The finite area scheme allows a description in terms of surface partial differential equations (Deckelnick et al., 2005), which leads to simple and expressive governing equations. However, this comes at the cost of a complex three-dimensional surface mesh. Projection of the governing equations on a plane surface following e.g. Bouchut and Westdickenberg (2004) may be beneficial for some applications. The three-dimensional surface mesh can also be an advantage, allowing a simple coupling with three-dimensional ambient two-phase models for powder clouds (Sampl and Zwinger, 2004). The presented meshing method, creating a finite volume and the corresponding finite area mesh, is viable for such simulations as well.

Future steps will incorporate optimisation of the solver in terms of stability and execution time. Mesh generation and the integration of geographic information systems will be further streamlined. We aim to implement more complex models, suitable for mixed snow avalanches (e.g., Bartelt et al., 2015; Issler et al., 2017) and debris flow (e.g., Iverson and George, 2014; Mergili et al., 2017) in the near future. Coupling of the here proposed dense flow model with three-dimensional two-phase models for the powder cloud regime (e.g. Cheng et al., 2017; Chauchat et al., 2017) is planned as well.

Code and data availability. The OpenFOAM solver, core utilities and the presented case study are available in the OpenFOAM community repository (https://develop.openfoam.com/Community/avalanche) and integrated as a module within OpenFOAM-v1712. The complete code (based on foam-extend-4.0) including python scripts for GIS integration and the simulation setup including the underlying raw data is included in the supplementary material and available at https://bitbucket.org/matti2/fasavagehutterfoam.

Appendix A: Understanding projections in surface partial differential equations

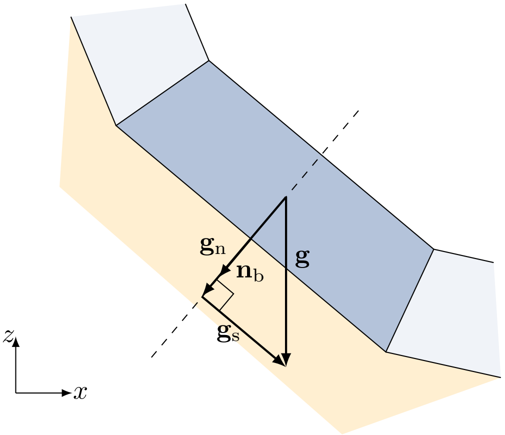
Here we shortly explain the concept of projections within the framework of surface partial differential equations. These projections are widely used in computational fluid dynamics, usually when surfaces in three dimensional space are considered. We do not focus on mathematical formalities and this section can not replace the formal derivation of Rauter and Tuković (2018). We want to emphasize that no surface aligned coordinate system is required throughout the whole process and the reader is encouraged to stick to global Cartesian coordinates. For simplicity we present a discretised finite area cell, which has been extruded by flow thickness h to present the flowing mass, see Fig. A1.

We begin by splitting a simple vectorial entity, the gravitational acceleration $\mathbf{g} \in \mathbb{R}^3$, into a surface normal component, $\mathbf{g}_\mathrm{n} \in \mathbb{R}^3$, and a surface tangential component, $\mathbf{g}_\mathrm{s} \in \mathbb{R}^3$, as shown in Fig. A1. The magnitude of the surface normal component can be calculated using the scalar-product and the surface normal vector,

$$\|\mathbf{g}_\mathrm{n}\| = \mathbf{n}_\mathrm{b} \cdot \mathbf{g}, \tag{A1}$$

which corresponds to a projection of $\mathbf{g}$ on $\mathbf{n}_\mathrm{b}$. The surface normal component points in the same direction as the surface normal vector, which allows calculation of the vectorial surface normal component. Rearranging of vector multiplications yields the known form,

$$\mathbf{g}_\mathrm{n} = \mathbf{n}_\mathrm{b}\,\|\mathbf{g}_\mathrm{n}\| = \mathbf{n}_\mathrm{b}\,(\mathbf{n}_\mathrm{b} \cdot \mathbf{g}) = (\mathbf{n}_\mathrm{b}\,\mathbf{n}_\mathrm{b}) \cdot \mathbf{g}. \tag{A2}$$



$\mathbf{g}_{\mathrm{n}}$
$\mathbf{n}_{\mathrm{b}}$
$\mathbf{g}$
$\mathbf{g}_{\mathrm{s}}$
z
x

Figure A1. Splitting gravitational acceleration into a surface tangential and surface normal part with simple projections to the surface normal vector $\mathbf{n}_{\mathrm{b}}$.

The surface tangential component follows by subtracting the surface normal component from total gravitational acceleration,

$$\mathbf{g}_{\mathrm{s}} = \mathbf{g} - \mathbf{g}_{\mathrm{n}} = \mathbf{g} - (\mathbf{n}_{\mathrm{b}}\,\mathbf{n}_{\mathrm{b}}) \cdot \mathbf{g} = (\mathbf{I} - \mathbf{n}_{\mathrm{b}}\,\mathbf{n}_{\mathrm{b}}) \cdot \mathbf{g}. \tag{A3}$$

Movement in surface normal direction is constrained by the basal topography, which yields the basal pressure. Therefore, the surface normal component $\mathbf{g}_{\mathrm{n}}$ has to contribute to basal pressure p_{b} (Eq. 3), and only the surface tangential component contributes to local acceleration $\frac{\partial h\,\overline{\mathbf{u}}}{\partial t}$ (Eq. 2). The total gravitational acceleration can be reconstructed by summing up both components,

$$\mathbf{g} = \mathbf{g}_{\mathrm{n}} + \mathbf{g}_{\mathrm{s}} = (\mathbf{n}_{\mathrm{b}}\,\mathbf{n}_{\mathrm{b}}) \cdot \mathbf{g} + (\mathbf{I} - \mathbf{n}_{\mathrm{b}}\,\mathbf{n}_{\mathrm{b}}) \cdot \mathbf{g} = \mathbf{I} \cdot \mathbf{g} = \mathbf{g}, \tag{A4}$$

reassuring perfect conservation of three dimensional momentum.

The same concept can be applied to fluxes through the boundary of the control volume, leading to the concept of surface partial differential operators, $\boldsymbol{\nabla}_{\mathrm{s}}$ and $\boldsymbol{\nabla}_{\mathrm{n}}$. Figure A2 shows a momentum flux $\boldsymbol{\nabla} \cdot \mathbf{m}$, which could represent convective momentum transport $\boldsymbol{\nabla} \cdot (h\,\overline{\mathbf{u}}\,\overline{\mathbf{u}})$ or lateral pressure gradient $\frac{1}{2\,\rho}\boldsymbol{\nabla}\,(p_{\mathrm{b}}\,h)$. The net flux leaving the control volume can be calculated as the sum of all fluxes and leads to the definition of the divergence operator,

$$\boldsymbol{\nabla} \cdot \mathbf{m} = \frac{1}{S_{\mathrm{b}}}\left(\mathbf{m}_{\mathrm{out}} - \mathbf{m}_{\mathrm{in}}\right), \tag{A5}$$

in accordance to Gauss' theorem. S_{b} is the surface area of the cell. For the exact formulation in terms of finite areas, the reader is refereed to Rauter and Tuković (2018). Note that the net flux is a three dimensional vector without any particular direction in relation to the basal surface. Hence, it has a surface tangential component and a surface normal component. It can thus be treated similar to gravitational acceleration, yielding the surface normal component

$$\boldsymbol{\nabla}_{\mathrm{n}} \cdot \mathbf{m} = \mathbf{n}_{\mathrm{b}}\,\|\boldsymbol{\nabla}_{\mathrm{n}} \cdot \mathbf{m}\| = \mathbf{n}_{\mathrm{b}}\,(\mathbf{n}_{\mathrm{b}} \cdot \boldsymbol{\nabla} \cdot \mathbf{m}) = (\mathbf{n}_{\mathrm{b}}\,\mathbf{n}_{\mathrm{b}}) \cdot \boldsymbol{\nabla} \cdot \mathbf{m}. \tag{A6}$$

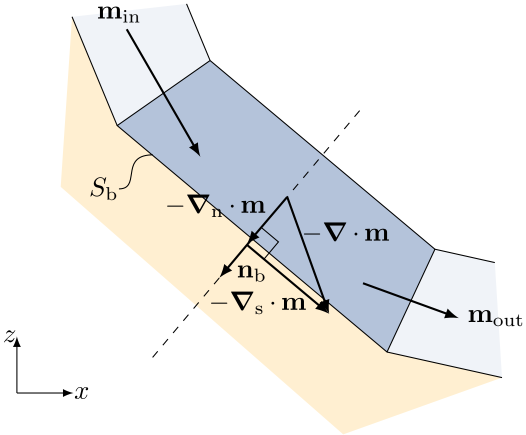

$\mathbf{m}_{\text{in}}$
S_b
$-\boldsymbol{\nabla}_\text{n} \cdot \mathbf{m}$
$-\boldsymbol{\nabla} \cdot \mathbf{m}$
$\mathbf{n}_\text{b}$
$-\boldsymbol{\nabla}_\text{s} \cdot \mathbf{m}$
$\mathbf{m}_{\text{out}}$
z
x

Figure A2. Splitting net fluxes into a surface tangential and surface normal part with simple projections to the surface normal vector $\mathbf{n}_\text{b}$. Note that the flux, entering the control volume, $-\boldsymbol{\nabla} \cdot \mathbf{m}$, is shown.

and the surface tangential component

$$\boldsymbol{\nabla}_\text{s} \cdot \mathbf{m} = \boldsymbol{\nabla} \cdot \mathbf{m} - \boldsymbol{\nabla}_\text{n} \cdot \mathbf{m} = \boldsymbol{\nabla} \cdot \mathbf{m} - (\mathbf{n}_\text{b}\,\mathbf{n}_\text{b}) \cdot \boldsymbol{\nabla} \cdot \mathbf{m} = (\mathbf{I} - \mathbf{n}_\text{b}\,\mathbf{n}_\text{b}) \cdot \boldsymbol{\nabla} \cdot \mathbf{m}. \tag{A7}$$

Surface normal and tangential components contribute to local acceleration and basal pressure for reasons discussed in terms of gravitational acceleration. Three dimensional conservation is reassured for fluxes as well, if the three dimensional flux $\boldsymbol{\nabla} \cdot \mathbf{m}$ is conservative. Finally, we want to note that velocity is a three-dimensional vector field and its direction is not fixed a priori. However, velocity will always be aligned with the surface because only surface tangential components are present in the respective conservation equation.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. We thank Mark Olesen and Andrew Heather (ESI-OpenCFD) for help regarding OpenFOAM and review of our solver code. We thank Matthias Granig and Felix Oesterle (WLV) for support regarding SamosAT and for providing the respective software. We thank our colleges, Iman Bathaeian, Jan-Thomas Fischer and Fabian Schranz for valuable comments on the manuscript. We thank the OpenFOAM, ParaView and QGIS communities for sharing their code and providing helpful advice. We gratefully acknowledge the financial support by the OEAW project "beyond dense flow avalanches". The computational results presented have been achieved (in part) using the HPC infrastructure LEO of the University of Innsbruck.