faSavageHutterFOAM 1.0: Depth-integrated simulation of dense snow avalanches on natural terrain with OpenFOAM

Matthias Rauter[1,2], Andreas Kofler[2], Andreas Huber[3], and Wolfgang Fellin[1]

[1]University of Innsbruck, Institute of Infrastructure, Division of Geotechnical and Tunnel Engineering
[2]Department of Natural Hazards, Austrian Research Centre for Forests (BFW), Innsbruck, Austria
[3]University of Innsbruck, Institute of Infrastructure, Division of Hydraulic Engineering

Abstract.

Numerical models for dense snow avalanches have become a central part in hazard zone mapping and mitigation. Several commercial and free applications, which are used on a regular basis, implement such models. In this study we present a tool based on the open-source toolkit OpenFOAM® as an alternative to the established solutions. The proposed tool implements a depth-integrated shallow flow model in accordance to current practice. The solver combines advantages of the extensive OpenFOAM infrastructure with popular models from the avalanche community. OpenFOAM allows assembling of custom physical models with build-in primitives and implements the numerical solution at a high level. OpenFOAM supports an extendable solver structure, making the tool well-suited for future developments and rapid prototyping. We introduce the basic solver, implementing an incompressible, single-phase model for natural terrain, including entrainment. The respective workflow, consisting of meshing, pre-processing, numerical solution and post-processing, is presented. We demonstrate data transfer from and to a geographic information system (GIS), to allow a simple application in practice. The tool-chain is based entirely on open-source applications and libraries and can be easily customized and extended. Simulation results for a well documented avalanche event are presented and compared to previous numerical studies and historical data.

1 Introduction

Numerical avalanche modelling has become an important and well-accepted ingredient to hazard zone mapping. All popular tools rely on depth-integrated flow models (Pudasaini and Hutter, 2007) and only a few academic exceptions are known (Domnik et al., 2013; Kröner, 2013; von Boetticher et al., 2016, 2017; Barker and Gray, 2017). Depth-integrated flow models, widely known as Shallow Water Equations, have a long tradition in hydraulic modelling (e.g., Vreugdenhil, 1994), dating back to Barré de Saint-Venant (1871). This approach is commonly applied in academia and practice because it reduces the computational effort to a level, where physical simulations of realistic flows are feasible. The first application to gravitational mass flows is attributed to Grigorian et al. (1967), the first formal derivation and analysis of the underlying model to Savage and Hutter (1989, 1991). Since then, the mechanical model has been continuously improved and extended to e.g., simple, two-dimensional surfaces (Greve et al., 1994), complex, shallow surfaces (Gray et al., 1999) or curved and twisted flow paths (Pudasaini et al., 2005a, b). Finally, respective models have been adapted to natural, i.e. arbitrary but mildly curved terrain,

making simulations of real case avalanches possible. Denlinger and Iverson (2004) proposed a model embedded in an ordinary Cartesian coordinate system as an alternative to the complex curvilinear coordinate system used by Savage and Hutter (1989, 1991). Bouchut and Westdickenberg (2004), Hergarten and Robl (2015) and recently Rauter and Tuković (2018) follow a similar approach. Christen et al. (2010) apply a non-orthogonal local coordinate system (Fischer et al., 2012), however, without

5   incorporating the respective correction terms (Hergarten and Robl, 2015). A Lagrangian solution, which has some advantages for natural terrain, has been presented by Hungr (1995) and later on by Sampl and Zwinger (2004) and Sampl and Granig (2009).

Beside improvement of the underlying mechanical model[1] various physical processes have been added to governing equations, such as multiple phases (e.g., Pudasaini, 2012; Kowalski and McElwaine, 2013; Iverson and George, 2016), entrainment

10   (e.g., Issler, 2014), improved basal friction relations (e.g., Voellmy, 1955; Norem et al., 1987; Pouliquen and Forterre, 2002; Bartelt et al., 2006; Issler and Gauer, 2008; Baker et al., 2016; Rauter et al., 2016), (for a review see Ancey, 2007), compressibility (e.g., Iverson and George, 2014; Bartelt et al., 2015) or thermodynamic processes (e.g., Vera Valero et al., 2015).

There are several numerical methods to solve the respective mathematical equations. Basically, most methods can be classified as finite difference method (e.g., Wang et al., 2004), finite element method (e.g., Hanert et al., 2005), finite volume method

15   (e.g., Christen et al., 2010) or as Lagrangian particle method (e.g., Sampl and Granig, 2009). Specialised differencing schemes (e.g., upwind, TVD, NVD) prevent oscillations (e.g., Jasak et al., 1999).

Shallow granular flow models have been carefully validated over the last few decades. This includes back-calculations of small scale experiments (for a review see Pudasaini and Hutter, 2007), large scale experiments (e.g., Christen et al., 2010), historic snow avalanches (e.g, Fischer et al., 2015) and rock avalanches (e.g., Mergili et al., 2017). Shallow flow models

20   have various weaknesses, such as the limitation to mildly curved terrain or the missing resolution in surface normal direction. However, they have proven to be a good trade-off between accuracy and computing time and thus useful for many applications.

Shallow flow models gained popularity through commercial software packages: DAN (Hungr, 1995), SamosAT (Sampl and Zwinger, 2004), FLATModel (Medina et al., 2008) and RAMMS (Christen et al., 2010) implement such models and are used regularly in practice. Open-source alternatives include TITAN2D (Pitman et al., 2003; Patra et al., 2005), r.avaflow (Mergili

25   et al., 2012, 2017) and an extension to the CFD-toolkit (computation fluid dynamics) GERRIS (Hergarten and Robl, 2015). From an academic viewpoint, open-source applications have various advantages over their commercial counterparts; e.g. users can view and modify the source code to gain a better understanding of the software and adapt the flow model without re-implementing basic models and numerical methods from scratch.

Geographic Information Systems (GIS) are commonly applied in hazard zone mapping. Therefore numerical simulation

30   tools are usually incorporated or linked to these systems to streamline the respective workflow. GIS allows user friendly data input, post-processing and production of publication quality maps.

---

[1]We distinguish between mechanical model, in the sense of basic conservation equations and their reformulation, e.g. in terms of depth-integration and process models, in the sense of closure of governing equations with e.g. constitutive models. The combination of the mechanical model and all closures is called flow model or physical model throughout this work.

Recently, Rauter and Tuković (2018) proposed a shallow granular flow model, expressed in terms of surface partial differential equations (Deckelnick et al., 2005; Tuković and Jasak, 2012) and presented an open-source implementation based on the CFD-toolkit OpenFOAM® (OpenCFD Ltd., 2004). The underlying mechanical model is widely similar to the classic Savage and Hutter (1989, 1991) model and its derivations.

5   One particular advantage of an OpenFOAM solver is the well-designed, object-oriented source code. This makes the code cleaner than comparable solutions as it hides implementation details, such as numerical schemes, I/O or inter-process communication, behind well defined interfaces. The top-level solver mimics the tensorial notation of partial differential equations and specific implementations of e.g., interpolation schemes, are exchangeable without changing the top-level source code. This enables separation of physical models and numerical solution, which allows a streamlined interdisciplinary development process.

10   Process models, e.g. entrainment and basal friction can be incorporated similarly, keeping the source code clean and easy to extend.

    The OpenFOAM solver, presented in here, implements an incompressible single-phase model including various basal friction and entrainment closures. This model is, to some extent, suitable for dense snow avalanches and constitutes the baseline for complex flow models, as employed by e.g. Bartelt et al. (2015) or Mergili et al. (2017). Moreover, the underlying method has

15   been developed to simplify coupling with three-dimensional ambient flows (Tuković and Jasak, 2012; Marschall et al., 2014; Dieter-Kissling et al., 2015a, b; Pesci et al., 2015), which enables development of models for mixed snow avalanches (e.g., Sampl and Zwinger, 2004) and turbidity currents (e.g., Huang et al., 2005).

    The purpose of this paper is to present the capability of the new OpenFOAM solver and the Rauter and Tuković (2018) model. The solver is evaluated and validated for snow avalanches on natural terrain. We present the basic flow model, as

20   well as methods and tools to incorporate natural terrain and GIS data in OpenFOAM simulations. Also export of OpenFOAM results to a GIS for post-processing and visualisation is demonstrated. Results for a well documented avalanche event are presented and compared to historical records and results of SamosAT. All underlying source code (except SamosAT) and data are available for free to encourage reproduction, improvement and cross validation.

2   Method

25   2.1   Flow model

Historically, shallow granular flow models have been set up in surface aligned, curvilinear coordinates, leading to a two-dimensional system of partial differential equations (e.g., Savage and Hutter, 1989, 1991). Rauter and Tuković (2018) follow a different approach (see also, e.g., Denlinger and Iverson, 2004; Bouchut and Westdickenberg, 2004; Hergarten and Robl, 2015) and formulate the mechanical model in terms of surface partial differential equations (SPDEs, e.g., Deckelnick et al.,

30   2005). Respective SPDEs are defined on a surface Γ, embedded in three-dimensional space, which represents the mountain topography. This approach, popular in the thin liquid film community (e.g., Craster and Matar, 2009), avoids transformations into the surface aligned coordinate system and thus complex metric tensors. Considering the relative shallowness of the avalanche, it can be treated as thin layer flowing along the mountain surface. The governing equations describe the motion
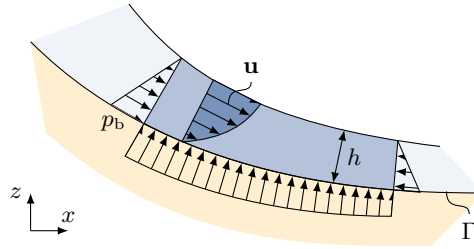
Figure 1. Definition of velocity $\mathbf{u}$, flow thickness h and basal pressure p_b on a control volume. A hydrostatic and linear pressure distribution is assumed. The shape of the velocity profile is commonly ignored in governing equations (Baker et al., 2016). Flow thickness h is measured normal on the basal surface Γ. The curvature radius of the surface Γ is assumed to be much bigger than the flow thickness.

of the avalanche in three-dimensional space along this surface. Consequently, velocity is a three-dimensional vector field and contains all information on flow direction and respective effects, such as centrifugal forces. Resulting SPDEs can be solved with various methods, e.g. finite element method (e.g., Olshanskii et al., 2009) or finite area method, a modified finite volume method (Tuković and Jasak, 2012).

5   A basic shallow granular flow model can be written in terms of surface partial differential equations as[2]

$$\frac{\partial h}{\partial t} + \boldsymbol{\nabla} \cdot (h\,\overline{\mathbf{u}}) = \frac{\dot{q}}{\rho}, \tag{1}$$

$$\frac{\partial (h\,\overline{\mathbf{u}})}{\partial t} + \boldsymbol{\nabla}_\mathrm{s} \cdot (h\,\overline{\mathbf{u}}\,\overline{\mathbf{u}}) = -\frac{1}{\rho}\boldsymbol{\tau}_\mathrm{b} + h\,\mathbf{g}_\mathrm{s} - \frac{1}{2\rho}\boldsymbol{\nabla}_\mathrm{s}\,(h\,e\,p_\mathrm{b}), \tag{2}$$

$$\boldsymbol{\nabla}_\mathrm{n} \cdot (h\,\overline{\mathbf{u}}\,\overline{\mathbf{u}}) = h\,\mathbf{g}_\mathrm{n} - \frac{1}{2\rho}\boldsymbol{\nabla}_\mathrm{n}\,(h\,p_\mathrm{b}) - \frac{1}{\rho}\mathbf{n}_\mathrm{b}\,p_\mathrm{b}. \tag{3}$$

Variables and mathematical operators are explained below. Equations (1) to (3) are equivalent to a Savage-Hutter like system,
10 consistently extended to complex but mildly curved terrain and entrainment. Note that the earth pressure theory (e.g., Savage and Hutter, 1989, 1991) has been replaced with the hydrostatic pressure assumption, as in most practical applications (e.g., Christen et al., 2010). The notation as SPDE makes extension to complex terrain straight-forward and implementation into SPDE environments, e.g. OpenFOAM, possible. A formal derivation is given by Rauter and Tuković (2018). Here, we aim to deliver a short and descriptive introduction.

15 Equation (1) represents the depth-integrated continuity equation, Eq. (2) the surface tangential momentum conservation equation and Eq. (3) its surface normal counter part, defined in all points $\mathbf{x}_\mathrm{b}$ on the surface $\Gamma \subset \mathbb{R}^3$, representing the mountain surface. The time is denoted as t. The unknown fields are the surface normal flow thickness $h(\mathbf{x}_\mathrm{b})$ (see Fig. 1), the depth-averaged flow velocity $\overline{\mathbf{u}}(\mathbf{x}_\mathrm{b}) \in \mathbb{R}^3$, defined as

$$\overline{\mathbf{u}}(\mathbf{x}_\mathrm{b}) = \frac{1}{h(\mathbf{x}_\mathrm{b})} \int\limits_0^{h(\mathbf{x}_\mathrm{b})} \mathbf{u}(\mathbf{x}_\mathrm{b} - \mathbf{n}_\mathrm{b}\,z')\,\mathrm{d}z', \tag{4}$$

---

[2]Multiplications between vectors represent the outer product $\mathbf{u}\,\mathbf{v} = \mathbf{u} \otimes \mathbf{v}$

and the basal pressure $p_b(\mathbf{x}_b)$. In the framework of SPDEs, the normal vector field $\mathbf{n}_b(\mathbf{x}_b) \in \mathbb{R}^3$ of the surface Γ is sufficient to describe all major curvature effects. The density ρ is assumed to be constant. The gravitational acceleration $\mathbf{g}$ is split into a surface tangential component,

$$\mathbf{g}_s = (\mathbf{I} - \mathbf{n}_b\,\mathbf{n}_b) \cdot \mathbf{g}, \tag{5}$$

and surface normal component,

$$\mathbf{g}_n = (\mathbf{n}_b\,\mathbf{n}_b) \cdot \mathbf{g}. \tag{6}$$

The gradient operator $\boldsymbol{\nabla}$ denotes the three-dimensional derivative along the surface (Deckelnick et al., 2005). If the responding result is a three-dimensional vector field (e.g. gradient of a scalar field or divergence of a tensor field), it can be split, similar to the gravitational acceleration, into a surface tangential component,

$$\boldsymbol{\nabla}_s = (\mathbf{I} - \mathbf{n}_b\,\mathbf{n}_b) \cdot \boldsymbol{\nabla}, \tag{7}$$

and surface normal component,

$$\boldsymbol{\nabla}_n = (\mathbf{n}_b\,\mathbf{n}_b) \cdot \boldsymbol{\nabla}. \tag{8}$$

Surface tangential and normal components contribute to local acceleration and basal pressure, respectively. This follows from the assumption that movement is constrained in surface normal direction, which is enforced by a mechanical force, namely the basal pressure. For simply curved surfaces, the given relation matches the model of Greve et al. (1994), as shown by Rauter and Tuković (2018).

There are various user-selectable models, describing basal friction $\boldsymbol{\tau}_b(\mathbf{x}_b)$ and entrainment rate $\dot{q}(\mathbf{x}_b)$, to close the system of equations. To reassemble the traditional model (often called Voellmy or Voellmy-Salm model, Christen et al., 2010), as applied by e.g. Fischer et al. (2015), the basal friction is described following Voellmy (1955),

$$\boldsymbol{\tau}_b = \mu\, p_b \frac{\overline{\mathbf{u}}}{|\overline{\mathbf{u}}| + u_0} + \frac{\rho\, g}{\xi} |\overline{\mathbf{u}}|\,\overline{\mathbf{u}}. \tag{9}$$

Therein μ and ξ are constant parameters, although they may depend on avalanche size and surface roughness (Salm et al., 1990) or flow regime (Köhler et al., 2016). The small value u_0 avoids divisions by zero and regularizes the relation near stillstand, where the original function is discontinuous. This regularisation, combined with the employed time integration scheme (implicit three-level second-order, Ferziger and Peric, 2002), leads to a well-defined behaviour in the runout zone, where the velocity is nearly zero (Rauter and Tuković, 2018). Previously, this issue has been addressed with operator splitting and explicit stress reduction (e.g., Mangeney-Castelnau et al., 2003; Zhai et al., 2015; Mergili et al., 2017), which is not required in the proposed scheme.

The entrainment rate is calculated, based on an empirical erosive entrainment model, as

$$\dot{q} = \begin{cases} \dfrac{\boldsymbol{\tau}_b \cdot \overline{\mathbf{u}}}{e_b} & \text{for} \quad h_{msc} > 0, \\ 0 & \text{for} \quad h_{msc} = 0, \end{cases} \tag{10}$$

where e_b is the specific erosion energy (Fischer et al., 2015). Entrainment is restricted by the available mountain snow cover thickness h_msc. The initial mountain snow cover thickness is calculated following Fischer et al. (2015), using a linear approach,

$$h_\mathrm{msc}(z) = \left(H_\mathrm{msc}(z_0) + \frac{\partial H_\mathrm{msc}}{\partial z}(z - z_0) \right) \cos(\zeta), \tag{11}$$

5 where z is the surface elevation and z_0 the elevation of a reference station, which has to be provided by the user, alongside with the base value $H_\mathrm{msc}(z_0)$ and the growth rate $\frac{\partial H_\mathrm{msc}}{\partial z}$. ζ is the angle between the gravitational acceleration and the surface normal vector. Its further evolution is described by the conservation equation

$$\frac{\partial h_\mathrm{msc}}{\partial t} = -\frac{\dot{q}}{\rho}. \tag{12}$$

Undershoots, i.e. $h_\mathrm{msc} < 0$, are prevented with a regularisation similar to Eq. (9).

10 The governing equations are solved with an implicit, conservative, finite area method (Rauter and Tuković, 2018), using the respective OpenFOAM library (Tuković and Jasak, 2012). The finite area method is similar to the well-known finite volume method (e.g. Jasak, 1996), however with appropriate differential operators for SPDEs, Eqs. (7) and (8). We apply first- (upwind scheme) and second-order accurate spatial differencing schemes. First-order schemes converge slower in terms of mesh refinement due to their high numerical diffusivity. However, numerical diffusivity effectively prevents oscillations and increases

15 the stability of the solver. Oscillations in second-order accurate simulations are prevented with a normalised variable diagram (NVD) scheme for unstructured meshes, known as Gamma scheme (Jasak et al., 1999). NVD schemes blend upwind and a higher order scheme to combine advantages of both methods.

As mentioned before, OpenFOAM utilises capabilities of C++ to make top-level source code appear similar to the tensor notation of partial differential equations. The conservation equation (1), e.g., can be solved with the following lines of code

20 using OpenFOAM:

```
faScalarMatrix hEqn
(
    fam::ddt(h)
  + fam::div(phis, h)
 ==
    dqdt/rho
);
hEqn.solve();
```

`phis` is the velocity edge field (see Rauter and Tuković, 2018, for details) and `dqdt` the source term incorporating entrainment.

30 Momentum conservation equations (2) and (3) look similar and conservation equations for arbitrary fields (e.g. random kinetic energy, Bartelt et al., 2015) can be added with the same syntax.

2.2 Simulation evaluation

We use an established implementation of the same flow model, SamosAT (version 2017_07_05) (Sampl and Zwinger, 2004; Sampl and Granig, 2009), for comparison. The main difference between SamosAT and the presented OpenFOAM solver is the solution method. SamosAT solves similar governing equations, slightly adapted to fit into the respective framework, with

5   smoothed-particle hydrodynamics (SPH). This approach follows a Lagrangian description, making handling of complex terrain simpler (Sampl and Zwinger, 2004). Therefore, SamosAT provides an excellent reference to validate avalanche models for complex terrain. The second term on the right hand side of Eq. (3) was deactivated in OpenFOAM computations to reassemble the mechanical model as implemented in SamosAT. This term is usually small and can be safely neglected (Rauter and Tuković, 2018). However, it is shown in equations to preserve the similarity between Eqs. (2) and (3).

10   We compare simulations using the $1\,\mathrm{kPa}$-isoline of the dynamic peak pressure, defined as

$$p_{\mathrm{dyn}}(\mathbf{x}_{\mathrm{b}}) = \max_{t}\left(\rho\,|\overline{\mathbf{u}}(\mathbf{x}_{\mathrm{b}},t)|^2\right). \tag{13}$$

Definitions of hazard zones are based on this threshold in many European countries (Jóhannesson et al., 2009) and therefore often used for evaluation of respective models (e.g., Fischer et al., 2015; Rauter et al., 2016).

In addition to the comparison with a reference implementation, we present a comparison with historical records from a

15   catastrophic event. A common method to document avalanches is the delineation of deposition. This information is also available for the presented case study. Deposition processes are not explicitly included in the flow model due to depth-integration. However, the general form and size of the deposition should be reproduced by the model to be useful for hazard zone mapping. This is problematic in some implementations, e.g., SamosAT, due to missing regularisation of the friction term, but possible with the proposed method.

20   We apply model parameters (μ, ξ, e_{b}) optimised for SamosAT (Fischer et al., 2015) and the comparison is conducted on a qualitative level.

Finally, we evaluate OpenFOAM simulations with regard to convergence during mesh refinement to give a quantitative estimation of numerical uncertainties as recommended by Roache (1997). The numerical solution should converge to the unknown analytical solution with increasing grid resolution and the numerical uncertainty should decay with the order of the

25   applied method. Richardson extrapolation allows to estimate the numerical uncertainty, using results of three different meshes. This way, the expected convergence can be verified and the numerical uncertainty quantified.

2.3 Simulation setup

The precondition to conduct simulations in OpenFOAM is a mesh, describing the geometry of the problem. For SPDEs, e.g. shallow flow models, a surface mesh, matching the slope topography, is sufficient and no volume mesh is required. In

30   practice, however, three-dimensional meshing tools can be used to create a volume mesh, the boundary of which can be used as surface mesh.

Topography is usually available as digital elevation model (DEM) in GIS formats, yielding elevation on a regular two-dimensional grid. The relevant part of the topography is re-sampled with cubic splines, triangulated and stored as STL file
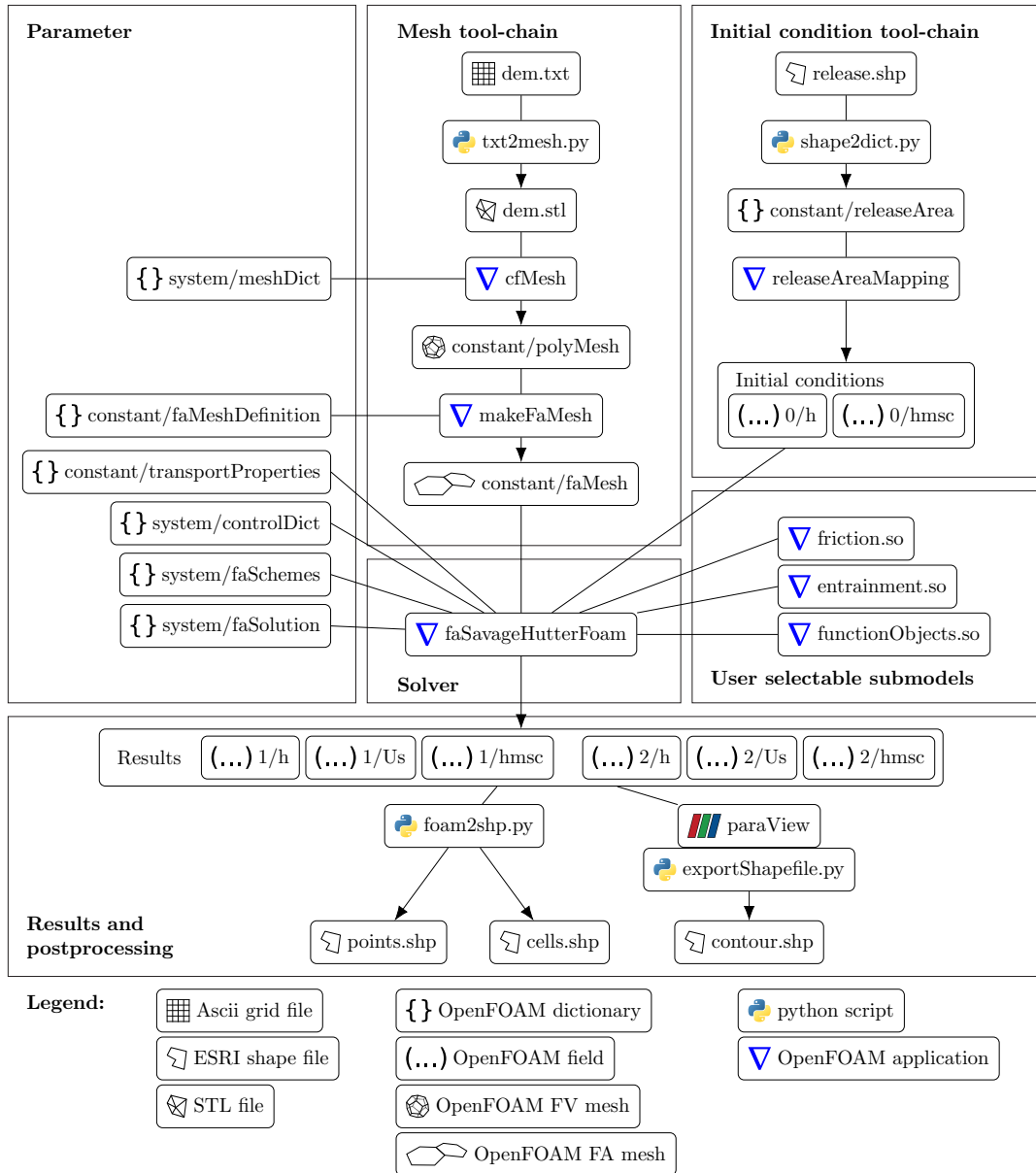
Figure 2. Simulation setup and tool-chain. The tool-chain consists exclusively of open-source applications. Individual applications and process models can be replaced with custom ones. Parameters for python scripts are provided via command line interface. Parameters for OpenFOAM applications are provided through OpenFOAM dictionaries. Domain decomposition and reconstruction, which is handled by separate applications, is not shown. OpenFOAM reads initial conditions from the folder "0" and writes results to folders, named after the respective timestep ("1", "2", etc.). Details on OpenFOAM formats can be found in OpenCFD Ltd. (2004).

(e.g., Kai et al., 1997) to prepare it for meshing. We chose the meshing application cfMesh (Juretić, 2015), because of its good integration in OpenFOAM and its clean boundary meshes. cfMesh requires a closed triangulated surface to create a volume mesh. This is the case for all general purpose meshing tools and cfMesh can be easily replaced in our tool-chain, for example with Netgen (Schöberl, 1997) (see Rauter and Tuković, 2018, for an application). The closed surface can be assembled from a

5   triangulation of the mountain surface, sidewalls and the respective top boundary. The resulting surface and volume mesh are presented in Figs. 3b and 3c. Refinement near the mountain surface reduces the amount of required volume cells, while keeping the number of surface cells high. The resulting mesh is also valid for three-dimensional simulations with e.g. Navier Stokes Equations, as conducted by e.g. Sampl and Zwinger (2004); Dutykh et al. (2011); Kröner (2013); von Boetticher et al. (2016, 2017); Huang et al. (2005). The boundary mesh, describing the mountain surface, is shown in Fig. 3d. The shallow flow model

10 is solved on this surface mesh. We used polygonal-dominated (volumetric polyhedral-dominated, respectively) meshes for simulations because of stability and accuracy reasons (Juretić, 2005). Triangular (volumetric tetrahedral, respectively) meshes have been evaluated as well. However, second-order accurate simulations on triangular meshes failed, while first-order accurate simulations are virtually identical to the respective simulations on polygonal-dominated meshes.

The release area, acting as initial condition, is provided as a polygon in ESRI shapefile format (ESRI, 1998). To find all

15 surface cells within the given polygon, the Hormann and Agathos (2001) algorithm as implemented in OpenFOAM is applied. The mountain snow cover h_{msc} of respective cells is then transferred to the flow thickness h to create a suitable initial condition. The release area for our case study, taken from Fischer et al. (2015), is shown in Fig. 7a as polygon and as set of surface cells in Fig. 3d.

The solver reads the surface mesh and initial conditions, as well as physical models, numerical schemes and constants to

20 initialise the simulation (see Fig. 2). The respective entries can be found in the designated locations, according to usual practice in OpenFOAM (OpenCFD Ltd., 2004). The solver can run on multiple processors using domain decomposition (Weller et al., 1998) and message passing interface (MPI).

User defined friction and entrainment models can be loaded at run-time, meaning that the user does not have to recompile the solver to add a custom friction or entrainment model. The same is the case for general purpose functions which are triggered

25 at the end of every time-step. Here, we used this interface to calculate and record the dynamic peak pressure at run-time, without the necessity to save multiple time-steps or to change solver source code. Similar functions can be used to check mass, momentum or energy conservation, record specific data (e.g. time-line at a certain point) or to manipulate fields during run-time, e.g. to trigger secondary slabs.

Simulation results are written to hard disk in the usual OpenFOAM file format (OpenCFD Ltd., 2004) for post-processing,

30 evaluation and simulation restart. The simulation setup, all involved applications and all intermediate and final files are presented in Fig. 2. The tool-chain is modularly assembled from various open-source applications. Single modules, such as mesher, solver or friction model, can be easily replaced.
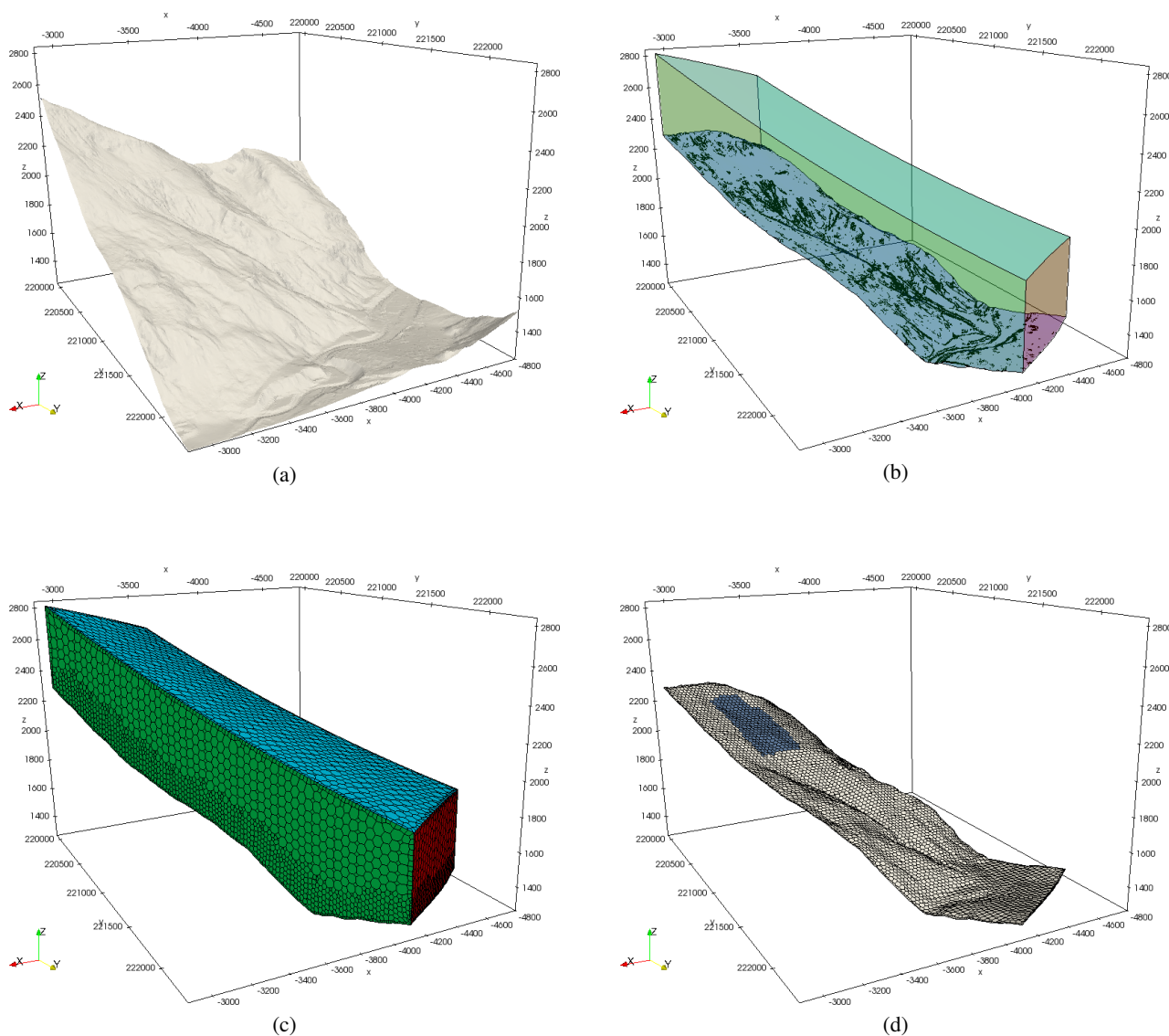
Figure 3. Meshing tool-chain: The terrain data is usually available as raster data (a). Triangulation of the relevant area and adding walls and a top boundary yields a closed triangulated surface (b, sharp edges are highlighted black). This surface can be processed by most meshing tools, here we apply cfMesh to get a polyhedral-dominated finite volume mesh (c). The bottom boundary surface of the finite volume mesh builds the foundation for the finite area mesh used for simulations (d). Note that we show a very coarse mesh for the sake of visibility of edges. Terrain data: Amt der Tiroler Landesregierung (AdTLR). EPSG: 31254.
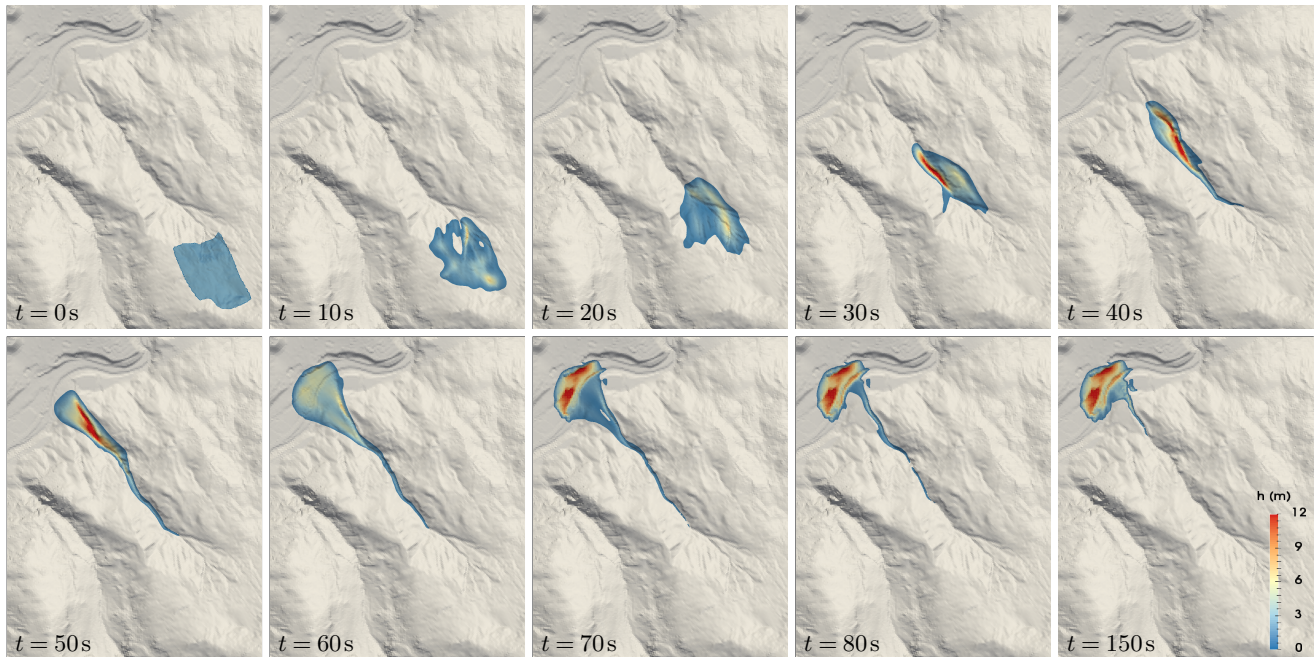
Figure 4. Time series of an OpenFOAM simulation with mean cell size $\Delta = 7.45\,\mathrm{m}$ and first-order interpolations in ParaView. The colour scale represent flow thickness, which is clipped at $0.5\,\mathrm{m}$. Terrain data: AdTLR.

2.4 Post-processing and visualisation

Post-processing and visualisation of OpenFOAM simulations is commonly performed using ParaView® (Ahrens et al., 2005; Ayachit, 2015) (see Figs. 3, 4 and 5). ParaView is an open-source data analysis and visualisation application. It can read and visualise OpenFOAM files and they can be used for further operations, such as the calculation of contour lines. To integrate GIS applications in post-processing, results can be exported to common GIS file formats. Contour lines can be exported to ESRI shapefile format with a custom python extension based on the library pyshp (Figs. 6c, 7 and 9). Alternatively, individual cells and respective field values, can be exported as polygons (Fig. 6a) or points (Fig. 6b) to ESRI shapefiles.

To generate regular raster files, the unstructured OpenFOAM mesh and associated fields have to be mapped to a structured Cartesian grid (Figs. 6d and 9). These and other approaches allow an almost seamless integration into general purpose GIS applications, as shown in the following case study. Here, we utilise foam-extend 4.0 with a custom solver, python 2.7.12 with numpy 1.11.0, scipy 0.17.0 and pyshp 1.2.3 for shapefile export, ParaView 5.0.1 and QGis 2.8.6.
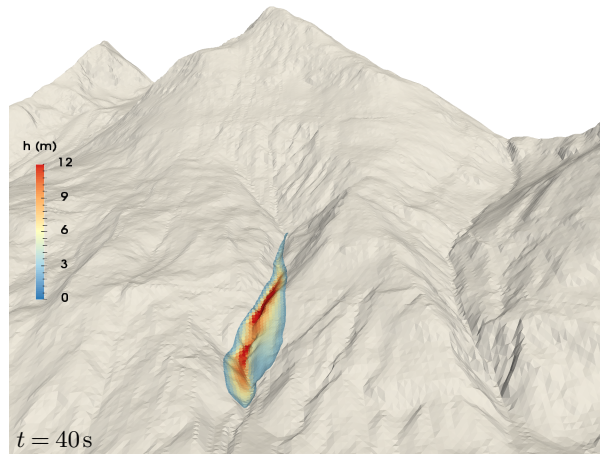
Figure 5. Perspective view on the OpenFOAM simulation with mean cell size $\Delta = 7.45\,\mathrm{m}$ and first-order interpolations in ParaView. The colour scale represent flow thickness, which is clipped at $0.5\,\mathrm{m}$. Terrain data: AdTLR.

3   Case study

In this work we focus on the Wolfsgruben avalanche. The event from the 13th March 1988, when the avalanche struck inhabited areas has been repeatedly used as benchmark for avalanche simulations, latest by Fischer et al. (2015). We chose this example because the respective data is freely available, making reproduction and cross validation possible.

The mountain snow cover thickness for the specific event can be described with parameters $H_{\mathrm{msc}}(z_0) = 1.61\,\mathrm{m}$, $z_0 = 1289\,\mathrm{m}$, $\frac{\partial H_{\mathrm{msc}}}{\partial z} = 8 \cdot 10^{-4}$. Physical parameters to reassemble the runout properly are $\mu = 0.26$, $\xi = 8650\,\mathrm{m\,s^{-2}}$ and $e_{\mathrm{b}} = 11500\,\mathrm{J\,kg^{-1}}$. These parameters have been optimised in a previous study using SamosAT (Fischer et al., 2015).

Numerical parameters for OpenFOAM (see Rauter and Tuković, 2018) have been chosen such, that they do not influence the results, while keeping the solver as stable as possible. The appropriate mesh resolution for OpenFOAM has been identified using a mesh refinement study, which is presented alongside the results. The simulation duration has been set to $150\,\mathrm{s}$. This duration is sufficient to reach still-stand in the runout zone ($\overline{\mathbf{u}} < 10^{-5}\,\mathrm{m\,s^{-1}}$) and thus virtually unchanging deposition. We decomposed the simulation domain into four parts for OpenFOAM and all simulations have been conducted on a Quadcore Intel Core i7-7700K @ $4.20\,\mathrm{GHz}$ and $32\,\mathrm{GB}$ DDR4 Ram @ $2.667\,\mathrm{GHz}$.

SamosAT utilises a grid with $5\,\mathrm{m}$ resolution and we follow recommendations in terms of appropriate particle numbers and other numerical parameters. The interpolation method has been varied between interpolation on grid (SPH-mode 0) and interpolation on particles (SPH-mode 1) to get an insight into the numerical uncertainty.

ParaView renderings are presented in Fig. 4 for multiple time-steps, showing the dynamic behaviour of the avalanche. A perspective ParaView rendering is shown in Fig. 5. The avalanche follows the narrow channel directly beneath the release area. Small portions of the avalanche overflow the left and right humps in some simulations, which can be seen in the peak dynamic pressure, Fig. 7.
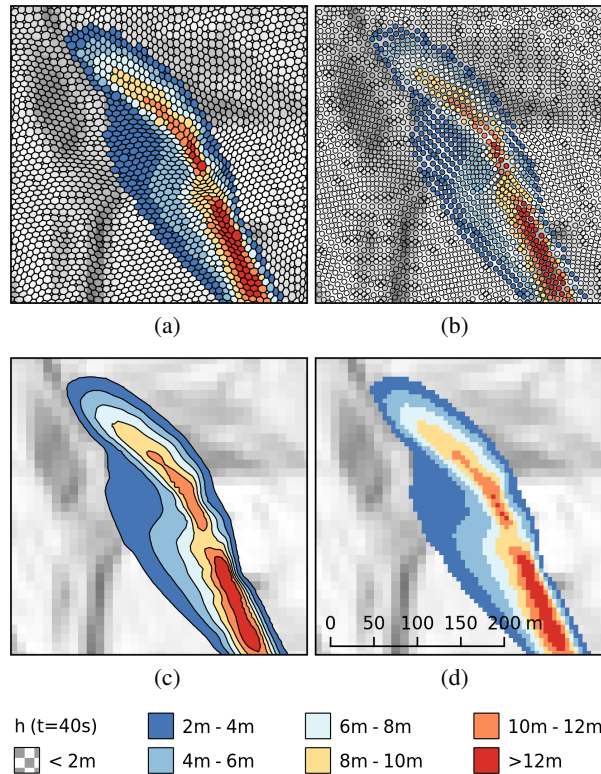
Figure 6. The flow thickness field h at time $t = 40\,$s for a simulation with mean cell size $\Delta = 7.45\,$m, first-order interpolations. The figure shows four methods to export and analyse results in GIS: Export of cells as polygons (a). Export of cell centres as points (b). Export of contour lines as polygons (c). Remapping of the unstructured finite area mesh to regular raster (d). The raster has been created by converting point data to raster file in QGIS. The resolution of the DEM is $10\,$m, results have been mapped to a $5\,$m grid. Terrain data: AdTLR.

The results at time-step $t = 40\,$s have been exported to QGIS using various methods, see Fig. 6. Affected areas (i.e. $1\,$kPa-isolines), as predicted by OpenFOAM and SamosAT are shown in Fig. 7. Variations due to different interpolation schemes are shown for both implementations, to give an insight into the numerical uncertainty.

The influence of the mesh resolution on the affected area is shown in Fig. 8 for the OpenFOAM solver. Respective mean cell sizes, an estimation of the numerical uncertainty following Roache (1997) and execution times (excluding time for mesh generation, which may take several minutes) are presented in Tab. 1. Here, the runout is defined as the length of the central avalanche path (see Fig. 8) within the affected area. The central avalanche path has been taken from Fischer et al. (2015). The mean cell size is defined as the square root of the mean cell area. For comparison, execution times for SamosAT are $98\,$s (SPH-mode 0) and $368\,$s (SPH-mode 1), respectively. One should keep in mind that SamosAT utilises solely a single processor core while OpenFOAM utilises all available cores. Moreover, execution times should be seen as rough estimations because they depend on various factors, such as the number of saved time-steps, debug messages and compile options.

Table 1. Mesh size, runout, error estimation and execution time for different OpenFOAM simulations. Base cell size and refinements refer to parameters of cfMesh.

interpolation	base cell size	refinements	number of cells	mean cell size	runout	num. uncertainty	exec. time
1st-order	40 m	2	40 899	7.45 m	2 145 m		173 s
1st-order	30 m	2	72 166	5.61 m	2 137 m	46 m	396 s
1st-order	20 m	2	161 364	3.75 m	2 112 m	6 m	1 261 s
1st-order	15 m	2	285 892	2.82 m	2 107 m		3 051 s
2nd-order	40 m	2	40 899	7.45 m	2 156 m		353 s
2nd-order	30 m	2	72 166	5.61 m	2 142 m	66 m	810 s
2nd-order	20 m	2	161 364	3.75 m	2 109 m	2 m	2 737 s
2nd-order	15 m	2	285 892	2.82 m	2 107 m		6 952 s
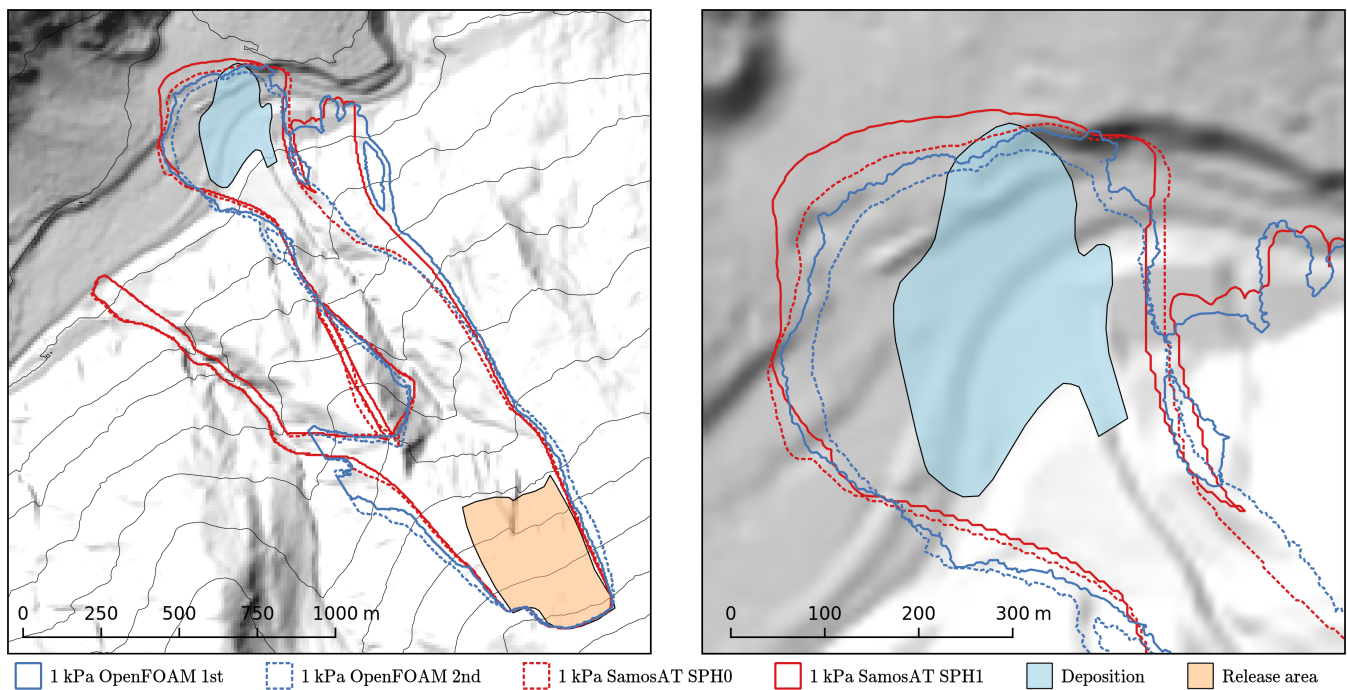
Figure 7. Comparison of OpenFOAM first-order (blue), OpenFOAM second-order (blue, dashed), SamosAT SPH 0 (red, dashed) and SamosAT SPH 1 (red) in terms of 1 kPa-isolines (affected area). OpenFOAM results are based on the mesh with cell size $\Delta = 7.45$ m. The documented release area (orange area) and documented deposition area (blue area) are shown for orientation. The shape and reach of the main avalanche branch are similar in all simulations, secondary branches differ to some extend. Overview (left) and focus on the runout zone (right). Terrain data: AdTLR.

Deposition (i.e. flow thickness field h in the last time-step) of the OpenFOAM solution is shown in Fig. 9 alongside with the documentation.
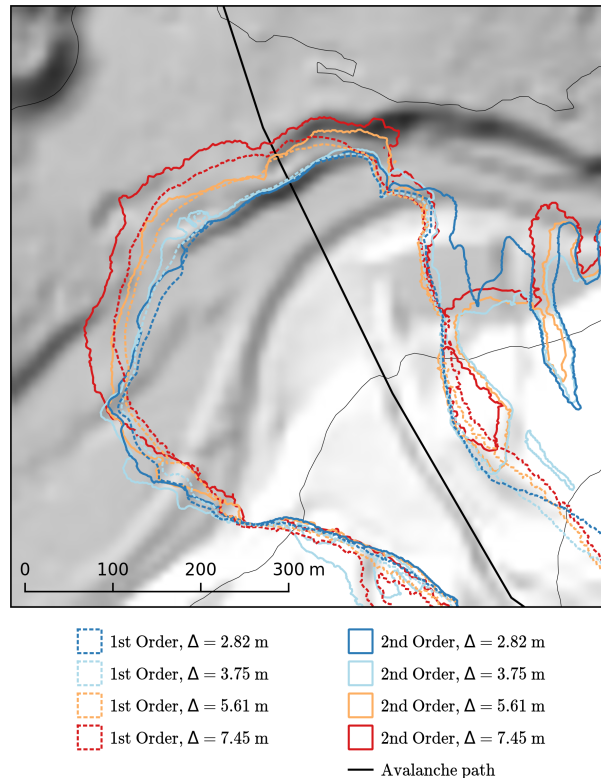
Figure 8. Mesh refinement and convergence study for the OpenFOAM solver. Four mesh sizes and both interpolation schemes, first-order upwind (dashed line) and second-order Gamma (solid line) have been evaluated. The central avalanche path from Fischer et al. (2015) is shown in black. Terrain data: AdTLR.

4   Discussion and conclusion

Results of the new OpenFOAM solver are widely similar to SamosAT. Differences between SamosAT and OpenFOAM are in the range of numerical uncertainty and differences between interpolation methods are of comparable size. This uncertainty has to be expected; in fact, it is well known in the CFD community, that numerical schemes and implementation details influence results dramatically, even for physically simple models (e.g., Sod, 1978). In the case of gravitational mass flows, numerical uncertainty plays a minor role, since underlying models, parameters, terrain and snow cover data are afflicted with substantially higher uncertainty. This is shown by comparison of the documented deposition with the result of an OpenFOAM simulation in Fig. 9. Although parameters have been optimised to the specific event, all simulations differ significantly from documentation. Especially the large bulge on the orographic right side of the deposition area is not matched by any simulation. However, some details, such as the form of the tail and the position where the deposition expands, are accurately simulated by the OpenFOAM solver. Significant differences between simulation and documentation are not limited to the presented case and have been observed before by e.g. Rauter et al. (2016). We deduce that numerical errors are much smaller than the expected model error.
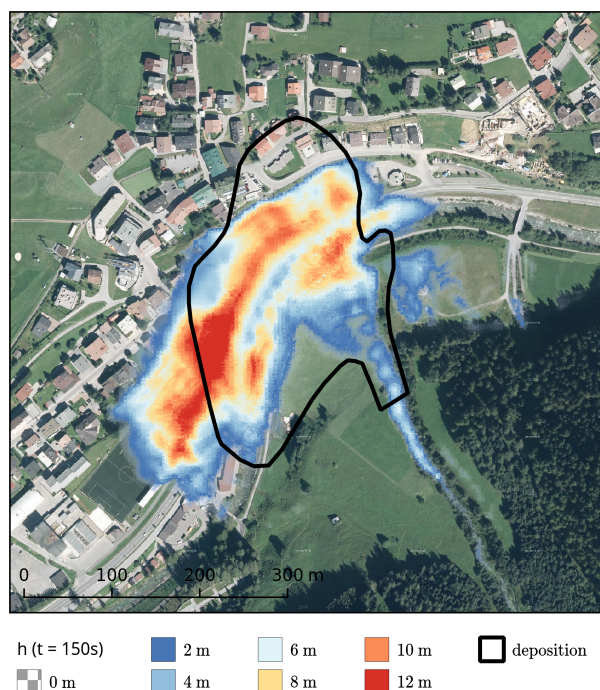
Figure 9. Flow thickness field h at $t = 150\,\mathrm{s}$ of the second-order OpenFOAM simulation ($\Delta = 3.75\,\mathrm{m}$) and the documented deposition area. The flow thickness field in the last time step should roughly replicate the deposition. The bulge on the orographic right side of the deposition area is not matched by any simulation. However, some interesting details, such as the tail of the avalanche are represented well in OpenFOAM simulations. Map data: basemap.at.

Under these circumstances, a quantitative comparison between implementations (as by e.g. Rauter et al., 2016, for basal friction models) is not appropriate.

The refinement study shows that in the presented case, the simulated runout reduces with increasing mesh refinement (Fig. 8). Simulations on fine meshes are stopped by the first embankment, simulations on coarser grids overflow it and reach the next

5   embankment. This is reasonable, considering the higher diffusivity and lower curvature of coarser meshes. However, this trend should not be taken for granted for other cases and a refinement study should always be conducted to get an insight into the numerical uncertainty. Results indicate that a cell size of approximately $3.75\,\mathrm{m}$ is required in OpenFOAM to achieve convergence with respect to practical applications. The numerical uncertainty can not be calculated for the coarsest and finest mesh, since three simulations are required to conduct a Richardson extrapolation. It has to be noted that all simulations are

10   based on the same DEM with a grid size of $10\,\mathrm{m}$. The influence of terrain model quality (see, e.g., Bühler et al., 2011) on simulation results is not investigated.

The execution time of the OpenFOAM solver is acceptable for coarse meshes but increases with the square of the number of cells, because the time-step duration has to be reduced similarly to cell size. The OpenFOAM solver is noticeably slower than SamosAT, especially when considering OpenFOAMs multiprocessing capabilities. For applications where fast execution

is imperative, such as parameter studies, SamosAT may be the appropriate choice. There is potential for future optimisation in OpenFOAM, especially the implicit time integration scheme is expensive and should be replaced with a simpler explicit one. However, the implicit solution strategy, in combination with the regularised friction relation, leads to a satisfying behaviour in the runout zone. In contrast, the simple explicit solution strategy from e.g. SamosAT leads to a continuous creeping of the

5   deposition, meaning that the final flow thickness can not be compared with the deposition, as noticed by Fischer et al. (2015) and Rauter et al. (2016).

Stability of the OpenFOAM solver is strongly influenced by mesh quality. Simulations with polygonal-dominated surface meshes showed an acceptable stability for first- and second-order interpolations. The high influence of the three-dimensional mesh on stability and its computationally expensive creation is the main drawback of the proposed method. This is, however,

10   also a big advantage, allowing simple coupling with three-dimensional ambient flows, as conducted by Sampl and Zwinger (2004).

5   Summary and Outlook

This paper shows the application of a finite area scheme for shallow granular flows (Rauter and Tuković, 2018) to snow avalanches on natural terrain. Specific processes, such as entrainment, have been added to the basic model to replicate the

15   traditional model as implemented in SamosAT (Fischer et al., 2015).

Various simulations with the new OpenFOAM solver have been conducted. Methods and tools to incorporate the OpenFOAM solver in GIS have been presented. These tools allow integration of OpenFOAM in hazard mapping workflows and thus to validate the OpenFOAM solver with a reference implementation, herein SamosAT.

Application of three-dimensional Cartesian coordinates allows simple coupling with GIS applications because no coordinate

20   transformations are required. Unstructured meshes, on the other hand, require re-sampling to structured meshes or data transfer in form of polygons. This incorporates an additional effort compared to simulations on structured meshes, as conducted by e.g. Christen et al. (2010).

The OpenFOAM solver roughly reproduces results of SamosAT. Differences are within the expected numerical uncertainty. A comparison of numerical results to a documented event suggests that model uncertainty is substantially higher than numerical

25   uncertainties.

The major advantage of OpenFOAM is the object-oriented open-source code, which can be easily extended. The flexible code structure allows fast application of new models to real case examples. This especially qualifies the proposed method perfectly for model development and academic purposes. Moreover, the vast majority of source code is shared within the OpenFOAM community, leading to faster development of core features and higher code quality.

30   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
5 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
10 (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.

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.
15 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.

References

Ahrens, J., Geveci, B., Law, C., Hansen, C., and Johnson, C.: ParaView: An End-User Tool for Large Data Visualization, in: The Visualization Handbook, pp. 717–731, Elsevier, 2005.

Ancey, C.: Plasticity and geophysical flows: a review, Journal of Non-Newtonian Fluid Mechanics, 142, 4–35, https://doi.org/10.1016/j.jnnfm.2006.05.005, 2007.

Ayachit, U.: The ParaView Guide: A Parallel Visualization Application, Kitware, Inc., 2015.

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.

Barker, T. and Gray, J.: Partial regularisation of the incompressible $\mu(I)$-rheology for granular flow, Journal of Fluid Mechanics, 828, 5–32, https://doi.org/10.1017/jfm.2017.428, 2017.

Barré de Saint-Venant, A. J. C.: Théorie du mouvement non permanent des eaux, avec application aux crues des rivières et à l'introduction des marées dans leurs lits, Comptes Rendus des séances de l'Académie des Sciences, 73, 237–240, 1871.

Bartelt, P., Buser, O., and Platzer, K.: Fluctuation–dissipation relations for granular snow avalanches, Journal of Glaciology, 52, 631–643, https://doi.org/10.3189/172756506781828476, 2006.

Bartelt, P., Buser, O., Vera Valero, C., and Bühler, Y.: Configurational energy and the formation of mixed flowing/powder snow and ice avalanches, Annals of Glaciology, 57, 179–188, https://doi.org/10.3189/2016AoG71A464, 2015.

Bouchut, F. and Westdickenberg, M.: Gravity driven shallow water models for arbitrary topography, Commun. Math. Sci., 2, 359–389, 2004.

Bühler, Y., Christen, M., Kowalski, J., and Bartelt, P.: Sensitivity of snow avalanche simulations to digital elevation model quality and resolution, Annals of Glaciology, 52, 72–80, https://doi.org/10.3189/172756411797252121, 2011.

Chauchat, J., Cheng, Z., Nagel, T., Bonamy, C., and Hsu, T.-J.: SedFoam-2.0: a 3-D two-phase flow numerical model for sediment transport, Geoscientific Model Development, 10, 4367, https://doi.org/0.5194/gmd-10-4367-2017, 2017.

Cheng, Z., Hsu, T.-J., and Calantoni, J.: SedFoam: A multi-dimensional Eulerian two-phase model for sediment transport and its application to momentary bed failure, Coastal Engineering, 119, 32–50, https://doi.org/10.1016/j.coastaleng.2016.08.007, 2017.

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

Craster, R. V. and Matar, O. K.: Dynamics and stability of thin liquid films, Rev. Mod. Phys., 81, 1131–1198, https://doi.org/10.1103/RevModPhys.81.1131, 2009.

Deckelnick, K., Dziuk, G., and Elliott, C. M.: Computation of geometric partial differential equations and mean curvature flow, Acta numerica, 14, 139–232, https://doi.org/10.1017/S0962492904000224, 2005.

Denlinger, R. P. and Iverson, R. M.: Granular avalanches across irregular three-dimensional terrain: 1. Theory and computation, Journal of Geophysical Research: Earth Surface, 109, F01 014, https://doi.org/10.1029/2003JF000085, 2004.

Dieter-Kissling, K., Marschall, H., and Bothe, D.: Direct Numerical Simulation of droplet formation processes under the influence of soluble surfactant mixtures, Computers & Fluids, 113, 93–105, https://doi.org/10.1016/j.compfluid.2015.01.017, 2015a.

Dieter-Kissling, K., Marschall, H., and Bothe, D.: Numerical method for coupled interfacial surfactant transport on dynamic surface meshes of general topology, Computers & Fluids, 109, 168–184, https://doi.org/10.1016/j.compfluid.2014.12.017, 2015b.

Domnik, B., Pudasaini, S. P., Katzenbach, R., and Miller, S. A.: Coupling of full two-dimensional and depth-averaged models for granular flows, Journal of Non-Newtonian Fluid Mechanics, 201, 56–68, https://doi.org/10.1016/j.jnnfm.2013.07.005, 2013.

Dutykh, D., Acary-Robert, C., and Bresch, D.: Mathematical Modeling of Powder-Snow Avalanche Flows, Studies in Applied Mathematics, 127, 38–66, https://doi.org/10.1111/j.1467-9590.2010.00511.x, 2011.

ESRI: ESRI Shapefile Technical Description, https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf, last checked: 17.12.2017, 1998.

Ferziger, J. H. and Peric, M.: Computational methods for fluid dynamics, Springer, 3 edn., 2002.

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

Fischer, J.-T., Kofler, A., Wolfgang, F., Granig, M., and Kleemayr, K.: Multivariate parameter optimization for computational snow avalanche simulation in 3d terrain, Journal of Glaciology, pp. 875–888, https://doi.org/10.3189/2015JoG14J168, 2015.

Gray, J. M. N. T., Wieland, M., and Hutter, K.: Gravity-driven free surface flow of granular avalanches over complex basal to-
10 pography, Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 455, 1841–1874, https://doi.org/10.1098/rspa.1999.0383, 1999.

Greve, R., Koch, T., and Hutter, K.: Unconfined Flow of Granular Avalanches along a Partly Curved Surface. I. Theory, Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 445, 399–413, https://doi.org/10.1098/rspa.1994.0068, 1994.

15 Grigorian, S. S., Eglit, M. E., and Iakimov, Y. L.: A new formulation and solution of the problem of snow avalanche motion, Snow, Avalanches & Glaciers., 12, 104–113, 1967.

Hanert, E., Le Roux, D. Y., Legat, V., and Deleersnijder, E.: An efficient Eulerian finite element method for the shallow water equations, Ocean Modelling, 10, 115–136, https://doi.org/10.1016/j.ocemod.2004.06.006, 2005.

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

Hormann, K. and Agathos, A.: The point in polygon problem for arbitrary polygons, Computational Geometry, 20, 131–144, https://doi.org/10.1016/S0925-7721(01)00012-8, 2001.

Huang, H., Imran, J., and Pirmez, C.: Numerical model of turbidity currents with a deforming bottom boundary, Journal of Hydraulic Engineering, 131, 283–293, https://doi.org/10.1061/(ASCE)0733-9429(2005)131:4(283), 2005.

25 Hungr, O.: A model for the runout analysis of rapid flow slides, debris flows, and avalanches, Canadian Geotechnical Journal, 32, 610–623, https://doi.org/10.1139/t95-063, 1995.

Issler, D.: Dynamically consistent entrainment laws for depth-averaged avalanche models, Journal of Fluid Mechanics, 759, 701–738, https://doi.org/10.1017/jfm.2014.584, 2014.

Issler, D. and Gauer, P.: Exploring the significance of the fluidized flow regime for avalanche hazard mapping, Annals of Glaciology, 49,
30 193–198, https://doi.org/10.3189/172756408787814997, 2008.

Issler, D., Jenkins, J. T., and McElwaine, J. N.: Comments on avalanche flow models based on the concept of random kinetic energy, Journal of Glaciology, p. 1–17, https://doi.org/10.1017/jog.2017.62, 2017.

Iverson, R. M. and George, D. L.: A depth-averaged debris-flow model that includes the effects of evolving dilatancy. I. Physical basis, Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 470,
35 https://doi.org/10.1098/rspa.2013.0819, 2014.

Iverson, R. M. and George, D. L.: Modelling landslide liquefaction, mobility bifurcation and the dynamics of the 2014 Oso disaster, Géotechnique, 66, 175–187, https://doi.org/10.1680/jgeot.15.LM.004, 2016.

Jasak, H.: Error analysis and estimation for the finite volume method with applications to fluid flows, Ph.D. thesis, Imperial College, University of London, 1996.

Jasak, H., Weller, H. G., and Gosman, A. D.: High resolution NVD differencing scheme for arbitrarily unstructured meshes, Int. J. Numer. Meth. Fluids, 31, 431–449, https://doi.org/10.1002/(SICI)1097-0363(19990930)31:2<431::AID-FLD884>3.0.CO;2-T, 1999.

5 Jóhannesson, T., Gauer, P., Issler, P., and Lied, K.: The design of avalanche protection dams. Recent practical and theoretical developments, No. EUR 23339 in Climate Change and Natural Hazard Research Series 2, 978-92-79-08885-8, 2009.

Juretić, F.: Error analysis in finite volume CFD, Ph.D. thesis, Imperial College London, University of London, 2005.

Juretić, F.: cfMesh User Guide, Creative Fields, Zagreb, 2015.

Kai, C. C., Jacob, G. G., and Mei, T.: Interface between CAD and rapid prototyping systems. Part 1: a study of existing interfaces, The
10 International Journal of Advanced Manufacturing Technology, 13, 566–570, https://doi.org/10.1007/BF01176300, 1997.

Köhler, A., McElwaine, J. N., Sovilla, B., Ash, M., and Brennan, P.: The dynamics of surges in the 3 February 2015 avalanches in Vallée de la Sionne, Journal of Geophysical Research: Earth Surface, 121, 2192–2210, https://doi.org/10.1002/2016JF003887, 2016.

Kowalski, J. and McElwaine, J. N.: Shallow two-component gravity-driven flows with vertical variation, Journal of Fluid Mechanics, 714, 434–462, https://doi.org/10.1017/jfm.2012.489, 2013.

15 Kröner, C.: Experimental and numerical description of rapid granular flows and some baseline constraints for simulating 3-dimensional granular flow dynamics, Ph.D. thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, http://hss.ulb.uni-bonn.de/2014/3694/3694a.pdf, last checked: 17.12.2017, 2013.

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,
20 https://doi.org/10.1029/2002JB002024, 2003.

Marschall, H., Boden, S., Lehrenfeld, C., Falconi D., C. J. Hampel, U., Reusken, A., Wörner, M., and Bothe, D.: Validation of interface capturing and tracking techniques with different surface tension treatments against a Taylor bubble benchmark problem, Computers & Fluids, 102, 336–352, https://doi.org/10.1016/j.compfluid.2014.06.030, 2014.

Medina, V., Hürlimann, M., and Bateman, A.: Application of FLATModel, a 2D finite volume code, to debris flows in the northeastern part
25 of the Iberian Peninsula, Landslides, 5, 127–142, https://doi.org/10.1007/s10346-007-0102-3, 2008.

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

Mergili, M., Fischer, J.-T., Krenn, J., and Pudasaini, S. P.: r.avaflow v1, an advanced open-source computational framework for the propagation and interaction of two-phase mass flows, Geoscientific Model Development, 10, 553–569, https://doi.org/10.5194/gmd-10-553-2017,
30 2017.

Norem, H., Irgens, F., and Schieldrop, B.: A continuum model for calculating snow avalanche velocities, in: Proceedings of the Symposium on Avalanche Formation, Movement and Effects, Davos, Switzerland, pp. 14–19, 1987.

Olshanskii, M. A., Reusken, A., and Grande, J.: A Finite Element Method for Elliptic Equations on Surfaces, SIAM Journal on Numerical Analysis, 47, 3339–3358, https://doi.org/10.1137/080717602, 2009.

35 OpenCFD Ltd.: OpenFOAM - The Open Source CFD Toolbox - User Guide, https://www.openfoam.com/documentation/user-guide/, last checked: 31.01.2018, 2004.

Patra, A. K., Bauer, A., Nichita, C., Pitman, E. B., Sheridan, M., Bursik, M., Rupp, B., Webber, A., Stinton, A., Namikawa, L., et al.: Parallel adaptive numerical simulation of dry avalanches over natural terrain, Journal of Volcanology and Geothermal Research, 139, 1–21, https://doi.org/10.1016/j.jvolgeores.2004.06.014, 2005.

Pesci, C., Dieter-Kissling, K., Marschall, H., and Bothe, D.: Finite volume/finite area interface tracking method for two-phase flows with fluid
5   interfaces influenced by surfactant, in: Computational Methods for Complex Liquid-Fluid Interfaces, pp. 373–409, CRC Press, Taylor & Francis Group, https://doi.org/10.1201/b19337-22, 2015.

Pitman, E. B., Nichita, C. C., Patra, A., Bauer, A., Sheridan, M., and Bursik, M.: Computing granular avalanches and landslides, Physics of Fluids, 15, 3638–3646, https://doi.org/10.1063/1.1614253, 2003.

Pouliquen, O. and Forterre, Y.: Friction law for dense granular flows: application to the motion of a mass down a rough inclined plane,
10   Journal of Fluid Mechanics, 453, 133–151, https://doi.org/10.1017/S0022112001006796, 2002.

Pudasaini, S., Wang, Y., and Hutter, K.: Modelling debris flows down general channels, Natural Hazards and Earth System Science, 5, 799–819, https://doi.org/10.5194/nhess-5-799-2005, 2005a.

Pudasaini, S. P.: A general two-phase debris flow model, Journal of Geophysical Research: Earth Surface, 117, https://doi.org/10.1029/2011JF002186, 2012.

15   Pudasaini, S. P. and Hutter, K.: Avalanche Dynamics: Dynamics of Rapid Flows of Dense Granular Avalanches, Springer, https://doi.org/10.1007/978-3-540-32687-8, 2007.

Pudasaini, S. P., Wang, Y., and Hutter, K.: Rapid motions of free-surface avalanches down curved and twisted channels and their numerical simulation, Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 363, 1551–1571, https://doi.org/10.1098/rsta.2005.1595, 2005b.

20   Rauter, M. and Tuković, Ž.: A finite area scheme for shallow granular flows on three-dimensional surfaces, Computer & Fluids, https://doi.org/10.1016/j.compfluid.2018.02.017, 2018.

Rauter, M., Fischer, J.-T., Fellin, W., and Kofler, A.: Snow avalanche friction relation based on extended kinetic theory, Natural Hazards and Earth System Sciences, 16, 2325–2345, https://doi.org/10.5194/nhess-16-2325-2016, 2016.

Roache, P. J.: Quantification of uncertainty in computational fluid dynamics, Annual Review of Fluid Mechanics, 29, 123–160,
25   https://doi.org/10.1146/annurev.fluid.29.1.123, 1997.

Salm, B., Gubler, H. U., and Burkard, A.: Berechnung von Fliesslawinen: eine Anleitung für Praktiker mit Beispielen, WSL Institut für Schnee-und Lawinenforschung SLF, Davos, 1990.

Sampl, P. and Granig, M.: Avalanche Simulation with Samos-AT, in: Proceedings of the International Snow Science Workshop, September 27-October 2, 2009, pp. 519–523, 2009.

30   Sampl, P. and Zwinger, T.: Avalanche simulation with SAMOS, Annals of glaciology, 38, 393–398, https://doi.org/10.3189/172756404781814780, 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.

Savage, S. B. and Hutter, K.: The dynamics of avalanches of granular materials from initiation to runout. Part I: Analysis, Acta Mechanica,
35   86, 201–223, https://doi.org/10.1007/BF01175958, 1991.

Schöberl, J.: NETGEN An advancing front 2D/3D-mesh generator based on abstract rules, Computing and visualization in science, 1, 41–52, https://doi.org/10.1007/s007910050004, 1997.

Sod, G. A.: A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, Journal of Computational Physics, 27, 1–31, https://doi.org/10.1016/0021-9991(78)90023-2, 1978.

Tuković, Ž. and Jasak, H.: A moving mesh finite volume interface tracking method for surface tension dominated interfacial fluid flow, Computers & fluids, 55, 70–84, https://doi.org/doi:10.1016/j.compfluid.2011.11.003, 2012.

5   Vera Valero, C., Jones, K. W., Bühler, Y., and Bartelt, P.: Release temperature, snow-cover entrainment and the thermal flow regime of snow avalanches, Journal of Glaciology, 61, 173–184, https://doi.org/10.3189/2015JoG14J117, 2015.

Voellmy, A.: Über die Zerstörungskraft von Lawinen., Schweizerische Bauzeitung, 73, 1955.

von Boetticher, A., Turowski, J. M., McArdell, B. W., Rickenmann, D., and Kirchner, J. W.: DebrisInterMixing-2.3: a finite volume solver for three-dimensional debris-flow simulations with two calibration parameters – Part 1: Model description, Geoscientific Model Development,
10   9, 2909–2923, https://doi.org/10.5194/gmd-9-2909-2016, 2016.

von Boetticher, A., Turowski, J. M., McArdell, B. W., Rickenmann, D., Hürlimann, M., Scheidl, C., and Kirchner, J. W.: DebrisInterMixing-2.3: a finite volume solver for three-dimensional debris-flow simulations with two calibration parameters – Part 2: Model validation with experiments, Geoscientific Model Development, 10, 3963–3978, https://doi.org/10.5194/gmd-10-3963-2017, 2017.

Vreugdenhil, C. B.: Numerical methods for shallow-water flow, Kluwer Academic Publishers, 1994.

15   Wang, Y., Hutter, K., and Pudasaini, S. P.: The Savage-Hutter theory: A system of partial differential equations for avalanche flows of snow, debris, and mud, ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik, 84, 507–527, https://doi.org/10.1002/zamm.200310123, 2004.

Weller, H. G., Tabor, G., Jasak, H., and Fureby, C.: A tensorial approach to computational continuum mechanics using object-oriented techniques, Computers in physics, 12, 620–631, https://doi.org/10.1063/1.168744, 1998.

20   Zhai, Q., Zhang, R., and Wang, X.: A hybridized weak Galerkin finite element scheme for the Stokes equations, Science China Mathematics, 58, 2455–2472, https://doi.org/10.1007/s11425-015-5030-4, 2015.