MOX-Report No. 15/2025

Efficient particle generation for depth-averaged and fully 3D MPM using TIFF image data

Fois, M.; de Falco, C.; Formaggia L.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it https://mox.polimi.it

Efficient particle generation for depth-averaged and fully 3D MPM using TIFF image data

Marco Fois [(1)] Carlo de Falco [(1)] Luca Formaggia [(1)]

March 14, 2025

[(1)] MOX – Modelling and Scientific Computing
Dipartimento di Matematica, Politecnico di Milano
Piazza Leonardo da Vinci, 20133 Milano, Italy
marco.fois@polimi.it
carlo.defalco@polimi.it
luca.formaggia@polimi.it

Keywords: MPM, Depth-Averaged MPM, Particle generation, TIFF images.

Abstract

In this work, we present a comprehensive framework for the generation and efficient management of particles in both fully three-dimensional (3D) and depth-averaged Material Point Method (DAMPM) simulations. Our approach leverages TIFF image data to construct initial conditions for large-scale geophysical flows, with a primary focus on landslide modeling. We describe the algorithms developed for particle initialization, distribution, and tracking, ensuring consistency and computational efficiency across different MPM formulations. The proposed methods enable accurate representation of complex topographies while maintaining numerical stability and adaptability to diverse material behaviors. Although the primary application is landslide simulation, the methodologies outlined are broadly applicable to other fields involving granular flows, fluid-structure interactions, and large-deformation processes. Performance evaluations demonstrate the efficiency and robustness of our approach, highlighting its potential for advancing high-fidelity simulations in geomechanics and beyond.

1 Introduction

The Material Point Method (MPM) is a widely used computational technique, developed in the late '90s, to solve problems involving large deformations, multiphase interactions, and complex material behaviors [1, 2]. In MPM, a continuous material body is discretized into a collection of N_p material points (particles), each of which carries all relevant physical properties, such as mass, velocity, stress, and strain. These properties depend on the specific problem being analyzed and evolve over time according to governing physical laws [3, 4, 5]. This particle-based representation allows MPM to effectively capture history-dependent behaviors, such as plasticity and damage accumulation, making it particularly suitable for problems involving extreme deformations and complex material responses [6, 7, 8, 4, 9, 10].

To facilitate computations, a background grid with N_v nodes is introduced over the computational domain Ω. The grid can be either Cartesian or non-Cartesian and is used for solving the governing equations by computing differential terms directly on the grid nodes. Unlike the conventional mesh-based methods, the MPM grid does not store any historical information about the material but serves as a computational tool to facilitate numerical operations, after which particle properties are updated, and the grid is reset at each time step [11].

One of the main advantages of MPM lies in the relatively straightforward manner in which input data, often derived from digital images such as CT scans, TIFF, or PNG files, can be transformed into initial data suitable for spatial discretization. Specifically, these input data are used to generate the particles, which serve as the discrete representation of the continuous domain [12, 13, 14, 15, 10]. This feature makes MPM particularly appealing in applications where input data originate from imaging techniques, enabling efficient conversion of geometric and material properties into numerical representations.

A classical approach in the literature for initializing the computational domain consists of mapping each pixel in the input image to a material point. The material properties are typically inferred from pixel attributes, such as color or intensity values [11, 16, 17, 18]. This method has been employed in various contexts, including biomedical simulations and geotechnical applications. However, a major challenge associated with this approach concerns the resolution of the input image and, consequently, the number of material points generated for discretization.

In the specific context of geophysical modeling, the initial topographic and geotechnical conditions are often derived from DTMs generated through satellite interferometry [19, 20, 21, 22, 23, 24, 25]. The DTMs, typically provided in geoTIFF format, represent the terrain as raster data, where each pixel is associated with an absolute elevation value. The spatial resolution of such models generally does not exceed 5 meters per pixel.

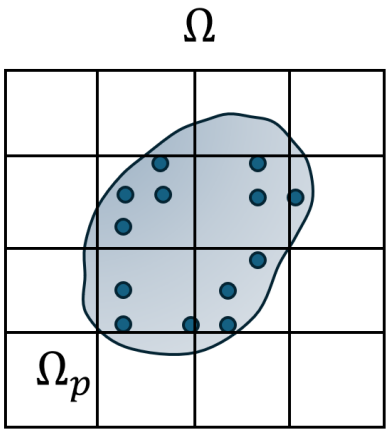
When generating a particle system from a TIFF image, it is crucial to ensure that the number of particles is sufficient for accurate material tracking. Insufficient particle density could compromise the precision of the simulation, leading to an inaccurate representation of material flow or distribution over time. Therefore, it is advisable to adopt interpolation or refinement strategies to ensure an optimal distribution of particles within the simulation space.

In this work we present an efficient and robust algorithm for particle generation in the MPM framework for both the depth-averaged and fully 3D cases. In particular, the paper is organized as follows: in Section 2, we briefly introduce the classical formulation of the MPM and describe the particle generation algorithm for the depth-averaged case. Section 3 focuses on the extension to the fully 3D case, where

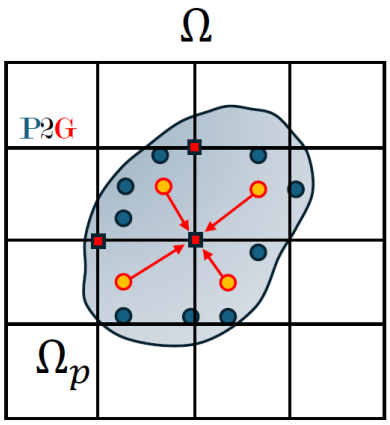
we propose a conversion algorithm between the Depth-Averaged MPM (DAMPM) and the fully three-dimensional formulation, ensuring numerical conservation of mass and momentum. Section 4 presents numerical validation tests, demonstrating the accuracy and efficiency of the proposed approach. Finally, in Section 5, we draw conclusions and discuss potential directions for future works.

2 The particle generation in DAMPM

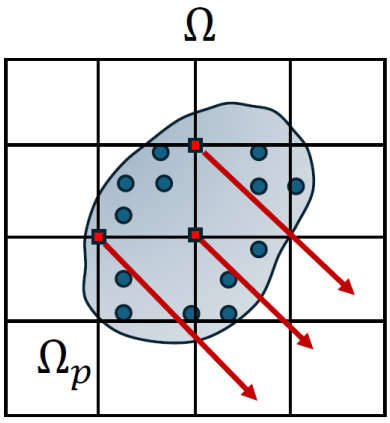
The classical MPM is a numerical approach used to simulate the behavior of continua by discretizing the material into a set of N_p material points. These points serve as carriers of all relevant physical quantities, such as mass, velocity, and stress. The material is embedded in a computational domain Ω, where a background grid of N_v nodes is defined. This grid is used for computations involving the differential terms of the problem, facilitating the numerical solution, see Figure 1a.



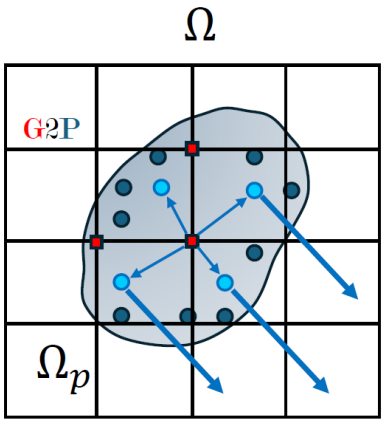
(a) A grid is defined over the domain Ω, while the continuum material is discretized into a set of N_p Lagrangian particles, each carrying specific physical properties.



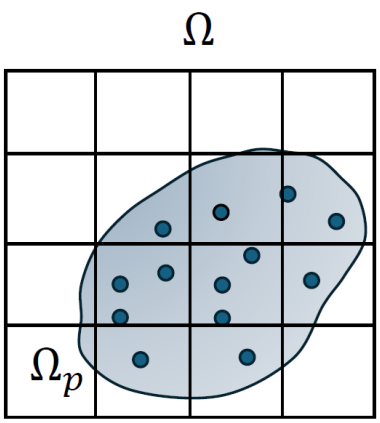
(b) Particle-to-Grid (P2G) procedure: masses and momenta are accumulated onto the corresponding grid nodes using the finite element basis functions. This process allows for assembling the nodal forces.



(c) Advective Phase on grid nodes: nodal accelerations and velocities, illustrated with red arrows, are computed using the nodal momenta and the total nodal forces.



(d) The G2P procedure. The advective phase is projected back to the particles.



(e) Final stage of the method: once the particle stresses are calculated, the particle positions and their depths are updated. This completes one iteration of the scheme, allowing the method to restart.

Figure 1: Illustration of the classic MPM algorithm.

.

MPM is a particle-based, semi-Lagrangian method that consists of four main steps. The first step, known as Particle-to-Grid (P2G), involves transferring mass and momentum from the material points to the grid nodes using, for instance, the finite element shape functions defined at the nodes (Figure 1b). This step also enables the computation of internal and external forces acting on the material. The

second step is the advection phase (Figure 1c), where nodal velocities and accelerations are computed using the accumulated masses, forces, and momenta. The third step, Grid-to-Particle (G2P), updates the positions, velocities, and stresses of the material points based on the grid data (Figure 1d). Finally, the last phase advances the simulation in time, resetting the grid in preparation for the next iteration of the cycle (Figure 1e).

In this Section, we focus on the generation of particles starting from a DTM in TIFF format within the context of the depth-averaged formulation of the MPM, in which the state variables such as pressure and velocities are averaged in the vertical direction. In particular, each material point can be assimilated to a vertical column with a proper height h_p and volume V_p. An extensive description of the DAMPM method can be found in [7, 26, 27].

In a computational framework where a domain Ω is represented by a DTM in TIFF format, the simplest approach to defining a computational grid is to align each pixel of the TIFF image with a grid node. This results in a structured mesh composed of square elements.

The generated grid is inherently defined in a 2D space, where the nodes $\mathbf{x}_i$, for $i \in \{1, \ldots, N_v\}$, correspond directly to the pixels of the TIFF image. The corresponding elevation values $Z(\mathbf{x}_i)$ represent the topographic height at each node $\mathbf{x}_i$, as shown in Figure 2.

However, in the context of depth-averaged modeling, these elevation values do not play an active role in the computational process except through their gradients ∇Z, which contribute to the representation of terrain-induced effects, such as driving and resisting forces in the governing physical model.

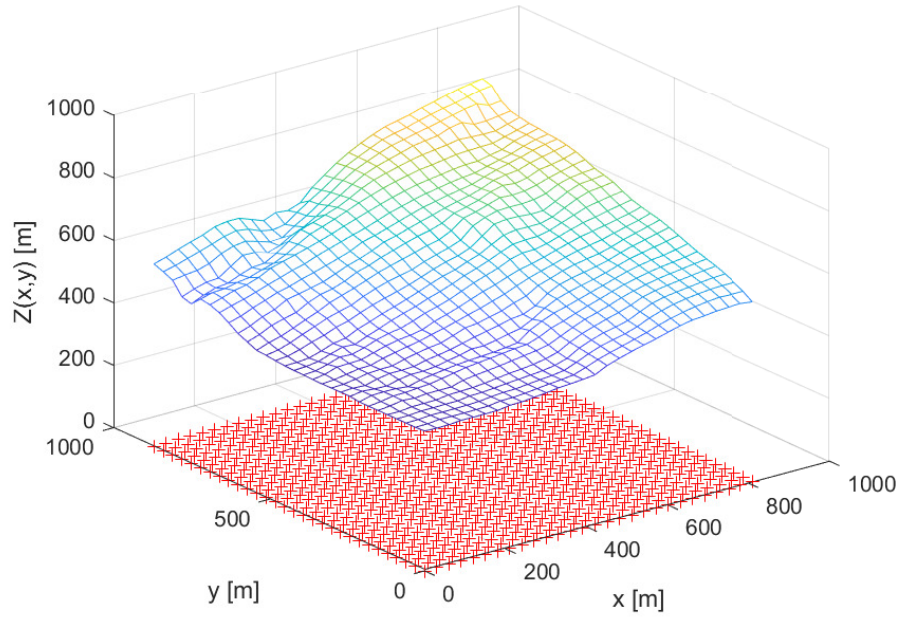


Figure 2: The computational grid is shown with red crosses overlaid on the DTM, which represents the domain Ω. Each grid node $\mathbf{x}_i$ of the grid is associated with an elevation value $Z(\mathbf{x}_i)$ representing the terrain height.

Regarding the initial condition of the material of interest, it must be discretized using a finite set of particles $p \in \{1, \ldots, N_p\}$, each characterized by physical properties such as mass, velocity, volume, and stresses. An intuitive and straightforward approach, given that the material positions, elevations, and initial perimeter are known, is to align the particle positions with the pixels of the TIFF image, thereby coinciding with the grid nodes $\mathbf{x}_i$, as shown in Figure 3.

A critical issue related to the use of this technique is related to the resolution of the available DTM raster, which, as mentioned in Section 1, typically does not exceed 5 m per pixel when using satellite-derived data. Constraining the model so rigidly to the DTM resolution can lead to numerical instability and poor discretization of the continuous material, significantly affecting the accuracy of the simulation.

Another important issue arising from this approach is related to P2G transfer phase within the MPM framework. Specifically, in a standard grid configuration, each grid node receives mass contributions from

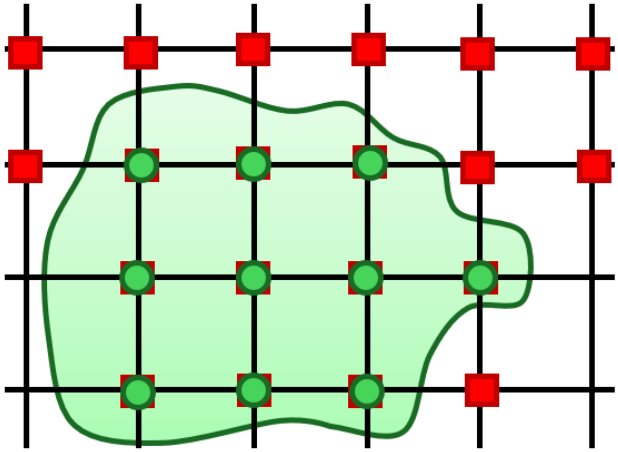


Figure 3: The continuous material is embedded within the Cartesian grid generated from the DTM of the area of interest. The continuous material is discretized into a set of particles (green circles) positioned at the same locations as the grid nodes (red squares), each associated with its own elevation value.

a limited number of particles [28, 9, 29, 30, 31, 32] and when this happens, information loss and numerical instabilities, such as cell-crossing artifacts, may occur. These considerations are especially crucial in geomechanical applications, such as landslide modeling, where accurately capturing the evolution of material displacement and stress distribution is fundamental to reliable predictions.

To mitigate these challenges, we propose a simple technique that enables the generation of the initial condition on the continuum material with an arbitrary number of particles, starting by the TIFF image.

Let us consider a DTM in TIFF format, as shown in left panel of Figure 4, where the topography Z and an initial condition Ψ on the material are provided. The process begins by selecting the computational domain Ω from the DTM and the simplest way to get it is to extract a square pixel region from the TIFF data. Then, the procedure consists in defining a two-dimensional square grid over Ω, with nodes x_i corresponding to the pixels of the DTM, as shown in the right panel of Figure 4.

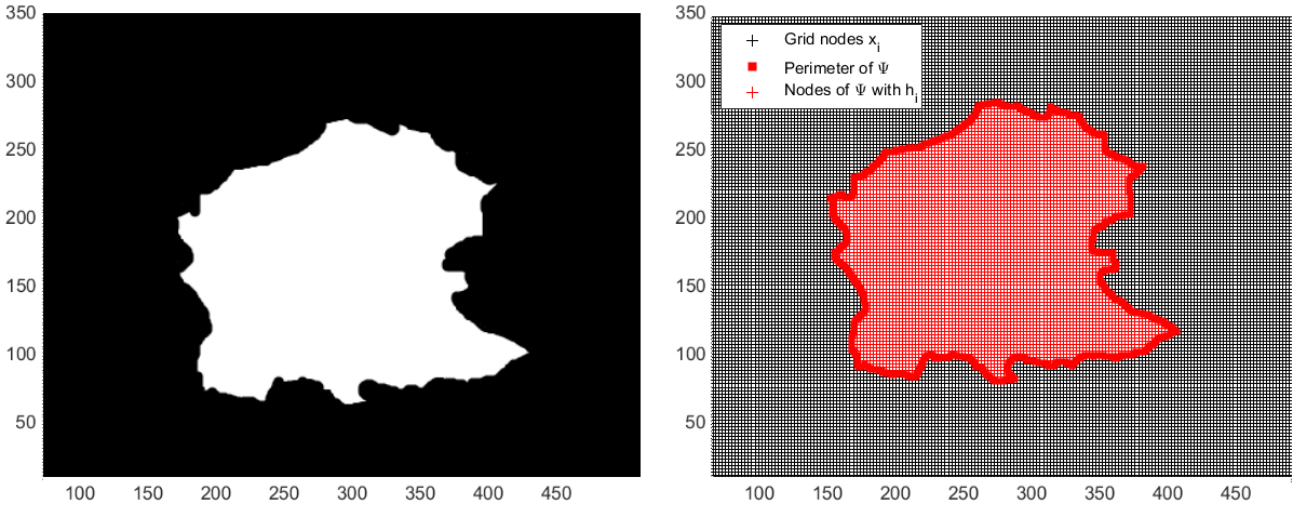


Figure 4: Particle generation: starting from the TIFF input file, the region of interest Ψ is delimited and defined on the grid nodes to get h_i (left). Finally, the region of interest is filled with a user-defined number of particles (right).

At each node x_i, the associated elevation h_i is extracted to define the topographic surface Z. Additionally, the gradient ∇Z_i at the grid nodes is computed and stored. All information regarding nodes, elements, and grid spacing can be organized within a structured data variable.

Following the terrain definition, the initial condition on the material is processed by treating Ψ as a pixel region within Ω. The TIFF data of Ψ is typecast into either a 16-bit unsigned integer or a double-

Algorithm 1 Particle generation algorithm for DAMPM

Require
DTM in TIFF format data for topography Z and initial condition Ψ on the material.

Do
From DTM, select the computational domain Ω.
Define a 2D square grid for Ω on the pixels of the DTM with $\mathbf{x}_i$ nodes.
From the nodes $\mathbf{x}_i$, select the corresponding elevations h_i and define the topography Z.
Compute and store gradient ∇Z_i on the grid nodes.

Store all information on nodes, elements and spacing in a `struct` variable.

Do
Define the initial condition on the material as pixel region Ψ in Ω.
Typecast Ψ as `uint16` or `double`.
Select the boundary domain of Ψ by generating the polygonal line $\mathcal{P}$ of $\partial\Psi$.
Associate to each node $\mathbf{x}_i$ of Ψ the corresponding height h_i (given by the DTM).

Create a user-defined 2D set of N_p particles with location $\mathbf{x}_p$ inside Ψ.
Define the particle initial condition Ψ_p, by interpolating h_i with $\mathbf{x}_p$ used as query points.
Get the particle elevations h_p.

Store in a `struct` all the info on particle locations $\mathbf{x}_p$ and elevations h_p

Get
Total mass `M`
Total volume `V`
Check for conservation of mass and volume with respect to the input data.

For $p \in \{1, ..., N_p\}$
Define physical initial conditions on each particle of Ψ_p (mass, velocity, stresses, volume, ...).
end

Store all the information on a usable input file (e.g. JSON file, h5 file or CSV file)

precision floating-point format. The boundary domain of Ψ is determined by generating its polygonal contour $\partial\Psi$, and each grid node x_i belonging to Ψ is assigned a corresponding height h_i, as shown in right panel of Figure 4.

A set of N_p particles is then generated within the region Ψ by defining a user-specified two-dimensional distribution of points x_p. The simplest approach is to sample these points uniformly within Ψ, ensuring a regular spatial distribution. However, an alternative strategy involves refining the sampling in regions where the topographic gradient ∇Z is more pronounced, allowing for a higher concentration of particles in areas with significant elevation variations. Once the particle locations x_p are established, the initial condition Ψ_p at each particle is defined by interpolating the values of h_i, using x_p as query points. This interpolation provides the corresponding particle elevations h_p. Figure 9 shows an example of this procedure. All the relevant information regarding the particle locations x_p and elevations h_p is stored in a structured variable. We strongly emphasize that the particles constituting Ψ_p in the DAMPM framework are, in fact, columns of material, whose height h_p is vertically averaged, in accordance with the depth-averaged formulation of the model [27, 7, 26]. Figure 5 illustrates the initial discretized condition Ψ_p with $1.5 \cdot 10^4$ particles within the domain Ω.

The algorithm proceeds by computing the total mass M and total volume V, verifying the conservation of mass and volume with respect to the input data. The last step of the procedure is to define a physical initial condition on mass, velocity, stresses, and volume on each particle of Ψ_p.

Finally, all the processed data is stored in a structured format suitable for numerical simulations,

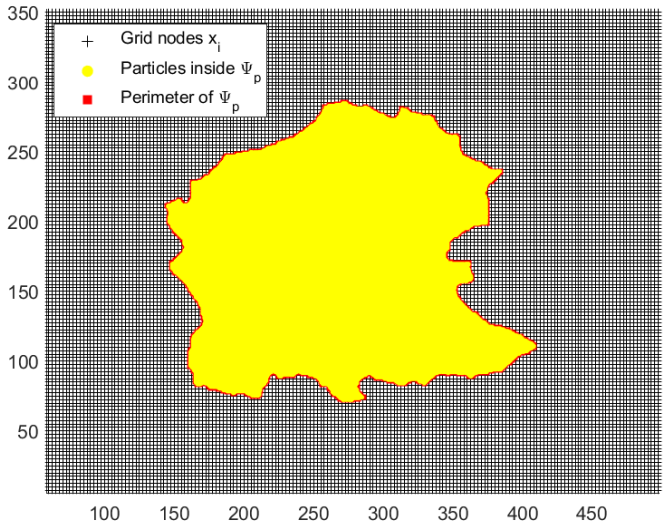


Figure 5: Particle-based initial condition Ψ_p of the continuum material Ψ discretized with $1.6 \cdot 10^4$ particles.

such as a JSON file, HDF5 file, or CSV file. The complete procedure is summarized in Algorithm 1.

We point out that, since the initial mass and volume are evenly distributed among all particles, their conservation between the continuous material description and the discrete particle representation, denoted as Ψ_p, is automatic. In MPM, the mass assigned to each particle remains constant throughout the entire simulation. This implies that the total mass of the system is simply the sum of the masses of all particles and is neither lost nor created, thus ensuring mass conservation at the particle level [11, 1, 2]. Furthermore, if the basis functions defined on the grid satisfy the partition of unity property, then the mass and moments computed on the particles are equal to those computed on the grid nodes [7, 11].

3 A sequential conversion from 2D to 3D

In this Section, we describe a sequential procedure that enables the conversion of the depth-averaged particle discretization Ψ_p, represented as columns of material, into a set Ψ_p^3 of fully three-dimensional particles. Each of these 3D particles retains its own physical properties, ensuring the conservation of mass, volume, and momentum throughout the transformation process [33].

Such a conversion approach is particularly relevant in the context of impact scenario analysis, especially when considering phenomena such as landslides or debris flows. The advantage of transitioning from a depth-averaged representation to a fully 3D particle-based description lies in the improved quantitative analysis along the vertical direction. While depth-averaged models provide an efficient approximation of large-scale flow dynamics [34, 26, 7, 35, 27], they inherently neglect detailed vertical variations, which can be crucial for accurately capturing impact forces, material redistribution, and localized interactions with obstacles. A 3D representation allows for a more precise evaluation of these effects, leading to better predictive capabilities in hazard assessment and mitigation strategies [36, 37, 38, 39].

The algorithm aims to convert depth-averaged particle data into a set of fully three-dimensional particles while ensuring the conservation of relevant physical quantities. The procedure requires the input of orography data, denoted as Z, and depth-averaged particle data, including positions, velocities, stresses, and momenta. The process begins by applying the G2P procedure to map the initial depth-averaged particles onto the nodal topography Z, obtaining the projected particle elevation Z_p. A spacing parameter $r \in \mathbb{R}^+$ is then selected to define the vertical separation between the new 3D particles. For each particle $p \in \{1, ..., N_p\}$, the number of 3D particles forming the vertical column is computed as

$$M_p = \left\lfloor \frac{\|h_p - Z_p\|_\infty}{r} \right\rfloor, \tag{1}$$

where h_p represents the initial height of the depth-averaged particle. If $M_p < 1$, the minimum value is enforced as $M_p = 1$ to ensure at least one 3D particle per column.

The attributes of the newly generated 3D particles are then defined and computed. The elevation of each new particle $i \in \{1, ..., M_p\}$ is determined by

$$z_p^3 = Z_p + \frac{i-1}{M_p - 1}(h_p - Z_p), \tag{2}$$

which distributes the particles uniformly along the vertical axis. The horizontal positions remain unchanged from the depth-averaged representation, such that $\mathbf{x}_p^3 = \mathbf{x}_p$. To ensure numerical conservation, the mass and volume of each 3D particle are assigned as $m_p^3 = \frac{m_p}{M_p}$ and $V_p^3 = \frac{V_p}{M_p}$ respectively. The velocity and acceleration components inherit the horizontal components from the depth-averaged particles, while the vertical counterparts are initially set to zero as

$$\mathbf{v}_p^3 = \begin{bmatrix} v_{x,p} \\ v_{y,p} \\ 0 \end{bmatrix}, \quad \mathbf{a}_p^3 = \begin{bmatrix} a_{x,p} \\ a_{y,p} \\ 0 \end{bmatrix}. \tag{3}$$

Finally, the stress tensor is updated according to the applied constitutive model.

After processing, the algorithm ensures data consistency by eliminating possible NaN or infinite values, verifying mass and momentum conservation, and storing the updated particle information in an external output file. The complete procedure is summarized in Algorithm 2.

Since the 3D discretization depends on both the horizontal spacing between the columns h_x and the vertical spacing between the 3D particles h_z, the error in the conservation of mass and momentum is $O(h_x^2)$ and $O(h_z)$, respectively. This implies that the error is quadratic with respect to the horizontal spacing and linear with respect to the vertical spacing. The proof of these estimates, in the context of the MPM formulation with linear basis functions, is provided in the Section 3.1.

Algorithm 2 2D-3D conversion algorithm

Require
Orography Z
Depth-averaged particle data: positions, velocities, stresses, momenta, ...

Do
Apply the G2P procedure to the nodal topography Z to get the particle projection Z_p.
Select a spacing $r \in \mathbb{R}^+$ between the new particles.

For $p \in \{1, ..., N_p\}$
Compute the number of 3D particles in each column as: $M_p = \lfloor \frac{\|h_p - Z_p\|_\infty}{r} \rfloor$

If $M_p < 1$
$M_p = 1$
EndIf
Define/Compute attributes on the new particles as:
Elevation: $z_p^3 = Z_p + \frac{i-1}{M_p - 1}(h_p - Z_p)$, for $i = 1, ..., M_p$
Positions: $\mathbf{x}_p^3 = \mathbf{x}_p$
Mass and volume: $m_p^3 = \frac{m_p}{M_p}$ and $V_p^3 = \frac{V_p}{M_p}$

Velocities: $v_{x,p}^3 = v_{x,p}$, $v_{y,p}^3 = v_{y,p}$, $v_{z,p}^3 = 0$
Accelerations: $a_{x,p}^3 = a_{x,p}$, $a_{y,p}^3 = a_{y,p}$, $a_{z,p}^3 = 0$
Update Stress tensor according to the constitutive model applied.
EndFor

Eliminate possible NaN or Inf values. Check for mass and momentum conservation. Store the new particle info in an input file (e.g. JSON, csv, h5)

From a practical perspective, Algorithms 1 and 2 can be coupled to generate useful input data for the simulation of a runout scenario followed by the impact of landslide events. From a computational standpoint, it is more efficient to simulate the landslide propagation phase using a depth-averaged solver, where the initial condition Ψ_p is generated by Algorithm 1. Subsequently, it is advantageous to apply Algorithm 2 to convert the depth-averaged representation into a full 3D model for a detailed analysis of the impact phenomenon.

The application of these two techniques is entirely independent of each other. In fact, the runout simulation can be performed first, and once it is stopped, the state variables and domain topology can be collected [33, 40]. At this stage, the depth-averaged output can be converted into a 3D input file for an external solver. This independence and versatility of the two algorithms constitute additional advantages of the proposed approach. In Figure 6, an example of the conversion between the DAMPM and 3D MPM models is shown. Each column of particles, depicted in blue, has been proportionally divided into a set of 3D particles, in red. For illustrative purposes, the initial condition was discretized with a very limited number of DAMPM particles.

However, it is evident that a higher concentration of particle columns leads to greater accuracy in the 3D discretization, thus reducing the loss of information regarding mass, volume, and momentum.

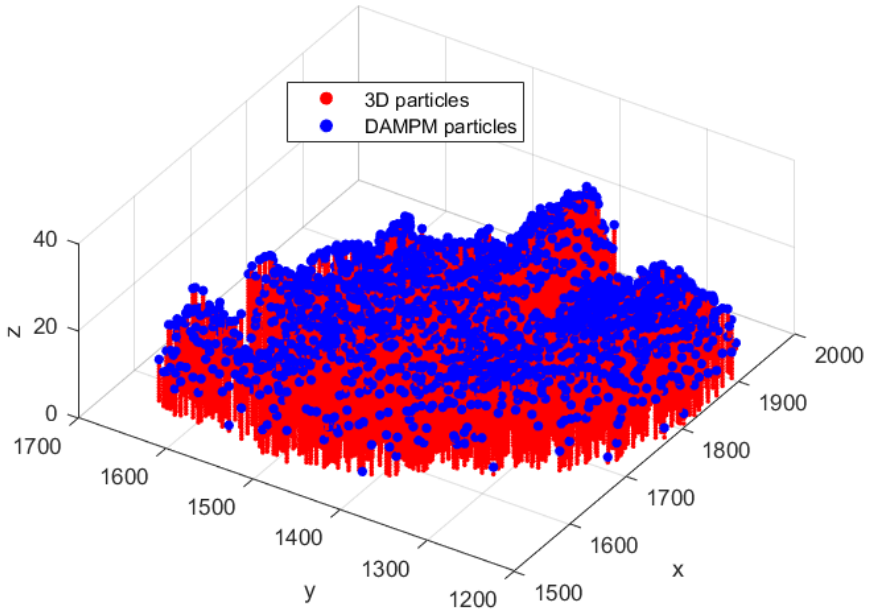


Figure 6: Comparison between DAMPM discretization (with blue dots) of a real landslide final state and the respective 3D MPM (with red dots).

3.1 Conservation of mass and momenta during the conversion phase

In this Section we prove that Algorithm 2 satisfies the numerical conservation of masses and momentum.

To this end, without losing of generality, we consider only the x-momentum and the mass and, by following [11], define the error ϵ for both the quantities, respectively as

$$\epsilon_m = \sqrt{\frac{\sum_p V_p \left\| m_p - M_{e,p} \right\|^2}{V_{tot}}}, \qquad \epsilon_{m\mathbf{v}_x} = \sqrt{\frac{\sum_p V_p \left\| (m\mathbf{v}_x)_p - (m\mathbf{v})_{e,p} \right\|^2}{V_{tot}}}, \tag{4}$$

where V_p, m_p, $(m\mathbf{v}_x)_p$ are the volumes, masses, $x-$momentum of the 3D particles respectively while the quantities $M_{e,p}$ and $(m\mathbf{v})_{e,p}$ are the exact counterparts. In (4), the quantity $\|\cdot\|$ denotes the Euclidean

norm, while V_{tot} is the total volume of the material. We now proceed to estimate the asymptotic trend of the error ϵ_m and $\epsilon_{m\mathbf{v}_x}$. It holds

$$\begin{aligned}\epsilon_m^2 &= \frac{\sum_p V_p \, \|m_p - M_{e,p}\|^2}{V_{tot}} \\ &\leq \frac{\sum_p V_p \, \|m_p\|^2 + \sum_p V_p \|M_{e,p}\|^2}{V_{tot}}.\end{aligned} \tag{5}$$

We can set $m_p = \rho V_p = \rho h_{x,p}^2 \, h_z$, where $h_{x,p}^2$ and h_z are the area of the p^{th} particle, with $h_x = h_y$, and the constant vertical spacing between the 3D material points, respectively.

Moreover, since we are assuming that the mass is equidistributed on each particle p, we can estimate the particle exact mass as $M_{e,p} = M_e/N_p$, where N_p is the total number of material points and M_e is the total exact mass. From this, by assuming that the density ρ is constant, it follows that

$$\begin{aligned}M_{e,p} &= \frac{1}{N_p} \int_\Omega \rho(\mathbf{x}) \, d\mathbf{x} \\ &\simeq \frac{1}{N_p} \rho \sum_{p=1}^{N_p} h_{x,p}^2 \, h_z., \\ &= \frac{1}{N_p} \rho h_z \sum_{p=1}^{N_p} h_{x,p}^2,\end{aligned} \tag{6}$$

since h_z does not depend on p and ρ is constant.

Under these hypotheses, by setting $h_x = \max_p h_{x,p}$ we have

$$\begin{aligned}\epsilon_m^2 &\leq \frac{\sum_p V_p \, |\rho \, h_{x,p}^2 \, h_z|^2 + \sum_p V_p \left| \frac{1}{N_p} \rho \, h_z \, \sum_p h_{x,p}^2 \right|^2}{V_{tot}} \\ &\leq \frac{|\rho|^2 |h_z|^2 \sum_p V_p \, \sum_p |\, h_{x,p}^2|^2 + |\rho|^2 \, |h_z|^2 \sum_p V_p \left| \frac{1}{N_p} \, \sum_p h_{x,p}^2 \right|^2}{V_{tot}} \\ &\leq \frac{2 N_p \mathcal{V}}{V_{tot}} \, |h_x^2 h_z|^2 \, |\rho|^2,\end{aligned} \tag{7}$$

where $\mathcal{V} = \max_p V_p$. Hence, $\epsilon_m = O(h_x^2 \, h_z)$.

Regarding the momentum $m\mathbf{v}_x$, by setting $\mathbf{v}_x = \max_p \mathbf{v}_{x,p}$, an analogous reasoning shows that

$$\begin{aligned}\epsilon_{m\mathbf{v}_x}^2 &= \frac{\sum_p V_p \, \|(m\mathbf{v}_x)_p - (m\mathbf{v}_x)_{e,p}\|^2}{V_{tot}} \\ &\leq \frac{\sum_p V_p \, \|(m\mathbf{v}_x)_p\|^2 + \sum_p V_p \|(m\mathbf{v}_x)_{e,p}\|^2}{V_{tot}} \\ &\leq \frac{N_p \, \mathcal{V} \, |\rho|^2 |h_x^2 \, h_z|^2 \|\mathbf{v}_x\|^2 + N_p \, \mathcal{V} \, |\rho|^2 |h_x^2 \, h_z|^2 \, \|\mathbf{v}_{e,p}\|^2}{V_{tot}} \\ &= \frac{2 N_p \, \mathcal{V} \, |\rho|^2}{V_{tot}} \, |h_x^2 \, h_z|^2 \, (\|\mathbf{v}_x\|^2 + \|\mathbf{v}_e\|^2) \\ &\leq \frac{2 N_p \, \mathcal{V} \, |\rho|^2}{V_{tot}} \, |h_x^2 \, h_z|^2 \, (2\|\mathbf{v}_e\|^2) \\ &= \frac{4 N_p \, \mathcal{V} \, |\rho|^2}{V_{tot}} \, |h_x^2 \, h_z|^2 \, \|\mathbf{v}_e\|^2,\end{aligned} \tag{8}$$

Since $\mathbf{v}_x = \max_p \mathbf{v}_{x,p} \leq \mathbf{v}_e$ and the total exact velocity $\mathbf{v}_e$ coincides with the particle exact velocity $\mathbf{v}_{e,p}$. Again, we obtain $\epsilon_{m\mathbf{v}_x} = O(h_x^2 \, h_z)$.

4 Numerical results

In this Section, we present some numerical results on particle generation for DAMPM, starting from TIFF images. We also demonstrate the application of Algorithm 2 for the conversion to the fully 3D MPM, with a numerical verification of mass and momentum conservation.

4.1 Depth-averaged particles generation

In this Section, we test Algorithm 1 in an ideal setting where the initial condition Ψ coincides with the entire domain Ω, which is extracted from a squared TIFF image of size 100×100 pixels. Figure 7 illustrates the set Ψ, where the nodes correspond to the pixels of the TIFF image. The initial condition on h_i is defined by the analytical formula

$$h_i(x_i, y_i) = 1 + 2\exp\left[-\frac{1}{150}\left((x_i - 50)^2 + (y_i - 50)^2\right)\right], \qquad (x_i, y_i) \in \Psi. \tag{9}$$

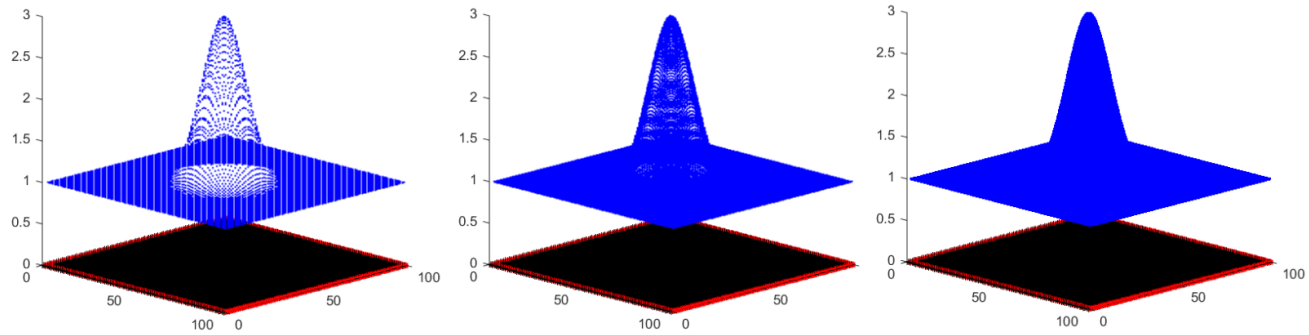


Figure 7: Initial condition Ψ_p discretized with a different number of particles. From left to right: 10^4 (1 particle per pixel), 10^5 and 10^6. On the bottom, the polygon $\partial\Psi$ shown in red, with the background grid, in black.

We applied Algorithm 1 to generate a different number of particles, starting with a number equal to the total pixels in the image, i.e., 10^4, and increasing up to 4 million particles, with a CPU time never exceeding 25 seconds. Figure 7 also shows three configurations of Ψ_p with a different number of particles, while the computational time trend is shown in Figure 8, where it can be observed that the execution time scales linearly with the number of particles.

Naturally, in practical applications, it is not necessary to generate such a high number of particles. However, the goal of this test was to assess the execution speed of the algorithm under extreme conditions [33] and these computations have been executed in double precision on a laptop with a 12th Gen Intel Core i7-1255U processor clocked at 1.70 GHz and 16GB of RAM.

4.2 Application to a real topography

In this Section, we apply Algorithm 1 to generate an initial condition for a landslide mass, defined within a domain $\Omega = [0, 1350] \times [0, 1200]\ m^2$ derived from the TIFF of a real DTM.

The initial resolution of the TIFF is 10 m per pixel; therefore, associating each material point with a single pixel would result in an excessively coarse distribution and a significant loss of information. Once the perimeter of the initial condition, highlighted in red in Figure 9 (left), is defined, we apply Algorithm 1 by specifying an arbitrary number of particles, independent of the DTM resolution. To achieve this, starting from the coordinates of each pixel, we consider the polygonal curve defining the perimeter of the initial condition Ψ, shown on the right side of Figure 9. We then interpolate the known elevation data from the DTM using the positions of the newly generated particles (chosen arbitrarily) as query points.

The result is a depth-averaged particle configuration, consisting of columns, on which any physical quantity required by the chosen constitutive model can be defined. Figure 10 shows the initial condition Ψ_p discretized with $1.1 \cdot 10^4$ particles.

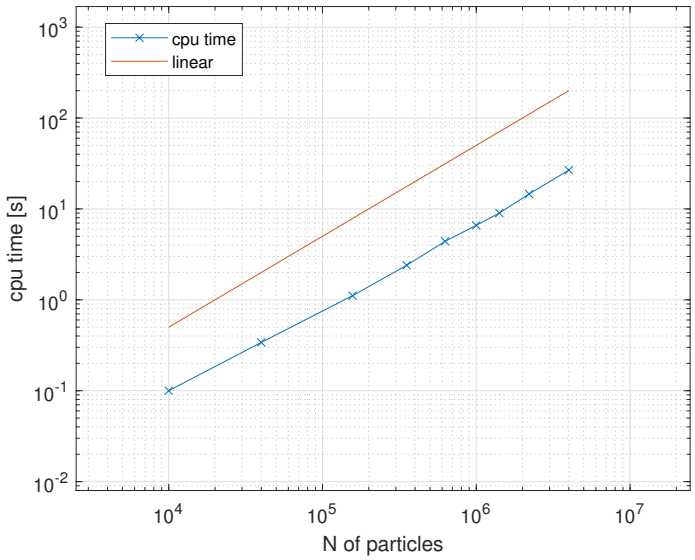


Figure 8: Particle-based initial condition Ψ_p of the continuum material Ψ discretized with $1.6 \cdot 10^4$ particles.

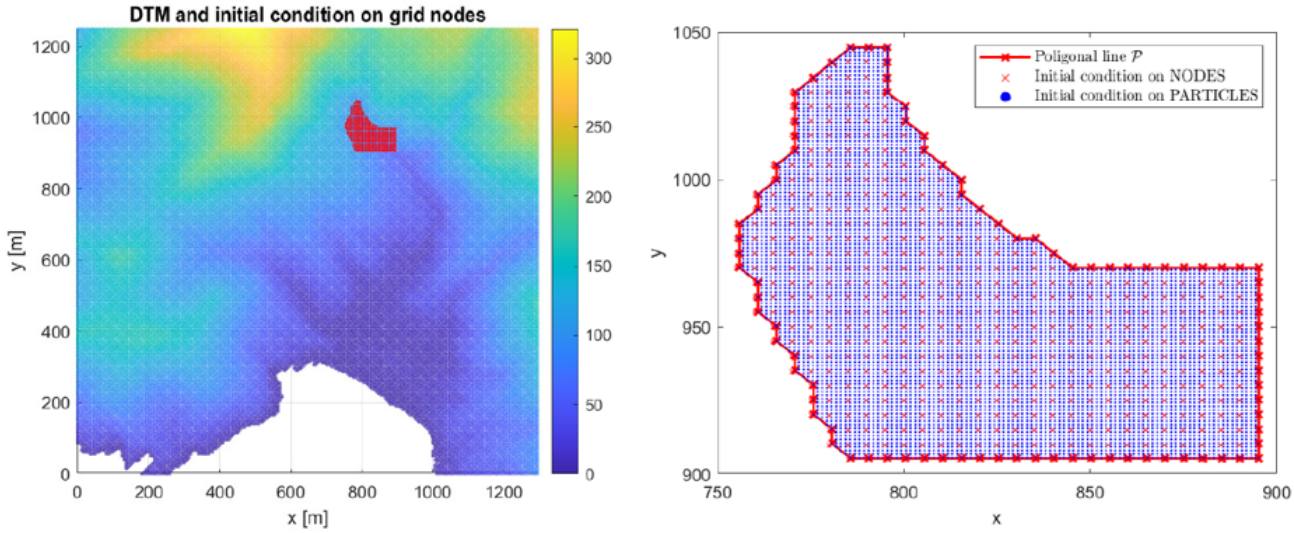


Figure 9: Particle generation: starting from the TIFF input file, the region of interest Ψ is delimited and defined on the grid nodes to get h_i (left). Finally, the region of interest is filled with a user-defined number of particles (right).

4.3 Mass and momentum during the conversion phase

The second test we carried out deals with the numerical conservation of both mass and momentum during the conversion phase between DAMPM simulation and 3D MPM.

To this end, we have considered an idealized scenario in which the computational domain Ω is defined by the square $[0, 5]^2$ m and the DAMPM initial conditions on height h and velocities $\mathbf{v}$ are respectively given by

$$\begin{aligned} h(x, y, 0) &= 10 - \sin(2x)\cos(y), \\ u(\mathbf{x}, 0) &= 2.001, \;\; v(\mathbf{x}, 0) = 0. \end{aligned} \tag{10}$$

We applied Algorithm 2 by varying the number of particles. This can be done by considering different values for the vertical spacing h_z, i.e.

$$h_z = \{0.01, 0.001, 0.0001, 0.00005, 0.00001\},$$

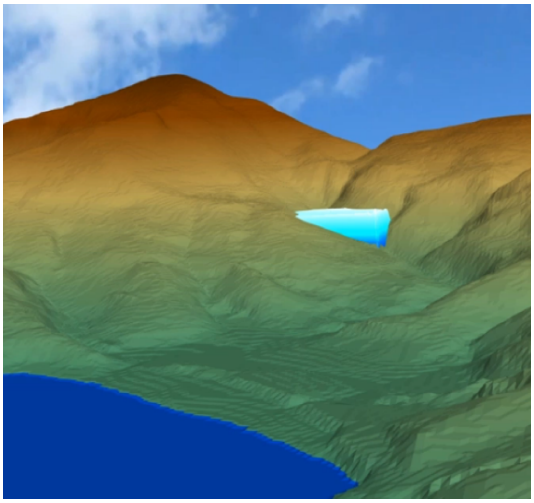


Figure 10: Particle-based initial condition Ψ_p of the continuum material Ψ after post-processing.

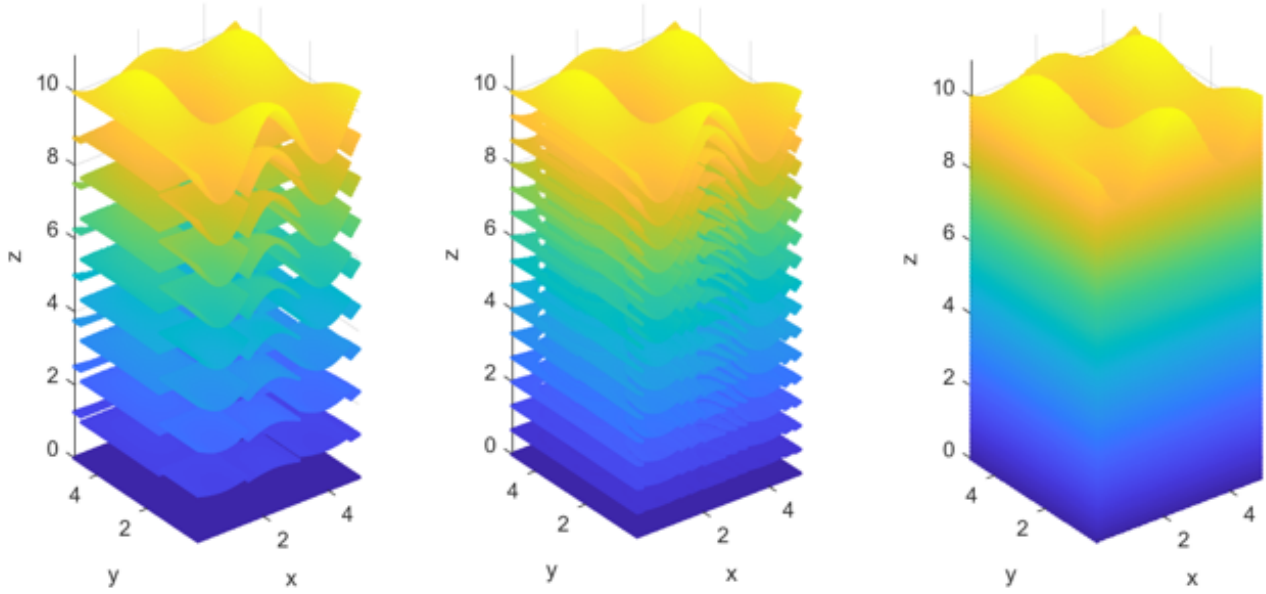


Figure 11: Different layer of discretization for the 3D MPM. From left to right $1.0 \cdot 10^5$, $5.6 \cdot 10^5$ and $2 \cdot 10^6$ particles.

and for the horizontal spacing h_x as

$$h_x = \{0.02, 0.01, 0.005, 0.0033, 0.0025, 0.0017\}.$$

In Figure 11, different layer configurations are shown for the transition from a depth-averaged approach to a fully 3D representation, by varying the number of vertical particles inside each column.

The convergence rate can be numerically verified by fixing each time one of the two spacing between h_z and h_x and varying the other one.

The top panel of Figure 12, shows the trend of the errors related to the mass and x-momentum with respect the spacing h_z. It can be observed that as the vertical spacing between particles h_z decreases, there is a clear improvement in the accuracy of both mass and momentum measurements. Specifically, the decreasing trend of both errors follows a linear behavior with respect to h_z, reflecting the estimate obtained in (8).

On the other hand, the bottom panel of Figure 12 shows the trend of the errors ϵ_m and $\epsilon_{m\mathbf{v}_x}$ with respect to the spacing h_x. Even in this case, the theoretical estimate in (7) is verified, since both the errors follow a quadratic trend, as h_x decreases.

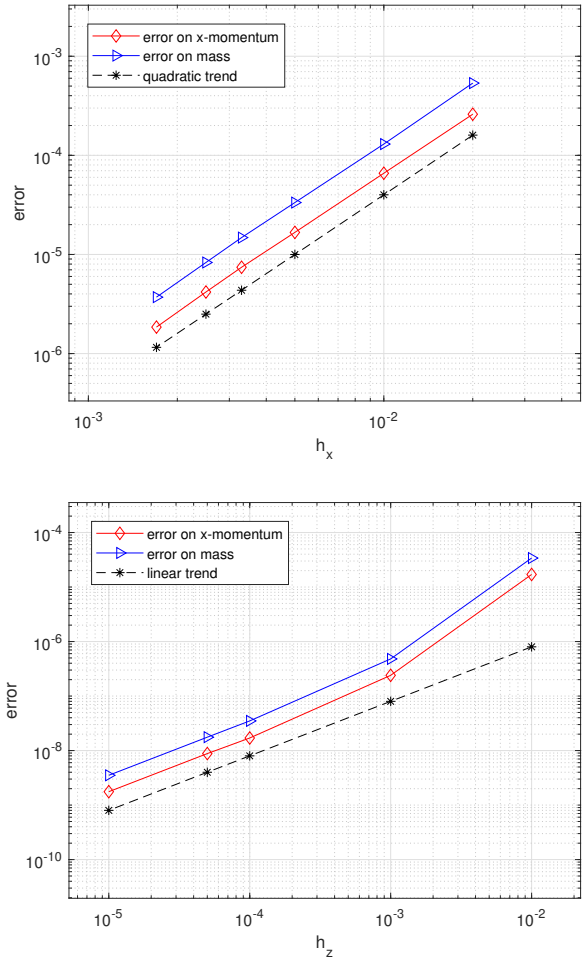


Figure 12: Conservation of masses and momentum during the conversion phase. On the top panel, the trend of the errors on masses and x-momentum with respect to h_z. On the bottom panel, the trend of the two errors with respect to h_x. In both cases, the errors on masses and momenta are expressed in kg and $kg \cdot m/s$, respectively.

4.4 Application to multiscale simulations

In this Section, we show two examples of application to multiscale simulations. The first test deals with a sliding mass that impacts a rigid barrier over an inclined plane. The second test concerns a runout simulation of a landslide followed by an impact against a rigid barrier. The code related to the DAMPM was developed entirely in-house, whereas the one used for the full 3D simulations is a software called KRATOS Multiphysics developed by [41, 42, 43].

4.4.1 Sliding to the wall

The idealized test we carried out deals with a dam break of a water column along a domain $\Omega = [0, 25] \times [0, 5]\ m^2$ described by the topography

$$Z(\mathbf{x}) = \begin{cases} 10 - \frac{1}{3}x & \text{if } \mathbf{x} \in [0, 10] \times [0, 5] \\ 6.67 & \text{otherwise} \end{cases}. \tag{11}$$

The initial conditions on height h and velocities u, v of the sliding material are prescribed as

$$h(\mathbf{x}, 0) = \begin{cases} 9.5 - Z(\mathbf{x}) & \text{if } \mathbf{x} \in [1.5, 5] \times [0, 5] \\ 0 & \text{otherwise} \end{cases}, \tag{12}$$
$$u(\mathbf{x}, 0) = v(\mathbf{x}, 0) = 0, \quad \forall x \in \mathbf{\Omega}.$$

In Figure 14 are depicted the final moments preceding the impact of the water column against a wall

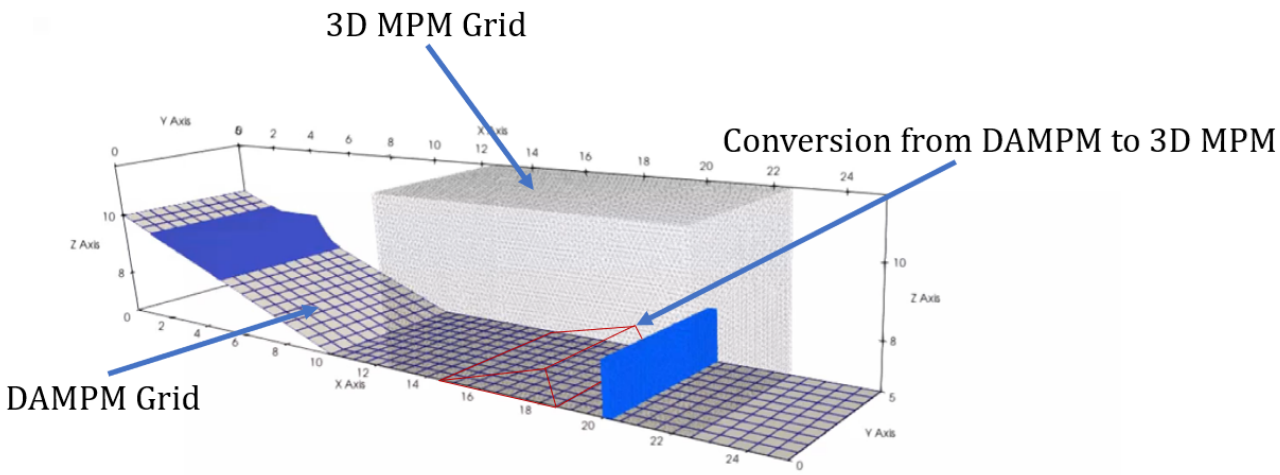


Figure 13: Sketch of the DAMPM and coupled MPM computational domain in the context of sliding to the wall test.

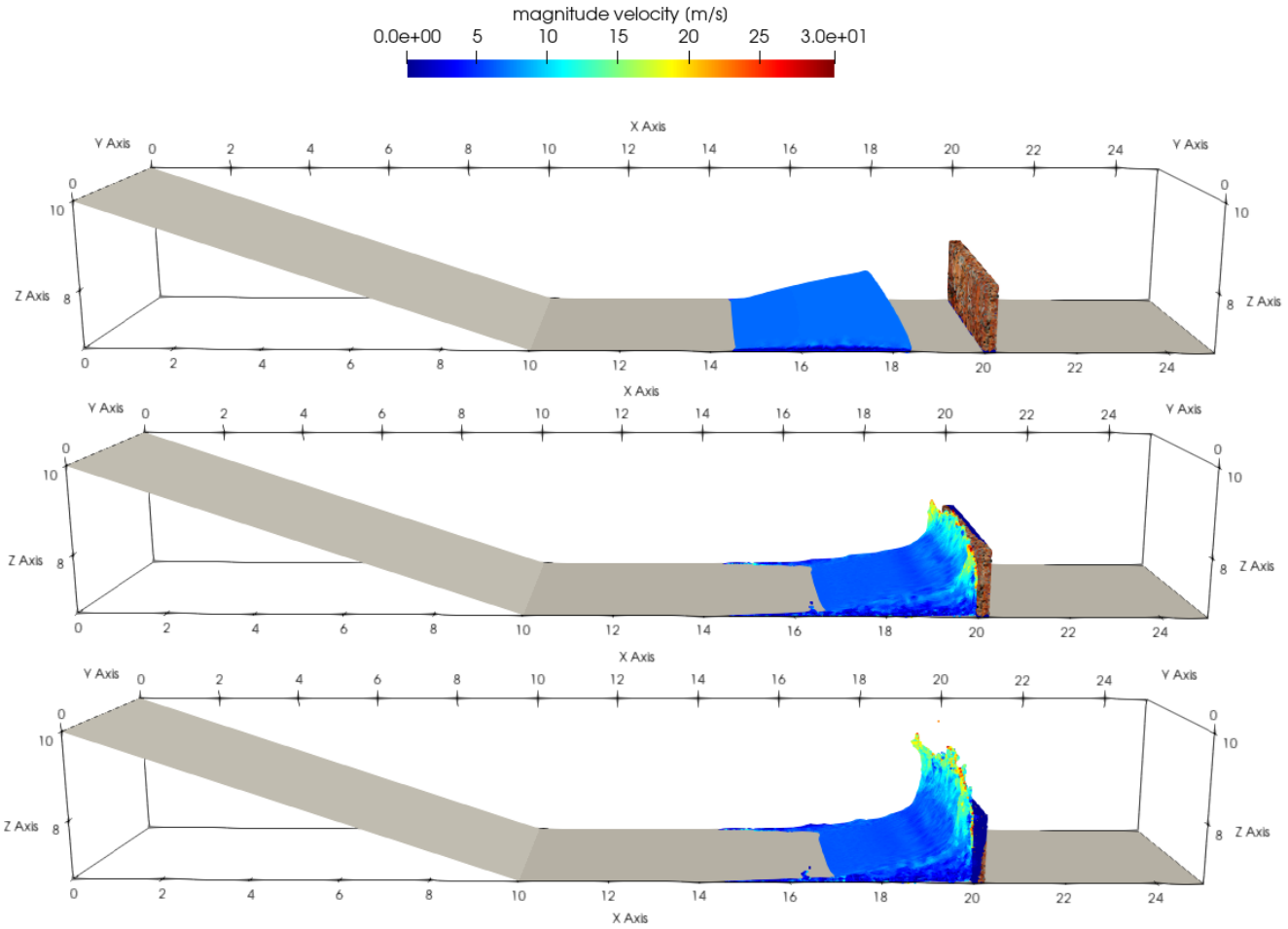


Figure 14: Snapshots of three different time steps of the Coupled MPM, at $T = 3\,s$ (on the top panel), $T = 3.3\,s$ (central panel) and $T = 3.5\,s$ (bottom panel).

located at $x = 20$, after the conversion from the depth-averaged model to the Coupled model made at time $T = 3\,s$, which is shown with a red line in Figure 13. The DAMPM simulation was carried out using $7 \cdot 10^4$ particles, and upon conversion to the 3D model, $5.03 \cdot 10^5$ particles were generated. The DAMPM computational domain has been discretized by using 500 elements, while the coupled MPM counterpart has about $1.1 \cdot 10^4$ elements.

As shown in Figure 14, the water mass at time $T = 3\,s$ exhibits a constant velocity of approximately $6.8\,m/s$, which aligns with theoretical expectations derived from motion along an inclined plane of about $18°$ with respect to the horizontal line, followed by motion on a frictionless flat domain. After the impact, occurring at time $T = 3.3\,s$, the water front surges, generating velocity peaks exceeding $20\,m/s$ and surpassing the barrier, reaching a height of nearly $10\,m$.

In the context of a frictionless dam-break, the lack of friction allows the water to propagate with maximum momentum, leading to a more pronounced stress response upon impact. Principal stresses develop as the kinetic energy of the water front is abruptly transferred to the barrier, inducing compression and shear effects in the fluid mass. However, the response of the barrier to impact also reveals that the stresses within the barrier reach similar magnitudes. This indicates a direct and substantial transfer of forces from the water mass to the barrier, confirming that the barrier is absorbing a significant portion of the kinetic energy transferred during the impact.

4.4.2 Realistic scenario

We considered a realistic scenario in which the computational domain defined by $\Omega = [0, 750] \times [0, 1000] m^2$ derived from a DTM in TIFF format. The study area is located on a hill in northern Italy, near Lecco (LC), which is characterized by complex terrain and natural slopes up to $60° - 68°$ prone to landslides, as shown in the left panel of Figure 15.

Specifically, the qualitative behavior of a mudflow impacting a rigid barrier placed along the path of the moving mass was investigated. In this case, the sliding material initially occupied a volume of approximately $5.8 \cdot 10^3\ m^3$, with a material density ρ set at $1300\ kg/m^3$.

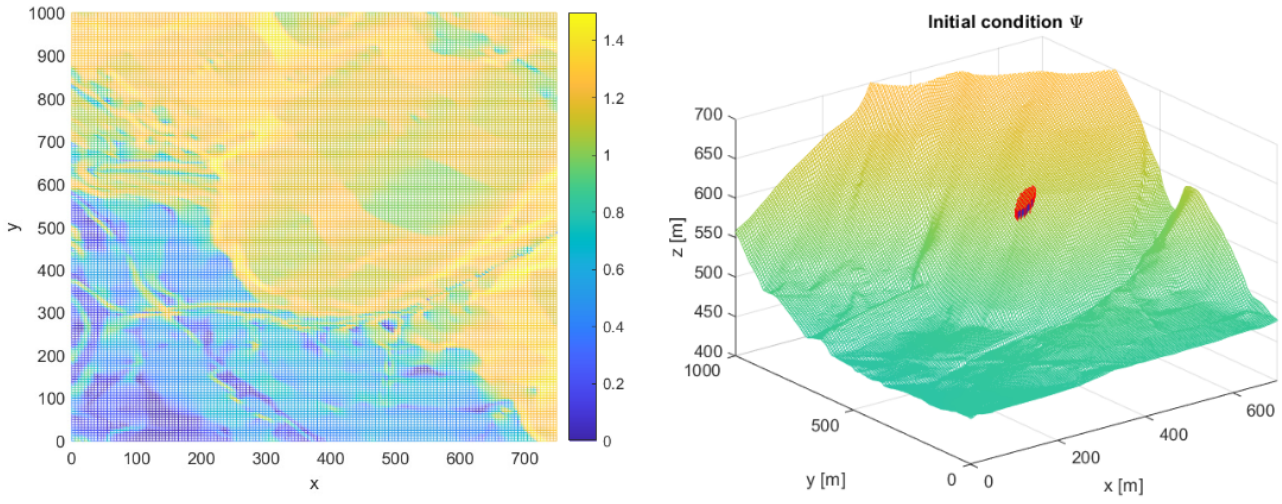


Figure 15: On the left, the colormap of the slope magnitude, in radiants, derived by the DTM data. On the right, the initial condition Ψ highlighted in red over the computational domain Ω.

The initial conditions Ψ on the material height h and its velocity field $\mathbf{v}$, shown in right panel of Figure 15, have been defined as

$$\begin{aligned} h(\mathbf{x}, 0) &= 1.02 Z(\mathbf{w}(\mathbf{x})) - Z(\mathbf{w}(\mathbf{x})) \\ u(\mathbf{x}, 0) = v(\mathbf{x}, 0) &= 0, \quad \forall x, \end{aligned} \tag{13}$$

where $\mathbf{w} = \left\{(x, y) \in \Omega : \frac{(y-550)^2}{4} + \frac{(X-420)^2}{5}) \leq 100\right\}$ and Z coincides with the DTM provided. In this test we have considered a total simulation time of $38\ s$.

From a computational perspective, the simulation was initially performed using the DAMPM for the first $30\,s$, which allowed for a more efficient simulation of the large-scale flow behavior. During this phase, $7.0 \cdot 10^4$ particles were used to represent the mudflow. To better capture the detailed dynamics near the barrier and the interaction with the complex 3D topography, the simulation was switched to a fully 3D Coupled MPM model for the remaining $8\ s$. This conversion resulted in a significantly finer resolution with $5.6 \cdot 10^5$ particles. Regarding the domain discretization, we used a cartesian grid with 30351 elements

to cover the entire Ω for the DAMPM simulation. For the impact simulation, we considered just a portion of $\Omega_{3D} \subset \Omega$ consisting in the square $\Omega_{3D} = [200, 400] \times [400, 600]\, m^2$, filled with $2.1 \cdot 10^5$ elements, as shown in the left panel of Figure 16.

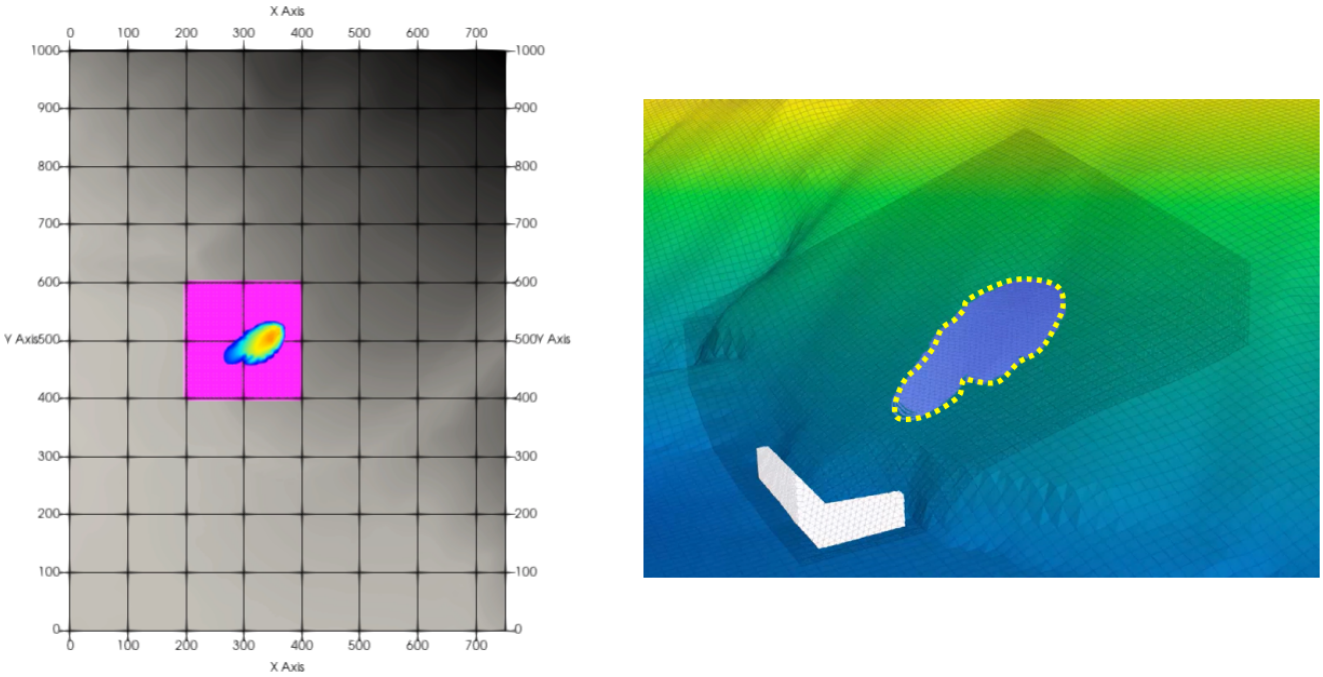


Figure 16: Realistic case test. On the left, the juxtaposition of the DAMPM computational domain, in gray, and the coupled MPM one, in magenta. On the right, an enlargement of the 3D MPM domain, with the L-shape fixed barrier, in white, and the 3D input landslide.

A rigid L-shaped barrier was strategically placed at the base of the hill, as depicted in the right panel of Figure 16. This barrier, with a height of 20 meters and a length of 100 meters, was designed to contain the advancing sliding mass and prevent further damage to the surrounding area. The effectiveness of the barrier in containing the flow and dissipating the kinetic energy of the mudflow was one of the key outcomes of this simulation.

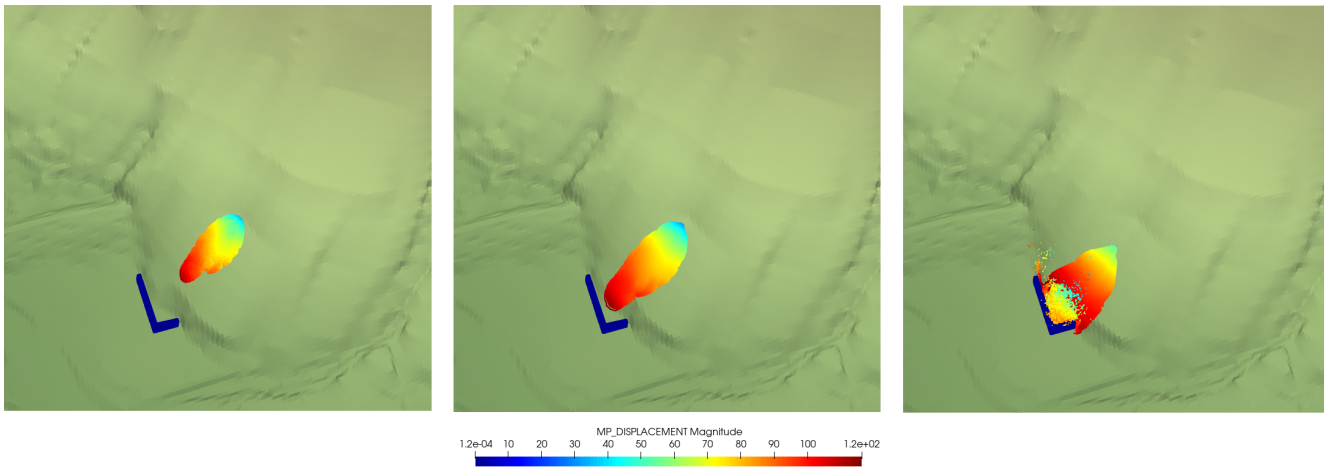


Figure 17: Snapshots of three different time steps of the Coupled MPM, at $T = 30\,s$ (left panel), $T = 33\,s$ (central panel) and $T = 38\,s$ (right panel).

Figure 17 shows different time steps of the front advancement following the conversion to the coupled model. Specifically, the instances shown are at $T = 30\,s$, $T = 34\,s$ and $T = 38\,s$. Although the front advancement speed was sustained and was estimated to be around $20\,m/s$ in the moments just before the impact, the presence of the barrier prevented the landslide mass from reaching the flat areas at the base of the hill.

5 Conclusions

In this work, we have presented a simple yet efficient technique for particle generation in the Depth-averaged Material Point Method based on DTM data in TIFF format, regardless the resolution of the data. Additionally, we have introduced a sequential method for converting between the depth-averaged and fully three-dimensional models within the MPM framework. The proposed approach ensures the conservation of mass and volume, which has been verified both in the 2D particle generation procedure and in the conversion process between models. Furthermore, we have validated the developed methodologies through a series of numerical tests. Potential improvements to the presented algorithms include the parallelization of the described procedure and the application of the conversion algorithm using particle generation techniques that differ from uniform sampling. Instead of relying on predefined spatial distributions, particle placement and resolution could dynamically adapt based on the local velocity field and acting forces. This would enhance the accuracy of particle-based methods, particularly in regions of high gradients or complex flow structures. Additionally, an important potential extension involves the implementation of inverse coupling techniques, transitioning from the fully 3D formulation to a depth-averaged representation.

Acknowledgements

M.F., C.d.F. and L.F. were supported by the Accordo Attuativo ASI-POLIMI "Attività di Ricerca e Innovazione" n. 2018-5-HH.0. C.d.F. and L.F. also acknowledge the support of "Dipartimento di Eccellenza 2023-2027". C.d.F. and L.F. have been partially funded by the Italian Research Center on High-Performance Computing, Big Data and Quantum Computing (ICSC), European Union - Next Generation EU. All authors are members of the Gruppo Nazionale Calcolo Scientifico-Istituto Nazionale di Alta Matematica (GNCS-INdAM).

References

[1] D. Sulsky, Z. Chen, H. Schreyer, A particle method for history-dependent materials, Computer Methods in Applied Mechanics and Engineering 118 (1993) 179–196.

[2] D. Sulsky, Erratum: Application of a particle-in-cell method to solid mechanics, Computer Physics Communications (1995).

[3] E. Wyser, Y. Alkhimenkov, M. Jaboyedoff, Y. Y. Podladchikov, A fast and efficient matlab-based mpm solver: fmpmm-solver v1.1, Geoscientific Model Development 13 (12) (2020) 6265–6284. doi:10.5194/gmd-13-6265-2020.

[4] S. Bardenhagen, E. Kober, The generalized interpolation material point method, CMES - Computer Modeling in Engineering and Sciences 5 (06 2004).

[5] S. Andersen, L. Andersen, Modelling of landslides with the material-point method, Computational Geosciences 14 (2010) 137–147. doi:10.1007/s10596-009-9137-y.

[6] M. Llano, M. Farias, D. Pedroso, An assessment of the material point method for modelling large scale run-out processes in landslides, Landslides 13 (12 2015). doi:10.1007/s10346-015-0664-4.

[7] M. Fois, C. de Falco, L. Formaggia, A semi-conservative depth-averaged material point method for fast flow-like landslides and mudflows, Commun. in Nonlin. Sci. and Numer. Simulat. 138 (2024) 108202. doi:https://doi.org/10.1016/j.cnsns.2024.108202.

[8] T. J. Charlton, W. M. Coombs, C. E. Augarde, igimp: An implicit generalised interpolation material point method for large deformations, Computers Structures 190 (2017) 108–125.

[9] A. Stomakhin, C. Schroeder, L. Chai, J. Teran, A. Selle, A material point method for snow simulation, ACM Trans. Graph. 32 (4) (jul 2013). doi:10.1145/2461912.2461948.

[10] W. Gao, H. Li, Z. Wang, J. Zhao, X. Zhang, A multi-material material point method for simulating soil–water–structure interaction in levee erosion, Computers and Geotechnics 100 (2018) 114–124.

[11] A. de Vaucorbeil, V. P. Nguyen, S. Sinaie, J. Y. Wu, Chapter two - material point method after 25 years: Theory, implementation, and applications, in: S. P. Bordas, D. S. Balint (Eds.), Advances in Applied Mechanics, Vol. 53 of Advances in Applied Mechanics, Elsevier, 2020, pp. 185–398. doi:https://doi.org/10.1016/bs.aams.2019.11.001.

[12] J. Guilkey, J. Hoying, W. J., Computational modeling of multicellular constructs with the material point method., J Biomech. 39(11) (2006) 2074–2086. doi:10.1016/j.jbiomech.2005.06.017.

[13] S. Bardenhagen, A. Brydon, J. Guilkey, Insight into the physics of foam densification via numerical simulation, Journal of the Mechanics and Physics of Solids 53 (3) (2005) 597–617. doi:https://doi.org/10.1016/j.jmps.2004.09.003.

[14] J. Nairn, Material point method simulations of transverse fracture in wood with realistic morphologies, Holzforschung 61 (06 2007). doi:10.1515/HF.2007.057.

[15] C. Jiang, C. Schroeder, J. Teran, A. Stomakhin, A. Selle, The material point method for simulating continuum materials, in: ACM SIGGRAPH 2016 Courses, SIGGRAPH '16, Association for Computing Machinery, New York, NY, USA, 2016. doi:10.1145/2897826.2927348.

[16] X. Wang, X. Zhang, W. Wang, Image-based material point method for simulation of complex materials, IEEE Transactions on Visualization and Computer Graphics 26 (7) (2019) 2247–2259.

[17] D. Hong, X. Zhang, X. Wang, Image-driven material point method for modeling heterogeneous materials, Journal of Computational Physics 418 (2020) 109615.

[18] W. Li, X. Hu, Y. Wang, A new material point method based on image analysis for simulating flow-like landslides, Engineering Geology 253 (2019) 169–180.

[19] M. Xie, W. Zhao, N. Ju, C. He, H. Huang, Q. Cui, Landslide evolution assessment based on insar and real-time monitoring of a large reactivated landslide, wenchuan, china, Engineering Geology 277 (2020) 105781. doi:https://doi.org/10.1016/j.enggeo.2020.105781.

[20] L. Brocca, F. Ponziani, T. Moramarco, F. Melone, N. Berni, W. Wagner, Improving landslide forecasting using ascat-derived soil moisture data: A case study of the torgiovannetto landslide in central italy, Remote Sensing 4 (5) (2012) 1232–1244. doi:10.3390/rs4051232.

[21] S. Mirzaee, M. Motagh, B. Akbari, Landslide monitoring using insar time-series and gps observations, case study: Shabkola landslide in northern iran, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-1/W1 (2017) 487–492. doi:10.5194/isprs-archives-XLII-1-W1-487-2017.

[22] A. Ferretti, C. Prati, F. Rocca, Permanent scatterers in sar interferometry, IEEE Transactions on Geoscience and Remote Sensing 39 (1) (2001) 8–20.

[23] C. Colesanti, A. Ferretti, C. Prati, F. Rocca, Monitoring landslides and tectonic motions with the permanent scatterers technique, Engineering Geology 68 (1-2) (2003) 3–14.

[24] N. Casagli, D. Colombo, A. Ferretti, A. Fumagalli, Satellite sar interferometry for wide-area mapping of landslide hazard, Sensors 10 (3) (2010) 1861–1875.

[25] J. Wasowski, F. Bovenga, Satellite remote sensing of landslide motion with advanced methods: A review, Landslides 11 (2014) 141–167.

[26] L. Guillet, L. Blatny, B. Trottet, D. Steffen, J. Gaume, A depth-averaged material point method for shallow landslides: Applications to snow slab avalanche release, Journal of Geophysical Research: Earth Surface 128 (8) (2023) e2023JF007092, e2023JF007092 2023JF007092. arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2023JF007092, doi:https://doi.org/10.1029/2023JF007092.

[27] K. Abe, K. Konagai, Numerical simulation for runout process of debris flow using depth-averaged material point method, Soils and Foundations 56 (5) (2016) 869–888, special Issue on the International Symposium on Geomechanics from Micro to Macro IS-Cambridge 2014. doi:https://doi.org/10.1016/j.sandf.2016.08.011.

[28] C. Jiang, C. Schroeder, A. Selle, J. M. Teran, A. Stomakhin, The affine particle-in-cell method, ACM Transactions on Graphics (TOG) 34 (4) (2015) 1–10.

[29] J. E. Guilkey, J. A. Weiss, Implicit time integration for the material point method: Quantitative and algorithmic comparisons with the finite element method, International Journal for Numerical Methods in Engineering 57 (9) (2003) 1323–1338.

[30] S. Bardenhagen, Energy conservation error in the material point method for solid mechanics, Journal of Computational Physics 180 (2002) 383–403. doi:10.1006/jcph.2002.7103.

[31] X. Wang, X. Zhang, A variational framework for image-based material point method simulations, Computer Graphics Forum 36 (8) (2017) 145–157.

[32] J. Su, L. Chen, Z. Huang, A gpu-based adaptive material point method for simulating elastoplastic materials, Computer Methods in Applied Mechanics and Engineering 347 (2019) 352–368.

[33] M. Fois, Particle-based numerical methods for landslides: run-out and impact analysis, Ph.D. thesis, MOX - Dept. of Mathematics, Politecnico di Milano (2024).

[34] F. Gatti, M. Fois, C. de Falco, S. Perotto, L. Formaggia, Parallel simulations for fast-moving landslides: Space-time mesh adaptation and sharp tracking of the wetting front, Int J Numer Meth Fluids 95 (8) (2023) 1286–1309. doi:https://doi.org/10.1002/fld.5186.

[35] F. Gatti, C. de Falco, M. Fois, L. Formaggia, A scalable well-balanced taylor-galerkin scheme for a lava flow depth-integrated model with point source vents, Computers and Mathematics with Applications 184 (2025) 153–167. doi:https://doi.org/10.1016/j.camwa.2025.02.014.
URL https://www.sciencedirect.com/science/article/pii/S0898122125000677

[36] V. Singer, A. Larese, A. Börst, R. Wüchner, K. Bletzinger, Partitioned mpm-fem coupling approach for advanced numerical simulation of massmovement hazards impacting flexible protective structures., 10th edition of the International Conference on Computational Methods for Coupled Problems in Science and Engineering (2023).

[37] Pasqua, Andrea, Leonardi, Alessandro, Pirulli, Marina, Coupling depth-averaged and 3d numerical models to study debris flow: Saint-vincent event, E3S Web of Conf. 415 (2023) 02015. doi:10.1051/e3sconf/202341502015.
URL https://doi.org/10.1051/e3sconf/202341502015

[38] F. Mintgen, M. Manhart, A bi-directional coupling of 2d shallow water and 3d reynolds-averaged navier–stokes models, Journal of Hydraulic Research 56 (6) (2018) 771–785.

[39] B. Domnik, et al., Coupling of full two-dimensional and depth-averaged models for granular flows, Journal of Non-Newtonian Fluid Mechanics 201 (2013) 56–68.

[40] M. Fois, A. Katili, C. de Falco, A. Larese, L. Formaggia, Landslide run-out simulations with depth-averaged models and integration with 3d impact analysis using the material point method, 16th WCCM/PANACM 2024, 2024. doi:10.23967/c.wccm.2024.054.

[41] M. V., P. Bucher, R. Zorrilla, R. Rossi, J. Cotela, C. Velázquez, M. Celigueta, J. Maria, T. Teschemacher, C. Roig, M. Maso, G. Casas, S. Warnakulasuriya, M. Núñez, P. Dadvand, S. Latorre, I. de Pouplana, J. González, F. Arrufat, et al., Kratosmultiphysics/kratos: Release 9.2 (v9.2)., Zenodo (2022). doi:https://doi.org/10.5281/zenodo.3234644.

[42] P. Dadvand, R. Rossi, M. Gil, X. Martorell, J. Cotela, E. Juanpere, S. Idelsohn, E. Oñate, Migration of a generic multi-physics framework to hpc environments, Computers & Fluids 80 (2013) 301–309. doi:10.1016/j.compfluid.2012.02.004.

[43] P. Dadvand, R. Rossi, E. Oñate, An object-oriented environment for developing finite element codes for multi-disciplinary applications, Archives of Computational Methods in Engineering 17 (3) (2010) 253–297. doi:10.1007/s11831-010-9045-2.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

13/2025 Scimone, R.; Menafoglio, A.; Secchi, P.
Estimating Non-Stationarity in Spatial Processes: an approach based on Random Domain Decomposition

14/2025 Nicolussi, F.; Masci, C.
Stratified Multilevel Graphical Models: Examining Gender Dynamics in Education

12/2025 Alessandro Andrea Zecchi, Claudio Sanavio, Simona Perotto e Sauro Succi
Improved amplitude amplification strategies for the quantum simulation of classical transport problems

11/2025 Tonini, A.; Dede', L.
Enhanced uncertainty quantification variational autoencoders for the solution of Bayesian inverse problems

10/2025 Botti, M.; Mascotto, L.; Vacca, G.; Visinoni, M.
Stability and interpolation estimates of Hellinger-Reissner virtual element spaces

09/2025 Quarteroni, A.; Gervasio, P.; Regazzoni, F.
Combining physics-based and data-driven models: advancing the frontiers of research with Scientific Machine Learning

08/2025 Botti, M.; Fumagalli, I.; Mazzieri, I.
Polytopal discontinuous Galerkin methods for low-frequency poroelasticity coupled to unsteady Stokes flow

07/2025 Patanè, G.; Nicolussi, F.; Krauth, A.; Gauglitz, G.; Colosimo, B. M.; Dede', L.; Menafoglio, A.
Functional-Ordinal Canonical Correlation Analysis With Application to Data from Optical Sensors

06/2025 Torzoni, M.; Manzoni, A.; Mariani, A.
Enhancing Bayesian model updating in structural health monitoring via learnable mappings

04/2025 Andrini, D.; Magri, M.; Ciarletta, P.
Nonlinear morphoelastic theory of biological shallow shells with initial stress