INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 0000; **00**:1–28
Published online in Wiley Online Library (www.onlinelibrary.wiley.com). DOI: 10.1002/fld.4723

High order divergence-free velocity reconstruction for free surface flows on unstructured Voronoi meshes

Walter Boscheri*[1], Giuseppe Roberto Pisaturo[2], Maurizio Righetti[2]

[1] *Department of Mathematics and Computer Science, University of Ferrara, Via Machiavelli 30 - 44121 Ferrara, Italy*
[2] *Faculty of Science and Technology, Free University of Bozen, Piazza Universitá 1 - 39100 Bolzano, Italy*

SUMMARY

In this paper we present an efficient semi-implicit scheme for the solution of the Reynolds-Averaged-Navier-Stokes (RANS) equations for the simulation of hydrostatic and non-hydrostatic free surface flow problems. A staggered unstructured mesh composed by Voronoi polygons is used to pave the horizontal domain, while parallel layers are adopted along the vertical direction. Pressure, velocity and vertical viscosity terms are taken implicitly, while the nonlinear convective terms as well as the horizontal viscous terms are discretized explicitly by using a semi-Lagrangian approach, which requires an interpolation of the three-dimensional velocity field to integrate the flow trajectories backward in time. To this purpose, a high order reconstruction technique is proposed, which is based on a *constrained* least squares operator that guarantees a *globally* and *pointwise divergence-free* velocity field. A comparison with an analogous reconstruction which is not divergence-free preserving is also presented to give evidence of the new strategy. This allows the continuity equation to be satisfied up to machine precision even for high order spatial discretizations. The reconstructed velocity field is then used for evaluating high order terms of a Taylor method that is here adopted as ODE integrator for the flow trajectories. The proposed semi-implicit scheme is validated against a set of academic test problems, and proof of convergence up to fourth order of accuracy in space is shown. Copyright © 0000 John Wiley & Sons, Ltd.

Received . . .

KEY WORDS: free surface flows, high order, divergence-free, semi-implicit, hydrostatic, non-hydrostatic, Voronoi mesh

1. INTRODUCTION

Free surface flows represent one of the most widespread phenomena which characterize environmental problems in engineering, namely involving oceanic currents, propagation of tsunami waves, sea tides and general water flows in mountain rivers and lakes. As a consequence, a lot of efforts have been invested in the development of numerical methods for the solution of the free surface governing equations, that are given by the incompressible Navier-Stokes equations. Because such a physical model is quite complex, several simplified models have been derived in order to solve specific flow problems, and different numerical schemes have also been designed accordingly.

*Correspondence to: Walter Boscheri, epartment of Mathematics and Computer Science, University of Ferrara, Via Machiavelli 30 - 44121 Ferrara, Italy. Email: walter.boscheri@unife.it

In this work we consider the three-dimensional equations describing free surface flows that are obtained after averaging over turbulent time scales the incompressible Navier-Stokes equations, hence leading to the so called Reynolds-Averaged-Navier-Stokes (RANS) equations. The presence of viscous terms in both the horizontal and the vertical direction, together with the free surface wave speed, typically arises a very severe CFL stability condition which penalizes explicit numerical schemes, by drastically reducing the admissible time step that guarantees the stability of the numerical method. For this reason, in [1, 2, 3, 4] a set of very efficient *semi-implicit* schemes have been proposed for the solution of different simplified models derived from the incompressible Navier-Stokes equations. Specifically, the shallow water equations have been considered in two [1] and three [2] space dimensions, while a semi-implicit finite element scheme has been discussed in [3]. For a better and rigorous simulation of vertical flows in the three-dimensional case, the hydrostatic pressure assumption must be released and the non-hydrostatic component of the pressure has to be included in the model by solving the vertical momentum equation, as done in the approach proposed in [5, 6]. In all those works, a staggered grid is adopted where the scalar normal velocity components are defined at the cell interfaces, whereas the pressure, i.e. the free surface elevation, is evaluated at the center of the control volumes. There, the pressure terms as well as the terms related to the velocity and the vertical viscosity are discretized *implicitly*, while the nonlinear convective and horizontal viscous terms are discretized *explicitly*. Specifically, the implicit velocities are used for a conservative discretization of the continuity equation and for the bed friction term in the momentum equation. On the other hand, a semi-Lagrangian method, like the ones presented in [7, 8, 9, 10], is employed for the remaining explicit terms. Such a choice is motivated by the fact that this is the only explicit method that is not constrained by a CFL-type stability condition and allows for large time steps. However, it requires the integration of the fluid trajectories backward in time, thus it needs an appropriate interpolation of the vector velocity field from the known scalar normal velocity components.

A finite-difference scheme is used for the discretization of the momentum equations, differently from the finite volume type approach employed for the free surface equation. As a consequence the method is fully conservative for mass, but not for momentum. A formulation that conserves momentum has been presented in [11], while the development of a mass-conservative treatment of wetting and drying is addressed in [12]. The efficient use of subgrid resolution has also been included in the same framework in [13].

Semi-implicit methods have been extended to Discontinuous Galerkin (DG) schemes [14, 15, 16, 17, 18, 19, 20] for viscous and inviscid hydrodynamics and magnetohydrodynamics, while in [21, 22] they are used for the simulation of compressible flows in tubes. Fully implicit DG methods can be found in [23, 24, 25, 26] for the numerical solution of both compressible and incompressible Navier-Stokes equations.

In [1, 2, 3, 4] a fractional step method is adopted to achieve at most second order of accuracy in time, and a second order interpolation of the velocity field is also included for evaluating the fluid trajectories needed in the semi-Lagrangian discretization of the convective terms. To improve the spatial accuracy of the scheme, a high order semi-implicit method has been designed in [27] in the context of polygonal Voronoi meshes, which makes use of a polynomial based high order reconstruction, on the lines of [28, 29].

In this paper we extend the method presented in [27] to *non-hydrostatic flows* and we propose a strategy for preserving the *divergence-free* constraint imposed by the equation for mass conservation. This condition is automatically fulfilled for a first order scheme simply by adopting a staggered mesh discretization with the scalar normal velocity components defined at cell interfaces, while this does not hold anymore in the case of a high order method, since the high order reconstruction polynomial that is recovered for the velocity field is *not*, in principle, divergence-free. In [30] a linear velocity field is reconstructed within a particle tracking numerical method that is able to preserve local mass conservation on two-dimensional triangular meshes. Here, a methodology will be illustrated that aims at satisfying the continuity equation of a free surface flow up to machine precision in fully three-dimensional flows. A *globally* and *pointwise* divergence-free velocity field is reconstructed for each control volume of the computational mesh, which is discretized by using Voronoi prisms.

The outline of this article is as follows: Section 2 presents the physical model with the governing equations considered in this work, while Section 3 contains all the details of the semi-implicit numerical scheme. The high order divergence-free reconstruction is fully discussed in Section 4 and in Section 5 a wide set of benchmark test problems is run in order to assess the accuracy and the validity of the new scheme. Finally, conclusions and outlook to future work are given in Section 6.

2. GOVERNING EQUATIONS

The conservation laws which govern free surface flows can be derived by the incompressible Navier-Stokes equations after averaging over turbulent time-scales. Let the physical space be described by the position vector $\mathbf{x} = (x, y, z)$, while let the velocity components in x, y and z direction be denoted by $u(x, y, z, t)$, $v(x, y, z, t)$ and $w(x, y, z, t)$, respectively, with t being the time. Furthermore, $g = 9.81 \left[m/s^2\right]$ is the gravitational acceleration and ν represents a viscosity coefficient which is assumed to be constant, hence no turbulence model is required. Finally, the fluid pressure $p(x, y, z, t)$ is taken to be the sum of three contributions, namely the atmospheric pressure $p_a(x, y, t)$, the hydrostatic pressure $p_h(x, y, z, t) = g\left(\eta(x, y, t) - z\right)$ and the non-hydrostatic part $q(x, y, z, t)$, hence explicitly yielding

$$p(x, y, z, t) = g\left(\eta(x, y, t) - z\right) + q(x, y, z, t), \tag{1}$$

with the assumption of $p_a(x, y, t) := 0$ and $\eta(x, y, t)$ denoting the free surface elevation, as illustrated in Figure 1.

Then, in the three-dimensional space, conservation of mass and momentum is given by

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0, \tag{2}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} = -g\frac{\partial \eta}{\partial x} - \frac{\partial q}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right), \tag{3}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} = -g\frac{\partial \eta}{\partial y} - \frac{\partial q}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right), \tag{4}$$

$$\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} = -\frac{\partial q}{\partial z} + \nu\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right), \tag{5}$$

$$\frac{\partial \eta}{\partial t} + \frac{\partial}{\partial x}\left[\int_{-h}^{\eta} u dz\right] + \frac{\partial}{\partial y}\left[\int_{-h}^{\eta} v dz\right] = 0. \tag{6}$$

The last equation (6) is referred to as free surface equation and is obtained by integrating the continuity equation (2) over depth and using a kinematic condition at the free surface. According to Figure 1, $h(x, y)$ is the bottom topography measured from the undisturbed water depth given at $z = 0$, whereas $H(x, y, t) = h(x, y) + \eta(x, y, t)$ represents the total water depth.

3. NUMERICAL SCHEME

In order to discretize the governing system (2)-(6), we propose to adopt the semi-implicit scheme presented in [6], in which pressure, velocity and vertical viscosity terms are discretized *implicitly*, whereas the terms of nonlinear convection and horizontal viscosity are taken *explicitly*. According to [27], unstructured Voronoi meshes are used to discretized the computational domain on the $x - y$ plane, while the velocity field is reconstructed by means of a high order polynomial which exactly preserves the divergence-free constraint, that is nothing but the continuity equation (2).

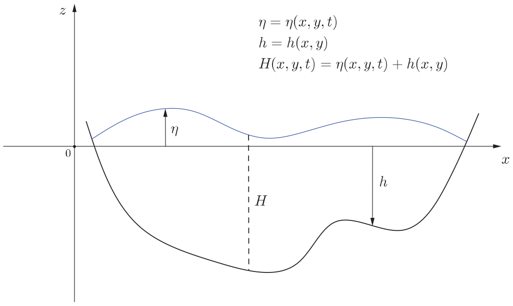

Figure 1. Notation adopted along the vertical direction z: free surface elevation η, bottom topography h and total water depth H.

3.1. Staggered unstructured Voronoi meshes

The computational domain Ω is first split into two parts, that is the horizontal domain Ω_{xy} and the vertical domain Ω_z. On the horizontal plane $x - y$ the computational grid is composed by a Voronoi tessellation which is, by construction, an *orthogonal unstructured mesh*, as required by the algorithm developed in [6]. In other words, a Voronoi mesh guarantees that the segment connecting two centroids of two adjacent elements always crosses orthogonally the shared edge that they have in common. In this work, such a mesh is constructed fully in parallel on several MPI threads as follows:

- the starting point is a coarse primary triangular unstructured mesh that covers the horizontal flow domain with non-overlapping control volumes C_γ, as shown in Figure 2. This mesh counts a total number of N_C triangles, hence $\gamma \in [1, N_C]$;
- the coarse mesh is then partitioned using the free graph partitioning software METIS [31] among the total number of threads N_{CPU} and each MPI rank *locally* refines its portion of the primary mesh by an arbitrary refinement factor $\boldsymbol{\chi}$. A structured refinement is used, where each big triangle C_γ is divided into a total number of $\boldsymbol{\chi}^2$ sub-triangles T_k. Both N_{CPU} and $\boldsymbol{\chi}$ have to be specified by the user at the beginning of the mesh generation. As a result, Ω_{xy} is discretized by a fine primary triangular grid composed by N_T cells T_k with $k \in [1, N_T]$;
- at this point each thread proceeds by generating the associated dual mesh, which is designed to be a Voronoi tessellation: for each triangle T_k the corresponding circumcenter is computed and all circumcenters provide the nodes for the Voronoi elements. Contrariwise, the centroids of the Voronoi cells are simply given by the nodes of the primary triangular mesh. Consequently, a dual Voronoi polygon P_i is obtained by connecting all circumcenters of the triangular elements T_k which share the primary vertex i under consideration. Both primary and dual meshes are displayed in Figure 3 to clarify how the primary and the dual grids are linked together. The horizontal computational domain results in a discretization of N_P non-overlapping Voronoi cells P_i with $i \in [1, N_P]$.

The only requirement for the successful construction of the Voronoi mesh as suggested above is that the coarse primary triangular grid must be a *Delaunay* tessellation with all angles of the mesh being less than 90 degrees of amplitude. Under this assumption, an unstructured Voronoi mesh can always be generated, fulfilling all the properties required by the algorithm:

1. orthogonality;
2. non-empty intersection between the segment connecting two centroids and the associated shared edge of the Voronoi cell on the entire computational mesh.

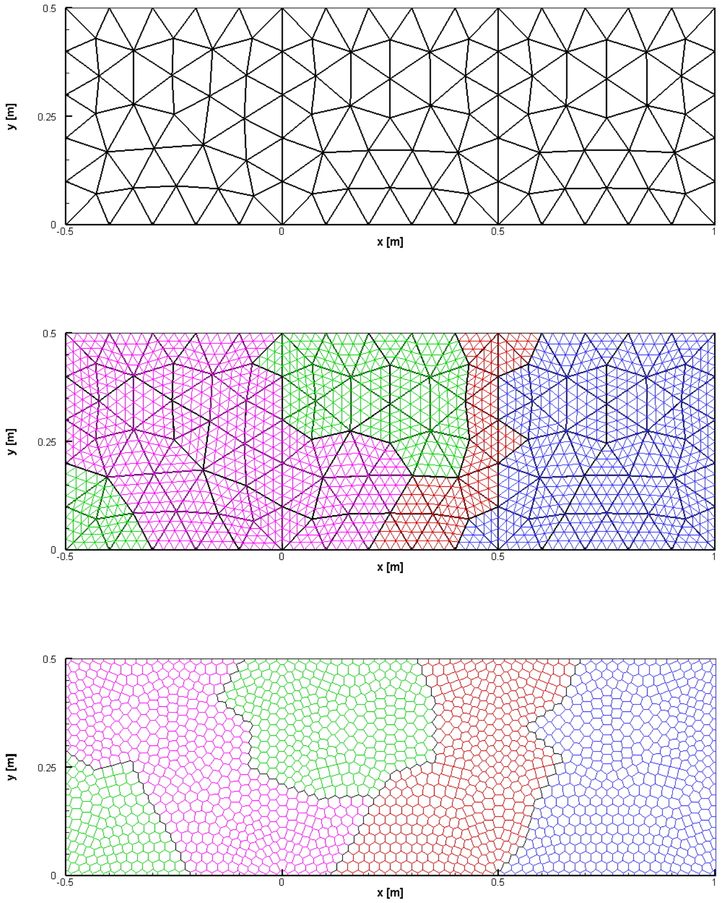

Figure 2. Steps for the generation of an unstructured Voronoi mesh on parallel threads with $N_{CPU} = 4$: primary coarse triangular mesh with N_C cells (top), primary fine triangular mesh with N_T cells (middle), dual Voronoi mesh with N_P polygons (bottom). Each MPI thread is identified with a different color.

Note that the first condition can be always obtained if the primary mesh is a Delaunay triangulation, while for the second requirement we additionally need to deal only with acute angles in the primary grid.

The final (time-dependent) three-dimensional computational domain $\Omega(t)$ consists of prisms whose horizontal bottom and top faces are given by a Voronoi cell P_i, and whose height is a vertical thickness Δz, that results from a discretization of the water column along the vertical direction. Figure 3 plots a portion of the resulting 3D computational grid, which is $\Omega(t) = \left\{ \mathbf{x} \in \mathbb{R}^3 | (x,y) \in \Omega_{xy} \wedge -h(x,y) \leq z \leq \eta(x,y,t) \right\}$.

A staggered discretization is then adopted on the Voronoi unstructured mesh, which is illustrated in Figure 4. Let P_i be the Voronoi cell and ∂P_i its boundary, which is composed by a total number of N_j edges denoted by λ_j. The left and right elements that share a common edge λ_j are addressed with $L(j)$ and $R(j)$, respectively, and the straight line segment connecting their corresponding centroids $\mathbf{c}_{L(j)}$ and $\mathbf{c}_{R(j)}$ is δ_j, according to Figure 4. Furthermore, we introduce a grid orientation through a sign function $\sigma_{i,j}$ which is defined for each edge λ_j between $L(j)$ and $R(j)$ as follows:

$$\sigma_{i,j} = \frac{R(j) - 2i + L(j)}{R(j) - L(j)}. \tag{7}$$

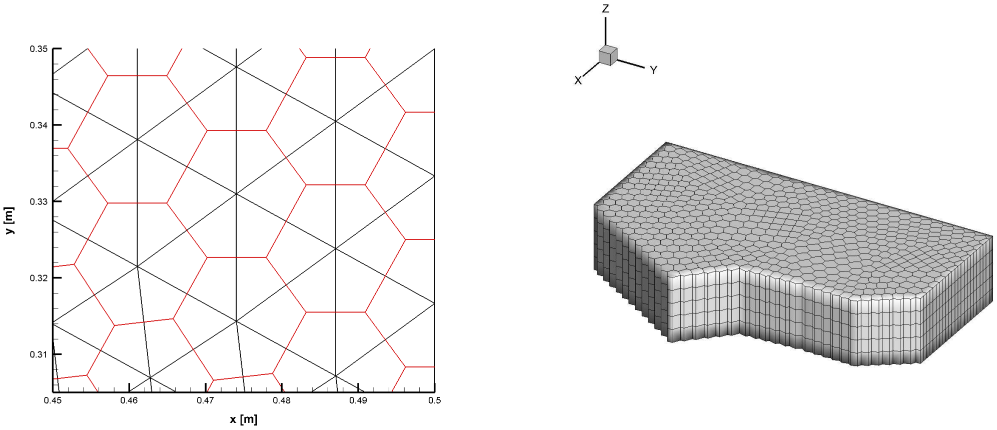

Figure 3. Left: primary triangular (black lines) and associated dual Voronoi (red lines) mesh. Right: three-dimensional computational grid composed by prisms with Voronoi base and thickness Δz^n.

The so-called $z-$layer discretization is used along the vertical direction, hence partitioning each water column by layers of thickness $\Delta z^n_{j,k}$, which can be either uniform or non-uniform. The number of active layers $N^n_{z,j}$, i.e. those layers which are wet, is a function of the free surface level that changes in time. Specifically, the index of the layer containing the bottom is addressed with m_j, whereas the index of the layer crossed by the free surface is denoted with M^n_j, hence $N^n_{z,j} = \left(M^n_j - m_j + 1\right)$. If no water is present on edge λ_j, then $N^n_{z,j} = 0$ and that edge is dry. We underline that we do not consider sediment transport phenomena, hence the bottom index m_j is not time-dependent.

At the aid of Figure 4, let us also explain where the physical variables appearing in the governing system (2)-(6) are defined on the staggered unstructured Voronoi grid:

- the free surface η_i is evaluated at the centroid $\mathbf{c}_i$ of polygon P_i and is taken to be constant within each Voronoi cell;
- the bottom topography h_j as well as the total water depth H^n_j are located on each edge λ_j;
- due to the unstructured grid, the horizontal velocity is no longer stored in terms of u and v components, but a face-averaged velocity $u^n_{j,k}$ is computed in *normal* direction to each face of the prism given by $\left[P_i \times \Delta z^n_{j,k}\right]$. In this way, the horizontal velocity is defined at the intersection point between the edge λ_j and the segment δ_j at height $z_{j,k}$ which is the center along the vertical direction of the k^{th} layer;
- the non-hydrostatic component $q^n_{i,k}$ is defined for each cell at the centroid $\mathbf{c}_i$ at height $z_{i,k}$;
- finally, the vertical velocity component $w^n_{i,k+\frac{1}{2}}$ is computed at the centroid $\mathbf{c}_{i,k}$ at $z_{i,k+\frac{1}{2}}$ for each active layer in vertical direction.

3.2. *Semi-implicit algorithm for hydrostatic and non-hydrostatic free surface flows*

We adopt a semi-implicit scheme to solve numerically the free surface flow equations (2)-(6), following the work presented in [6]. Velocity, pressure and vertical viscosity are discretized *implicitly*, while the convective and diffusive terms are computed *explicitly*, so that the stability of the resulting numerical algorithm is independent of the celerity, but only depends on the discretization of the nonlinear convective and the horizontal viscosity terms. The scheme is composed of two main steps, namely the hydrostatic prediction and the non-hydrostatic correction, which will be detailed hereafter.

Hydrostatic prediction The first part of the algorithm aims at computing *provisional* values for the free surface elevation as well as for the flow velocity by neglecting the implicit contribution

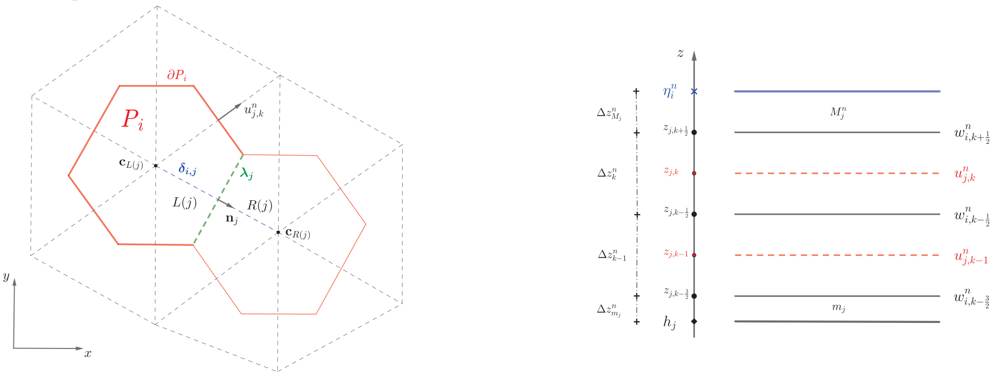

Figure 4. Notation and variable definition used on a staggered unstructured Voronoi mesh on the horizontal plane $x - y$ (left) and along the vertical direction z (right).

of the non-hydrostatic pressure. Relying on Galilean invariance, one can write the horizontal momentum conservation laws (3) and (4) in the normal direction $\mathbf{n}_j$ for each computational edge λ_j and each active layer k, hence obtaining only one evolution equation for the scalar normal component of the horizontal velocity. According to [6], a semi-implicit finite difference discretization is adopted for the momentum conservation laws of $\tilde{u}^{n+1}_{j,k}$ and $\tilde{w}^{n+1}_{i,k+\frac{1}{2}}$, while a mass-conservative finite volume scheme is used for the evolution of the free surface $\tilde{\eta}^{n+1}_i$:

$$\begin{aligned}
\tilde{u}^{n+1}_{j,k} &= Fu^n_{j,k} - (1-\theta)\frac{\Delta t}{\delta_j}\left[g\left(\eta^n_{R(j)} - \eta^n_{L(j)}\right) + \left(q^n_{R(j),k} - q^n_{L(j),k}\right)\right] \\
&- \theta g \frac{\Delta t}{\delta_j}\left(\tilde{\eta}^{n+1}_{R(j)} - \tilde{\eta}^{n+1}_{L(j)}\right) \\
&+ \nu \frac{\Delta t}{\Delta z^n_{j,k}}\left[\frac{\tilde{u}^{n+1}_{j,k+1} - \tilde{u}^{n+1}_{j,k}}{\Delta z^n_{j,k+\frac{1}{2}}} - \frac{\tilde{u}^{n+1}_{j,k} - \tilde{u}^{n+1}_{j,k-1}}{\Delta z^n_{j,k-\frac{1}{2}}}\right] \quad \forall k \in [m_j, M^n_j], \qquad (8) \\
\tilde{w}^{n+1}_{i,k+\frac{1}{2}} &= Fw^n_{i,k+\frac{1}{2}} - (1-\theta)\frac{\Delta t}{\Delta z^n_{i,k+\frac{1}{2}}}\left[q^n_{i,k+1} - q^n_{i,k}\right] \\
&+ \nu \frac{\Delta t}{\Delta z^n_{i,k+\frac{1}{2}}}\left[\frac{\tilde{w}^{n+1}_{i,k+\frac{3}{2}} - \tilde{w}^{n+1}_{i,k+\frac{1}{2}}}{\Delta z^n_{i,k+1}} - \frac{\tilde{w}^{n+1}_{i,k+\frac{1}{2}} - \tilde{w}^{n+1}_{i,k-\frac{1}{2}}}{\Delta z^n_{i,k}}\right] \quad \forall k \in [m_i, M^n_i - 1], \qquad (9) \\
\tilde{\eta}^{n+1}_i &= \eta^n_i - \theta \frac{\Delta t}{|P_i|}\sum_{\lambda_j \in \partial P_i}\left[|\lambda_j|\sigma_{i,j}\sum_{k=m_j}^{M^n_j}\Delta z^n_{j,k}\tilde{u}^{n+1}_{j,k}\right] \\
&- (1-\theta)\frac{\Delta t}{|P_i|}\sum_{\lambda_j \in \partial P_i}\left[|\lambda_j|\sigma_{i,j}\sum_{k=m_j}^{M^n_j}\Delta z^n_{j,k}u^n_{j,k}\right] \quad \forall i \in [1, N_P]. \qquad (10)
\end{aligned}$$

Here, $\Delta t = t^{n+1} - t^n$ denotes the time step, $|P_i|$ is the surface of the polygonal Voronoi cell P_i, while the symbol $\tilde{}$ is used to indicate the provisional values that need to be corrected in the second step of the algorithm. $Fu^n_{j,k}$ and $Fw^n_{i,k+\frac{1}{2}}$ represent two explicit operators that contain the discretization of the nonlinear convective and horizontal viscous terms. As done in [27], a semi-Lagrangian scheme is employed, hence obtaining

$$
\begin{aligned}
Fu^n_{j,k} &= \left(\mathbf{u}(\mathbf{x}^L_{j,k}) + \nu\Delta t\,\nabla^2_h \mathbf{u}(\mathbf{x}^L_{j,k})\right)\cdot\mathbf{n}_j, && (11)\\
Fw^n_{i,k+\frac{1}{2}} &= w(\mathbf{x}^L_{i,k+\frac{1}{2}}) + \nu\Delta t\,\nabla^2_h w(\mathbf{x}^L_{i,k+\frac{1}{2}}), && (12)
\end{aligned}
$$

with $\mathbf{x}^L_{j,k}$ denoting the foot of the Lagrangian trajectory that has started from point $\mathbf{x}^n_{j,k}$, which is defined on edge λ_j and layer k for $Fu^n_{j,k}$. Similarly, for $Fw^n_{i,k+\frac{1}{2}}$ the trajectory begins at point $\mathbf{x}^n_{i,k+\frac{1}{2}}$, which is located at centroid $\mathbf{c}_i$ and height $z^n_{i,k+\frac{1}{2}}$, and it ends at its foot $\mathbf{x}^L_{i,k+\frac{1}{2}}$. $\mathbf{u}(\mathbf{x}) = (u(\mathbf{x}), v(\mathbf{x}))$ and $w(\mathbf{x})$ are a continuous interpolation of the horizontal velocity field that arises from the high order divergence-free reconstruction technique described in Section 4, whereas $\nabla^2_h\mathbf{u}$ and $\nabla^2_h w$ represent a discretization of the horizontal Laplace operator evaluated at the foot of the characteristics, which will be discussed in Section 4.1. To improve the time accuracy of the method, the implicitness factor θ has been introduced in (8)-(9), and for the sake of stability it must be bounded within the range $\theta \in [0.5, 1]$, see [32]. In our algorithm, the stability of the scheme only depends on the horizontal viscosity, therefore the time step must obey the following condition:

$$
\Delta t \leq \frac{l^2_{min}}{4\nu}, \tag{13}
$$

with $l_{min} = \min\sqrt{P_i}$ being the smallest characteristic mesh size of the entire computational grid. To ease the notation, let us define the vectors

$$
\tilde{\mathbf{U}}^{n+1}_j = \begin{bmatrix} \tilde{u}^{n+1}_{j,m_j} \\ \vdots \\ \tilde{u}^{n+1}_{j,k} \\ \vdots \\ \tilde{u}^{n+1}_{j,M^n_j} \end{bmatrix}, \quad
\mathbf{U}^n_j = \begin{bmatrix} u^n_{j,m_j} \\ \vdots \\ u^n_{j,k} \\ \vdots \\ u^n_{j,M^n_j} \end{bmatrix}, \quad
\tilde{\mathbf{W}}^{n+1}_i = \begin{bmatrix} \tilde{w}^{n+1}_{i,m_i} \\ \vdots \\ \tilde{w}^{n+1}_{i,k+\frac{1}{2}} \\ \vdots \\ \tilde{w}^{n+1}_{i,M^n_i-1} \end{bmatrix}, \quad
\mathbf{\Delta Z}^n_j = \begin{bmatrix} \Delta z^n_{j,m_j} \\ \vdots \\ \Delta z^n_{j,k} \\ \vdots \\ \Delta z^n_{j,M^n_j} \end{bmatrix},
$$

$$
\mathbf{A}^n_j = \begin{bmatrix}
\Delta z^n_{j,m_j} + \nu\frac{\Delta t}{\Delta z^n_{j,m_j+\frac{1}{2}}} & -\nu\frac{\Delta t}{\Delta z^n_{j,m_j+\frac{1}{2}}} & 0 \\
 & \vdots & \\
-\nu\frac{\Delta t}{\Delta z^n_{j,k-\frac{1}{2}}} & \Delta z^n_{j,k} + \nu\frac{\Delta t}{\Delta z^n_{j,k-\frac{1}{2}}} + \nu\frac{\Delta t}{\Delta z^n_{j,k+\frac{1}{2}}} & -\nu\frac{\Delta t}{\Delta z^n_{j,k+\frac{1}{2}}} \\
 & \vdots & \\
0 & -\nu\frac{\Delta t}{\Delta z^n_{j,M^n_j-\frac{1}{2}}} & \Delta z^n_{j,M^n_j} + \nu\frac{\Delta t}{\Delta z^n_{j,M^n_j-\frac{1}{2}}}
\end{bmatrix},
$$

$$
\mathbf{A}^n_i = \begin{bmatrix}
\Delta z^n_{i,m_i+\frac{1}{2}} + \nu\frac{\Delta t}{\Delta z^n_{i,m_i+1}} & -\nu\frac{\Delta t}{\Delta z^n_{i,m_i+1}} & 0 \\
 & \vdots & \\
-\nu\frac{\Delta t}{\Delta z^n_{i,k}} & \Delta z^n_{i,k+\frac{1}{2}} + \nu\frac{\Delta t}{\Delta z^n_{i,k}} + \nu\frac{\Delta t}{\Delta z^n_{i,k+1}} & -\nu\frac{\Delta t}{\Delta z^n_{i,k+1}} \\
 & \vdots & \\
0 & -\nu\frac{\Delta t}{\Delta z^n_{i,M^n_i-1}} & \Delta z^n_{i,M^n_i-\frac{1}{2}} + \nu\frac{\Delta t}{\Delta z^n_{i,M^n_i-1}}
\end{bmatrix},
$$

$$\mathbf{T}_i^n = \begin{bmatrix} \Delta z^n_{i,m_i+\frac{1}{2}}\left(Fw^n_{i,m_i+\frac{1}{2}} - (1-\theta)\Delta t\left(q^n_{i,m_i+1}-q^n_{i,m_i}\right)\right)\\ \vdots \\ \Delta z^n_{i,k+\frac{1}{2}}\left(Fw^n_{i,k+\frac{1}{2}} - (1-\theta)\Delta t\left(q^n_{i,k+1}-q^n_{i,k}\right)\right)\\ \vdots \\ \Delta z^n_{i,M_i^n-\frac{1}{2}}\left(Fw^n_{i,M_i^n-\frac{1}{2}} - (1-\theta)\Delta t\left(q^n_{i,M_i^n}-q^n_{i,M_i^n-1}\right)\right) \end{bmatrix},$$

$$\mathbf{G}_j^n = \begin{bmatrix} \Delta z^n_{j,m_j}\left(Fu^n_{j,m_j} - (1-\theta)\frac{\Delta t}{\delta_j}\left(g\left(\eta^n_{R(j)}-\eta^n_{L(j)}\right)+q^n_{R(j),m_j}-q^n_{L(j),m_j}\right)\right)\\ \vdots \\ \Delta z^n_{j,k}\left(Fu^n_{j,k} - (1-\theta)\frac{\Delta t}{\delta_j}\left(g\left(\eta^n_{R(j)}-\eta^n_{L(j)}\right)+q^n_{R(j),k}-q^n_{L(j),k}\right)\right)\\ \vdots \\ \Delta z^n_{j,M_j^n}\left(Fu^n_{j,M_j^n} - (1-\theta)\frac{\Delta t}{\delta_j}\left(g\left(\eta^n_{R(j)}-\eta^n_{L(j)}\right)+q^n_{R(j),M_j^n}-q^n_{L(j),M_j^n}\right)\right) \end{bmatrix}, \tag{14}$$

(15)

which are used to rewrite in compact matrix-vector form the discretized governing equations (8)-(10) as

$$\begin{aligned} \mathbf{A}_j^n\tilde{\mathbf{U}}_j^{n+1} &= \mathbf{G}_j^n - \theta g\frac{\Delta t}{\delta_j}\left(\tilde\eta^{n+1}_{R(j)}-\tilde\eta^{n+1}_{L(j)}\right)\mathbf{\Delta Z}_j^n, \qquad (16)\\ \mathbf{A}_i^n\tilde{\mathbf{W}}_i^{n+1} &= \mathbf{T}_i^n \qquad (17)\\ \tilde\eta_i^{n+1} &= \tilde\eta_i^n - \theta\frac{\Delta t}{|P_i|}\sum_{\lambda_j\in\partial P_i}|\lambda_j|\sigma_{i,j}(\mathbf{\Delta Z}_j^n)^T\tilde{\mathbf{U}}_j^{n+1}\\ &- (1-\theta)\frac{\Delta t}{|P_i|}\sum_{\lambda_j\in\partial P_i}|\lambda_j|\sigma_{i,j}(\mathbf{\Delta Z}_j^n)^T\mathbf{U}_j^n. \qquad (18) \end{aligned}$$

Boundary conditions have already taken into account by the auxiliary matrices $\mathbf{A}_j^n$ and $\mathbf{A}_i^n$, imposing a no-slip wall condition on the bottom, i.e. $\tilde u_{j,m_j-\frac{1}{2}}=0$ and $\tilde w_{i,m_i-\frac{1}{2}}=0$, while requiring zero stress on the free surface, hence $\frac{\partial \tilde u_{j,M_j^n+\frac{1}{2}}}{\partial z}=0$. The vertical momentum equation (9) is defined up to layer M_i^n-1, hence no free surface boundary condition is needed for building matrix $\mathbf{A}_i^n$. Other choices are possible, considering friction coefficients at $z=-h$ or wind stresses on $z=\eta$, see [6] for more details.

For each cell P_i, the linear tridiagonal system (17) is solved for the vertical velocity unknowns $\tilde w^n_{i,k+\frac{1}{2}}$ by a direct method, namely the Thomas algorithm. This is not the case for the horizontal momentum equations (16) that are coupled with the equations for the unknown free surface elevation $\tilde\eta_i^{n+1}$. Following [6], formal substitution of expression (16) into (18) yields a symmetric and positive definite linear system of N_P equations with the provisional values $\tilde\eta_i^{n+1}$ as the only unknowns:

$$\begin{aligned} |P_i|\tilde\eta_i^{n+1} &- g\theta^2\Delta t^2\sum_{\lambda_j\in\partial P_i}\frac{|\lambda_j|\sigma_{i,j}}{\delta_j}\left[\mathbf{\Delta Z}^T\mathbf{A}^{-1}\mathbf{\Delta Z}_j\right]_j^n\left(\tilde\eta^{n+1}_{R(j)}-\tilde\eta^{n+1}_{L(j)}\right) = \\ & |P_i|\eta_i^n - (1-\theta)\Delta t\sum_{\lambda_j\in\partial P_i}|\lambda_j|\sigma_{i,j}\left[\mathbf{\Delta Z}^T\,\mathbf{U}\right]_j^n \\ &- \theta\Delta t\sum_{\lambda_j\in\partial P_i}|\lambda_j|\sigma_{i,j}\left[\mathbf{\Delta Z}^T\mathbf{A}^{-1}\mathbf{G}\right]_j^n. \qquad (19) \end{aligned}$$

System (19) is solved with the conjugate gradient method by iteration until the residual norm becomes less than a prescribed tolerance $\epsilon_\eta = 10^{-12}$. Once the free surface elevation is computed, the provisional horizontal normal velocities $\tilde{\mathbf{U}}_j^n$ are efficiently evaluated using again the Thomas algorithm for the tridiagonal systems (16).

Non-hydrostatic correction The provisional values $\tilde{\eta}_i^{n+1}$, $\tilde{u}_{j,k}^{n+1}$ and $\tilde{w}_{i,k+\frac{1}{2}}^{n+1}$ are now employed for the non-hydrostatic correction of the pressure. The pressure equation (1) can be discretized by using either the values at time t^{n+1} (which are still unknown) or the provisional values coming from the hydrostatic prediction, that is

$$\begin{aligned} p_{i,k}^{n+1} &= g\left(\tilde{\eta}_i^{n+1} - z_k\right) + \tilde{q}_{i,k}^{n+1}, && (20) \\ p_{i,k}^{n+1} &= g\left(\eta_i^{n+1} - z_k\right) + q_{i,k}^{n+1}. && (21) \end{aligned}$$

By assuming that the pressure is hydrostatic in the cells containing the free surface, the pressure correction $\tilde{q}_{i,M_i^n}^{n+1}$ is obtained at the aid of the following hydrostatic relation:

$$p_{i,M}^{n+1} = g\left(\tilde{\eta}_i^{n+1} - z_M\right) = g\left(\eta_i^{n+1} - z_M\right) + \tilde{q}_{i,M}^{n+1}. \tag{22}$$

Now, according to [6], a finite volume discretization of the continuity equation (2) for all the cells below the free surface yields

$$\sum_{\lambda_j \in \partial P_i} |\lambda_j| \sigma_{i,j} \Delta z_{j,k}^n u_{j,k}^{n+1} + |P_i| \left(w_{i,k+\frac{1}{2}}^{n+1} - w_{i,k-\frac{1}{2}}^{n+1} \right) = 0 \qquad \forall k \in [m_i, M_{i-1}^n], \tag{23}$$

where the new velocity components $u_{j,k}^{n+1}$ and $w_{i,k+\frac{1}{2}}^{n+1}$ are defined by means of the non-hydrostatic correction as

$$\begin{aligned} u_{j,k}^{n+1} &= \tilde{u}_{j,k}^{n+1} - \theta \frac{\Delta t}{\delta_j} \left(\tilde{q}_{R(j),k}^{n+1} - \tilde{q}_{L(j),k}^{n+1} \right), && (24) \\ w_{i,k+\frac{1}{2}}^{n+1} &= \widetilde{w}_{i,k+\frac{1}{2}}^{n+1} - \theta \frac{\Delta t}{\Delta z_{i,k+\frac{1}{2}}^n} \left(\tilde{q}_{i,k+1}^{n+1} - \tilde{q}_{i,k}^{n+1} \right). && (25) \\ & && (26) \end{aligned}$$

At $z = \eta$ we consider a finite difference approximation of the free surface equation (6), which is given by

$$\begin{aligned} |P_i| \eta_i^{n+1} = |P_i| \eta_i^n \quad & - \quad \theta \Delta t \sum_{\lambda_j \in \partial P_i} \left[\sigma_{i,j} |\lambda_j| \Delta z_{j,M_j}^n u_{j,M_j}^{n+1} \right] + \theta \Delta t |P_i| w_{i,M-\frac{1}{2}}^{n+1} \\ & - \quad (1-\theta) \Delta t \left[\sum_{\lambda_j \in \partial P_i} |\lambda_j| \sigma_{i,j} \sum_{k=m}^{M^n} \Delta z_{j,k}^n u_{j,k}^n \right], \end{aligned} \tag{27}$$

after using the discrete continuity equation (23) and setting $w_{i,m-\frac{1}{2}}^{n+1} = 0$. Following [6], relation (22) is used to write the above expression (27) as

$$\begin{aligned} |P_i| \tilde{q}_{i,M}^{n+1} = g |P_i| \left(\eta_i^n - \tilde{\eta}_i^{n+1} \right) \quad & - \quad g\theta \Delta t \sum_{\lambda_j \in \partial P_i} \left[\sigma_{i,j} |\lambda_j| \Delta z_{j,M_j}^n u_{j,M_j}^{n+1} \right] + g\theta \Delta t |P_i| w_{i,M-\frac{1}{2}}^{n+1} \\ & - \quad g(1-\theta) \Delta t \left[\sum_{\lambda_j \in \partial P_i} |\lambda_j| \sigma_{i,j} \sum_{k=m}^{M^n} \Delta z_{j,k}^n u_{j,k}^n \right]. \end{aligned} \tag{28}$$

At this point the velocity discretizations (24) and (25) are substituted into the approximations (23) and (28), leading to the following systems of equations for the non-hydrostatic correction below the free surface $\tilde{q}_{i,k}^{n+1}$ and at the free surface $\tilde{q}_{i,M}^{n+1}$, respectively:

$$\begin{aligned}\theta\Delta t & \left[\sum_{\lambda_j\in\partial P_i}|\lambda_j|\sigma_{i,j}\Delta z_{j,k}^n\frac{\tilde{q}_{L(j),k}^{n+1}-\tilde{q}_{R(j),k}^{n+1}}{\delta_j}+|P_i|\left(\frac{\tilde{q}_{i,k}^{n+1}-\tilde{q}_{i,k+1}^{n+1}}{\Delta z_{i,k+\frac{1}{2}}^n}-\frac{\tilde{q}_{i,k-1}^{n+1}-\tilde{q}_{i,k}^{n+1}}{\Delta z_{i,k-\frac{1}{2}}^n}\right)\right] \\ = & \; |P_i|\left(\tilde{w}_{i,k+\frac{1}{2}}^{n+1}-\tilde{w}_{i,k-\frac{1}{2}}^{n+1}\right)+\sum_{\lambda_j\in\partial P_i}|\lambda_j|\sigma_{i,j}\Delta z_{j,k}^n\tilde{u}_{j,k}^{n+1} \qquad \forall k\in[m_i,M_{i-1}^n],\end{aligned} \tag{29}$$

and

$$\begin{aligned}\theta\Delta t & \left[\sum_{\lambda_j\in\partial P_i}|\lambda_j|\sigma_{i,j}\Delta z_{j,M}^n\frac{\tilde{q}_{L(j),M}^{n+1}-\tilde{q}_{R(j),M}^{n+1}}{\delta_j}-|P_i|\frac{\tilde{q}_{i,M-1}^{n+1}-\tilde{q}_{i,M}^{n+1}}{\Delta z_{i,M-\frac{1}{2}}^n}\right]+\frac{|P_i|}{g\theta\Delta t}\tilde{q}_{i,M}^{n+1} \\ = & \; |P_i|\tilde{w}_{i,M-\frac{1}{2}}^{n+1}-\sum_{\lambda_j\in\partial P_i}|\lambda_j|\sigma_{i,j}\Delta z_{j,M}^n\tilde{u}_{j,M}^{n+1}+\frac{|P_i|}{\theta\Delta t}\left(\eta_i^n-\tilde{\eta}_i^{n+1}\right) \\ - & \; \frac{(1-\theta)}{\theta}\left[\sum_{\lambda_j\in\partial P_i}|\lambda_j|\sigma_{i,j}\sum_{k=m}^{M^n}\Delta z_{j,k}^n u_{j,k}^n\right].\end{aligned} \tag{30}$$

Expressions (29)-(30) constitute a positive definite system with at most $N_P \times \max\left\{N_{z,i}^n\right\}$ that is solved relying on a conjugate gradient algorithm. The iterations stop when the residual norm is smaller than a given tolerance $\epsilon_q = 10^{-10}$. In vertical direction, an impenetrable boundary is used at the bottom, hence $\tilde{w}_{i,m_i} = 0$.

On the horizontal plane, zero normal flux is imposed across wall-type boundary conditions, whereas a prescribed value for $\tilde{u}_{j,k}^{n+1}$ or $\tilde{q}_{i,k}^{n+1}$ is provided for open boundaries. Once the solution of systems (29)-(30) is obtained, the horizontal normal velocity is updated according to (24), while for the sake of conservation the vertical velocity component is derived from the continuity equation (23) as

$$w_{i,k+\frac{1}{2}}^{n+1} = w_{i,k-\frac{1}{2}}^{n+1} - \frac{1}{|P_i|}\sum_{\lambda_j\in\partial P_i}|\lambda_j|\sigma_{i,j}\Delta z_{j,k}^n u_{j,k}^{n+1}. \tag{31}$$

Finally, the new free surface elevation and the non-hydrostatic component of the pressure are given by

$$\eta_i^{n+1} = \tilde{\eta}_i^{n+1} + \frac{\tilde{q}_{i,M_i^n}}{g} \qquad \forall i\in[1,N_P] \tag{32}$$

$$q_{i,k}^{n+1} = \tilde{q}_{i,k}^{n+1} - \tilde{q}_{i,M_i^n} \qquad \forall i\in[1,N_P], \quad \forall k\in[m_i^n,M_i^n], \tag{33}$$

where the last relation (33) directly follows from equating the discrete forms (20)-(21) of the pressure equation (1).

4. HIGH ORDER DIVERGENCE-FREE VELOCITY RECONSTRUCTION

The numerical scheme presented so far makes use of pointwise horizontal normal velocity components $u_{j,k}^n$ and pointwise vertical velocity components $w_{i,k+\frac{1}{2}}^{n+1}$. Nevertheless, a continuous interpolation of the velocity field is required for the computation of the Lagrangian trajectories needed to discretize the nonlinear convective terms $Fu_{j,k}^n$ and $Fw_{i,k+\frac{1}{2}}^n$. Therefore, a *reconstruction* step is carried out in order to obtain a high order description of the velocity field in each computational cell. Here, a polynomial of degree N is defined within each control volume $P_i \times \left[z_{k-\frac{1}{2}}, z_{k+\frac{1}{2}}\right]$ and is used to recover a spatial order of accuracy $N+1$ for the velocity field. First, a reconstruction step is performed on the horizontal plane $x-y$ for each active layer

k, which generates the high order polynomials $\mathbf{u}_{i,k}(\mathbf{x}) = (u_{i,k}(\mathbf{x}), v_{i,k}(\mathbf{x}))$ and $\mathbf{w}_{i,k}(\mathbf{x})$. Then, along the vertical direction z, the velocity reconstruction at a generic point $\mathbf{x} = (x, y, z)$ is based on classical Lagrange interpolation through a set of $N + 1$ velocity values (degrees of freedom) computed using the horizontal velocity polynomials defined on a local set of layers. Furthermore, the reconstruction developed in this work is designed to preserve the divergence-free constraint up to machine precision. The choice of a decoupled reconstruction for horizontal and vertical flow directions allows the scheme to be computationally less expensive compared to a fully three-dimensional reconstruction.

The final reconstruction polynomial will then be referred to as $\bar{\mathbf{u}}(\mathbf{x}) = [\mathbf{u}(\mathbf{x}), \mathbf{w}(\mathbf{x})]$ which gives a fully three-dimensional high order interpolation of the velocity field at any generic point $\mathbf{x}$ in the entire computational domain.

The high order divergence-free reconstruction procedure is carried out following the steps listed hereafter.

Reconstruction on the horizontal plane The unknown high order polynomials which reconstruct the horizontal velocity field are expressed at the current time t^n in terms of a normalized Taylor series of degree N. The normalizing factor $l_i = \sqrt{P_i}$ is needed in order to avoid ill-conditioned reconstruction matrices [33], while the centroid $\mathbf{c}_i = (x_i, y_i)$ is used as center of expansion. For the horizontal normal velocity, the reconstruction polynomial is defined at height $z^n_{i,k}$, whereas for the vertical velocity the horizontal plane is considered at height $z^n_{i,k+\frac{1}{2}}$, according to the definition of the corresponding velocities displayed in Figure 4. To be clear, Einstein notation is adopted in the formulae which follow hereafter, implying summation over repeated indexes.

For each control volume, the horizontal reconstruction step aims at defining two polynomials which read

$$\bar{\mathbf{v}}^n_{i,k}(x,y) \;=\; \sum_{p+q\le N} \frac{(x-x_i)^p}{p!\, l_i^p}\frac{(y-y_i)^q}{q!\, l_i^q}\cdot\frac{\partial^{p+q}}{\partial x^p \partial y^q}\hat{\mathbf{v}}^n_{i,k} \;:= \psi_l\, \hat{\mathbf{v}}^l_{i,k}, \tag{34}$$

$$\bar{\mathbf{w}}^n_{i,k+\frac{1}{2}}(x,y) \;=\; \sum_{p+q\le N} \frac{(x-x_i)^p}{p!\, l_i^p}\frac{(y-y_i)^q}{q!\, l_i^q}\cdot\frac{\partial^{p+q}}{\partial x^p \partial y^q}\hat{w}^n_{i,k+\frac{1}{2}} \;:= \psi_l\, \hat{\mathbf{w}}^l_{i,k+\frac{1}{2}}, \tag{35}$$

with the unknown expansion coefficients given by the normalized derivatives, hence

$$\begin{array}{llll} \hat{\mathbf{v}}^l_{i,k} & := \frac{\partial^{p+q}}{\partial x^p \partial y^q}\hat{\mathbf{v}}^n_{i,k} & \text{with} & l \in [1, 2\mathcal{N}], \\ \hat{\mathbf{w}}^l_{i,k+\frac{1}{2}} & := \frac{\partial^{p+q}}{\partial x^p \partial y^q}\hat{w}^n_{i,k+\frac{1}{2}} & \text{with} & l \in [1, \mathcal{N}]. \end{array} \tag{36}$$

The *modal* basis functions are denoted by ψ_l which are from the space of piecewise polynomials $\mathbb{P}^N$. The mono-index l covers all degrees of freedom which sum to $\mathcal{N} = \frac{(N+1)(N+2)}{2}$ for each spatial direction. The unknowns are recovered by solving a linear system composed by a set of equations defined on a certain *reconstruction stencil* that provides the known values, which the reconstruction procedure is built on. For the horizontal normal velocity, the stencil $\mathcal{S}_i^{\lambda}$ is composed by a total number $n_{\mathcal{S}}$ of edges λ_j, while for the vertical velocity we consider entire cells P_j for filling the stencil $\mathcal{S}_i^P$. Typically, on unstructured meshes, the number of items (edges or polygons) in the stencil is bigger than the necessary minimum number $\mathcal{N}$ of unknown expansion coefficients required to obtain the formal order of accuracy $\mathcal{O}(N) = N + 1$. Therefore, we set $n_{\mathcal{S}} = 2\mathcal{N}$, so that *overdetermined* linear systems arise in the context of high order reconstructions on polygonal grids. The reconstruction stencil is obtained by a recursive algorithm which adds neighbor items (either edges or cells) to the stencil until the prescribed number $n_{\mathcal{S}}$ is reached. Specifically, the first item in the stencil is always given by the element P_i under consideration, then the Neumann neighborhood enters the stencil, that is composed by the direct side neighbors surrounding P_i. The stencil search algorithm proceeds by recursively adding Neumann neighbors of the previous neighbors, thus the higher is the reconstruction polynomial degree N the larger is the associated reconstruction stencil. For cells lying on physical boundaries no special care is adopted, therefore the reconstruction stencil is filled in the same manner, resulting in a one-sided stencil because stencil items can be added only where the mesh exists, i.e. the internal part of the computational domain. As usually done for high order finite volume schemes [34, 29], the reconstruction relies on integral conservation of known quantities, thus allowing the reconstruction system to be fed by the following $n_{\mathcal{S}}$ equations for $\bar{\mathbf{v}}^n_{i,k}(x,y)$ and $\bar{\mathbf{w}}^n_{i,k+\frac{1}{2}}(x,y)$:

$$\frac{1}{|\lambda_j|}\int_{\lambda_j} \bar{\mathbf{v}}^n_{i,k}(x,y)\,\cdot\mathbf{n}_j\,ds \;=\; u^n_{j,k} \qquad \forall \lambda_j \in \mathcal{S}^\lambda_i, \tag{37}$$

$$\frac{1}{|P_j|}\int_{P_j} \bar{\mathbf{w}}^n_{i,k+\frac{1}{2}}(x,y)\,dA \;=\; w^n_{j,k+\frac{1}{2}} \qquad \forall P_j \in \mathcal{S}^P_i. \tag{38}$$

Furthermore, for the vertical velocity component, an additional linear constraint is imposed to the system, which requires that equation (38) holds *exactly* on polygon P_i at height $z_{k+\frac{1}{2}}$, that is

$$\frac{1}{|P_i|}\int_{P_i} \bar{\mathbf{w}}^n_{i,k+\frac{1}{2}}(x,y)\,dA = w^n_{i,k+\frac{1}{2}}. \tag{39}$$

All the integrals are computed using Gaussian quadrature rules of suitable order of accuracy, see [35]. The reconstruction systems (37) and (38)-(39) set up above can be written in matrix-vector form as

$$\mathbf{M}_u\mathbf{V} = \mathbf{B}^n_u, \tag{40}$$

$$\left\{ \begin{array}{l} \mathbf{M}_w\mathbf{W} = \mathbf{B}^n_w \\ \mathbf{C}_w\mathbf{W} = \mathbf{D}_w\mathbf{B}^n_w \end{array} \right. , \tag{41}$$

with $\mathbf{B}^n_u$ and $\mathbf{B}^n_w$ representing the known edge- and cell-defined velocity values, that are listed in the corresponding reconstruction stencils $\mathcal{S}^\lambda_i$ and $\mathcal{S}^P_i$. The vectors of unknown expansion coefficients $\mathbf{V}$ and $\mathbf{W}$ contain the reconstructed velocity components $(\hat{u},\hat{v})$ and $\hat{w}$ on the horizontal plane and all higher order normalized spatial derivatives, i.e.

$$\mathbf{V}^T = \quad \hat{\mathbf{v}}^l_{i,k} \quad = \left(\hat{u},\hat{u}_x,\hat{u}_y,\hat{u}_{xx},\hat{u}_{xy},\hat{u}_{yy},\ldots\hat{v},\hat{v}_x,\hat{v}_y,\hat{v}_{xx},\hat{v}_{xy},\hat{v}_{yy}\ldots\right)^n_{i,k}, \tag{42}$$

$$\mathbf{W}^T = \quad \hat{\mathbf{w}}^l_{i,k+\frac{1}{2}} \quad = \left(\hat{w},\hat{w}_x,\hat{w}_y,\hat{w}_{xx},\hat{w}_{xy},\hat{w}_{yy},\ldots\right)^n_{i,k+\frac{1}{2}}. \tag{43}$$

The solution of (40) is directly computed using the *normal equation*

$$\mathbf{V} = \mathbf{M}^T_u\left(\mathbf{M}^T_u\mathbf{M}_u\right)^{-1}\mathbf{B}^n_u := \mathbf{R}_u\,\mathbf{B}^n_u, \tag{44}$$

where the reconstruction matrix $\mathbf{R}_u$ is evaluated once and for all in the preprocessing stage for all cells. On the other hand, the *constrained* least squares (CLSQ) method is adopted for solving system (41), therefore a functional $g(\mathbf{W})$ is defined and minimized by requiring its derivatives to vanish, thus

$$g(\mathbf{W}) \;=\; \left(\mathbf{M}_w\mathbf{W}-\mathbf{B}^n_w\right)^T\left(M_w\mathbf{W}-\mathbf{B}^n_w\right) - \boldsymbol{\mu}^T\left(\mathbf{C}_w\mathbf{W}-\mathbf{D}_w\mathbf{B}^n_w\right), \tag{45}$$

$$\frac{\partial g}{\partial \mathbf{W}} \;=\; 2\mathbf{M}^T_w\mathbf{M}_w\mathbf{W} - 2\mathbf{M}^T_w\mathbf{B}^n_w - \mathbf{C}^T_w\boldsymbol{\mu} = 0, \tag{46}$$

$$\frac{\partial g}{\partial \boldsymbol{\mu}} \;=\; -\left(\mathbf{C}_w\mathbf{W}-\mathbf{D}_w\mathbf{B}^n_w\right) = 0, \tag{47}$$

where $\boldsymbol{\mu}$ is the vector of Lagrange multipliers to enforce the linear constraint. The associated enlarged linear system of normal equations then reads

$$\left(\begin{array}{cc} 2\mathbf{M}^T_w\mathbf{M}_w & -\mathbf{C}^T_w \\ \mathbf{C}_w & \mathbf{0} \end{array}\right)\left(\begin{array}{c} \mathbf{W} \\ \boldsymbol{\mu} \end{array}\right) = \left(\begin{array}{c} 2\mathbf{M}^T_w \\ \mathbf{D}_w \end{array}\right)\mathbf{B}^n_w, \tag{48}$$

whose solution can be simply obtained as

$$\left(\begin{array}{c} \mathbf{W} \\ \boldsymbol{\mu} \end{array}\right) = \left(\begin{array}{cc} 2\mathbf{M}^T_w\mathbf{M}_w & -\mathbf{C}^T_w \\ \mathbf{C}_w & \mathbf{0} \end{array}\right)^{-1}\left(\begin{array}{c} 2\mathbf{M}^T_w \\ \mathbf{D}_w \end{array}\right)\mathbf{B}^n_w := \mathbf{R}^L\,\mathbf{B}^n_w. \tag{49}$$

We are only interested in the solution of the expansion coefficients $\mathbf{W}$, therefore the reconstruction matrix $\mathbf{R}_w$ for the vertical velocity on plane $x-y$ is a subset of the CLSQ matrix $\mathbf{R}^L$, that is $\mathbf{R}_w := \mathbf{R}^L_{[\mathcal{N}\times n_{\mathcal{S}}]}$.

Once the reconstruction matrices have been computed in the preprocessing stage for all cells, at each time step t^n the reconstruction step on the horizontal plane is efficiently carried out by two matrix-vector products, namely

$$\mathbf{V}^n = \mathbf{R}_u\,\mathbf{B}^n_u, \qquad \mathbf{W}^n = \mathbf{R}_w\,\mathbf{B}^n_w. \tag{50}$$

Remark on the total water depth. The *same* reconstruction matrices $\mathbf{R}_u$ and $\mathbf{R}_w$ can be directly used to obtain a high order description of the bottom topography b_j and the free surface elevation η_i, respectively. The corresponding unknown expansion coefficients, $\mathbf{b} = \left(\hat{b}, \hat{b}_x, \hat{b}_y, \hat{b}_{xx}, \hat{b}_{xy}, \hat{b}_{yy}, \ldots\right)$ and $\boldsymbol{\eta} = (\hat{\eta}, \hat{\eta}_x, \hat{\eta}_y, \hat{\eta}_{xx}, \hat{\eta}_{xy}, \hat{\eta}_{yy}, \ldots)$, are simply evaluated as

$$\mathbf{b}(x,y) = \mathbf{R}_u\,\mathbf{B}_b, \qquad \boldsymbol{\eta}^n(x,y) = \mathbf{R}_w\,\mathbf{B}^n_\eta. \tag{51}$$

Therefore, the total water depth is easily and accurately evaluated both on each edge λ_j and at each polygonal centroid $\mathbf{c}_i$ by

$$H^n_j = \max\left[0,\, b_j + \boldsymbol{\eta}^n(x_j,y_j)\right], \qquad H^n_i = \max\left[0,\, \mathbf{b}(x_i,y_i) + \eta^n_i\right]. \tag{52}$$

Reconstruction along the vertical direction In order to obtain a continuous high order polynomial even along the $z-$axis at a generic coordinate z, a classical one-dimensional Lagrange interpolation is used. The staggered discretization implies that the horizontal reconstruction for the normal velocity u is defined at height z_k, while at $z_{k+\frac{1}{2}}$ the high order polynomials for w are reconstructed, according to Figure 4. Therefore, depending on the chosen reconstruction degree N, the *vertical reconstruction stencil* is composed by a total number of $N+1$ layers, that are selected starting from the kth layer containing the z coordinate at which the vertical reconstruction has to be carried out. For the sake of clarity and without loss of generality, if $N=3$ the vertical reconstruction stencil for w is given by $S^w_{i,k} = \left(k-\frac{3}{2}, k-\frac{1}{2}, k+\frac{1}{2}, k+\frac{3}{2}\right)$, and the corresponding stencil for u results in $S^v_{i,k} = (k-1, k, k+1, k\pm 2)$. The choice of the last stencil point $k\pm 2$ is taken according to an upwind choice that is based on the local interpolation of the vertical velocity w, thus if $w<0$ the stencil is filled with $k+2$, otherwise layer $k-2$ is used. Close to the bottom and the free surface, the vertical stencils are filled with layers containing water, thus obtaining fully upward and downward pointing stencils for $k \in [m, m+N]$ and $k \in [M^n - N, M^n]$, respectively.

Let $L^{v,w}_{k+r}(z)$ be the Lagrange polynomials passing through the points z_{k+r} and let $\bar{\mathbf{w}}^n_{i,k+r-\frac{1}{2}}(x,y)$ and $\bar{\mathbf{v}}^n_{i,k+r}(x,y)$ denote the high order velocities on plane $x-y$ obtained as previously described. The vertical reconstruction polynomial for a cell P_i, which gives a *fully* three-dimensional representation of the velocity field, is then given by

$$\mathbf{w}_{i,k}(\mathbf{x}) = \sum_{r\in\mathcal{S}^w_{i,k}} L^w_{k+r}(z)\,\bar{\mathbf{w}}^n_{i,k+r-\frac{1}{2}}(x,y), \qquad \mathbf{u}_{i,k}(\mathbf{x}) = \sum_{r\in\mathcal{S}^{v,\pm}_{i,k}} L^v_{k+r}(z)\,\bar{\mathbf{v}}^n_{i,k+r}(x,y), \tag{53}$$

with

$$L^{v,w}_{k+r}(z) = \frac{\prod\limits_{m\neq r}(z - z_{k+m})}{\prod\limits_{m\neq r}(z_{k+r} - z_{k+m})}, \qquad \forall\, m, r \in \mathcal{S}^{v,w}_{i,k}. \tag{54}$$

Divergence-free reconstruction The reconstruction polynomials obtained so far constitute a *preliminary* result, and they must be modified in order to preserve the divergence-free constraint dictated at the continuous level by the continuity equation (2). Indeed, at point $\mathbf{x}_P = (x_P, y_P, z_P)$ the high order polynomials $\mathbf{w}(\mathbf{x}_P)$ and $\mathbf{u}(\mathbf{x}_P)$ in (53) do in general yield the inequality

$$\frac{\partial}{\partial x}\mathbf{u}(\mathbf{x}_P) + \frac{\partial}{\partial y}\mathbf{u}(\mathbf{x}_P) + \frac{\partial}{\partial z}\mathbf{w}(\mathbf{x}_P) \neq 0. \tag{55}$$

To fulfill the constraint, we rely on a *constrained L_2 projection* [36] that aims at *slightly* modifying the degrees of freedom $\mathbf{V}^T$ for the horizontal velocity previously evaluated from (40). More precisely, the sought divergence-free polynomial $\mathbf{v}(\mathbf{x})$ has to be as close as possible to the original reconstruction polynomial $\bar{\mathbf{v}}(\mathbf{x})$ and has to satisfy the following two requirements:

- the velocity field has to be *locally* and *globally* divergence-free, thus $\mathbf{v}(\mathbf{x})$ together with $\mathbf{w}(\mathbf{x})$ must obey the continuity equation at each and every point inside the three-dimensional control volume $P_{i,k} = \left[P_i \times \Delta z_{i,k}^n\right]$;
- the normal velocity components $u_{j,k}$ defined on each edge $\lambda_{j,k}$ must be *exactly* reconstructed when evaluating $\mathbf{v}(x_{j,k}, y_{j,k})$. This condition has to be imposed on all edges of polygon P_i except one because if the velocity field is locally divergence-free, Gauss' theorem ensures that the boundary integral is automatically divergence-free.

Therefore, the divergence-free reconstruction is obtained by *minimizing*

$$\int\limits_{P_{i,k}} \left(\mathbf{v}_{i,k}^n(x,y) - \bar{\mathbf{v}}_{i,k}^n(x,y)\right)^2 dA, \tag{56}$$

under the following constraints:

$$\begin{aligned} \int\limits_{P_{i,k}} \psi_q^{'} \left(\frac{\partial}{\partial x}\mathbf{v}_{i,k}^n(x,y) + \frac{\partial}{\partial y}\mathbf{v}_{i,k}^n(x,y)\right) dA &= -\int\limits_{P_{i,k}} \psi_q^{'} \frac{\partial}{\partial z}\mathbf{w}(x,y,z_k)\, dA, & (57) \\ \int_{\lambda_j \in P_i^{\subset}} \frac{1}{|\lambda_j|}\mathbf{v}_{i,k}^n(x,y) \cdot \mathbf{n}_j \, ds &= u_{j,k}^n \qquad \forall \lambda_j \in P_i^{\subset}, & (58) \end{aligned}$$

with $P_i^{\subset}$ denoting an arbitrary subset of edges of cell P_i excluding one of them and $\psi_q^{'}$ representing a set of test functions (which are chosen to be equal to the Taylor basis functions) belonging to the space of piecewise polynomials of degree $N-1$. In our approach only the horizontal degrees of freedom $\mathbf{V}_l^T$ in $\bar{\mathbf{v}}_{i,k}^n(x,y)$ are modified, while the vertical expansion coefficients $\mathbf{W}^T$ are shifted to the right hand side of (57), hence they remain the same obtained after the reconstruction (49). Note that the right hand side of Equation 57 contains the derivative along the vertical direction $\frac{\partial}{\partial z}$, that has to be approximated using a Lagrange interpolation polynomial for $\mathbf{w}(x,y,z_k)$ with degree $N+1$ in (53), in order to guarantee and maintain the formal accuracy of the horizontal reconstruction. Adopting a Lagrangian multiplier technique [29] as done for the horizontal reconstruction of the vertical velocity (49), the *divergence-free reconstruction functional* $\mathcal{G}$ to be minimized takes the form

$$\begin{aligned} \mathcal{G} &= \int\limits_{P_{i,k}} \left(\mathbf{v}_{i,k}^n(x,y) - \bar{\mathbf{v}}_{i,k}^n(x,y)\right)^2 dA \\ &- \boldsymbol{\mu}\left(\int\limits_{P_{i,k}} \psi_q^{'} \left(\frac{\partial}{\partial x}\mathbf{v}_{i,k}^n(x,y) + \frac{\partial}{\partial y}\mathbf{v}_{i,k}^n(x,y)\right) dA + \int\limits_{P_{i,k}} \psi_q^{'} \frac{\partial}{\partial z}\mathbf{w}(x,y,z_k)\, dA\right) \\ &- \boldsymbol{\gamma}\left(\int_{\lambda_j \in P_i^{\subset}} \frac{1}{|\lambda_j|}\mathbf{v}_{i,k}^n(x,y) \cdot \mathbf{n}_j \, ds - u_{j,k}^n\right), \end{aligned} \tag{59}$$

leading to the following linear system for the unknown *divergence-free degrees of freedom* $\hat{\mathbf{u}}^l(P_{i,k})$ as well as for the Lagrange multipliers $\boldsymbol{\mu}$ and $\boldsymbol{\gamma}$:

$$\begin{aligned}
\frac{\partial \mathcal{G}}{\partial \hat{\mathbf{u}}^l} &= \left(2 \int\limits_{P_{i,k}} \psi_q \psi_l \, dA \right) \left(\hat{\mathbf{u}}^l - \hat{\mathbf{v}}^l \right) - \boldsymbol{\mu} \left(\int\limits_{P_{i,k}} \psi_q^{'} \nabla \psi_l \, dA \right) - \boldsymbol{\gamma} \left(\int_{\lambda_j \in P_i^{\subset}} \frac{1}{|\lambda_j|} \psi_l \, \mathbf{n}_j \, ds \right) = 0, \\
\frac{\partial \mathcal{G}}{\partial \boldsymbol{\mu}} &= \left(\int\limits_{P_{i,k}} \psi_q^{'} \nabla_{xy} \psi_l \, dA \right) \cdot \hat{\mathbf{u}}^l + \left(\int\limits_{P_{i,k}} \psi_q^{'} \, \nabla_z \psi_l \, dA \right) \cdot \hat{\mathbf{w}}^l = 0, \\
\frac{\partial \mathcal{G}}{\partial \boldsymbol{\gamma}} &= \left(\int\limits_{P_{i,k}} \frac{1}{|\lambda_j|} \psi_l \cdot \mathbf{n}_j \, ds \right) \cdot \hat{\mathbf{u}}^l - u_{j,k}^n = 0, \qquad (60)
\end{aligned}$$

with ψ_l denoting the Taylor basis functions used in (34)- (35) as well as for the approximation of $\mathbf{v}(\mathbf{x})$. The above system can be written in matrix-vector notation and cast into form (49), then solved with a constrained least squares method as done for the CLSQ system for the vertical velocity. At this point, the vector of expansion coefficients $\mathbf{V}^n$ in (50) for the horizontal velocity is *overwritten* and assigned the new values $\hat{\mathbf{u}}^l$ computed from the solution of (60).

4.1. Operator F^n: nonlinear convection and horizontal diffusion

The explicit operators $Fu_{j,k}^n$ and $Fw_{i,k+\frac{1}{2}}^n$ appearing in the discrete momentum equations (8) and (9) are computed via a semi-Lagrangian discretization for the convective terms and a finite volume formulation for the horizontal diffusion, which will be briefly discussed hereafter.

Convection Following the work presented in [27], the Lagrangian trajectory of a flow particle located at a generic spatial point $\mathbf{x} = (x, y, z)$ is described by the ODE system

$$\frac{d\mathbf{x}}{d\tau} = -\bar{\mathbf{u}}(\mathbf{x}), \qquad \text{with} \qquad \tau \in [0, \Delta t], \qquad (61)$$

where τ is the rescaled time w.r.t. the global time step Δt. The velocity vector $\bar{\mathbf{u}}(\mathbf{x}) = (\mathbf{v}(x), w(\mathbf{x}))$ is evaluated relying on the high order three-dimensional divergence-free velocity field which is available from the reconstruction procedure discussed in the previous section. The initial condition of (61) is $\mathbf{x}(0) = \mathbf{x}_{j,k}^n$ for the edge based operator $Fu_{j,k}^n$, whereas it reads $\mathbf{x}(0) = \mathbf{x}_{i,k+\frac{1}{2}}^n$ for $Fw_{i,k+\frac{1}{2}}^n$. The ODE system (61) is then approximated by means of a high order Taylor expansion, leading to the new point position at local time τ_{l+1}:

$$\mathbf{x}_{l+1} = \mathbf{x}_l + \Delta\tau \frac{d\mathbf{x}}{d\tau} + \frac{\Delta\tau^2}{2} \frac{d^2\mathbf{x}}{d\tau^2} + \frac{\Delta\tau^3}{6} \frac{d^3\mathbf{x}}{d\tau^3} + (\mathcal{O}4) \quad \text{with} \quad \Delta\tau = c \, min \left(\frac{d_i}{|\mathbf{v}(\mathbf{x}_l)|}, \frac{\Delta z_k^n}{|w(\mathbf{x}_l)|} \right), \qquad (62)$$

where the local time step $\Delta\tau$ is determined under a local CFL condition with $c = 0.5$, by considering the minimum between a local horizontal and vertical restriction. We use the current element P_i with diameter d_i and layer k in which the current point $\mathbf{x}_l$ is located, and the velocity vector evaluated at point $\mathbf{x}_l$ for determining $\Delta\tau$ in (62). At the aid of the chain rule, the time derivatives in (62) are replaced by spatial derivatives that are readily computed from the high order reconstruction polynomials $\mathbf{v}(\mathbf{x})$ and $\mathbf{w}(\mathbf{x})$, hence obtaining the following method for the backward integration of the trajectory equation (61):

$$\mathbf{x}_{l+1} = \mathbf{x}_l - \Delta\tau \bar{\mathbf{u}}_l + \frac{\Delta\tau^2}{2} \left(\frac{\partial \bar{\mathbf{u}}}{\partial \mathbf{x}} \bar{\mathbf{u}} \right)_l - \frac{\Delta\tau^3}{6} \left(\frac{\partial^2 \bar{\mathbf{u}}}{\partial \mathbf{x}^2} \bar{\mathbf{u}} \bar{\mathbf{u}} + \left(\frac{\partial \bar{\mathbf{u}}}{\partial \mathbf{x}} \right)^2 \bar{\mathbf{u}} \right)_l + (\mathcal{O}4). \qquad (63)$$

The integration stops either when $\tau = \Delta t$ or when a boundary of the computational domain $\Omega(t^n)$ is reached by the trajectory. If no boundaries are crossed, the foot of the characteristics is simply given by the last position vector $\mathbf{x}^L(\tau) = \mathbf{x}(\Delta t)$, otherwise we take the intersection $\mathbf{x}_b$ between the boundary edge and the particle trajectory, i.e. $\mathbf{x}^L(\tau) = \mathbf{x}_b$. The order of accuracy for the Taylor expansion (62) is chosen to be the same of the order adopted in the reconstruction procedure, so that all high order terms can be easily evaluated.

Horizontal viscosity In order to discretize the horizontal viscosity terms, a conservative finite volume method is employed together with the numerical viscous flux proposed in [37]. Such a discretization holds for each polygon P_i and each layer k, and both velocities and gradients are computed again using the high order velocity field reconstruction:

$$\nabla_i^2 \bar{\mathbf{u}}_k = \frac{1}{|P_i|} \int\limits_{\partial P_i} \nabla_h \bar{\mathbf{u}}_k \cdot \mathbf{n}_j \, ds \quad \text{with} \quad \nabla_h \bar{\mathbf{u}}_k \cdot \mathbf{n}_j = \frac{1}{2} \left(\nabla \bar{\mathbf{u}}_k^+ + \nabla \bar{\mathbf{u}}_k^- \right) \cdot \mathbf{n}_j + \frac{1}{\delta_j \sqrt{\frac{1}{2}\pi}} \left(\bar{\mathbf{u}}_k^+ - \bar{\mathbf{u}}_k^- \right), \tag{64}$$

where $\bar{\mathbf{u}}_k^-$ and $\bar{\mathbf{u}}_k^+$ denote the high order boundary extrapolated data on ∂P_i from within cell P_i and from the neighbor polygon, respectively. The viscosity term for feeding the operators F^n is taken in those element containing the foot of the characteristics $\mathbf{x}^L$, i.e. $\nabla^2 \bar{\mathbf{u}}(\mathbf{x}^L)$, previously determined by the backward integration of the Lagrangian trajectory.

5. NUMERICAL RESULTS

Several test problems are presented and discussed in order to validate the numerical scheme proposed in this work. In total, five different numerical experiments have been chosen, each one targeting a specific part of the algorithm:

1. high order velocity field reconstruction (Section 4): convergence studies with and without the novel divergence-free technique on a sequence of successively refined computational meshes;
2. nonlinear convective terms (Section 4.1): integration of circular trajectories on the horizontal $x-y$ plane as well as along the vertical $x-z$ plane;
3. horizontal viscosity terms (Section 4.1): first problem of Stokes;
4. solution of the hydrostatic pressure component (Section 3.2): steady vortex flow with a perfect balance between hydrostatic pressure gradient and centrifugal force;
5. non-hydrostatic pressure correction (Section 3.2): oscillating basin with a non-negligible vertical velocity.

All simulations are carried out with the implicitness factor $\theta = 0.55$ and a divergence-free reconstruction degree of $N = 2$, if not explicitly written.

5.1. Convergence studies

The accuracy of the high order divergence-free reconstruction presented in this paper is assessed by considering a smooth velocity field which simply reads:

$$\begin{aligned} u &= \sin\left(\frac{\pi}{2}x\right) \cos\left(\frac{\Pi}{2}y\right) \cos\left(\frac{\Pi}{2}z\right), \\ v &= \cos\left(\frac{\pi}{2}x\right) \sin\left(\frac{\Pi}{2}y\right) \cos\left(\frac{\Pi}{2}z\right), \\ w &= \cos\left(\frac{\pi}{2}x\right) \cos\left(\frac{\Pi}{2}y\right) \sin\left(\frac{\Pi}{2}z\right) (-2). \end{aligned} \tag{65}$$

The computational domain is the unit cube $\Omega = [0, 1]^3$ and is discretized with a fixed number of *active* vertical layers $N_z = 25$. Each horizontal layer is then paved with an unstructured Voronoi mesh and a sequence of four successively refined grids is considered. Once the reconstruction procedure has been carried out, the errors ϵ are evaluated on the entire computational domain, i.e. for each polygon $P_i \in [1, N_P]$ lying on each layer $k \in [1, N_z]$ the L_1, L_2 and L_∞ norms are given by

$$\begin{aligned} \epsilon_{L_1} &= \int_\Omega \|(s_e(\mathbf{x}) - s_h((\mathbf{x})))\| \, dA, \qquad \epsilon_{L_2} = \sqrt{\int_\Omega \|(s_e(\mathbf{x}) - s_h((\mathbf{x})))\|^2 \, dA}, \\ \epsilon_{L_\infty} &= \max_{P_i} \|(s_e(\mathbf{x}) - s_h((\mathbf{x})))\| \, , \end{aligned} \tag{66}$$

where s represents a generic velocity component, while $s_e(\mathbf{x})$ and $s_h(\mathbf{x})$ denote the exact and the numerical (reconstructed) solution, respectively. Table I shows the convergence rates for the high order *divergence-free* reconstruction, while Table II reports the results for a high order velocity reconstruction without the divergence-free algorithm, as done in [27].

In both cases the desired order of accuracy is achieved for $N = 2$ and $N = 3$, but the divergence-free reconstruction systematically generates a lower error in terms of absolute value for all norms, except the L_∞ with $N = 3$, as depicted in Figure 5.

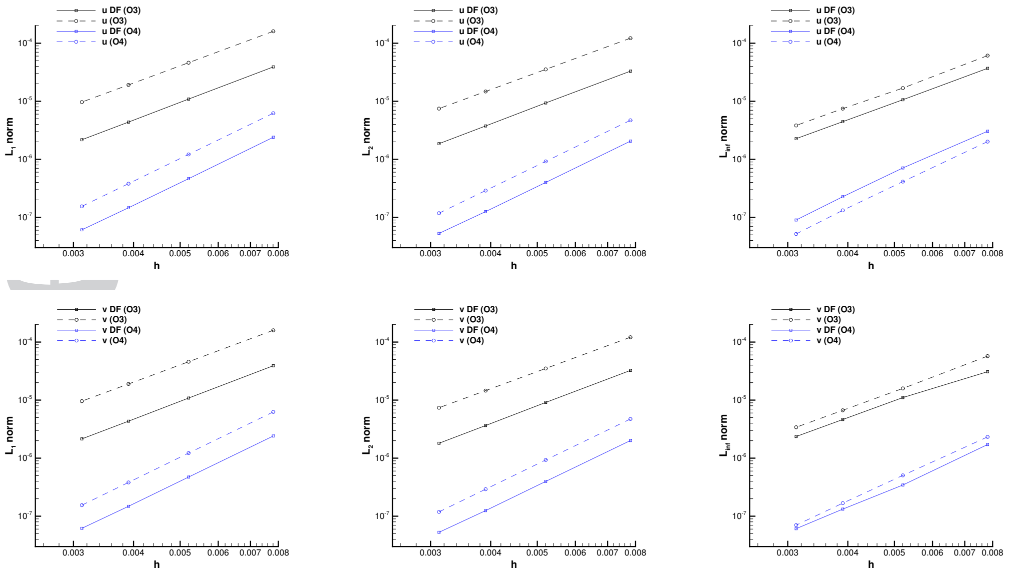

Figure 5. Dependency of the error norms (L_1, L_2, L_∞) reported in Table II on the mesh size for $N = 2$ (drawn in black) and $N = 3$ (drawn in blue). The divergence-free results (solid lines) are addressed with "DF" and they are compared against the error norms obtained without using the divergence-free reconstruction technique (dashed lines).

Furthermore, we explicitly want to demonstrate that our new reconstruction technique is *divergence-free up to machine precision*, therefore we have also computed the integral average of the divergence of the velocity field over the whole computational domain, that is

$$\mathbf{div}_h(\bar{\mathbf{u}}) = \int\limits_{P_{i,k}} \left(\frac{\partial \psi_l}{\partial x} \hat{\mathbf{v}}^l + \frac{\partial \psi_l}{\partial y} \hat{\mathbf{v}}^l + \frac{\partial \psi_l}{\partial z} \hat{\mathbf{w}}^l \right) dA \qquad \forall P_i \in [1, N_P], \quad \forall k \in [1, N_z]. \tag{67}$$

Table III clearly demonstrates the capability of our new reconstruction to obtain a fully divergence-free velocity field, contrarily to what can be achieved by simply applying a high order reconstruction technique without any special care about the discrete divergence operator.

Table I. Numerical convergence rates obtained with third and fourth order accurate reconstruction **with** divergence-free constraints. Errors in L_1, L_2 and L_∞ are reported for all velocity components u, v, w. The characteristic mesh size is $h(\Omega) = \max_{P_i} \sqrt{|P_i|}$.

$\mathbf{N = 2}$

horizontal velocity u

$h(\Omega)$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$
7.81E-03	3.9061E-05	-	3.3009E-05	-	3.6911E-05	-
5.21E-03	1.0901E-05	3.1	9.3511E-06	3.1	1.0619E-05	3.1
3.90E-03	4.3764E-06	3.2	3.7546E-06	3.2	4.4496E-06	3.0
3.12E-03	2.1732E-06	3.1	1.8592E-06	3.1	2.2676E-06	3.0

horizontal velocity v

$h(\Omega)$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$
7.81E-03	3.9107E-05	-	3.2459E-05	-	3.0867E-05	-
5.21E-03	1.0840E-05	3.2	9.1198E-06	3.1	1.1060E-05	2.5
3.90E-03	4.3332E-06	3.2	3.6474E-06	3.2	4.6291E-06	3.0
3.12E-03	2.1481E-06	3.1	1.8056E-06	3.2	2.3574E-06	3.0

vertical velocity w

$h(\Omega)$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$
7.81E-03	3.2521E-04	-	2.4981E-04	-	1.3927E-04	-
5.21E-03	9.3169E-05	3.1	7.2113E-05	3.1	4.2367E-05	2.9
3.90E-03	3.8609E-05	3.1	2.9975E-05	3.1	1.8185E-05	2.9
3.12E-03	1.9564E-05	3.0	1.5211E-05	3.0	9.5321E-06	2.9

$\mathbf{N = 3}$

horizontal velocity u

$h(\Omega)$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$
7.81E-03	2.3978E-06	-	2.0559E-06	-	3.0432E-06	-
5.21E-03	4.6373E-07	4.1	3.9941E-07	4.0	7.0964E-07	3.6
3.90E-03	1.4616E-07	4.0	1.2562E-07	4.0	2.2667E-07	4.0
3.12E-03	6.0971E-08	3.9	5.2869E-08	3.9	9.0077E-08	4.1

horizontal velocity v

$h(\Omega)$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$
7.81E-03	2.4080E-06	-	2.0114E-06	-	1.7160E-06	-
5.21E-03	4.7213E-07	4.0	3.9734E-07	4.0	3.4434E-07	4.0
3.90E-03	1.4808E-07	4.0	1.2482E-07	4.0	1.3235E-07	3.3
3.12E-03	6.1823E-08	3.9	5.2787E-08	3.9	6.0953E-08	3.5

vertical velocity w

$h(\Omega)$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$
7.81E-03	1.3919E-05	-	1.2097E-05	-	1.0554E-05	-
5.21E-03	2.5474E-06	4.2	2.2506E-06	4.1	2.6183E-06	3.4
3.90E-03	7.9034E-07	4.1	7.0286E-07	4.0	7.4232E-07	4.4
3.12E-03	3.2785E-07	3.9	2.9228E-07	3.9	3.0097E-07	4.0

Table II. Numerical convergence rates obtained with third and fourth order accurate reconstruction **without** divergence-free constraints. Errors in L_1, L_2 and L_∞ are reported for all velocity components u, v, w. The characteristic mesh size is $h(\Omega) = \max_{P_i} \sqrt{|P_i|}$.

$\mathbf{N = 2}$

horizontal velocity u

$h(\Omega)$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$
7.81E-03	1.6048E-04	-	1.2212E-04	-	6.1184E-05	-
5.21E-03	4.6141E-05	3.1	3.5407E-05	3.1	1.6865E-05	3.2
3.90E-03	1.9126E-05	3.1	1.4738E-05	3.0	7.4675E-06	2.8
3.12E-03	9.6775E-06	3.1	7.4755E-06	3.0	3.8281E-06	3.0

horizontal velocity v

$h(\Omega)$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$
7.81E-03	1.5954E-04	-	1.2113E-04	-	5.7251E-05	-
5.21E-03	4.5796E-05	3.1	3.5083E-05	3.1	1.5912E-05	3.2
3.90E-03	1.8956E-05	3.1	1.4574E-05	3.1	6.6745E-06	3.0
3.12E-03	9.5921E-06	3.1	7.3902E-06	3.0	3.4053E-06	3.0

vertical velocity w

$h(\Omega)$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$
7.81E-03	3.2521E-04	-	2.4981E-04	-	1.3927E-04	-
5.21E-03	9.3169E-05	3.1	7.2113E-05	3.1	4.2367E-05	2.9
3.90E-03	3.8609E-05	3.1	2.9975E-05	3.1	1.8185E-05	2.9
3.12E-03	1.9564E-05	3.0	1.5211E-05	3.0	9.5321E-06	2.9

$\mathbf{N = 3}$

horizontal velocity u

$h(\Omega)$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$
7.81E-03	6.2277E-06	-	4.7003E-06	-	2.0167E-06	-
5.21E-03	1.2131E-07	4.0	9.2033E-07	4.0	4.1289E-07	3.9
3.90E-03	3.8013E-07	4.0	2.8912E-07	4.0	1.3210E-07	4.0
3.12E-03	1.5479E-07	4.0	1.1796E-07	4.0	5.1742E-08	4.2

horizontal velocity v

$h(\Omega)$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$
7.81E-03	6.2584E-06	-	4.7268E-06	-	2.3223E-06	-
5.21E-03	2.5474E-06	4.2	9.3110E-07	4.0	5.0537E-07	3.8
3.90E-03	7.9034E-07	4.1	2.9177E-07	4.0	1.6783E-07	3.8
3.12E-03	3.2785E-07	3.9	1.1868E-07	4.0	7.0099E-08	3.9

vertical velocity w

$h(\Omega)$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$
7.81E-03	1.3919E-05	-	1.2097E-05	-	1.0554E-05	-
5.21E-03	2.5474E-06	4.2	2.2506E-06	4.1	2.6183E-06	3.4
3.90E-03	7.9034E-07	4.1	7.0286E-07	4.0	7.4232E-07	4.4
3.12E-03	3.2785E-07	3.9	2.9228E-07	3.9	3.0097E-07	4.0

Table III. Errors in L_1, L_2 and L_∞ norm of the integral average of the divergence of the velocity field computed using the third and fourth order accurate reconstruction with and without the divergence-free constraints.

Reconstruction **with** divergence-free constraints

	$\mathbf{N=2}$			$\mathbf{N=3}$		
$h(\Omega)$	ϵ_{L_1}	ϵ_{L_2}	ϵ_{L_∞}	ϵ_{L_1}	ϵ_{L_2}	ϵ_{L_∞}
7.81E-03	3.4866E-15	1.0760E-16	2.2118E-17	5.5908E-15	1.5331E-16	1.9461E-17
5.21E-03	4.9333E-15	1.0249E-16	1.0286E-17	7.3968E-15	1.3998E-16	1.1953E-17
3.90E-03	6.4804E-15	1.0340E-16	9.8662E-18	9.4952E-15	1.3571E-16	1.0991E-17
3.12E-03	8.2151E-15	1.0766E-16	7.3319E-18	1.1326E-14	1.3048E-16	7.3319E-18

Reconstruction **without** divergence-free constraints

	$\mathbf{N=2}$			$\mathbf{N=3}$		
$h(\Omega)$	ϵ_{L_1}	ϵ_{L_2}	ϵ_{L_∞}	ϵ_{L_1}	ϵ_{L_2}	ϵ_{L_∞}
7.81E-03	1.1441E-03	2.9545E-05	2.0314E-06	1.0354E-04	2.2937E-06	1.1829E-07
5.21E-03	4.3618E-04	7.4789E-06	3.8500E-07	3.0019E-05	4.4344E-07	1.5588E-08
3.90E-03	2.1157E-04	2.7161E-06	9.8958E-08	1.2968E-05	1.4502E-07	4.4339E-09
3.12E-03	1.3533E-04	1.4530E-06	3.7736E-08	7.0176E-06	6.7277E-08	2.8425E-09

5.2. Integration of circular trajectories

The assessment of the high order Taylor method (63) for the backward integration of the Lagrangian trajectories is carried out by prescribing a rigid body rotation as velocity field, which reads

$$\bar{\mathbf{u}}(\mathbf{x}) = \omega \times \mathbf{x}, \tag{68}$$

where the angular velocity vector ω is chosen to be either $\omega = (0, 0, 1)$ or $\omega = (0, 1, 0)$ for a circular rotation on the $x - y$ plane or on the $x - z$ plane, respectively. The final time is set to $t_f = 2\pi$, so that one complete period of rotation is considered and the final point of the trajectory exactly matches the starting point. In other words, starting from a generic point $\mathbf{x}_P \in \Omega$, the exact solution is simply given by a closed circle, which is numerically evaluated using the divergence-free reconstruction technique together with the Taylor method for the integration of the trajectory. The computational domain is the box $\Omega = [-0.5; 0.5] \times [-0.5; 0.5] \times [0; 1]$, which is discretized by using a uniform vertical mesh spacing of $\Delta z = 0.1$ and a total number of polygons $N_P = 1177$ for each layer, as shown in Figure 6.

Figure 7 depicts the numerical trajectories obtained for both test problems with a comparison against the exact solution. An excellent agreement is achieved for the third order scheme, while if a second order Taylor method is used in (63), the numerical trajectory significantly deviates from the analytical path. Thus, high order of accuracy is a key ingredient for obtaining a better resolution in the discretization of the nonlinear convective terms of the governing equations (2)-(6).

5.3. First problem of Stokes

The aim of this test case is to validate the discretization of the horizontal viscosity terms given by (64). The first problem of Stokes considers an infinite plate which is initially at rest. Suddenly, the plate is subject to an acceleration with constant speed parallel to the plate itself. To simulate this problem, a coordinate system which moves with the plate is chosen, therefore the equivalent problem results in solving an initially uniform flow with constant velocity field $\bar{\mathbf{u}}(\mathbf{x}) = (1, 0, 0)$, in which a flat plate at rest is located at time $t = 0$. The computational domain is the box $\Omega = [0; 1] \times [0; 0.5] \times [0; 1]$ and is discretized with $N_P = 3469$ Voronoi cells and a total number of $N_z = 5$ vertical layers. According to [38], the analytical solution reduces to a linear scalar diffusion

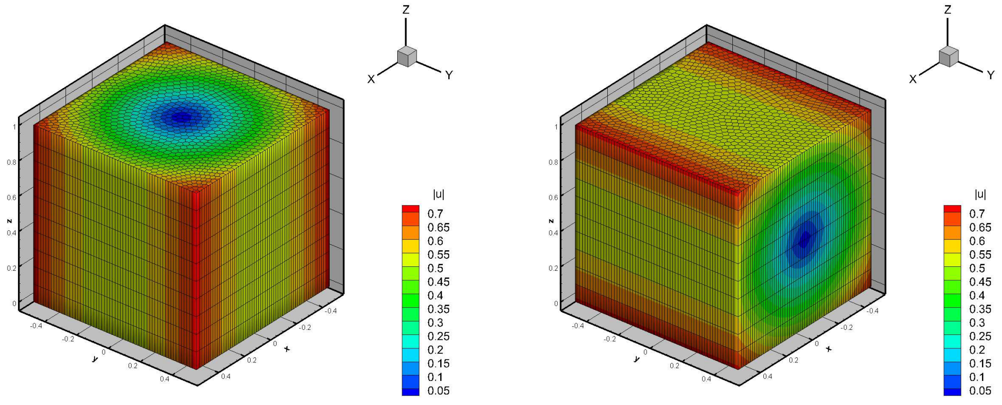

Figure 6. Computational mesh and absolute value of the velocity for the circular flow on the $x - y$ plane (left) and on the $x - z$ plane (right).

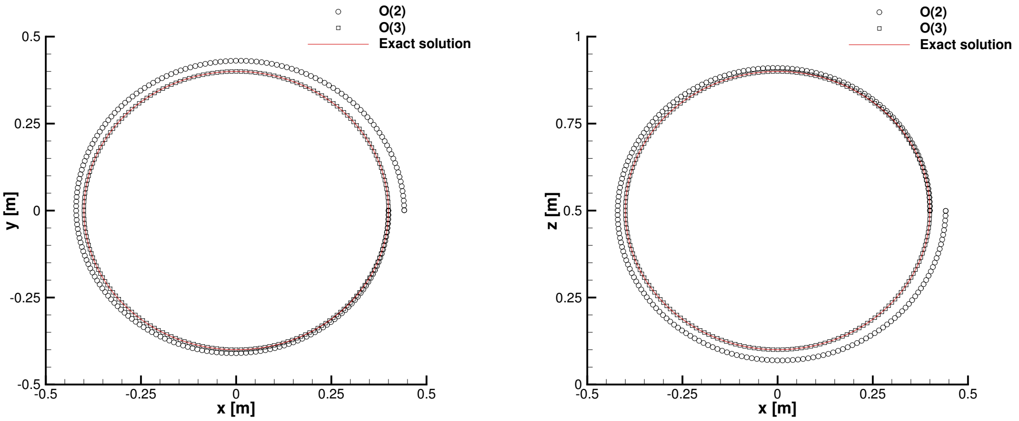

Figure 7. Trajectories for the circular flow on the $x - y$ plane (left) and on the $x - z$ plane (right) with starting point $\mathbf{x}_P = (0.4, 0, 0.5)$. Second and third order results are reported and compared against the exact solution highlighted by a solid line.

equation for the horizontal velocity u:

$$\frac{\partial u}{\partial t} + v\frac{\partial^2 u}{\partial y^2} = 0 \qquad u(x, 0, t) = 0, \tag{69}$$

with the exact solution

$$u = \text{erf}\left(\frac{y}{2\sqrt{\nu t}}\right), \tag{70}$$

where the horizontal viscosity coefficient is $\nu = 10^{-3}$. The initial free surface elevation is $\eta = 1$ and the simulation is carried out until a final time of $t = 0.5$ with second and third order of accuracy. Figure 8 depicts the horizontal velocity profile along the $y-$direction at $x = 0.5$ obtained with second and third order numerical schemes. The higher accurate scheme generates better results which are closer to the analytical solution, as expected. Figure 9 shows a three-dimensional view of the third order accurate horizontal velocity component u on the entire computational domain at different output times.

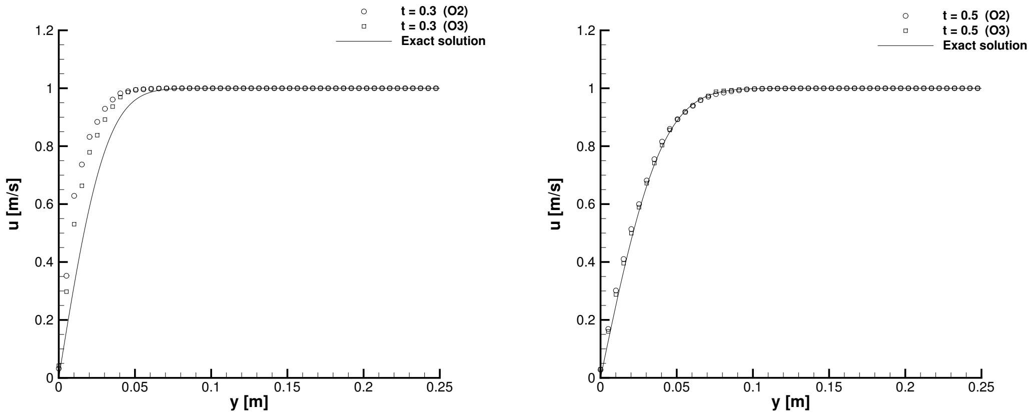

Figure 8. Second and third order results compared against exact solution for the first problem of Stokes. Horizontal velocity component along the $y-$direction at $x = 0.5$ is plot at output time $t = 0.3$ (left) and $t = 0.5$ (right).

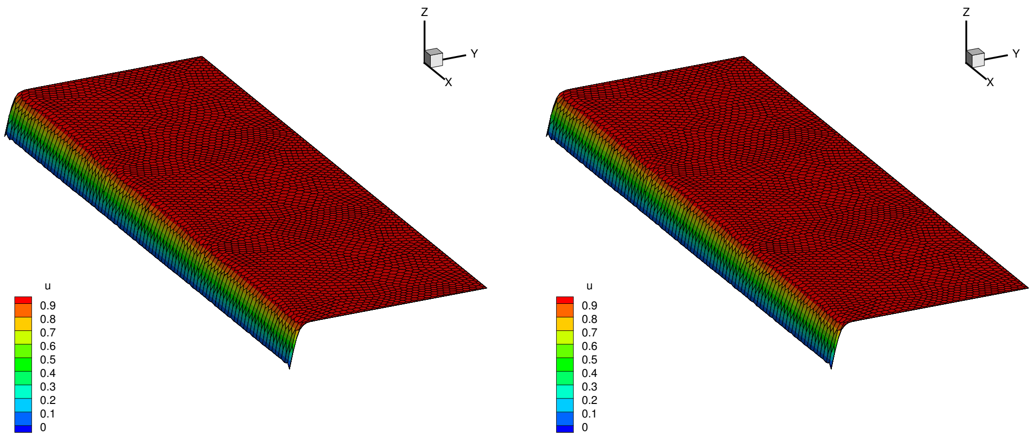

Figure 9. Third order horizontal velocity distribution for the first problem of Stokes at output time $t = 0.3$ (left) and $t = 0.5$ (right).

To compare the results with the work presented in [27], the first problem of Stokes is simulated again with $N = 2$ but using an implicitness factor of $\theta = 1$, hence recovering a first order accurate scheme in time. A comparison against the exact solution is depicted in Figure 10, in which a better matching is achieved w.r.t. the results obtained with $\theta = 0.55$ shown in Figure 8. That result is due to the effect of the numerical viscosity and is *not* related to the physical kinematic viscosity. Furthermore, on the right panel of Figure 10 we also show the discrete divergence of the velocity field computed according to (67) for all the simulations of the first problem of Stokes. The divergence-free reconstruction proposed in this work maintains the velocity field with divergence at machine accuracy for all computations, while for classical high order reconstruction schemes the discrete divergence ranges in the interval $[0, -0.17]$, never reaching the divergence-free condition. Nevertheless, it is interesting to note that the higher order scheme better approaches the divergence-free reconstruction, compared to the second order results.

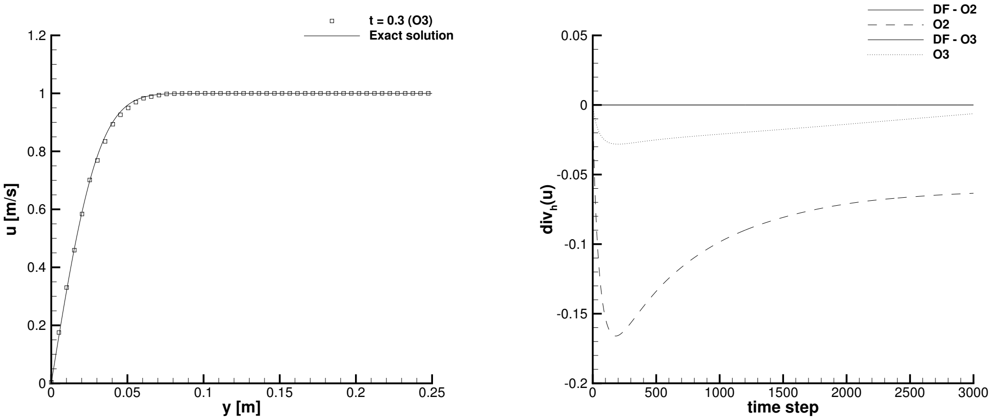

Figure 10. Left: comparison between exact solution and numerical results with $N = 2$ and $\theta = 1$ for the problem of Stokes. Right: evolution of the discrete velocity divergence for second and third order reconstructions with and without the divergence constraint.

5.4. *Stationary vortex*

We propose to solve a stationary flow motion involving a vortex, that is generated by a perfect balance between the pressure gradient and the centrifugal force. Thus, viscous terms are neglected and hydrostatic pressure is assumed within the entire simulation up to the final time $t = 0.5$. Following [27], the fluid is initially given an angular velocity

$$u_\phi = r \exp\left(-\frac{1}{2}(r^2 - 1)\right), \tag{71}$$

and a free surface elevation of

$$\eta(r) = \int \frac{u_\phi^2}{gr} dr = \eta_\infty - \frac{1}{2g} \exp\left(1 - r^2\right) \qquad \text{with} \qquad \eta_\infty := 1, \tag{72}$$

which results from the integration of the radial momentum balance equation

$$\frac{\partial \eta}{\partial r} = \frac{u_\phi^2}{gr}, \tag{73}$$

with $r = \sqrt{x^2 + y^2}$ denoting the generic radial position. Equations (71) and (72) describe a stationary flow, therefore the exact solution coincides with the initial condition. When solving this test problem numerically, the scheme produces a damping of the solution in time, due to numerical diffusion errors. The higher accurate is the numerical method, the less diffusion is generated. For our purposes, a circular domain with radius $r = 10$ is discretized on the horizontal plane with a total number of $N_P = 2340$ polygonal elements and $N_z = 10$ vertical layers bounded in the interval $z \in [0; 1]$. Figure 11 contains a cut along $y = 0$ of the free surface elevation and the numerical results have been produced using from second up to fourth order divergence-free velocity reconstructions. Indeed, the choice of the reconstruction degree affects the accuracy of the results, as evident from Figure 11. The fourth order scheme generates less dissipation compared to the others, hence achieving a better stationary solution. Figure 12 depicts a three-dimensional view of the free surface elevation at the final time with $N = 2$ as well as the two-dimensional Voronoi mesh used to carry out all simulations.

Finally, we show in Figure 13 a comparison between the results obtained with and without divergence-free reconstruction. Looking at the left panel, for $N = 3$ the free surface elevation of

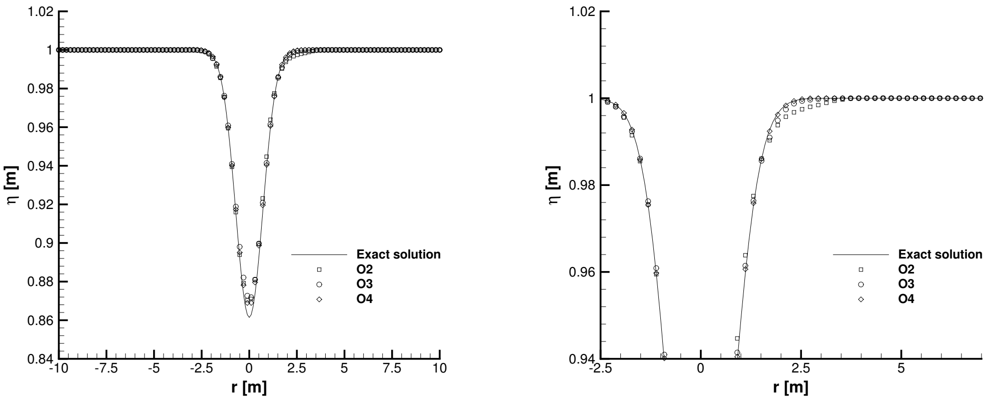

Figure 11. Stationary vortex test case: free surface elevation against exact solution (left) for second, third and fourth order version of our numerical scheme. A zoom on the vortex curvature is also plot (right).

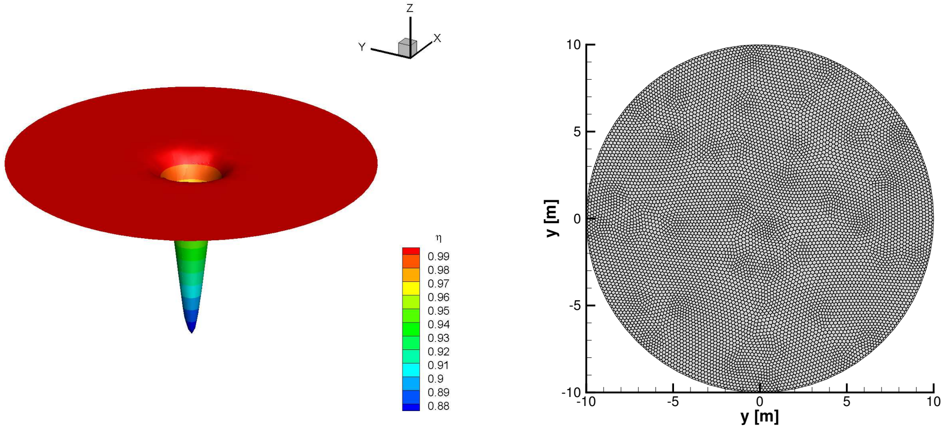

Figure 12. Free surface elevation (left) and Voronoi mesh (right) for the stationary vortex problem with $N = 2$.

the vortex is better recovered by the divergence-free scheme, while on the right panel for second, third and fourth order schemes the divergence of the velocity field is zero up to machine precision. The high order reconstructions do not preserve the divergence constraint, however the most accurate scheme has the smallest error, as already demonstrated by the results shown in Figure 10.

5.5. Oscillating basin

This test problem has been proposed in [6] in order to check and demonstrate the need of the non-hydrostatic pressure correction when vertical velocity plays an important role in the fluid flow. The basin is given by the cube $\Omega = [0; 10]^3$ and is discretized on the horizontal plane with a characteristic mesh size of $h = 0.25$, while along the $z-$direction a total constant number of $N_z = 20$ layers has been used. The flow is initially at rest and it is driven by an initial pressure gradient given by the following free surface elevation:

$$\eta(x) = 0.02x - 0.1. \tag{74}$$

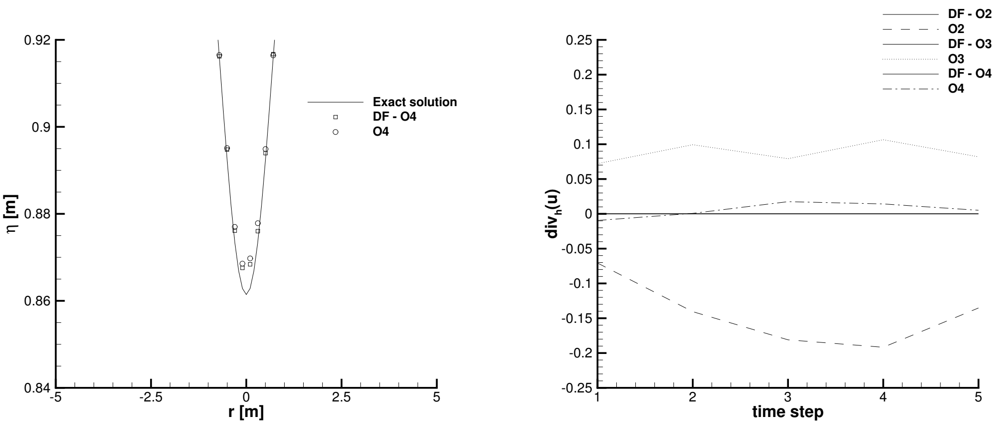

Figure 13. Left: Comparison between fourth order results for free surface elevation with and without divergence-free reconstruction. Right: evolution of the discrete velocity divergence for second, third and fourth order reconstructions with and without the divergence constraint for the stationary vortex.

According to [6], horizontal and vertical viscosity are neglected and the final time of the simulation is $t = 10$, that is reached with a time step of $\Delta t = 10^{-3}$. The exact solution is a standing wave of length $\lambda = 2L$ and frequency $f = \frac{c}{\lambda}$, with $L = 10$ denoting the length of the basin and $c = \sqrt{\frac{g\lambda}{2\pi}\tanh\left(\frac{2\pi H}{\lambda}\right)}$ being the dispersion relation.

The time history of the free surface elevation at $(x, y) = (10, 5)$ is shown in Figure 14, where we compare the third order numerical results computed with and without the non-hydrostatic correction for the pressure. As expected, the hydrostatic assumption is no longer valid for achieving reasonably good results and the analytical solution is much better recovered by the non-hydrostatic scheme, in terms of both amplitude and frequency.

6. CONCLUSIONS

In this paper we have presented an efficient semi-implicit numerical scheme for the solution of the three-dimensional Reynolds-Averaged-Navier-Stokes (RANS) equations. A staggered polygonal Voronoi mesh is employed for the discretization of the horizontal domain together with $z-$layers along the vertical direction. The pressure, the horizontal velocity and the vertical viscosity terms are discretized implicitly, while the convective and horizontal viscous terms are taken explicitly, namely relying on a semi-Lagrangian scheme that requires the integration of the Lagrangian trajectories of the flow backward in time. To this end, a new globally and pointwise divergence-free reconstruction has been designed, which generates a continuous high order velocity field representation within each control volume. We have proved that the continuity equation is satisfied up to machine precision, hence the divergence-free constraint is fully respected. The integration of the ODE that governs the Lagrangian trajectories is subsequently performed using the high order information obtained by the reconstruction. The proposed method applies to both hydrostatic as well as non-hydrostatic flows and a set of test problems has been shown in order to assess and validate the new numerical method. Whenever possible, the numerical results obtained with different order of accuracy have been compared against the exact solution.

Future work will consider the use of a turbulence model of the type $k - \omega$ as well as the inclusion of suspended sediment transport phenomena, in order to simulate more realistic configurations typically occurring close to hydroelectric power plants.

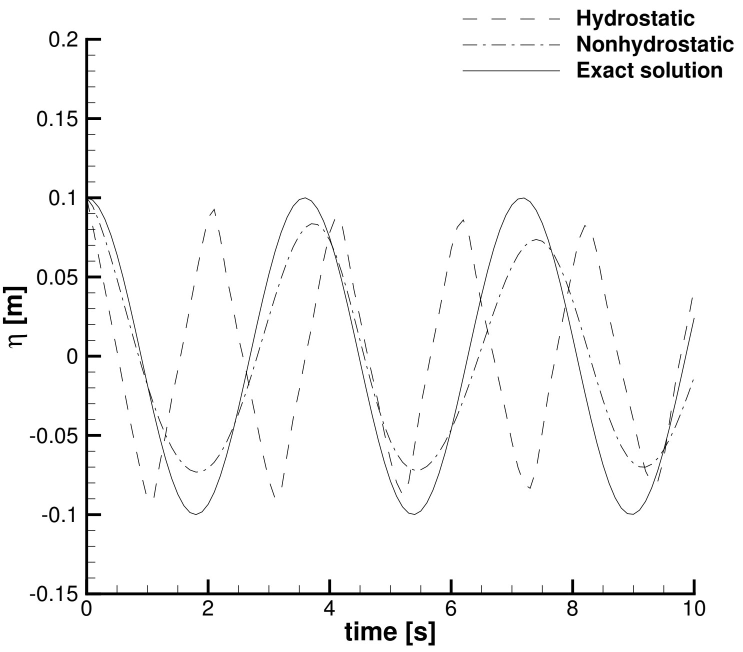

Figure 14. Free surface elevation for the oscillating basin test case. Analytical solution (solid line) is compared against the hydrostatic (dashed line) and non-hydrostatic (dashed-dotted line) third order simulation.

ACKNOWLEDGMENTS

This research has been carried out under the project *Sediplan-r* (n. FESR-1002) funded by the European Regional Development Fund (EFRE-FESR 2014-2020) of the province of Bolzano (Italy). The Authors also acknowledge the project "Thermo Fluid Dynamics, infrastructures for applied research to business and industry in South Tyrol (LTFD)", n. ERDF-1773.

The Authors would like to acknowledge the two anonymous referees for their comments and suggestions that helped to improve the quality and the clarity of this article.

Conflict of interest

The Authors declare no potential conflict of interests.

REFERENCES

1. Casulli V.. Semi-implicit finite difference methods for the two–dimensional shallow water equations. *Journal of Computational Physics.* 1990;86:56–74.
2. Casulli V., Cheng R.T.. Semi-implicit finite difference methods for three–dimensional shallow water flow. *International Journal for Numerical Methods in Fluids.* 1992;15:629–648.
3. Walters R.A., Casulli V.. A robust finite element model for hydrostatic surface water flows. *Communications in Numerical Methods in Engineering.* 1998;14:931–940.
4. Casulli V., Walters R.A.. An unstructured grid, three–dimensional model based on the shallow water equations. *International Journal for Numerical Methods in Fluids.* 2000;32:331–348.
5. Casulli V.. A semi-implicit finite difference method for non-hydrostatic free-surface flows. *International Journal for Numerical Methods in Fluids.* 1999;30:425–440.
6. Casulli V., Zanolli P.. Semi-Implicit Numerical Modeling of Nonhydrostatic Free-Surface Flows for Environmental Problems. *Mathematical and Computer Modelling.* 2002;36:56–74.

7. Courant R., Isaacson E., Rees M.. On the solution of nonlinear hyperbolic differential equations by finite differences. *Comm. Pure Appl. Math.*. 1952;5:243–255.
8. Lentine M., Grétarsson Jón Tómas, Fedkiw R.. An unconditionally stable fully conservative semi-Lagrangian method. *Journal of Computational Physics*. 2011;230:2857–2879.
9. Qiu Jing-Mei, Shu Chi-Wang. Conservative high order semi-Lagrangian finite difference WENO methods for advection in incompressible flow. *Journal of Computational Physics*. 2011;230:863-889.
10. Iske A., Käser M.. Conservative Semi-Lagrangian Advection on Adaptive Unstructured Meshes. *Numerical Methods for Partial Differential Equations*. 2004;20:388–411.
11. Stelling G.S., Duinmeijer S.P.A.. A staggered conservative scheme for every Froude number in rapidly varied shallow water flows. *International Journal for Numerical Methods in Fluids*. 2003;43:1329–1354.
12. Casulli V.. A high-resolution wetting and drying algorithm for free-surface hydrodynamics. *International Journal for Numerical Methods in Fluids*. 2009;60:391–408.
13. Casulli V., Stelling G.S.. Semi-implicit subgrid modelling of three-dimensional free-surface flows. *International Journal for Numerical Methods in Fluids*. 2011;67:441–449.
14. Tavelli M., Dumbser M.. A staggered semi-implicit discontinuous Galerkin method for the two dimensional incompressible Navier–Stokes equations. *Applied Mathematics and Computation*. 2014;248:70–92.
15. Tavelli M., Dumbser M.. A staggered space–time discontinuous Galerkin method for the three-dimensional incompressible Navier–Stokes equations on unstructured tetrahedral meshes. *Journal of Computational Physics*. 2016;319:294–323.
16. Fambri F., Dumbser M.. Semi-implicit discontinuous Galerkin methods for the incompressible Navier–Stokes equations on adaptive staggered Cartesian grids. *Computer Methods in Applied Mechanics and Engineering*. 2017;324:170–203.
17. Fambri F., Dumbser M., Zanotti O.. Space–time adaptive ADER-DG schemes for dissipative flows: Compressible Navier–Stokes and resistive MHD equations. *Computer Physics Communications*. 2017;220:297–318.
18. Dolejsi V.. Semi-implicit interior penalty discontinuous Galerkin methods for viscous compressible flows. *Communications in Computational Physics*. 2008;4:231–274.
19. Dolejsi V., Feistauer M.. A semi-implicit discontinuous Galerkin finite element method for the numerical solution of inviscid compressible flow. *Journal of Computational Physics*. 2004;198:727–746.
20. Dolejsi V., Feistauer M., Hozman J.. Analysis of semi-implicit DGFEM for nonlinear convection-diffusion problems on nonconforming meshes. *Computer Methods in Applied Mechanics and Engineering*. 2007;196:2813–2827.
21. Dumbser M., Iben U., Ioriatti M.. An efficient semi-implicit finite volume method for axially symmetric compressible flows in compliant tubes. *Applied Numerical Mathematics*. 2015;89:24–44.
22. Ioriatti M., Dumbser M.. Semi-implicit staggered discontinuous Galerkin schemes for axially symmetric viscous compressible flows in elastic tubes. *Computers and Fluids*. 2018;167:166–179.
23. Nigro A., Renda S., Bartolo C. De, Hartmann R., Bassi F.. A high-order accurate discontinuous Galerkin finite element method for laminar low Mach number flows. *International Journal for Numerical Methods in Fluids*. 2013;72:43–68.
24. Crivellini A., Bassi F.. An implicit matrix-free Discontinuous Galerkin solver for viscous and turbulent aerodynamic simulations. *Computers and Fluids*. 2011;50:81–93.
25. Bassi F., Franchina N., Ghidoni A., Rebay S.. Spectral p–multigrid discontinuous Galerkin solution of the Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*. 2011;67:1540–1558.
26. Bassi F., Crivellini A., Pietro D.A. Di, Rebay S.. An implicit high-order discontinuous Galerkin method for steady and unsteady incompressible flows. *Computers and Fluids*. 2007;36:1529–1546.
27. Boscheri W., Dumbser M., Righetti M.. A semi-implicit scheme for 3D free surface flows with high-order velocity reconstruction on unstructured Voronoi meshes. *International Journal for Numerical Methods in Fluids*. 2013;72:607–631.
28. Käser M., Iske A.. ADER schemes on adaptive triangular meshes for scalar conservation laws. *Journal of Computational Physics*. 2005;205:486–508.
29. Dumbser M., Käser M.. Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. *Journal of Computational Physics*. 2007;221:693–723.
30. Ketefian G.S., Gross E.S., Stelling G.S.. Accurate and consistent particle tracking on unstructured grids. *International Journal for Numerical Methods in Fluids*. 2015;80:648–665.
31. Karypis G., Kumar V.. Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed Computing*. 1998;48:96–129.
32. Casulli V., Cattani E.. Stability, accuracy and efficiency of a semi-implicit method for three-dimensional shallow water flow. *Computers and Mathematics with Applications*. 1994;27:99–112.
33. Abgrall R.. On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation. *Journal of Computational Physics*. 1994;144:45–58.
34. Boscheri W.. High Order Direct Arbitrary-Lagrangian-Eulerian (ALE) Finite Volume Schemes for Hyperbolic Systems on Unstructured Meshes. *Archives of Computational Methods in Engineering*. 2016;:1–51.
35. Stroud A.H.. *Approximate Calculation of Multiple Integrals*. Englewood Cliffs, New Jersey: Prentice-Hall Inc.; 1971.
36. Balsara D.S., Dumbser M.. Divergence-free MHD on unstructured meshes using high order finite volume schemes based on multidimensional Riemann Solvers. *Journal of Computational Physics*. 2015;299:687–715.
37. Gassner G., Lörcher F., Munz C.D.. A contribution to the construction of diffusion fluxes for finite volume and discontinuous Galerkin schemes. *Journal of Computational Physics*. 2007;224:1049–1063.
38. Schlichting H., Gersten K.. *Grenzschicht-Theorie*. Springer; 2005.