



# Fluid-evolving landform interaction by a surface-tracking method

Jorge Molina<sup>1</sup> · Pablo Ortiz<sup>1</sup> · Rafael Bravo<sup>1</sup>

Received: 4 December 2023 / Accepted: 13 February 2024  
© The Author(s) 2024

## Abstract

This paper introduces a continuous finite element model to simulate fluid flow–bedform interaction problems. The approach utilizes a non-oscillatory finite element algorithm to compute the fluid dynamics by solving the complete Navier–Stokes equations. Additionally, it addresses the evolution of the fluid–bedform interface as a consequence of spatially non-balanced sediment fluxes through the solution of a conservation equation for the erodible layer thickness. A sign preservation algorithm is particularly relevant for landform tracking because a positive definite thickness of the erodible sediment layer is essential to model the interaction between evolving cohesionless sediment layers and rigid beds. The fluid/terrain interface is explicitly captured through a surface tracking methodology. First, new nodes fitting the interface are incorporated into the finite element mesh; then, elements beneath this interface are deactivated, while intersected elements are restructured to get a mesh composed exclusively of tetrahedral elements. Numerical experiments demonstrate capabilities of the method by exploring relevant problems related with civil engineering, such as the evolution of trenches and the scour of a submerged pile.

**Keywords** Sediment transport · Coupled model · Adaptive grid · Positive definite algorithm

## 1 Introduction

Prediction of fluid–terrain interaction involving the transport of cohesionless sediment is fundamental to understanding the morphodynamics of natural systems as deserts, rivers or coastal regions and its effects on civil structures. For this kind of problems we consider three different interfaces: fluid/erodible sediment layers, erodible/non-erodible layers and fluid/non-erodible beds. Interfaces limiting with non-erodible beds could appear when the fluid or the moving sediment particles are in contact with cohesive strata or with structures and obstacles. Some examples of predictions including these interfaces are about the evolution of riverbeds around bridge foundations, barchan and transverse dunes in deserts and coastal areas, subaqueous dunes for the maintenance of channels, and silting of reservoirs.

Apart from its role in the sediment dynamics, evolutionary bedforms play an important role in determining the fluid flow, given that bed shape changes create boundary forcings to the flow due to momentum variations. In practice, timescale of the flow variability is frequently much shorter than that for

the bed evolution, enabling the use of two widely established strategies. First plan of action is to neglect the two-phase interaction. Second plan is to adopt a quasi-steady approximation. To illustrate the first category, in Refs. [23, 26] fluid and sediment dynamics are separated and a semi-analytic solution is integrated to link both domains. A representative example of a quasi-steady procedure can be found in Ref. [27], where authors integrate the flow equations past a frozen landform to a steady state, and update the boundary shape subsequently. Computational cost of this line of action is low and the procedure captures the basic physics of problems with evolving bedforms, although results do not always fit experimental data completely. An affordable approach to relax the limitations of the quasi-steady approximation is to use a coupled model for a *severe flow scenario*, shrinking the gap between timescales. An hypothetical severe-scenario is adopted in Refs. [19, 20], amplifying the sediment flux few orders of magnitude, and at the same time improving the efficacy of computation.

An effective numerical procedure of the fluid/sediment interface must include a straightforward treatment of the erodible/non-erodible limit. In previous works some of the present authors were concentrated on the coupling of the sediment transport with free surface flows, either by depth-integrated flow solutions [16–18], or by fully incompressible

✉ Jorge Molina  
jorgemolina@ugr.es

<sup>1</sup> University of Granada, Esc. Ing. Caminos, Campus Fuentenueva, 18071 Granada, Spain

free surface flow solutions [14]. Both procedures incorporate a Non-Oscillatory Finite Element Method (NFEM) that permits the concurrent simulation of erodible and non-erodible interfaces. The method avoids nonphysical computed values by preserving bounds for the thickness of the erodible stratum in a conservative form. Sign preservation given by the NFEM is essential both for flow computation and for interface tracking procedure. In the first case, this property avoids unphysical overshoots in pressure and velocity fields. In the second case, a positive definite thickness of the erodible layer of sediment is mandatory to account for interaction between evolving cohesionless sediment layers and rigid beds. An alternate methodology was adopted in Refs. [3, 4], where an explicit particle Lagrangian procedure was used to solve the fluid and sediment flows. This methodology proves to be efficient in several practical applications, nevertheless treatment of non erodible obstacles is not apparent. In the two-dimensional NFEM approach of Ref. [14] the solid domain is deactivated, while fluid-solid interface is updated by solving a conservation equation with mass fluxes derived from the fluid flow solution.

In this work we expand the numerical model developed in Ref. [14] to solve complete three dimensional problems involving the interaction of fluids with changing erodible beds, rigid strata, and immersed structures. The simulation of problems with embedded moving interfaces has been a challenge for decades due to the necessity of handling large jumps in physical properties. Numerical methods documented in the literature can be categorised into two widely recognised groups, those based on surface capturing and those based on surface tracking [5, 7]. First class of procedures establishes the interface implicitly by means of a phase function governed by a transport equation. In these methods, it is usual to resort to mesh refinement algorithms [12] or to enrichment functions [2] to improve the solution quality at the interface. Otherwise, in second class of methods the discontinuity matches with a mesh boundary or it is defined by faces of mesh elements. These methodologies usually give better accuracy, although the mesh has to be updated as interface evolves, thus grid adaptation or remeshing is ineludible. Some fields where this technique is applied are the propagation of fractures [8, 21], two fluids flows [22] and fluid-solid interactions [15, 24], among others. In previous references, mesh adaptation typically follows three steps: refinement of the discontinuity neighbourhood, definition of new nodes fitting the interface, and movement of background nodes to improve the mesh quality.

We propose a surface tracking method. However, the remeshing algorithm is simpler than usual, because it does not need an improved resolution at the discontinuity. Reasons for this simplification are the following: (a) solid phase is essentially at rest in typical problems with coupled fluid and erodible beds, while interface dynamics is governed by a

saltation process, thus the solid phase is not considered in the computation (and jumps) of variables through the interface (e.g., pressure gradient); (b) velocity field does not have to be reproduced in detail at the vicinity of the bedform because we assume a logarithmic wall law to compute the friction velocity; (c) in the sediment transport problems treated here, the interface is frequently smooth with slope changes in a larger scale than usual spatial discretization sizes, then intersections between interface and elements can be considered as planar. Also, the natural slope limiter (avalanche effect) is usually active for cohesionless materials, helping to regularize the interface. Therefore, present grid adaptation procedure adds new nodes fitting the interface and deactivates those elements belonging to the solid phase. Refinement step and movement of background nodes are omitted with the consequent reduction of computational cost.

Section 2 reports the governing equations for fluid and sediment (2.1), and the numerical model formulation (2.2) including details of the mesh adaptation algorithm. Moreover, particulars of the method regarding flux limiting procedure are provided in Appendix. Numerical experiments are described in Sect. 3. The first series of tests (3.1) explores the evolution of trenches due to both water flow and wind, and involves simplified two-dimensional and full three-dimensional simulations. Second test (Sect. 3.2) concerns the scour of submerged piles due to a water flow. Finally, conclusions in Sect. 4 close the paper.

## 2 Governing equations and numerical solution

### 2.1 Governing equations

Simulation of sediment transport involves the solution of two sets of equations: those governing the fluid flow, and those governing the sediment transport and the fluid/sediment interface evolution. The continuous model for fluid flow solution is the set of Navier–Stokes equations for a Newtonian incompressible fluid with density  $\rho_f$  and dynamic viscosity  $\mu_f$ . The system is written as

$$\nabla \cdot \mathbf{u} = 0 \quad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) = -\frac{1}{\rho_f} \nabla p + \frac{1}{\rho_f} \nabla \cdot \boldsymbol{\tau} \quad (2)$$

in  $\Omega \in \mathbf{R}^D$ ,  $t \in [t_0, T]$ . Here,  $\mathbf{u}(\mathbf{x}, t)$  is the velocity field ( $\mathbf{x} = (x_l)$ ,  $l = 1, D$ ),  $D$  is the number of space dimensions,  $p = p' - \rho_f g z$  is the dynamic pressure where  $p'$  is the total pressure,  $g$  is the gravity acceleration,  $z$  is the vertical coordinate,  $\boldsymbol{\tau} = \mu_f \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$  is the viscous stress tensor and  $[t_0, T]$  is the time interval. Initial conditions are given by

$\mathbf{u}(\mathbf{x}, t_0) = \bar{\mathbf{u}}_0(\mathbf{x}), p(\mathbf{x}, t_0) = \bar{p}_0(\mathbf{x})$ , and boundary conditions are

$$\mathbf{u} = \bar{\mathbf{u}}_f, \frac{\partial p}{\partial \mathbf{n}_b} = 0 \text{ on } \Gamma_f^- \quad (\text{a})$$

$$\mathbf{u} \cdot \mathbf{n}_b = 0, \mathbf{t}_b \cdot \boldsymbol{\tau} \cdot \mathbf{n}_b = 0, \mathbf{b}_b \cdot \boldsymbol{\tau} \cdot \mathbf{n}_b = 0,$$

$$\frac{\partial p}{\partial \mathbf{n}_b} = 0 \text{ on } \Gamma_s^- \quad (\text{b})$$

$$\frac{\partial \mathbf{u}}{\partial \mathbf{n}_b} = 0, p = \bar{p} \text{ on } \Gamma^+ \quad (\text{c}).$$

The boundary of the domain  $\Omega$  is designated as  $\Gamma$ ,  $\Gamma = \Gamma^- + \Gamma^+$ , and  $\bar{\mathbf{u}}_0, \bar{p}$  and  $\bar{\mathbf{u}}_f$  are known functions (from now on overline designates known values). Further,  $\mathbf{n}_b, \mathbf{t}_b, \mathbf{b}_b$  are the normal, tangent and bitangent to the boundary, respectively. Boundary  $\Gamma^- = \Gamma_f^- \cup \Gamma_s^-$  comprises those parts of the border with inlet or wall conditions ( $\mathbf{u} \cdot \mathbf{n}_b \leq 0$ ), where subscripts f and s denote fixed velocity condition and slip condition, respectively, and  $\Gamma^+$  represents the part of the boundary with outlet conditions.

Second set of equations consists of a law for the sediment flux  $\mathbf{q}_s$ , and a conservation equation of the effective sediment layer thickness  $h(\tilde{\mathbf{x}}, t)$  establishing the solid–fluid interface as

$$\rho_s \frac{\partial h}{\partial t} + \nabla_H \cdot \mathbf{q}_s = \nabla_H \cdot (\rho_s \mathcal{K} \nabla_H h^*) \quad (3)$$

in  $(\Gamma_h \subset \Gamma_f^-) \in \mathbf{R}^{D-1}, t \in [t_0, T]$  with initial condition  $h(\tilde{\mathbf{x}}, t_0) = \bar{h}_0(\tilde{\mathbf{x}})$  ( $\tilde{\mathbf{x}} = (\tilde{x}_l), l = 1, D - 1$ ), and boundary condition  $h = \bar{h}$  on  $\partial \Gamma_h$ . In Eq. (3),  $h^*(\tilde{\mathbf{x}}, t) = h_s(\tilde{\mathbf{x}}) + h(\tilde{\mathbf{x}}, t)$  denotes the vertical position of the bottom depth, where  $h_s$  is the upper position of a non-erodible stratum. Moreover,  $\rho_s = \rho_m (1 - \lambda)$  is the bulk density,  $\rho_m$  is the density of the grain material, and  $\lambda$  is the porosity (volume fraction of voids). The sediment mass flux is considered as vertically integrated over the sediment layer, and  $\nabla_H = \left( \frac{\partial}{\partial \tilde{x}_1}, \dots, \frac{\partial}{\partial \tilde{x}_{D-1}} \right)$ . In the right-hand side of the equation, the avalanche flux  $\rho_s \mathcal{K} \nabla_H h^*$  operates as a natural slope limiter. This term is computed as an anisotropic diffusion, where the diffusion coefficient  $\mathcal{K} \equiv \beta \max(0, \text{sgn}(\|\nabla_H h^*\| - s_C))$ , where  $s_C = \tan \alpha$  is the critical slope and  $\alpha$  the angle of friction; hence,  $\mathcal{K}$  depends critically on the bottom slope. The diffusivity  $\beta$  is specified in terms of temporal and spatial resolution of the numerical model [19].

In this work, we consider aeolian transport problems we consider sediment transport problems where sand is the prevailing fraction of the grain mixture. Thus, the sediment flux  $\mathbf{q}_s$  is essentially ascribed to the saltation process, involving the direct momentum transfer from fluid to the grains, in addition to the ejection due to collisions between grains. Efficacy of saltation flux models depends significantly on a detailed calculation of the wall shear stress field. Among existing

models to compute bedload sediment fluxes (see a comprehensive review in Ref. [14]), the general adjusted Bagnold’s law [9] is widely used for aeolian transport and is selected in this work,

$$\mathbf{q}_s = C \frac{\rho_f}{g} \mathbf{u}_* \|\mathbf{u}_*\|^2 \max \left( 0, 1 - \frac{u_t}{\|\mathbf{u}_*\|} \right). \quad (4)$$

In the flux law  $C = 4.2\sqrt{d/d_1}$  is an empirical coefficient [9],  $d$  is the grain size,  $d_1 = 2.5 \times 10^{-4}$  m,  $g$  is the gravity acceleration,  $\mathbf{u}_* = u_* \mathbf{u} \|\mathbf{u}\|^{-1}$  is the friction velocity and  $u_* = \sqrt{\rho_f^{-1} \tau_w}$  ( $\tau_w$  denotes the wall shear stress). Lastly,  $u_t$  is the threshold value of the friction velocity, which considers the angle  $\gamma$  between the local flow current direction and slope (see derivation in [19]),

$$u_t = \sqrt{\frac{\sin \theta}{s_C} \cos \gamma + \sqrt{\frac{\sin^2 \theta}{s_C^2} (\cos^2 \gamma - 1) + \cos^2 \theta}} u_{t0}, \quad (5)$$

where  $\theta$  is the local slope angle, and  $u_{t0}$  the threshold friction velocity for a flat bed.

### 2.2 Numerical model

Numerical solutions of Eqs. (1)–(2) and Eq. (3) are performed using the non-oscillatory finite element method. The NFEM strategy employs flux correction techniques to correct a conservative and sign-preserving predictor algorithm with anti-diffusive contributions, maintaining the properties of the predictor scheme. Corrections are calculated by limiting the difference between the contributions of a high order (HO) method and those of a predictor or low order (LO) method. Therefore, complete solutions of fluid dynamics and conservation of the sediment layer demand two numerical approximations to perform the flux correction limiting process. Algorithms defining HO and LO approaches could be related. For instance, in Ref. [10], computation of LO solution is achieved by adding extra diffusion to the HO solution. The NFEM makes use of independent high order and low order solutions. In the case of the high order algorithm, solution of Eqs.(1)–(2) is the Characteristic Based Scheme (CBS), incorporating a second-order accuracy extension for transient advective fields [16], and solution of Eq. (3) is the second-order accuracy Characteristic-Galerkin algorithm [16]. On the other hand, the low order method for the NFEM is such that it has a nearly minimum added diffusion to preserve positivity. The best candidate is a first-order upwind finite element procedure.

For completeness, we include an Appendix where the reader can find the basis of the flux correction procedure proposed, as well as a simple one-dimensional illustration of

the algorithm. In the next subsection, we introduce the finite element high order and low order solutions.

### 2.2.1 High and low order solutions

First, we specify the characteristic-based-split (CBS) algorithm as the HO procedure for the fluid dynamics (Eqs. (1)–(2)). After time discretization, the velocity solution can be written as

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta \mathbf{u}^* + \Delta \mathbf{u}^{**} \tag{6}$$

where the velocity at time level  $(n + 1)\Delta t$  is computed as the sum of a predictor velocity  $\mathbf{u}^* = \mathbf{u}^n + \Delta \mathbf{u}^*$  and a pressure correction velocity increment  $\Delta \mathbf{u}^{**}$ . Finite element discretization for  $\Delta \mathbf{u}^*$  is determined by the Characteristic Galerkin method (see e.g. [16]), and its formulation (reproduced concisely below) can be consulted in detail in Refs. [11, 12, 14]. Finite element discretizations are equipped with the finite element spaces  $\mathcal{V}_i^h, \mathcal{U}_i^h, \mathcal{W}^h$ , and  $\mathcal{P}^h$ , ( $i = 1, D$ ) stated as

$$\begin{aligned} \mathcal{V}_i^h \subset \mathcal{V}_i &= \{v_i \in H^1(\Omega) \mid v_i = 0 \text{ on } \Gamma_i^-\}, \\ \mathcal{U}_i^h \subset \mathcal{U}_i &= \{u_i \in H^1(\Omega) \mid u_i = \bar{u}_i \text{ on } \Gamma_i^-\}, \\ \mathcal{W}^h \subset \mathcal{W} &= \{w \in H^1(\Omega) \mid w = 0 \text{ on } \Gamma^+\}, \\ \mathcal{P}^h \subset \mathcal{P} &= \{p \in H^1(\Omega) \mid p = \bar{p} \text{ on } \Gamma^+\}. \end{aligned}$$

Thus, the predictor solution is determined as: Find  $\Delta u_i^{*h} \in \mathcal{U}_i^h$  for all  $t \in [t_0, T]$  such that

$$\begin{aligned} \left( v_i^h, \frac{\Delta u_i^{*h}}{\Delta t} \right)_\Omega &= - \left( v_i^h, \nabla \cdot \left( (\mathbf{u}^h)^{n+1/2} (u_i^h)^n \right) \right)_\Omega \\ &+ \left( v_i^h, f_i^h \right)_\Omega \\ &- \frac{\Delta t}{2} \left\{ \left\langle \nabla \cdot \left( v_i^h (\mathbf{u}^h \mathbf{u}^h)^{n+1/2} \right), \nabla (u_i^h)^n \right\rangle_{\Omega_I} \right. \\ &- \left. \left\langle v_i^h \left( (\mathbf{u}^h)^{n+1/2} \cdot \nabla (\mathbf{u}^h)^n \right), \nabla (u_i^h)^n \right\rangle_{\Omega_I} \right. \\ &\left. + \left( v_i^h, (\mathbf{u}^h)^{n+1/2} \cdot (\nabla (f_i^h)^n) \right)_{\Omega_I} \right\} \tag{7} \end{aligned}$$

where  $\mathbf{f} = \frac{\mu_f}{\rho_f} \nabla^2 \mathbf{u}$  and

$$\begin{aligned} (v, w)_\Omega &= \int_\Omega v w \, d\Omega \\ \langle \mathbf{v}, \mathbf{w} \rangle_\Omega &= \int_\Omega (\mathbf{v} \cdot \mathbf{w}) \, d\Omega \end{aligned}$$

and  $[v, \mathbf{c}]_\Gamma = \int_\Gamma v \mathbf{c} \cdot \mathbf{n}_b \, d\Gamma$

for vector fields  $\mathbf{v}, \mathbf{w}, \mathbf{c}$  and scalars  $v$  and  $w$ . Moreover,  $\Omega_I$  is the domain without elements with sides belonging to the boundary. Specifics about the computation of the velocity field at  $(n + 1/2)\Delta t$  in Eq. (7) are discussed in Ref. [14].

Second step involves pressure computation from the continuity equation. The solution is formulated as: Find  $(p^h)^{n+1} \in \mathcal{P}^h$  for all  $t \in [t_0, T]$ , such that

$$\begin{aligned} \frac{\Delta t}{\rho_f} \left\langle \nabla w^h, \nabla (\Delta p^h) \right\rangle_\Omega &= \left\langle \nabla w^h, \Delta \mathbf{u}^{*h} \right\rangle_\Omega \\ &- \frac{\Delta t}{\rho_f} \left\langle \nabla w^h, \nabla p^h \right\rangle_\Omega^n - [w^h, \Delta \mathbf{u}^h]_\Gamma \end{aligned} \tag{8}$$

where  $\Delta \mathbf{u}^h = (\mathbf{u}^h)^{n+1} - (\mathbf{u}^h)^n$ . Final pressure field is  $p^{n+1} = p^n + \Delta p$ . Last step updates velocity field through Eq. (6), where  $\Delta \mathbf{u}^{**}$  is given by

$$\begin{aligned} \left( v_i^h, \frac{\Delta u_i^{**h}}{\Delta t} \right)_\Omega &= - \frac{1}{\rho_f} \left( v_i^h, (\nabla p^h)_i^{n+1} \right)_\Omega \\ &- \frac{\Delta t}{2\rho_f} \left( \nabla \cdot \left( v_i^h (\mathbf{u}^h)^{n+1/2} \right), (\nabla p^h)_i^{n+1} \right)_{\Omega_I}. \end{aligned} \tag{9}$$

Second, the high order solution for the advective-diffusion transport law (3) is attained by the Characteristic Galerkin method and is stated in terms of the spaces  $\mathcal{S}^h$  and  $\mathcal{H}^h$ , defined as  $\mathcal{S}^h \subset \mathcal{S} = \{s \in H^1(\Gamma_h) \mid s = 0 \text{ on } \partial\Gamma_h\}$ ,  $\mathcal{H}^h \subset \mathcal{H} = \{h \in H^1(\Gamma_h) \mid h = \bar{h} \text{ on } \partial\Gamma_h\}$ , where  $\Gamma_h$  (specified after Eq. (3)) corresponds with the solid/fluid interface surface. Then, the HO solution is formulated as: Find  $h^h \in \mathcal{H}^h$  for all  $t \in [t_0, T]$ , such that

$$\begin{aligned} \left( s^h, \frac{\Delta h^h}{\Delta t} \right)_{\Gamma_h} &= \left( \mathbf{u}_s^{n+1/2} \cdot \nabla_H s^h, (h^h)^n \right)_{\Gamma_h} \\ &- \left[ s^h, \frac{\bar{\mathbf{q}}_s}{\rho_s} \right]_{\partial\Gamma_h^+}^n - \left( \nabla_H s^h, \mathcal{K}(\nabla_H h^{*h}) \right)_{\Gamma_h}^n + \\ &\left[ s^h, \mathcal{K}(\nabla_H h^{*h}) \right]_{\partial\Gamma_h}^n \\ &- \frac{\Delta t}{2} \left\{ \left\langle \nabla_H \cdot \left( s^h (\mathbf{u}_s^h \mathbf{u}_s^h)^{n+1/2} \right), \nabla_H (h^h)^n \right\rangle_{\Gamma_{hI}} \right. \\ &\left. \left\langle s^h \left( (\mathbf{u}_s^h)^{n+1/2} \cdot \nabla_H (\mathbf{u}_s^h)^n \right), \nabla_H (h^h)^n \right\rangle_{\Gamma_{hI}} \right. \\ &\left. + \left( \nabla_H \cdot s^h (\mathbf{u}_s^h)^{n+1/2}, \left( h^h \nabla_H \cdot \mathbf{u}_s^h \right)^n \right)_{\Gamma_{hI}} \right\} \end{aligned} \tag{10}$$

where  $\mathbf{u}_s \equiv \frac{\mathbf{q}_s}{h^h \rho_s}$ ,  $\partial\Gamma_h^+$  is defined as the portion of the bed-form boundary with a prescribed sediment mass flux  $\bar{\mathbf{q}}_s$ , and

$$(v, w)_{\Gamma_h} = \int_{\Gamma_h} v w \, d\Gamma_h$$

$$\langle \mathbf{v}, \mathbf{w} \rangle_{\Gamma_h} = \int_{\Gamma_h} (\mathbf{v} \cdot \mathbf{w}) \, d\Gamma_h$$

and  $[v, \mathbf{c}]_{\partial\Gamma_h} = \int_{\partial\Gamma_h} v \mathbf{c} \cdot \mathbf{n}_b \, d\partial\Gamma_h$ .

Computation of sediment mass flux requires stress recovery from the updated velocity field attained in the flow solution. In this work, shear stresses at the sediment/fluid interface are estimated by assuming a logarithmic wall velocity law; then, friction velocity is calculated as

$$(\mathbf{u}_*)^{n+1} = \kappa \frac{\mathbf{u}_\Delta^{n+1} - (\mathbf{u}_\Delta^{n+1} \cdot \mathbf{n}_b) \mathbf{n}_b}{\ln(z_\Delta/z_0)}, \tag{11}$$

where  $\mathbf{u}_\Delta = \mathbf{u}(z_\Delta)$  is the velocity value taken at height  $z_\Delta$  over the bedform,  $\kappa = 0.41$  is the Von Karman constant, and  $z_0$  is the equivalent roughness length. The height  $z_\Delta$  is estimated in terms of mesh resolution, and its value in Eq. (11) is assumed at time  $n \Delta t$ . Prior to the solution (10), computation of  $\mathbf{q}_s^{n+1}$  and  $\mathbf{u}_s^{n+1/2} = (\mathbf{u}_s^n + \mathbf{u}_s^{n+1})/2$  is achieved by elementwise usage of the law stated by Eq. (4) and an a posteriori nodal recovery.

Now, we examine the low-order solutions. The proposed low-order approach is an upwind finite element monotone scheme independent of the high-order procedure. From now on, we introduce the notation  $\mathcal{LO}(\cdot)$  comprising upwind fluxes computation such that

$$\mathcal{LO}(\psi, \mathbf{a}) = (\mathbf{a} \cdot \nabla \psi) + [\nabla \cdot (\mathbf{k} \nabla \psi)] \tag{12}$$

for a scalar field  $\psi$  and a vector field  $\mathbf{a}$ , where  $\mathbf{k}$  is a diffusivity tensor corresponding to the artificial diffusion of the finite element upwind scheme (see specifics in Eq. (13) of Ref. [14]). Results in the computation of the LO solution are essentially the same by median dual finite volume method or Galerkin linear finite elements with lumped mass approximation [16]. Here, the low-order predictor method is written via the Galerkin formulation for the sake of simplicity.

A monolithic implementation of the low-order solution for fluid flow requires three steps similar to Eqs. (7)–(9). Then, this course of action would entail the computation of two implicit solutions for pressure (Eq. (8) and the corresponding for the LO solution). Both calculations should be performed every time step followed by the correction reported in the Appendix. A cost-effective way to circumvent this double computation is by performing correction only for the predictor velocity field. Then, the abbreviated procedure has the following steps:

1. Computation of the HO velocity predictor  $\mathbf{u}^{*h} = (\mathbf{u}^h)^n + \Delta \mathbf{u}^{*h}$  given by Eq. (7).

2. Calculation of the LO velocity predictor  $\mathbf{u}_{LO}^{*h} = (\mathbf{u}^h)^n + \Delta \mathbf{u}_{LO}^{*h}$ , where the intermediate velocity increment is

$$\begin{aligned} & \left( v_{iL}^h, \frac{\Delta u_{iLO}^{*h}}{\Delta t} \right)_{\Omega} \\ &= \left( v_i^h, \mathcal{LO} \left( (u_i^h + \frac{\Delta t}{2} f_i^h)^n, (\mathbf{u}^h)^n \right) \right)_{\Omega} + \left( v_i^h, f_i^h \right)_{\Omega} \end{aligned} \tag{13}$$

where  $v_{iL}^h$  is the lumped mass matrix and  $LO$  subindex signifies low-order solution.

3. Computation of the corrected velocity predictor  $\tilde{\mathbf{u}}^{*h}$  by the correction algorithm (Appendix, see Eq. (23)).
4. Calculation of the updated pressure by solving the implicit equation (8) and replacing  $\Delta \mathbf{u}^{*h}$  by the corrected velocity increment  $\Delta \tilde{\mathbf{u}}^{*h} = \tilde{\mathbf{u}}^{*h} - (\mathbf{u}^h)^n$ .
5. Get the updated velocity field by computing  $\Delta \mathbf{u}^{**}$  with Eq. (9).

Low-order form for the bedform evolution (3) is formulated as: Find  $(h^h)_{LO}^{n+1} \in \mathcal{H}^h$  for all  $t \in [t_0, T]$ , such that

$$\begin{aligned} \left( s_L^h, \frac{\Delta h_{LO}^h}{\Delta t} \right)_{\Gamma_h} &= \left( s^h, \mathcal{LO} \left( (h^h)^n, (\mathbf{u}_s^h)^n \right) \right)_{\Gamma_h} \\ &- \left\langle \nabla_H s^h, \mathcal{K}(\nabla_H h^{*h})^n \right\rangle_{\Gamma_h} + \left[ s^h, \mathcal{K}(\nabla_H h^{*h}) \right]_{\partial\Gamma_h}^n, \end{aligned} \tag{14}$$

where  $s_L^h$  corresponds to the lumped mass matrix, and  $(h^h)_{LO}^{n+1} = (h^h)^n + \Delta h_{LO}^h$ .

All series of experiments in preceding works [11–14] have shown that the use of this reduced method instead of the monolithic procedure has no noticeable differences in results and retains the capability to preserve interface resolution. The average extra computational cost of the abridged NFEM respect to the procedure with no corrections is about 1% for serial calculation of incompressible flows.

### 2.2.2 Mesh adaptation algorithm

In this subsection we report the proposed grid adaptation method to the practical tracking of the fluid–bedform interface. Grid adaptation requires the following initial mesh data: (a) coordinates of all nodes and their connectivity; (b) all edges and the pair of nodes defining each one; and (c) edges defining each element of the mesh. These data are conveniently stored by linked lists. Once values of  $h^{n+1}$  and bed position are updated, grid adaptation is conducted by three steps:

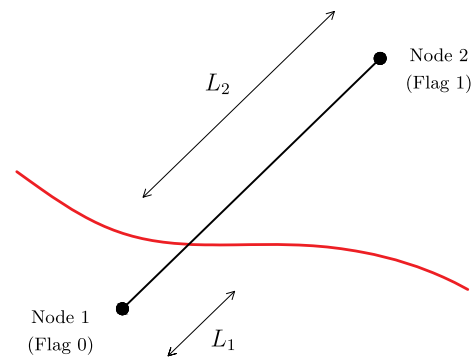
1. Loop over nodes:  $h_i^*$  is calculated at coordinates  $(x_i, y_i)$  of every node  $i \in \Omega$  (with vertical coordinate  $z_i$  of the original mesh). Nodes are labelled as
  - Under the interface:  $z_i + \zeta < h_i^*$ , Node flag is 0 (inactive).
  - Over the interface:  $z_i - \zeta > h_i^*$ . Node flag is 1 (active).
  - Coincident with the interface:  $z_i - \zeta \leq h_i^* \leq z_i + \zeta$ . Node flag is 2 (coincident).

The number  $\zeta$  is of the order of round-off value.

2. Loop over edges: pair of nodes delimiting every edge are labelled as
  - Both nodes have flag 0: edge is under the bedform; edge flag is 0 (inactive).
  - Both nodes have flag 1: edge is over the bedform; edge flag is 1 (active).
  - Both nodes have flag 2: edge is contained in the interface; edge label is 2 (coincident).
  - One node has flag 2 and the other node has flag 0 or 1; edge flag is 0 or 1 depending on the flag of the second node.
  - One node has flag 0 and the other node has flag 1; edge is cut by the bedform and its flag is 3 (intersected). In this situation, coordinates of the cut-off node are calculated and stored. Nevertheless, if the intersection is close to one of the edge nodes, element could be too small in compare with those of the original mesh. To avoid this inconvenience, we define a tolerance  $\xi$  such that, if  $\frac{L_1}{L_1 + L_2} < \xi$  (see Fig. 1), flag of node 1 is designated as 2. Practical values for the tolerance are  $\xi \in [0.05, 0.10]$ . If this special case occurs, step 2 must be run again to avoid indeterminacy of other edges containing modified nodes.

3. Loop over elements: we consider tetrahedrons and assume elements small enough such that the cut caused by the bedform is plane. Hence, the following cases can be produced,

- There are not nodes with flag 1: the tetrahedron is completely below the bedform, then it is deactivated and not included in the new mesh.
- The sum of nodes with flag 1 and nodes with flag 2 is four: tetrahedron is over (or tangent to) the bedform, then the element is active and its geometry is identical to that of the original mesh.
- One edge has flag 2 and other edge has flag 3: the element is split and a new element is formed with (a) the active node, (b) the cut-off node of the edge with label 3 and (c) the two nodes with flag 2 (see Fig. 2).
- Two edges have flag 3 and one node has flag 2: we can distinguish two cases. The first one occurs when just one node is label 1, in this case the new element



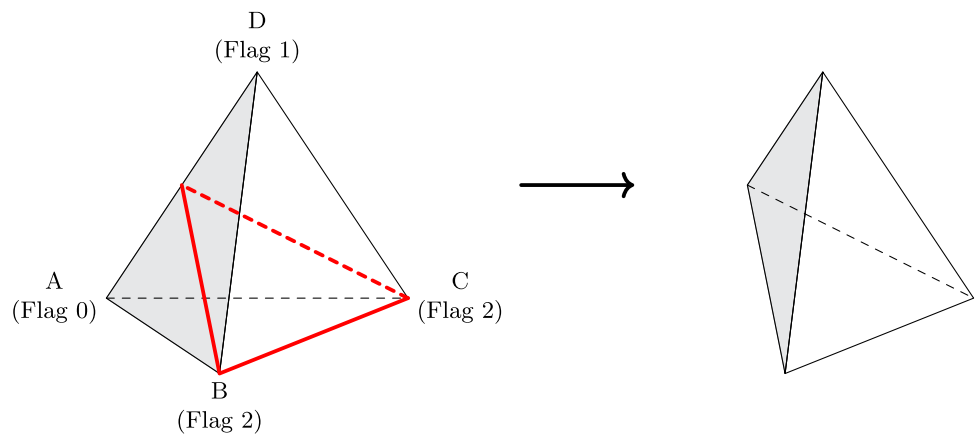
**Fig. 1** Edge (black line) cut by the bedform (red line). (Color figure online)

is formed with (a) the active node, (b) the two cut-off nodes of edges with label 3 and (c) the node with flag 2 (see Fig. 3). Second case takes place when two nodes have flag 1; the resulting square pyramid is divided into two tetrahedrons by splitting the pyramid base by the shortest diagonal (see Fig. 4). Then, the two new elements are formed by (a) active nodes, (b) the node with flag 2 and (c) the two cut-off nodes of edges with label 3.

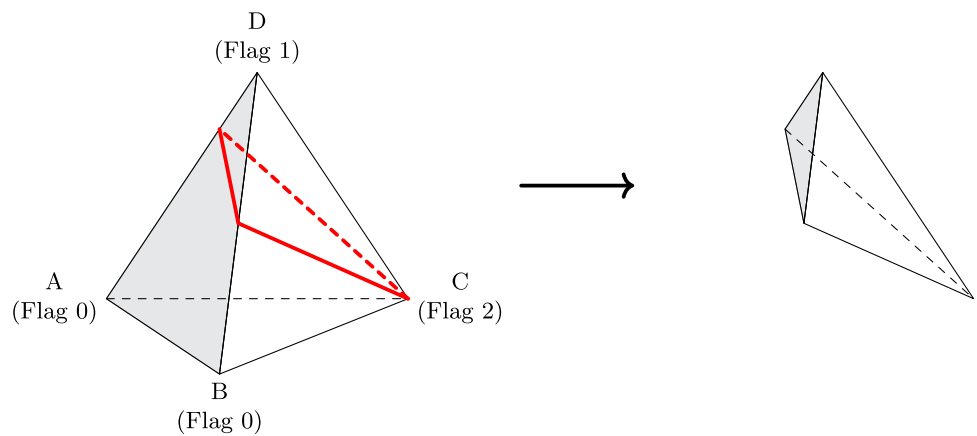
- Three edges have label 3: here two cases are identified. The first one takes place when just one node has label 1, in this case, the new element is formed by (a) the three cut-off nodes and (b) the node with flag 1 (see Fig. 5). Second case occurs when there are three nodes with flag 1; in this situation, the resulting triangular prism is split into three tetrahedrons by subdividing the square faces (see Fig. 6). Special care must be taken when faces are split to avoid an un-tetrahedralizable polyhedron.
- Four edges have label 3: here the resulting triangular prism is divided into three tetrahedrons by subdividing the square faces as in the previous case (see Fig. 7).

Once activated elements are established, the subsequent steps involve the assembling of the new mesh and the prescription of boundary conditions on those edges in contact with the bedform. Pressure values at new nodes on edges with label 3 are taken from previous nodes belonging to the same edge. Errors introduced in this process are negligible as the fluid-bedform interface movement has a time scale variability much larger than the flow and the updating timestep; hence, new nodes are very close to the former ones. Furthermore, in this instance, the grid adaptation algorithm does not need to be performed every time step. Otherwise, when a “severe flow scenario” is adopted, defined either by similar time scales of fluid flow and saltation flux or by the saltation flux amplified a few orders of magnitude [20], mesh adaptation every time step is recommended.

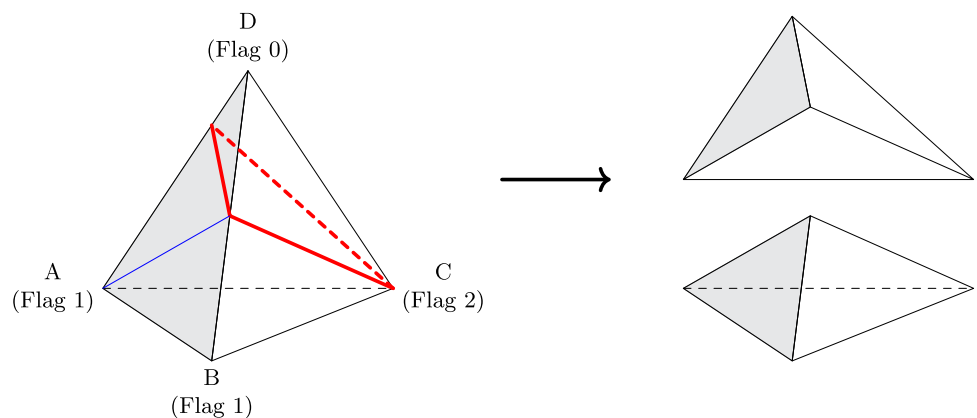
**Fig. 2** Case with one edge with label 2 and other edge with label 3. Red line indicates the cut made by the bedform. (Color figure online)



**Fig. 3** Case with two edges with label 3, one node with label 2 and one node with label 1. Red line indicates the cut made by the bedform. (Color figure online)



**Fig. 4** Case with two edges with label 3, one node with label 2 and two nodes with label 1. Red line indicates the cut made by the bedform and blue line shows face subdivision. (Color figure online)



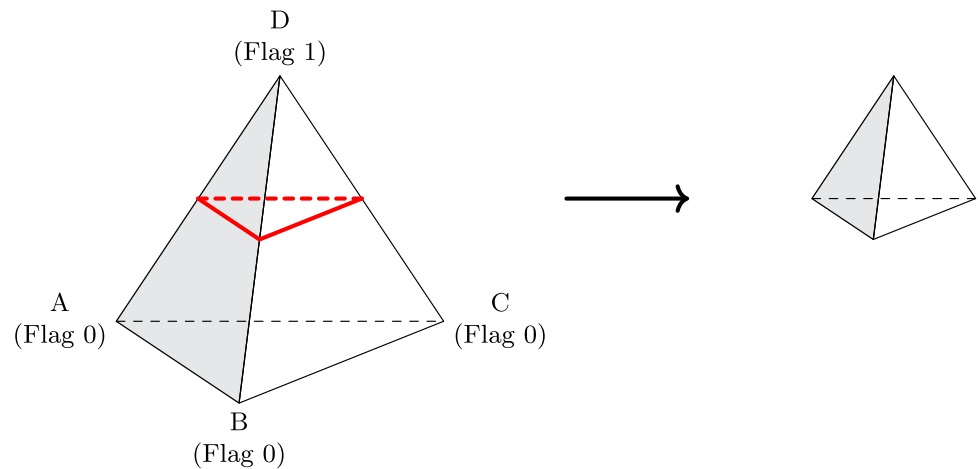
### 2.2.3 A brief on parallel coding development

The Finite Element model described above has been programmed in C++ and parallelized with MPI (Message Passing Interface). In addition, OpenMP (Open Multi-Processing) is also employed for those tasks requiring shared memory. In summary, the procedure is as follows: the root process reads grids for fluid solution and for solid–fluid interface solution; then, grids are divided in the  $X$ – $Y$  plane into  $N$  subdomains, being  $N$  the number of available computational cores. Each core receives data of all nodes and elements

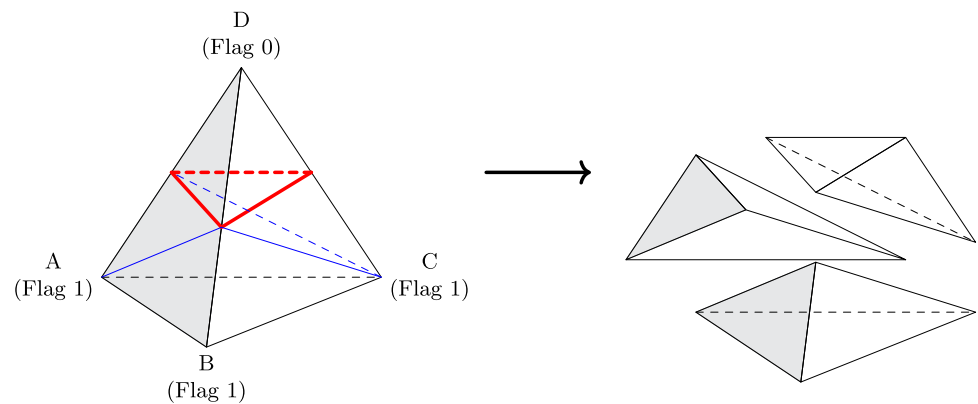
within its domain, in addition to data of ‘shared’ elements (those with at least one node belonging to the subdomain). On the basis of this information, each core is able to get local nodal vectors resulting from the computation of the right hand side of Eqs. (7)–(10), (13) and (14). Hence, solutions of aforementioned equations are calculated employing a Pre-conditioned Conjugate Gradient (PCG), which also allows for MPI parallelization as the coefficient matrix does not have to be assembled.

There are some modules of the code that use data from the whole mesh. For example, the sediment fluxes compu-

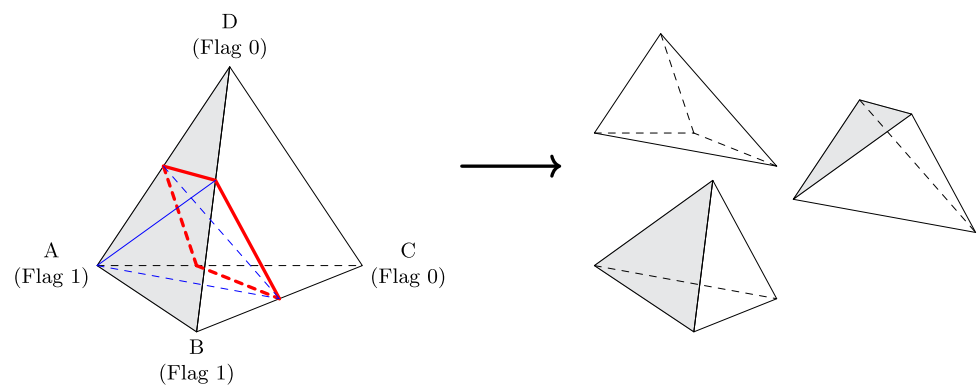
**Fig. 5** Case with three edges with label 3 and one node with flag 1. Red line indicates the cut made by the bedform. (Color figure online)



**Fig. 6** Case with three edges with label 3 and three nodes with flag 1. Red line indicates the cut made by the bedform and blue lines show faces subdivision. (Color figure online)



**Fig. 7** Case with four edges with label 3. Red line indicates the cut made by the bedform and blue lines show faces subdivision. (Color figure online)



tation (Eq. (4)) requires the recovering of the velocity field from the fluid dynamics solution and involves a search of elements surrounding each node of the fluid-solid interface (see details in Ref. [14]). Other examples are steps 2 and 3 of the mesh adaptation algorithm (Sect. 2.2.2), whose implementation is more efficient by using shared memory parallelism. As mentioned before, parallelization in these modules is performed with OpenMP. In some tests carried out on an Intel Core i9-12900K, we found that the MPI parallelization accelerated the execution up to 3 times respect to the same code but entirely parallelized with OpenMP.

## 3 Results

### 3.1 Evolution of trenches

In this section we examine the evolution of different trenches due to the effect of a water flow and of the wind. In the first simulation we assess the performance of the numerical model when a non-erodible stratum is present within the domain. For this purpose we reproduce one of the experiments reported by Struiksmma [25]. It was performed in a straight channel with 11.5 m length, 0.2 m width, and a vertical side rigid wall of 0.5 m. Initial configuration of the



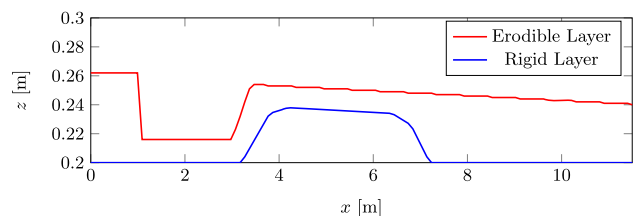


Fig. 8 Struiksmas experiment: initial set-up

test is depicted in Fig. 8. As can be seen, the flume is partially filled with a non-erodible layer forming an obstacle (blue line in Fig. 8) and, above it, there is an erodible bedform of sand with  $d = 0.6$  mm. Moreover, a trench with 0.04 m depth and 2 m length is dug in the erodible layer (red line in Fig. 8). In the experiment, a water flow of 9.2 l/s is discharged from left to right in Fig. 8, and the free surface is placed at  $z = 0.362$  m. The employed two dimensional structured mesh has an average element size<sup>1</sup>  $\delta = 0.005$  m, and its dimensions are  $17.25 \times 0.237$  m. Regarding boundary conditions, no-slip condition is imposed at the bottom, left boundary has a prescribed influx condition and right boundary reproduces a water discharge over a weir to keep a subcritical flow in the flume. In the simulation, we solve the dynamics of a two-phase flow (air and water), where the gas-liquid interface is captured using a conservative level-set method (see formulation in Ref. [14]). Air has  $\rho_1 = 1.205$  kg/m<sup>3</sup> and  $\mu_1 = 1.808 \times 10^{-5}$  kg/(ms), while water has  $\rho_2 = 998$  kg/m<sup>3</sup>,  $\mu_2 = 1.0 \times 10^{-3}$  kg/(ms). Sediment flux is computed by using the formula proposed by Struiksmas [25],

$$q_s/\rho_s = C_\epsilon \mathbf{u} \|\mathbf{u}\|^4, \tag{15}$$

where  $C_\epsilon = 0.00049$  s<sup>4</sup>/m<sup>3</sup> and  $\mathbf{u}$  is the depth-averaged water velocity. Note that the equation (15) is not universally applicable, as it was tuned in the laboratory specifically for the current experiment; hence, its use should be restricted to this particular setup. Furthermore, fluxes obtained from Eq. (15) are scaled by a factor of 60 to transform 4 min of simulated time into 4 h of real time. Simulations last 255 s and the timestep is  $\Delta t = 0.001$  s. During the first 15 s, sediment transport is deactivated to reach an initial steady-state in hydrodynamics; subsequently, the bedload transport law is activated.

Figure 9 shows experimental and numerical results by representing the bottom level at  $t = 1, 2, 3$  and 4 h. Results are shown superimposed on the initial bed. Numerics match satisfactory with laboratory results, even when the non-erodible bed becomes uncovered (Fig. 9c, d). The sharp front position and the silting process of the trench are accurately captured at  $t = 1, 2$  and 3 h, with  $L_2$ -norm errors  $e(L_2)_{1h} = 6.94 \times 10^{-3}$

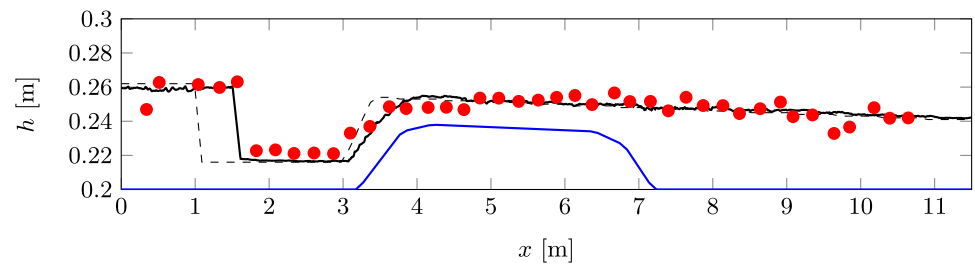
<sup>1</sup> Average element size for 2D structured meshes is  $\delta = \sqrt{\Delta x^2 + \Delta y^2}$ .

m,  $e(L_2)_{2h} = 7.41 \times 10^{-3}$  m, and  $e(L_2)_{3h} = 9.70 \times 10^{-3}$  m. However, the front at  $t = 4$  h is not evinced by the lab experiment (Fig. 9d), and the error increases to  $e(L_2)_{4h} = 1.40 \times 10^{-2}$  m. Otherwise, position of the bedform downstream the trench is very similar to laboratory outputs for  $t = 1$  h and 2 h, and somewhat overpredicted for  $t = 3$  h and 4 h. This slight overprediction can be ascribed to deviations in the sediment outflow through the right boundary respect to the laboratory set-up. Despite the fact that the sediment flux law (15) has not a threshold condition for sediment onset of motion, erosion excess is apparent only in a reduced region close to the upper boundary of the non-erodible obstacle. Similar effect can be recognized in Refs. [18, 25].

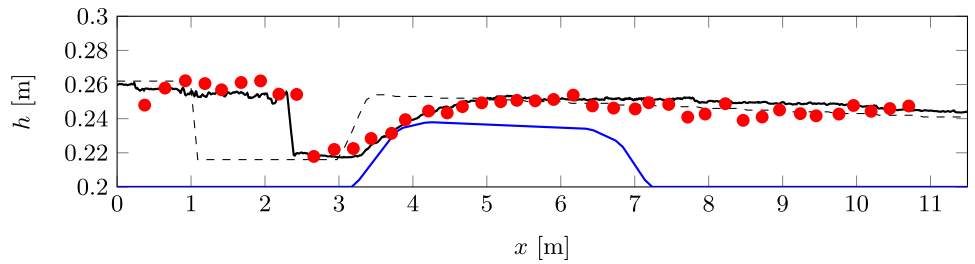
In the second series of simulations, we study the evolution of two different trenches due to the wind action. The first trench has a depth of 0.05 m, a bottom length of 0.2 m, and a side slope of 45°; and the second trench has a 0.025 m depth, a bottom length of 0.1 m, and side slope of 22.5° (see the solid line in Figs. 10 and 11). In both cases, the employed mesh is structured, with a 1.0 m length, 0.05 m width, and 0.2 m height, and has an average element size<sup>2</sup> of  $\delta = 0.0023$  m. Physical parameters of the air are  $\rho_f = 1.205$  kg/m<sup>3</sup>,  $\mu_f = 1.808 \times 10^{-5}$  kg/(ms). The air is considered at rest at the initial time. Left boundary of the mesh has inlet conditions, and air enters the domain according to a boundary layer law:  $u(z) = \frac{0.5}{\kappa} \log\left(\frac{z}{z_0}\right)$ . Regarding the sediment,  $\rho_s = 1987.5$  kg/m<sup>3</sup>,  $d = 2.5 \times 10^{-4}$  m,  $z_0 = 2.5 \times 10^{-5}$  m,  $u_{t0} = 0.22$  m/s, and  $\alpha = 45^\circ$ . Besides, to accelerate the computation of sediment flux, it is scaled by a time multiplier of 50, assuming the time scale of flow variability much shorter than that for bed evolution. We run the simulation for a total of 8 s with  $\Delta t = 5 \times 10^{-5}$  s, which is equivalent to 400 real seconds for the erodible bed. Figure 10 shows the deep trench profile at real times  $t = 0, 100, 200,$  and 400 s, and Fig. 11 shows the profile for the shallow case at  $t = 0, 25, 100, 200,$  and 400 s. Additionally, a 3D view of some time steps is shown in Figs. 12 and 13. As can be seen, in both cases the trench moves in the wind direction, i.e. there is sediment deposition on the upwind (left) side and erosion on the downwind (right) side. However, the velocity migration of the deep trench is lower than the shallow case, in line with previous numerical simulations [20] and experimental observations [1]. We also notice that erosion on the right side is more intense than deposition on the left side. This fact results in a progressive change of the trench shape leading to partial silting (see trench depth and bottom length evolution in Figs. 10 and 11). Moreover, it is manifest that the left side of the trench starts to migrate

<sup>2</sup> Average element size for three-dimensional meshes is defined as  $\delta = \sqrt[3]{\frac{\sum_{i=1}^N V_i}{N}}$ , where  $V_i$  is the volume of cell  $i$ , and  $N$  is the total number of cells.

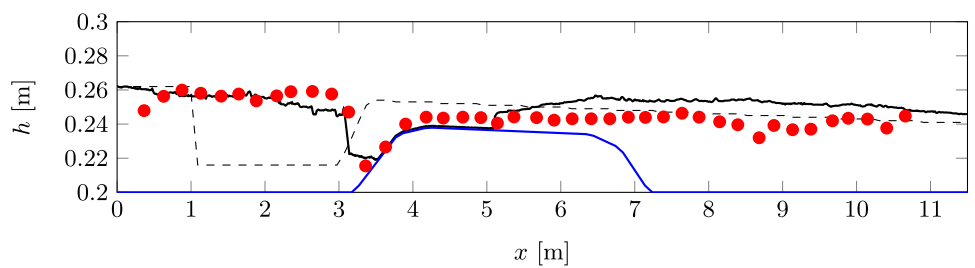
**Fig. 9** Struikma's experiment. Black dashed line and blue line are the initial erodible and non-erodible bed profiles, respectively, black solid line indicates numerical results and red dots show laboratory outputs. (Color figure online) [25]



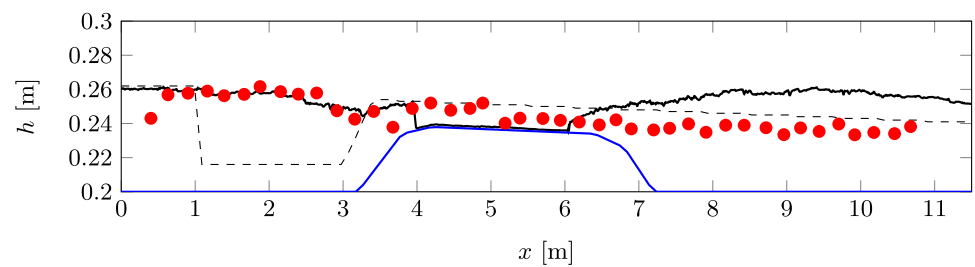
(a) 1 hour



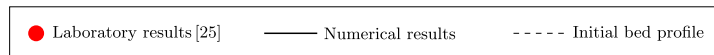
(b) 2 hours



(c) 3 hours



(d) 4 hours



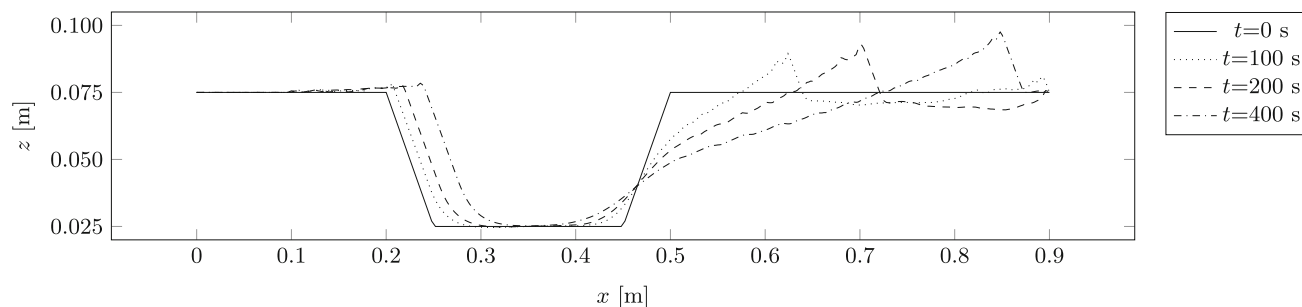


Fig. 10 Deep trench evolution

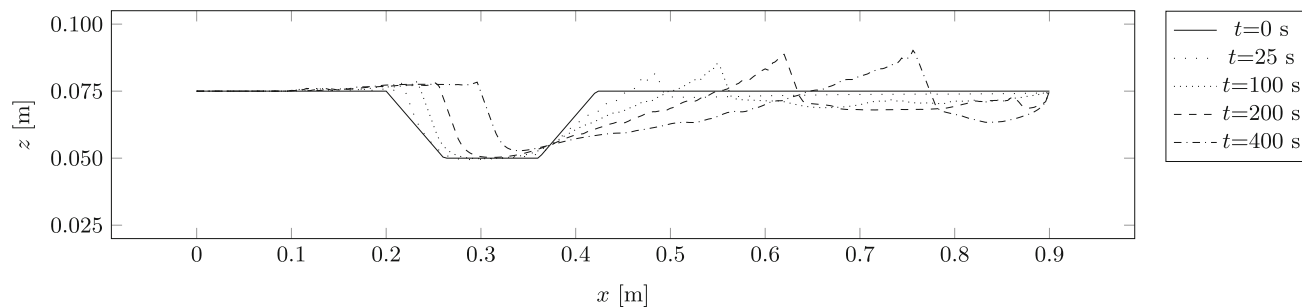


Fig. 11 Shallow trench evolution

just after reaching its critical slope. This is notorious in the shallow case (see Fig. 11), which starts with a side slope lower than the critical. Otherwise, sediment subtracted from the right side forms a secondary pile downwind advancing faster than the downwind side of the trench. This fact is consistent with the classic relation stating that dune velocity is inversely proportional to its height (see e.g. Ref. [14], Fig. 16). Sediment contribution to this sandpile is determined by downwind slope, affecting air flow and ultimately sediment transport. Then, fluid/sediment dynamics can give rise to two opposed scenarios. If there is no sediment supply upstream the trench (current case), its shape would be similar to an inverted barchan dune and, depending on the depth, it would migrate or disappear with time [20]. On the other hand, if there exists an unlimited sediment source, the trench will ultimately get silted regardless of its depth.

### 3.2 Scour in a pile

In this section, we examine the local scour around a submerged vertical pile. In particular, we reproduce laboratory test 21 of Ref. [29]. This test was conducted in a water flume with an erodible bedform composed of sand with  $d = 0.40$  mm,  $\rho_m/\rho_f = 2.69$ , and  $\alpha = 35^\circ$ . A cylindrical structure with a diameter  $D = 0.1$  m was inserted into the sediment in such a manner that the height  $h_c$  not buried by sediment was measured in the laboratory (see figure 3b of Ref. [29]), and it correlates with the logarithmic law  $u(z) = \frac{u_*}{\kappa} \ln\left(\frac{z}{z_0}\right)$ , where

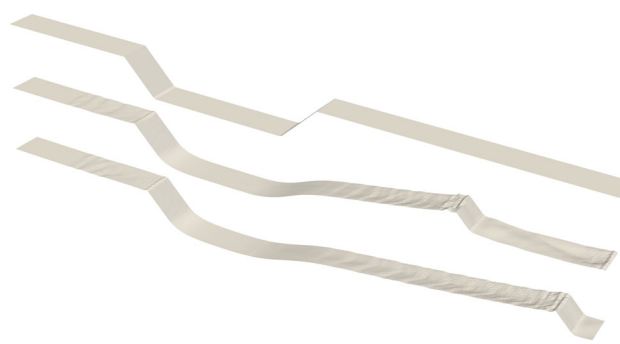


Fig. 12 3D visualization of the deep trench at times  $t = 0, 200$  and  $400$  s



Fig. 13 3D visualization of the shallow trench at times  $t = 0, 200$  and  $400$  s

$u_* = 0.0483$  m/s and  $z_0 = 0.0043$  m, resulting in a mean flow velocity of 0.441 m/s. The numerical domain employed

in this simulation is a prism of  $20D$  length,  $10D$  width, and  $7D$  height with a cylinder-shaped hole in the middle (see Fig. 14). The domain was discretised by using tetrahedrons with  $\delta = 0.076D$  (unstructured). Physical parameters for water are  $\rho_f = 1000 \text{ kg/m}^3$ ,  $\mu_f = 0.001 \text{ kg/ms}$  and, for the sediment,  $u_{t0} = 0.0148 \text{ m/s}$  according to data provided in Ref. [29]. Regarding boundary conditions, a fixed inlet velocity was imposed at the left boundary with a law coinciding with the laboratory setup; outlet condition described in Sect. 2.1 was applied at the right boundary. For the bedform and the cylinder surface, we employ non-slip condition, while a slip condition is used for side boundaries. At the free surface, we also applied slip boundary condition, resembling standard simulations involving scour around structures. The basis of this simplification lies in the independence of the scour depth of the water depth when the ratio of water depth to cylinder diameter is greater than 4 (see [29] and references therein). This assumption is not expected to have an appreciable effect on the numerics. At initial time, the bedform is assumed horizontal and placed at  $z = 0$  (see the reference system in Fig. 14). For the computation of sediment fluxes we employ the equation specifically calibrated for water flows proposed by Duran et al. [6],

$$\mathbf{q}_s = \frac{5.78\rho_s}{(\rho_m/\rho_f - 1)g} \mathbf{u}_* \max\left(0, \|\mathbf{u}_*\|^2 - u_t^2\right), \quad (16)$$

which was used in previous works (see e.g. [14]) giving good results.

The simulation was run with  $\Delta t = 0.005 \text{ s}$  until reaching the scour depth equilibrium. This state is represented in Fig. 15, which shows a pit around the pile and a longitudinal trench downstream. These features arise due to a horseshoe vortex and vortex shedding downstream the pile, that erode the sediment located close to the cylinder surface and its wake. The equilibrium takes place when the pit sides take a slope lower or equal to the critical (see Fig. 15b), and the flow velocity is very low in the pit (see Fig. 15a), resulting in a friction velocity lower than the threshold. Under these conditions, sediment transport ceases, and the scour depth remains constant. Additionally, irregular ripples are observed upstream the pile, consistent with the report of Ref. [29] for tests with a mean inflow velocity of  $0.441 \text{ m/s}$ . The resulting bedform is not symmetric with respect to the horizontal axis, particularly downstream the pile, due to the vortex shedding depicted by the streamlines in Fig. 15a. This pattern is also appreciated in laboratory experiments. Figure 15c shows the scour depth around the pile, ranging from  $0.91D$  to  $1.22D$ . While there are slight variations in scour depth upstream and downstream of the pile ( $\theta = 180^\circ$  and  $\theta = 0^\circ$ , respectively) compared to the laboratory results, the calculated mean scour depth of  $1.06D$  (see the dashed line in Fig. 15c) closely aligns with the reference value of  $1.03D$  reported in Table 1 of Ref. [29].

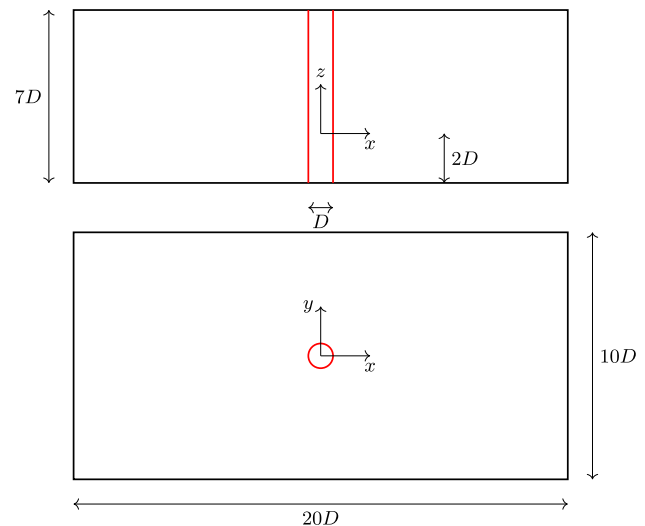
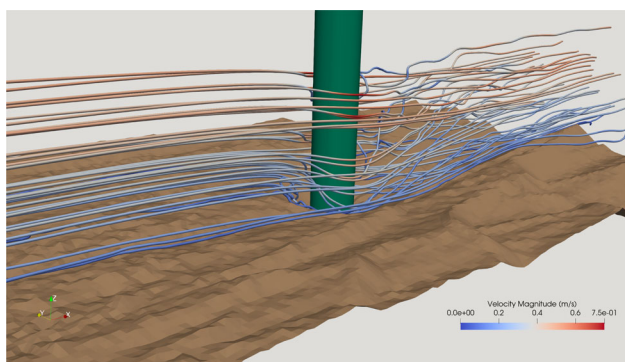


Fig. 14 Scour in a pile: sketch of the numerical domain

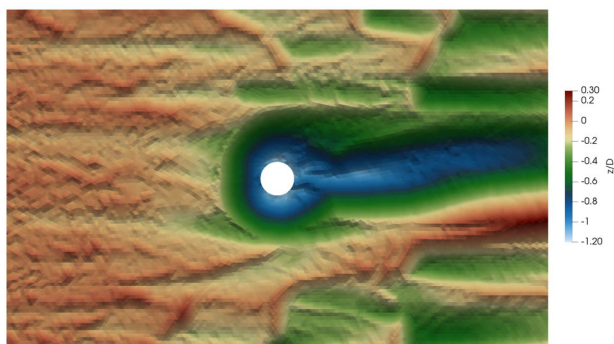
## 4 Conclusions

The numerical model proposed in this work combines a complete three-dimensional flow solution with an interface tracking algorithm to adapt the mesh to an evolving bedform, governed by a dimension-reduced conservation equation. This method admits beds formed by a combination of cohesionless and non-removable materials, making necessary the implementation of the non-oscillatory finite element method (NFEM) to assure the positivity of the erodible layer thickness. The coupling permits free surface flows simulations and takes advantage of NFEM benefits to shrink instabilities in the hydrodynamics, particularly near interfaces. In contrast with previous works where hydrodynamics was solved by using depth-integrated fluid flow equations, here we achieve a more detailed hydrodynamics solution, in special vertical dynamics and vortexes distribution, allowing computation of a more realistic wall shear stress field as well as erodible layer dynamics.

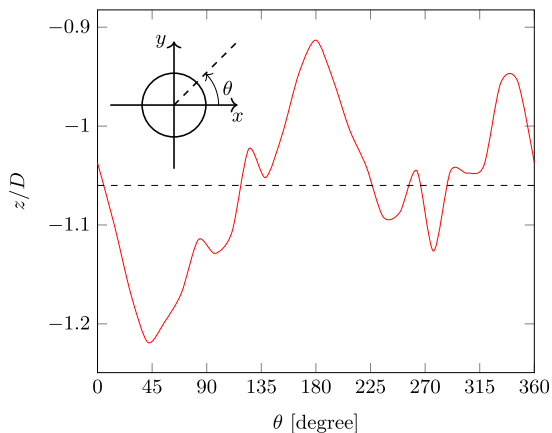
Numerical simulations explore scenarios connected with relevant engineering problems. In the first series of tests, we study the evolution of trenches under the influence of both a water flow and wind. In the two-dimensional case of a channel with a partially erodible bed, the complex dynamics resulting from the presence of a deep trench and a medium gravel obstacle lead to the exposure of the obstacle and the burial of the trench. This experiment demonstrates the numerical model capability to handle several interfaces (i.e., air/water, water/cohesionless sediment, cohesionless sediment/non-erodible bed, and water/non-erodible bed), with results consistent with laboratory experiments. In the three-dimensional case, we investigate the migration and shape evolution of two trenches in a sandy environment. We observe that migration velocity increases as trench depth



(a) Streamlines around the pile.



(b) Bed topography map.



(c) Scour depth distribution around the pile. The dashed line indicates the mean scour depth.

**Fig. 15** Scour in a pile: numerical results once the equilibrium scour depth is reached

decreases, in addition to a partial silting in both tests, particularly noticeable in the shallow case. This pattern aligns with numerical simulations from prior studies and resembles the behavior seen in barchan dunes. In fact, as illustrated in numerical results, the trench adopts a shape similar to an inverted dune, with an upstream side slope steeper than the downstream one. Finally, in the last test, we study the scour

around a submerged pile caused by a steady current. The process is governed by the combination of a horseshoe vortex and vortex shedding, which washes the sediment around the pile and in the wake of the cylinder, forming a pit and a trench. The numerical model also yields encouraging results similar to outputs observed in the laboratory, particularly in terms of the mean scour depth around the pile.

Present model assumes the hypothesis of saturated sediment flows, which gives reasonable results for phenomena simulated in this work, as substantiated by numerics. However, the extension of the numerical model to reproduce non-saturated sediment flows is of great interest to study a wider spectrum of physical processes, and will be pursued and reported in future publications.

**Acknowledgements** The research was supported by the Grant #PID 2020-115778GB-I00 funded by MCIN/AEI/10.13039/501100011033.

**Funding** Funding for open access publishing: Universidad de Granada/CBUA. Open access charge was funded by Universidad de Granada / CBUA.

**Declarations**

**Conflict of interest** The authors declare no conflicts of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

**Appendix: Flux correction basis**

Corrected solution of Eqs. (1)–(3) is established in terms of a high order (HO) and low order (LO) finite element solutions denoted here as **B** and **b**, respectively. High order corrected solution at node *i* of the finite element mesh  $\Omega = \bigcup \Omega_j$ , ( $j = 1, E$ ), at time  $(n+1)\Delta t$  is designated as  $\tilde{\mathbf{B}}_i^{n+1}$ , and is built as

$$\tilde{\mathbf{B}}_i^{n+1} = \mathbf{b}_i^{n+1} + \sum_{j=1}^e \tilde{\mathbf{A}}_j = \mathbf{b}_i^{n+1} + \sum_{j=1}^e \mathbf{c}_j \mathbf{A}_j = \mathbf{b}_i^{n+1} + \sum_{j=1}^e \mathbf{c}_j (\mathbf{A}_{HO} - \mathbf{A}_{LO})_j \tag{17}$$

resulting from updating the LO solution **b** at time  $(n + 1)\Delta t$  by the sum of corrected anti-diffusive element contributions, denoted as  $\tilde{\mathbf{A}}$ , that counterbalance the truncation error of the

LO scheme. In Eq. (17),  $\tilde{\mathbf{A}}_j$  is the corrected anti-diffusive contribution of element  $j$ , while  $\mathbf{A}_j$  is the difference between the contribution of the high order scheme  $\mathbf{A}_{HO}$  and the contribution of the low order scheme  $\mathbf{A}_{LO}$  corresponding to element  $j$ . Sum extends over  $e$ , the number of elements  $j$  surrounding node  $i$ ; summation convention is not used.

Elementwise correction functions  $\mathbf{c}_j$  impose desired properties of LO solution (positivity and, under certain conditions, monotonicity) in the corrected HO solution. Range of  $\mathbf{c}_j$ 's is:  $0 \leq c_{jm} \leq 1$ ,  $c_{jm} \in \mathbf{c}_j$ , where  $m$  denotes the number of variables of the problem ( $m \in [1, D + 1]$  for Eqs. (1) and (2), and  $m = 1$  for Eq. (3)). Note that if  $c_{jm} = 1$  the corrected solution equals the HO solution, while for  $c_{jm} = 0$  solution equals the LO solution [17]. To determine correcting functions values the method defines nodal bounds  $B_i^{min}$ ,  $B_i^{max}$  (here written for a scalar equation to simplify notation) such that  $B_i^{min} \leq \tilde{b}_i^{n+1} \leq B_i^{max}$ . For instance, appropriate and successful nodal bounds were given by Ref. [28] (see equations (17') and (18') therein),

$$B_i^{min} = \min_{j=1,e} (B_i^n, b_i^{n+1}, B_k^n, b_k^{n+1}), \quad \forall (\text{nodes } k \neq i) \in j \quad (18)$$

$$B_i^{max} = \max_{j=1,e} (B_i^n, b_i^{n+1}, B_k^n, b_k^{n+1}), \quad \forall (\text{nodes } k \neq i) \in j \quad (19)$$

where all variables in Eqs. (18) and (19) are positive definite. Correcting functions  $\mathbf{c}_j$  depend on element anti-diffusive contribution  $\mathbf{A}_j$  to the node, on the bounds  $B_i^{min}$ ,  $B_i^{max}$ , and on the LO nodal solution  $b_i^{n+1}$ .

To illustrate practical details of the procedure, consider Fig. 16, where elements and nodes surrounding a node  $i$  are sketched; nodes are numbered as  $i + 1 \dots i + 5$  to simplify notation. Anti-diffusive contributions to node  $i$  have superscript + or -, either if the elementwise flux increases or decreases the value of  $b_i^{n+1}$ . Then, given that corrected anti-diffusive contribution should fulfill the constraint

$$B_i^{min} \leq \mathbf{b}_i^{n+1} + \sum_{j=1}^e \tilde{\mathbf{A}}_j^+ - \sum_{j=1}^e |\tilde{\mathbf{A}}_j^-| \leq B_i^{max}, \quad (20)$$

the following two sufficient conditions arise for the case depicted in Fig. 16,

$$b_i^{n+1} - |\tilde{\mathbf{A}}_2^-| - |\tilde{\mathbf{A}}_3^-| - |\tilde{\mathbf{A}}_5^-| \geq B_i^{min}, \quad (21)$$

$$b_i^{n+1} + \tilde{\mathbf{A}}_1^+ + \tilde{\mathbf{A}}_4^+ \leq B_i^{max}. \quad (22)$$

To find the equation defining correcting functions, we rewrite in compact form the corrected solution (17) at node

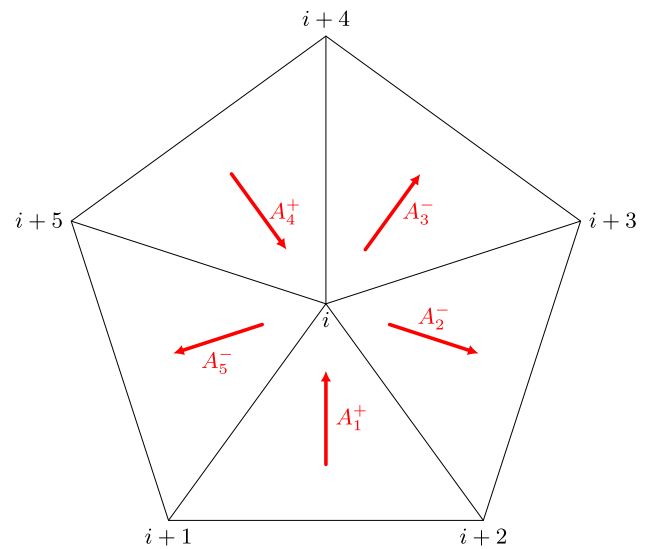


Fig. 16 Anti-diffusive contributions to node  $i$

$i$  (see Eq. (15) in [16]) for a scalar quantity as

$$\tilde{b}_i^{n+1} = b_i^{n+1} + \sum_{j=1}^e \frac{1}{2} [c_j^+ (A_j + |A_j|) + c_j^- (A_j - |A_j|)] \quad (23)$$

and conditions (21)–(22) result in

$$b_i^{n+1} - c_i^- (|A_2^-| + |A_3^-| + |A_5^-|) \geq B_i^{min}, \quad (24)$$

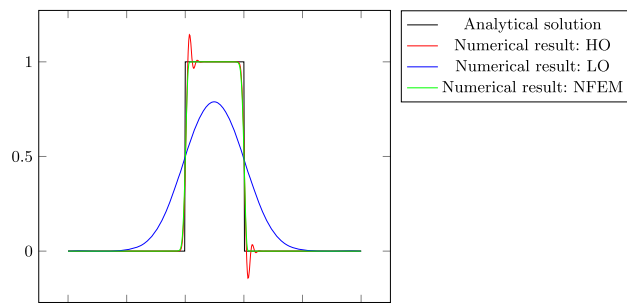
$$b_i^{n+1} + c_i^+ (A_1^+ + A_4^+) \leq B_i^{max}. \quad (25)$$

In these equations,  $c_i^+$ ,  $c_i^-$  indicate nodal correcting functions. A general form of these conditions can be easily reached as,

$$c_i^+ = \min \left( 1, \frac{B_i^{max} - b_i^{n+1}}{\sum_{j=1}^e \frac{1}{2} [A_j + |A_j|] + \zeta} \right)$$

$$c_i^- = \min \left( 1, \frac{b_i^{n+1} - B_i^{min}}{\sum_{j=1}^e \frac{1}{2} [|A_j| - A_j] + \zeta} \right)$$

where a small number  $\zeta$  is integrated to avoid the vanishing of the denominator. Whereas correcting functions are established at nodes, elementwise coefficients used in Eq. (23) are  $c_j^+ = \min(c_k^+)$ ,  $c_j^- = \min(c_k^-)$ ,  $\forall$  nodes  $k \in j$ . If Eq. (17) is employed to recover a single correction coefficient as  $c_j = \min(c_j^+, c_j^-)$ , the corrected solution will be more diffusive and closer to LO answer than that obtained from Eq. (23). Direct extension of scalar correction ( $m = 1$ ) for the complete set of equations of motion ( $m = 1, D + 1$ ) could result in inappropriate constraints on some of the variables. However, the action only on velocity components [10]



**Fig. 17** Analytical solution and numerical results for the advection of a heaviside step function

is an effective approach to coordinate corrections. Now the correcting functions for a node  $i$  are defined as  $c_i = \min(c_{ip})$ , where  $p$  ( $p = 1, D$ ) indicates the velocity component.

As an example to complement this Appendix, the non-oscillatory finite element method is utilized in the simulation of the pure advection of a Heaviside step function  $f(x, t)$  such that, at  $t = 0$ ,

$$f(x, 0) = \begin{cases} 1 & \text{if } x \in [0, 1] \\ 0 & \text{otherwise} \end{cases}.$$

The 1D equation to solve is

$$\frac{\partial f}{\partial t} + \frac{\partial(uf)}{\partial x} = 0,$$

where advective velocity is  $u = 1.0$  m/s. The finite element mesh has  $\Delta x = 0.01$  m, time step  $\Delta t = 0.002$  s and final time is  $T = 20$  s. Figure 17 shows results at  $t = T$  by applying three different methods: a) Characteristic-Galerkin (high order), b) Upwind scheme (low order) and c) NFEM. Analytical solution is superimposed in Fig. 17. It is apparent the conservation of bounds of the low order scheme, although it introduces additional diffusion that spoils the experiment, giving a standard  $L_2$ -norm error of  $e(L_2)_{LO} = 2.005 \times 10^{-1}$ . High order algorithm achieves an error  $e(L_2)_{HO} = 5.672 \times 10^{-2}$ , and, as expected, without bounds preservation. Finally, NFEM reaches a very similar accuracy of the HO solution:  $e(L_2)_{NFEM} = 5.677 \times 10^{-2}$  and solution is monotone.

## References

- Andreotti B, Claudin P, Douady S (2002) Selection of dune shapes and velocities. Part 1: dynamics of sand, wind and barchans. *E.P.J. B Condens Matter Complex Syst* 28:321–339
- Belytschko T, Parimi C, Moës N, Sukumar N, Usui S (2003) Structured extended finite element methods for solids defined by implicit surfaces. *Int J Numer Methods Eng* 56:609–635
- Bravo R, Becker P, Ortiz P (2017) Numerical simulation of evolutionary erodible bedforms using the particle finite element method. *Comp Part Mech* 4:297–305
- Bravo R, Ortiz P, Idelsohn S, Becker P (2020) Sediment transport problems by the particle finite element method. *Comp Part Mech* 7:139–149
- Cruchaga M, Battaglia L, Storti M, D'Elía J (2016) Numerical modeling and experimental validation of free surface flow problems. *Arch Comput Methods Eng* 23:139–169
- Duran O, Andreotti B, Claudin P (2012) Numerical simulation of turbulent sediment transport, from bed load to saltation. *Phys Fluids* 24:103306
- Gibou F, Hyde D, Fedkiw R (2019) Sharp interface approaches and deep learning techniques for multiphase flows. *J Comput Phys* 380:442–463
- Labergere C, Rassinoux A, Saanouni K (2014) Numerical simulation of continuous damage and fracture in metal-forming processes with 2D mesh adaptive methodology. *Finite Elem Anal Des* 82:46–61
- Lettau H, Lettau K (1978) Exploring the world's driest climate. Report of an expedition to Pampa de la Joya, Peru, in July 1964. Center for Climatic Research, U. of Wisconsin-Madison
- Löhner R, Morgan K, Peraire J, Vahdati M (1987) Finite element flux-corrected transport (FEM-FCT) for the Euler and Navier-Stokes equations. *Int J Numer Methods Fluids* 7:1093–1109
- Molina J, Ortiz P (2019) A conservative flux-corrected continuous FEM for fluid interface dynamics. *Int J Numer Methods Fluids* 91:287–310
- Molina J, Ortiz P (2020) A continuous finite element solution of fluid interface propagation for emergence of cavities and geysering. *Comput Methods Appl Mech Eng* 359:112746
- Molina J, Ortiz P (2022) Propagation of large air pockets in ducts. Analytical and numerical approaches. *Appl Math Model* 110:633–662
- Molina J, Ortiz P, Bravo R (2023) A finite element method for partially erodible bed evolution coupled with multiphase flows. *Comput Methods Appl Mech Eng* 405:115853
- Nagarajan A, Soghrati S (2018) Conforming to interface structured adaptive mesh refinement: 3D algorithm and implementation. *Comput Mech* 62:1213–1238
- Ortiz P (2009) A positive definite continuous FEM model for advection. *Adv Water Resour* 32(8):1359–1371
- Ortiz P (2012) Non-oscillatory continuous FEM for transport and shallow water flows. *Comput Methods Appl Mech Eng* 223–224:55–69
- Ortiz P, Anguita J, Riveiro M (2015) Free surface flows over partially erodible beds by a continuous finite element method. *Environ Earth Sci* 74:7357–7370
- Ortiz P, Smolarkiewicz PK (2006) Numerical simulation of sand dune evolution in severe winds. *Int J Numer Methods Fluids* 50:1229–1246
- Ortiz P, Smolarkiewicz PK (2009) Coupling the dynamics of boundary layers and evolutionary dunes. *Phys Rev E* 79(4):041307-1–041307-11
- Poluektov M, Figiel Ł (2022) A cut finite-element method for fracture and contact problems in large-deformation solid mechanics. *Comput Methods Appl Mech Eng* 388:114234
- Quan S, Schmidt DP (2007) A moving mesh interface tracking method for 3D incompressible two-phase flows. *J Comput Phys* 221:761–780
- Schwämmle V, Herrmann S (2004) Modelling transverse dunes. *Earth Surf Process Landf* 29:769–784
- Soghrati S, Nagarajan A, Liang B (2017) Conforming to interface structured adaptive mesh refinement: new technique for the automated modeling of materials with complex microstructures. *Finite Elem Anal Des* 125:24–40
- Struiksmá N (1999) Mathematical modelling of bedload transport over non-erodible layers. In: IAHR symposium on river, coastal and estuarine morphodynamics, Genoa, pp 89–98
- Weng WS, Hunt JCR, Carruthers DJ, Warren A, Wiggs GFS, Livingstone I, Castro I (1991) Air flow and sand transport over sand-dunes. *Aeolian Grain Transport*. Springer, Vienna, pp 1–22

27. Wippermann FK, Gross G (1986) The wind-induced shaping and migration of an isolated dune: a numerical experiment. *Bound Layer Meteorol* 36:319–334
28. Zalezak ST (1979) Fully multidimensional flux-corrected transport algorithms for fluids. *J C Phys* 31:335–362
29. Zhao M, Cheng L, Zang Z (2010) Experimental and numerical investigation of local scour around a submerged vertical circular cylinder in steady currents. *Coast Eng* 57:709–721

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.