AvaFrame com1DFA (version 1.3): a thickness-integrated computational avalanche module - Theory, numerics and testing

Matthias Tonnel[1], Anna Wirbel[1], Felix Oesterle[1], and Jan-Thomas Fischer[1]

[1]Dep. of Natural Hazards, BFW - Austrian Research Center for Forests, Innsbruck, Austria

Correspondence: Felix Oesterle (felix@avaframe.org)

Abstract.

Simulation tools are important to investigate and predict mobility and the destructive potential of gravitational mass flows (e.g. snow avalanches). AvaFrame - the open avalanche framework - offers well established computational modeling approaches, tools for data handling and analysis as well as ready-to-use modules for evaluation and testing. This paper presents the theoretical background, derivation and model verification for one of AvaFrame's core modules, the thickness-integrated computational model for avalanches with flow or mixed form of movement, named com1DFA. Particular emphasis within the description of the utilized numerical particle–grid method is given to the computation of spatial gradients and the accurate implementation of driving and resisting forces. The implemented method allows to provide a time-space criterion connecting the numerical particles, grid and time discretization. The convergence and robustness of the numerical implementation is checked with respect to the spatio-temporal evolution of the flow variables using tests with a known analytical solution. In addition we present a new test for verifying the accuracy of the numerical simulation in terms of runout (angle and distance). This test is derived from the total energy balance along the center-of-mass path of the avalanche. This manuscript, particularly in combination with the code availability (open-source code repository) and detailed online documentation provides a description of an extendable framework for modeling and verification of avalanche simulation tools.

1 Introduction

Simulation tools for gravitational mass flows - with a focus on snow avalanches in this manuscript - are in great demand for operational engineering applications, scientific model development and gain increasing attention in academic education. Each of these application requires different outputs. For operational engineering applications, the runout outcome for different scenarios is usually of highest interest. Scientific applications aim at better understanding the processes and will require outputs such as flow variable evolution. Existing tools for simulating snow avalanches cover a wide range of numerical implementations and vary from proprietary (e.g. Christen et al., 2010; Sampl and Zwinger, 2004; Zugliani and Rosatti, 2021; Li et al., 2021) to open source software (e.g. Hergarten and Robl, 2015; Mergili et al., 2017; Rauter et al., 2018). The latter ones are generally more focused towards scientific and academic issues whereas the first are more geared towards operational applications. AvaFrame - the open avalanche framework - strives to fill the gap between operations and scientific development combining over a decade of operational application (using e.g. SamosAT, see Sampl and Zwinger (2004); Fischer et al. (2014)) with an open source

scientific development environment. Using a modular structure AvaFrame adds in-depth testing and analysis modules to the core flow modules. Further modules provide interfaces for visualization and geodata handling for all kinds of existing and emerging simulation tools. It enables to combine, further develop and extend the different tools to best suit the users needs.

At their core avalanche simulation tools are based on a large variety of flow models, differing by their basic assumptions (what physical processes they include, degree of complexity), mathematical derivation and/or their numerical implementation. These range from Eulerian methods (Christen et al., 2010; Mergili et al., 2017; Rauter and Tuković, 2018; Zugliani and Rosatti, 2021) using spatially fixed meshes to Lagrangian methods (Sampl and Zwinger, 2004) where the mass is discretized. Some methods are combinations of Eulerian and Lagrangian approaches such as the Material Point method (Stomakhin et al., 2013). AvaFrame's **com1DFA** dense flow avalanche (DFA) module is based on a flow model that is derived from the thickness integration of the conservation equations of mass and momentum. Classic shallow water models, e.g. Saint-Venant, integrate in the direction of gravity, often called "depth" integration. Other approaches, such as Savage Hutter models (Savage and Hutter, 1989) integrate in the slope-normal direction but also call it "depth" integrated. To be more consistent with operational terminology, we propose to call the integration in the slope-normal direction "thickness" integration. In this way, we can highlight the special case of gravitational mass flows in steep terrain .

The resulting equations are solved using a mixed particle–grid method, in which mass is tracked using particles. Pressure gradients are computed using a smooth particle hydrodynamics (SPH) method adapted to steep terrain and flow thickness is computed on the grid (Sampl and Zwinger, 2004; Monaghan, 2005; Liu and Liu, 2010; Granig et al., 2016). To avoid nonphysical behavior at starting and stopping, **com1DFA** applies the method proposed in Mangeney-Castelnau et al. (2003), allowing for a friction balanced starting and stopping of the flow.

Verifying and validating the methods applied in our implementation is a challenging but crucial step, as it is for all simulation tools. Verification is done by comparing the numerical model results to an analytical solution (e.g. Zugliani and Rosatti, 2021; Rauter and Tuković, 2018). Validation can be tackled in different ways, either by comparing the model results to observations (e.g. Christen et al., 2010) or by comparing them to the results of already trusted numerical models. In this manuscript, the focus is on model verification, and the numerical model results are compared to tests with known (semi-) analytical solutions using two different approaches.

The flow variable tests (Hutter et al., 1993; Faccanoni and Mangeney, 2013) allow us to investigate the local spatio-temporal evolution of flow thickness and velocity. This enables us to verify the proper implementation of the pressure gradient and friction force computation, among others. In contrast, the energy line test is based on the investigation of the total energy balance (Körner, 1980). It focuses on the accuracy of the global kinetic energy (velocity) along the path and the corresponding center-of-mass runout. Thereby the proper implementation of stopping behavior, that is to say the proper balancing of driving and friction forces, can be assessed. It also provides a test where the runout, a quantity which is important for operational applications, is tested. We explore and explain the limitations of these two approaches.

The manuscript itself is structured as follows: Section 2 summarizes the underlying flow model, including fundamental assumptions and derivations of the thickness-integrated equations, building the foundation for the gravitational mass flow simulations. In Sect. 3 the spatial and temporal discretizations of the model equations are described using a particle–grid

approach. The implementation in the AvaFrame computational module **com1DFA** is outlined in Sect. 4. Model verification tests are presented in Sect. 5, targeting the correct implementation of the mathematical model as well as the convergence and robustness of the numerical model code. In addition to employing test cases with a known (semi-)analytical solution, a new energy line test is introduced in Sect. 5.2.

65   Besides the in–depth description within this manuscript, Oesterle et al. (2022) provide a combination of code and corresponding documentation. Users find more information according to their individual scientific, operational or educational focus and the reader is invited to contribute to the future development. It is important to note that this manuscript presents the latest development state of the **com1DFA** module. It differs slightly from the implementation of the **com1DFA** module used for operational purposes, which is described in the online AvaFrame documentation. For example, differences include improvements

70   of the SPH gradient computation method and how friction forces are taken into account.

2   Mathematical model: From 3D equations to thickness-integrated Lagrangian equations

In this section, the mathematical model and associated equations used to simulate DFA processes are presented. The derivation is based on the thickness integration of the three-dimensional Navier-Stokes equations, using a Lagrangian approach with a terrain-following coordinate system. The equations are simplified using the assumption of shallow flow on a mildly curved

75   topography, meaning flow thickness is considerably smaller than the length and width of the avalanche and it is considerably smaller than the topography curvature radius.

     We consider snow as the material, however the choice of material does not influence the derivation in the first place. We assume constant density ρ_0 and a flow on a surface $\mathcal{S}$, subjected to the gravity force and friction on the surface $\mathcal{S}$. If needed, additional processes such as entrainment or other external effects can be taken into account. These processes are included in

80   the following derivations but will not be considered for model verification (Sect. 5), as test cases with an analytical solution are only available for simplified conditions where entrainment or any additional forces are discarded. The mass conservation equation applied to a Lagrangian volume of material $V(t)$ reads:

$$\frac{\mathrm{d}}{\mathrm{d}t} \underbrace{\int_{V(t)} \rho_0 \, \mathrm{d}V}_{m(t)} = \rho_0 \frac{\mathrm{d}V(t)}{\mathrm{d}t} = \oint_{\partial V(t)} q^{\mathrm{ent}} \, \mathrm{d}A, \tag{1}$$

where the source term q^{ent} represents the snow entrainment rate (incoming mass flux). The momentum conservation equation

85   applied to same volume of material $V(t)$ reads:

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{V(t)} \rho_0 \mathbf{u} \, \mathrm{d}V = \underbrace{\oint_{\partial V(t)} \boldsymbol{\sigma} \cdot \mathbf{n} \, \mathrm{d}A}_{\text{surface forces}} + \underbrace{\int_{V(t)} \rho_0 \mathbf{g} \, \mathrm{d}V}_{\text{body force}} + \mathbf{F}^{\mathrm{ext}}, \tag{2}$$

where $\mathbf{u}$ is the fluid velocity and $\mathbf{g}$ the gravity acceleration. $\boldsymbol{\sigma} = -p\mathbb{1} + \mathbf{T}$ represents the stress tensor, where $\mathbb{1}$ is the identity tensor, p the pressure and $\mathbf{T}$ the deviatoric part of the stress tensor. $\mathbf{n}$ is the normal vector to $\partial V(t)$. F^{ext} represent additional

forces due to snow entrainment (force needed to break and compact the entrained snow) or due to added resistance (trees, obstacles, etc.). F^{ext} is assumed to be surface-parallel. Problem-specific assumptions are needed to solve the mass and momentum conservation equations (closure equation on the stress tensor 2.2). These are introduced in the following sections alongside a local coordinate system, boundary conditions and constitutive relations.

2.1 Natural Coordinate System (NCS) and thickness-integrated quantities

In order to solve the previously described equations, a local coordinate system is introduced. The avalanche flows on a surface S, a 2D manifold embedded in the 3D Euclidian space. Different approaches exist to define a coordinate system on this curved surface. Some define a tangent space on S based on the coordinate lines, which leads to an orthogonal but not orthonormal coordinate system for a curved surface S (e.g. Luca et al., 2016). Instead of this and because of the Lagrangian approach used here, we define the local tangent plane to S at any point using the velocity direction and the normal to the surface S at this position. This results in a time-dependent orthonormal coordinate system that is advected along with the flow, referred to as the Natural Coordinate System (NCS).

A control volume $V(t)$ is assumed to be a small truncated pyramid shape extending from the bottom surface $\mathcal{S}_b$ (lying on the topography $\mathcal{S}$) up to the free surface in the surface-normal direction $\mathbf{N^b}$ as illustrated in Fig. 1. With the assumption of moderately curved surfaces, this is close to being a prism shape since the normals of the lateral surfaces are almost parallel to the bottom. Note that the bottom surface $\mathcal{S}_b$ of area A^b has no predefined shape. The octagonal shape used in Fig. 1 is just one possible option.

The normal vector $\mathbf{N^b}$ to the bottom surface is pointing upwards whereas $\mathbf{n^b} = -\mathbf{N^b}$ is the bottom normal vector to the Lagrangian control volume (pointing out of the volume).

$$V(t) = \int\limits_{V(t)} \mathrm{d}V = \int\limits_{\mathcal{S}_b} \left(\int\limits_{b}^{s} \det(\mathbf{J})\,\mathrm{d}x_3 \right) \mathrm{d}A, \tag{3}$$

where $\mathbf{J}$ is the transformation matrix from the Cartesian coordinate system to the NCS. The NCS is an orthonormal coordinate system $(\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3})$ aligned with the bottom surface. $\mathbf{v_3} = \mathbf{N^b} = -\mathbf{n^b}$ is the normal vector to the bottom surface pointing upwards. $\mathbf{v_1}$ is pointing in the direction of the thickness-integrated fluid velocity $\overline{\mathbf{u}}$ (introduced below).

$$\mathbf{v_1} = \frac{\overline{\mathbf{u}}}{\|\overline{\mathbf{u}}\|}, \quad \mathbf{v_2} = \frac{\mathbf{v_3} \wedge \mathbf{v_1}}{\|\mathbf{v_3} \wedge \mathbf{v_1}\|}, \quad \mathbf{v_3} = \mathbf{N^b}. \tag{4}$$

In the case of shallow flow on moderately curved surfaces, $\det(\mathbf{J}) = (1 - \kappa_1 x_3)(1 - \kappa_2 x_3) \approx 1$. $\kappa_{\{1,2\}}$ represent the surface curvature in $\mathbf{v}_{\{1,2\}}$ directions and x_3 is the elevation from the bottom surface in the direction $\mathbf{N^b}$. This approximation is valid if the curvature radius is much larger than the flow thickness h. In this case, the control volume reads:

$$V(t) \approx \int\limits_{\mathcal{S}_b} \underbrace{\int\limits_{b}^{s} \mathrm{d}x_3}_{h(t)} \mathrm{d}A. \tag{5}$$
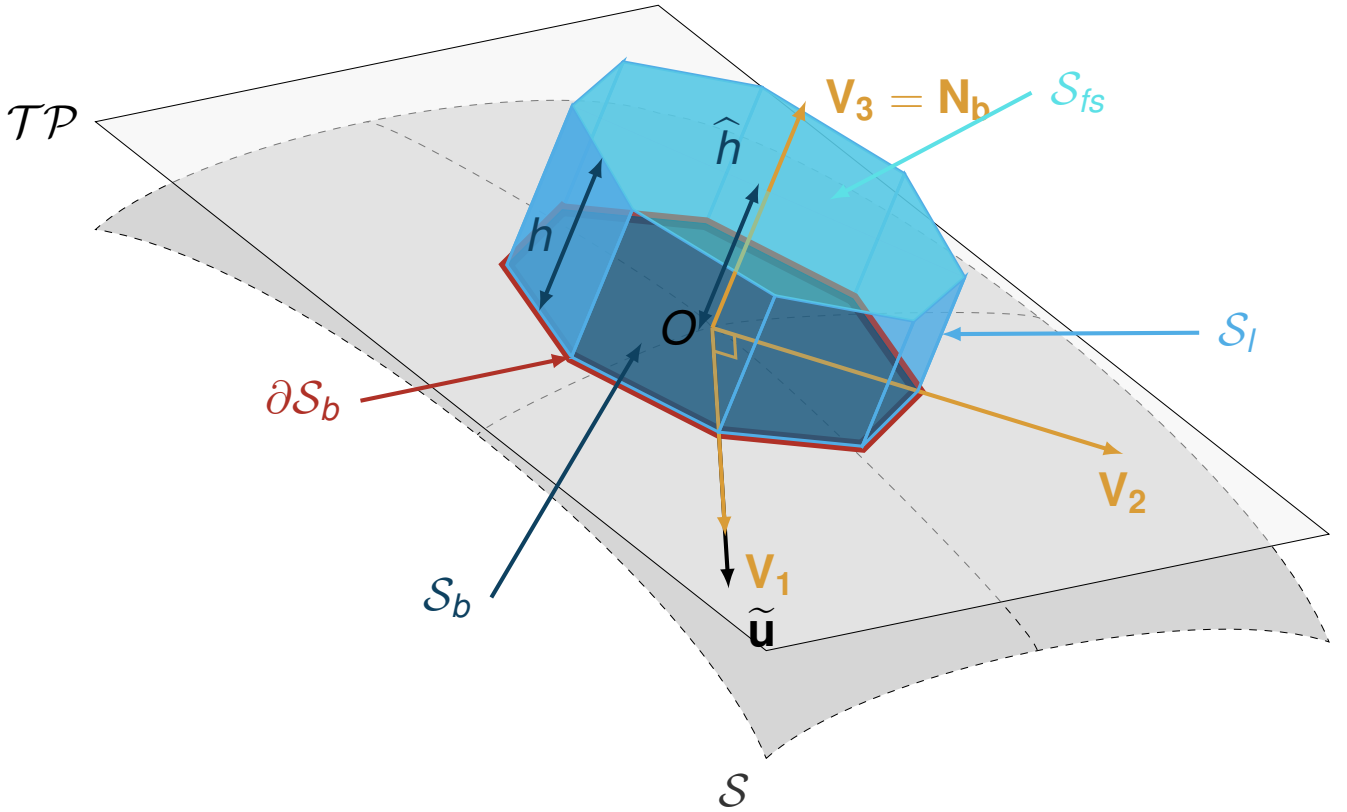
Figure 1. Example of a small Lagrange volume considered in the equations and corresponding notations. The gray surface (denoted $\mathcal{S}$) represents the bottom surface (topography) and $\mathcal{TP}$ represents the tangent plane to the surface at the the point O. The normal vector to $\mathcal{S}$ and $\mathcal{TP}$ in O is $\mathbf{v_3} = \mathbf{N^b}$. The control volume, represented in blue, has a basal surface $\mathcal{S}_b$ lying in $\mathcal{TP}$, a lateral surface $\mathcal{S}_l$ aligned with $\mathbf{N^b}$ and a free surface $\mathcal{S}_{fs}$.

The following volume (indicated by the superscript $\widetilde{\square}$), area (indicated by the superscript $\widehat{\square}$) and thickness (indicated by the superscript $\overline{\square}$) averaged quantities are introduced (where f is a scalar or vector function of $\Omega \subset \mathcal{R}^3$):

$$\widetilde{f} = \frac{1}{V(t)} \int\limits_{V(t)} f \, \mathrm{d}V$$

$$\widehat{f} = \frac{1}{A^b(t)} \int\limits_{\mathcal{S}_b} f \, \mathrm{d}A \qquad \text{and}$$

$$\overline{f} = \frac{1}{h(t)} \int\limits_{0}^{h(t)} f \, \mathrm{d}x_3$$

$$\widetilde{f}(x_3) = \frac{1}{A^b(\widehat{h} - x_3)} \int\limits_{\mathcal{S}_b} \left(\int\limits_{x_3}^{h(t)} f \det(\mathbf{J}) \, \mathrm{d}x_3 \right) \mathrm{d}A \approx \frac{1}{A^b(\widehat{h} - x_3)} \int\limits_{\mathcal{S}_b} \left(\int\limits_{x_3}^{h(t)} f \, \mathrm{d}x_3 \right) \mathrm{d}A$$

$$\overline{f}(x_3) = \frac{1}{(h - x_3)} \int\limits_{x_3}^{h(t)} f \, \mathrm{d}x_3. \tag{6}$$

120   Note that $\widetilde{f}(0) = \widetilde{f}$ and $\overline{f}(0) = \overline{f}$. When the control volume goes to 0, i.e. basal area goes to a point, $\widetilde{f} \xrightarrow{A^b \to 0} \overline{f}$ and $\widehat{f} \xrightarrow{A^b \to 0} f$. Also, note that we assume integration on a tangent plane being equivalent to integration on a small surface in

the manifold defined by the terrain (or 3D space in which the terrain is embedded). This is justified by the relative smallness of the basal particle surface compared to the curvature.

The NCS has some interesting properties that will be useful for projecting and solving the equations. Because of the orthogonality of this NCS, we have $\mathbf{v}_i \cdot \mathbf{v}_j = \delta_{ij}$, $\{i, j\} \in \{1, 2, 3\}^2$, which gives after time derivation:

$$\frac{\mathrm{d}(\mathbf{v}_i \cdot \mathbf{v}_j)}{\mathrm{d}t} = \mathbf{v}_i \cdot \frac{\mathrm{d}\mathbf{v}_j}{\mathrm{d}t} + \frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} \cdot \mathbf{v}_j = 0, \tag{7}$$

meaning that:

$$\begin{cases} \dfrac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} \cdot \mathbf{v}_i = 0 \implies \dfrac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} \perp \mathbf{v}_i \\ \dfrac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} \cdot \mathbf{v}_j = -\mathbf{v}_i \cdot \dfrac{\mathrm{d}\mathbf{v}_j}{\mathrm{d}t}, \quad i \neq j. \end{cases} \tag{8}$$

It is possible to express $\frac{\mathrm{d}\mathbf{v}_1}{\mathrm{d}t}$ in terms of $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ and using orthogonality of $\frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t}$ and $\mathbf{v}_i$:

$$\frac{\mathrm{d}\mathbf{v}_1}{\mathrm{d}t} = \alpha_i \mathbf{v}_i = \cancel{\alpha_1 \mathbf{v}_1}^{\,\mathbf{0}} + \alpha_2 \mathbf{v}_2 + \alpha_3 \mathbf{v}_3, \quad \alpha_i = \frac{\mathrm{d}\mathbf{v}_1}{\mathrm{d}t} \cdot \mathbf{v}_i. \tag{9}$$

The derivative of the thickness-integrated velocity decomposes to:

$$\frac{\mathrm{d}\overline{\mathbf{u}}}{\mathrm{d}t} = \frac{\mathrm{d}(\overline{u}_1 \mathbf{v}_1)}{\mathrm{d}t} = \overline{u}_1 \frac{\mathrm{d}\mathbf{v}_1}{\mathrm{d}t} + \frac{\mathrm{d}\overline{u}_1}{\mathrm{d}t} \mathbf{v}_1 = \overline{u}_1 (\alpha_2 \mathbf{v}_2 + \alpha_3 \mathbf{v}_3) + \frac{\mathrm{d}\overline{u}_1}{\mathrm{d}t} \mathbf{v}_1 \tag{10}$$

2.2 Boundary conditions

To complete the conservation Eqs. 1 and 2 the following boundary conditions at the bottom ($\mathcal{S}_b$) and free ($\mathcal{S}_{fs}$) surfaces are introduced. σ_s and σ_b represent the restriction of σ to the free surface $\mathcal{S}_{fs}$ and bottom surface $\mathcal{S}_b$, respectively:

– traction-free top surface:

$$\boldsymbol{\sigma_s} \cdot \mathbf{n_s} = \mathbf{0} \quad \text{on } \mathcal{S}_{fs} \tag{11}$$

– impenetrable bottom surface without detachment:

$$\mathbf{u^b} \cdot \mathbf{n^b} = \mathbf{0} \quad \text{on } \mathcal{S}_b \tag{12}$$

– bottom friction law:

$$\boldsymbol{\tau^b} = \boldsymbol{\sigma^b} \cdot \mathbf{n^b} - ((\boldsymbol{\sigma^b} \cdot \mathbf{n^b}) \cdot \mathbf{n^b}) \mathbf{n^b} = \mathbf{f}(\boldsymbol{\sigma^b}, \overline{\mathbf{u}}, h, \rho_0, t, \mathbf{x}) = -f(\boldsymbol{\sigma^b}, \overline{\mathbf{u}}, h, \rho_0, t, \mathbf{x}) \mathbf{v}_1 \quad \text{on } \mathcal{S}_b \tag{13}$$

2.3 Constitutive relation: friction force

To close the momentum equation, a constitutive equation describing the basal shear stress tensor $\boldsymbol{\tau^b}$ as a function of the avalanche flow state is required:

$$\boldsymbol{\tau^b} = \mathbf{f}(\boldsymbol{\sigma^b}, \overline{\mathbf{u}}, h, \rho_0, t, \mathbf{x}). \tag{14}$$

In the following, we employ a Mohr–Coulomb friction model that describes the friction interaction between two solids. The bottom shear stress reads:

$$\boldsymbol{\tau^b} = -\tan\delta\,\boldsymbol{\sigma^b} \cdot \mathbf{n^b}\,\frac{\overline{\mathbf{u}}}{\|\overline{\mathbf{u}}\|}, \tag{15}$$

where δ is the friction angle and $\mu = \tan\delta$ is the friction coefficient. The bottom shear stress linearly increases with the normal stress component p^b.

With Mohr–Coulomb friction an avalanche starts to flow if the slope inclination exceeds the friction angle δ. In the case of an infinite slope of constant inclination, the avalanche velocity would increase indefinitely.However, because of its relative simplicity, the Mohr–Coulomb friction model is convenient for deriving analytical solutions and testing numerical implementations. For a more detailed discussion of friction laws an their applicability we refer to Salm and Gubler (1985); Gubler (1987); Gauer (2014).

Different friction models accounting for the influence of flow velocity, flow thickness, etc. have been proposed. Three friction models are available in the **com1DFA** module. First, a Coulomb one, which is used in this paper, second a Voellmy friction model (Voellmy, 1955) and third, the samosAT friction model, which is the one used for hazard mapping by Austrian federal agencies (Sampl, 2007).

2.4 Expression of surface forces in the NCS

Taking advantage of the NCS and using the boundary conditions, it is possible to split the surface forces into bottom, lateral and free surface forces and perform further simplifications:

$$\oint_{\partial V(t)} \boldsymbol{\sigma} \cdot \mathbf{n}\,\mathrm{d}A = \int_{\mathcal{S}_b} \boldsymbol{\sigma^b} \cdot \mathbf{n^b}\,\mathrm{d}A + \int_{\mathcal{S}_{fs}} \boldsymbol{\sigma_s} \cdot \underbrace{\mathbf{n_s}}_{\mathbf{0}}\,\mathrm{d}A + \int_{\mathcal{S}_{lat}} \boldsymbol{\sigma} \cdot \mathbf{n}\,\mathrm{d}A = \underbrace{\int_{\mathcal{S}_b} \boldsymbol{\sigma^b} \cdot \mathbf{n^b}\,\mathrm{d}A}_{\text{bottom force}} + \underbrace{\oint_{\partial\mathcal{S}_b} \left(\int_0^h \boldsymbol{\sigma} \cdot \mathbf{n}\,\mathrm{d}x_3 \right)\,\mathrm{d}l}_{\text{lateral force}}. \tag{16}$$

Using the notations introduced in Sect. 2.1 and the decomposition of the stress tensor, the bottom force can be expressed as a surface-normal component and a surface-tangential one:

$$\int_{\mathcal{S}_b} \boldsymbol{\sigma^b} \cdot \mathbf{n^b}\,\mathrm{d}A = \int_{\mathcal{S}_b} (-p^b\mathbb{1} + \mathbf{T}) \cdot \mathbf{n^b}\,\mathrm{d}A = -\int_{\mathcal{S}_b} p^b \mathbf{n^b}\,\mathrm{d}A + \int_{\mathcal{S}_b} \mathbf{T} \cdot \mathbf{n^b}\,\mathrm{d}A = -\int_{\mathcal{S}_b} p^b \mathbf{n^b}\,\mathrm{d}A + \int_{\mathcal{S}_b} \boldsymbol{\tau^b}\,\mathrm{d}A = -A^b\widehat{p^b\mathbf{n^b}} + A^b\widehat{\boldsymbol{\tau^b}}, \tag{17}$$

where $\boldsymbol{\tau^b}$ is the basal friction term (introduced in Sect. 2.2). Applying Green's theorem, the lateral force reads:

$$\oint_{\partial\mathcal{S}_b} \left(\int_0^h \boldsymbol{\sigma} \cdot \mathbf{n}\,\mathrm{d}x_3 \right)\,\mathrm{d}l = \oint_{\partial\mathcal{S}_b} \left(\int_0^h (-p\mathbb{1} + \mathbf{T})\,\mathrm{d}x_3 \right) \cdot \mathbf{n}\,\mathrm{d}l = -\oint_{\partial\mathcal{S}_b} \left(\int_0^h p\,\mathrm{d}x_3 \right) \cdot \mathbf{n}\,\mathrm{d}l + \oint_{\partial\mathcal{S}_b} \left(\int_0^h \mathbf{T}\,\mathrm{d}x_3 \right) \cdot \mathbf{n}\,\mathrm{d}l$$

$$= -\oint_{\partial\mathcal{S}_b} h\overline{p}\mathbf{n}\,\mathrm{d}l + \oint_{\partial\mathcal{S}_b} h\overline{\mathbf{T}} \cdot \mathbf{n}\,\mathrm{d}l = -\int_{\mathcal{S}_b} \boldsymbol{\nabla}(h\overline{p})\,\mathrm{d}A + \int_{\mathcal{S}_b} \boldsymbol{\nabla} \cdot (h\overline{\mathbf{T}})\,\mathrm{d}A = -A^b\widehat{\boldsymbol{\nabla}(h\overline{p})} + A^b\widehat{\boldsymbol{\nabla} \cdot (h\overline{\mathbf{T}})}. \tag{18}$$

Equations 17 and 18 represent the thickness-integrated form of the surface forces and can now be used to write the thickness integrated momentum equation.

2.5 Thickness-integrated momentum equation

Using the definitions of average values given in Sect. 2.1 and the decomposition of the surface forces given by Eqs. 17 and 18, the momentum equation reads:

$$\rho_0 \frac{\mathrm{d}(V(t)\widetilde{\mathbf{u}})}{\mathrm{d}t} = \rho_0 V \frac{\mathrm{d}\widetilde{\mathbf{u}}}{\mathrm{d}t} + \rho_0 \widetilde{\mathbf{u}} \frac{\mathrm{d}V}{\mathrm{d}t} = \oint_{\partial V(t)} \boldsymbol{\sigma} \cdot \mathbf{n}\, \mathrm{d}A + \rho_0 V \mathbf{g} + \mathbf{F}^{\mathrm{ext}}, \tag{19}$$

175 which leads to:

$$\rho_0 V \frac{\mathrm{d}\widetilde{\mathbf{u}}}{\mathrm{d}t} = \underbrace{-A^b \widehat{p\mathbf{n^b}}}_{\substack{\text{bottom}\\\text{normal force}}} \underbrace{+A^b \widehat{\boldsymbol{\tau^b}}}_{\substack{\text{bottom}\\\text{shear force}}} \underbrace{-A^b \widehat{\boldsymbol{\nabla}(h\overline{p})}}_{\substack{\text{lateral}\\\text{pressure force}}} \underbrace{+A^b \overbrace{\boldsymbol{\nabla}\cdot h\mathbf{T}}^{O(\epsilon^2)}}_{\substack{\text{lateral}\\\text{shear force}}} + \rho_0 V \mathbf{g} + \mathbf{F}^{\mathrm{ext}} \underbrace{-\widetilde{\mathbf{u}} \oint_{\partial V(t)} q^{\mathrm{ent}}\, \mathrm{d}A}_{\substack{\text{speed loss due}\\\text{to entrainment}}}. \tag{20}$$

The lateral shear stress term is neglected because of its relative smallness in comparison to the other terms as shown by the dimensional analysis carried out in Gray and Edwards (2014). The mass conservation reads:

$$\rho_0 \frac{\mathrm{d}V}{\mathrm{d}t} = \oint_{\partial V(t)} q^{\mathrm{ent}}\, \mathrm{d}A, \tag{21}$$

180 Using the approximations from Sect. 2.1, the equation of motion becomes:

$$\rho_0 V \frac{\mathrm{d}\overline{\mathbf{u}}}{\mathrm{d}t} = -A^b p\mathbf{n^b} + A^b \boldsymbol{\tau^b} - A^b \boldsymbol{\nabla}(h\overline{p}) + \rho_0 V \mathbf{g} + \mathbf{F}^{\mathrm{ext}} - \overline{\mathbf{u}} \oint_{\partial V(t)} q^{\mathrm{ent}}\, \mathrm{d}A, \tag{22}$$

where all quantities are evaluated at the center of the basal area (point O in Fig. 1). This equation is projected in the normal direction $\mathbf{v_3} = \mathbf{N^b}$ to get the expression of the basal pressure p^b. The projection of this same equation on the tangential plane leads to the differential equations satisfied by $\overline{\mathbf{u}}$.

2.5.1 Pressure distribution, thickness-integrated pressure and pressure gradient

We can project the momentum equation (Eq. 22), using the volume between x_3 and the surface h, in the normal direction $(\mathbf{v_3} = \mathbf{N^b} = -\mathbf{n^b})$. Applying the properties of the NCS (Eq. 10) the surface-normal component of Eq. 22 reads:

$$
\begin{aligned}
\rho_0 V(x_3, t) \frac{\mathrm{d}\overline{\mathbf{u}}(x_3)}{\mathrm{d}t} \cdot \mathbf{v_3} =\ & \rho_0 A^b (h - x_3)\overline{u}_1(x_3) \frac{\mathrm{d}\mathbf{v_1}}{\mathrm{d}t} \cdot \mathbf{v_3} \\
=\ & -\rho_0 A^b (h - x_3)\overline{u}_1(x_3)\mathbf{v_1} \cdot \frac{\mathrm{d}\mathbf{v_3}}{\mathrm{d}t} = -\rho_0 A^b (h - x_3)\overline{\mathbf{u}}(x_3) \cdot \frac{\mathrm{d}\mathbf{N^b}}{\mathrm{d}t} \\
=\ & -A^b p\,\mathbf{n^b} \cdot \overbrace{\mathbf{N^b}}^{-1} + A^b \boldsymbol{\tau^b} \cdot \overbrace{\mathbf{N^b}}^{0} - A^b \boldsymbol{\nabla}\{(h - x_3)\overline{p}\} \cdot \mathbf{N^b} \\
& + \rho_0 V \mathbf{g} \cdot \overbrace{\mathbf{N^b}}^{g_{N^b}} + \mathbf{F}^{\mathrm{ext}} \cdot \overbrace{\mathbf{N^b}}^{0} - \overline{\mathbf{u}} \oint_{\partial V(t)} q^{\mathrm{ent}}\, \mathrm{d}A \cdot \overbrace{\mathbf{N^b}}^{0}.
\end{aligned} \tag{23}
$$

Neglecting the normal component of the pressure gradient gives the expression for pressure. Under the condition that $\overline{\mathbf{u}}_1(x_3)$ is independent of x_3, pressure follows a linear profile from the bottom surface to the free surface:

$$p(x_3) = \rho_0(h - x_3)\left\{-g_{N^b} - \overline{\mathbf{u}} \cdot \frac{\mathrm{d}\mathbf{N^b}}{\mathrm{d}t}\right\} \quad \text{and} \quad p(x_3 = 0) = p^b = \rho_0 h\left\{-g_{N^b} - \overline{\mathbf{u}} \cdot \frac{\mathrm{d}\mathbf{N^b}}{\mathrm{d}t}\right\}. \tag{24}$$

Note that the bottom pressure should always be positive. A negative pressure is nonphysical and means that the material is not in contact with the bottom surface anymore. This can happen in the case of large velocities on convex topography. If so, the material should be in a free-fall state until it gets back in contact with the topography. A description on how this is handled within the numerical implementation can be found in Sect. 4.3.

Using Eq. 24, it is possible to express the thickness-integrated pressure $\overline{p}$:

$$h\overline{p} = \int\limits_0^h p(x_3)\,\mathrm{d}x_3 = -\rho_0 \frac{h^2}{2}\left(g_{\mathbf{N^b}} + \overline{\mathbf{u}} \cdot \frac{\mathrm{d}\mathbf{N^b}}{\mathrm{d}t}\right) = -\rho_0 \frac{h^2}{2}\,g^{\mathrm{eff}}, \tag{25}$$

where g^{eff} is the effective normal acceleration acting on the volume, including the normal component of gravity and a curvature component. Because of the utilized Lagrangian approach, the curvature terms are expressed as the temporal derivative of the normal vector $\frac{\mathrm{d}\mathbf{N^b}}{\mathrm{d}t}$ which is directly linked to the particle positions. This does not require the computation of the $\kappa_{\{1,2\}}$ used in Eulerian approaches (Fischer et al., 2012).

The expression of the thickness-integrated pressure is used to derive the pressure gradient $\boldsymbol{\nabla}(h\overline{p})$. Assuming g^{eff} to be locally constant (otherwise g^{eff} would remain inside the gradient operator), leads to:

$$\boldsymbol{\nabla}(h\overline{p}) = -\rho_0\,g^{\mathrm{eff}}\,h\,\boldsymbol{\nabla}h. \tag{26}$$

2.5.2 Tangential momentum equation

Using the derived expression of the thickness-integrated pressure (Eq. 26), we project the momentum balance (Eq. 22) in the tangent plane, which leads to the following equation:

$$\rho_0 V\left(\frac{\mathrm{d}\overline{\mathbf{u}}}{\mathrm{d}t} - \left(\frac{\mathrm{d}\overline{\mathbf{u}}}{\mathrm{d}t} \cdot \mathbf{v}_3\right)\mathbf{v}_3\right) = A^b\boldsymbol{\tau^b} - \rho_0\,g^{\mathrm{eff}}\,hA^b\,\boldsymbol{\nabla}_s h + \rho_0 V\mathbf{g}_s + \mathbf{F}^{\mathrm{ext}} - \overline{\mathbf{u}}\oint\limits_{\partial V(t)} q^{\mathrm{ent}}\,\mathrm{d}A, \tag{27}$$

where $\boldsymbol{\nabla}_s = \boldsymbol{\nabla} - (\boldsymbol{\nabla} \cdot \mathbf{N^b})\mathbf{N^b}$ and $\mathbf{g_s} = \mathbf{g} - (\mathbf{g} \cdot \mathbf{N^b})\mathbf{N^b}$ are the tangential component of the gradient operator and of the gravity acceleration, respectively.

After replacing the velocity derivative component in the normal direction by the expression developed in Eq. 23, Eq. 27 reads:

$$\rho_0 V\frac{\mathrm{d}\overline{\mathbf{u}}}{\mathrm{d}t} = A^b\boldsymbol{\tau^b} - \rho_0\,g^{\mathrm{eff}}\,hA^b\,\boldsymbol{\nabla}_s h + \rho_0 V\mathbf{g}_s + \mathbf{F}^{\mathrm{ext}} - \overline{\mathbf{u}}\oint\limits_{\partial V(t)} q^{\mathrm{ent}}\,\mathrm{d}A - \underbrace{\rho_0 V\left(\overline{\mathbf{u}} \cdot \frac{\mathrm{d}\mathbf{v}_3}{\mathrm{d}t}\right)\mathbf{v}_3}_{\text{curvature acceleration}}. \tag{28}$$

The curvature acceleration is in the normal direction to the tangent plane in order to keep the flow on the surface.

3 Numerical method: particle–grid approach

In the previous section, the equation of motion was derived using a Lagrangian approach. In order to solve this set of equations numerically, we employ a mix of particle and grid approaches. We discretize the material into numerical particles and solve the equation of motion. The grid is used to compute several parameters that are required for the computations as for example surface-normal vectors and flow thickness. Combining both approaches allows us to best exploit the advantages of each. The particle approach is used to track the mass, compute the curvature terms and the gradient of the flow thickness as well as to update the particle positions. The grid is used to handle the topography information and compute the flow thickness and artificial viscosity. We found this to help with numerical stability and it is more efficient as it decreases the required amount of particles. A theoretical convergence criterion is described in the last section.

3.1 Interpolation between particle and grid values

Topography information is usually provided in a raster format which corresponds to a regular rectilinear mesh, from hereon referred to as grid. In order to get information on the surface elevation and normal vectors, the topography information needs to be interpolated at the particle locations, and this needs to be repeated at each time step since the particles are moving. Similarly, the particle properties such as mass or momentum, which translate into flow thickness and velocity, also need to be interpolated onto the grid. Grid velocity is required to compute the artificial viscosity term, ensuring numerical stability, see Sect. 4.2. Grid flow thickness is used to compute the particle flow thickness which is required for computing the frictional forces. The mesh being regular and rectilinear, we use bilinear interpolation for simplicity and efficiency. This also ensures the conservation of mass or momentum when interpolating from the particles to the grid and back. The properties of the grid and the interpolation method are detailed in the AvaFrame documentation (https://docs.avaframe.org/en/latest/DFAnumerics.html#interpolation).

3.2 The particle momentum equation

Discretizing the material into particles (particle quantities are denoted by the subscript k, e.g. $m_k = \rho_0 V_k$ is the mass of particle k) leads to the following continuity equation:

$$\frac{\mathrm{d}}{\mathrm{d}t} m_k = A_k^{\mathrm{ent}} q^{\mathrm{ent}}, \tag{29}$$

where A_k^{ent} is the interface area between the particle and the entrainable material while q_k^{ent} represents the flux of entrained mass.

By assuming that the Lagrangian control volume V can be represented by a particle, we can derive the particle momentum equation in the normal direction and in the tangent plane:

$$\begin{cases} p_k^b = \rho_0 \, h_k \, g_k^{\mathrm{eff}} \\ m_k \dfrac{\mathrm{d}\overline{\mathbf{u}}_k}{\mathrm{d}t} = A_k^b \boldsymbol{\tau}^b - m_k \, g_k^{\mathrm{eff}} \, \boldsymbol{\nabla}_s h + m_k \mathbf{g}_s + \mathbf{F}_k^{\mathrm{ext}} - \overline{\mathbf{u}}_k A_k^{\mathrm{ent}} q_k^{\mathrm{ent}} - m_k \left(\overline{\mathbf{u}}_k \cdot \dfrac{\mathrm{d}\mathbf{v}_{3,k}}{\mathrm{d}t} \right) \mathbf{v}_{3,k}. \end{cases} \tag{30}$$

In this equation, flow thickness gradient, basal friction and curvature acceleration terms need to be further developed and discretized.

3.3 Flow thickness and its gradient

3.3.1 Flow thickness gradient computation using SPH

To assess the flow thickness gradient, we employ an SPH method (Smoothed Particles Hydrodynamics Method Liu and Liu (2010)), where the gradient is directly derived from the particles and does not require any mesh. In contrast, a mesh method or an MPM (Material Point Method) would directly use a mesh formulation to approximate the gradient or interpolate the particle properties on an underlying mesh and then approximate the gradient of the flow thickness using a mesh formulation. There are two main advantages of using SPH: First is the computational efficiency, since only the particles of interest are computed without needing to compute the area of the whole terrain (as with Eulerian methods). Second, mass transfer is not required because it is handled by the particles directly, making the implementation easier (Granig et al., 2016).

In theory, an SPH method does not require any mesh to compute the gradient. However, applying this method requires finding neighbor particles. This process can be sped up with the help of an underlying grid; different neighbor search methods are presented in Ihmsen et al. (2014), a "uniform grid method" is used in this paper.

The SPH method is used to express a quantity (the flow thickness in our case) and its gradient at any particle location as a weighted sum of its neighbors properties. The principle of the method is well described in Liu and Liu (2010) and the basic formula reads:

$$f(x) \simeq \langle f(x) \rangle = \sum_l f_l A_l W(x, x_l)$$
$$\boldsymbol{\nabla} f(x) \simeq \langle \boldsymbol{\nabla} f(x) \rangle = -\sum_l f_l A_l \, \boldsymbol{\nabla} W(x, x_l), \tag{31}$$

where W represents the SPH-Kernel function (we employ the spiky kernel, see Eq. B2) and the subscript l enumerates the neighbouring particles. This kernel function is designed to satisfy the unity condition, be an approximation of the Dirac function and have a compact support domain (Liu and Liu, 2010).

This method is usually either used in a 3D space, in which particles move freely and where the weighting factor for the summation is the volume of the particle. Or on a 2D horizontal plane where the weighting factor for the summation is the area of the particle and the gradient is 2D. Here we want to compute the gradient of the flow thickness on a 2D surface (the topography) embedded in 3D space. The method used is analogous to the SPH gradient computation on the 2D horizontal plane but the gradient is 3D and tangent to the surface (colinear to the local tangent plane). The theoretical derivation in B2 shows that the SPH computation is equivalent to applying the 2D SPH method in the local tangent plane instead of in the horizontal plane. This leads to the following SPH expression of the flow thickness gradient:

$$\boldsymbol{\nabla} h_k \simeq -\sum_l h_l A_l \, \boldsymbol{\nabla} W(x_k, x_l) = -\sum_l \frac{m_l}{\rho} \, \boldsymbol{\nabla} W(x_k, x_l). \tag{32}$$

3.3.2 Flow thickness computation

The particle flow thickness is computed with the help of the grid. The mass of the particles is interpolated onto the grid using a bilinear interpolation method (described in Sect. 3.1). Then, dividing the mass at the grid cells by the area of the grid cells, while taking the slope of the cell into account, returns the flow thickness field on the grid. This is interpolated back to the particles, which leads to the particle flow thickness property.

We do not compute the flow thickness directly from the particle properties (mass and position) using an SPH method because it induced instabilities. Indeed, the cases where too few neighbors are found lead to very small flow thickness, which becomes an issue for flow thickness-dependent friction laws. Note that using such an SPH method would lead to a full particle method. But since the flow thickness is only used in some cases for the friction force computation, using the previously described grid method should not affect the computation significantly.

3.4 Friction force discretization

The Coulomb friction force term in Eq. 30 for a particle reads:

$$A_k^b \, \boldsymbol{\tau^b} = -A_k^b \, \tan\delta \, p_k^b \, \mathbf{v}_1 = -A_k^b \, \tan\delta \, \rho_0 \, h_k \, g_k^{\text{eff}} \, \mathbf{v}_1 = -m_k \, \tan\delta \, g_k^{\text{eff}} \, \mathbf{v}_1 = -\left\| \mathbf{F}_k^{\text{fric}} \right\|_{\max} \mathbf{v}_1. \tag{33}$$

This relation stands if the particle is moving. The starting and stopping processes satisfy a different equation and are handled differently in the numerical implementation (using the same equation would lead to non-physical behavior). This is described in more detail in Sect. 4.5.

3.5 Time discretization

The momentum equation is solved numerically in time using an Euler time scheme. The time derivative of any quantity f is approximated by:

$$\frac{\mathrm{d}f_k}{\mathrm{d}t} \approx \frac{f_k^{n+1} - f_k^n}{\Delta t}, \tag{34}$$

where Δt represents the time step and $f^n = f(t^n)$, $t^n = n\Delta t$. For the velocity this reads:

$$\frac{\mathrm{d}\overline{\mathbf{u}}_k}{\mathrm{d}t} \approx \frac{\overline{\mathbf{u}}_k^{n+1} - \overline{\mathbf{u}}_k^n}{\Delta t}. \tag{35}$$

The positions of the particles are updated using a centered Euler scheme:

$$\mathbf{x}_k^{n+1} = \mathbf{x}_k^n + \frac{\Delta t}{2m} \left(\overline{\mathbf{u}}_k^{n+1} + \overline{\mathbf{u}}_k^n \right). \tag{36}$$

The forces are taken into account in two subsequent steps as forces acting on the particles can be sorted into driving forces and friction forces. Friction forces act against the particle motion, only affecting the magnitude of the velocity. They can in no case become driving forces. This is why in a first step the velocity is updated with the driving forces before updating in a second step the velocity magnitude applying the friction force.

3.6 Convergence

We are looking for a criterion that relates the properties of the spatial and temporal discretization to ensure convergence of the numerical solution. Simply decreasing the time step and increasing the spatial resolution, by decreasing the grid cell size and kernel radius and increasing the number of particles, does not ensure convergence. Moussa and Vila (2000) also analyzed hyperbolic non-linear transport equations with a particle and SPH method and showed that the kernel radius size cannot be varied independently from the time step and number of particles. Indeed, they show that the numerical solution converges towards the solution of the equation under the following condition:

$$
\begin{cases}
r_{\text{part}} \to 0 \\
r_{\text{kernel}} \to 0 \qquad \text{and} \quad dt \leq C r_{\text{kernel}}, \\
\dfrac{r_{\text{part}}^{m}}{r_{\text{kernel}}^{m+1}} \to 0 \quad m = 2
\end{cases}
\tag{37}
$$

where r_{part} represents the "size" of a particle , r_{kernel} represents the SPH kernel radius, dt is the time step and C a constant. The conditions in Eq. 37 mean that both r_{part} (particle size) and r_{kernel} (SPH kernel radius) need to go to zero but also that the particle size needs to go faster to zero than the SPH kernel radius. Finally, the time step needs to go to zero at the same rate as r_{kernel}. The particle size can be expressed as a function of the SPH kernel radius:

$$
r_{\text{part}} = \left(\frac{A^{b}}{\pi} \right)^{1/2} = \left(\frac{A_{\text{kernel}}}{\pi n_{\text{ppk}}} \right)^{1/2} = \frac{r_{\text{kernel}}}{n_{\text{ppk}}^{1/2}},
\tag{38}
$$

where the particle basal area was assumed to be a circle. Note that this does not affect the results except adding a different shape factor in front of this expression. n_{ppk} is the number of particles per kernel support area and defines the density of the particles when initializing a simulation. Let n_{ppk} be defined by a reference number of particles per kernel support area $n_{\text{ppk}}^{0} > 0$, a reference kernel radius $r_{\text{kernel}}^{0} > 0$ and a real exponent α:

$$
n_{\text{ppk}} = n_{\text{ppk}}^{0} \left(\frac{r_{\text{kernel}}}{r_{\text{kernel}}^{0}} \right)^{\alpha}.
\tag{39}
$$

This leads to:

$$
r_{\text{part}} = \left(\frac{r_{\text{kernel}}^{0}{}^{\alpha}}{n_{\text{ppk}}^{0}} \right)^{1/2} r_{\text{kernel}}^{1-\alpha/2}.
\tag{40}
$$

Replacing r_{part} by the previous equation in Eq. 37 leads to the following condition:

$$
\frac{r_{\text{kernel}}^{0}{}^{\alpha}}{n_{\text{ppk}}^{0}} r_{\text{kernel}}^{-1-\alpha} \to 0.
\tag{41}
$$

This brings us to the following choice:

$$
\begin{cases}
dt = C_{\text{time}} r_{\text{kernel}} \\
n_{\text{ppk}} = n_{\text{ppk}}^{0} \left(\dfrac{r_{\text{kernel}}}{r_{\text{kernel}}^{0}} \right)^{\alpha},
\end{cases}
\tag{42}
$$

which satisfies the convergence criterion if:

$$\alpha < -1. \tag{43}$$

Note that this criterion leaves some freedom of choice for the exponent α and that there are no constraints on the reference kernel radius r_{kernel}^0 and reference number of particles per kernel radius n_{ppk}^0. Nevertheless, it seems appropriate to require a minimum number of particles per kernel radius so that enough neighbors are available to get a reasonable estimate of the gradient. These parameters should be adjusted according to the expected accuracy of the results and/or available compute power. Determining the optimal parameter values for α, r_{kernel}^0 and n_{ppk}^0, for example according to a user's needs in terms of accuracy and computational efficiency, requires a specific and detailed investigation of the considered case. In Sect. 5, we will explore model convergence using the condition Eq. 43 with different values of α.

4 Numerical implementation / Solver

In this section, the numerical implementation and algorithm of the **com1DFA** module are described. The following sections are organized following the workflow used in the **com1DFA** code, which is also illustrated in Fig. 2. First, the release mass is discretized into particles and the grid is initialized. As a result of the partial differential equations considered and the time discretization used, stability issues might arise. Hence, artificial viscosity is added in order to ensure the stability of the solution. As a next step, driving forces (including curvature effects) are accounted for. Friction forces are taken into account subsequently, in order to ensure proper starting and stopping behavior. Finally, a reprojection step is needed to ensure that particles lie on the topography and that particle velocities are tangent to the topography. For simplicity and because they are not considered in the verification tests in Sect. 5, entrainment and added resistance effects are not included in what follows. Additional information about entrainment or resistance forces is available in the theory section of the AvaFrame documentation.

4.1 Initialization

To start a simulation with **com1DFA**, input information about topography, material properties and initial conditions is required. Topography is described by a DEM (digital elevation model) using the ESRI ASCII format. It supplies a grid with (x,y) Cartesian locations of the lower left cell center, a cell size, number of cells in x- and y-direction and contains information about the elevation z. The material is characterized by its density and some friction properties. The initial condition is given by release areas, polygons describing the initial material location, and the release thickness, in our case measured in the surface-normal direction.

Then the material is discretized into particles. The field of normal-vectors to the surface is computed from the input DEM and the different grid fields are initialized. The details of the initialization process are given in the initialization section of the AvaFrame documentation.

4.2 Numerical stability

Because the lateral shear force term was removed when deriving the model equations (because of its relative smallness, Gray and Edwards, 2014), Eq. 22 is hyperbolic. Hyperbolic systems have the characteristic of carrying discontinuities or shocks which will cause numerical instabilities. They would fail to converge if for example an Euler forward in time scheme is used (LeVeque, 1990). Several methods exist to stabilize the numerical integration of a hyperbolic system of differential equations. All aim at adding some upwinding in the discretization scheme. Some methods tackle this problem by introducing some upwinding in the discretization of the derivatives (Harten et al., 1983; Harten and Hyman, 1983). Others introduce some artificial viscosity (as in Monaghan, 1992), which is also implemented in **com1DFA**. The following artificial viscosity force acting on particle k is added to stabilize the momentum equation:

$$
\begin{aligned}
\mathbf{F_k^{visc}} &= -\frac{1}{2}\rho_0 C_{Lat} A_k^{\text{Lat}} \|\mathbf{d\overline{u}}_k\|^2 \frac{\mathbf{d\overline{u}}_k}{\|\mathbf{d\overline{u}}_k\|} \\
&= -\frac{1}{2}\rho_0 C_{Lat} A_k^{\text{Lat}} \|\mathbf{d\overline{u}}_k\| \mathbf{d\overline{u}}_k,
\end{aligned}
\tag{44}
$$

where the velocity difference reads $\mathbf{d\overline{u}}_k = \mathbf{\overline{u}}_k - \widehat{\mathbf{\overline{u}}}_k$ ($\widehat{\mathbf{\overline{u}}}_k$ represents the averaged velocity of the neighbor particles and is practically the grid velocity interpolated at the particle position). C_{Lat} is a coefficient that controls the viscous force. It would be the equivalent of C_{Drag} in the case of the drag force. C_{Lat} is a numerical parameter that depends on the grid size.

In this expression, let $\mathbf{\overline{u}}_k^n$ be the velocity at the beginning of the time step and $\mathbf{\overline{u}}_k^{n+1\blacktriangle}$ be the velocity after adding the numerical viscosity (Fig. 2). In the norm term $\|\mathbf{d\overline{u}}_k\|$ the particle and grid velocity at the beginning of the time step are used. This ensures explicit time discretization with respect to the norm $\|\mathbf{d\overline{u}}_k\|$ and $\widehat{\mathbf{\overline{u}}}$ in $\mathbf{d\overline{u}}_k$. In contrast, an implicit formulation is used in $\mathbf{d\overline{u}}_k$ because the new value of the velocity is used there. The artificial viscosity force now reads:

$$
\mathbf{F_k^{visc}} = -\frac{1}{2}\rho_0 C_{Lat} A_k^{\text{Lat}} \left\| \mathbf{\overline{u}}_k^n - \widehat{\mathbf{\overline{u}}}_k^n \right\| \left(\mathbf{\overline{u}}_k^{n+1\blacktriangle} - \widehat{\mathbf{\overline{u}}}_k^n \right).
\tag{45}
$$

Updating the velocity then gives:

$$
\mathbf{\overline{u}}_k^{n+1\blacktriangle} = \frac{\mathbf{\overline{u}}_k^n - C_{vis}\widehat{\mathbf{\overline{u}}}_k^n}{1 + C_{vis}}
\tag{46}
$$

with

$$
C_{vis} = \frac{1}{2}\rho_0 C_{Lat} A_k^{\text{Lat}} \left\| \mathbf{\overline{u}}_k^n - \widehat{\mathbf{\overline{u}}}_k^n \right\| \frac{\Delta t}{m}.
\tag{47}
$$

This approach to stabilize the momentum equation (Eq. 30) is not optimal for different reasons. Firstly, it introduces a new coefficient C_{vis} which is not a physical quantity and will require to be calibrated. Secondly, it artificially adds a force that should be described physically. So it would be more interesting to take the physical force into account in the first place.

Potential solutions could be taking the physical shear force into account, using for example the µ-I rheology (Gray and Edwards, 2014; Baker et al., 2016). Another option would be to replace the artificial viscosity with a purely numerical artifact aiming to stabilize the equations such as an SPH version of the Lax–Friedrich scheme as presented in Ata and Soulaïmani (2005).

4.3 Curvature acceleration term

The last term of the particle momentum equation (Eq. 30) as well as the effective gravity g^{eff} are the final terms to be discretized before the numerical integration. In both of these terms, the remaining unknown is the curvature acceleration term $\overline{\mathbf{u}}_k \cdot \frac{\mathrm{d}\mathbf{v}_{3,k}}{\mathrm{d}t}$.

Using the forward Euler time discretization for the temporal derivative of the normal vector $\mathbf{v}_{3,k}$ gives:

$$\frac{\mathrm{d}\mathbf{v}_{3,k}}{\mathrm{d}t}\bigg|^n \approx \frac{\mathbf{v}_{3,k}^{n+1} - \mathbf{v}_{3,k}^n}{\Delta t}. \tag{48}$$

$\mathbf{v}_{3,k}^n$ is a known quantity, the normal vector of the bottom surface at $\mathbf{x}_k^n$ wich is interpolated from the grid normal vector values at the position of the particle k at time t^n. $\mathbf{v}_{3,k}^{n+1}$ is unknown since $\mathbf{x}_k^{n+1}$ is not known yet, hence we estimate $\mathbf{x}_k^{n+1}$ based on the position $\mathbf{x}_k^n$ and the velocity at t^n:

$$\mathbf{x}_k^{n+1} = \mathbf{x}_k^n + \Delta t \, \overline{\mathbf{u}}_k^{n+1\,\blacktriangle}. \tag{49}$$

This position at t^{n+1} is projected onto the topography and $\mathbf{v}_{3,k}^{n+1}$ can be interpolated from the grid normal vector values.

Note that the curvature acceleration term is needed to compute the bottom pressure (Eq. 24), which is used for the bottom friction computation and for the pressure gradient computation. The curvature acceleration term can lead to a negative value, which means detachment of the particles from the bottom surface. In **com1DFA**, surface detachment is not allowed and if pressure becomes negative, it is set back to zero forcing the material to remain in contact with the topography.

4.4 Account for driving forces

Adding the driving forces is done after adding the artificial viscosity as described in Fig. 2. The velocity is updated as follows ($\overline{\mathbf{u}}_k^{n+1\,\star}$ is the velocity after taking the driving force into account):

$$\overline{\mathbf{u}}_k^{n+1\,\star} = \overline{\mathbf{u}}_k^{n+1\,\blacktriangle} + \frac{\Delta t}{m_k}\mathbf{F}_k^{\text{drive}} = \overline{\mathbf{u}}_k^{n+1\,\blacktriangle} + \frac{\Delta t}{m_k}\left(-m_k g_k^{\text{eff}}\,\boldsymbol{\nabla}_s h + m_k \mathbf{g}_s - m_k\left(\overline{\mathbf{u}}_k^{n+1\,\blacktriangle}\cdot\frac{\mathrm{d}\mathbf{v}_{3,k}}{\mathrm{d}t}\bigg|^n\right)\mathbf{v}_{3,k}^n\right), \tag{50}$$

where the flow thickness gradient term is computed using the SPH formulation in 32.

4.5 Account for friction forces

The friction force related to the bottom shear force needs to be taken into account in the momentum equation and the velocity needs to be updated accordingly. Friction force acts against motion, hence it only affects the magnitude of the velocity and cannot be a driving force (Mangeney-Castelnau et al., 2003). Moreover, the friction force magnitude depends on the particle state, i.e. if it is flowing or at rest. If the velocity of the particle k reads $\overline{\mathbf{u}}_k^{n+1\,\star}$ after adding the driving forces, adding the friction force leads, depending on the sign of $\frac{m_k\left\|\overline{\mathbf{u}}_k^{n+1\,\star}\right\|}{\Delta t} - \left\|\mathbf{F}_k^{\text{fric}}\right\|_{max}$ (where $\left\|\mathbf{F}_k^{\text{fric}}\right\|_{max}$ depends on the chosen friction law introduced in Sect. 3.4), to:

- $\left\|\mathbf{F}^{\text{fric}}\right\| = \left\|\mathbf{F}_k^{\text{fric}}\right\|_{\text{max}}$ and $\overline{\mathbf{u}}_k^{n+1} = \overline{\mathbf{u}}_k^{n+1\,\star}\left(1 - \frac{\Delta t}{m_k}\frac{\left\|\mathbf{F}_k^{\text{fric}}\right\|_{\text{max}}}{\left\|\overline{\mathbf{u}}_k^{n+1\,\star}\right\|}\right)$, if $\frac{m_k\left\|\overline{\mathbf{u}}_k^{n+1\,\star}\right\|}{\Delta t} > \left\|\mathbf{F}_k^{\text{fric}}\right\|_{\text{max}}$

– $\left\|\mathbf{F}_k^{\mathrm{fric}}\right\| \leq \left\|\mathbf{F}_k^{\mathrm{fric}}\right\|_{\max}$ and the particle stops moving ($\overline{\mathbf{u}}_k^{n+1} = 0$) before the end of the time step, if $\frac{m_k\left\|\overline{\mathbf{u}}_k^{n+1\,\star}\right\|}{\Delta t} \leq \left\|\mathbf{F}_k^{\mathrm{fric}}\right\|_{\max}$.

410 This method prevents the friction force to become a driving force and nonphysically change the direction of the velocity. This would lead to oscillations of the particles instead of stopping. Adding the friction force following this approach (Mangeney-Castelnau et al., 2003) allows the particles to start and stop flowing properly.

4.6 Reprojection

The last term in Eq. 30 (accounting for the curvature effects) adds a non-tangential component allowing the new velocity
415 to lie in a different plane than the one from the previous time step. This enables the particles to follow the topography. But because the curvature term was only based on an estimation (see Sect. 4.3), the new particle position is not necessarily on the topography and the new velocity does not necessarily lie in the tangent plane at this new position. Furthermore, in case of a strong convex curvature and high velocities, the particles can theoretically be in a free-fall state (detachment), as mentioned in Sect. 2.5.1. **com1DFA** does not allow detachment of the particles and the particles are forced to stay on the topography. This
420 is a limitation of the model/method, which will lead to nonphysical behavior in special cases (material flowing over a cliff). In both of the previously mentioned cases, the particle positions are projected back onto the topography and the velocity direction is corrected to be tangential to the topography. The position reprojection is done using an iterative method that attempts to conserve the distance traveled by each particle between t^n and t^{n+1}. The velocity reprojection changes the direction of the velocity but its magnitude is conserved.

425 ## 5 Model Verification

In this section, the numerical implementation of the mathematical model is tested. We present different tests where, for specific conditions, a (semi-) analytical solution exists. The tests described here are implemented in the **ana1Tests** module from AvaFrame. In the first set of tests, the flow variable tests, we compare the temporal and spatial evolution of the flow thickness (h) and the thickness-integrated momentum flux ($h\overline{\mathbf{u}}$) of the **com1DFA** simulation results to a (semi-) analytical solution. With
430 these tests, we aim at verifying the numerical discretization and implementation of the solver as well as checking the validity of the convergence criterion described in Sect. 3.6.

In the second test, the energy line test, we investigate global variables such as mass-averaged position and kinetic energy that are derived from the DFA simulations. This test is based on energy conservation considerations for simplified topographies. This allows to verify the accuracy of the DFA simulations in terms of mass-averaged runout. All the tests presented and used
435 in what follows are implemented and available in AvaFrame (both data and helper functions). All results and figures can be reproduced using the code available on the AvaFrame github repository.
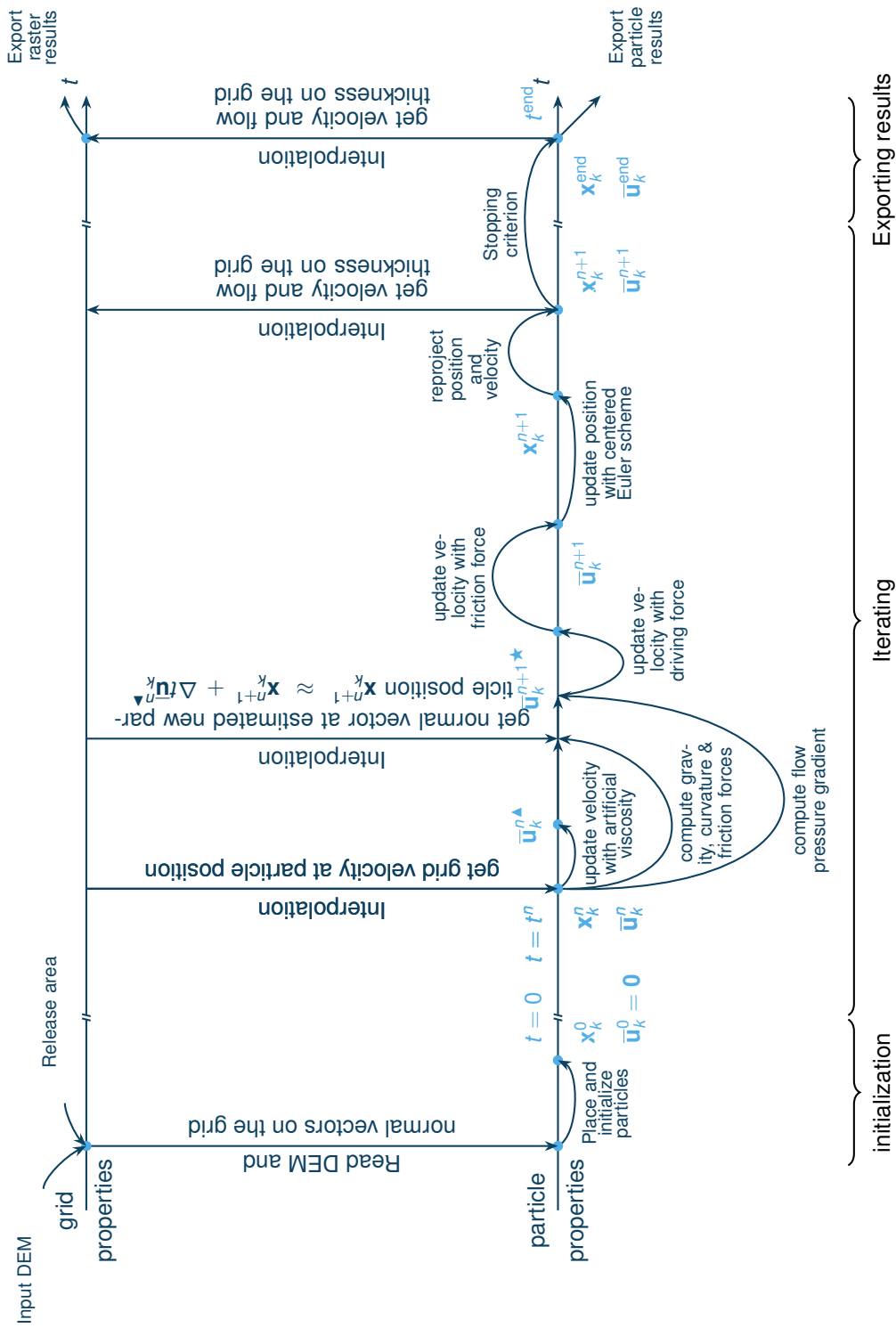
Figure 2. Dense Flow Avalanche solver (**com1DFA**) workflow.

5.1 Flow variable tests

Before performing the above mentioned similarity solution and dam break tests, it is necessary to describe the quantities that are compared and the measures that are used to assess the convergence, accuracy or precision of the numerical model. Both the flow thickness (h) and thickness-integrated flow momentum flux ($h\overline{\mathbf{u}}$) are used to compare the analytical solution to the simulation results. Two different deviation measures are used to quantify the deviation between a reference solution and the simulation result on a domain (one- or two-dimensional). The first is based on the $\mathcal{L}_{\max}$ norm (uniform norm), the second on the Euclidean norm ($\mathcal{L}_2$ norm). Let f_{num} be the numerical solution and f_{ref} the reference solution defined on a domain Ω. The local deviation is defined by $\mathcal{E}(x) = f_{\text{num}}(x) - f_{\text{ref}}(x)$ and the global deviation by:

- The uniform norm ($\mathcal{L}_{\max}$) measures the largest absolute value of the deviation $\mathcal{E}$ on Ω:

$$\mathcal{L}_{\max}(\mathcal{E}) = \sup_{x \in \Omega}(|\mathcal{E}(x)|)$$

. This norm is applied to one- or two-dimensional results. It can also be normalized by dividing the uniform norm of the deviation by the uniform norm of the reference. In this case we refer to the relative deviation:

$$\mathcal{R}\mathcal{L}_{\max} = \frac{\mathcal{L}_{\max}(\mathcal{E})}{\mathcal{L}_{\max}(f_{\text{ref}})}$$

.

- The Euclidean norm ($\mathcal{L}_2$ norm) gives an overall measure of the deviations

$$\mathcal{L}_2(\mathcal{E}) = \int_{x \in \Omega} ||\mathcal{E}(x)||^2 \, dx$$

. It is useful to normalize the norm of the deviation either by dividing with the norm of the reference solution:

$$\mathcal{R}\mathcal{L}_2 = \frac{\mathcal{L}_2(\mathcal{E})}{\mathcal{L}_2(f_{\text{ref}})}$$

or by the measure of the interval ($\mathcal{L}_2(1) = \int_{x \in \Omega} dx$):

$$\mathcal{R}\mathcal{S}\mathcal{L}_2 = \sqrt{\frac{\mathcal{L}_2(\mathcal{E})}{\mathcal{L}_2(1)}}$$

. The first normalization approach will give a relative deviation in % whereas the second will give an average deviation of f on Ω.

5.1.1 Similarity solution test

The similarity solution is one of the few cases where a semi-analytic solution is available for solving the thickness-integrated equations. This makes it very useful for validating the implementation of dense flow avalanche numerical methods (here **com1DFA**). In this problem, we consider an avalanche governed by a dry-friction law (Coulomb friction) flowing down an

inclined plane. The release mass is initially distributed in an ellipse (diameters of length L_x and L_y) with a parabolic thickness shape (H in the middle). This mass is released at $t = 0$ and flows down the inclined plane, as illustrated in Fig. 3 (a) for the initial time step and at some later time t. This semi-analytic solution can be derived under the following assumptions: First, 455 the ratio $\epsilon = H/L$ of flow thickness over spatial extent in flow and cross-flow direction should be smaller than 1. Second, the slope angle must be larger than the bed friction angle. Third, the solution is assumed to retain the symmetry properties of the initial configuration relative to the moving center of mass. The full description of the conditions and assumptions as well as the derivation of the solution is presented in detail by Hutter et al. (1993). The term semi-analytic is used here because the method enables to transform the PDE (partial differential equation) of the problem into an ODE (ordinary differential equation) using 460 a similarity analysis method. However, solving the ODE still requires a numerical integration but this one is more accurate (when conducted properly, here using an explicit Runge-Kutta method of fourth order) and requires less computational power than solving the PDE.

This test is implemented in the **ana1Tests** module of AvaFrame which offers functions to compute the semi-analytic solution, to compare it to the output from the DFA computational module and to visualize the results.
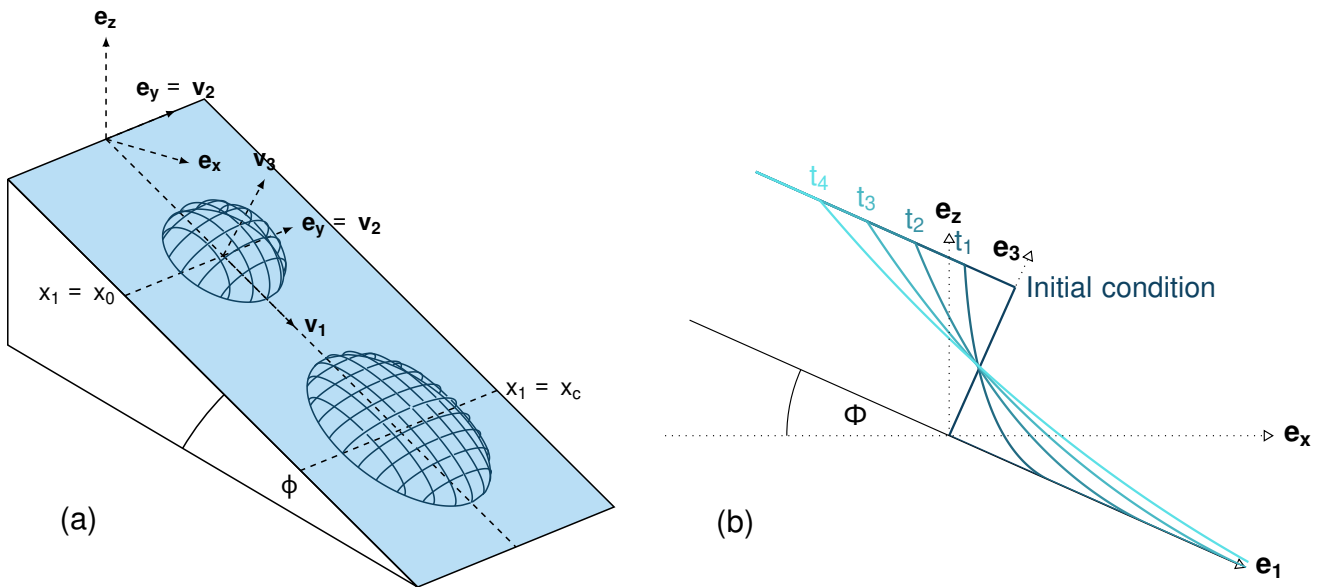
Figure 3. Flow variable test setups: (a) Similarity solution: 3D view of the material on the inclined plane at $t = 0$ (center located in $x_1 = x_0$) and at a general time t (center located in $x_1 = x_c$). The footprint of the material on the inclined plane at the initial time is an ellipse of diameters L_x and L_y, the thickness follows a parabolic distribution with the maximum thickness of H in the middle of the ellipse and 0 on the edge. (b) Dam break: 2D view of the material on the inclined plane (profile in flow direction) at $t = 0$ and for different $t > 0$.

test	sphKernelRadius	nPPK0	aPPK	cMax
similarity solution	$\{\mathbf{10}, \mathbf{8}, \mathbf{6}, 5, 4, 3\}$	$\{\mathbf{15}, 20, 30, 40\}$	$\{\mathbf{0}, -\mathbf{0.5}, -\mathbf{1}, -1.5, -\mathbf{2}, -2.5, -\mathbf{3}\}$	$\{0.04, \mathbf{0.02}, 0.01, 0.005\}$
dam break	$\{\mathbf{10}, \mathbf{8}, \mathbf{6}, 5, 4, 3\}$	$\{\mathbf{15}, 20, 30\}$	$\{\mathbf{0}, -\mathbf{0.5}, -\mathbf{1}, -1.5, -\mathbf{2}, -2.5, -\mathbf{3}\}$	$\{0.04, \mathbf{0.02}, 0.01, 0.005\}$

Table 1. Parameter variation used to study convergence of the DFA simulation solution for both similarity solution and dam break test (in bold the parameters used for the figures presented in this manuscript).

5.1.2 Dam break test

The dam break test is the second test for which an analytical solution of the thickness-integrated equations is known. In this test, we also consider an avalanche governed by a dry-friction law (Coulomb material), released from rest on an inclined plane (see Fig. 3 (b)). In the case of a thickness-integrated model as derived by Savage and Hutter (e.g. in Hutter et al., 1993), an analytical solution exists under the assumption of shallowness of the flow. Furthermore the friction angle has to be smaller than the slope of the inclined plane. This test, in contrast to the similarity solution test focuses on the very early stages of the flow and not on the evolution over time and lateral spreading. The derivation of the dam break solution is described in Faccanoni and Mangeney (2013) and corresponds to a Riemann problem. It has the following initial conditions:

$$(h, \mathbf{u})(x, t = 0) = \begin{cases} (h_0, \mathbf{0}), & \text{if } x \leq 0 \\ (0\ , \mathbf{0}), & \text{if } x > 0. \end{cases} \tag{51}$$

The dam break assumes an invariance in the y direction which is achieved using a wide enough domain in the y direction so that lateral effects can be neglected.

5.1.3 Results

DFA simulations are computed using the **com1DFA** module in AvaFrame varying the different numerical parameters listed in Tab. 1. The range of α values (called "aPPK" in the code) is determined by the convergence criterion (Eq. 43). The SPH kernel radius r_{kernel} (called "sphKernelRadius" in the code) is varied around 5m which is the raster cell size value currently used for operational hazard mapping in Austria. The tested values of n_{ppk0} (called "nPPK0" in the code) and C_{time} (called "cMax" in the code) are listed in Tab. 1. A large n_{ppk0} and small C_{time} lead to very long computation time which makes it unrealistic and impractical to use. Instructions on how to reproduce the results presented below are provided in the supplementary material.

For both of the tests, the numerical schemes to apply friction and the method used to compute the SPH gradient are crucial to capture a proper starting and stopping behavior of the flow. Some intermediate developments showed that adding the friction with different methods as the one presented here and computing the SPH gradient without taking the slope inclination into account leads to unsatisfactory results. This is why the friction force is added as described in Sect. 4.5 and the SPH force is computed as described in Appendix B2. In what follows, artificial viscosity is added (in the following results the $C_{\text{Lat}} = 10$) in the similarity solution test, which stabilizes the solution without degrading the match with the semi-analytic solution. For the

dam break test, adding artificial viscosity has a negative impact on the solution and the following results were produced with
no artificial viscosity ($C_{\text{Lat}} = 0$).

Fig. 4 shows an example where a DFA simulation is compared to the semi-analytical solution of the similarity solution test case. The chosen parameters for this example are $r_{\text{kernel}} = 3$m, $\alpha = -3$, $n_{ppk}^0 = 15$ and $C_{\text{time}} = 0.02$s and correspond to the most accurate of the simulations presented here. The reader can find the results of the similarity solution test with the standard parameters in the supplementary material. The top panels show the flow thickness and momentum profiles in and across flow direction after 20 seconds of flow. The bottom right panel shows the evolution of the $\mathcal{RL}_2$ and $\mathcal{RL}_{\text{max}}$ deviation with time. The deviation at the initial time step ($t = 0s$) is rather high (this is related to the random process to initialize the particles in the simulation) and then quickly decreases after a few seconds of simulation due to the reorganization of the particles. The deviation then increases again as the numerical inaccuracies accumulate. When varying the numerical parameters in the DFA simulations (according to Eq. 42), the computed $\mathcal{L}_2$ deviations between DFA results and the semi-analytical solution decrease (see Fig. 5). In Fig. 6, the comparison between a DFA simulation and the analytical solution of the dam break test is shown for flow thickness, flow velocity and momentum at t=15s (upper panel). The lower left panel shows a top view of the flow colored by flow thickness. This panel also shows the domain on which the deviation between analytical and numerical solution is computed. The lower right panel shows the relative deviations $\mathcal{RL}_2$ and $\mathcal{RL}_{\text{max}}$ on flow thickness and momentum. The same behavior as for the similarity solution test is observed regarding the time evolution of the deviation. Computation was done with $r_{\text{kernel}} = 3$m, $\alpha = -3$, $n_{ppk}^0 = 15$ and $C_{\text{time}} = 0.02$s. The reader can find the results of the similarity solution test with the standard parameters in the supplementary material.

Results from both the similarity solution tests and the dam break test validate the convergence criterion from Moussa and Vila (2000). Indeed, with an α exponent smaller than -1, decreasing the SPH kernel radius and varying the other parameters according to Eq. 42 leads to a decrease in the deviation, whereas for larger exponents, $\alpha = -0.5$ for both tests or $\alpha = 0$ for the dam break test (Fig. 7), the deviation decreases only slightly or even increases. Moreover, it is observed (not shown in the figure) that decreasing the time step (decreasing the C_{max} parameter) with all other parameters fixed leads to a decreasing deviation. Finally, for these two specific cases, DFA simulation results converge towards the semi-analytical or analytical solution.

5.2 Energy and runout testing

The Energy line test compares the results of the **com1DFA** simulation to a geometrical solution derived from the total energy of the system. Solely considering Coulomb friction, this solution is motivated by the first principle of energy conservation along a simplified topography. In this case, the friction force only depends on the slope angle. The analytical runout is the intersection of the path profile with the geometrical line (α line) defined by the friction angle α. From the geometrical line it is also possible to extract information about the flow mass-averaged velocity at any time or position along the path profile (see example in Fig. 9). For the detailed theory of this test please refer to Appendix A.
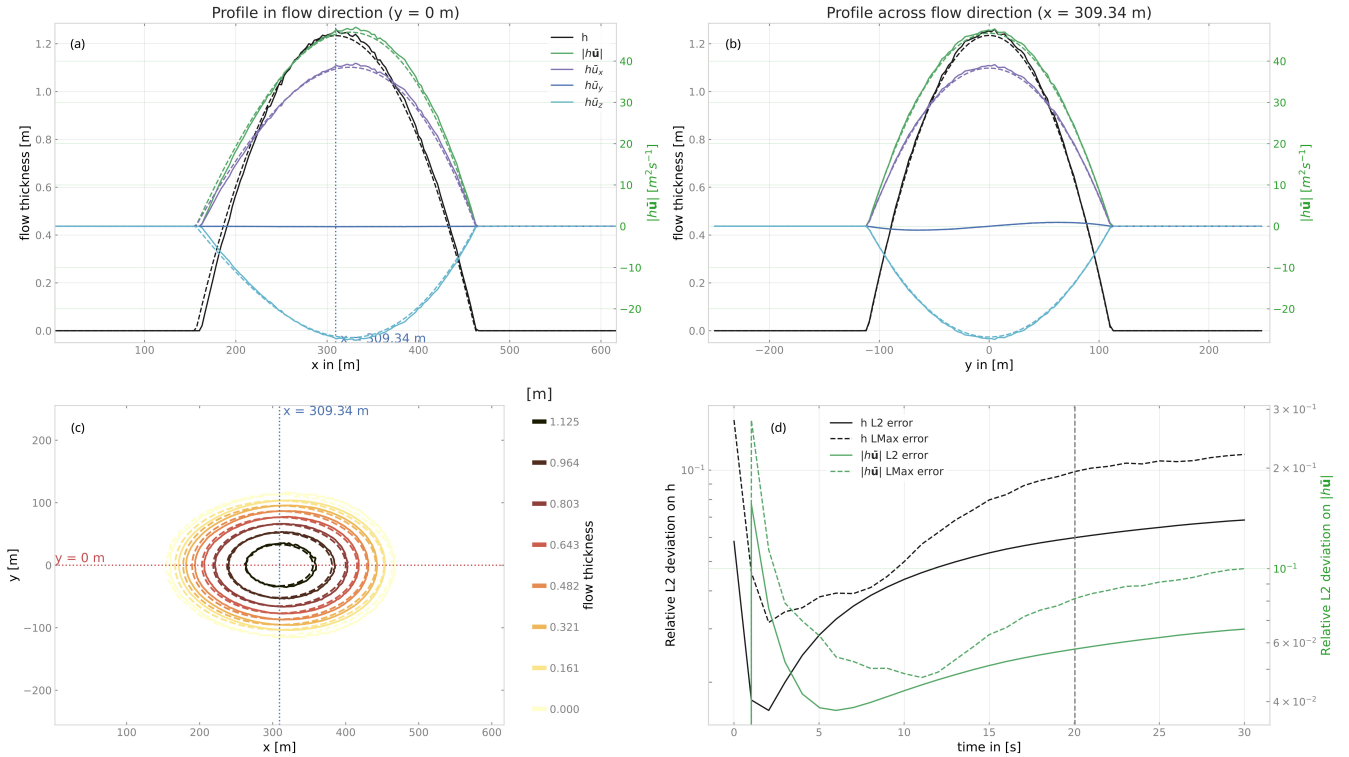
Figure 4. Comparison of the analytical (dashed lines) and numerical (solid lines) solution for the similarity solution test case at $t = 20.04s$. Panel (a) shows the profile in the flow direction (along the x axis) whereas (b) shows the profiles across flow direction (along the y axis). Panel (c) provides a top view of the flow thickness (flow thickness contour lines). Panel (d) shows the time evolution of the deviation (both $\mathcal{RL}_{\max}$ and $\mathcal{RL}_2$) on flow thickness h and momentum $\|h\overline{\mathbf{u}}\|$ between analytic and numerical solution ($r_{\text{kernel}} = 3$m, $\alpha = -3$, $n^0_{ppk} = 15$ and $C_{\text{time}} = 0.02$s). The vertical dashed line in (d) marks the time at which data in panels (a), (b) and (c) are shown.

In this test, we use the α line to evaluate the DFA simulation. Computing the mass-averaged path profile for the particles (each particle corresponding to a material point) in the simulation and comparing it to the α line allows us to compute four error indicators. Fig. A1 illustrates the concept.

The first three are related to the analytical runout point defined by the intersection between the α line and the mass-averaged path profile. The last one is related to the velocity:

- The horizontal distance between the analytical runout point and the end of the path profile defines the $\epsilon_s = \overline{s}_\gamma - \overline{s}_\alpha$ error in meters.

- The vertical distance between the analytical runout point and the end of the path profile defines the $\epsilon_z = \overline{z}_\gamma - \overline{z}_\alpha$ error in meters.

525
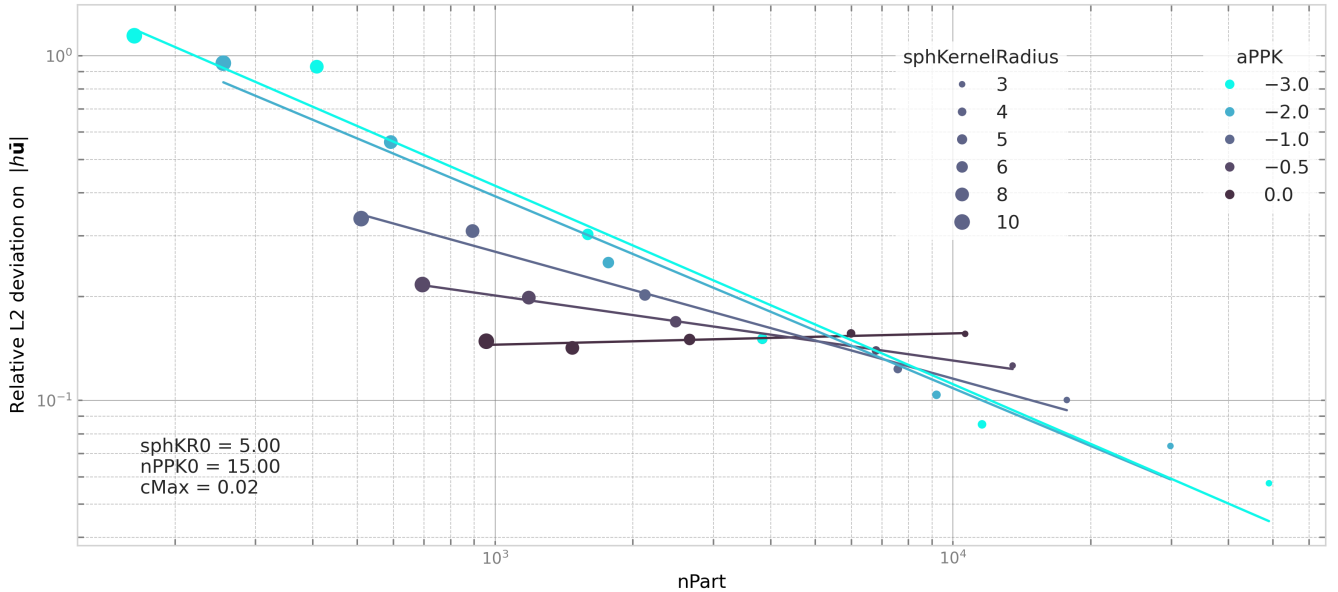
Figure 5. $\mathcal{RL}_2$ deviation on flow momentum in the similarity solution test case for different α exponents and SPH kernel radius r_{ppk}. Other parameters are kept fixed (reference kernel radius r_{ppk}^0, time constant C_{time} and reference number of particles per kernel radius n_{ppk}^0). The points corresponding to simulations with $r_{ppk} = 5\,\mathrm{m}$ are all overlapping since they are all identical. The colored lines are added to give an idea of the convergence speed trend associated to each α scenario. One can observe that decrease in deviation is stronger for lower α exponents and that no or little decrease is observed for $\alpha = 0$ or $\alpha = -0.5$.

- The runout angle difference between the α line angle and the DFA simulation runout line defines the $\epsilon_\alpha = \gamma - \alpha$ runout angle error.

- The Root Mean Square Error (RMSE) between the α line and the DFA simulation energy points defines an error on the velocity altitude $\overline{\frac{v^2}{2g}}$.

5.2.1 Limitations and applicability

It is essential to state where the assumptions of this test hold. One of the important hypotheses when developing the energy solution is the material point flowing in the steepest slope direction, i.e. where $dl = \frac{ds}{\cos\theta}$. If this hypothesis fails, as illustrated in Fig. 8, it is not possible to derive the analytical energy solution. In the 3D case, the distance vector $\mathbf{dl}$ traveled by the particles reads $dl = \frac{ds}{\cos\gamma}$, where γ is the angle between the $\mathbf{dl}$ vector and the horizontal plane which can differ from the slope angle θ ($\gamma \leq \theta$). Here the energy solution is not the solution to the problem and hence cannot be used as reference. In this case, it would not be possible to distinguish what deviation is caused by the numerical error or because of the hypothesis being violated.
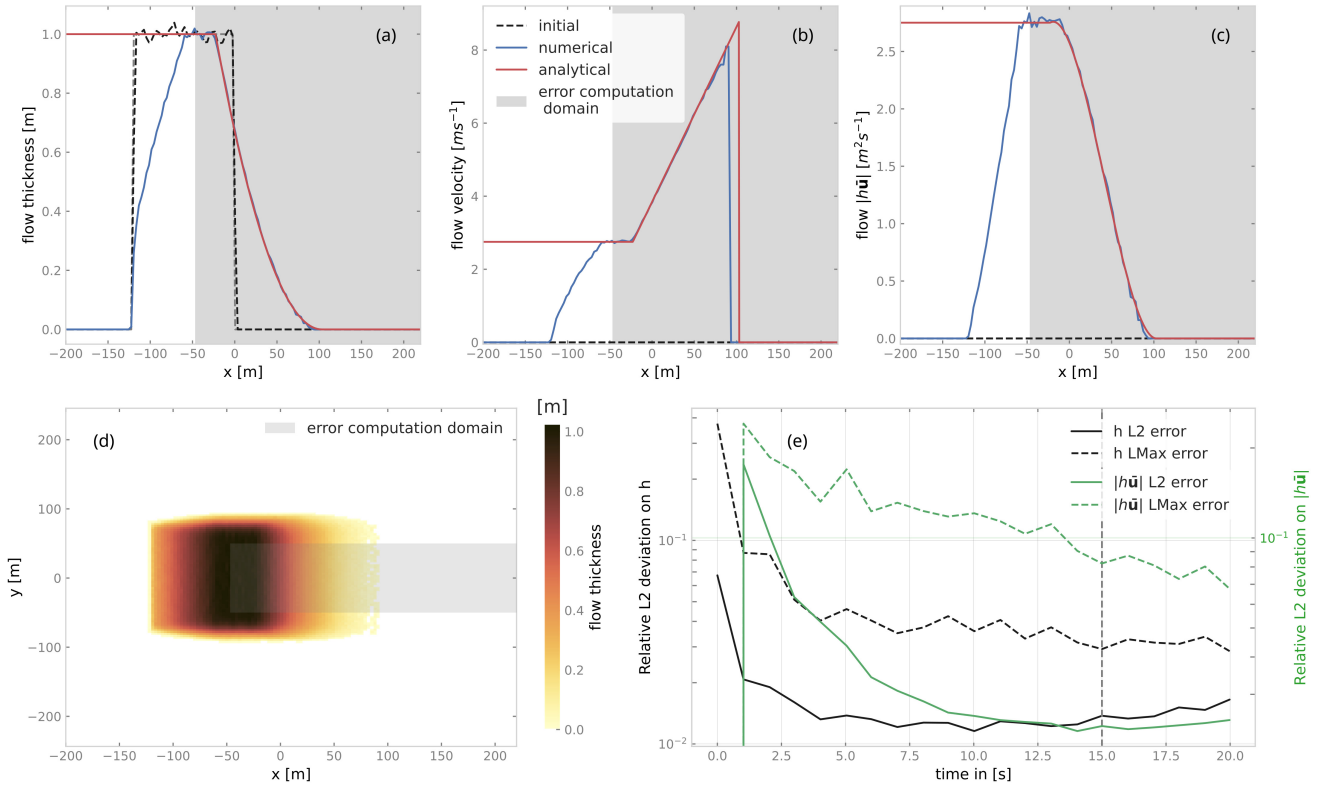
Figure 6. Comparison of the analytical and numerical solution for the dam break test case at $t = 15.0s$. Top panels show flow thickness (a), velocity (b) and momentum profiles (c) in flow direction. Panel (d) provides a top view of the flow thickness. The gray shaded rectangle represents the domain on which the deviations are computed. Panel (e) shows the time evolution of the deviation (both $\mathcal{RL}_{\max}$ and $\mathcal{RL}_2$) on flow thickness h and momentum $\|h\overline{u}\|$ between analytic and numerical solution ($r_{\text{kernel}} = 3$m, $\alpha = -3$, $n_{ppk}^0 = 15$ and $C_{\text{time}} = 0.02$s). The vertical dashed line in (e) marks the time at which data in the other panels are shown.

The α line can be used to study the effect of terms such as curvature acceleration, artificial viscosity or pressure gradients. For example, the curvature acceleration modifies the friction term, depending on topography curvature and particle velocity, leading to a mismatch between the energy solution and the DFA simulation. Figure 10 shows this curvature effect. The topography considered here is an inclined plane that smoothly transitions into a horizontal plane, so curvature only occurs in the transition part. The energy line test for this case shows that there is added dissipation only in the transition part, seen by squares not following the α line. Once all particles have reached the horizontal plane, the squares follow the α line again (with a shift in s coordinate).

Finally, the effects of the pressure force can be studied. For example, to show that adding the pressure forces does not influence the simulation runout (not shown here). This can be explained by the fact that pressure forces do not dissipate

545
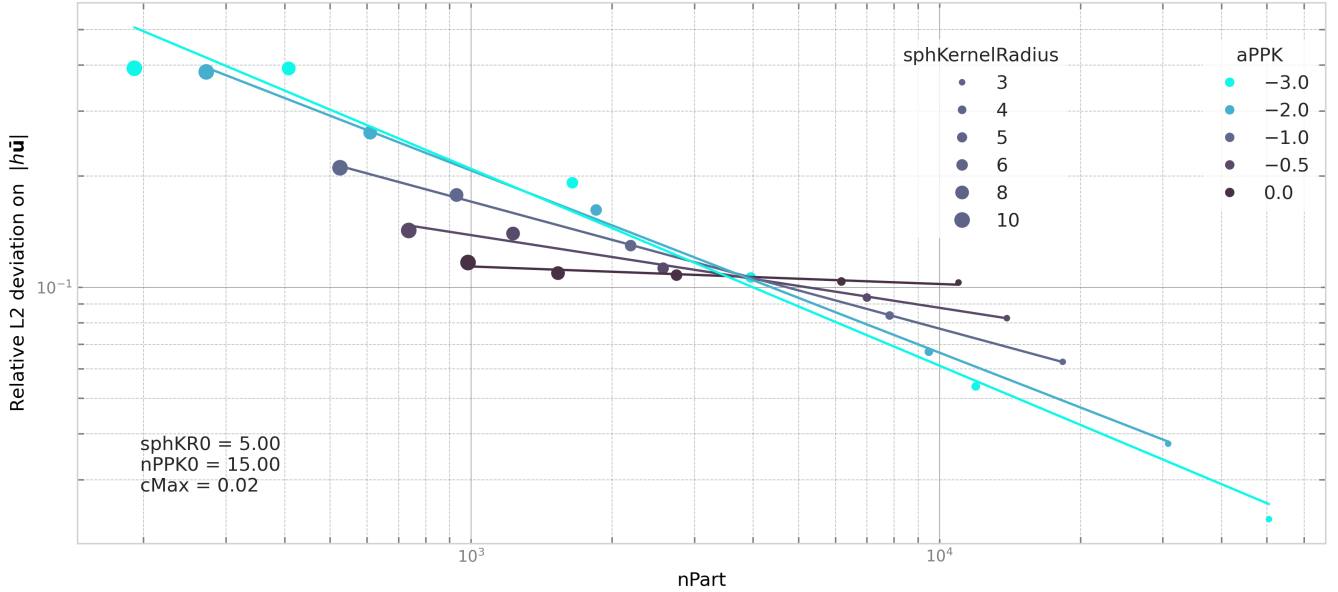
Figure 7. $\mathcal{RL}_2$ deviation on flow momentum in the dam break case for different α exponents and SPH kernel radius r_{ppk}. Other parameters are kept fixed (reference kernel radius r_{ppk}^0, time constant C_{time} and reference number of particles per kernel radius n_{ppk}^0). The points corresponding to simulations with $r_{ppk} = 5$m are all overlapping since they are all identical. The colored lines are added to give an idea of the convergence speed trend associated to each α scenario. One can observe that decrease in deviation is stronger for lower α exponents and that no or little decrease is observed for $\alpha = 0$ or $\alpha = -0.5$.
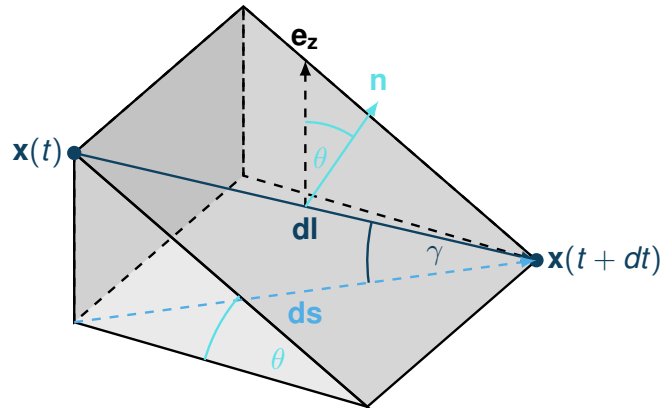
Figure 8. Example of trajectory where the steepest descent path hypothesis fails. The mass point is traveling from $\mathbf{x}(t)$ to $\mathbf{x}(t+dt)$. The slope angle θ and travel angle γ are also illustrated. Here $(\mathbf{e_z} \cdot \mathbf{n})dl = cos\theta \frac{\mathrm{d}s}{cos\gamma} \neq ds$.
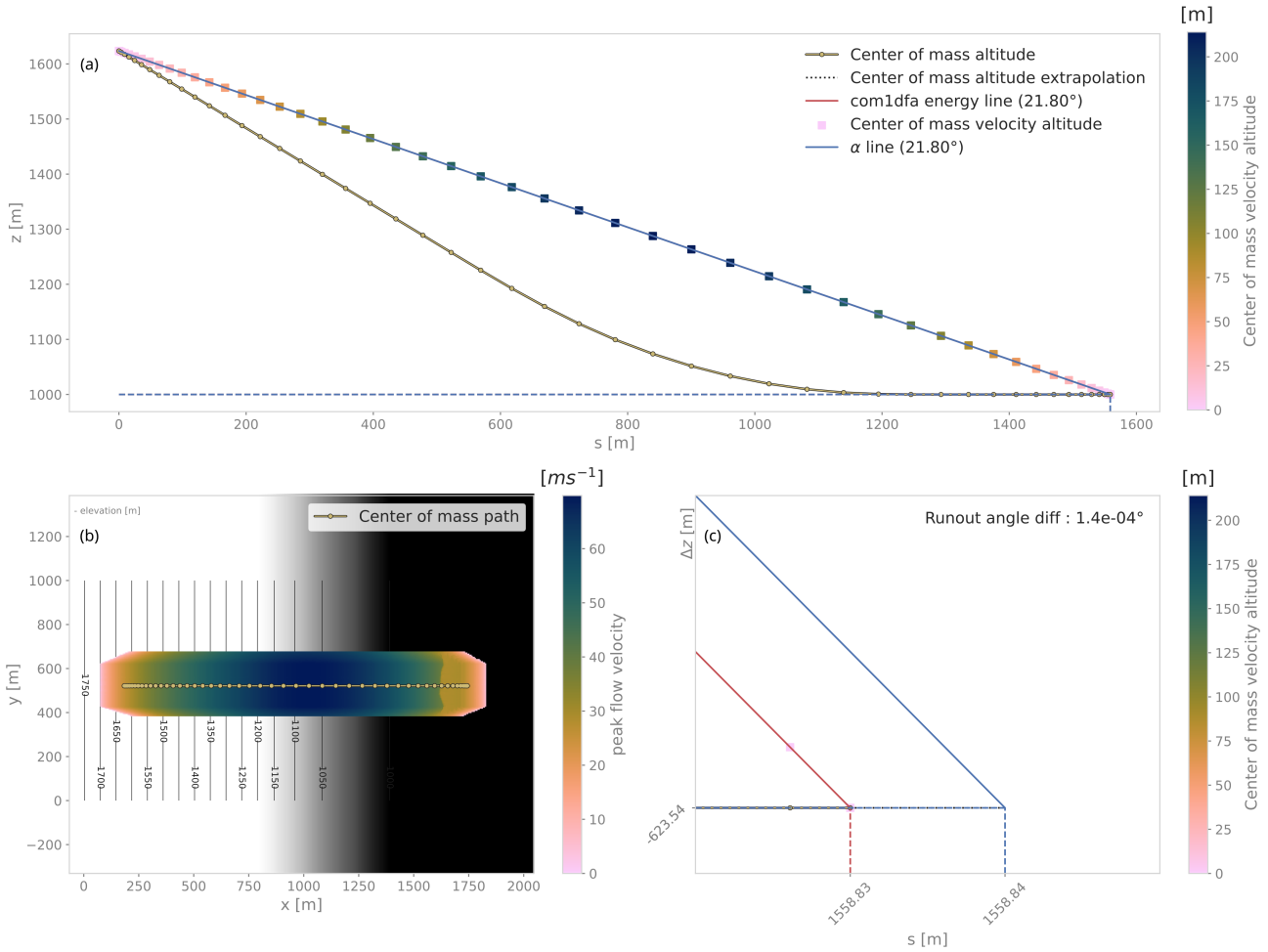
Figure 9. Energy line test for an inclined plane smoothly transitioning to a horizontal plane with particles following Coulomb friction and not subject to pressure forces. All conditions for the energy-line test to be applicable are satisfied and the geometrical solution can be used as reference to compute the numerical error (here less than 10^{-2}m and $10^{-4\circ}$). Panel (a) shows the center of mass profile, which is indicated in the top view of peak flow velocity in panel (b). Panel (c) shows a zoom-in to the lower end of the profile.

550 any energy and hence should not affect the energy balance. However, pressure forces lead to particle trajectories that do not necessarily follow the steepest direction, which means that the fundamental hypothesis illustrated in Fig. 8 is not satisfied.

5.2.2 Grid orientation effect

The energy line test previously described is also used to test whether the numerical method implemented in **com1DFA** performs independently of the grid orientation. Indeed we saw in Sect. 3.1 that **com1DFA** uses a regular grid to update some variables such as flow thickness or flow velocity. In order to quantify the effect of the grid orientation on the simulation results, we

555
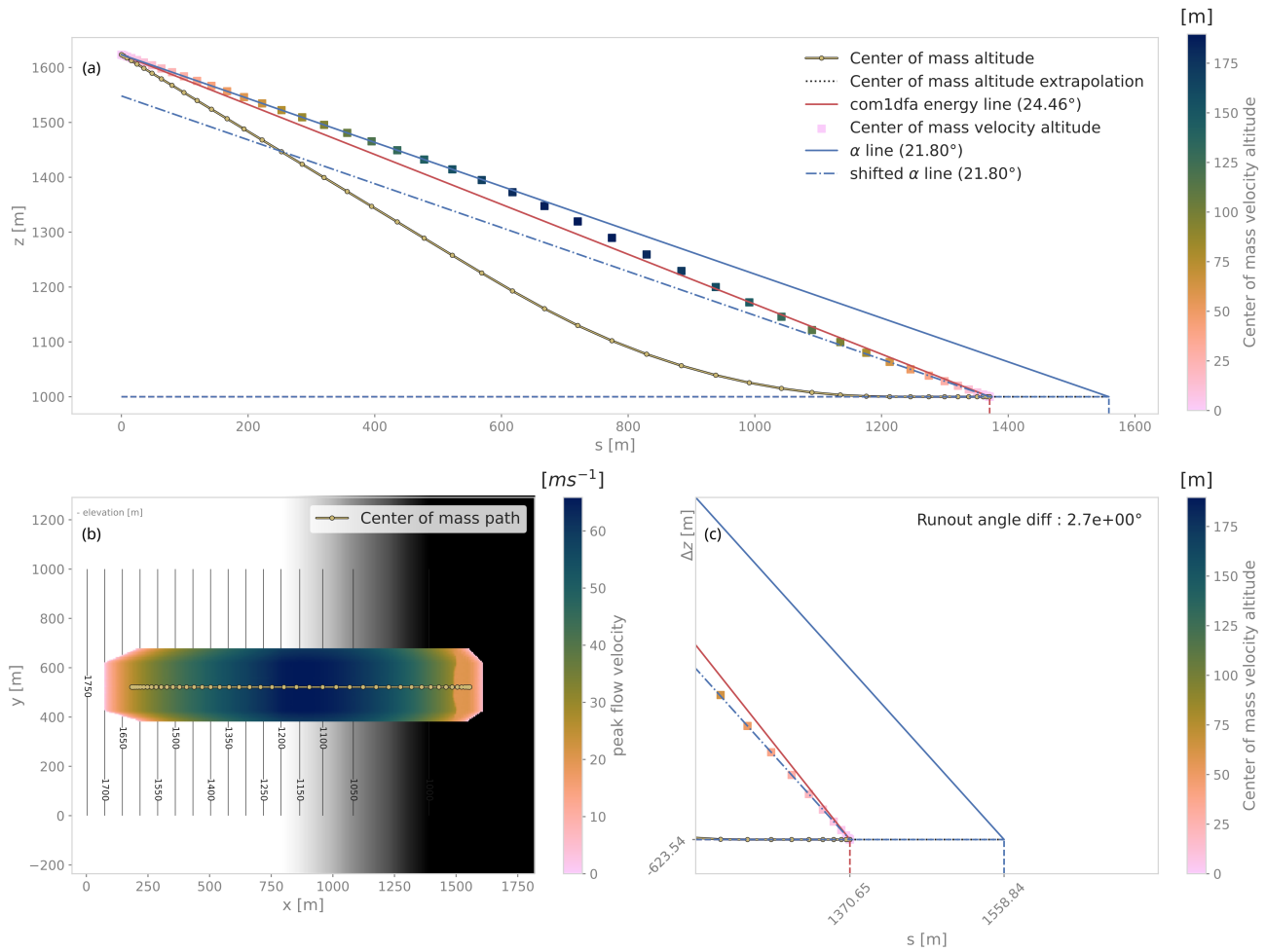
Figure 10. Energy line test for an inclined plane smoothly transitioning to a horizontal plane with particles following Coulomb friction, not subject to pressure forces but taking curvature acceleration into account. The geometrical solution and the numerical solution match on the inclined plane and horizontal parts and differ on the curved part. This shows the effect of curvature on the runout (decrease of runout because of the added friction due to curvature). Panel (a) shows the center of mass profile, which is indicated in the top view of peak flow velocity in panel (b). Panel (c) shows a zoom-in to the lower end of the profile.

perform tests where grid orientation is changed while keeping the grid cell size and topography the same. Here we show two examples where we consider a parabolic slope, i.e. the topography varies only in one direction, and a bowl shape, i.e. the topography with a rotational symmetry about its center. The main axis of the flow is not always aligned with the grid and we provide 3 cases. First a 0° case in which the slope is invariant in the y direction (main flow direction aligned with the grid). Second a 225° case, with the slope being invariant in a direction angled 225° from the y direction (main flow direction aligned with the grid diagonal) and third a 120° case meaning that the slope is invariant in a direction angled 120° from the y direction

560

(main flow direction is neither aligned with the grid nor with the grid diagonal). For each of these test cases, two simulations are performed, with or without pressure gradients. The results of these tests and the instructions on how to reproduce them are provided in the supplementary material. The results are very satisfying. The runout distances differ by only a few centimeters between the 0°, 120° and 225° cases. This is of the same order of magnitude as the numerical errors computed with the energy line tests. This proves the rather low dependence of the simulation results on the grid orientation.

6   Conclusions

The presented **com1DFA** module within the open-source simulation framework AvaFrame is aimed at and developed for the simulation of avalanches with flow or mixed form of movement. These may mostly be classified as A2B7C1D-E7F7G1H7J1, according to the morphological classification code (De Quervain et al., 1981). Effects of snow entrainment or additional sources of resistance can be included. The corresponding theoretical background is not described in this publication, but is available in the AvaFrame online documentation (Oesterle et al., 2022).

The module targets very- to extremely large avalanches of catastrophic intensity, corresponding to avalanches of size 4-5 of the European Avalanche Warning Service (EAWS) size classification, or a 150-year return period in the case of Austrian hazard mapping (Johannesson et al., 2009). Additional calibrated parameter sets for small and medium-sized avalanches and wet snow avalanches (albeit experimental), are also provided within the framework. However, these are exclusively optimized on a subset of Austrian test cases. Other types of avalanches, such as powder snow avalanches or similar, are unsuitable applications for the **com1DFA** module.

Com1DFA is based on the slope-normal thickness integration of the mass and momentum conservation equations which are solved with a mixed particle–grid method. Particles are used to track mass and to compute the pressure forces following a smooth particle hydrodynamics (SPH) approach adapted to the special case of steep terrain. An underlying Eulerian grid is used to compute the flow thickness from the distribution of the particles. Velocity is also interpolated from the particles to the grid in order to compute the stabilizing artificial viscosity term. Physical starting and stopping behaviour is ensured by considering differences in the friction force, whether the material is flowing or at rest. A time-space criterion ensuring the convergence of the implemented numerical method is provided. Of course limitations and assumptions exist, one of the main ones being the assumption of a moderately curved surface. For currently used resolutions of topographies (about 5-10m at the time of writing) terrain features remain relatively smooth, but it already requires forcing of particles to stay on the topography. For potential higher resolution applications in the future, this assumption needs to be re-evaluated

To verify the numerical implementation of **com1DFA**, we apply a series of tests separated into two categories. Flow variable tests, i.e. similarity solution and dam break tests, are used for checking the proper spatio-temporal evolution of flow thickness and velocity. Runout testing checks the accuracy of global variables such as center-of-mass runout and kinetic energy. The tests show the validity of the chosen time-space criterion as well as the accuracy and precision of the **com1DFA** numerical solution.

Note that the computational efficiency of the **com1DFA** module is not a topic in this manuscript since simulations with the standard setup compute within seconds to minutes on current computer hardware. To achieve better computational efficiency,

595 we implemented parallel execution of multiple serial simulations. However, topics like computational efficiency, further in-depth testing and application to real topographies will be treated in future publications.

Current and future potential improvements to **com1DFA** include topics such as the improvement of entrainment and de-trainment (e.g. concerning forests), a more numerically sound representation of dams/walls or a probabilistic approach to uncertainties. These topics will need both conceptual developments as well as numerical improvements, but some are already

600 being tackled. Possible applications to other types of gravitational mass flows might also be an exciting development, but our current focus lies on snow avalanches. However, since AvaFrame is an open-source framework, we invite everyone to use the presented modules and welcome feedback and contributions, regardless of the topic.

Appendix A: Theory Energy line test

The conservation of energy for a material point (block model) flowing down-slope from point O to point B, assuming only

605 Coulomb friction is at work and neglecting internal energy dissipation through heating, reads:

$$E_B^{tot} - E_O^{tot} = E_B^{kin} + E_B^{pot} - (E_O^{kin} + E_O^{pot}) = \int_O^B \delta E_{fric}$$

$$= \frac{1}{2}mv_B^2 + mgz_B - \frac{1}{2}mv_O^2 - mgz_O \qquad (A1)$$

$$= \int_O^B \mathbf{F}_{fric} \cdot \mathbf{dl} = - \int_O^B \mu \|\mathbf{N}\| \frac{\mathbf{dl}}{dl} \cdot \mathbf{dl} = - \int_O^B \mu mg(\mathbf{e_z} \cdot \mathbf{n})dl,$$

where δE_{fric} is the energy dissipation due to friction, $\mathbf{N}$ represents the normal (to the bottom surface) component of the gravity force, $\mathbf{n}$ the normal vector to the slope surface and $\mathbf{dl}$ is the elementary vector on the path profile traveled by the material point between O and B. The vertical component of the normal vector reads $\mathbf{e_z} \cdot \mathbf{n} = \cos\theta$, where θ is the slope angle. m represents the

610 mass of the material point, g the gravity, $\mu = \tan\alpha$ the friction coefficient and friction angle, z the elevation and v the velocity of the material point. Note that in the 2D case, $dl = \frac{ds}{\cos\theta}$, which means that the material point is flowing in the steepest slope direction ($\mathbf{ds}$ is the horizontal component of $\mathbf{dl}$). Now considering O as the origin position ($s_O = 0$ and $v_O = 0$) leads to the following simplification:

$$E_B^{tot} - E_O^{tot} = \frac{1}{2}mv_B^2 + mgz_B - \frac{1}{2}mv_O^{2\;\nearrow^0} - mgz_O^{\;\nearrow^{z_0}}$$

$$= \int_O^B \delta E_{fric} = - \int_{s'=s_O}^{s'=s_B} \mu mg ds' = -\mu mg(s_B - s_O^{\;\nearrow^0}). \qquad (A2)$$

615 Speaking in terms of altitude, the energy conservation equation can be rewritten (the subscript B is dropped):

$$z_O = z + \frac{v^2}{2g} + \tan\alpha. \qquad (A3)$$

Considering a system of material points flowing down a slope with Coulomb friction, we can sum the previous equation (Eq. A3) of each material point after weighting it by its mass. This leads to the mass-averaged energy conservation equation:

$$\overline{z}_O = \overline{z} + \frac{\overline{v^2}}{2g} + \overline{s}\tan\alpha, \tag{A4}$$

where the mass-averaged value $\overline{a}$ of a quantity a is (k indicates the points indices):

$$\overline{a} = \frac{\sum_k m_k a_k}{\sum_k m_k}. \tag{A5}$$

In this way, we can define the center-of-mass path $(\overline{x}, \overline{y})$ and the center-of-mass path profile $(\overline{s}, \overline{z})$. The mass-averaged quantities also follow the same energy conservation law when expressed in terms of altitude. This result is illustrated in Fig. A1 (b) and applies to both the material point equation (Eq. A3) and the mass-averaged energy conservation equation (Eq. A4). The light blue line in Fig. A1 (b) is obtained by evaluating the mass-averaged energy conservation (Eq. A4) at the final time (t_{end}) and position (($\overline{s}_{\text{end}}$, $\overline{z}_{\text{end}}$)), where $v^2 = 0$. This leads to the α line (also called energy line) equation:

$$\overline{z} = \overline{z}_O - \overline{s}\tan\alpha. \tag{A6}$$
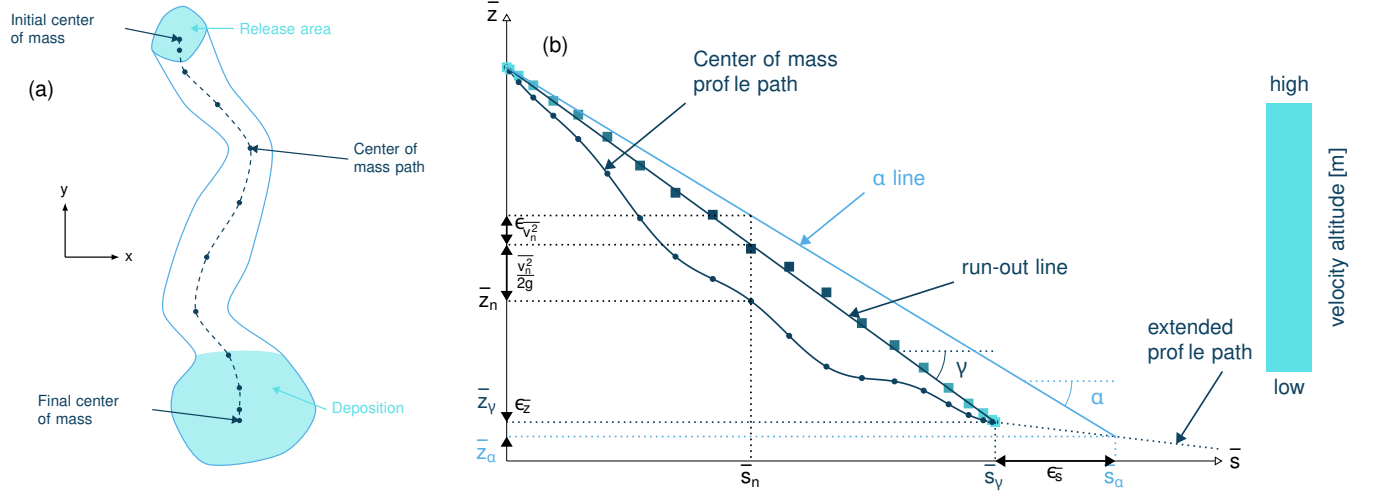
Figure A1. Panel (a) shows a top-down view of the avalanche simulation and extracted path. (b) shows simulation path profile (dark blue curve and dots) with the runout line (dark blue line and velocity altitude squares), α line and runout error indicators (ϵ_s and ϵ_z)

Appendix B: SPH pressure gradients

The SPH method used in shallow water equations is in most applications applied on a horizontal surface. The theoretical development on a horizontal plane is described in B1. The dense flow avalanche model described in this paper should be expressed on the bottom surface which is not necessarily horizontal. The SPH gradient computation development is detailed in B2.

B1 Standard method

Let us start with the computation of the gradient of a scalar function $f\colon \mathbb{R}^2 \to \mathbb{R}$ on a horizontal plane. Let $P_k = \mathbf{x}_k = (x_{k,1}, x_{k,2})$ and $Q_l = \mathbf{x}_l = (x_{l,1}, x_{l,2})$ be two points in $\mathbb{R}^2$ defined by their coordinates in the Cartesian coordinate system $(P_k, \mathbf{e_1}, \mathbf{e_2})$. $\mathbf{r}_{kl} = \mathbf{x}_k - \mathbf{x}_l$ is the vector from Q_l to P_k and $r_{kl} = \|\mathbf{r}_{kl}\|$ the length of this vector. Now consider the kernel function W:

$$W\colon \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}, (P_k, Q_l, r_0) \mapsto W(P_k, Q_l, r_0). \tag{B1}$$

$r_0 \in \mathbb{R}$ is the smoothing kernel length (or radius). In the case of the spiky kernel, W reads (2D case):

$$
\begin{aligned}
W_{kl} &= W(\mathbf{x}_k, \mathbf{x}_l, r_0) = W(\mathbf{x}_k - \mathbf{x}_l, r_0) = W(\mathbf{r}_{kl}, r_0) \\
&= \frac{10}{\pi r_0^5}
\begin{cases}
(r_0 - \|\mathbf{r}_{kl}\|)^3, & 0 \le \|\mathbf{r}_{kl}\| \le r_0 \\
0, & r_0 < \|\mathbf{r}_{kl}\|.
\end{cases}
\end{aligned}
\tag{B2}
$$

$\|\mathbf{r}_{kl}\| = \|\mathbf{x}_k - \mathbf{x}_l\|$ is the distance between particles k and l and r_0 the smoothing length. Using the chain rule to express the gradient of W in the Cartesian coordinate system (x_1, x_2) leads to:

$$\boldsymbol{\nabla} W_{kl} = \frac{\partial W}{\partial r} \cdot \boldsymbol{\nabla} r, \quad r = \|\mathbf{r}\| = \sqrt{(x_{k,1} - x_{l,1})^2 + (x_{k,2} - x_{l,2})^2}, \tag{B3}$$

with

$$\frac{\partial W}{\partial r} = -3 \frac{10}{\pi r_0^5}
\begin{cases}
(r_0 - \|\mathbf{r}_{kl}\|)^2, & 0 \le \|\mathbf{r}_{kl}\| \le r_0 \\
0, & r_0 < \|\mathbf{r}_{kl}\|
\end{cases}
\tag{B4}$$

and

$$\frac{\partial r}{\partial x_{k,i}} = \frac{(x_{k,i} - x_{l,i})}{\sqrt{(x_{k,1} - x_{l,1})^2 + (x_{k,2} - x_{l,2})^2}}, \quad i = \{1, 2\}, \tag{B5}$$

which leads to the following expression for the gradient:

$$\boldsymbol{\nabla} W_{kl} = -3 \frac{10}{\pi r_0^5}
\begin{cases}
(r_0 - \|\mathbf{r}_{kl}\|)^2 \dfrac{\mathbf{r}_{kl}}{r_{kl}}, & 0 \le \|\mathbf{r}_{kl}\| \le r_0 \\
0, & r_0 < \|\mathbf{r}_{kl}\|.
\end{cases}
\tag{B6}$$

The gradient of f is then simply:

$$\boldsymbol{\nabla} f_k = -\sum_l f_l A_l \, \boldsymbol{\nabla} W_{kl}. \tag{B7}$$

B2 2.5D SPH method

We now want to express a function f and its gradient on a curved surface and express this gradient in the three-dimensional Cartesian coordinate system $(P_k, \mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3})$. Let us consider a smooth surface $\mathcal{S}$ and two points $P_k : \mathbf{x}_k = (x_{k,1}, x_{k,2}, x_{k,3})$

655 and $Q_l : \mathbf{x}_l = (x_{l,1}, x_{l,2}, x_{l,3})$ on $\mathcal{S}$. We can define $\mathcal{TP}_k$ the tangent plane to $\mathcal{S}$ in P_k. If $\mathbf{u}_k$ is the (non-zero) velocity of the particle at P_k, it is possible to define the local orthonormal coordinate system $(P_k, \mathbf{V_1}, \mathbf{V_2}, \mathbf{V_3} = \mathbf{n})$ with time-dependent $\mathbf{V_1} = \frac{\mathbf{u}_k(t)}{\|\mathbf{u}_k(t)\|}$ and $\mathbf{n}$ the normal to $\mathcal{S}$ at P_k. Locally, $\mathcal{S}$ can be assimilated to $\mathcal{TP}_k$ and Q_l to its projection Q'_l on $\mathcal{TP}_k$ (see Fig. B1). Similarly, we assimilate a function $f : \mathcal{S} \subset \mathbb{R}^3 \to \mathbb{R}$ to a function $f' : \mathcal{TP}_k \subset \mathbb{R}^3 \to \mathbb{R}$ satisfying $f(Q) = f'(Q')$.
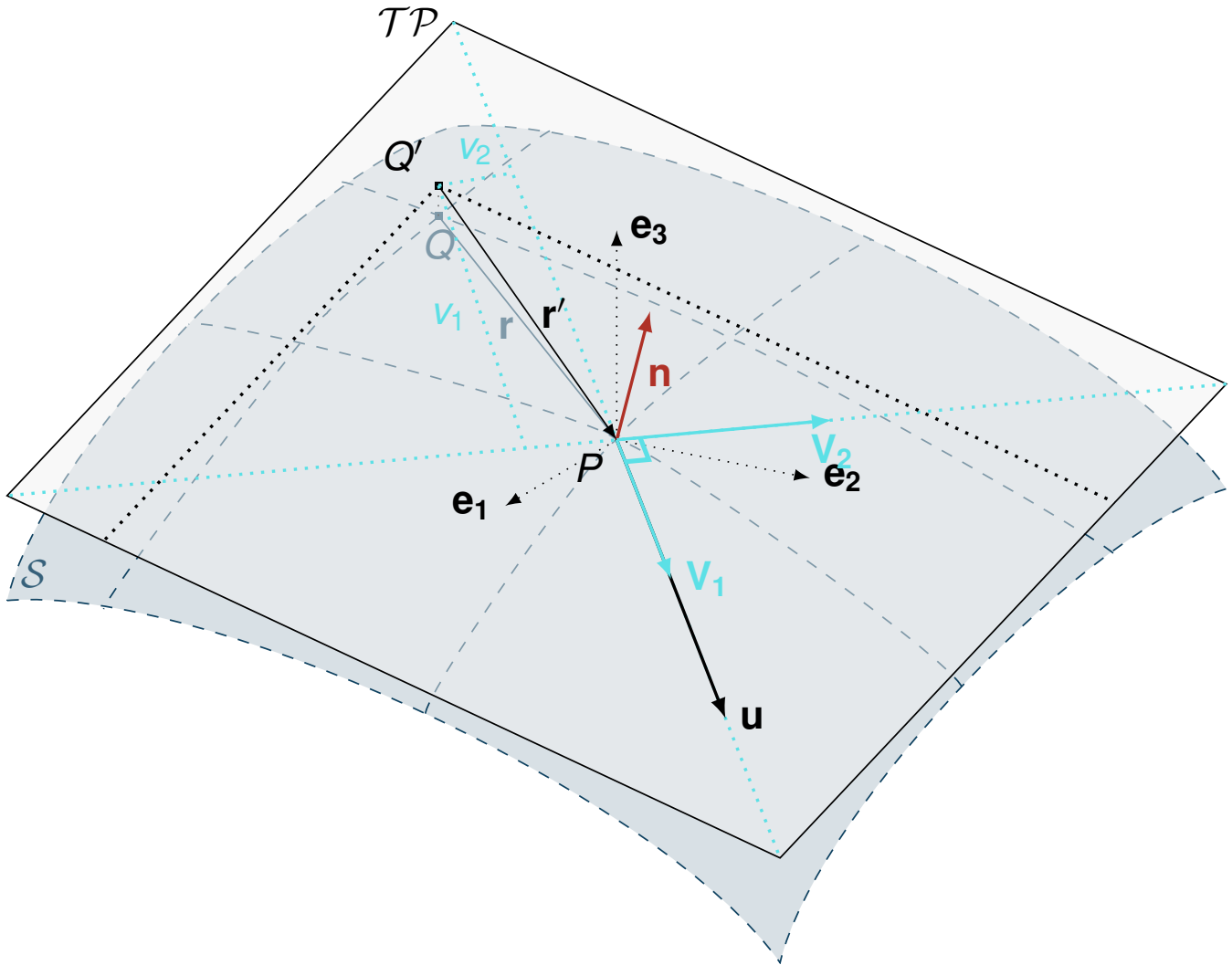
Figure B1. Tangent plane and local coordinate system used to apply the SPH method

The vector $\mathbf{r}'_{kl} = \mathbf{x}_k - \mathbf{x}'_l$ from Q'_l to P_k lies in $\mathcal{TP}_k$ and can be expressed in the local basis of $\mathcal{TP}_k$:

660 $\mathbf{r}'_{kl} = \mathbf{x}_k - \mathbf{x}'_l = v_{kl,1}\mathbf{V_1} + v_{kl,2}\mathbf{V_2}.$ (B8)

It is important to properly define f and its gradient:

$$f': \mathcal{TP}_k \subset \mathbb{R}^3 \to \mathbb{R}$$
$$(x_1, x_2, x_3) \mapsto f'(x_1, x_2, x_3) = f'(x_1(v_1, v_2), x_2(v_1, v_2)) = g'(v_1, v_2). \tag{B9}$$

Indeed, since (x_1, x_2, x_3) lies in $\mathcal{TP}_k$, x_3 is not independent of (x_1, x_2):

$$g': \mathcal{TP}_k \subset \mathbb{R}^2 \to \mathbb{R}$$
$$(v_1, v_2) \mapsto g'(v_1, v_2) = g'(v_1(x_1, x_2), v_2(x_1, x_2)) = f'(x_1, x_2, x_3). \tag{B10}$$

The target is the gradient of g' in terms of the $\mathcal{TP}_k$ variables (v_1, v_2). Let us call this gradient $\nabla_{\mathcal{TP}}$. It is then possible to apply the B1 method to compute this gradient:

$$\nabla_{\mathcal{TP}} W_{kl} = \frac{\partial W}{\partial r} \cdot \nabla_{\mathcal{TP}} r, \quad r = \|\mathbf{r}\| = \sqrt{v_{kl,1}^2 + v_{kl,2}^2}, \tag{B11}$$

This leads to:

$$\nabla_{\mathcal{TP}} W_{kl} = -3 \frac{10}{\pi r_0^5} \frac{(r_0 - \|\mathbf{r}'_{kl}\|)^2}{r'_{kl}} \begin{cases} v_{kl,1} \mathbf{V_1} + v_{kl,2} \mathbf{V_2}, & 0 \leq \|\mathbf{r}'_{kl}\| \leq r_0 \\ 0, & r_0 < \|\mathbf{r}'_{kl}\| \end{cases} \tag{B12}$$

$$\nabla_{\mathcal{TP}} g'_k = - \sum_l g'_l A_l \nabla_{\mathcal{TP}} W_{kl}. \tag{B13}$$

This gradient can now be expressed in the Cartesian coordinate system. It is clear that the change of coordinate system was not needed:

$$\nabla_{\mathcal{TP}} W_{kl} = -3 \frac{10}{\pi r_0^5} \frac{(r_0 - \|\mathbf{r}'_{kl}\|)^2}{r'_{kl}} \begin{cases} r_{kl,1} \mathbf{e_1} + r_{kl,2} \mathbf{e_2} + r_{kl,3} \mathbf{e_3}, & 0 \leq \|\mathbf{r}'_{kl}\| \leq r_0 \\ 0, & r_0 < \|\mathbf{r}'_{kl}\|. \end{cases} \tag{B14}$$

Computing the gradient in the local coordinate system is, however, advantageous if the components (in flow direction or in cross flow direction) need to be treated differently.

Code and data availability. The AvaFrame software is publicly available at https://github.com/avaframe/AvaFrame/ and https://doi.org/10.5281/zenodo.4721446 (Oesterle et al., 2022). The associated online documentation is available at https://docs.avaframe. org/en/latest/. The code is released under the European Union Public license (EUPL version 1.2). The version of the code (including config- uration files) and documentation for reproducing the results presented in this paper are available at https://github.com/avaframe/AvaFrame/ tree/theoryPaperCode and https://docs.avaframe.org/en/theorypapercode/index.html

Author contributions. AW, FO and MT are the principal programmers and jointly developed the Dense Flow Avalanche numerical model code. MT prepared the manuscript and performed the model simulations to produce the results and figures of this manuscript with contribu- tions from AW. FO and JTF jointly supervised the project and contributed to the discussion and paper.

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

Acknowledgements. The authors would like to acknowledge Peter Sampl who developed the theory and source code for SamosAT which represented the starting point of the **com1DFA** module of AvaFrame. Thank you to Marie von Busse and Wolfgang Fellin for their help in developing the energy line test. We also would like to thank our anonymous reviewer and Dieter Issler for the constructive review that immensely helped improve the manuscript's clarity and quality.

690 *Financial support.* AvaFrame is supported by the Austrian Federal Ministry for Agriculture, Forestry, Regions and Water Management through a cooperation between the Austrian Avalanche and Torrent Control (WLV) and the Austrian Research Centre for Forests (BFW). Additional support is provided by the international cooperation project "AvaRange - Particle Tracking in Snow Avalanches" supported by the German Research Foundation (DFG, project No. 421446512) and the Austrian Science Fund (FWF, project No. I 4274-N29).

References

695   Ata, R. and Soulaïmani, A.: A stabilized SPH method for inviscid shallow water flows, International Journal for Numerical Methods in Fluids, 47, 139–159, https://doi.org/10.1002/fld.801, 2005.

Baker, J. L., Barker, T., and Gray, J. M. N. T.: A two-dimensional depth-averaged $\mu(I)$-rheology for dense granular avalanches, Journal of Fluid Mechanics, 787, 367–395, https://doi.org/10.1017/jfm.2015.684, 2016.

Christen, M., Kowalski, J., and Bartelt, P.: RAMMS: Numerical simulation of dense snow avalanches in three-dimensional terrain, Cold
700   Regions Science and Technology, 63, 1–14, https://doi.org/10.1016/j.coldregions.2010.04.005, 2010.

De Quervain, R. et al.: Avalanche atlas, Unesco, Paris, http://unesdoc.unesco.org/images/0004/000480/048004MB.pdf, 1981.

Faccanoni, G. and Mangeney, A.: Exact solution for granular flows, International Journal for Numerical and Analytical Methods in Geomechanics, 37, 1408–1433, https://doi.org/10.1002/nag.2124, 2013.

Fischer, J. T., Kowalski, J., and Pudasaini, S.: Topographic curvature effects in applied avalanche modeling, Cold Regions Science and
705   Technology, 74-75, 21–30, https://doi.org/10.1016/j.coldregions.2012.01.005, 2012.

Fischer, J.-T., Fromm, R., Gauer, P., and Sovilla, B.: Evaluation of probabilistic snow avalanche simulation ensembles with Doppler radar observations, Cold Regions Science and Technology, 97, 151–158, https://doi.org/https://doi.org/10.1016/j.coldregions.2013.09.011, 2014.

Gauer, P.: Comparison of avalanche front velocity measurements and implications for avalanche models, Cold Regions Science and Technology, 97, 132–150, https://doi.org/10.1016/j.coldregions.2013.09.010, 2014.

710   Granig, M., Sampl, P., Fischer, J., Kofler, A., and Joerg, P.: Adaption and further development of the numerical solution in the avalanche simulation model SamosAT, in: 13th Congress INTERPRAEVENT, 2016.

Gray, J. and Edwards, A.: A depth-averaged (I)-rheology for shallow granular free-surface flows, Journal of Fluid Mechanics, 755, 297–329, https://doi.org/10.1017/jfm.2014.450, 2014.

Gubler, H.: Measurements and modelling of snow avalanche speeds, IAHS Publ, 162, 405–420, 1987.

715   Harten, A. and Hyman, J. M.: Self adjusting grid methods for one-dimensional hyperbolic conservation laws, Journal of Computational Physics, 50, 235–269, https://doi.org/https://doi.org/10.1016/0021-9991(83)90066-9, 1983.

Harten, A., Lax, P. D., and Leer, B. v.: On upstream differencing and Godunov-type schemes for hyperbolic conservation laws, SIAM Review, 25, 35–61, https://doi.org/10.1137/1025002, 1983.

Hergarten, S. and Robl, J.: Modelling rapid mass movements using the shallow water equations in Cartesian coordinates, Natural Hazards
720   and Earth System Sciences, 15, 671–685, https://doi.org/10.5194/nhess-15-671-2015, 2015.

Hutter, C., Siegel, M., Savage, S., and Nohguchi, Y.: Two-dimensional spreading of a granular avalanche down an inclined plane Part I. Theory, Acta Mechanica, 100, 37–68, https://doi.org/10.1007/BF01176861, 1993.

Ihmsen, M., Orthmann, J., Solenthaler, B., Kolb, A., and Teschner, M.: SPH Fluids in Computer Graphics, in: Eurographics 2014 - State of the Art Reports, edited by Lefebvre, S. and Spagnuolo, M., The Eurographics Association, https://doi.org/10.2312/egst.20141034, 2014.

725   Johannesson, T., Gauer, P., Issler, P., and Lied, K.: The design of avalanche protection dams, Recent practical and theoretical developments European Commission. Directorate General for Research, Brussels, Belgium, 2009.

Körner, H. J.: The energy-line method in the mechanics of avalanches, Journal of Glaciology, 26, 501–505, https://doi.org/10.3189/s0022143000011023, 1980.

LeVeque, R. J.: Numerical methods for conservation laws, https://doi.org/http://dx.doi.org/10.1007/978-3-0348-8629-1, 1990.

Li, X., Sovilla, B., Jiang, C., and Gaume, J.: Three-dimensional and real-scale modeling of flow regimes in dense snow avalanches, Land-
slides, 18, 3393–3406, https://doi.org/10.1007/s10346-021-01692-8, 2021.

Liu, M. and Liu, G.: Smoothed Particle Hydrodynamics (SPH): an overview and recent developments, Archives of Computational Methods
in Engineering, 17, 25–76, https://doi.org/10.1007/s11831-010-9040-7, 2010.

Luca, I., Tai, Y.-C., and Kuo, C.-Y.: Shallow Geophysical Mass Flows down Arbitrary Topography, Springer International Publishing,
https://doi.org/10.1007/978-3-319-02627-5, 2016.

Mangeney-Castelnau, A., Vilotte, J.-P., Bristeau, M.-O., Perthame, B., Bouchut, F., Simeoni, C., and Yerneni, S.: Numerical modeling of
avalanches based on Saint Venant equations using a kinetic scheme, Journal of Geophysical Research: Solid Earth, 108, 2527–2544,
https://doi.org/10.1029/2002JB002024, 2003.

Mergili, M., Fischer, J.-T., Krenn, J., and Pudasaini, S. P.: r. avaflow v1, an advanced open-source computational framework for the propaga-
tion and interaction of two-phase mass flows, Geoscientific Model Development, 10, 553–569, https://doi.org/10.5194/gmd-10-553-2017,
2017.

Monaghan, J.: Smoothed particle hydrodynamics, Annual Review of Astronomy and Astrophysics, 30, 543–574, https://doi.org/10.1146/
annurev.aa.30.090192.002551, 1992.

Monaghan, J. J.: Smoothed particle hydrodynamics, Reports on Progress in Physics, 68, 1703–1759, https://doi.org/10.1088/0034-
4885/68/8/r01, 2005.

Moussa, B. and Vila, J. P.: Convergence of SPH method for scalar nonlinear conservation laws, Siam Journal on Numerical Analysis - SIAM
J NUMER ANAL, 37, https://doi.org/10.1137/S0036142996307119, 2000.

Oesterle, F., Tonnel, M., Wirbel, A., and Fischer, J.-T.: avaframe/AvaFrame: Version 1.3, https://doi.org/10.5281/zenodo.7189007, 2022.

Rauter, M. and Tuković, : A finite area scheme for shallow granular flows on three-dimensional surfaces, Computers & Fluids, 166, 184–199,
https://doi.org/10.1016/j.compfluid.2018.02.017, 2018.

Rauter, M., Kofler, A., Huber, A., and Fellin, W.: faSavageHutterFOAM 1.0: depth-integrated simulation of dense snow avalanches on natural
terrain with OpenFOAM, Geoscientific Model Development, 11, 2923–2939, https://doi.org/10.5194/gmd-11-2923-2018, 2018.

Salm, B. and Gubler, H.: Measurement and analysis of the motion of dense flow avalanches, Annals of Glaciology, 6, 26–34, https://doi.org/
10.1017/s0260305500009939, 1985.

Sampl, P.: SamosAT - Modelltheorie und Numerik, AVL List GMBH, Tech. rep., https://doi.org/10.5281/zenodo.8007681, 2007.

Sampl, P. and Zwinger, T.: Avalanche simulation with SAMOS, Annals of Glaciology, 38, 393–398, 2004.

Savage, S. B. and Hutter, K.: The motion of a finite mass of granular material down a rough incline, Journal of Fluid Mechanics, 199,
177–215, https://doi.org/10.1017/s0022112089000340, 1989.

Stomakhin, A., Schroeder, C., Chai, L., Teran, J., and Selle, A.: A material point method for snow simulation, ACM Transactions on Graphics,
32, 1–10, https://doi.org/10.1145/2461912.2461948, 2013.

Voellmy, A.: Über die Zerstörungskraft von Lawinen, Schweizerische Bauzeitung, Sonderdruck aus dem 73. Jahrgang, 1–25, 1955.

Zugliani, D. and Rosatti, G.: TRENT2D: An accurate numerical approach to the simulation of two-dimensional dense snow avalanches in
global coordinate systems, Cold Regions Science and Technology, 190, 103 343, https://doi.org/10.1016/j.coldregions.2021.103343, 2021.