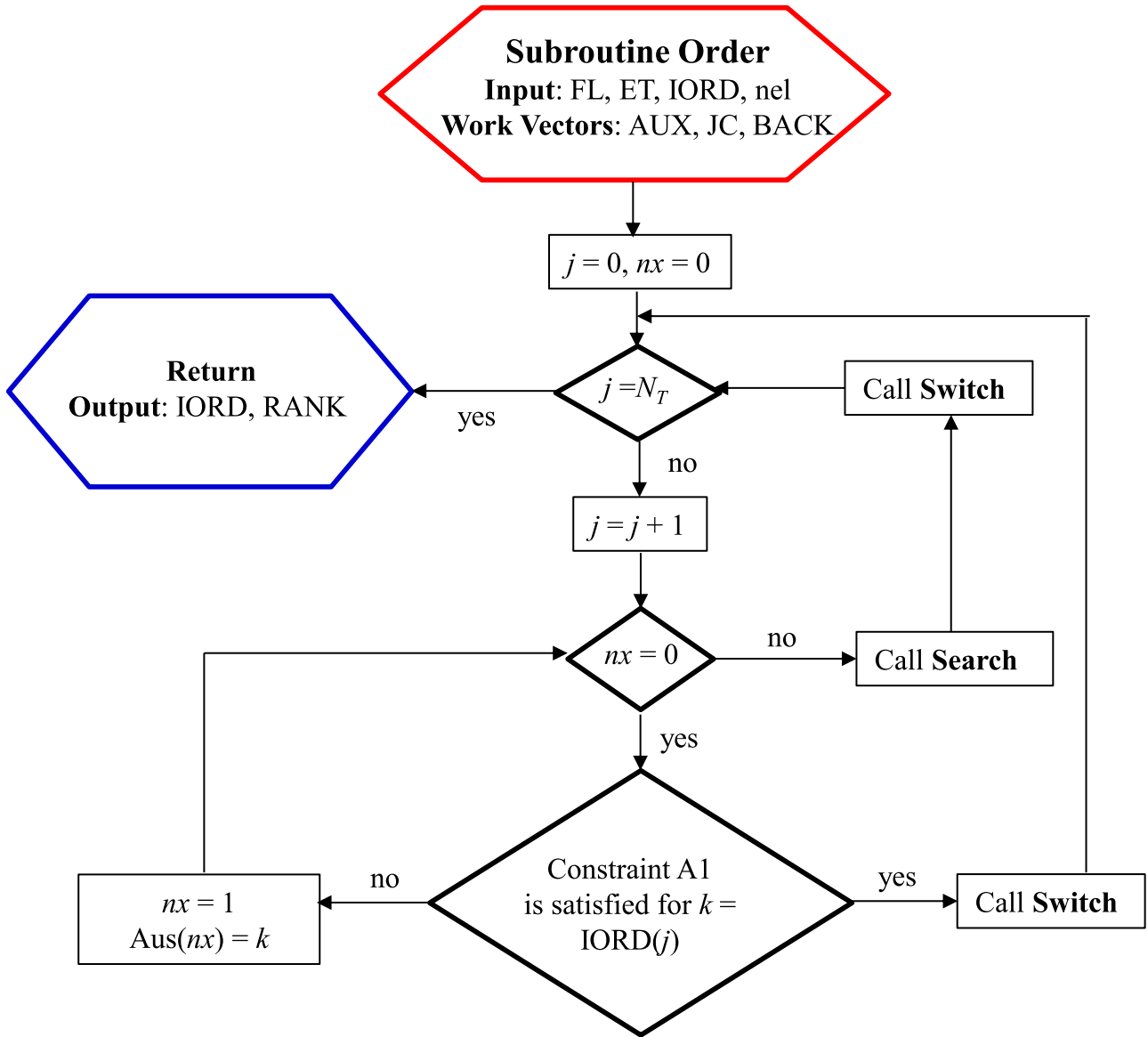
Call  $ET(j,k)$  the index of the  $j$ th tetrahedron neighbor to tetrahedron  $k$  and  $Fl(j,k)$  the flux between tetrahedrons  $k$  and  $ET(j,k)$ .  $ET(j,k) = 0$  if the  $j$ th face of tetrahedron  $k$  is a boundary face. Flux  $FL(j,k)$  is positive if it goes from  $k$  to  $ET(j,k)$ , negative otherwise. We assume the fluxes of two neighbor tetrahedrons  $k$  and  $k_p$  to have the same norm and opposite sign in the shared

face. The **Order** input are the integer matrices  $ET$ ,  $FL$  and the initial **IORD** vector.

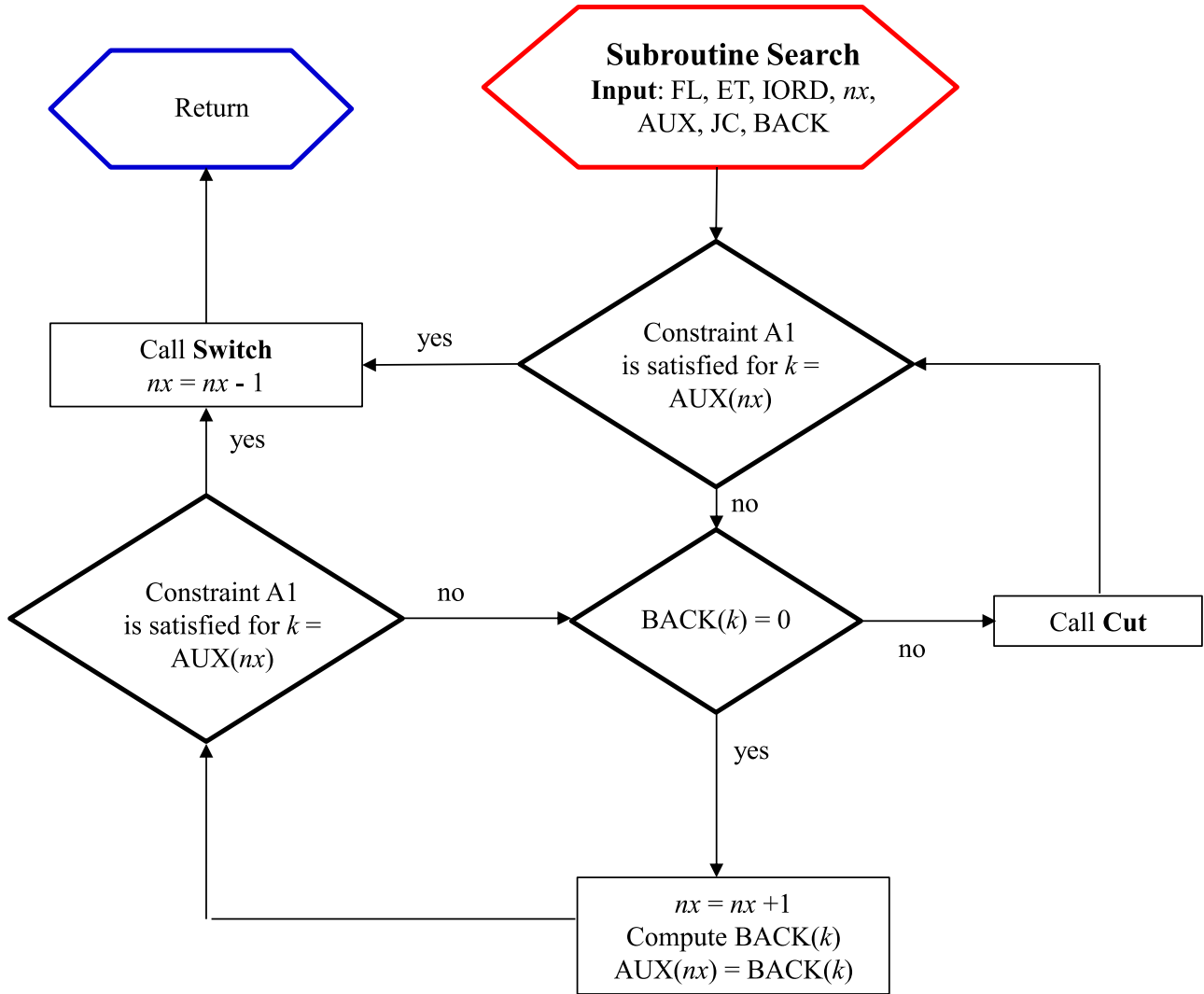
Call  $INV(k)$  the position  $m$  of tetrahedron  $k$  in the **IORD** vector, such that  $INV(IORD(m)) = m$ . Vector **INV** is initialized according to the input **IORD** vector. Initialize also two other auxiliary vectors **AUX** and **BACK** and one auxiliary matrix **JC** with size  $[4, N_T]$ , where  $N_T$  is the number of tetrahedrons. All vectors **RANK**, **BACK** and **JC** are initialized with 0.

The **Order** algorithm computes the vector **RANK** and updates the vector **IORD**. The output vector **IORD** provides, for each index  $m$ , the tetrahedron  $k = IORD(m)$  with the following properties:

$$\begin{aligned}
 &(\text{RANK}(ET(j, k)) > 0 \text{ or } \text{JC}(j, k) \neq 0) \text{ if } (\text{FL}(j, k) \\
 &< 0 \text{ and } \text{ET}(j, k) > 0) \quad j = 1, \dots, 4 \quad (A1) \\
 &(\text{RANK}(k) > \text{RANK}(ET(j, k)) \text{ or } \text{JC}(j, k) \neq 0) \\
 &\text{if } (\text{FL}(j, k) < 0 \text{ and } \text{ET}(j, k) > 0)
 \end{aligned}$$



**Figure A1.** Flow-chart of the Order subroutine.



**Figure A2.** Flow-chart of the Search subroutine.

$$\begin{aligned}
 & j = 1, \dots, 4 & (A2) \\
 & (\text{INV}(\text{ET}(j, k)) < \text{INV}(k) \text{ or } \text{JC}(j, k) \neq 0) \\
 & \text{if } (\text{FL}(j, k) < 0 \text{ and } \text{ET}(j, k) > 0) \\
 & \text{for any value } k \\
 & = \text{iord}(s) \text{ and } 0 < s < m & (A3)
 \end{aligned}$$

The general strategy is to compute the  $m$ th tetrahedron of **IORD** and its rank  $\text{RANK}(\text{IORD}(m))$  after the computation of the previous tetrahedrons  $\text{IORD}(1), \dots, \text{IORD}(m-1)$  and their rank. **Order** adopts the following subroutines, where apex  $i$  marks input variables and apex  $o$  marks output variables:

### A1.1. Subroutine Switch

Input:  $nx$ ,  $AUX$ ,  $\text{IORD}$ ,  $\text{INV}$ ,  $\text{FL}$ ,  $\text{ET}$ ,  $\text{RANK}$

Output:  $\text{RANK}$ ,  $\text{IORD}$ ,  $\text{INV}$

Given the known index  $r = \text{AUX}(nx)$  of a tetrahedron which satisfies constraints A1 and has  $\text{RANK}^i(r) = 0$ , compute the new rank of  $r$  as the maximum rank of the neighbor tetrahedrons that satisfy constraint (A1), plus one. This allows

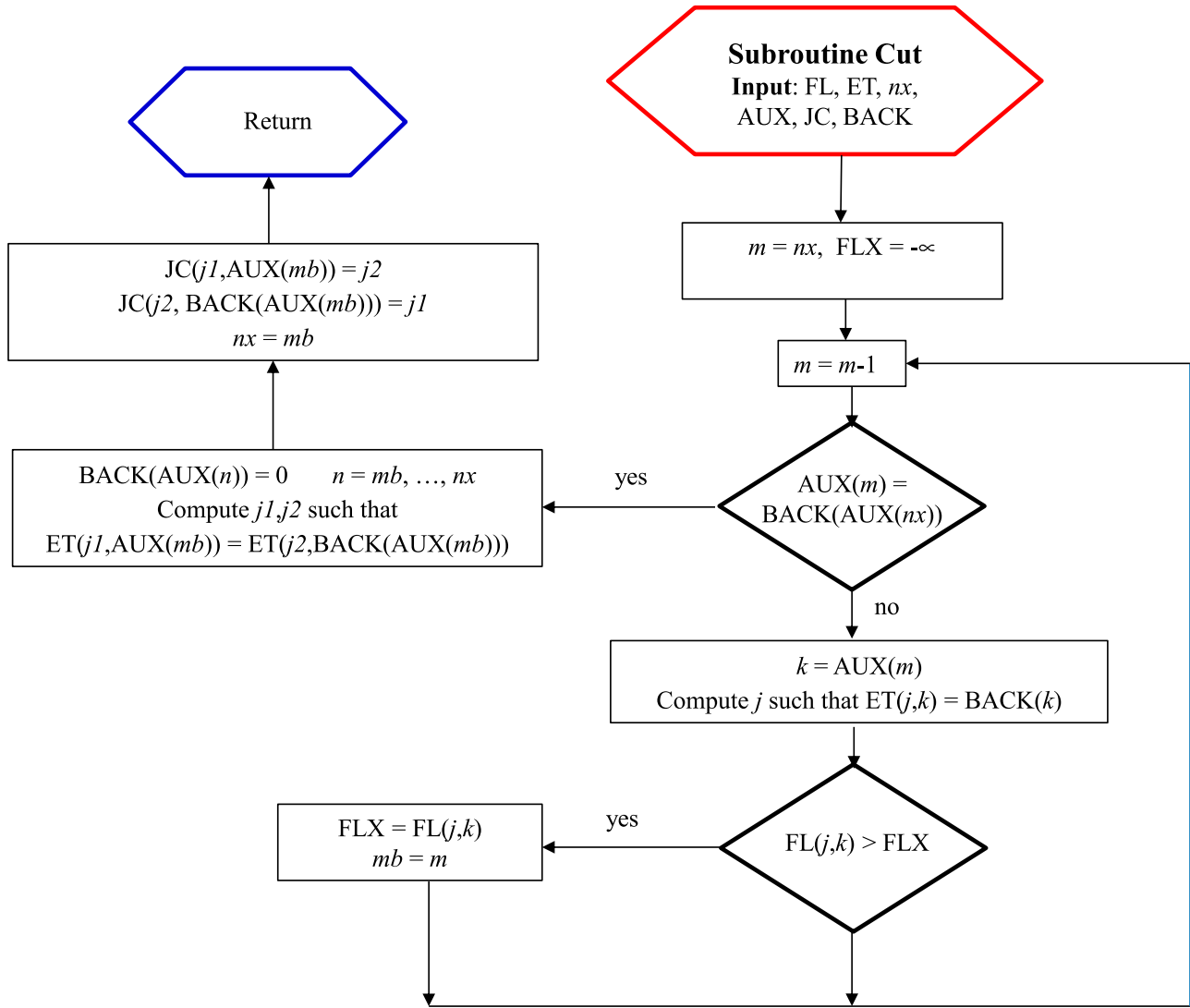
tetrahedron  $r$  to satisfy also requirement A2. Switch the position of tetrahedrons  $r$  and  $\text{IORD}^i(m)$  in the  $\text{IORD}^o$  vector, by setting:  $s = \text{INV}^i(r)$ ,  $\text{IORD}^o(m) = r$ ,  $\text{IORD}^o(s) = \text{IORD}^i(m)$ ,  $\text{INV}^o(r) = m$ ,  $\text{INV}^o(\text{IORD}^i(m)) = s$ . See the flow-chart in Figure A1.

### A1.2. Subroutine Search

Input:  $nx$ ,  $AUX$ ,  $\text{IORD}$ ,  $\text{INV}$ ,  $\text{FL}$ ,  $\text{ET}$ ,  $\text{RANK}$ ,  $\text{BACK}$ ,  $\text{JC}$

Output:  $nx$ ,  $AUX$ ,  $\text{BACK}$ ,  $\text{RANK}$ ,  $\text{IORD}$ ,  $\text{INV}$ ,  $\text{JC}$

Call  $\text{BACK}(k)$  the neighbor of tetrahedron  $k$  with  $\text{RANK}(k) = 0$  and  $\text{RANK}(\text{BACK}(k)) = 0$  with the maximum flux going from  $k$  to  $\text{BACK}(k)$ . Given a length  $nx^i \geq 1$ , check if constraint (A1) is satisfied for  $k = \text{AUX}(nx)$ . If it is satisfied, apply **Switch** and set  $nx^o = nx^i - 1$ . Otherwise, select the neighbor tetrahedron  $kp$  with the minimum (maximum absolute value) entering flux smaller than zero and with  $\text{RANK}^i(kp) = 0$ . If  $\text{BACK}(kp) \neq 0$ , we have a loop. In this case apply subroutine **Cut** and iterate the check. If constraint (A1) is not satisfied, select the neighbor tetrahedron  $kp$  with  $\text{RANK}(kp) = 0$  and minimum entering flux, set  $\text{BACK}(k) = kp$ , update  $nx^o$  with



**Figure A3.** Flow-chart of the Cut subroutine.

$nx^i + 1$ , set  $AUX(nx^0) = kp$ , and iterate until constraint (A1) is satisfied. See the flow-chart in Figure A2.

### A1.3. Subroutine Cut

Input:  $nx$ ,  $AUX$ ,  $BACK$ ,  $JC$ ,  $FL$ ,  $ET$

Output:  $nx$ ,  $BACK$ ,  $JC$

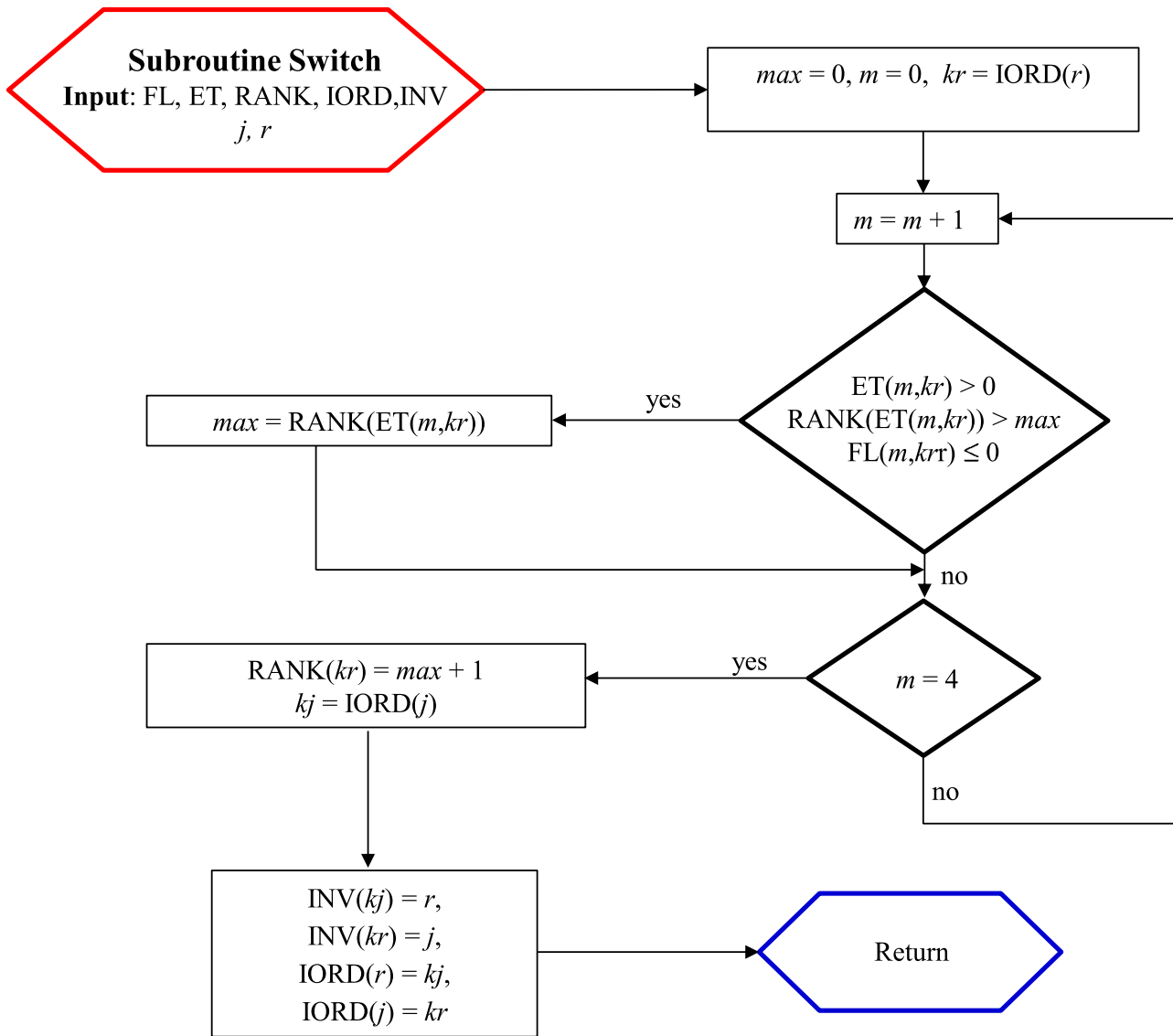
If a loop is found, compute the index  $mb$  and the tetrahedron  $kb = AUX^i(mb)$  corresponding to the minimum positive flux going from  $BACK(AUX^i(m))$  to  $AUX^i(m)$  for  $mp \leq m \leq nx^i$ , where  $AUX^i(mp) = BACK(AUX^i(nx^i))$ . Set  $nx^0 = mb$  and  $BACK^0(AUX^i(m)) = 0$ , for  $mb \leq m \leq nx^i$ . Set  $ka = BACK^0(kb)$ ,  $JC^0(n1, ka) = kb$  and  $JC^0(n2, kb) = ka$ , where  $n1$  and  $n2$  are the local indices of the face common to tetrahedrons  $ka$  and  $kb$ . See the flow-chart in Figure A3.

In **Order** we loop the index  $p$  from 1 to  $nel$ . At each iteration, if  $nx > 0$  we apply **Search**. If  $nx = 0$ , we test the tetrahedron

$k = IORD(p)$ . If constraint (A1) is satisfied, we apply **Switch**. If it is not satisfied, we set  $nx = 1$ ,  $AUX(nx) = k$  and apply **Search**. See the flow-chart in Figure A4.

Observe that, if zero loops and zero flux sign changes occur, the solution obtained in the MAST procedure is the same with any adopted sorting rule providing a sequence vector **IORD** also different from the vector computed in the **Order** subroutine, if constraint (A3) is satisfied for all the tetrahedrons. Due to loop cuts and flux sign changes, this is not true and a much better solution turns out to be the one obtained by solving sequentially all the tetrahedrons with the same rank, starting from 1, also using parallel computing if available. Any open source subroutine, like QUICKSORT, can be used to order all the tetrahedrons according to their rank value after the **RANK** solution of **Order** is found.

See the flow-chart of the **Order**, **Search**, **Cut** and **Switch** subroutines in Figures A1–A4.



**Figure A4.** Flow-chart of the Switch subroutine.