Brief communication: Depth-averaging of 3D depth-resolved MPM simulation results of geophysical flows for GIS visualization

Hervé Vicari[1,2,3], Michael Lukas Kyburz[1,2,3], and Johan Gaume[1,2,3]

[1]WSL Institute for Snow and Avalanche Research SLF, Davos Dorf, Switzerland
[2]Climate Change, Extremes, and Natural Hazards in Alpine Regions Research Center CERC, Davos Dorf, Switzerland
[3]Institute for Geotechnical Engineering, ETH Zürich, Zürich, Switzerland

Correspondence: Hervé Vicari (herve.vicari@slf.ch)

Abstract. Significant advances in full-3D modeling of geophysical flows have provided deeper insights into complex processes and predictive potential. However, practical application in the natural hazard community remains limited due to inadequate GIS integration of simulation results. This study addresses the oxymoronic transformation of 3D *depth-resolved* MPM simulation outputs into simplified *depth-averaged* results, such as flow depth and thickness, and slope-parallel and slope-normal velocities. Specifically, we present an algorithm that rasterizes scattered MPM outputs into a 2D format, enhancing their utility for hazard mapping and mitigation. We demonstrate our approach by applying it to an ice avalanche event, which is simulated using MPM and visualized in GIS. Notably, the 3D MPM shows slope-normal flow velocity components over terrain jumps, and our algorithm enables the identification of flow detachment from the terrain, which depth-averaged models typically neglect.

1 Introduction

With the emergence of computers, numerical modelling of geophysical flows (e.g., debris flows, rock, snow, and ice avalanches) has become an integrated part of natural hazards mapping and mitigation. Starting from the 1960s, depth-averaged models that simulate gravitational mass movements along a prescribed 2D topography were developed (see, e.g., the review by Eglit et al. (2020) for snow avalanche models; Hungr (1995) for debris flows). Later, depth-averaged models have been extended to 3D topography, also aided by higher spatial coverage and resolution of Digital Terrain Models (DTMs). Several depth-averaged models have been proposed to model rock avalanches (e.g., Mergili et al., 2012), debris flows (e.g., Denlinger and Iverson, 2004; McDougall and Hungr, 2004; Iverson and George, 2014; Tayyebi et al., 2021), and snow avalanches ~~(e.g., Christen et al., 2010; ?)~~(e.g., Christen et al., 2010; Vicari and Issler, 2025). The extensive scientific advancement and adoption of depth-averaged models can largely be attributed to their computational efficiency, as these models modify the 3D conservation equations by integrating them in the vertical direction or perpendicular to the topography—thus reducing the number of conserved flow variables that need to be resolved—and assume simplified rheologies. Additionally, their straightforward integration into Geographic Information Systems (GIS) has also contributed to their popularity, particularly among natural hazard agencies and practitioners focused on hazard mapping.

Recently, advancements in computing power have enabled the resolution of 3D conservation equations within a feasible (yet extensive) timeframe, making it possible to simulate geophysical flows at the slope scale. 3D models have been applied for

debris flows (e.g., Kwan et al., 2015; Koo et al., 2018), snow avalanches (e.g., Gaume et al., 2019; Li et al., 2021; Kyburz et al., 2024), rock and ice avalanches (e.g., Cicoira et al., 2022), and landslides (e.g., Franci et al., 2020; Tran et al., 2024). Nonetheless, although these models are scientifically popular, their use in practical applications for evaluating and mitigating natural hazards remains limited. We believe that this gap can be attributed to multiple factors: (i) inexperience within the natural hazards community and scarcity of guidelines; (ii) limited laboratory experiments to determine the constitutive parameters of geomaterials and challenges in determining these parameters in the field; (iii) non-user-friendly software; (iv) higher computational cost compared to depth-averaged models; (v) poor integration of simulation results with GIS tools. The latter issue stems from a mismatch between the output of these 3D *depth-resolved* simulations (i.e., where conservation equations are also resolved in the direction normal to the terrain) and the 2D raster-based formats used in GIS platforms. Indeed, in these simulations, physical quantities such as material positions and their velocities vary continuously along the terrain-normal direction and are expressed in absolute 3D coordinates. However, 2D raster data can represent only a single value per grid cell, requiring some form of averaging and reduction of the depth-resolved physical quantities to the local topography. This dimensional mismatch limits the straightforward use of 3D simulation results in hazard mapping. Within this work, we aim at addressing this last point ~~,~~ by developing a tool to export the results of fully 3D Material Point Method (MPM) simulations into a rasterized format. ~~In other terms~~Specifically, our aim is to convert ~~the *depth-resolved*~~ this depth-resolved MPM particle information into *depth-averaged* variables that can be saved in a 2D raster format and visualized in GIS. ~~Toward the end of our research, we became aware of a similar tool, very recently developed by Su et al. (2024) ,~~ Han et al. (2020) and Su et al. (2024) already proposed algorithms to extract physical quantities from 3D Smoothed-Particle Hydrodynamics (SPH) simulations. Nevertheless, considering the significance of MPM for modeling gravitational mass movements, there is a need for a dedicated algorithm; our method also introduces novel elements, especially concerning the exported depth-averaged variables. We illustrate the practical application of the depth-averaging algorithm by presenting an ice avalanche event simulated using MPM and subsequently exported to GIS. We then examine the depth-averaged variables derived from the 3D MPM simulation and discuss potential future enhancements of the exporting tool.

2 Methods

2.1 3D depth-resolved MPM model and material parameters

MPM is a hybrid Eulerian–Lagrangian method ~~for simulating~~ commonly used to simulate the behavior of ~~continuous materials . In the ice avalanche simulations discussed in this paper,~~ granular and fluid materials within a continuum mechanics framework. In our case study, we use a Drucker–Prager yield criterion ~~is used to model ice~~to model the constitutive behavior of ice. However, alternative material models may also be employed within the MPM model to simulate other materials, such as snow or soil (e.g., Gaume et al., 2019; Cicoira et al., 2022). The Drucker–Prager yield function is defined as:

$$q - \gamma(p + a) \leq 0, \tag{1}$$

where q and p are the deviatoric and mean stress, respectively. The material parameters γ and a represent friction and attraction, respectively. These can be derived from the Mohr-Coulomb yield criterion parameters—i.e., the friction angle φ and the cohesion c—given that the two yield criteria are assumed to be equivalent under triaxial compression conditions: $\gamma = (6\sin\varphi)/(3-\sin\varphi)$ and $a = c/\tan\varphi$. We assume a non-associative flow rule with zero dilatancy. The model incorporates softening, similar to the model of Cicoira et al. (2022), wherein we assume that the friction angle decreases from an initial $\varphi_i = 40°$ to a residual $\varphi_r = 20°$, and cohesion decreases from an initial $c_i = 1$ kPa to 0, depending on the accumulated deviatoric plastic strain ε_d^p. Unlike the simulations of Cicoira et al. (2022), we do not model explicitly erosion of glacier's ice and snow cover. ~~Therefore~~Instead, we assume that the basal friction is higher when the flow is in contact with the glacier ($\mu = 0.65$) ~~is higher than the basal friction in contact with bedrock and~~ than with bedrock or soil ($\mu = 0.55$). This ~~assumption is made to consider for potential erosion of wet compactsnow, resulting in decreased flow mobility(Li et al., 2022)~~frictional contrast yielded a reasonable match to the observed inundation area. One possible explanation for this assumption is that the erosion of compact, wet snow on the glacier surface may lead to reduction in flow mobility, as also suggested by Li et al. (2022). At $t = 0$, the ice bulk density is set equal to $\rho_0 = \rho_{p0} = 850$ kg/m^3, where ρ_p denotes the material points' density. The simulation is carried out with a grid size of $gs = 1$ m, with $n_g = 6$ material points per cell. A complete list of the material parameters is reported in the Supplements (S2).

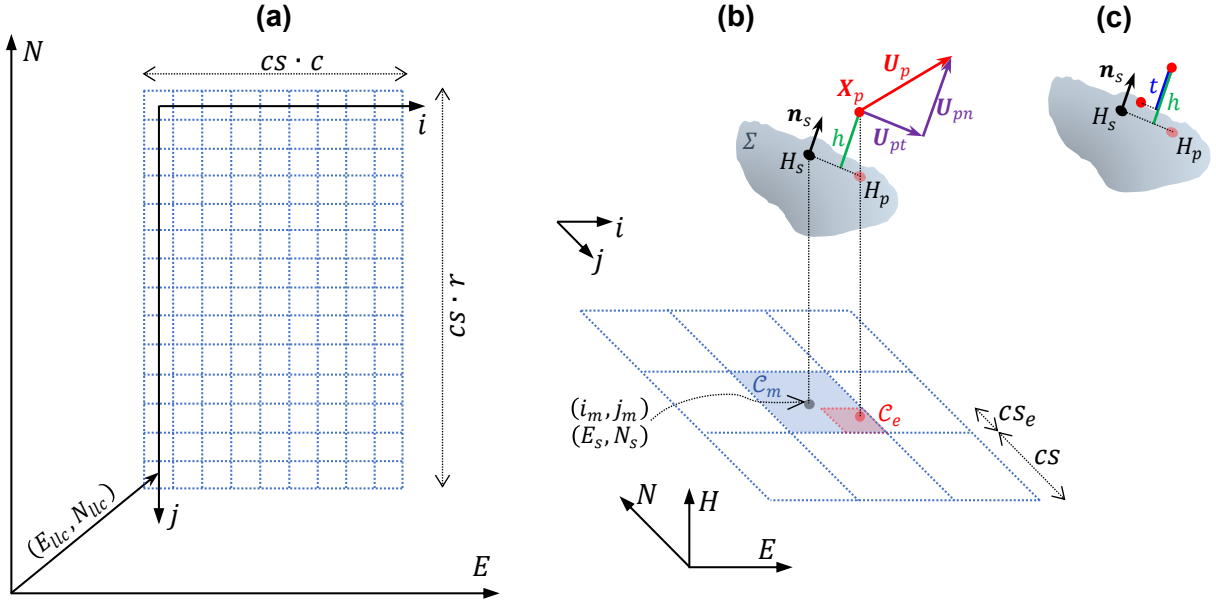

Figure 1. Discretization of the topography in ASCII format and calculation of flow-related physical variables: (a) Global and local coordinate system; (b) Calculation of the surface normals and mapping of the material points to their corresponding cell, with definition of flow velocity components, which are parallel ($\boldsymbol{U}_{pt}$) and normal ($\boldsymbol{U}_{pn}$) to the terrain; (c) Schematic of flow depth h, defined as the maximum distance from the topography among the material points within a cell; and flow thickness t, defined as the difference between the maximum and minimum distances from the topography among all material points within a cell. The two quantities may differ notably when the flow detaches from the terrain.

2.2 Topography representation

MPM simulations are performed on a generic topography Σ, whose elevation is defined in a global Euclidean system (ENH) as $H(E,N)$ (Fig. 1). We denote the latitude and longitude coordinates of sampled points on the surface as (E_s, N_s), and H_s their respective elevations on Σ. Each sampled coordinate (E_s, N_s) specifies the center of a cell with index (i,j), which is separated by a distance cs (the cell size of the DTM) from the adjacent cell (Fig. 1a):

$$i = \frac{E_s - E_{llc}}{cs}, \tag{2a}$$

$$j = \frac{-N_s + N_{llc} + cs \cdot (r-1)}{cs}, \tag{2b}$$

where (E_{llc}, N_{llc}) define the planar coordinates of the centre of the lower left cell of the DTM. The number of cells of the discretized topography in the east and north directions is indicated by c (i.e., columns) and r (i.e., rows), respectively. The parameters (E_{llc}, N_{llc}), $(cs \cdot c, cs \cdot r)$, cs, define the location, extension and resolution of the DTM, respectively.

~~Discretization of the topography in ASCII format: (a) Global and local coordinate system; (b) Calculation of the surface normals and mapping of the material points to their corresponding cell.~~

To project the simulation results on the topography, it is necessary to define a terrain orientation metric. The unit vector normal to the topography (outward-pointing) is expressed as (see, Fischer et al., 2012):

$$\boldsymbol{n}_s = \frac{(-\partial_E H, -\partial_N H, 1)^T}{\sqrt{1 + (\partial_E H)^2 + (\partial_N H)^2}}. \tag{3}$$

In the exporting code, the spatial gradients of the surface height are approximated using central difference at each grid point:

$$\partial_E H(i,j) \approx \frac{H(i+1,j) - H(i-1,j)}{2 \cdot cs}, \tag{4a}$$

$$\partial_N H(i,j) \approx \frac{H(i,j-1) - H(i,j+1)}{2 \cdot cs}. \tag{4b}$$

2.3 Depth-averaging of MPM results

The results of the MPM simulation at a certain time t_e include particles (p) with their position $\boldsymbol{X}_p(t_e) = (X_p, Y_p, Z_p)$ and velocity $\boldsymbol{U}_p(t_e) = (U_p, V_p, W_p)$, expressed in the global coordinate system ENH. The algorithm initially finds the indexes (i_m, j_m) correspondent to the DTM's cell $\mathcal{C}_m$ to which each particle p belongs (Fig. 1b):

$$i_m(t_e) = \underset{i \in \mathbb{Z} \cap [0, c-1]}{\operatorname{argmin}} \left| \frac{X_p(t_e) - E_{llc} - cs \cdot i}{cs} \right|, \tag{5a}$$

$$j_m(t_e) = \underset{j \in \mathbb{Z} \cap [0, r-1]}{\operatorname{argmin}} \left| \frac{-Y_p(t_e) + N_{llc} + cs \cdot (r-1) - cs \cdot j}{cs} \right|. \tag{5b}$$

Unlike Han et al. (2020) and Su et al. (2024), which project particles along the terrain normal, our approach uses vertical projection to assign particles to raster cells (see discussion in Sec. 2.3.2). The code offers the possibility to export the depth-averaged results to a new digital model with a different cell size (cs_e) than the original DTM's cell size (cs). The indices (i_e, j_e) for each cell $\mathcal{C}_e$ in the new model are thus computed using equations similar to Eqs. 5a and 5b, but with cs_e.

2.3.1 Flow heights

Once the DTM's cell $\mathcal{C}_m$ has been identified, the elevation H_p of the surface at the planar coordinate (X_p, Y_p) is calculated imposing the coplanarity condition

$$\boldsymbol{n}_s^T(i_m, j_m) \cdot [X_p(t_e) - E_s(i_m),\ Y_p(t_e) - N_s(j_m),\ H_p(t_e) - H_s(E_s, N_s)]^T = 0. \tag{6}$$

For each non-empty cell $\mathcal{C}_e$, the maximum flow depth (~~distance to~~ defined as the maximum distance from the topography Σ ~~,~~ in the surface-normal direction among all the material points within $\mathcal{C}_e$, see Fig. 1c) ~~, projected normal to the topography,~~ can be found as

$$h(i_e, j_e, t_e) = \max_{\boldsymbol{X}_p(t_e) \in \mathcal{C}_e} [(Z_p(t_e) - H_p) \cdot n_{sH}(i_m, j_m)], \tag{7}$$

where n_{sH} is the component of the normal-to-topography unit vector in the vertical direction.

Similarly, the maximum flow thickness (~~elevation difference between highest and lowest flow particle,~~ defined as the difference between the maximum and minimum distances from the topography Σ in the surface-normal direction among all material points within $\mathcal{C}_e$, see Fig. 1c) ~~, projected normal to the topography,~~ is calculated as

$$t(i_e, j_e, t_e) = \max_{\boldsymbol{X}_p(t_e) \in \mathcal{C}_e} Z_p h(i_e, j_e, t_e) - \min_{\boldsymbol{X}_p(t_e) \in \mathcal{C}_e} \left[\left(Z_p(t_e) - H_p \right) \cdot n_{sH}(i_m, j_m) \right]. \tag{8}$$

In both Eqs. 7 and 8, we provisionally neglect the increase in flow height associated to the volume of the material points. The code also allows setting a percentile to get the flow depth and thickness from all heights and thickness values of all particles within each cell, thus potentially excluding scattered particles.

2.3.2 Flow velocities

The velocity $\boldsymbol{U}_p$ of each particle p within a cell $\mathcal{C}_e$ is decomposed into the slope-normal and slope-parallel components (Fig. 1b):

$$\boldsymbol{U}_{pn}(\boldsymbol{X}_p, t_e) = [\boldsymbol{U}_p(\boldsymbol{X}_p, t_e) \cdot \boldsymbol{n}_s(i_m, j_m)] \boldsymbol{n}_s(i_m, j_m), \tag{9}$$

$$\boldsymbol{U}_{pt}(\boldsymbol{X}_p, t_e) = \boldsymbol{U}_p(\boldsymbol{X}_p, t_e) - \boldsymbol{U}_{pn}(\boldsymbol{X}_p, t_e). \tag{10}$$

The slope-normal and slope-parallel velocities can therefore be depth-integrated using (density-weighted) Favre-averaging:

$$\tilde{\boldsymbol{U}}_n(E,N,t_e) = \frac{1}{\int_0^h \rho(E,N,z,t_e)dz} \int_0^h \rho(E,N,z,t_e)\boldsymbol{U}_n(E,N,z,t_e)dz, \tag{11}$$

$$\tilde{\boldsymbol{U}}_t(E,N,t_e) = \frac{1}{\int_0^h \rho(E,N,z,t_e)dz} \int_0^h \rho(E,N,z,t_e)\boldsymbol{U}_t(E,N,z,t_e)dz, \tag{12}$$

where z is the coordinate normal to the topography and ρ is the flow *bulk* density. However, in the code, the integration is not performed in the normal-to-topography direction, but vertically for all particles within a cell $\mathcal{C}_e$. Although this is formally not consistent with the definitions of the depth-averaged velocities of Eqs. 11 and 12, this calculation avoids inconsistencies on ~~concave~~ concave-up topographies (i.e., duplication of particles to multiple cells is avoided; see also the discussion in Hergarten, 2024, p. 783) and carries substantial simplifications in coding. We therefore calculate the depth-averaged velocities by replacing dz with the discrete particle volume $\mathcal{V}_p$, and ρ with the particle density ρ_p:

$$\begin{aligned}\tilde{\boldsymbol{U}}_n(i_e,j_e,t_e) &\approx \frac{1}{\sum_{\boldsymbol{X}_p(t_e)\in\mathcal{C}_e} \rho_p(\boldsymbol{X}_p,t_e)\mathcal{V}_p(\boldsymbol{X}_p,t_e)} \sum_{\boldsymbol{X}_p(t_e)\in\mathcal{C}_e} \rho_p(\boldsymbol{X}_p,t_e)\boldsymbol{U}_{pn}(\boldsymbol{X}_p,t_e)\mathcal{V}_p(\boldsymbol{X}_p,t_e) \\ &= \frac{1}{n_p(t_e)} \sum_{\boldsymbol{X}_p(t_e)\in\mathcal{C}_e} \boldsymbol{U}_{pn}(\boldsymbol{X}_p,t_e),\end{aligned} \tag{13}$$

$$\begin{aligned}\tilde{\boldsymbol{U}}_t(i_e,j_e,t_e) &\approx \frac{1}{\sum_{\boldsymbol{X}_p(t_e)\in\mathcal{C}_e} \rho_p(\boldsymbol{X}_p,t_e)\mathcal{V}_p(\boldsymbol{X}_p,t_e)} \sum_{\boldsymbol{X}_p(t_e)\in\mathcal{C}_e} \rho_p(\boldsymbol{X}_p,t_e)\boldsymbol{U}_{pt}(\boldsymbol{X}_p,t_e)\mathcal{V}_p(\boldsymbol{X}_p,t_e) \\ &= \frac{1}{n_p(t_e)} \sum_{\boldsymbol{X}_p(t_e)\in\mathcal{C}_e} \boldsymbol{U}_{pt}(\boldsymbol{X}_p,t_e).\end{aligned} \tag{14}$$

Since the (solid) mass of each particle, $\rho_p\mathcal{V}_p$, is constant in MPM[1], the expressions on the far right of Eqs. 13 and 14 are derived, where n_p is the number of particles in each cell $\mathcal{C}_e$. Notably, in this context, the Favre-averaged velocities correspond to the *cell-averaged* velocities. In certain circumstances, however, the flow may dilate, which would lead to a reduction in the flow bulk density ρ from its initial value ρ_{p0}—this change in bulk density is not always purely due to plastic volumetric deformation from the constitutive model (i.e., resulting in the particles' density changing from ρ_{p0} to ρ_p; see, also, Li et al., 2021), but can also result from the formation of cracks and voids between granules of particles, as well as particles moving apart while air gets mixed into the flow (even though air is not explicitly modeled). Therefore, it would become necessary to replace $\rho_p\mathcal{V}_p$ in Eqs. 13 and 14 with ~~$[\rho_{p0}\mathcal{V}_{p0} + \rho_a(\mathcal{V} - \mathcal{V}_{p0})]$~~ $[\rho\mathcal{V} = \rho_{p0}\mathcal{V}_{p0} + \rho_a(\mathcal{V} - \mathcal{V}_{p0})]$, where ρ_a is the air density and $(\mathcal{V} - \mathcal{V}_{p0})$ is the extra-air volume ingested in the flow. Given that ρ_a is small and assuming limited flow dilation, we neglect the term $\rho_a(\mathcal{V} - \mathcal{V}_{p0})$.

[1]This assumption holds for the temporal evolution of a given particle's mass ($\rho_p\mathcal{V}_p = \rho_{p0}\mathcal{V}_{p0}$) and requires that the densities of various materials within a certain cell $\mathcal{C}_e$ are the same. In this study, we simulate only a single material type (ice), and so the mass of the particles remains unchanged.

2.3.3 Choice of exporting parameters for GIS visualization

GIS results are exported with a specified cell size cs_e. Choosing an appropriate value for cs_e is crucial: If the cells are too small, it may result in too many empty cells, especially in areas of the flow where the flow is dilated and particles are dispersed; Conversely, if the cells are too large, it might overestimate the flow extent, especially in areas where the flow has compacted. For an initial estimate, one can use the reference volume $\mathcal{V}$ around a particle as a surrogate for cell size~~:~~

$$]\mathcal{V}\approx$$

$$]\frac{\rho_{p0}\mathcal{V}_{p0}}{\rho},$$

. This volume is derived from mass conservation of a MPM particle, neglecting the extra-air mass (see Sec. 2.3.2), such that $\rho\mathcal{V} \approx \rho_{p0}\mathcal{V}_{p0}$, and therefore:

$$]\mathcal{V}\approx$$

$$]\frac{\rho_{p0}\mathcal{V}_{p0}}{\rho}.$$

(15)

~~where the extra-air mass has been neglected to obtain the far right side of the equation.~~ As the flow bulk density ρ may not be constant spatially and temporally~~(see discussion in Sec. 2.3.2)~~, the choice of cs_e requires some compromise and may cause the total flow mass (as derived from the exported flow height results) to be not conserved at all time steps. For instance, at the initial time step, results may be exported with $cs_e(t_0) \approx \sqrt[3]{\mathcal{V}_{p0}} = \sqrt[3]{gs^3/n_g}$. If, say, during the flow, ρ becomes smaller than ρ_{p0}, results should ideally be exported with $cs_e(t_e) > cs_e(t_0)$. This becomes particularly difficult when results must be exported not just at a specific time step, but also require the export of spatial distribution of the maximum value of a certain depth-averaged variable $\mathcal{F}$ (any of Eqs. 7, 8, 13, 14) over all time steps (this forcedly requires selecting a constant cs_e):

$$\mathcal{F}_{\max}(i_e, j_e) = \max_{t_e = k\cdot\Delta t_e,\ k=0,1,\ldots,\frac{t_{\text{sim}}}{\Delta t_e}} \mathcal{F}(i_e, j_e, t_e), \tag{16}$$

where t_{sim} is the simulated flow duration and Δt_e is the exported time step. Furthermore, to avoid spatial-loss of information of maximum values of the depth-averaged flow variables, Δt_e should be chosen as

$$\Delta t_e \approx \frac{cs_e}{\|\tilde{\boldsymbol{U}}\|_{\max}}, \tag{17}$$

where $\|\tilde{\boldsymbol{U}}\|_{\max}$ is the maximum (over time and space) depth-averaged speed projected in the EN plane.

3 Results and discussion

We first validate the exporting tool on a simplified case of a frictionless block moving on ~~an~~ a 2D inclined plane and then falling from a cliff (see Supplements, S1). Subsequently, we simulate in MPM the ice avalanche that resulted from the collapse of a $150000\ \mathrm{m}^3$ portion of the Whymper serac (Courmayeur, Italy) in 1998 (Fig. 2a). The results are exported using $cs_e = 1\ \mathrm{m}$ (estimated from Eq. 15) and $\Delta t_e = 0.2\ \mathrm{s}$ and visualized in Figs. 2b–g. Since the current version of the code is not parallelized, and consequently considerably slow, we provisionally used a large time step Δt_e, about 10 times greater than what suggested by Eq. 17.

The simulated deposit is shown in Fig. 2b (flow depth h, normal to the terrain) and the flow ~~extent~~ inundation area is shown in Fig. 2c (plotted in terms of maximum depth-averaged slope-parallel flow speed). The runout and flow extent reasonably match the mapped extent of the event. Nonetheless, the simulated left bank lobe of the avalanche stops approximately $300\ \mathrm{m}$ up-slope compared to the mapped runout. Similarly, the right lobe of the simulated avalanche exhibits a shorter runout compared to the actual event. This reduced simulated mobility may result from excessive material deposition in the crevasses (the real terrain DTM at the time of the ice avalanche is unknown and could have had smaller crevasses than the DTM used for simulations) and not explicitly modeling the erodible glacier cover. Furthermore, we only made a few attempts to alter the parameters of the ice material in MPM (in particular, the residual friction and the basal frictions) without further optimization of the simulation results.

~~In this section, we~~ We now briefly analyze the influence of complex terrain topography on flow behavior, which can be effectively captured by the depth-averaged results, shown in Figs. 2d–i at $t_e = 50\ \mathrm{s}$. In regions with curved topographies and jumps (e.g., cliffs, crevasses), high values of the depth-averaged slope-normal speed (see Fig. 2d) are calculated, whereas on more gentle terrain the ice flows approximately parallel to the terrain (see Fig. 2e). Specifically, on ~~convex~~ convex-up topographies along the flow direction ~~,~~ (such as overhangs, or cliff edges), the ice flows in the out-of-slope direction (i.e., positive, blue values of $\boldsymbol{n}_s^T \cdot \tilde{\boldsymbol{U}}_n$)~~, while it flows toward the slope on concave topographies along the flow direction~~ . In contrast, on concave-up sections (such as gullies or terrain depressions) the flow is directed toward the topography (i.e., negative, red values of $\boldsymbol{n}_s^T \cdot \tilde{\boldsymbol{U}}_n$). Figure 2h shows the calculated depth-averaged velocities extracted along a linear profile of the glacier (the profile line is shown in Fig. 2d–g). A cliff is located between $\approx 60\ \mathrm{m}$ and $\approx 80\ \mathrm{m}$ ~~in the~~ of this profile, causing the flow to separate from the topography ($\boldsymbol{n}_s^T \cdot \tilde{\boldsymbol{U}}_n > 0$) at the jump and subsequently collide with the topography ($\boldsymbol{n}_s^T \cdot \tilde{\boldsymbol{U}}_n < 0$) upon landing. In other areas where the curvature of the topography is smaller and there are no terrain jumps, the calculated slope-normal component ~~also appears to be smaller, or even~~ is smaller, and mostly negligible compared to the slope-parallel velocity component. The flow detachment from the topography is also evident by comparing the flow thickness (Fig. 2f) to the flow depth (Fig. 2g). Figure 2i presents the comparison of these two parameters along the profile line: across the jump and upon landing (splash), the flow depth significantly exceeds the flow thickness—signifying the flow's detachment from the surface (see also Fig. 1c)—whereas the values of t and h are ~~similar~~ not so far apart when the flow sticks to the terrain (see also the additional analysis in the Supplements, S3). Unlike in depth-averaged models, the detachment from the terrain in our 3D simulations allows to correctly capture that, while in airborne motion, the material does not experience basal flow

resistance. Similarly, when the flow impacts the terrain, the 3D MPM can effectively capture the predominant motion towards the terrain, potentially leading to increased basal flow resistance, as well as compaction and/or dilation of the material. This may lead to more precise dynamics compared to depth-averaged models, particularly on complex terrains that exhibit jumps and substantial curvature[2]. These spatial insights highlight how our export tool reveals terrain-driven flow dynamics that are not easily interpreted directly from raw 3D MPM outputs.

[2]Depth-averaged models may also somehow mimic reduced/increased basal friction when flowing over convex/concave topographies, if centrifugal curvature-induced effects—thus reducing/increasing the slope normal stress—are being implemented. Moreover, frictional parameters are, in some operational models (e.g., Bartelt et al., 2017), adjusted heuristically based on the curvature of the terrain.

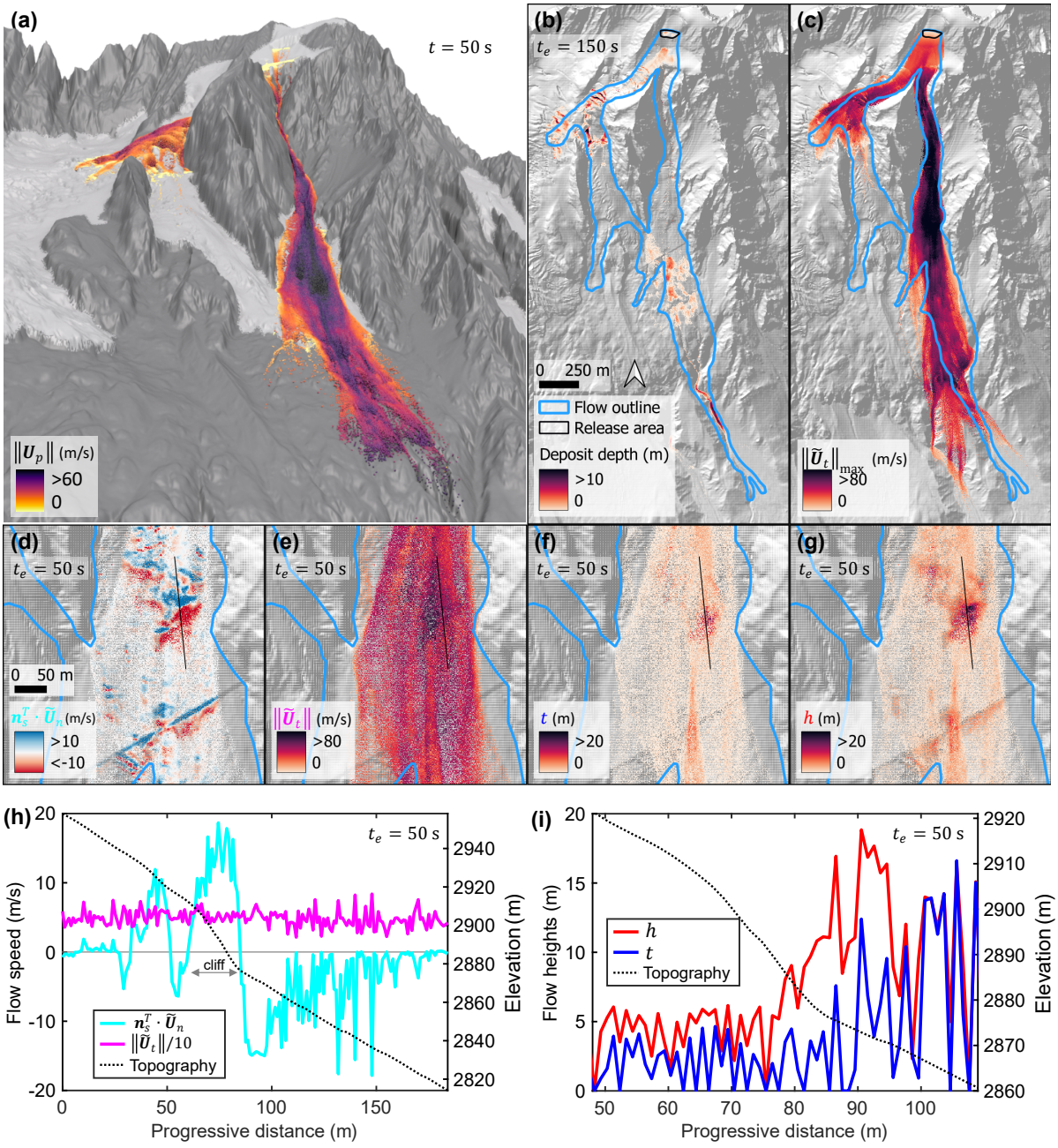

Figure 2. Depth-resolved MPM simulations and GIS visualization of exported depth-averaged results. (a) 3D visualization at $t = 50$ s of the MPM simulation of the Whymper ice avalanche in 1998. (b) Exported final ~~deposit~~deposition depth ($h(t_e = 150\ \text{s})$). (c) Maximum depth-averaged slope-parallel speed over all time steps ($\tilde{\boldsymbol{U}}_{t,\max}$). (d) Depth-averaged slope-normal velocity ($\tilde{\boldsymbol{U}}_n$) at $t_e = 50$ s. (e) Depth-averaged slope-parallel speed ($\tilde{\boldsymbol{U}}_t$) at $t_e = 50$ s. (f) Flow thickness (t) at $t_e = 50$ s. (g) Flow depth (h) at $t_e = 50$ s. (h) Slope-normal ($\tilde{\boldsymbol{U}}_n$) and slope-parallel ($\tilde{\boldsymbol{U}}_t$) speeds at $t_e = 50$ s along a profile line. (i) Flow thickness (t) and depth (h) at $t_e = 50$ s along a profile line (black line in d–g).

4 Conclusion and outlook

In this study, we presented a methodology to transform 3D MPM simulation outputs into 2D rasterized formats that can be visualized with GIS tools. This methodology entails the conversion of *depth-resolved* MPM variables into *depth-averaged* variables. More specifically, we described the computation of flow height metrics, like flow depth and thickness, and flow velocity metrics, like Favre-averaged slope-normal and slope-parallel velocities. Thus, this approach facilitates harvesting the benefits of 3D numerical methods (Li et al., 2021; Cicoira et al., 2022), including the simulation of geophysical flows over complex topographies, terrain jumps, cavities, and interacting with mitigation structures, while enabling practitioners to visualize the model results on GIS maps similar to traditional depth-averaged simulation outputs. Therefore, this tool represents a step forward in using 3D MPM for hazard assessment and mapping of geophysical flows.

The application of the depth-averaging tool to the simulation of an ice avalanche highlighted how complex terrain, like with cliffs and crevasses, can significantly alter flow velocities, causing them to deviate from being parallel to the ground. The computation of slope-normal velocities and the distinction of flow depth ~~and flow thickness~~ (distance between the terrain surface and the highest flow particle) and flow thickness (projected distance between the highest and lowest particles in each cell) revealed that the material might detach from the terrain during jumps and impact the terrain upon landing, potentially influencing flow dynamics through dilation, compaction, and variations in internal and basal stresses. Additional research will be needed to study in detail the effect of complex topographical features on avalanche characteristics. Furthermore, we observed that crevasses along the flow path could lead to significant deposition, further influencing subsequent flow dynamics. Thus, selecting an appropriate DTM appears critical for simulating ice avalanches over glaciers: this needs to be further investigated in the future. Alternative constitutive models for ice, such as incorporating rate dependency, will be required in future studies to achieve better agreement between the simulated ice avalanche and the actual observed runout.

The depth integration of 3D simulation results presented in this work will allow future comparison between 3D MPM simulations and depth-averaged simulations (Wirbel et al., 2024) to verify the assumptions made in depth-averaged numerical models, such as neglecting flow velocities normal to the terrain, assuming the flow sticking to the terrain, constant bulk flow density, and uniform velocity profiles. Additional work is required to broaden the utility of our depth-averaging tool. In the future, we plan to export density and velocity profiles along the flow depth direction. These data could be crucial for determining the pressure distribution of avalanches impacting obstacles. Such *vertical* information (e.g., Kyburz et al., 2024) may, for example, be exported onto raster, to replace traditional empirical analytical models of impact pressure, which (typically) rely solely on depth-averaged density and velocity. The tool could also be used to determine topography changes and erosion rates when an erodible bed layer is explicitly modeled in MPM (e.g., Li et al., 2022).

Finally, a similar framework could be applied in the future to depth-average other 3D particle-based models' results, like SPH and DEM ~~. Su et al. (2024) recently proposed an algorithm to extract~~ (Discrete Element Method). Recently, similar algorithms were proposed by Han et al. (2020) and Su et al. (2024) to extract depth-averaged physical quantities from 3D SPH simulations. ~~In their approach, flow particles quantities are assigned to the corresponding closest boundary particle , allowing for~~ Specifically, the ~~extraction~~ algorithm by Han et al. (2020) assigns each 3D SPH particle to a boundary particle by

projecting particle positions along the terrain-normal direction. Physical quantities (e.g., flow depth and basal velocities) are then computed for each boundary particle based on all SPH particles falling within its associated cell. In contrast, Su et al. (2024) identifies contributing SPH particles as those located within a cylindrical region whose longitudinal axis is centered on each boundary particle and aligned with the terrain normal. Our approach differs in that MPM particles are assigned to 2D raster cells based on vertical projection. While the computation of flow depth ~~(measured perpendicular to~~ and velocity components is conceptually similar across these methods, our implementation includes additional features: the ability to exclude scattered particles from the flow depth calculation, and the explicit computation of the ~~topography) and averaged velocities. Instead of exporting results on a~~ flow thickness. While Han et al. (2020) and Su et al. (2024) assign results to boundary particles, they can also export the data to 2D ~~rasterized grid format like ours, their results are saved on the boundary particles and visualized using ParaViewsoftware.~~ grids visualized in ParaView. Our method similarly produces rasterized outputs and can directly map quantities to a georeferenced 2D grid, which can be seamlessly exported to a GIS environment. Moreover, in contrast to the mesh-free SPH method, MPM solves momentum conservation on a background Eulerian grid. Thus, an alternative approach could rely on the direct extraction of physical quantities from the MPM code, straightforwardly utilizing the transferred velocities and masses at the grid nodes—thus eliminating the need for a separate export tool.

Code availability. The simulations were performed using the MPM code of Gaume et al. (2018). The MPM simulation results were first opened and visualized in the software Houdini. The depth-averaging code was implemented as a Python node within Houdini. In the Supplements, we provide the depth-averaging code.

Author contributions. All authors contributed to the conceptualization of this work. MLK initiated the code development, and HV wrote the current code and implemented the depth-averaging algorithm. HV conducted the numerical simulations and generated visual representations of the results. HV prepared the manuscript, with input and contributions from all co-authors.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Acknowledgements. The Digital Terrain Model and extent of the ice avalanche event were provided by Fondazione Montagna Sicura. We are grateful to Zheng Han and an anonymous referee for reviewing our manuscript and for their insightful comments.

References

Bartelt, P., Bieler, C., Bühler, Y., Christen, M., Deubelbeiss, Y., Graf, C., McArdell, B. W., Salz, M., and Schneider, M.: Manual RAMMS - Rapid Mass Movements Simulation, pp. 1–116, 2017.

Christen, M., Kowalski, J., and Bartelt, P.: RAMMS : Numerical Simulation of Dense Snow Avalanches in Three- Dimensional Terrain, Cold Regions Science and Technology, 63, 1–14, https://doi.org/10.1016/j.coldregions.2010.04.005, 2010.

Cicoira, A., Blatny, L., Li, X., Trottet, B., and Gaume, J.: Towards a Predictive Multi-Phase Model for Alpine Mass Movements and Process Cascades, Engineering Geology, 310, 106 866, https://doi.org/10.1016/j.enggeo.2022.106866, 2022.

Denlinger, R. P. and Iverson, R. M.: Granular Avalanches across Irregular Three-Dimensional Terrain: 1. Theory and Computation, Journal of Geophysical Research: Earth Surface, 109, https://doi.org/10.1029/2003JF000085, 2004.

Eglit, M., Yakubenko, A., and Zayko, J.: A Review of Russian Snow Avalanche Models—From Analytical Solutions to Novel 3D Models, Geosciences, 10, 77, https://doi.org/10.3390/geosciences10020077, 2020.

Fischer, J.-T., Kowalski, J., and Pudasaini, S. P.: Topographic Curvature Effects in Applied Avalanche Modeling, Cold Regions Science and Technology, 74–75, 21–30, https://doi.org/10.1016/j.coldregions.2012.01.005, 2012.

Franci, A., Cremonesi, M., Perego, U., Oñate, E., and Crosta, G.: 3D Simulation of Vajont Disaster. Part 2: Multi-failure Scenarios, Engineering Geology, 279, 105 856, https://doi.org/10.1016/j.enggeo.2020.105856, 2020.

Gaume, J., Gast, T., Teran, J., Van Herwijnen, A., and Jiang, C.: Dynamic Anticrack Propagation in Snow, Nature Communications, 9, 3047, https://doi.org/10.1038/s41467-018-05181-w, 2018.

Gaume, J., van Herwijnen, A., Gast, T., Teran, J., and Jiang, C.: Investigating the Release and Flow of Snow Avalanches at the Slope-Scale Using a Unified Model Based on the Material Point Method, Cold Regions Science and Technology, 168, 102 847, https://doi.org/10.1016/j.coldregions.2019.102847, 2019.

Han, Z., Su, B., Li, Y., Dou, J., Wang, W., and Zhao, L.: Modeling the Progressive Entrainment of Bed Sediment by Viscous Debris Flows Using the Three-Dimensional SC-HBP-SPH Method, Water Research, 182, 116 031, https://doi.org/10.1016/j.watres.2020.116031, 2020.

Hergarten, S.: MinVoellmy v1: A Lightweight Model for Simulating Rapid Mass Movements Based on a Modified Voellmy Rheology, Geoscientific Model Development, 17, 781–794, https://doi.org/10.5194/gmd-17-781-2024, 2024.

Hungr, O.: A Model for the Runout Analysis of Rapid Flow Slides, Debris Flows, and Avalanches, Canadian Geotechnical Journal, 32, 610–623, 1995.

Iverson, R. M. and George, D. L.: A Depth-Averaged Debris-Flow Model That Includes the Effects of Evolving Dilatancy. I. Physical Basis, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 470, https://doi.org/10.1098/rspa.2013.0819, 2014.

Koo, R. C., Kwan, J. S., Lam, C., Goodwin, G. R., Choi, C. E., Ng, C. W., Yiu, J., Ho, K. K., and Pun, W. K.: Back-Analysis of Geophysical Flows Using Three-Dimensional Runout Model, Canadian Geotechnical Journal, 55, 1081–1094, https://doi.org/10.1139/cgj-2016-0578, 2018.

Kwan, J. S., Koo, R. C., and Ng, C. W.: Landslide Mobility Analysis for Design of Multiple Debris-Resisting Barriers, Canadian Geotechnical Journal, 52, 1345–1359, https://doi.org/10.1139/cgj-2014-0152, 2015.

Kyburz, M. L., Sovilla, B., Bühler, Y., and Gaume, J.: Potential and Challenges of Depth-Resolved Three-Dimensional MPM Simulations: A Case Study of the 2019 'Salezer' Snow Avalanche in Davos, Annals of Glaciology, pp. 1–14, https://doi.org/10.1017/aog.2024.14, 2024.

Li, X., Sovilla, B., Jiang, C., and Gaume, J.: Three-Dimensional and Real-Scale Modeling of Flow Regimes in Dense Snow Avalanches, Landslides, 18, 3393–3406, https://doi.org/10.1007/s10346-021-01692-8, 2021.

Li, X., Sovilla, B., Ligneau, C., Jiang, C., and Gaume, J.: Different Erosion and Entrainment Mechanisms in Snow Avalanches, Mechanics Research Communications, 124, 103 914, https://doi.org/10.1016/j.mechrescom.2022.103914, 2022.

McDougall, S. and Hungr, O.: A Model for the Analysis of Rapid Landslide Motion across Three-Dimensional Terrain, Canadian Geotechnical Journal, 41, 1084–1097, https://doi.org/10.1139/t04-052, 2004.

Mergili, M., Schratz, K., Ostermann, A., and Fellin, W.: Physically-Based Modelling of Granular Flows with Open Source GIS, Natural Hazards and Earth System Sciences, 12, 187–200, https://doi.org/10.5194/nhess-12-187-2012, 2012.

Su, B., Li, Y., Han, Z., Ma, Y., Wang, W., Ruan, B., Guo, W., Xie, W., and Tan, S.: Topography-Based and Vectorized Algorithm for Extracting Physical Quantities in 3D-SPH Form and Its Application in Debris-Flow Entrainment Modeling, Engineering Geology, 340, 107 693, https://doi.org/10.1016/j.enggeo.2024.107693, 2024.

Tayyebi, S. M., Pastor, M., and Stickle, M. M.: Two-Phase SPH Numerical Study of Pore-Water Pressure Effect on Debris Flows Mobility: Yu Tung Debris Flow, Computers and Geotechnics, 132, 103 973, https://doi.org/10.1016/j.compgeo.2020.103973, 2021.

Tran, Q. A., Rogstad, A., Depina, I., Fernández, F., Alene, G. H., Grimstad, G., and Nordal, S.: 3D Large Deformation Modeling of the 2020 Gjerdrum Quick Clay Landslide, Canadian Geotechnical Journal, https://doi.org/10.1139/cgj-2024-0044, 2024.

Vicari, H. and Issler, D.: MoT-PSA: A Two-Layer Depth-Averaged Model for Simulation of Powder Snow Avalanches on 3-D Terrain, Annals of Glaciology, 65, e16, https://doi.org/10.1017/aog.2024.10, 2025.

Wirbel, A., Oesterle, F., Fischer, J.-T., Chambon, G., Faug, T., Gaume, J., Glaus, J., Hergarten, S., Issler, D., Jarosch, A., Jóhannesson, T., Martini, M., Mergili, M., Rauter, M., Robl, J., Rosatti, G., Spannring, P., Tollinger, C., Vicari, H., and Zugliani, D.: ISeeSnow - Initiating an Avalanche Simulation Tool Intercomparison, in: EGU General Assembly 2024, Copernicus Meetings, Vienna, Austria, https://doi.org/10.5194/egusphere-egu24-17750, 2024.

S1 Validation of the depth-averaging tool

The depth-averaging tool is validated on a simple geometrical configuration, consisting of a rigid block sliding on an inclined plane and then falling from a cliff. The plane is inclined at an angle $\theta_{\text{plane}} = 30^\circ$ and has an inclined length of $L_{\text{plane}} = 22$ m from the tail of the block. The cliff has height $H_{\text{cliff}} = 10$ m above a horizontal plane.

The block has height $H_{\text{block}} = 0.1$ m and length $L_{\text{block}} = 0.3$ m. The MPM simulation is initialized with the block vertically shifted by two grid sizes, $2 \cdot gs = 0.02$ m, to avoid the block sticking to the plane. Simulations are performed on a topography with cell size of $cs = 0.01$ m and results are exported with a resolution $cs_e = cs$.

Figure S1a shows the computed flow height h and thickness t (both projected normal to the topography) at $t_e = 0$, determined using Eqs. 7 and 8, respectively. The computed thickness t varies between 0 and 0.1 m, being constantly equal to 0.1 m where the top face of the block vertically projects within the basal face of the block. Similarly, the computed depth h varies between 0 and 0.12 m, this latter value being due to the vertical translation by $2 \cdot gs$ above the topography. Some oscillations are here observed, which may be due to the irregular sampling of material points. On the entire right-side of the block, the flow depth remains equal to ≈ 0.1 m, which is because the algorithm performs the particles search in a vertical direction, and the block is here overhanging (see discussion in Sec. 2.3.2). In summary, the computed initial flow heights are well calculated by the code.

As the block moves down along the inclined plane, and then falls from the cliff, the theoretical flow thickness can be calculated as:

$$\begin{cases} t = H_{\text{block}}, & \text{if } x < L_{\text{plane}} \cos\theta_{\text{plane}} \\ H_{\text{block}} \leq t \leq \sqrt{{L_{\text{block}}}^2 + {H_{\text{block}}}^2}, & \text{if } x \geq L_{\text{plane}} \cos\theta_{\text{plane}} \end{cases} \tag{S1}$$

where the last condition is because the block rotates during the free fall and might therefore be thicker than H_{block}. The blue line in Figure S1b (maximum value of t over all time steps) shows good agreement with Eq. S1; note the slight increase of $t_{\max}$ towards the end of the free fall, which is also when we observe the block rotating in the MPM simulation.

The theoretical flow depth can be calculated as:

$$\begin{cases} h \approx H_{\text{block}}, & \text{if } x < L_{\text{plane}} \cos\theta_{\text{plane}} \\ h \approx H_{\text{block}} + H_{\text{cliff}} - \tan\theta_{\text{plane}}(x - L_{\text{plane}} \cos\theta_{\text{plane}}) - \frac{1}{2} g \frac{(x - L_{\text{plane}} \cos\theta_{\text{plane}})^2}{v_{\text{term}}^2 \cos^2\theta_{\text{plane}}}, & \text{if } x \geq L_{\text{plane}} \cos\theta_{\text{plane}} \end{cases} \tag{S2}$$

where g is the acceleration due to gravity and v_{term} is the final speed of the block on the plane:

$$v_{\text{term}} = \sqrt{2 g L_{\text{plane}} \sin\theta_{\text{plane}}}. \tag{S3}$$

The red line in Figure S1b (maximum value of h over all time steps) shows good agreement with Eq. S2.

The theoretical slope-parallel speed can be calculated as:

$$\begin{cases} u_t = \sqrt{2 g x \tan\theta_{\text{plane}}}, & \text{if } x < L_{\text{plane}} \cos\theta_{\text{plane}} \\ u_t \approx v_{\text{term}} \cos\theta_{\text{plane}}, & \text{if } x \geq L_{\text{plane}} \cos\theta_{\text{plane}} \end{cases} \tag{S4}$$

The cyan line in Figure S1c (maximum value of u_t (Eq. 14) over all time steps) shows reasonable agreement with Eq. S4. However, the computed slope-parallel velocity slightly underestimates the theoretical one. We checked the MPM results by examining the particles' velocities at the end of the inclined plane, and the observed discrepancy can be attributed to numerical dissipations inherent in the APIC numerical scheme. Instead, the depth-averaging tool seems to export the correct results for the slope-parallel speed. Note that in the two cells located immediately before and after the cliff, the calculated value of u_t dips. This reduction is linked to the abrupt terrain transition, which influences the computation of the spatial gradients of the surface, as defined in Eq. 4a. As a result, the surface normals (Eq. 3) in these two cells are nearly horizontal. Consequently, while the normal slope velocity increases above 0 in these two cells (not visible in Figure S1d because it is out of scale), the slope-parallel velocity correspondingly experiences a decrease.

The theoretical slope-normal speed can be calculated as:

$$\begin{cases} u_n = 0, & \text{if } x < L_{\text{plane}} \cos\theta_{\text{plane}} \\ u_n \approx -v_{\text{term}} \sin\theta_{\text{plane}} - \frac{g(x - L_{\text{plane}} \cos\theta_{\text{plane}})}{v_{\text{term}} \cos\theta_{\text{plane}}}, & \text{if } x \geq L_{\text{plane}} \cos\theta_{\text{plane}} \end{cases} \tag{S5}$$

The magenta line in Figure S1d (minimum value of u_n (Eq. 13) over all time steps) shows good agreement with Eq. S5. Some discrepancy is observed in the free fall from the cliff, which is again due to dissipations in the MPM scheme.

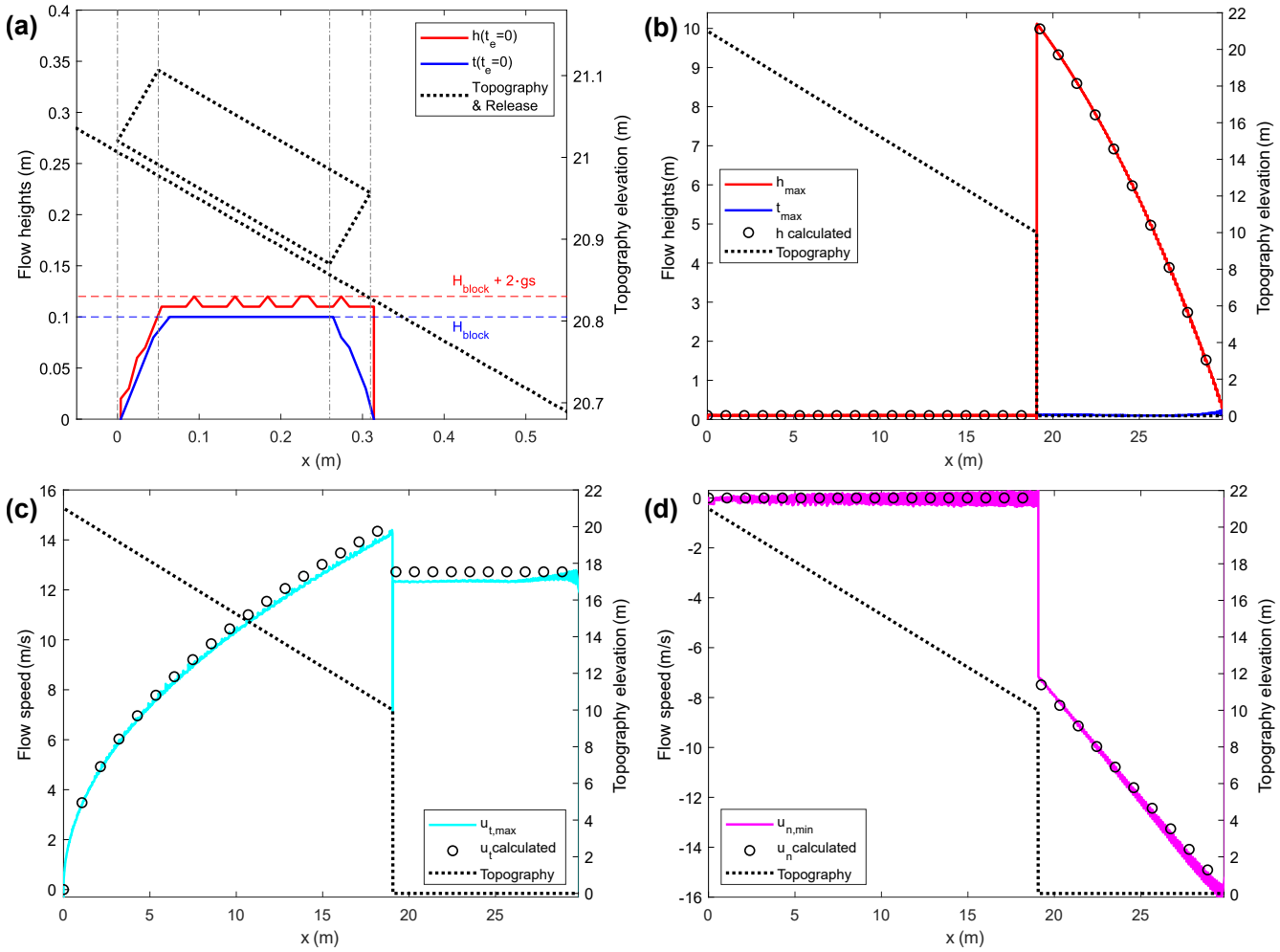

Figure S1. Validation of the depth-averaging code for a block sliding on an inclined plane and falling from a cliff. The computed flow variables are shown with coloured lines, while the calculated analytical solutions are shown with black circles. (a) Initial configuration and computed flow depth and thickness at $t = 0$. (b) Maximum flow depth and thickness over all time steps. (c) Maximum slope-parallel speed over all time steps. (d) Minimum slope-normal speed over all time steps.

S2 Material parameters for the ice avalanche simulation

Table S1 shows all the parameters for the ice avalanche simulations.

Table S1. Parameters for ice avalanche MPM simulation of the Whymper serac fall in 1998.

Parameter	Value
Initial bulk and particles' density, ρ_0	850 kg/m^3
Bulk modulus, K	6.7 MPa
Shear modulus, G	4.0 MPa
Initial friction angle, φ_i	40°
Residual friction angle, φ_r	20°
Initial cohesion, c_i	1 kPa
Basal friction, μ	0.65 (glacier) – 0.55 (bedrock)
Softening coefficient, ξ	10
Grid size, gs	1 m
Points per cell, n_g	6

S3 Flow detachment from the terrain

The depth-averaging algorithm enables us to capture flow detachment from the terrain, as observed in the slope-normal velocity component (Fig. 2h) and the difference between flow depth (measured from the terrain surface to the upper flow surface) and flow thickness (distance between the lowest and highest particles in a cell) (Fig. 2i). Figure 2i was computed using all particles (i.e., percentile set to P=100), including potentially scattered or isolated particles at the top of the flow.

To investigate the influence of filtering these outliers, we compare results obtained using different percentile values (P=20 to P=100), as shown in Fig. S2. Lower percentiles exclude the topmost particles, reducing both flow depth and flow thickness. In general, the trends along the profile remain similar regardless of the chosen percentile. In cells with sparse particle populations, percentile filtering has limited or no effect, as only a few particles are present (see Fig. S3). In zones where the flow is detached from the topography, using lower percentile values significantly reduces the flow thickness t, while the flow depth h remains comparatively stable, since it is still measured from the terrain to the upper percentile-bound particle. In contrast, in zones where the flow is attached to the terrain, changing the percentile value affects both flow depth and flow thickness in a similar way.

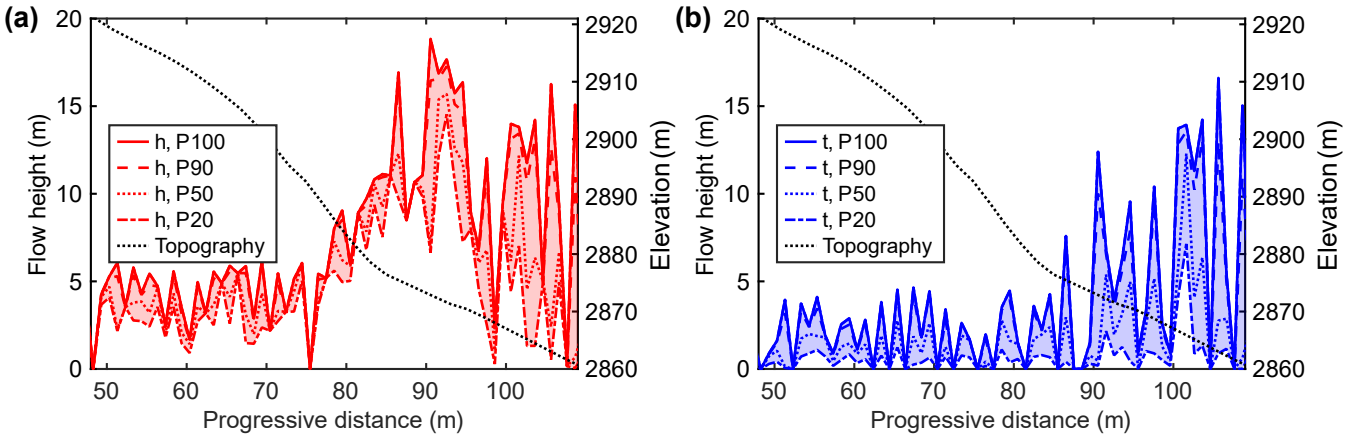

Figure S2. Flow depth (a) and flow thickness (b) exported with different percentile values. The shaded areas show the variability of the flow height metrics for the considered range (between P=20 and P=100).

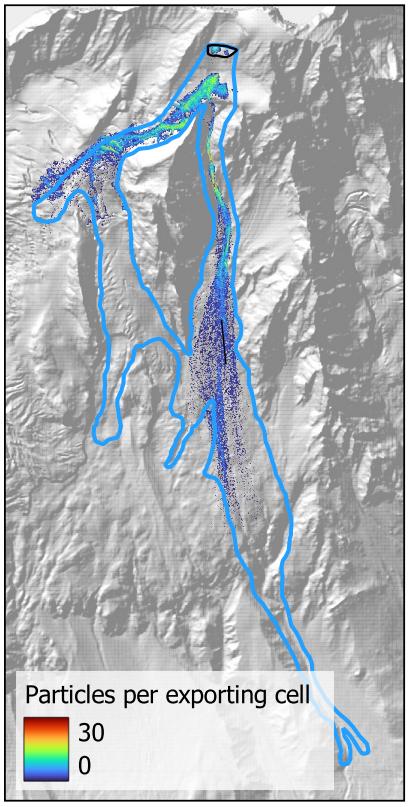

Figure S3. Number of particles per cell at $t_e = 50$ s.