Received: 13 June 2022 | Revised: 19 December 2022 | Accepted: 9 February 2023

DOI: 10.1002/fld.5186

RESEARCH ARTICLE

WILEY

Parallel simulations for fast-moving landslides: Space-time mesh adaptation and sharp tracking of the wetting front

Federico Gatti | Marco Fois | Carlo de Falco | Simona Perotto | Luca Formaggia

MOX – Department of Mathematics, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy

Correspondence
Federico Gatti, MOX – Department of Mathematics, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy.
Email: federico.gatti@polimi.it

Funding information
Agenzia Spaziale Italiana (Accordo Attuativo ASI-POLIMI "Attivit\a di Ricerca e Innovazione"), Grant/Award Number: 2018-5-HH.0

Abstract

We propose a highly scalable solver for a two-dimensional depth-integrated fluid dynamic model in order to simulate flow-like landslides, such as debris or mud flows. The governing equations are discretized on quadtree meshes by means of a two-step second-order Taylor–Galerkin scheme, enriched by a suitable flux correction in order to avoid spurious oscillations, in particular near discontinuities and close to the wetting-drying interface. A mesh adaptation procedure based on a gradient-recovery a posteriori error estimator allows us to efficiently deal with a discretization of the domain customized to the phenomenon under investigation. Moreover, we resort to an adaptive scheme also in time to prevent filtering out the landslide dynamics, and to an interface tracking algorithm to avoid an excessive refinement in noninterfacial regions while preserving details along the wetting-drying front. Finally, after verifying the performance of the proposed numerical method on idealized settings, we carry out a scalability analysis of the code both on idealized and real scenarios, to check the efficiency of the overall implementation.

KEYWORDS

depth-integrated model, flow-like landslides, parallel computing, quadtree mesh, space-time adaptation, Taylor–Galerkin scheme

1 | INTRODUCTION AND MOTIVATION

Hydrogeological instability is among the effects of climate change with the highest impact on the safety of people and of the built environment. In particular, landslides are responsible for significant human and economic losses worldwide.[1,2] The capability to predict landslides and to assess the risk connected with extreme events is of paramount importance to the safety of people and infrastructures,[3] especially in densely populated urban areas located in landslide-prone regions.[4] The dynamics of a landslide is characterized by a broad range of velocity scales, each being dominant in a particular phase of the event, from the steady creeping slip to a catastrophic avalanche, passing through the intermittent rapid slip. During these phases, the landslide displays different mechanical behaviors. In particular, during the triggering phase, the landslide behaves roughly like a rigid body and the driving process is the pore-pressure diffusion that causes the intermittent slipping of the involved material. Once the landslide is initiated, various behaviors take place, in particular a flow-like motion is typical of debris and mud flows, where the landslide follows a visco-plastic behavior and the overall process becomes advection-dominated.

In this paper, we focus on the "rapid" movement of flow-like landslides, that is, a movement characterized by a relatively large velocity. In other words, we assume that the mass has lost its equilibrium and we primarily analyze the associated run-out. We propose an efficient implementation of a two-dimensional depth-integrated fluid dynamic model able to simulate such kind of landslides. Following previous works,[5-7] the landslide model equations are numerically solved by using a Taylor-Galerkin method.[8,9] We implement a parallelized version of the two-step Taylor–Galerkin scheme[10] on h-adaptive quadtree meshes, in order to handle the different length-scales characterizing a landslide dynamics, while ensuring a reasonable computational burden. The spatial mesh adaptation is carried out via a metric-based iterative algorithm driven by a recovery-based a posteriori estimator for the discretization error.[11,12] Successively, we enrich the spatial adaptation with an adaptive choice also of the time step, following Reference 13. The quadtree data structure[14] allows us to exploit highly efficient algorithms for mesh traversal, partitioning, refinement, coarsening and balancing that are particularly well suited for scaling up to very large number of parallel processes on distributed-memory systems.[15-23] In particular, the use of such data structure for the mesh management allows to reduce the communication required in the error estimation and refinement/coarsening steps, that are often a bottleneck in the parallel implementation of adaptive simulation algorithms.[24]

The ability of being relatively easily deployed on large clusters and to fully exploit the available computing power represents a key requirement for the applicability of the proposed solver. Indeed, the main use case we have in mind is the integration of the code within a full stack of tools for environmental risk monitoring, which combines physics-based numerical simulations with a statistical data analysis, in order to predict catastrophic events and to assess the associated risk level.[25] In this framework, a large number of simulation runs may be needed on each scenario, either to evaluate the possible outcome of a specific event or to continuously calibrate the model parameters with respect to observations, for example, in a data assimilation process. For all these reasons, the algorithm design process we pursue is driven by the need to achieve maximum parallel scalability.

The paper is organized as follows. We present the governing equations for fast flowing landslide modeling in Section 2, while Section 3 gathers all the numerical tools used to approximate the selected model. In particular, Section 3.1 introduces the discrete spaces on balanced quadtree meshes with hanging nodes. Section 3.2 defines the two-step Taylor–Galerkin discretization algorithm. Section 3.3 provides details on the procedure used to track the wetting–drying interface. Finally, Sections 3.4 and 3.5 furnish the error estimator together with the metric-based adaptation procedure used to select both the spatial and the temporal computational mesh. In Section 4, we carry out the numerical assessment. In order to verify the performance of the proposed discretization setting, we compare discrete with corresponding analytical solutions. Moreover, we execute a scalability analysis, both in case of idealized and real configurations, in order to check the actual efficiency of the overall implementation. Finally, in Section 5, we draw some conclusions and offer perspectives for possible future developments of the current approximation framework.

2 | DEPTH-AVERAGED MODELING OF FLOW-LIKE LANDSLIDES

In this section, we introduce the governing equations used to model the fast landslide material (notice that the model here adopted is reliable as long as the substrate is not too slanted). Inside a Cartesian domain $\Omega \subset \mathbb{R}^2$ and in the time window $(0, T]$, we solve the de Saint-Venant equations

$$\begin{cases} \partial_t h + \partial_x U_x + \partial_y U_y = 0, \\ \partial_t U_x + \partial_x\left(\frac{U_x^2}{h} + \frac{1}{2}gh^2\right) + \partial_y\left(\frac{U_xU_y}{h}\right) = \dfrac{1}{\rho}f_x^B + \dfrac{1}{\rho}\partial_x(\sigma_{xx}h) + \dfrac{1}{\rho}\partial_y(\sigma_{xy}h) - gh\partial_x Z, \\ \partial_t U_y + \partial_x\left(\frac{U_yU_x}{h}\right) + \partial_y\left(\frac{U_y^2}{h} + \frac{1}{2}gh^2\right) = \dfrac{1}{\rho}f_y^B + \dfrac{1}{\rho}\partial_x(\sigma_{xy}h) + \dfrac{1}{\rho}\partial_y(\sigma_{yy}h) - gh\partial_y Z, \end{cases} \tag{1}$$

with respect to the unknown functions h and $\mathbf{U} = [U_x, U_y]^\top$ modeling the material depth and the horizontal mass flux, respectively, to be supplemented by proper initial and boundary conditions. In particular, following References 6 and 7, we distinguish in Ω a wet (Ω_w) and a dry $(\Omega \setminus \Omega_w)$ region, defined as the portion of the whole domain where $h > h_{\min}$ and $h \leq h_{\min}$, respectively, with $h_{\min}$ a threshold under which we have a null momentum flux. This trick allows us solving (1) on the whole Ω instead of on the wet region only. This leads to a considerable computational advantage taking into account that Ω_w varies in space and time. Moreover, we can assign boundary data directly on the boundary $\partial\Omega$ instead of on a curve changing in time. In particular, we apply nonreflecting boundary conditions.

In Equation (1), g is the gravitational acceleration, ρ is the density of the material, $\mathbf{f}^B = [f_x^B, f_y^B]^\top$ denotes the bed friction, $\boldsymbol{\sigma}$ is the deviatoric part of the Cauchy stress tensor, with components $[\sigma_{xx}, \sigma_{xy}, \sigma_{yy}]$, and Z describes the orography profile. In particular, the bed friction is defined by the Voellmy rheology, so that

$$\mathbf{f}^B = -\left(p_B \tan\delta \, \frac{1}{|\mathbf{v}|} + \rho g \, \frac{|\mathbf{v}|}{\xi}\right)\mathbf{v}, \tag{2}$$

where p_B is the bed pressure given by the Stevino law, being $p_B = p_S + \rho g h$, with p_S the atmospheric pressure, δ is the bed friction angle, $\mathbf{v} = [v_x, v_y]^\top = \mathbf{U}/h$ is the horizontal depth-averaged velocity, and ξ is a turbulence friction coefficient with the same dimension as for the acceleration.

Law (2) turns out to be particularly suited in the modeling of debris flows or rock avalanches.[26,27] We also remark that, as proposed in other works,[28-30] the bed friction angle should be considered time dependent due to consolidation processes. However, we consider δ as a constant since the main goal of the paper is to assess the scalability performance of the proposed implementation rather than focusing on modeling aspects that do not affect the performance of the overall framework.

Concerning the choice for $\boldsymbol{\sigma}$, we employ a depth-integrated visco-plastic Bingham stress model, that is

$$\boldsymbol{\sigma} = \left(\frac{\tau_Y}{\sqrt{I_2}} + 2\mu\right)\mathbf{D}, \tag{3}$$

where τ_Y is the yield shear stress, μ is the fluid viscosity, $\mathbf{D}$ denotes the horizontal strain rate tensor defined component-wise by $D_{ij} = \frac{1}{2}\left(\partial_{x_j} v_i + \partial_{x_i} v_j\right)$ for $i,j = 1,2$, with $x_1 = x$, $x_2 = y$, $v_1 = v_x$, $v_2 = v_y$, and where $I_2 = \frac{1}{2}\overline{\mathbf{D}} : \overline{\mathbf{D}}$ represents the second invariant of the three-dimensional depth-averaged strain rate tensor

$$\overline{\mathbf{D}} = \begin{bmatrix} D_{11} & D_{12} & \frac{1}{2}\partial_z v_x \\ D_{21} & D_{22} & \frac{1}{2}\partial_z v_y \\ \frac{1}{2}\partial_z v_x & \frac{1}{2}\partial_z v_y & -(D_{11} + D_{22}) \end{bmatrix},$$

component $\overline{D}_{33}$ being computed via the incompressibility constraint. In particular, following Reference 7, we evaluate I_2 under the hypotheses of steady state, laminar, simple shear flow, so that the depth-integrated rate of deformation tensor along the normal direction, z, is identified by relations

$$\partial_z v_i = \frac{3}{2+\zeta}\frac{v_i}{h}, \tag{4}$$

for $i = 1, 2$, where $\zeta = \tau_Y/\tau_B$, with $\zeta \in [0, 1]$ and τ_B the bed friction. Now, the relation between $\mathbf{v}$ and ζ,

$$|\mathbf{v}| = \frac{\tau_B}{6\mu}(1-\zeta)^2(2+\zeta), \tag{5}$$

results in the third-degree polynomial

$$\zeta^3 - (3+a)\zeta + 2 = 0, \tag{6}$$

in the unknown ζ, being $a = (6\mu|\mathbf{v}|)/(h\tau_Y)$. By applying the best second approximation of a third-degree polynomial which provides a maximum error equal to $1/32$, we obtain the equation

$$\frac{3}{2}\zeta^2 - \left(\frac{114}{32} + a\right)\zeta + \frac{65}{32} = 0, \tag{7}$$

that can be solved in a closed form. Once the equation is solved for ζ, we can compute quantities (4) (i.e., the two components $\overline{D}_{13} = \overline{D}_{31}$ and $\overline{D}_{23} = \overline{D}_{32}$) and, consequently, the invariant I_2.

3 | THE NUMERICAL METHOD

To simplify the discussion, we rewrite system (1) as a generic nonlinear system of conservation laws, that is, as

$$\partial_t \mathbf{q} + \nabla \cdot \mathbf{F} + \nabla \cdot \mathbf{D} = \mathbf{r}, \quad \text{in } \Omega \times (0, T], \tag{8}$$

to be supplemented with proper initial and boundary conditions. Here, $\mathbf{q} = \mathbf{q}(\mathbf{x}, t)$ is the vector of the conserved variables, being $\mathbf{x} = (x, y) \in \Omega \subset \mathbb{R}^2$ and $t \in (0, T]$, $\mathbf{F} = \mathbf{F}(\mathbf{q})$ and $\mathbf{D} = \mathbf{D}(\nabla \mathbf{q})$ denote the tensor of the transport and of the diffusive fluxes respectively, while $\mathbf{r} = \mathbf{r}(\mathbf{q})$ is the reaction term.

In particular, with reference to system (1), it turns out that $\mathbf{q} = [h, \mathbf{U}]^\top$, tensor $\mathbf{F}(\mathbf{q}) = [\mathbf{F}_x(\mathbf{q}), \mathbf{F}_y(\mathbf{q})]$ has components defined by

$$\mathbf{F}_x(\mathbf{q}) = \begin{bmatrix} U_x \\ \dfrac{U_x^2}{h} + \dfrac{1}{2} g h^2 \\ \dfrac{U_y U_x}{h} \end{bmatrix}, \quad \mathbf{F}_y(\mathbf{q}) = \begin{bmatrix} U_y \\ \dfrac{U_y U_x}{h} \\ \dfrac{U_y^2}{h} + \dfrac{1}{2} g h^2 \end{bmatrix},$$

the diffusive flux $\mathbf{D}(\nabla \mathbf{q}) = [\mathbf{D}_x(\nabla \mathbf{q}), \mathbf{D}_y(\nabla \mathbf{q})]$ is characterized by the components

$$\mathbf{D}_x(\nabla \mathbf{q}) = \begin{bmatrix} 0 \\ -\dfrac{1}{\rho} \sigma_{xx} h \\ -\dfrac{1}{\rho} \sigma_{xy} h \end{bmatrix}, \quad \mathbf{D}_y(\nabla \mathbf{q}) = \begin{bmatrix} 0 \\ -\dfrac{1}{\rho} \sigma_{xy} h \\ -\dfrac{1}{\rho} \sigma_{yy} h \end{bmatrix},$$

while the reaction term coincides with

$$\mathbf{r}(\mathbf{q}) = \begin{bmatrix} 0 \\ \dfrac{1}{\rho} f_x^B - g h \partial_x Z \\ \dfrac{1}{\rho} f_y^B - g h \partial_y Z \end{bmatrix}.$$

In the sequel, we deal with the space-time discretization scheme adopted to approximate problem (8) (i.e., (1)), endowed with a wetting-drying interface tracking algorithm. In particular, both the spatial and the temporal discretizations of the domain $\Omega \times (0, T]$ will be driven by a mesh adaptation procedure detailed in Sections 3.4 and 3.5.

3.1 | Discrete spaces on hierarchically refined Cartesian quadtree meshes

To offer a sharp description of the solution in wet regions and, at the same time, to prevent a smearing of the solution near wet-dry interfaces, we resort to a spatial mesh adaptation procedure. In more detail, we carry out an isotropic spatial adaptation on quadtree meshes, with the constraint to have adjacent cells whose size differs at most by a factor equal to two. The quadtree data structure easily enables both coarsening and refinement although it leads to nonconformal meshes characterized by the presence of hanging nodes. A mesh refinement consists in replacing an element with four children of equal size (see Figure 1, top), while coarsening occurs when removing four children and replacing them with a parent (see Figure 1, bottom).

Concerning the update of the discrete variables on adapted meshes, we compute the solution on the new grid by means of a bilinear interpolation of the solution computed on the previous mesh, while the orography characterizing

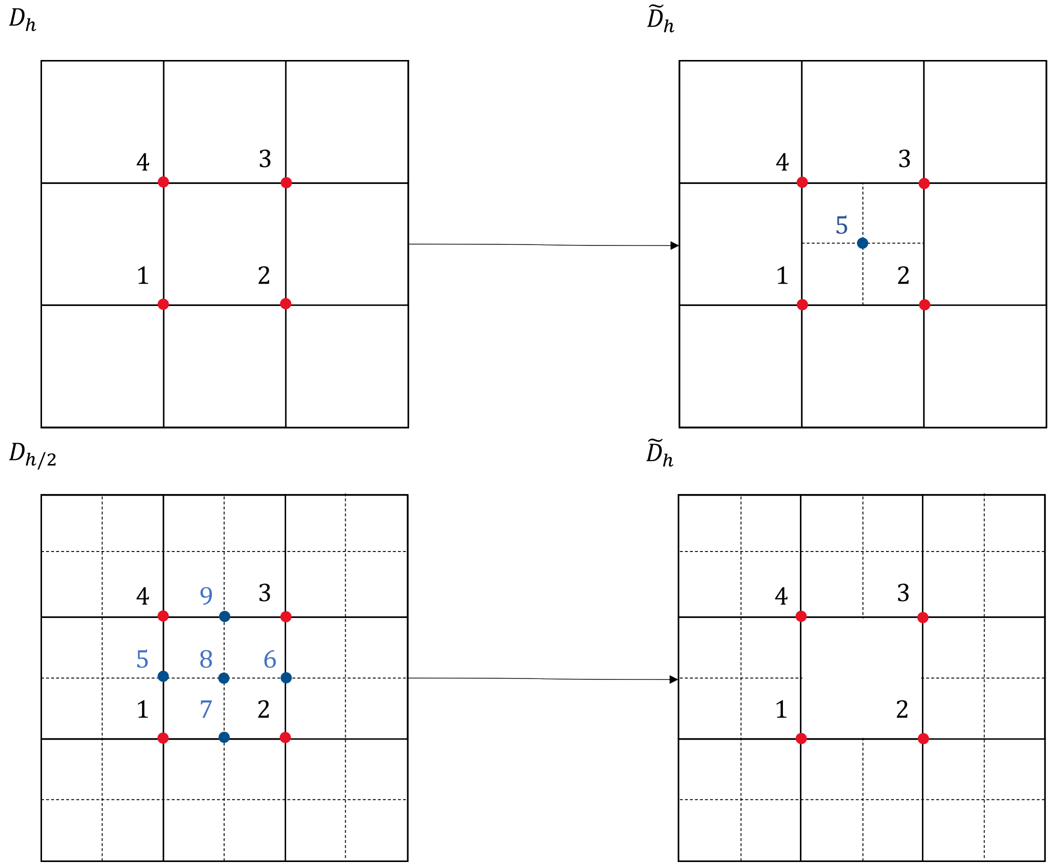

FIGURE 1 Example of refinement (top) and coarsening (bottom) for a quadtree mesh. The dofs characterizing space $\tilde{\mathbb{Q}}_1^h$ are highlighted (notice that the hanging nodes do not enrich the space). [Colour figure can be viewed at wileyonlinelibrary.com]

the source term is recomputed directly on the input orography, that is, via the Digital Terrain Model (DTM), associated with a regular quadrilateral grid. In particular, we update the slope since the source term of the momentum equation involves the terrain slant, both along the x- and y-direction. Thus, for each DTM cell, we compute, once and for all, the terrain slope along both the Cartesian directions with the divergence theorem, by considering as integration walk the diamond-shape path joining the centroids of the four cells sharing an edge with the considered DTM cell. Successively, at each quadtree element, we assign a unique slope value coinciding with the value taken by the orography at the centroid of the quadtree cell. We observe that this procedure is consistent with the case of a planar orography. Moreover, in view of a discrete setting, it corresponds to approximate the integrals involved in the weak formulation of problem (8) with a midpoint quadrature rule.

Following Reference 16, the presence of the hanging nodes is handled by modifying the discrete space instead of adding new degrees of freedom (see Figure 1), with the goal of preserving the partition of unity property. To this aim, we introduce the standard space $\mathbb{Q}_1^h$ of the continuous piecewise bilinear polynomials associated with the regular quadtree mesh $\mathcal{D}_h$ to discretize the material depth as well as the two components of the mass flux Reference 31 (chapter 2). Then, we define the modified discrete space $\tilde{\mathbb{Q}}_1^h \subset \mathbb{Q}_1^h$, which still collects continuous piecewise bilinear polynomials, where the value of the conservative quantities at the hanging nodes is identified with the arithmetic mean of the values taken by the same quantities at the corresponding parent nodes.

Now, we assume to have a maximum level M of uniform refinements. If we denote by $\{\phi_k^{(1),h/2^l},\ k = 1, \dots N_l^{(1)}\}$ the standard finite element basis of $\mathbb{Q}_1^{h/2^l}$, for $l = 0, \dots, M$, and by $\{\tilde{\phi}_i^{(1)},\ i = 1, \dots \tilde{N}^{(1)}\}$ the basis of $\tilde{\mathbb{Q}}_1^h$, we can identify two levels of refinement, say $l_1 = 0, \dots, M-1$ and $l_2 = l_1 + 1$, such that

$$\tilde{\phi}_i^{(1)}(\mathbf{x}) = \sum_{k=1}^{N_{l_1}^{(1)}} w_{k,l_1}\phi_k^{(1),h/2^{l_1}}(\mathbf{x}) + \sum_{k=1}^{N_{l_2}^{(1)}} w_{k,l_2}\phi_k^{(1),h/2^{l_2}}(\mathbf{x}), \tag{9}$$

for any $\mathbf{x} \in \Omega$. In particular, the weights w_{k,l_1}, w_{k,l_2} are determined by imposing the partition of unity property,

$$\sum_{i=1}^{\tilde{N}^{(1)}} \tilde{\phi}_i^{(1)}(\mathbf{x}) = 1, \tag{10}$$

for any $\mathbf{x} \in \Omega$, together with the Lagrangian property,

$$\tilde{\phi}_i^{(1)}(\mathbf{x}_j) = \delta_{ij}, \tag{11}$$

for $i, j = 1, \ldots, \tilde{N}^{(1)}$, with δ_{ij} the Kronecker delta.

3.2 | Two-step Taylor–Galerkin discretization on quadtrees

Following Reference 7, we approximate system (8) by resorting to a two-step Taylor–Galerkin (TG2) method on a spatial quadtree mesh characterized by a 2:1 balance ratio.

In more detail, to deal with the time discretization, we consider the Taylor series expansion of the conserved variable $\mathbf{q}$ around a given time t^n, which, when evaluated at time t^{n+1}, is given by

$$\mathbf{q}^{n+1} = \mathbf{q}^n + \Delta t\ \partial_t \mathbf{q}^n + \frac{\Delta t^2}{2}\ \partial_t^2 \mathbf{q}^n + \mathcal{O}(\Delta t^3), \tag{12}$$

with $\mathbf{q}^j = \mathbf{q}(\mathbf{x}, t^j)$ and $j = n, n+1$. Thus, indicating by $\mathbf{Q}^n \approx \mathbf{q}^n$ the time discrete conservative variable at the generic time t^n, we have

$$\mathbf{Q}^{n+1} = \mathbf{Q}^n + \Delta t\ \partial_t \left(\mathbf{Q}^n + \frac{\Delta t}{2}\ \partial_t \mathbf{Q}^n \right). \tag{13}$$

Now, after introducing an intermediate state, $\mathbf{Q}^{n+\frac{1}{2}}$, we can formalize the semi-discrete two-step temporal scheme as

$$\begin{cases} \mathbf{Q}^{n+\frac{1}{2}} = \mathbf{Q}^n + \frac{\Delta t}{2}\ \partial_t \mathbf{Q}^n, \\ \mathbf{Q}^{n+1} = \mathbf{Q}^n + \Delta t\ \partial_t \mathbf{Q}^{n+\frac{1}{2}}. \end{cases} \tag{14}$$

Concerning the spatial discretization, we employ standard finite elements, based on a quadtree partition, $\tilde{D}_h$, of Ω. We associate different approximation spaces with the two temporal steps in (14). We use the space $\mathbb{Q}_0$ of the continuous constant polynomials to discretize the first step (i.e., to compute the intermediate state $\mathbf{Q}^{n+\frac{1}{2}}$), and the space $\tilde{\mathbb{Q}}_1^h$, described in Section 3.1, to discretize the second step (i.e., to compute the final state $\mathbf{Q}^{n+1}$).

Thus, the discrete counterpart of the two steps in (14) is provided by the equations

$$\begin{aligned} &(\mathbf{Q}^{n+\frac{1}{2}}, \phi_j^{(0)}) = (\mathbf{Q}^n, \phi_j^{(0)}) - \frac{\Delta t}{2}\left(\nabla \cdot \mathbf{F}^n, \phi_j^{(0)}\right) + \frac{\Delta t}{2}\left(\mathbf{r}^n, \phi_j^{(0)}\right), \\ &(\mathbf{Q}^{n+1}, \tilde{\phi}_i^{(1)}) = (\mathbf{Q}^n, \tilde{\phi}_i^{(1)}) + \Delta t\left[\left(\mathbf{F}^{*,n+\frac{1}{2}}, \nabla\tilde{\phi}_i^{(1)}\right) + \left(\mathbf{r}^{n+\frac{1}{2}}, \tilde{\phi}_i^{(1)}\right) + \left(\mathbf{D}^n, \nabla\tilde{\phi}_i^{(1)}\right)\right] - \Delta t \int_{\partial\Omega} \left(\mathbf{F}^{*,n+\frac{1}{2}}\mathbf{n} + \mathbf{D}^n\mathbf{n}\right)\ \tilde{\phi}_j^{(1)} d\sigma, \end{aligned} \tag{15}$$

by properly varying indices j and i, where $\mathbf{Q}^n, \mathbf{Q}^{n+1} \in \tilde{\mathbb{Q}}_1^h$, $\mathbf{Q}^{n+\frac{1}{2}} \in \mathbb{Q}_0$, and where $\mathbf{F}^n$, $\mathbf{D}^n$ are the transport and the diffusive fluxes at time t^n, $\{\phi_j^{(0)}\}_i$ and $\{\tilde{\phi}_i^{(1)}\}_j$ denote the set of the basis functions for spaces $\mathbb{Q}_0$ and $\tilde{\mathbb{Q}}_1^h$, respectively, $(\cdot, \cdot)$ is the standard scalar product in $L^2(\Omega)$, $\mathbf{F}^{*,n+\frac{1}{2}}$ represents the numerical flux discussed below at time $t^{n+\frac{1}{2}}$, and $\mathbf{n}$ coincides with the outward unit normal to the boundary $\partial\Omega$. Both the equations in (15) exploit the conservation law in (8). However, we remark that the diffusive term is neglected in the intermediate state according to what done in Reference 7. The reason is that $\mathbf{D}$ depends on the gradient of $\mathbf{q}$, but the gradient is not computable in $\mathbb{Q}_0$. Vice versa, the diffusive contribution is computable in $\tilde{\mathbb{Q}}_1^h$. For this reason, the final state comprehends the diffusive term evaluated at time t^n since no corresponding approximation is available at the intermediate time $t^{n+\frac{1}{2}}$. This choice is similar to an operator splitting strategy and only mildly affects the convergence rate in time of the whole numerical procedure, consistently with what observed in Reference 7. We observe that the TG2 scheme can be conceived as a combination between the explicit Euler scheme, used to discretize the diffusion term, and a second order Taylor method to approximate the transport and the reaction contributions.

Concerning the boundary conditions closing problem (15), we note that the effect of the boundary data on the actual flow is not relevant as far as the wetting front does not reach the boundary $\partial\Omega$. Therefore, we employ nonreflecting boundary conditions for the transport contribution, while we impose null diffusive interface fluxes.

Finally, from a computational viewpoint, we avoid the expensive consistent mass matrix inversion associated with the second step in (15) by adopting a mass lumping technique.[32] This turns out to be a strategical choice in view of a parallel implementation of the numerical procedure. Indeed, we can avoid to build a global linear system, thus skipping expensive communications among processors.

The TG2 scheme in (15) is a second order space-time method. Nevertheless, for $\mathbf{F}^{*,n+\frac{1}{2}} = \mathbf{F}^{n+\frac{1}{2}}$, the scheme is neither monotone nor positive preserving.[10] Here, we are interested in setting up a numerical scheme able to avoid spurious oscillations near discontinuities while ensuring mass conservation. This requires the implementation of suitable flux limiter techniques, like the Flux Corrected Transport (FCT; we refer the interested reader to References 33-35 for further details). In particular, we resort to a first-0order monotone numerical flux (i.e., a “low-order” flux), weighted by a flux correction function that guarantees a first-order accuracy near discontinuities, in accordance with the Godunov theorem. We use a Lax-Friedrichs “low-order” numerical flux with Rusanov correction in order to modify the truncation error only, thus enabling for independent space and time adaptation. Let us focus on a single mesh element Q, with resolution $\Delta x \times \Delta y$. We define the Lax–Friedrichs anti-diffusive contribution with Rusanov correction,

$$\delta\mathbf{F}_Q^n = \max_{\left(\frac{\Delta x}{\Delta t},\frac{\Delta y}{\Delta t}\right)} \frac{1}{2\Delta t}(\nabla\mathbf{Q}^n, \phi_Q^{(0)}), \tag{16}$$

with $\phi_Q^{(0)}$ the generic $\mathbb{Q}_0$-basis function associated with the element Q. The $L^2(\Omega)$-product in (16) leads us to deal with a linear function of the ratios $\Delta x/\Delta t$ and $\Delta y/\Delta t$ for both the x- and the y-components. Now, since such ratios are upper bounded by the CFL condition, the maximum in (16) is reached when replacing $\Delta x/\Delta t$ and $\Delta y/\Delta t$ with the maximum simple wave speed with respect to both the Cartesian directions and in the considered element, this yielding the Rusanov anti-diffusive flux recipe.[36] Thus, according to the classical FCT strategy,[33-35] the numerical flux associated with the cell Q, $\mathbf{F}_Q^{*,n+\frac{1}{2}}$, reads,

$$\mathbf{F}_Q^{*,n+\frac{1}{2}} = (\mathbf{F}_Q^{n+\frac{1}{2}} - \delta\mathbf{F}_Q^n) + \alpha_Q\delta\mathbf{F}_Q^n, \tag{17}$$

with $\mathbf{F}_Q^{n+\frac{1}{2}}$ the restriction to Q of the transport flux $\mathbf{F}$ at time $t^{n+\frac{1}{2}}$, and where $\alpha_Q \in [0, 1]$ denotes the elementwise constant correction coefficient. We adopt the Zalesak multidimensional flux correction, as described in the corresponding seminal paper.[33] The main steps leading to this specific definition of α_Q are here summarized for completeness.

For each mesh node i and for each component z of the conserved variable $\mathbf{q}$, we compute the low-order updated discrete solution $z_i^{n+1,low}$ at time t^{n+1} by considering in the second step of (15) the low order numerical flux obtained by setting $\alpha_Q = 0$ in (17), for each Q . We define now the set, $\mathcal{N}_i$, of the elements sharing the node i, by including also the elements such that at least one hanging node has node i as a parent. Then, the corrected updated solution at node i at time t^{n+1} is given by

$$z_i^{n+1} = z_i^{n+1,low} + \frac{\Delta t}{m_i}\sum_{Q\in\mathcal{N}_i}\alpha_Q f_{Q,i}^z, \tag{18}$$

where m_i is the component of the lumped mass matrix associated with node i, while $f_{Q,i}^z$ represents the component related to z of the vector $\mathbf{f}_{Q,i} = (\delta\mathbf{F}_Q^n, \nabla\tilde{\phi}_i^{(1)})$, that is, of the anti-diffusive contribution to node i from element Q. Now, to determine α_Q, according to References 35,37, we introduce the auxiliary quantities

$$\begin{aligned} P_i^+ &= \tfrac{1}{\Delta t S_{i,x} S_{i,y}} \sum_{Q\in\mathcal{N}_i} \max\{0, f_{Q,i}^z\}, & P_i^- &= \tfrac{1}{\Delta t S_{i,x} S_{i,y}} \sum_{Q\in\mathcal{N}_i} \min\{0, f_{Q,i}^z\}, \\ W_i^+ &= \max_{Q\in\mathcal{N}_i} z_i^{n+1,\text{low}} - z_i^{n+1,\text{low}}, & W_i^- &= \min_{Q\in\mathcal{N}_i} z_i^{n+1,\text{low}} - z_i^{n+1,\text{low}}, \end{aligned} \tag{19}$$

which coincide with the sum of all the positive/negative anti-diffusive fluxes associated with node i and with the distance to the local extrema,[37] respectively, $S_{i,x}$ and $S_{i,y}$ denoting the x- and the y-components of the maximum simple wave speed at the node i associated with the updated low-order solution. Finally, after introducing the nodal quantities,

$$R_i^+ = \begin{cases} \min\{1, W_i^+/P_i^+\} & \text{if } P_i^+ \neq 0 \\ 1 & \text{if } P_i^+ = 0, \end{cases} \qquad R_i^- = \begin{cases} \min\{1, W_i^-/P_i^-\} & \text{if } P_i^- \neq 0 \\ 1 & \text{if } P_i^- = 0, \end{cases} \tag{20}$$

we define the correction factor associated with the generic element Q, that is,

$$\alpha_Q = \min_i \begin{cases} R_i^+, & \text{if } f_{Q,i}^z \geq 0 \\ R_i^-, & \text{if } f_{Q,i}^z < 0, \end{cases} \tag{21}$$

which guarantees the maximum discrete principle.

To conclude, the numerical flux $\mathbf{F}^{*,n+\frac{1}{2}}$ in (15) is defined by collecting the elementwise contributions in (17).

3.3 | Tracking of the Wetting–Drying interface

The discetization scheme in the previous section is enriched with the Lagrangian interface prediction strategy introduced in[38] in order to accurately track the evolution of the wetting–drying interface. The spatial mesh adaptation procedure detailed in the next section also will play a role in such a direction.

Thus, according to the modeling in Section 2, the wet region is discretized by setting a threshold, $h_{\min}$, for the depth h so that no momentum flux takes place under this value. This threshold has to be chosen sufficiently small to prevent unrealistic behaviors, although not too small to avoid the rise of numerical instabilities yielded by the division by h in the definition of the transport fluxes.

The strategy we adopt to track the wetting– interface is very basic. At a given time t^n, first we identify the wetting-drying interface elements, coinciding with the cells which have, at least, a dry node (where the discrete material height is under $h_{\min}$) as well as a wet node (where the discrete value for h is above $h_{\min}$). For each interface element, Q_I, we compute the average velocity. Then, we identify the neighbouring cell containing the centroid of Q_I when advected by such an average field. This localized search turns out to be possible by exploiting the CFL condition. Finally, the neighboring element is marked for refinement, up to a suitable refinement level (we refer to the next section, in particular to formula (27), for further details, and to Figure 2 for a sketch of the front-tracking procedure).

To avoid an excessive smearing of the solution when interpolating the physical quantities from one mesh to another, we avoid coarsening the cells along the interface.

3.4 | Space adaptation

The wetting–drying interface tracking process is complemented by an adaptive management of the spatial mesh, driven by an a posteriori error estimator. Among the several estimators for the discretization error available in the literature, we select a gradient recovery approach.[11,12,39] One of the main strengths of these estimators is the intrinsic independence from the problem at hand. In particular, the estimators depend on the chosen discrete space but not on the equations governing the problem. Moreover, other important properties are enjoyed, such as the computational effectiveness as well as the easiness of implementation.

Herein, we summarize the whole procedure we follow, for reader completeness. Starting from an assigned quadtree mesh, $\tilde{\mathcal{D}}_h^n$, at time t^n, we look for a *size function*, known as metric,[40] that specifies a new sizing for the mesh elements, so that a suitable norm of the discretization error remains below a user-defined tolerance, τ. In the original proposal by Zienkiewicz and Zhu, the chosen norm is the $H^1(\Omega)$-seminorm. In particular, we refer to the H^1-seminorm of the discretization error associated with the generic component z of the vector $\mathbf{q}$ in (8) at a given time t^n, that is, to

$$|e_h^{z,n}|_{H^1(\Omega)}^2 = \int_\Omega |\nabla z^n - \nabla z_h^n|^2 \, d\Omega, \tag{22}$$

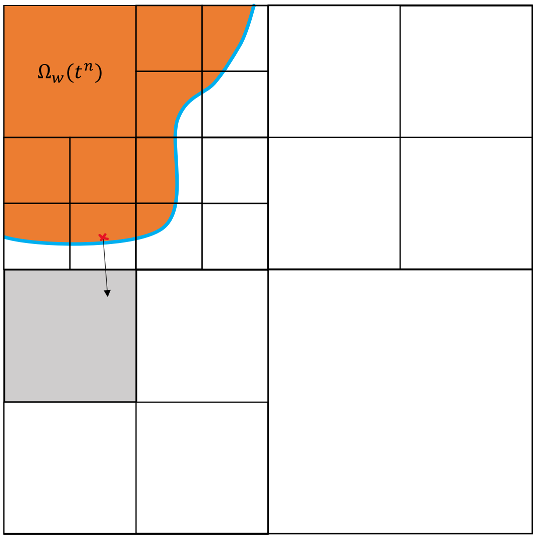

FIGURE 2 Tracking of the wetting–drying interface: wet domain at time t^n with associated interface curve highlighted in blue. The grey square corresponds to a cell marked for refinement since containing the centroid identified by the red cross when advected by the corresponding average field. [Colour figure can be viewed at wileyonlinelibrary.com]

where z_h denotes the discrete counterpart of z, while index n keeps trace of the selected time t^n. In References 11,12, and 39, the authors derive an error estimator for $|e_h^{z,n}|_{H^1(\Omega)}$ by replacing in (22) the exact gradient ∇z^n with a computable quantity, known as recovered gradient. In general, the recovered gradient is expected to provide an approximation to ∇z^n better than ∇z_h^n. Several methods are available in the literature to recover ∇z^n (see, e.g., References 41-43). In the sequel, we compute the recovered gradient $\nabla^R z_h^n$ by projecting, componentwise, the discrete gradient ∇z_h^n onto the bilinear space $\tilde{\mathbb{Q}}_1^h$, so that $\nabla^R z_h^n \in [\tilde{\mathbb{Q}}_1^h]^2$. For further details about the gradient recovery procedure we refer the interested reader to Section 10.3 in[16] and to.[24]

Thus, we can introduce the local recovery-based estimator for the discretization error, defined by

$$\eta_Q^2 = \int_Q |(\nabla^R z_h^n - \nabla z_h^n)|_Q|^2 \, dQ, \tag{23}$$

for each element $Q \in \tilde{\mathcal{D}}_h^n$, and where the integral is computed via a suitable quadrature formula (in the numerical assessment below, we employ the four-point Gaussian quadrature rule). The global error estimator is consequently assembled as

$$\eta^2 = \sum_{Q \in \tilde{\mathcal{D}}_h^n} \eta_Q^2. \tag{24}$$

Now, according to a metric-based approach,[16] we predict a new mesh size, h_Q^*, for each element Q of the grid $\tilde{\mathcal{D}}_h^n$ (which constitutes the so-called background mesh, that is, the grid where the computations are actually performed). To this aim, we impose that the global estimator η matches a user defined accuracy τ (i.e., $\eta = \tau$), in combination with an error equidistribution criterion, so that

$$\eta_Q^2 = \frac{\tau^2}{N}, \tag{25}$$

with N the cardinality of the background mesh $\tilde{\mathcal{D}}_h^n$. Then, the local error estimator is scaled with respect to an area information, by introducing the quantity $\tilde{\eta}_Q^2 = \eta_Q^2 / h_Q^2$. By exploiting the definition of the scaled estimator in (25), we can predict the new elementwise size,

$$h_Q^* = \frac{\tau}{\tilde{\eta}_Q \sqrt{N}}. \tag{26}$$

Notice that, since we deal with a quadtree mesh, we need to express the geometric information in (26) as a number of levels of refinement or coarsening. Actually, with each cell Q in $\tilde{\mathcal{D}}_h^n$, we have to associate an integer number, l_Q, which specifies the number of refinement (if $l_Q > 0$) or coarsening (if $l_Q < 0$) levels (where $l_Q = 0$ means that no action has to be taken on the cell Q). Under the hypothesis of invariance of $\tilde{\eta}_Q$ with respect to possible changes in the mesh size, so that

$$\tilde{\eta}_Q = \frac{\eta_Q}{h_Q} = \frac{\eta_Q^*}{h_Q^*},$$

with η_Q^* the error estimator computed on the mesh element identified by the predicted size h_Q^*, and since increasing the refinement level reduces to halve the characteristic cell size l_Q so that $2^{-l_Q} = h_Q^*/h_Q$, we derive that the level of refinement/coarsening coincides with

$$l_Q = \left\lceil \log_2\left(\frac{\eta_Q\sqrt{N}}{\tau}\right)\right\rceil. \tag{27}$$

In practice, we have introduced some controls on the metric-based mesh adaptation procedure to contain the associated computational effort. In particular, the user is demanded to set an upper bound, $N_{\max}$, on the maximum number of mesh elements jointly with a minimum space resolution, $\Delta_{\min}$, in the wet region. This allows us to avoid an over-refinement along the front. Moreover, we have set a minimum time, Δt_{ada}, between two consecutive mesh adaptation steps. Indeed, a mesh adaptation strategy combined with an interface tracking procedure at each iteration might be computationally too expensive and sometimes also useless from a modeling view point.

3.5 | Time adaptation

The possible heterogeneity in time characterizing the phenomena of interest justifies an adaptive choice of the time step. For a certain time t^{n-1}, the idea is to predict the next time step Δt^n, namely the next time t^n, by resorting to an a posteriori estimator for the discretization error in time. When predicting both the space and the time discretization, it is rather standard to keep distinct the space from the time error estimator in order to make the whole adaptation procedure more straightforward (see, e.g., References 44-47). Here, we adopt the same strategy. In particular, we refer the reader to Figure 3 where a sketch of the coupling between space and time mesh adaptation, together with the wetting–drying interface tracking procedure, is provided.

To drive the adaptive selection of the time step, we follow the approach proposed in Reference 13.

Consistently with Section 3.4, we refer to the generic component z of the vector $\mathbf{q}$ in (8). We denote by z_h the discrete counterpart of z, that we assume to be known at the times t^j, for $j \in \{0, \ldots, n-1\}$, with $\Delta t^j = t^j - t^{j-1}$ the jth time step in the temporal window $[0, T]$. To identify the next time step Δt^n, we again exploit a recovery-based a posteriori estimator

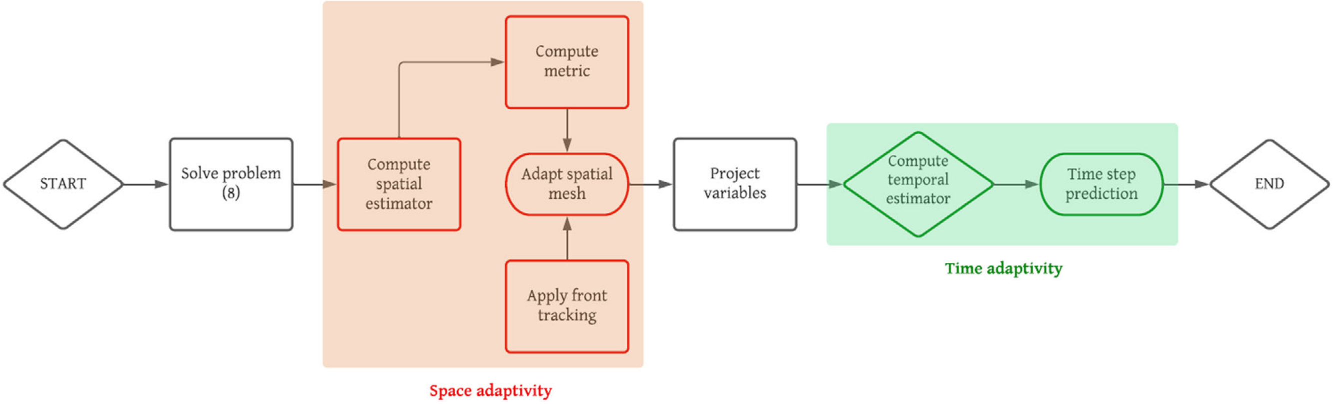

FIGURE 3 Block diagram of the whole space-time adaptive procedure. [Colour figure can be viewed at wileyonlinelibrary.com]

for the H^1-seminorm of the time discretization error,

$$|e^z_{h,t}(\mathbf{x})|^2_{H^1(I_{n-1})} = \int_{I_{n-1}} |\partial_t z - \partial_t z_h|^2 \, dt, \tag{28}$$

with $I_{n-1} = [t^{n-1}, t^n]$ and for $\mathbf{x} \in \Omega$. To make computable the right-hand side in (28), we approximate the derivative of the discrete solution by means of a standard finite difference scheme, so that

$$\partial_t z_h \simeq \frac{z_h^n - z_h^{n-1}}{\Delta t^{n-1}}, \tag{29}$$

d with $z_h^j = z_h(\mathbf{x}, t_j)$, for $\mathbf{x} \in \Omega$ and $j \in \{0, \ldots, n\}$. As far as the time derivative $\partial_t z$ is concerned, we replace the exact solution z with the recovered solution z^* coinciding with the quadratic interpolation of the pairs of data (t^{n-2}, z_h^{n-2}), (t^{n-1}, z_h^{n-1}), (t^n, z_h^n) (see, e.g., References 13 and 48). Thus, the square of the estimator in time associated with the interval I_{n-1} turns out to coincide with

$$\eta^2_{I_{n-1}}(\mathbf{x}) = \tilde{T} \int_{I_{n-1}} \left| \partial_t z^*(\mathbf{x}) - \frac{z_h^n - z_h^{n-1}}{\Delta t^{n-1}} \right|^2 dt, \tag{30}$$

for $\mathbf{x} \in \Omega$, the scaling factor $\tilde{T}$ being introduced to make $\eta_{I_{n-1}}$ dimensionally compatible with the space estimator in (23)-(24). Notice that, the integral in (30) can be computed exactly since the derivative $\partial_t z^*$ and the finite difference in (29) coincide with a linear and with a constant polynomial, respectively.

Now, in order to have a unique quantifier of the error on the time interval I_{n-1}, first we evaluate estimator $\eta_{I_{n-1}}$ at each vertex $\mathbf{v}$ of the current grid $\tilde{D}_h^{n-1}$. Successively, we compute an average value, $\eta_{I_{n-1,Q}}$, of the estimator on each element $Q \in \tilde{D}_h^{n-1}$, and, finally, we sum the elemental contributions throughout the mesh $\tilde{D}_h^{n-1}$. This allows us to define the time estimator for the interval I_{n-1} as

$$[\eta^T_{I_{n-1}}]^2 = \sum_{Q \in \tilde{D}_h^{n-1}} \eta^2_{I_{n-1,Q}} \quad \text{with} \quad \eta^2_{I_{n-1,Q}} = \frac{1}{4} \sum_{\mathbf{v} \in Q} \eta^2_{I_{n-1}}(\mathbf{v}). \tag{31}$$

Starting from (31), we are able predict the time step Δt_n. To this end, we rewrite estimator $\eta^T_{I_{n-1}}$ by applying a suitable scaling with respect to the temporal dimension Δt_{n-1}, being

$$[\eta^T_{I_{n-1}}]^2 = \tilde{T} \Delta t^2_{n-1} [\tilde{\eta}^T_{I_{n-1}}]^2, \tag{32}$$

with

$$[\tilde{\eta}^T_{I_{n-1}}]^2 = \frac{1}{\Delta t^2_{n-1}} \sum_{Q \in \tilde{D}_h^{n-1}} \frac{1}{4} \sum_{\mathbf{v} \in Q} \int_{I_{n-1}} \left| \partial_t z^*(\mathbf{v}) - \frac{z_h^n - z_h^{n-1}}{\Delta t^{n-1}} \right|^2 dt.$$

Then, we impose $\eta^T_{I_{n-1}} = \tau^{\Delta t}$, with $\tau^{\Delta t}$ a local tolerance, strictly dependent on the specific problem at hand. This choice, combined with relation (32), allows us to derive the new time length as

$$\Delta t_n = \frac{\tau^{\Delta t}}{\sqrt{\tilde{T}}\, \tilde{\eta}^T_{I_{n-1}}}. \tag{33}$$

We observe that time adaptation can be carried out only after the second time step. Indeed, the interpolation underling the definition of the recovered gradient in (30) involves the value of the discrete solution at three consecutive times. As a consequence, the first two time steps, Δt_0 and Δt_1, are freely set by the user. Moreover, to increase the effectiveness of the adaptive procedure, it is advisable to fix a minimum as well as a maximum value, $\Delta t_{\min}$ and $\Delta t_{\max}$, for the time step in order to lower and upper bound the time length predicted in (33). In the assessment below, we choose $\Delta t_{\max}$ starting from a numerical stability analysis based on the CFL condition, and we assign $\Delta t_0 = \Delta t_1 = \Delta t_{\max}$.

4 | NUMERICAL RESULTS

In this section we pursue a twofold goal. First, we explore the reliability of the proposed discretization framework on an ideal case study, taken from the literature.[7] Successively, we investigate the scalability performance of the implementation both on an ideal configuration and on a real case study.

We identify the computational domain Ω with the rectangle $(0, L) \times (0, H)$ m^2.

Concerning the tracking of the wetting/drying interface, we choose the threshold $h_{\min}$ equal to 10^{-5} m, so that no momentum flux takes place when the depth h is below such a value.

As far as the recovery-based error estimators is concerned, we choose the depth h to drive both the space and the time adaptation. In particular, we set the tolerance τ in (26) to 10^{-5} m, while, for the time adaptation, we choose the scaling factor in (30) as $\tilde{T} = \Delta t_{n-1}$, while setting the control steps $\Delta t_{\min}$ and $\Delta t_{\max}$ to 10^{-6} and to the value provided by the CFL condition, respectively.

4.1 | Reliability analysis: the Dam-break problem

We verify the performance of the numerical setting in Section 3 on a benchmark scenario, where Ω coincides with a square area, so that $L = H = 500$ m, in the presence of a flat and frictionless bed (i.e., $\mathbf{r}$ and $\mathbf{D}$ are null in (8)). These assumptions reduce system (8) to the classical shallow water equations.[49] The simple configuration here considered allows us to have an analytical solution taken as the target trend to be replicated.

We analyze two different setups, characterized by a fully wet and by a wetting/drying initial condition, respectively. In both cases, we apply nonreflecting boundary conditions on the whole boundary, $\partial\Omega$.

The grid resolution is set equal to $1/12$ m for the interface regions and to $1/6$ m for the wet areas, and a local tolerance $\tau^{\Delta t} = 100$ m for the time adaptivity. The spatial mesh adaptation procedure is performed every $\Delta t_{\text{ada}} = 10^{-3}$ s, with a maximum number of elements of the order of one million[1].

Let us focus on the former configuration. At the beginning, the domain is entirely covered by material at rest, with a material height equal to 70 m in the left half of the bed and to 7 m in the right part. The final time of interest coincides with $T = 2.2$ s.

Figure 4 compares the analytical with the numerical solution along the line $y = 250$ m, in terms of the elevation h and of the mass flux U_x, by highlighting a very good matching. We recognize a standard trend, that is, the generation of a shock along the downstream (i.e., on the right of the dam) direction, together with an upstream (i.e., on the left of the dam) rarefaction wave. We remark the effect of the flux limiter which allows us to avoid the generation, and consequently the propagation, of spurious oscillations in correspondence with the wetting/drying interface, thus guaranteeing the maximum discrete principle.

We consider now the latter setting. The material column, with an initial height equal to 70 m, is localized at the barycenter of the domain. The final time T is set to 0.8 s. The collapse of the material column, under the effect of the gravity, yields two rarefaction waves of equal intensity but symmetric with respect to the associated transverse axis, thus generating two nonlinear waves traveling along the same line but with an opposite direction.

In Figure 5, we provide the profile for the height and for the flux U_x at times $t = 0.3$ s and $t = 0.8$ s, both for the analytical and the approximate solution. The correspondence between the two solutions is remarkable. In this case, the bed is initially dry so the spatial adaptation, combined with the front-tracking technique, plays a crucial role to ensure a detailed preservation of the interface in the wetting/drying zone. Figure 6 (left and center panel) highlights this feature. We focus on the bottom-left quarter of the domain, that is, on the set $\{(x, y) : 0 \leq x, y \leq 250\,\text{m}\}$. The two plots show the material height distribution at two different times, superimposed to the quadtree adapted mesh. It is evident the sharp detection of the wetting/drying front provided by the spatial mesh adaptation as well as the correct tracking of the interface between the wet (in red) and the dry (in blue) zones. The panel on the right in the same figure displays the evolution of the time step yielded by the temporal adaptation process. In the very early stage of the simulation, the time step, initially set to 0.01s, drastically drops, in conjunction with the sudden collapse of the dam. Successively, the time step is, on average, reduced until time 0.15 s is reached. After this instant, Δt_n slightly increases until it reaches a constant value, about equal to 0.0018 s.

Finally, in Figure 7 we provide a barplot for the percentage relative error characterizing the mass conservation at different times. The interpolation step associated with the space adaptation does not introduce a significant mass loss, which remains always below 0.01%. This is a relevant feature since, a priori, the interpolation is not a conservative operation in

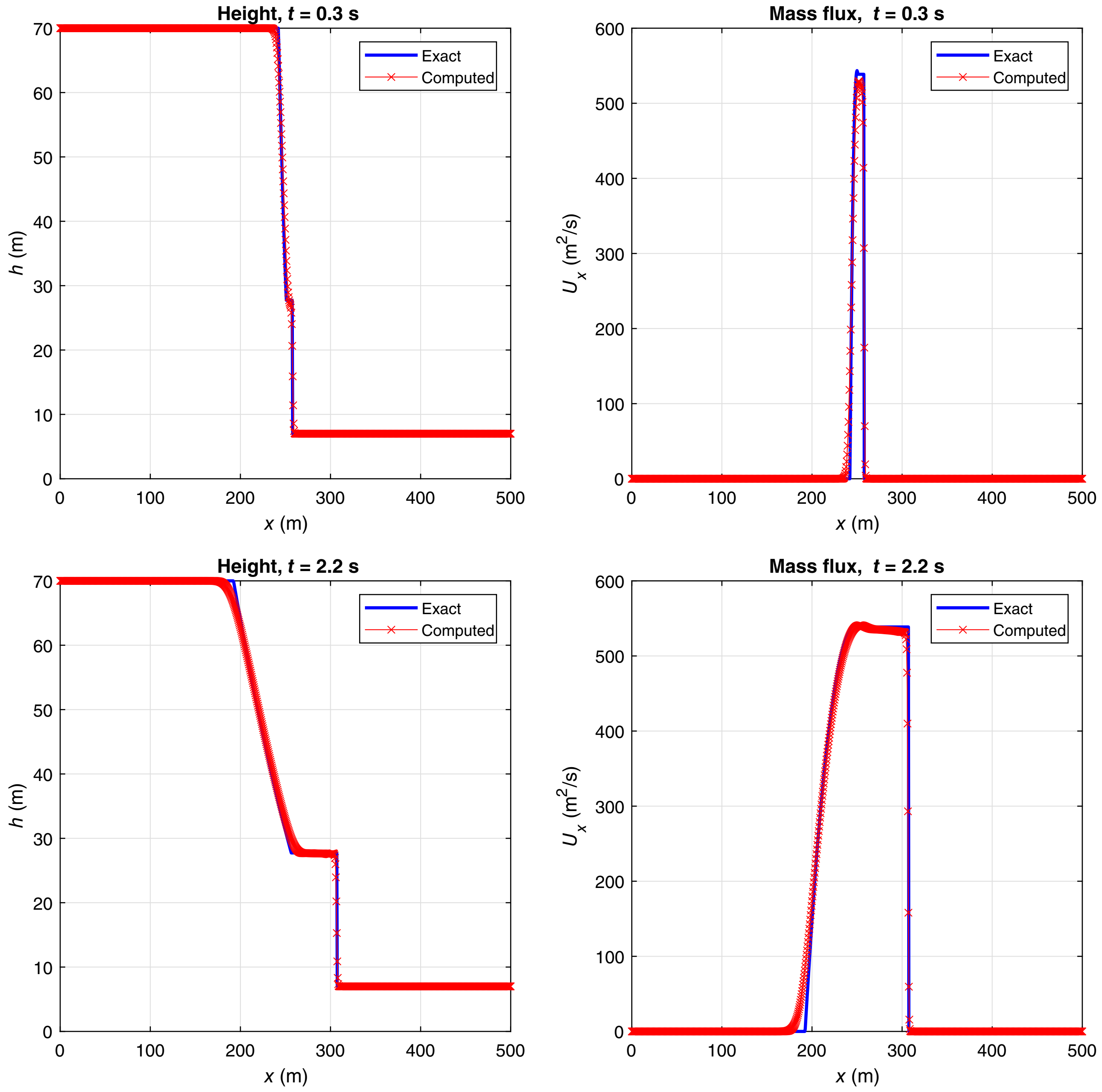

FIGURE 4 The dam-break problem (first scenario). Comparison between computed and analytical solution: height h (left) and mass flux U_x (right) at two different times (top-bottom). [Colour figure can be viewed at wileyonlinelibrary.com]

terms of mass. Note that we use the function *Integrate Variables* in Paraview[50] to compute the mass, also when evaluating the relative error in Figure 7, the reference quantity being represented by the mass at the initial time. The Paraview function introduces a discretization error of approximately 0.03% at the initial time, so that the higher error at the first time step is linked to the interpolation error of the initial data.

4.2 | Reliability analysis: Sensitivity to Δt_{ada}

A spatial mesh adaptation at each time step could be computationally too expensive and not necessarily useful from a modeling view point. Here, we carry out a sensitivity test to assess the influence of the temporal adaptation parameter

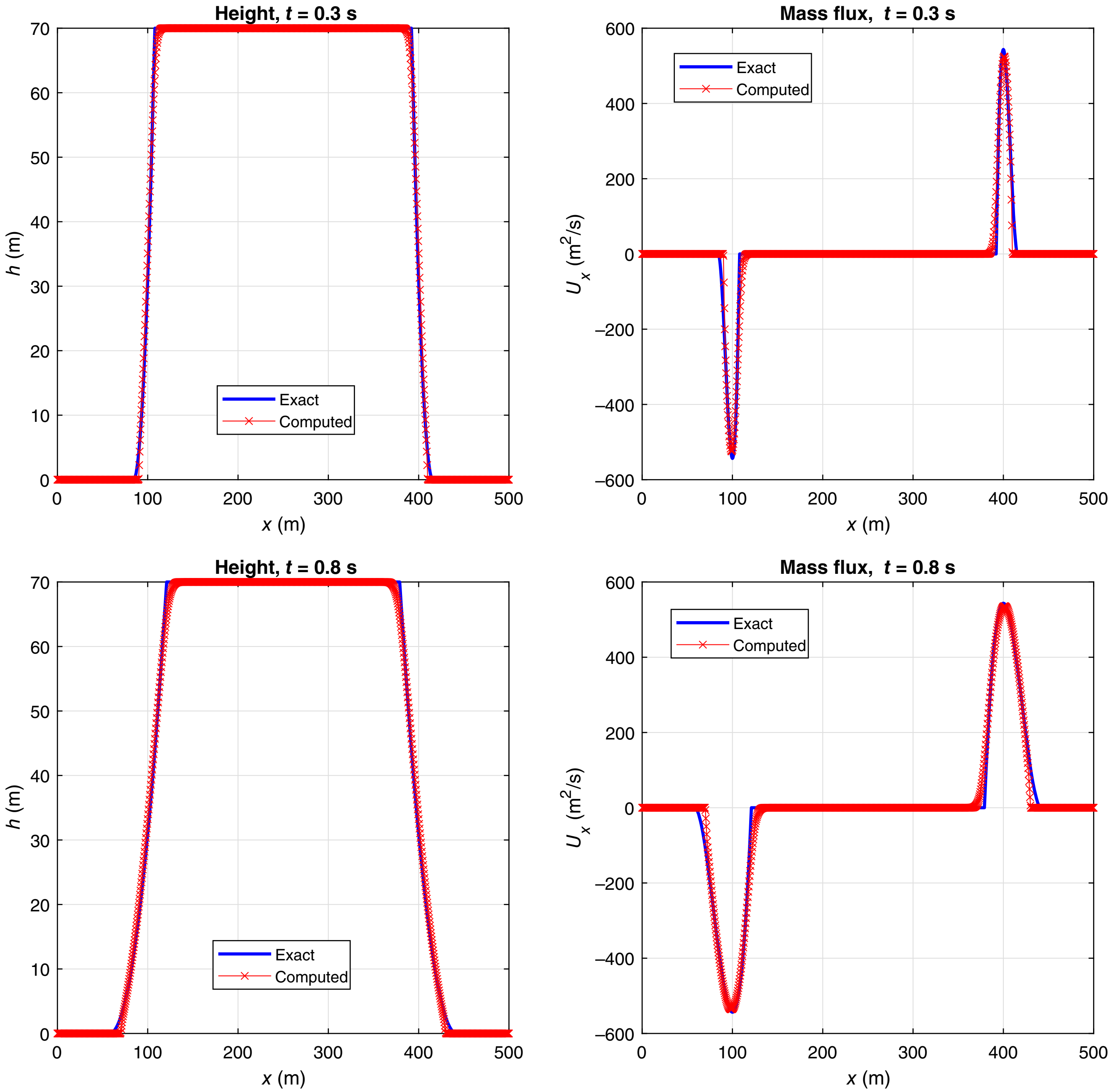

FIGURE 5 The dam-break problem (second scenario). Comparison between computed and analytical solution: height h (left) and mass flux U_x (right) at two different times (top-bottom). [Colour figure can be viewed at wileyonlinelibrary.com]

Δt_{ada} onto the numerical outcome. To this aim, we consider a flat square domain with dimensions $L = H = 5$ m, where we solve a wet-dry dam-break configuration initially at rest, characterized by a left and a right state equal to 2 and 0 m, respectively. The final time T is set to 0.1 s, while the tolerance on the time adaptation is chosen as $\tau^{\Delta t} = 10$ m. As far as the space adaptation is concerned, we set the maximum number of mesh elements equal to ten millions and a finest resolution equal to 0.625 and to 0.1562 m in wet and in the wet-dry region, respectively. We keep the CFL number equal to 0.9 for all the simulations.

We select a value for Δt_{ada} in the set

$$\{0,\ 0.0001,\ 0.0003,\ 0.0006,\ 0.0011,\ 0.0017,\ 0.0022,\ 0.0028,\ 0.0033,\ 0.0039,\ 0.0044,\ 0.0500\}.$$

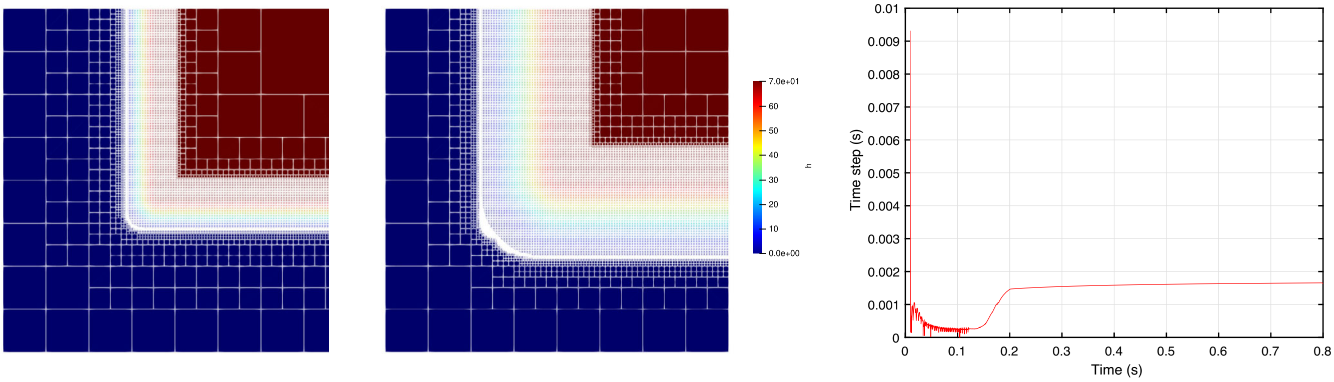

FIGURE 6 The dam-break problem (second scenario). Colormap of the material height h superimposed to the quadtree adapted mesh at $t = 0.3$ s (left) and $t = 0.8$ s (center); temporal evolution of the time step (right). [Colour figure can be viewed at wileyonlinelibrary.com]

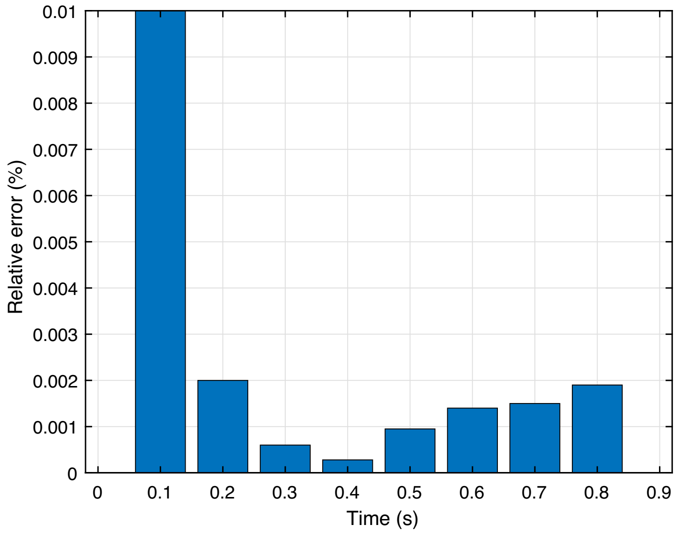

FIGURE 7 The dam-break problem (second scenario). Barplot of the percentage relative error associated with the mass conservation, at different times. [Colour figure can be viewed at wileyonlinelibrary.com]

Figure 8 gathers the results of such an analysis in terms of the number, $N_{\Delta t_{\text{ada}}/\Delta t}$, of the time steps performed before adapting the mesh (top-left), the CPU-time (top-right) and the accuracy (bottom), as a function of parameter Δt_{ada} in a logarithmic scale (so that $\Delta t_{\text{ada}} = 0$ s does coincide with $\Delta t_{\text{ada}} = 10^{-6}$ s). In particular, since the time step is adaptively selected, we provide the mean value of $N_{\Delta t_{\text{ada}}/\Delta t}$ over the time, while the accuracy is quantified with respect to the L^{∞}-norm associated with.

From the four panels, it is evident that it is possible to identify a region where the CPU-time and the accuracy strike a balance.

4.3 | Reliability analysis: accuracy test

We test now the rate of convergence of the adopted numerical scheme. To this aim, we consider an inviscid material initially at rest and characterized by the material height

$$h(r,0) = 1 + \frac{1}{10}\exp\left(-\frac{50}{L^2}r^2\right) \qquad \text{with } r^2 = \left(x - \frac{L}{2}\right)^2, \tag{34}$$

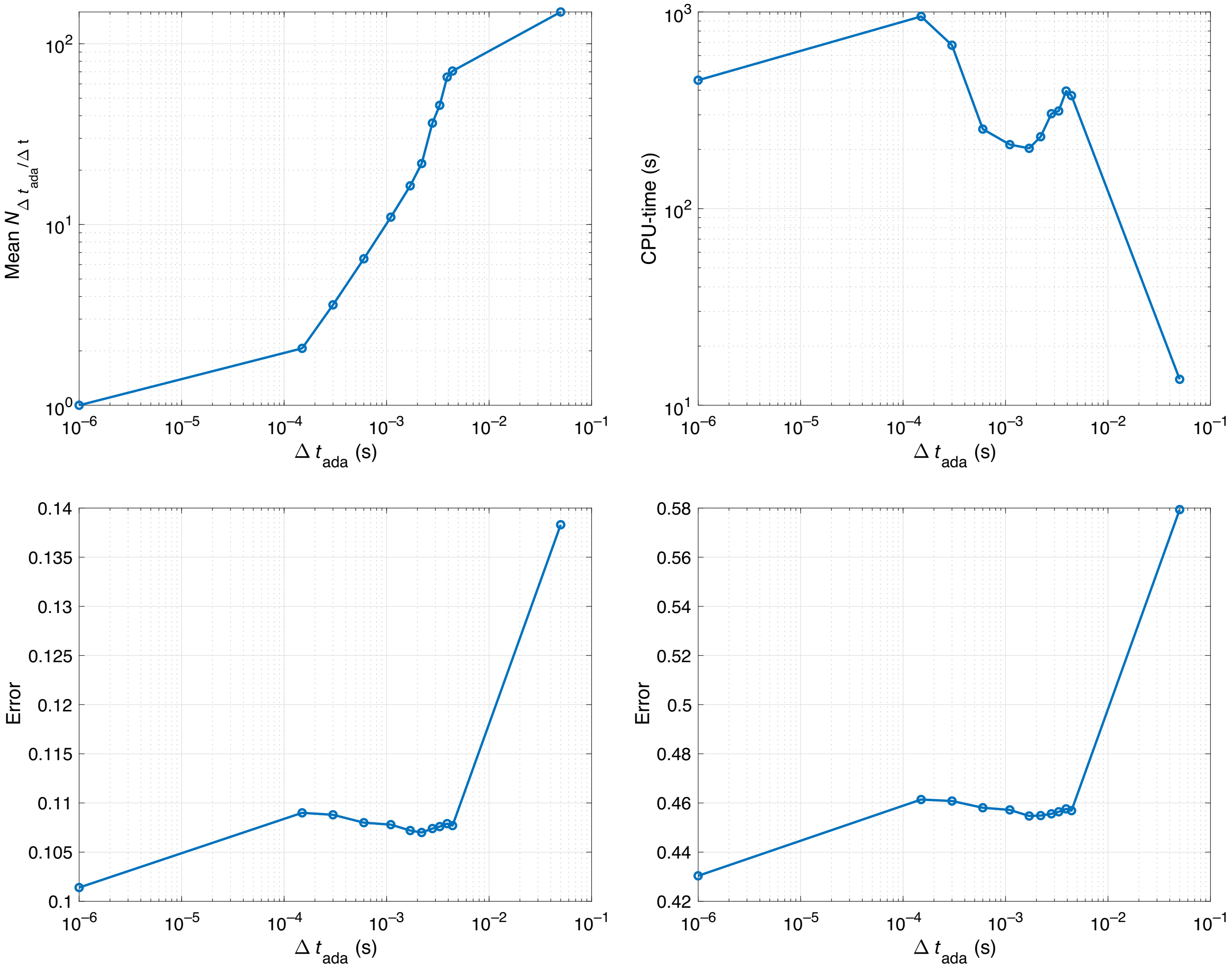

FIGURE 8 The sensitivity to Δt_{ada} test. Number of the time steps performed before adapting the mesh (top-left); CPU-time (top-right); L^∞-norm of the error (bottom) associated with the material height (left) and the mass flux (right). [Colour figure can be viewed at wileyonlinelibrary.com]

on a flat square domain with dimensions $L = H = 2$ m. The final time T is set to 0.1s, while the tolerance on the time adaptation is $\tau^{\Delta t} = 10$ m. We carry out the convergence analysis on a family of structured grids with uniform grid spacing $\Delta x = \{L/2^i\}_{i=6}^{10}$. We keep the CFL number equal to 0.8 for all the simulations that provides a diminishing maximum value of the time step for incrementing the space level of refinement.

Since the exact solution is not known explicitly, we adopt the solution computed on a structured mesh with grid spacing equal to $\Delta x = L/2^{11}$ as a reference (see Figure 9, top). Then, we compute the L^2-norm of the discretization error along the line $y = H/2$ m. In Figure 9, bottom we show the trend of the error associated with the mass flux (left panel) and with the material height (right panel). The estimated rate of convergence turns out to be equal to 1.72 and 1.50, respectively. This is in accordance with the expectation since the flux limiters might alter the second-order convergence by returning an intermediate (between one and two) rate.

4.4 | Efficiency analysis: scalability tests

Here, we assess the performance of the overall implementation by carrying out some intensive scalability tests, both on ideal and real scenarios. In particular, for the ideal tests, we save the whole numerical solution every 0.1 s, we switch on

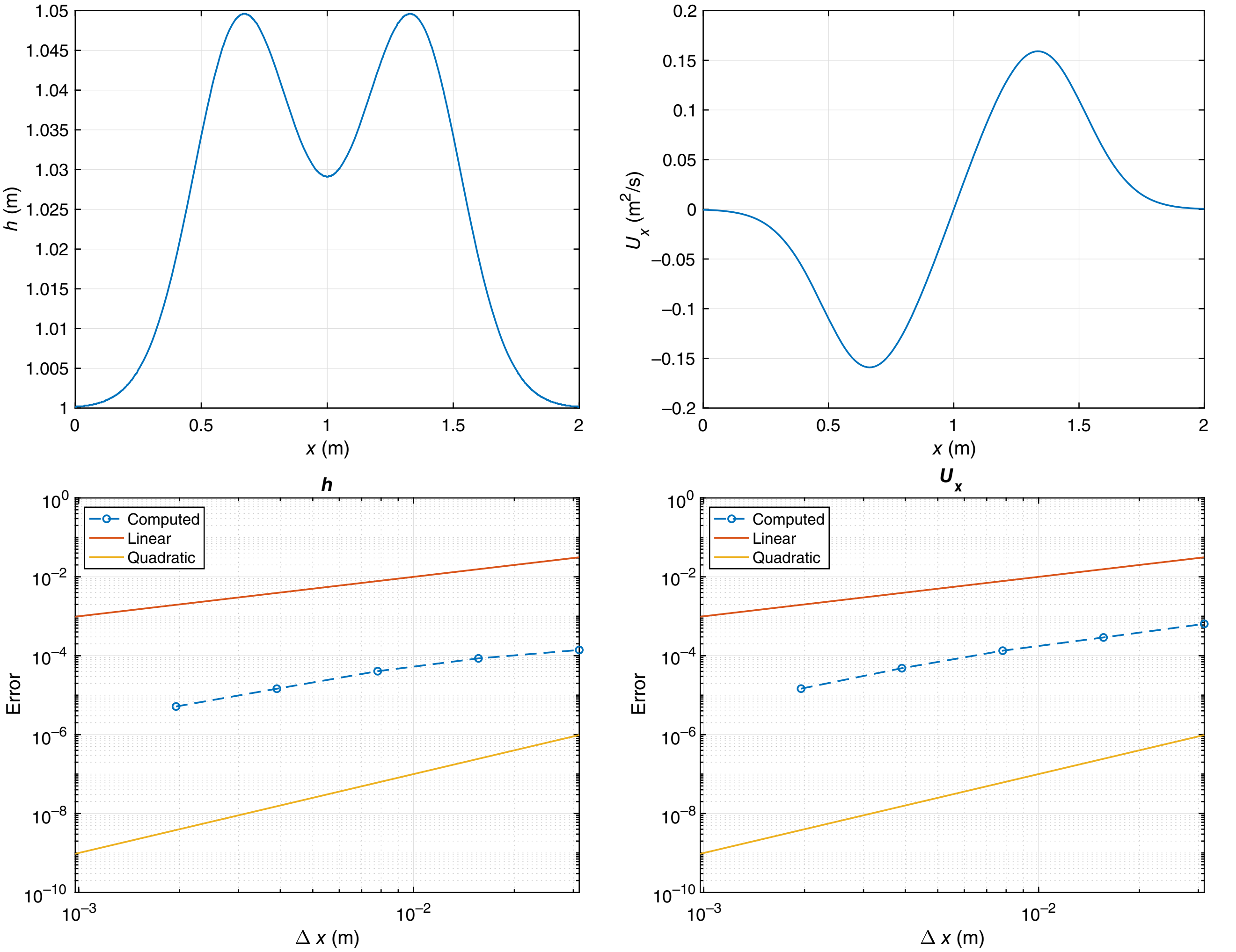

FIGURE 9 The accuracy test. Reference solution (top) along the line $y = H/2$ m: material height (left), x-component of the horizontal mass flux (right); convergence trend of the L^2-norm of the error (bottom) associated with the mass flux (left) and the material height (right). [Colour figure can be viewed at wileyonlinelibrary.com]

the spatial adaptation every $\Delta t_{\text{ada}} = 0.04$ s, while the time adaptation is always activated. Finally, the maximum number of quadtree elements is set equal to the order of 10 millions.

The simulations of this section have been run on the supercomputer CINECA GALILEO100 which is a new infrastructure co-funded by the European ICEI (Interactive Computing e-Infrastructure) project and engineered by DELL. This supercomputer is composed by 554 nodes, and each node has 384 GB RAM and two 24-cores processors Intel Cascade-Lake 8260 at 2.4 GHz. We have compiled and linked the application with the gcc-10 suite and OpenMPI 4.1.1. Finally, all the floating-point operations are performed in double precision.

4.4.1 | Ideal scenarios

Two settings are considered in this section. The first one coincides with the radial dam break problem. In such a case, we deal with the standard shallow water equations solved on a flat bottom, without any source term. We set the final simulation time T to 1 s. The computational domain is a square so that $L = H = 5$ m, while the initial height h is chosen as

$$h(r,0) = \begin{cases} 2 & \text{if } r \le 0.5 \\ 1 & \text{if } r > 0.5 \end{cases} \quad \text{with } r = \sqrt{(x - L/2)^2 + (y - L/2)^2}. \tag{35}$$

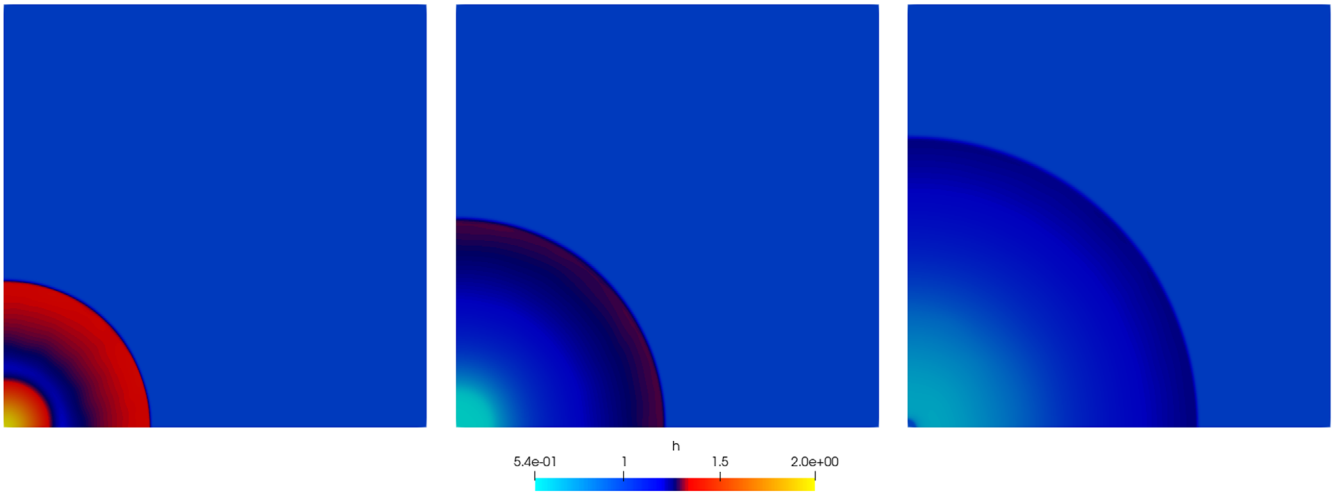

FIGURE 10 The radial dam break. Colormap of the material height h at $t = 0.3$ s (left), $t = 0.6$ s (center) and $t = 1$ s (right). [Colour figure can be viewed at wileyonlinelibrary.com]

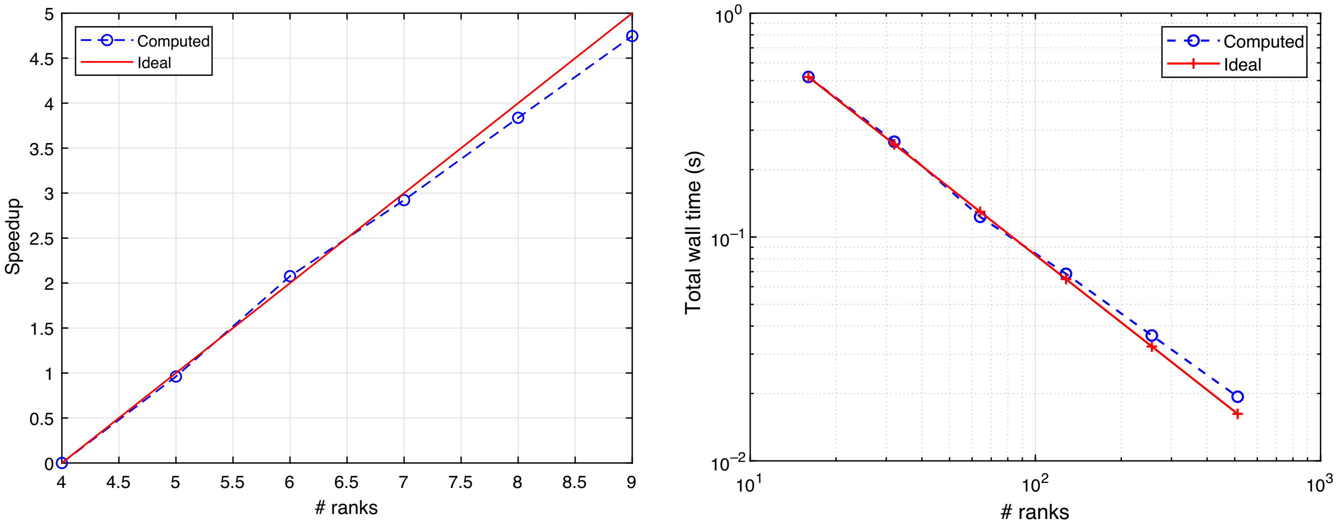

FIGURE 11 The radial dam break. Scalability test: speed-up in `log2` scale (left) and total wall time in `log-log` scale (right) as a function of the number of ranks. [Colour figure can be viewed at wileyonlinelibrary.com]

The material is initially at rest. We consider a gravitational field $g = 1\ \text{m/s}^2$ and we assign nonreflecting boundary conditions on the whole boundary domain. Figure 10 shows the material height distribution in the top-right quarter of the domain, $\{(x, y) : 2.5 \leq x, y \leq 5\text{m}\}$, at three different times after the dam collapsing, when employing a minimum spatial resolution equal to $0.125 \cdot 10^{-2}$ m and a local tolerance $\tau^{\Delta t} = 1$ m to constrain the time adaptivity.

We perform a scalability test according to the algorithm described in References 15, by using a different number of cores, or likewise of domain subdivisions, from 16 to 512. Figure 11 gathers the plot of the speed-up (on the left) and of the total simulation wall time (on the right) in seconds, as a function of the ranks. In both the plots, the values are scaled with respect to the solution associated with 16 ranks. It is evident that the numerical method proposed in Section 3 scales rather efficiently, up to the maximum number of cores we use. The total wall time employed with 16 ranks turns out to be approximately 27 times the one demanded by 512 ranks, resulting in a parallel efficiency roughly equal to 80%, for a total number of mesh elements equal to $6.79 \cdot 10^6$ at the final time.

As a second configuration, we consider a mass sliding along an inclined plane with bed friction, the gravitational field being now selected as $g = 9.81\ \text{m/s}^2$, and for a final simulation time T equal to 3 s. The sliding mass, characterized by a

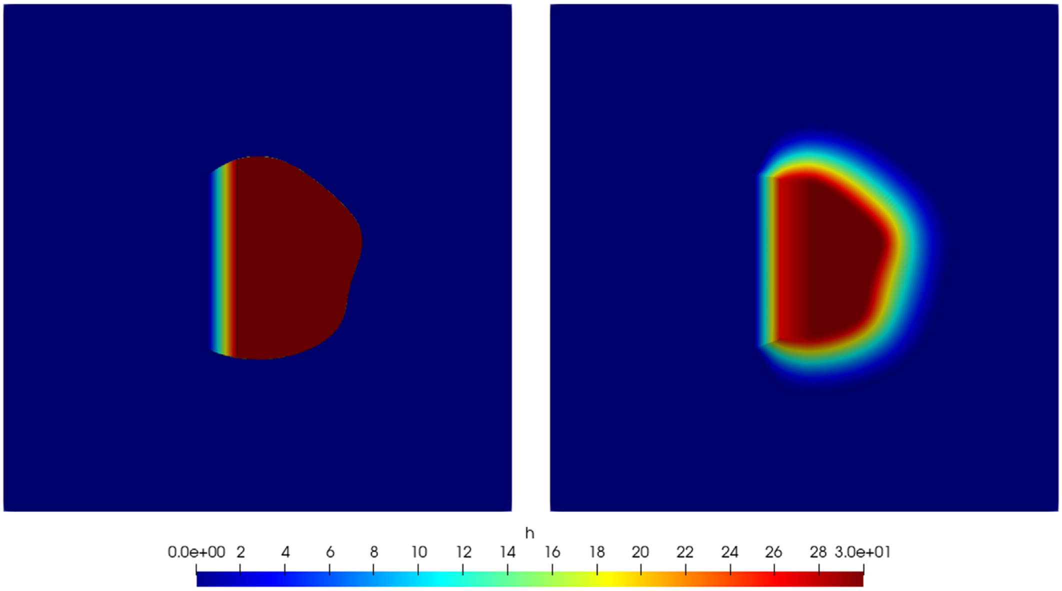

FIGURE 12 The mass sliding along an inclined plane. Distribution of the still level at times $t = 0$ s (left) and $t = 3$ s (right). [Colour figure can be viewed at wileyonlinelibrary.com]

density $\rho = 1400\,\text{kg/m}^3$, is initially placed at the center of the domain $\Omega = (0, 1000)^2\,\text{m}^2$, with a height equal to

$$h(\mathbf{x}, 0) = \begin{cases} \max\{0, \min\{500x/L - 200, 30\}\} & \text{for } \mathbf{x} \in W \\ 0 & \text{otherwise,} \end{cases} \tag{36}$$

with $L = 1000$ m, and where

$$W = \left\{ \mathbf{x} = (x, y) \in \mathbb{R}^2 : \frac{(x - L/2)^2}{L^2} + \frac{(x - L/2)^2}{L^2} \leq [0.2 + 0.01 \sin(10\,(y - L/2)\frac{\pi}{L^2})\,]^2 \right\}.$$

The material is initially at rest, while the plane has a slope of approximately 26° with respect to the horizontal axis, that is $Z(\mathbf{x}) = 500 - 500x$. The simulation is carried out in the absence of surface pressure and with a turbulence coefficient $\xi = 10^8\,\text{m/s}^2$ and a bed friction angle $\delta = 23°$. Nonreflecting boundary conditions are imposed on the whole boundary domain. The spatial resolution is set equal to 0.1667 m in the interface region, while we set a minimum spatial resolution equal to 0.25 m in the wet areas. In Figure 12 we show the distribution of the still level, that is, the sum of the material height h with the orographic surface Z, at the initial and at the final time. The contourlines are very sharply detected thanks to the combined action between mesh adaptation and wetting–drying interface tracking.

Now, we perform a scalability analysis by varying the number of ranks from 16 to 512. Figure 13 gathers the results of such an investigation. In the left panel, we show the speed-up, scaled with respect to the solution obtained when using 16 ranks, as a function of the number of processors. The right panel displays the trend of the total wall time after applying the same scaling, and for an increasing number of ranks. The two plots confirm the efficiency characterizing the proposed numerical setting, up to the tested number of cores. In this case, the total wall time characterizing the run with 16 ranks turns out to be approximately 22 times the wall time required by the simulation based on 512 processors, roughly resulting in a parallel efficiency equal to 70%, for a total number of mesh elements equal to $3.40 \cdot 10^6$ at the final time.

4.4.2 | A real scenario: the Bindo-Cortenova landslide

In this section we consider a real setting represented by a $1.2\,\text{km}^2$ wide translational landslide in Bindo-Cortenova, a small village located in the northern of Italy, close to Lecco. The sliding material is formed by very large conglomeratic rock blocks, up to 100 m in size, immersed in a gravelly sand matrix. The toe of the slope underwent a catastrophic failure in

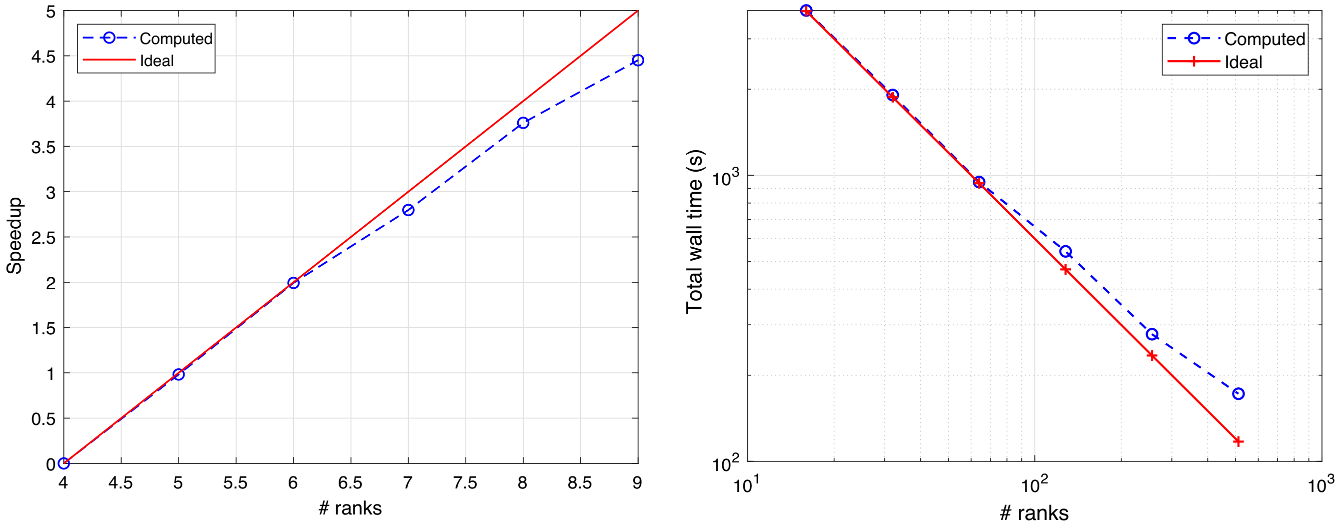

FIGURE 13 The mass sliding along an inclined plane. Scalability test: speed-up in `log2` scale (left) and total wall time in `log-log` scale (right) as a function of the number of ranks. [Colour figure can be viewed at wileyonlinelibrary.com]

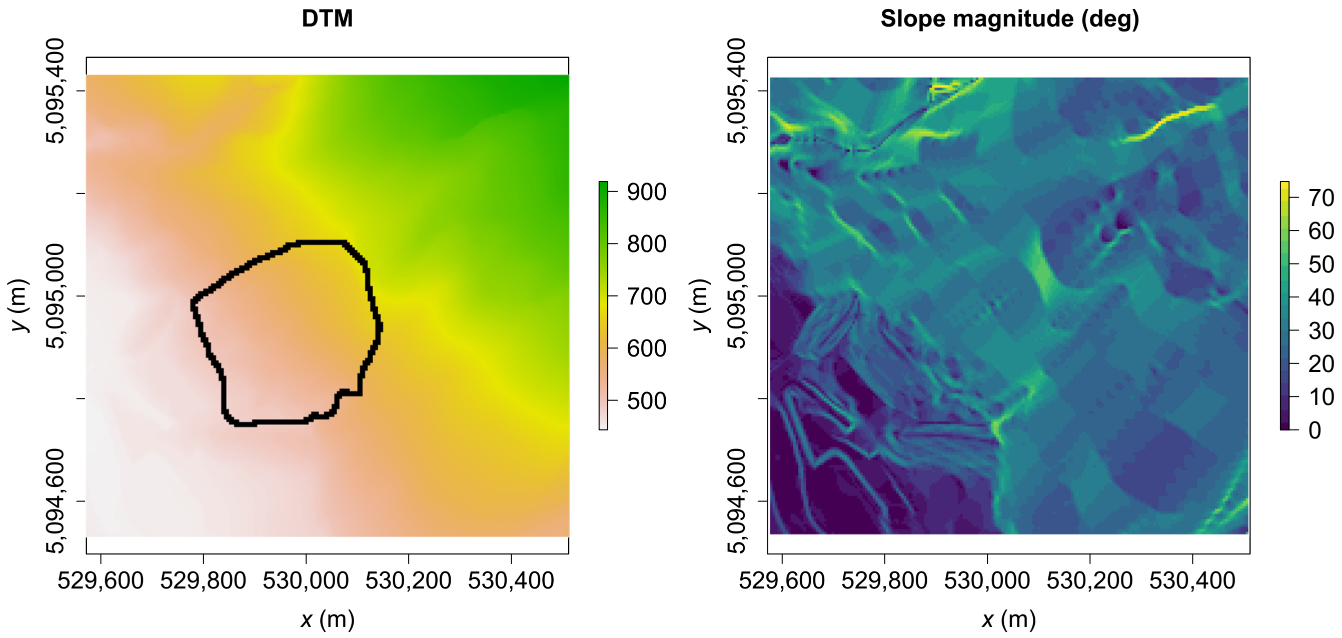

FIGURE 14 The Bindo-Cortenova landslide. Colormap of the input orography (left) and of the slope magnitude (right). [Colour figure can be viewed at wileyonlinelibrary.com]

December 2002 (with a material volume about equal to $1.2 \cdot 10^6$ m^3), after a period of extremely heavy rainfall. For more details about the considered scenario, we refer the interested reader, for instance, to Reference 51.

The input DTM, that is, the input orography Z, coincides with a 5-m-resolution raster. Figure 14 shows the DTM (left), where the contour of the detached material is black-highlighted, together with the corresponding slope map (in degrees), that is, the distribution of the arctan $|\nabla Z|$ (right). According to Reference 51, the average thickness of the sliding material is 38 m, and the mean slope is approximately 28°. We consider a bed friction angle equal to the residual friction angle, that is, 33.9°, while the material density ρ is set to 1291 kg/m^3, and, for simplicity, we consider a null surface pressure. Note that this assumption only removes the offset in the bed friction, in case of a nonnull surface pressure.

Concerning the rheological model, since we are interested in simulating a debris flow, we pick the fluid viscosity μ and the yield shear stress τ_Y equal to 50 Pa· s and $2 \cdot 10^3$ Pa, respectively. The turbulence coefficient ξ in the Voellmy

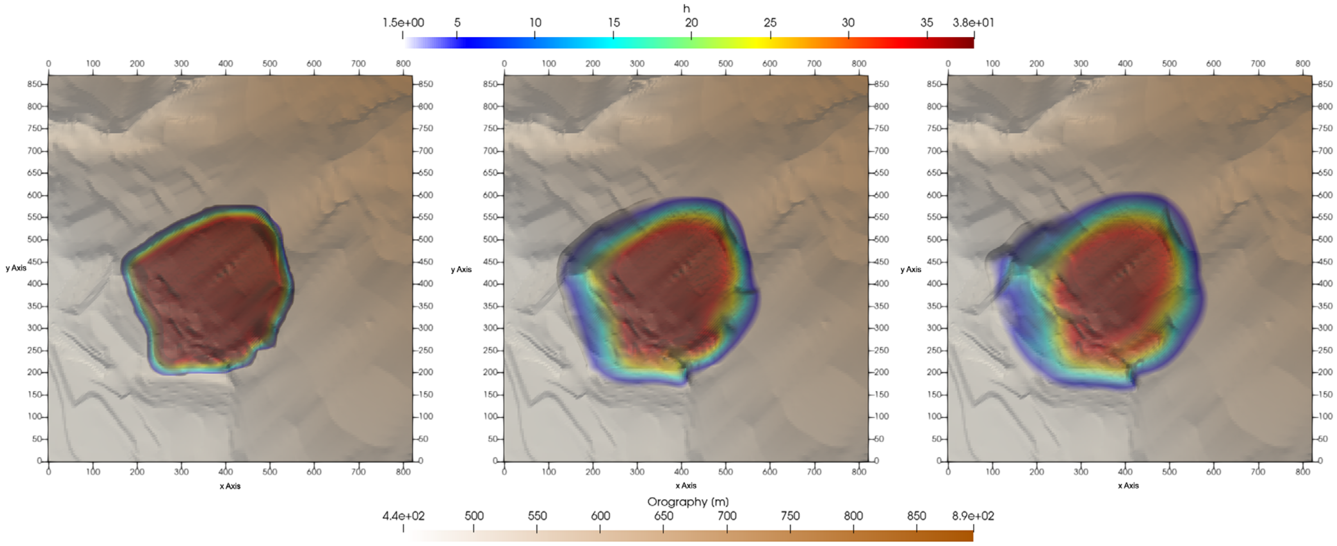

FIGURE 15 The Bindo-Cortenova landslide. Distribution of the material height at times $t = 1$ s (left), $t = 5$ s (center) and $t = 10$ s (right). [Colour figure can be viewed at wileyonlinelibrary.com]

rheological model (2) is chosen sufficiently large in order to prevent the time step numerically go to zero. In particular, via a trial-and-error approach, we set $\xi = 10^8$ m/s^2. We observe that the presence of the source term, together with the diffusion contribution, plays an important role in the numerical approximation, in terms of stability of the scheme. A possible remedy to this issue, although beyond the specific goal of the paper, consists in resorting to a second order implicit treatment of the source term. According to Reference 52, this approach does not affect the efficiency of the overall implementation.

To contain the memory effort, we adopt a saving time equal to 0.1 s, we activate the spatial adaptation every 0.04 s, while keeping the time adaptation always switched on, using a local time tolerance $\tau^{\Delta t} = 10$ m. The maximum number of quadtree mesh elements is of the order of 10 millions, as for the ideal simulations in the previous section. The final time is $T = 10$ s, while the computational domain coincides with a rectangle 820×870 m^2 (we refer to Figure 14 to identify the exact location of the area with respect to the DTM data). We employ a spatial resolution equal to 0.1667 m for the cells in the wetting–drying interface region, while we set a minimum resolution equal to 0.25 m for the wet regions. Finally, nonreflecting boundary conditions complete the problem along the boundary.

Figure 15 shows the material height distribution at three different times, by highlighting the slipping of the material all around the initial location of the landslide, with a nonhomogeneous advancement of the wetting–drying front which follows the complex local orography.

Figure 16 gathers some information about the space-time adaptation procedure. The left panel shows the spatial adapted mesh, superimposed to the material height distribution, at the final time $T = 10$ s. The landslide front is sharply identified in the computational mesh, with a thin refinement along the contour of the sliding material. The panel on the right displays the evolution of the time step during the whole simulation. The temporal discretization step rapidly reduces due to the sudden change in the dynamics of the water column, before assuming almost a constant trend (Δt_n stagnates around $0.25 \cdot 10^{-3}$) when the landslide gradually extends. Analogously to the dam-break case study, the time step reaches a constant value, after an initial phase when it gradually reduces (compare the right panels in Figures 6 and 16, respectively). This can be ascribed to the comparable phenomenological trend characterizing the two scenarios.

Finally, in Figure 17 we show the results of the scalability analysis, when varying the number of processors from 16 to 512, analogously to what done for the ideal simulation tests. The numerical method in Section 3 scales rather efficiently also in this more challenging configuration, at least up to the number of cores we tested. In more detail, we obtain a speed-up equal to 12 for the case of 512 cores with respect to the 16 core configuration. Concerning the wall time, from the right panel in the figure we derive that the total time associated with 16 ranks turns out to be approximately 23 times the time demanded when using 512 ranks, resulting in a parallel efficiency of roughly 70%, for a total number of mesh elements equal to $5.97 \cdot 10^6$ at the final time. All the values in the two plots of Figure 17 are scaled with respect to the solution yielded by the run based on 16 processors, analogously to Figures 11 and 13.

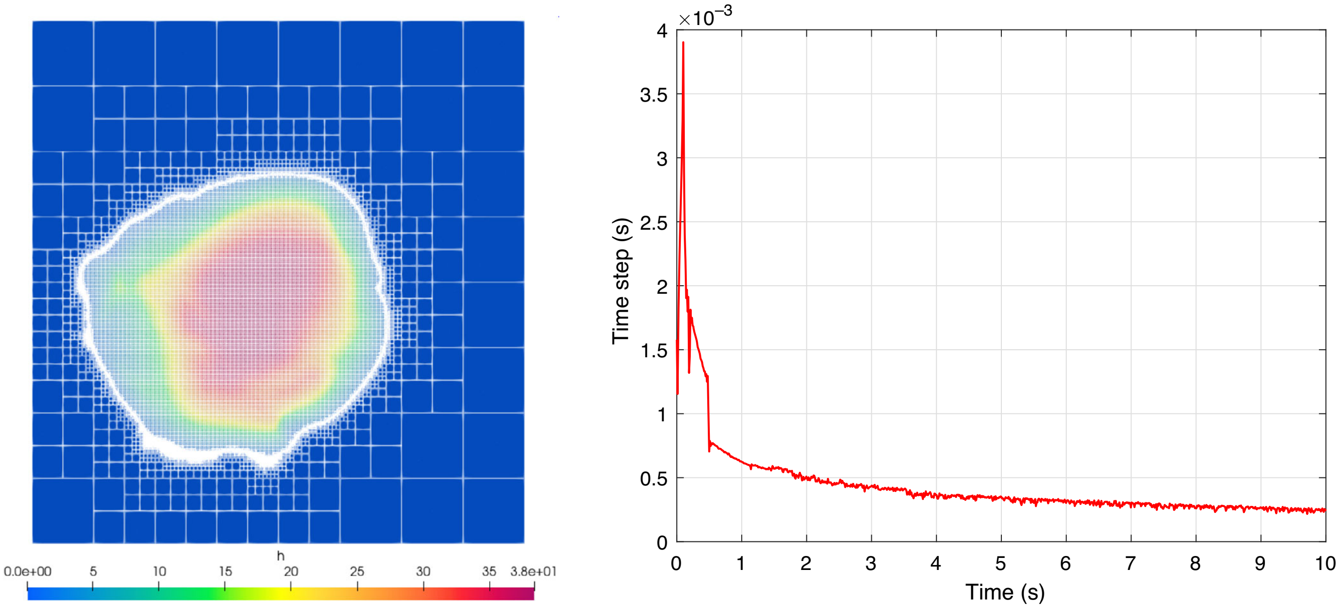

FIGURE 16 The Bindo-Cortenova landslide. Adapted spatial mesh at the final time (left); temporal evolution of the time step (right). [Colour figure can be viewed at wileyonlinelibrary.com]

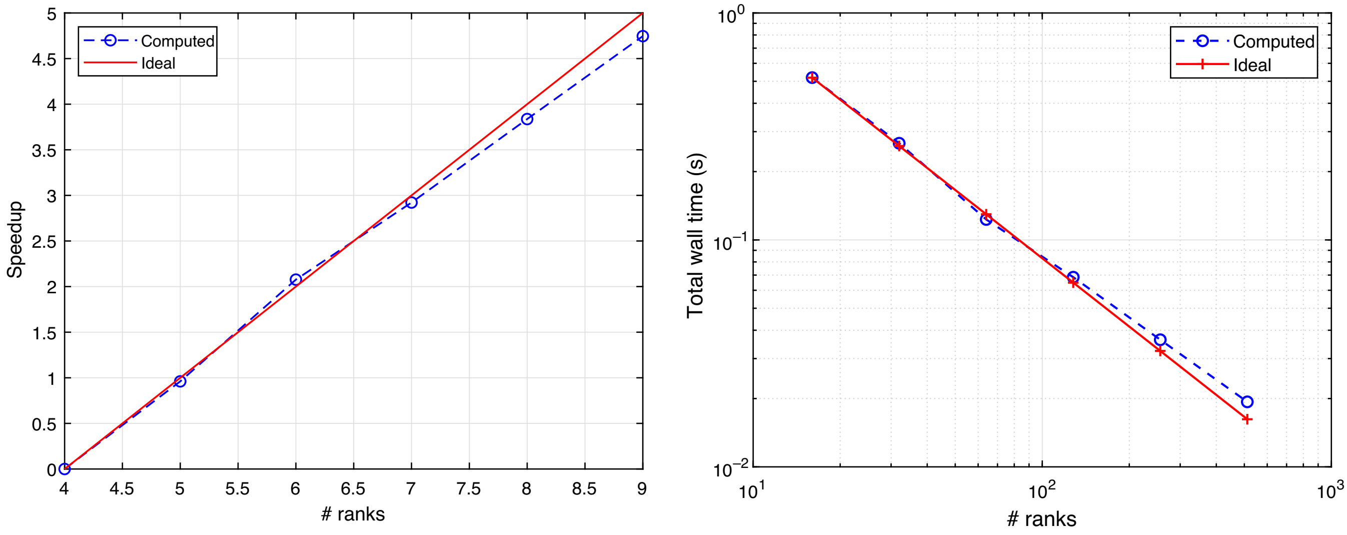

FIGURE 17 The Bindo-Cortenova landslide. Scalability test: speed-up in `log2` scale (left) and total wall time in `log-log` scale (right) as a function of the number of ranks. [Colour figure can be viewed at wileyonlinelibrary.com]

5 | CONCLUSIONS

We have proposed a scalable multiprocessor depth-integrated finite element quadtree-based numerical method to efficiently solve advection-dominated conservation laws, in particular for the simulation of fast landslides. We have numerically verified the reliability of the proposed numerical method, together with the effectiveness of a parallel implementation of the approach. A suitable combination of a space-time metric-based adaptation procedure with a tracking interface strategy guarantees a sharp and efficient modeling both for ideal configurations and real test-case scenarios, with a parallel efficiency ranging between 70% and 80%.

Among the possible future developments of this work, we mention the enrichment of the current simulation framework in order to include the activation/intermittent phase of the landslide dynamic, that is, the long-term time-scale dynamics. In such a context, the landslide dynamics is governed by the pore-pressure diffusion, and the landslide material behaves roughly like a rigid body. Another improvement is represented by the implicit treatment of the source term in order to avoid too restrictive time steps to ensure the stability of the numerical approach (see Reference 52) thus enabling to perform uncertainty quantification, for example, by polynomial chaos expansion (see Reference 53).

ACKNOWLEDGMENT

We thank Dott. Monica Corti for providing part of the data for the simulation of the real case study. Open Access Funding provided by Politecnico di Milano within the CRUI-CARE Agreement.

CONFLICT OF INTEREST

The authors declare no potential conflict of interests.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ENDNOTE

[1] The simulations in this section have been run with four ranks on a laptop with an Intel i7 CPU, 2.60 GHz clock frequency, 16GB RAM.

ORCID

Federico Gatti https://orcid.org/0000-0002-2318-4039
Marco Fois https://orcid.org/0000-0002-4470-112X

REFERENCES

1. Haque U, Blum P, Da Silva PF, et al. Fatal landslides in Europe. *Landslides*. 2016;13(6):1545-1554.
2. Haque U, Da Silva PF, Devoli G, et al. The human cost of global warming: deadly landslides and their triggers (1995–2014). *Sci Total Environ*. 2019;682:673-684.
3. Froude MJ, Petley DN. Global fatal landslide occurrence from 2004 to 2016. *Nat Hazards Earth Syst Sci*. 2018;18(8):2161-2181.
4. Dilley M. *Natural Disaster Hotspots: A Global Risk Analysis*. Vol 5. World Bank Publications; 2005.
5. Sai BVKS, Zienkiewicz OC, Manzari MT, Lyra PRM, Morgan K. General purpose versus special algorithms for high-speed flows with shocks. *Int J Numer Methods Fluids*. 1998;27(1–4):57-80. doi:10.1002/(SICI)1097-0363(199801)27:1/4<57::AID-FLD650>3.0.CO;2-C
6. Quecedo M, Pastor M. A reappraisal of Taylor–Galerkin algorithm for drying–wetting areas in shallow water computations. *Int J Numer Methods Fluids*. 2002;38(6):515-531. doi:10.1002/fld.225
7. Quecedo M, Pastor M, Herreros MI, Fernández Merodo JA. Numerical modelling of the propagation of fast landslides using the finite element method. *Int J Numer Methods Eng*. 2004;59(6):755-794. doi:10.1002/nme.841
8. Donea J. A Taylor–Galerkin method for convective transport problems. *Int J Numer Methods Eng*. 1984;20(1):101-119.
9. Löhner R, Morgan K, Zienkiewicz OC. The solution of non-linear hyperbolic equation systems by the finite element method. *Int J Numer Methods Fluids*. 1984;4(11):1043-1063.
10. Peraire J. *A Finite Element Method for Convection Dominated Flows*. PhD thesis. University College of Swansea; 1986.
11. Zienkiewicz OC, Zhu JZ. A simple error estimator and adaptive procedure for practical engineering analysis. *Int J Numer Methods Eng*. 1987;24(2):337-357.
12. Zienkiewicz OC, Zhu J. The superconvergent patch recovery and a posteriory error estimates II. Error estimates and adaptivity. *Int J Numer Methods Eng*. 1992;33(7):1365-1382.
13. Porta GM, Perotto S, Ballio F. A space-time adaptation scheme for unsteady shallow water problems. *Math Comput Simul*. 2012;82(12):2929-2950. doi:10.1016/j.matcom.2011.06.004
14. Samet H. The quadtree and related hierarchical data structures. *ACM Comput Surv*. 1984;16(2):187-260.
15. Burstedde C, Wilcox LC, Ghattas O. p4est: scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM J Sci Comput*. 2011;33(3):1103-1133.
16. Africa PC. *Scalable Adaptive Simulation of Organic Thin-Film Transistors*. PhD thesis. Politecnico di Milano; 2019.
17. Isaac T, Burstedde C, Wilcox LC, Ghattas O. Recursive algorithms for distributed forests of octrees. *SIAM J Sci Comput*. 2015;37(5):C497-C531. doi:10.1137/140970963
18. Burstedde C. Parallel tree algorithms for AMR and non-standard data access. *ACM Trans Math Softw*. 2020;46(32):1-31. doi:10.1145/3401990
19. bitpit/PABLO; 2018. https://optimad.github.io/bitpit/
20. Bergmann M, Fondanèche A, Iollo A. An Eulerian finite-volume approach of fluid-structure interaction problems on quadtree meshes. *J Comput Phys*. 2022;471:111647. doi:10.1016/j.jcp.2022.111647
21. p4est; 2011. Accessed December 16, 2022. https://p4est.github.io/
22. Basilisk; 2015. http://basilisk.fr/

23. Popinet S. A quadtree-adaptive multigrid solver for the serre–green–Naghdi equations. *J Comput Phys*. 2015;302:336-358.
24. Africa PC, Perotto S, de Falco C. Scalable recovery-based adaptation on quadtree meshes for advection-diffusion-reaction problems; 2022. doi:10.48550/ARXIV.2212.05945
25. Bernardi MS, Africa PC, de Falco C, Formaggia L, Menafoglio A, Vantini S. On the use of interferometric synthetic aperture radar data for monitoring and forecasting natural hazards. *Math Geosci*. 2021;53(8):1781-1812.
26. Evans S, Hungr O, Clague J. Dynamics of the 1984 rock avalanche and associated distal debris flow on mount Cayley, British Columbia, Canada; implications for the landslides hazard assessment on dissected volcanos. *Eng Geol*. 2001;61:29-51.
27. Hungr O. A model for the runout analysis of rapid flow slides, debris flows and avalanches. *Can Geotech J*. 1995;86:233-238.
28. Laigle D, Coussot P. Numerical modeling of mudflows. *J Hydraul Eng*. 1997;123(7):617-623.
29. Hutchinson J, Bhandari R. Undrained loading, a fundamental mechanism of mudflows and other mass movements. *Geotechnique*. 1971;21(4):353-358.
30. Pastor M, Quecedo M, Fernández Merodo J, Herrores M, Gonzalez E, Mira P. Modelling tailings dams and mine waste dumps failures. *Geotechnique*. 2002;52(8):579-591.
31. Boffi D, Brezzi F, Fortin M. *Mixed Finite Element Methods and Applications*. Springer; 2013.
32. Zienkiewicz OC, Morgan K, Morgan K. *Finite Elements and Approximation*. Courier Corporation; 2006.
33. Zalesak ST. Fully multidimensional flux-corrected transport algorithms for fluids. *J Comput Phys*. 1979;31(3):335-362.
34. Boris JP, Book DL. Flux-corrected transport. III. Minimal-error FCT algorithms. *J Comput Phys*. 1976;20(4):397-431.
35. Kuzmin D, Möller M, Turek S. High-resolution FEM–FCT schemes for multidimensional conservation laws. *Comput Methods Appl Mech Eng*. 2004;193(45–47):4915-4946.
36. Rusanov V. On difference schemes of third order accuracy for nonlinear hyperbolic systems. *J Comput Phys*. 1970;5(3):507-516. doi:10.1016/0021-9991(70)90077-X
37. Löhner R, Morgan K, Peraire J, Vahdati M. Finite element flux-corrected transport (FEM–FCT) for the euler and Navier–Stokes equations. *Int J Numer Methods Fluids*. 1987;7(10):1093-1109.
38. Laurmaa V, Picasso M, Steiner G. An octree-based adaptive semi-Lagrangian VOF approach for simulating the displacement of free surfaces. *Comput Fluids*. 2016;131:190-204.
39. Zienkiewicz OC, Zhu J. The superconvergent patch recovery and a posteriory error estimates I. the recovery technique. *Int J Numer Methods Eng*. 1992;33(7):1331-1364.
40. Frey PJ, George PL. *Mesh Generation. Application to Finite Elements*. 2nd ed. John Wiley & Sons, Inc.; 2008.
41. Li X, Wiberg N. A posteriori error estimate by element patch post-processing, adaptive analysis in energy and L2 norms. *Comput Struct*. 1994;53(4):907-919.
42. Maisano G, Micheletti S, Perotto S, Bottasso C. On some new recovery-based a posteriori error estimators. *Comput Methods Appl Mech Eng*. 2006;195(37–40):4794-4815.
43. Rodriguez R. Some remarks on Zienkiewicz-Zhu estimator. *Numer Methods Partial Differ Equ*. 1994;10(5):626-635.
44. Verfurth R. A posteriori error estimate for finite element discretizations of the heat equation. *CAL*. 2003;40:195-212.
45. Cascon J, Ferragut L, Asensio M. Space-time adaptive algorithm for the mixed parabolic problem. *Numer Math*. 2006;103:367-392.
46. Meidner D, Vexler B. Adaptive space-time finite element methods for parabolic optimization problems. *SIAM J Control Optim*. 2007;40(1):116-142.
47. Schmich M, Vexler B. Adaptivity with dynamic meshes for space-time finite element discretizations of parabolic equations. *SIAM J Control Optim*. 2008;30(1):369-393.
48. Lozinski A, Picasso M, Prachittham V. An anisotropic error estimator for the crank–nicolson method: application to a parabolic problem. *SIAM J Sci Comput*. 2009;31(4):2757-2783. doi:10.1137/080715135
49. Tan WY. *Shallow Water Hydrodynamics: Mathematical Theory and Numerical Solution for a Two-Dimensional System of Shallow-Water Equations*. Elsevier; 1992.
50. Ahrens J, Geveci B, Law C. Paraview: an end-user tool for large data visualization. *The Visualization Handbook*. 2005;836:717-731.
51. Secondi MM, Crosta G, di Prisco C, Frigerio G, Frattini P, Agliardi F. Landslide motion forecasting by a dynamic visco-plastic model. In: Margottini C, Canuti P, Sassa K, eds. *Landslide Science and Practice*. Springer; 2013:151-159.
52. Zhong X. Additive semi-implicit Runge–Kutta methods for computing high-speed nonequilibrium reactive flows. *J Comput Phys*. 1996;128(1):19-31. doi:10.1006/jcph.1996.0193
53. Crestaux T, Le Maıtre O, Martinez JM. Polynomial chaos expansion for sensitivity analysis. *Reliab Eng Syst Safe*. 2009;94(7):1161-1172.

How to cite this article: Gatti F, Fois M, de Falco C, Perotto S, Formaggia L. Parallel simulations for fast-moving landslides: Space-time mesh adaptation and sharp tracking of the wetting front. *Int J Numer Meth Fluids*. 2023;95(8):1286-1309. doi: 10.1002/fld.5186