To:	Norges Vassdrags- og Energidirektorat (NVE)
Attn.:	Odd Are Jensen
Copy to:	KGi, HHH
Date:	2025-02-20
Revision no./Rev.date:	0/
Document no.:	20230100-06-TN
Project:	AARN – Applied Avalanche Research in Norway
Project manager:	Kjersti Gleditsch Gisnås
Prepared by:	Dieter Issler
Reviewed by:	Callum Tregaskis

Basic Equations and Numerical Methods in MoT-Voellmy

Abstract

This note on the quasi-3D numerical avalanche run-out model MoT-Voellmy presents and explains the geometric setting, the basic depth-averaged conservation equations for mass and momentum and the options for including additional effects like centrifugal forces, entrainment and the braking effect of forest. These options can be turned on or off in the run configuration file, a simple text file whose structure is briefly explained. MoT-Voellmy uses the minimal version of the Method of Transport for solving the system of partial differential equations as efficiently as possible, neglecting the gravity waves due to hydrostatic pressure gradients. It thereby becomes similar to the Decoupled Hydrological Discretization scheme used by Cea and Bladé. The consequences of this simplification will be explored further in a separate publication.

Contents

1 Introduction **2**
2 The geometric setting **3**
3 Depth-averaged conservation equations **7**
4 Modeling options **12**
5 The numerical scheme: Method of Transport **14**
6 Concluding remarks **17**
Acknowledgements **18**
Bibliography **18**

Review and reference page **22**

1 Introduction

MoT-Voellmy is a program, developed and used at NGI since late 2011, for rapidly simulating the flow of dense avalanches in complex terrain. It implements the simple Voellmy friction law (Voellmy, 1955) in a quasi-3D (depth-averaged) setting and offers enhancements like entrainment and two-way interaction between forest stands and the avalanche flow.

The predecessor of MoT-Voellmy is the C++ code DIMAS, developed by Maja Adamska-Szatko for her PhD thesis (Adamska-Szatko, 2012) during a stay at NGI in the fall of 2011. Her task was to create a Cellular Automaton (CA) code for simulating snow avalanches with graphical animation as the computation progresses. A minimal variant of the Method of Transport for solving (hyperbolic) conservation laws in multiple dimensions (Fey & Jeltsch, 1993; Fey & Morel, 1995; Fey, 1998a,b; Noelle, 2000) with explicit time discretization satisfies all criteria for a CA except the discreteness of the node states and was therefore adopted. In contrast to other attempts of simulating gravity mass flows with CA, DIMAS conserves mass and linear momentum. For simplicity, the well-known Voellmy friction law was chosen, despite its known shortcomings. While DIMAS worked quite satisfactorily, simulations were very time-consuming and impractical due to the continuous generation of graphics output.

For this reason, the author implemented the same modeling concept but without graphics from scratch in a simple ISO-C code, which ran two to three orders of magnitude faster than DIMAS. Over the years, many improvements and enhancements have been made to MoT-Voellmy: The leading effect of terrain curvature is accounted for, hydrostatic pressure gradients are discretized in a more suitable way, entrainment models have been added, and two-way interaction with forest (the trees braking the avalanche and the avalanche breaking the trees) can be switched on or off. Thanks to its speed, MoT-Voellmy enabled the recently developed tool NAKSIN for automated generation of avalanche hazard indication maps (Issler *et al.*, 2023) to simulate approximately 1.3 million snow avalanche paths in Norway both with and without forest interaction.

Recently, Callum Tregaskis implemented a depth-averaged version of the $\mu(I)$ rheology (Jop *et al.*, 2006; Gray & Edwards, 2014; Barker *et al.*, 2017) by modifying the momentum source terms of MoT-Voellmy. The code MoT-muI (Tregaskis *et al.*, 2023) has very promising properties and will also be made publicly available after additional tests. Similarly, Hervé Vicari added a second set of balance equations for volume, mass and momentum to the MoT-Voellmy code to simulate mixed avalanches with a substantial powder-snow cloud as a two-layer flow (Vicari & Issler, 2025). Properly testing and calibrating this model is more demanding because of the larger number of parameters from multiple constitutive equations and exchange terms for mass or momentum between the snow cover, the two flow layers and the ambient air. This code, termed MoT-PSA, will also be released as open-source software once more experience from practical applications has been gained and recommendations can be given for choosing the model parameters.

The purpose of this Technical Note is to document the physical background of MoT-Voellmy, the set of equations defining the model, the method for solving them numerically, and the extra features that have been added to the code over the years. The validation of the code and suggestions for the choice of simulation settings are reported in another Technical Note. In addition, a scientific paper emphasizing the Method of Transport will be submitted to a peer-reviewed journal in due time.

2 The geometric setting

Coordinate system User input of geographical data (DTM, release area, forest stand density times average tree diameter, depth and shear strength of erodible snow) to MoT-Voellmy must always be in a regular raster (square grid cells) referring to a projected Cartesian coordinate system (X, Y, Z). Program output (flow depth, velocity, impact pressure, remaining erodible snow) refers to the same coordinate system and raster. In most cases, users will apply a suitable UTM (Universal Transverse Mercator) system, with the X and Y-coordinates approximately aligned with the East and North direction, respectively. It is, however, possible to use a rotated coordinate system – MoT-Voellmy is agnostic of such transformations.

Internally, the governing equations of MoT-Voellmy are formulated in a terrain-following curvilinear coordinate system (x, y, z). The surface $z = 0$ coincides with the terrain surface Σ, and a point $\mathrm{S} = (X, Y, Z(X, Y)) \in \Sigma$ in the Euclidean system has coordinates $(x = X, y = Y, z = 0)$ in the curvilinear system.

In the local coordinate system, the z-direction is chosen orthogonal to the local tangent plane spanned by the x and y-directions. The points $(x, y, z = \text{cst.})$ on the surfaces Σ_z have global coordinates $(X, Y, Z(X, Y)) + z\,\hat{\boldsymbol{n}}(X, Y, Z(X, Y))$, where $\hat{\boldsymbol{n}}$ is the unit normal vector on Σ_0. Thus, the points $(x, y, 0)$ and $(x, y, z_1 > 0)$ do not vertically project onto the same point on the plane $(X, Y, Z = 0)$ in general. The model is limited to flow depths well below R_{min}, the minimum radius of concave curvature on Σ_0, because z-coordinate lines will intersect and the coordinate system becomes singular. Another important feature is that the y-direction is not in general orthogonal to the x-direction, a property that must be accounted for when calculating the speed from the components of the velocity vector.

The vertical projection of the computational grid of MoT-Voellmy onto the horizontal plane consists of a raster of square cells with constant side length $\Delta X = \Delta Y$. MoT-Voellmy requires a rectangular raster.

This coordinate system is also used in the commercial code RAMMS::AVALANCHE (Christen *et al.*, 2010; Bartelt *et al.*, 2017). Geographical input data for RAMMS and some auxiliary files created by that program (notably raster files of friction parameters μ and ξ, but not the vector files for the release areas or forest) can be used in MoT-Voellmy. Both programs use the simple ESRI ASCII Grid (AAIGrid) format for input of raster data

(ESRI, 2020), which can be read and written by all popular geographical information systems (GIS). Users of MoT-Voellmy can choose between the output formats AAIGrid and BinaryTerrain 1.3 (Discoe, 2007).

Metric Tensor and Vectors The curvilinearity and non-orthogonality of the coordinate system (x, y, z) has two important and connected consequences: First, the *physical* side lengths Δx and Δy of computational cells differ from ΔX if the local slope angle is not 0, and in general they are not constant and $\Delta x \neq \Delta y$. Second, even on an inclined plane that is locally parallel to Σ, the length of a vector cannot be evaluated with the Euclidean formula $\|\boldsymbol{v}\| = (v_x^2 + v_y^2)^{1/2}$ but depends on the angle between the x and y-directions.

The correct formulas are obtained by using standard methods of differential geometry. The starting point is to consider the tangent vectors to Σ_0 along the coordinate lines $y = \text{cst.}$, $x = \text{cst.}$ and the unit normal vector, given by

$$\boldsymbol{\epsilon}_x = (1, 0, \partial_x Z)^{\mathrm{T}} \quad \text{and} \quad \boldsymbol{\epsilon}_y = (0, 1, \partial_y Z)^{\mathrm{T}}, \tag{1}$$

$$\hat{\boldsymbol{n}} = \boldsymbol{\epsilon}_z = \frac{\boldsymbol{\epsilon}_x \wedge \boldsymbol{\epsilon}_y}{\|\boldsymbol{\epsilon}_x \wedge \boldsymbol{\epsilon}_y\|} = \frac{(-\partial_x Z, -\partial_y Z, 1)^{\mathrm{T}}}{\sqrt{1 + (\partial_x Z)^2 + (\partial_y Z)^2}}. \tag{2}$$

Points above Σ_0 are parameterized by x, y and the *normal* distance z from Σ_0, thus their global coordinates are determined by $(x + z \cdot \hat{n}_x, y + z \cdot \hat{n}_y, Z(x, y) + z \cdot \hat{\boldsymbol{n}}_z)^{\mathrm{T}}$. A vector $\boldsymbol{v}$ in a tangent plane to Σ_0 can be expressed as

$$\boldsymbol{v} = v^x \boldsymbol{\epsilon}_x + v^y \boldsymbol{\epsilon}_y, \tag{3}$$

where (v_x, v^y) are the *contravariant* components of $\boldsymbol{v}$ because they behave oppositely to the basis $(\boldsymbol{\epsilon}_x, \boldsymbol{\epsilon}_y)$ under changes of the coordinate system so that $\boldsymbol{v}$ remains invariant.

Because of the curvature, coordinate increments Δx, Δy are not constant in physical length units. The corresponding information is contained in the metric tensor, which has the (covariant) form

$$\mathcal{G}(x, y, z) = \begin{pmatrix} G_{xx}(x, y, z) & G_{xy}(x, y, z) & 0 \\ G_{yx}(x, y, z) & G_{yy}(x, y, z) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{4}$$

The components of $\mathcal{G}$ on Σ_0 can readily be calculated from the formula $G_{ij} = \boldsymbol{\epsilon}_i \cdot \boldsymbol{\epsilon}_j$:

$$\begin{aligned} G_{xx}(x, y, 0) &= 1 + (\partial_x Z)^2, \\ G_{yy}(x, y, 0) &= 1 + (\partial_x Z)^2, \\ G_{xy}(x, y, 0) &= (\partial_x Z)(\partial_x Z) = G_{yx}(x, y, 0). \end{aligned} \tag{5}$$

Contravariant components u^i of a vector $\boldsymbol{u}$, say the flow velocity, can be converted to physical units (U_x, U_y, U_z) as

$$U_x = u^x \sqrt{G_{xx}}, \quad U_y = u^y \sqrt{G_{yy}}, \quad U_z = u^z. \tag{6}$$

We write (U_x, U_y, U_z) here to emphasize that these quantities are no longer contravariant because of this scaling to constant physical length units. (They are not covariant either.) The non-orthogonality of the coordinate system implies that the speed, $\|\boldsymbol{u}\|$, is not given by $\sqrt{(U_x)^2 + (U_y)^2}$ but by

$$\begin{aligned}\|\boldsymbol{u}\| &= \sqrt{G_{xx}(u^x)^2 + G_{yy}(u^y)^2 + 2G_{xy}u^x u^y} \\ &= \sqrt{U_x^2 + U_y^2 + 2U_x U_y \cos\alpha},\end{aligned} \tag{7}$$

where α, the angle between the tangent vectors $\boldsymbol{\epsilon}_x$ and $\boldsymbol{\epsilon}_y$, is obtained from

$$\cos\alpha = \frac{G_{xy}}{\sqrt{G_{xx}G_{yy}}}. \tag{8}$$

The correction (7) can be significant in steep terrain if the direction of steepest descent is close to the diagonal between coordinate lines. Assume the directional slope angles to be 45°, with the tangent vectors given by $\epsilon_x^{\mathrm{T}} = (1, 0, -1)$ and $\epsilon_y^{\mathrm{T}} = (0, 1, -1)$. If $\boldsymbol{u}$ is in the direction of steepest descent, $U^x = U^y$. Then $G_{xx} = G_{yy} = 2$, $G_{xy} = 1$, $\cos\alpha = 1/2$ and $\|\boldsymbol{U}\| = \sqrt{3/2}\sqrt{(U^x)^2 + (U^y)^2}$. Without this correction, the drag term in the Voellmy friction law would be underestimated by one third.

The gravitational acceleration, $\boldsymbol{g} = (0, 0, -g)^{\mathrm{T}}$ in the Cartesian system, can be decomposed into bed-normal and bed-parallel parts as $\boldsymbol{g} = g_z\boldsymbol{\epsilon}_z + \boldsymbol{g}_\parallel$. In the local coordinate system and using physical units, one finds

$$g_z = -\boldsymbol{g}\cdot\boldsymbol{\epsilon}_z = \frac{g}{\sqrt{1 + (\partial_X z)^2 + (\partial_Y Z)^2}}, \tag{9}$$

$$\boldsymbol{g}_\parallel = -\frac{g}{1 + (\partial_X Z)^2 + (\partial_Y Z)^2}\begin{pmatrix} \sqrt{1 + (\partial_X Z)^2}\,\partial_X Z \\ \sqrt{1 + (\partial_Y Z)^2}\,\partial_Y Z \\ 0 \end{pmatrix}. \tag{10}$$

Curvature The spatial variability of the metric tensor requires, in principle, replacing the derivative of a vector in a Cartesian system, e.g. $\partial_x v^y$, by the covariant derivative $D_x v^y = \partial_x v^y + \Gamma^y_{xx}v^x + \Gamma^y_{xy}v^y + \Gamma^y_{xz}v^z$, the connection coefficients Γ^i_{jk} being composed of derivatives of the metric tensor. A full implementation of this is complicated and computationally expensive (Bouchut & Westdickenberg, 2004; Issler, 2006; Peruzzetto *et al.*, 2021). MoT-Voellmy retains only the leading term, namely the centrifugal force due to the normal curvature in the flow direction, $\kappa_{\boldsymbol{U}}$. Textbooks or websites (Wikipedia authors, 2002–2024) on differential geometry explain how to obtain the curvature of a curve in Σ_0, $s \mapsto \gamma(s) = (x(s), y(s))^{\mathrm{T}}$, where the parameter s is the arc length along the curve. The needed quantities are the tangent vector $(\mathrm{d}x/\mathrm{d}s,\ \mathrm{d}y/\mathrm{d}s)^{\mathrm{T}}$, the metric tensor $\mathcal{G}$ and the coefficients L, M, N of the second fundamental form of Σ_0, which describes how strongly Σ_0 curves away from the local tangent plane:

$$\kappa_{\boldsymbol{\gamma}}(s) = L\left(\frac{\mathrm{d}x}{\mathrm{d}s}\right)^2 + 2M\frac{\mathrm{d}x}{\mathrm{d}s}\frac{\mathrm{d}y}{\mathrm{d}s} + N\left(\frac{\mathrm{d}y}{\mathrm{d}s}\right)^2. \tag{11}$$

L, M and N are given by the following scalar products:

$$\begin{aligned}(L, M, N) &= \frac{(-\partial_x Z,\, -\partial_y Z,\, 1)}{\sqrt{1 + (\partial_x Z)^2 + (\partial_y Z)^2}} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \partial_x^2 Z & \partial_x\partial_y Z & \partial_y^2 Z \end{pmatrix} \\ &= \left[1 + (\partial_x Z)^2 + (\partial_y Z)^2\right]^{-1/2} \left(\partial_x^2 Z,\, \partial_x\partial_y Z,\, \partial_y^2 Z\right). \end{aligned} \tag{12}$$

In the present case, $(\mathrm{d}x/\mathrm{d}s, \mathrm{d}y/\mathrm{d}s)^{\mathrm{T}}$ is to be replaced by the direction of the velocity vector, $\boldsymbol{U}/\|\boldsymbol{U}\|$, making use of Eqs. (7) and (8). Eq. (11) then evaluates to

$$\kappa_{\boldsymbol{U}} = \frac{LU_x^2 + 2MU_xU_y + NU_y^2}{U_x^2 + 2U_xU_y\cos\alpha + U_y^2}. \tag{13}$$

Numerical Approximation of Geometrical Quantities The Z-coordinate values given by the digital terrain model (DTM) represent the altitude averaged over the respective cell and are associated with the cell center. First, approximations to the tangent vectors to the coordinate lines are computed by central differences in the interior of the computational domain as

$$\boldsymbol{\epsilon}_{x,ij} = \begin{pmatrix} 1 \\ 0 \\ (\partial_X Z)_{ij} \end{pmatrix} \approx \begin{pmatrix} 1 \\ 0 \\ \frac{Z_{i+1,j} - Z_{i-1,j}}{2\Delta X} \end{pmatrix}, \qquad \boldsymbol{\epsilon}_{y,ij} = \begin{pmatrix} 0 \\ 1 \\ (\partial_Y Z)_{ij} \end{pmatrix} \approx \begin{pmatrix} 0 \\ 1 \\ \frac{Z_{i,j+1} - Z_{i,j-1}}{2\Delta Y} \end{pmatrix}. \tag{14}$$

where $\Delta X = \Delta Y$ is the uniform cell size in the horizontal projection; at the lower (upper) boundaries, right-sided (left-sided) differences are used instead. The oblique cell sizes in the terrain are computed as

$$\begin{pmatrix} \Delta x_{ij} \\ \Delta y_{ij} \end{pmatrix} = \Delta X \begin{pmatrix} \sqrt{1 + (\partial_X Z)_{ij}^2} \\ \sqrt{1 + (\partial_Y Z)_{ij}^2} \end{pmatrix}. \tag{15}$$

The cell area is approximated as the modulus of the vector product $\boldsymbol{\epsilon}_{x,ij} \wedge \boldsymbol{\epsilon}_{y,ij}$:

$$(\Delta A)_{ij} = (\Delta X)^2 \sqrt{1 + (\partial_X Z)_{ij}^2 + (\partial_Y Z)_{ij}^2}. \tag{16}$$

The approximations to the metric tensor and the coefficients of the Second Fundamental Form of Σ_0 are easily obtained from Eqs. (5), (8), and (12)–(16).

3 Depth-averaged conservation equations

Formulation of the balance equations Like most dynamical models for gravity mass flows in practical use today, MoT-Voellmy implements the fundamental balance laws for mass and momentum. The mass of Earth is considered infinite; this, together with the presence of gravity and friction forces between the ground and the avalanche, entails that the momentum of the avalanche is not constant but changes in a well-defined way. Moreover, flowing avalanches can entrain parts of the snow cover along their path so that their mass is not constant either.

No separate balance equation is imposed for the total energy; this implies that the kinetic energy of the avalanche increases according to the work gravity does on the avalanche, and decreases at the rate friction dissipates it. The tacit assumption is that the dissipated energy does not change the other properties of the avalanche. In reality, sliding friction generates heat, which may change the properties of the moving snow considerably through granulation (Steinkogler *et al.*, 2014, 2015), lubrication or cohesion (Vera Valero *et al.*, 2015; Bartelt *et al.*, 2015). Particle collisions convert kinetic energy (evaluated at the macro-scale described by the model) partly into granular temperature, i.e., fluctuating kinetic energy of the snow particles (meso-scale), and partly into heat (micro-scale). The granular temperature, in turn, generates a dispersive pressure in the avalanche body that tends to dilate the flow, which may reduce the effective friction by a large amount (Issler & Gauer, 2008, and references therein). These effects are not captured in MoT-Voellmy.

To reduce the computational load of avalanche simulations to a practical level, the vast majority of dynamical models—including MoT-Voellmy—use the shallow-water approximation, neglecting the effects of bed-normal motion and the non-uniformity of the velocity profile, i.e., using the depth-averaged value instead. This reduces the problem from three spatial dimensions to two and the number of partial differential equations from four to three. Many papers and textbooks, e.g., (Parker *et al.*, 1986) and (Pudasaini & Hutter, 2007), describe the procedure in detail. With the stated approximations concerning curvature effects, the set of equations can be written as

$$\rho_b \partial_t b = -q_e, \tag{17}$$

$$\rho_f \, \partial_t h + \rho_f \, \boldsymbol{\nabla} \cdot (h\bar{\boldsymbol{u}}) = q_e, \tag{18}$$

$$\rho_f \, \partial_t (h\bar{\boldsymbol{u}}) + \boldsymbol{\nabla} \cdot (h\overline{\boldsymbol{u}\boldsymbol{u}} + h\bar{p}) = \rho_f \, h\boldsymbol{g}_{\parallel} - \boldsymbol{\tau}_b. \tag{19}$$

b and h are the depths of the erodible snowcover and of the avalanche, measured normal to the local terrain, ρ_b and ρ_f are the respective densities. The vector $\boldsymbol{u} = (u, v)^{\mathrm{T}}$ is the avalanche velocity (projected on the local tangent plane to the terrain); it is measured in *physical* units, not in covariant ones. q_e designates the rate of mass transfer from the snow cover to the avalanche, $\boldsymbol{g}_{\parallel}$ is the projection of the gravitational acceleration to the local tangent plane, see Eqs. (9) and (10). $\bar{p}$ is the depth-averaged hydrostatic pressure (the isotropic part of the total stress tensor), while $\boldsymbol{\tau}_b$ is the bed shear stress and is also to be measured in physical units like $\boldsymbol{g}$ and $\boldsymbol{u}$.

Equation (17)—often called the Exner equation—keeps track of the remaining depth of

the snow cover and becomes 0 when all erodible snow has been entrained by the avalanche. The erosion rate appears as a source term with opposite signs in the Exner equation and in the mass balance equation (18) of the avalanche. Its functional dependence on the flow variables h and $\bar{\boldsymbol{u}}$ must be modeled (see Sec. 4). Equation (17) is, essentially, a set of trivial ordinary differential equations because only time derivatives appear as w_e is a function of h and $\bar{\boldsymbol{u}}$ without derivatives, as will be seen in Sec. 4. We also anticipate that the constitutive relations for $\bar{p}$ and $\boldsymbol{\tau}_b$ do not contain derivatives of the fields h, $\bar{\boldsymbol{u}}$. Then Eqs. (18) and (19) constitute a system of three first-order partial differential equations. Moreover, this system is hyperbolic, with the physical wave speeds

$$\boldsymbol{c} = \boldsymbol{u} + \sqrt{g_z h}\,(\cos\phi, \sin\phi, 0)^{\mathrm{T}}, \quad 0 \le \phi < 2\pi, \tag{20}$$

forming a circle of radius $\sqrt{g_z h}$ about the advection velocity $\boldsymbol{u}$.

MoT-Voellmy imposes von Neumann boundary conditions for h and $\bar{\boldsymbol{u}}$ along the perimeter of the computational domain so that the avalanche can flow out of the domain unhindered. As initial conditions, the user must provide the fields $b_0(\boldsymbol{x}) = b(\boldsymbol{x}, 0)$, $h_0(\boldsymbol{x}) = h(\boldsymbol{x}, 0)$, and $\bar{\boldsymbol{u}}_0(\boldsymbol{x}) = \bar{\boldsymbol{u}}(\boldsymbol{x}, 0)$. b_0 is only needed if entrainment is to be simulated, and $\bar{\boldsymbol{u}}_0$ can be left out if the simulation is to start from rest.

Closure relations The pressure p is assumed to be equal to the hydrostatic (or lithostatic) pressure component in the slope-nornmal direction, i.e.,

$$p = \rho_f\, g_z h = \rho_f\, g h \cos\theta. \tag{21}$$

For the bed shear stress $\boldsymbol{\tau}_b$, the Voellmy friction law (Voellmy, 1955) is assumed:

$$\boldsymbol{\tau}_b = \frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}\rho_f\,\left(\mu g_z h + k\boldsymbol{u}^2\right). \tag{22}$$

The first term represents dry, velocity-independent Coulomb friction while the second term can be considered as drag of avalanche snow on the sliding surface. Using this form of the drag allows to set the drag term exactly to 0, in contrast to the traditional form $(g/\xi)\boldsymbol{u}^2$ with $\xi \equiv g/k$, which would require $\xi \to \infty$. For an option to modify the drag term, see Sec. 4.

For the entrainment rate q_e, four different formulas termed ‘RAMMS’, ‘TJEM’, ‘AvaFrame’ and ‘GOEM’ have been implemented, but only two of them have been tested to some degree; it is not recommended to use the options ‘AvaFrame’ or ‘GOEM’ except for development and testing. ‘RAMMS’ refers to the entrainment formula available in early versions of RAMMS::AVALANCHE; it has since been removed from the code, but a more sophisticated variant is found in RAMMS::EXTENDED (Bartelt *et al.*, 2018). ‘TJEM’ stands for ‘Tangential-jump entrainment model’, which is briefly described in (Issler, 2014) and was proposed earlier by Fraccarollo & Capart (2002). In the programming and simulation evironment AvaFrame (Oesterle *et al.*, 2021), the computational module com1DFA offers an optional erosion module, which has been reproduced

in MoT-Voellmy. The option 'GOEM' (Grigorian–Ostroumov erosion model) is an extension of the entrainment formula by Grigorian & Ostroumov (2020) from the mid 1970s to quasi-3D models. They are briefly summarized here:

RAMMS The entrainment rate is assumed to depend solely on the avalanche velocity and a user-selectable proportionality coefficient,

$$q_e^{\text{RAMMS}} = \kappa \rho_b \|\boldsymbol{u}\|. \tag{23}$$

The RAMMS manual suggested $\kappa = 0.1 \ldots 0.5$, which amounts to full-depth entrainment at the avalanche front over a flow distance of typically 10 m or less. The shear strength of the erodible snow does not enter the formula. The only necessary extra input is the initial distribution of erodible snow, $b_0(\boldsymbol{x})$.

TJEM The key assumptions of this model are that (i) the bed shear stress τ_b of the avalanche must exceed the shear strength τ_c of the snow cover for entrainment to occur, (ii) the inclination of the snow–avalanche interface can be neglected, (iii) the shear stress at the interface is locked at $\tau_b = \tau_c$ and the difference $\tau_b - \tau_c$ accelerates the eroded material to the mean velocity of the avalanche, which leads to the entrainment rate

$$q_e^{\text{TJEM}} = \frac{R(\|\boldsymbol{\tau}\| - \tau_c)}{\|\boldsymbol{u}\|} \tag{24}$$

without any empirical parameters. $R(x)$ is the ramp function, which takes the value 0 for $x < 0$ and x for $x \geq 0$. The user must specify $b_0(\boldsymbol{x})$ and $\tau_c^{(s)}(\boldsymbol{x})$. The latter can either be constant throughout the depth of the snow cover, or it can increase linearly with depth with a space-independent gradient μ_b to take into account consolidation of the snow under its own weight. The shear strength is then derived from the instantaneous value of $b(\boldsymbol{x}, t)$ as

$$\tau_c(\boldsymbol{x}, t) = \tau_c^{(s)}(\boldsymbol{x}) + \mu_b \rho_f \, g_z [b_0(\boldsymbol{x}) - b(\boldsymbol{x}, t)]. \tag{25}$$

AvaFrame The AvaFrame computational module com1DFA offers two alternative entrainment modes, namely frontal entrainment by plowing and basal entrainment by scour. MoT-Voellmy implements only the latter because avalanche front tracking is not so easily done in an Eulerian model, where the computational cells do not move with the mass but stay fixed. The entrainment rate in the scouring mode is given by

$$q_e^{\text{AvaFrame}} = \frac{\|\boldsymbol{\tau}_b(h, \boldsymbol{u})\|}{e_b} \|\boldsymbol{u}\|, \tag{26}$$

where $\boldsymbol{tau}_b$ is the bed shear stress computed from the friction law and $e_b > 0$ is considered to be the specific deformation energy (work per unit mass) required for detaching and breaking up the eroded snow. e_b depends, in principle, on the snow properties. MoT-Voellmy expects the (local) values of e_b (in $\text{m}^2\,\text{s}^{-2}$) in the file listed in the run set-up file under the entry 'Bed shear strength filename'.

GOEM Grigorian and Ostroumov describe the interface between snow cover and avalanche as an inclined shock front and apply the jump conditions for mass and momentum across it, solving for the interface speed to obtain the entrainment rate. In doing so, they account for the (compressive) strength of the snow cover and also add a dynamic pressure $c\rho_f \boldsymbol{u}^2 \cos^2\alpha$ exerted by the avalanche with an empirical coefficient c. The latter is specified in the run set-up file instead of the entrainment coefficient of the option 'RAMMS'. The addition of the dynamic pressure has been criticized by Eglit & Demidov (2005) as a physically incorrect ad-hoc addition. Issler (2020) suggests that Grigorian and Ostroumov may have added this term in an attempt to compensate for the neglected jump condition for the interface-parallel component of momentum. The formula implemented in MoT-Voellmy is

$$q_e^{\text{GOEM}} = \frac{R(\sigma_\perp - \sigma_c)}{\|\boldsymbol{u}\| \cos\alpha}. \tag{27}$$

σ_c is the compressive strength of the snow cover, $\sigma_\perp$ the normal stress exerted by the avalanche on the interface, which is inclined at a variable angle $\alpha(\boldsymbol{x}, t)$ to the terrain; α is obtained from the gradient of the erodible snow depth, $b(\boldsymbol{x}, t)$. The formula for the avalanche pressure onto the erosion front is

$$\sigma_\perp = \rho_f \left(g_z h^2 + c\boldsymbol{u}^2\right) \cos^2\alpha. \tag{28}$$

There is a condition hidden in this way of writing the formula, namely $\rho_b < \rho_f$. This makes the GOEM not applicable to powder-snow avalanches and highly fluidized flows.

Discussion of approximations The modeling approach characterized by Eqs. (17)–(19) implies several simplifications that are easily overlooked or forgotten. We will briefly list and discuss them below; further important assumptions will be made when specifying the source terms q_e and $\boldsymbol{\tau}_b$ in Sec. 4.

- *Form factor:* In Eq. (19), $h\overline{\boldsymbol{u}\boldsymbol{u}}$ means that the quantity $\boldsymbol{u}\boldsymbol{u}$ should be averaged over the flow depth. Most models, including MoT-Voellmy, approximate it by $\bar{\boldsymbol{u}}\bar{\boldsymbol{u}}$. Depending on the shape of the velocity profile, these two expressions may differ significantly—by a factor 6/5 for laminar flow of a Newtonian fluid and by a factor 5/3 in the case of steady down-slope flow of a granular fluid.

- *Neglected contributions to divergence of stress tensor:* When depth-integrating the three-dimensional generalized Navier–Stokes equation to obtain the momentum balance (19), the divergence of the stress tensor gives rise to the fol-

lowing terms:

$$\int_0^h \partial_z(\tau_{xz}, \tau_{yz}, \tau_{zz})\mathrm{d}z = \boldsymbol{\tau}_s - \boldsymbol{\tau}_b \approx -\boldsymbol{\tau}_b, \tag{29}$$

$$\int_0^h \partial_x(\tau_{xx}, \tau_{xz})\mathrm{d}z = h\partial_x(\bar{\tau}_{xx}, \bar{\tau}_{xy}), \tag{30}$$

$$\int_0^h \partial_y(\tau_{xy}, \tau_{xz})\mathrm{d}z = h\partial_x(\bar{\tau}_{xx}, \bar{\tau}_{xy}). \tag{31}$$

The stresses on the upper surface of the avalanche, $\boldsymbol{\tau}_s$, arise from air drag and pressure; they can be safely neglected even at high speeds due to the low density of air, unless one attempts to model the generation of a suspension layer (powder snow cloud) or the motion of sub-aqueous debris flows. The contribution from the deviatoric normal stress, τ_{zz}, can also be neglected unless snow compression effects are to be included in the analysis. The terms $\partial_x(h\bar{\tau}_{zx})$ and $\partial_y(h\bar{\tau}_{zy})$ describe bed-normal forces on a control volume arising from shear in the bed-normal direction. They might play a limited and transient role at pronounced slope breaks; within the shallow-water approximation, they can also be safely neglected.

The most important terms after $\boldsymbol{\tau}_b$ are $\partial_y(h\bar{\tau}_{xy})$ and $\partial_x(h\bar{\tau}_{yx})$ because they describe the force parallel to the terrain that results from shear in a bed-normal plane. Such forces arise, e.g., when an avalanche flows in a narrow gully and friction at the gully sides slows down the outer regions of the flow: Then, these shear stress gradients tend to even out the transverse velocity profile. They have been incorporated in the code TITAN2D (Pitman *et al.*, 2003), but presently MoT-Voellmy follows the large majority of models and neglects these terms. It should, however, be interesting to investigate their effects further.

- *Constant flow density:* ρ_f is assumed constant despite ample evidence to the contrary from laboratory experiments on granular materials, field observations and avalanche experiments. The theory of granular materials shows that the shear and normal stresses grow as the square of the shear rate at constant density. As an avalanche accelerates, the dispersive normal stress would soon exceed the weight of the flow unless the flow expands; the interplay between dispersive effects and flow expansion is responsible for the much more gradual stress increase captured, e.g., in the $\mu(I)$ rheology (Jop *et al.*, 2006). The Voellmy model knows no dispersive stress; as a consequence, the true behavior of avalanche flow must be artificially captured by differently calibrating its empirical parameters for different avalanche sizes and terrain configurations.

- *Hydrostatic pressure:* The normal stresses inside the flowing mass and parallel to the local tangent plane are assumed to be hydrostatic. This is believed to be a valid approximation at least when the flow is steady and fluidized, but it is ostensibly invalid for snow at rest or close to stopping. In dry-snow avalanches,

partial fluidization due to particle collisions prevails during an event, except for the early acceleration and break-up phase and in the late stopping phase.

Non-hydrostatic pressure, distinguishing between active and passive earth pressure, has been incorporated in some models (e.g. Hutter *et al.*, 1993; Bartelt *et al.*, 1999; Pitman *et al.*, 2003; Christen *et al.*, 2010) but is not presently implemented in MoT-Voellmy. Therefore, one expects MoT-Voellmy to show more lateral spreading than those models.

- *No balance equation for energy:* The main consequence of this simplification—used in almost all models—is that the fate of the dissipated energy cannot be tracked. The kinetic energy transferred into heat will alter the properties of the flowing snow and thus the friction to some degree. Kinetic energy that is converted into random motion of the snow particles (granular temperature) instead, will induce dispersive pressure that drives the particles away from each other (fluidization) and may in this way reduce friction. When granular temperature and fluidization are neglected, their effect on the avalanche dynamics must be approximated by adjusting the friction parameters by extensive calibration.

4 Modeling options

In the set-up file for a simulation, besides the choice of entrainment model the user can activate a number of options to describe additional processes. They are briefly described here.

Centrifugal effects The centrifugal acceleration due to terrain curvature causes extra dry friction $\mu\rho_f\kappa_{\boldsymbol{u}}\boldsymbol{u}^2$, where $\kappa_{\boldsymbol{u}}$ is the terrain curvature in the flow direction (see Sec. 2 for the details). This term can be switched on or off.

Earth-pressure coefficient MoT-Voellmy does not distinguish between active and passive earth pressure. It is only possible to multiply the pressure gradient by a fixed coefficient throughout the entire simulation. In very complicated topography, with different branches of the avalanche colliding, a coefficient smaller than 1 may be required to avoid numerical instabilities. However, whenever possible this coefficient should be left at its default value 1.

Modification of the drag term The effective friction coefficient in the Voellmy model depends on the velocity and flow depth as

$$\mu_{\text{eff}} = \mu + k\frac{\mathbf{u}^2}{gh\cos\theta} = \mu + k\text{Fr}^2. \tag{32}$$

In deeply channelized flows, the Froude number Fr can become small and thus can μ_{eff} become unrealistically small. If desirable, this effect can be eliminated

by modifying the drag term as

$$k(\boldsymbol{x}) \longrightarrow \frac{k(\boldsymbol{x})}{1 - \mathrm{e}^{-h_{\text{drag}}/h(\boldsymbol{x},t)}}. \tag{33}$$

(To avoid division by 0, $h(\boldsymbol{x}, t)$ is limited from below by the user-set parameter h_{min}.) For $h_{\text{drag}} \gg h$, the original Voellmy friction law is recovered. For $h_{\text{drag}} \ll h$, MoT-Voellmy effectively behaves like the PCM model (Perla *et al.*, 1980), with

$$\mu_{\text{eff}} \approx \mu + k\frac{h}{h_{\text{drag}}}\frac{\mathbf{u}^2}{gh\cos\theta} = \mu + \frac{k}{h_{\text{drag}}}h\text{Fr}^2, \tag{34}$$

which corresponds to $M/D = h_{\text{drag}}/k$.

Braking effect of forest Especially for small avalanches, forest can have an important protective effect. MoT-Voellmy treats forest differently from RAMMS::AVALANCHE and RAMMS::EXTENDED: The former sets the drag coefficient k (respectively $\xi = g/k$) to a fixed value, irrespective of the forest density and tree size. The latter assumes that detrainment, i.e., the stopping of a fraction of the avalanche mass upstream of the trees, is the dominant braking effect. MoT-Voellmy assumes that each tree exerts a braking force on the avalanche that is independent of the contributions of the other trees, i.e., non-linear effects due to jet interactions and the like are neglected. The contributions from all trees in a computational cell are then averaged and expressed as an extra (bed) shear stress. The latter is therefore proportional to n, the number of trees per (oblique) surface area, the average tree diameter, D, and the instantaneous local flow depth, h.

The calculation of the force exerted by a single tree of diameter D is based on results from experiments with granular flows around a cylindrical obstacle and some discrete-element simulation (Chehata *et al.*, 2003; Wassgren *et al.*, 2003). They can be expressed as contributions to the dry-friction and drag terms:

$$\Delta\mu = 1.25nDh \quad \text{and} \quad \Delta k = 0.5nDh. \tag{35}$$

For historical reasons, the formula for Δk can be modified by a factor C_d (drag coefficient) specified in the run set-up file, but it is highly recommended leaving $C_d = 1$.

Forest destruction by the avalanche The reduction in run-out length due to the modified friction parameters Eq. (35) is far too large in the case of large avalanches because they destroy the forest, which then loses much of its retarding effect. To capture this, MoT-Voellmy compares, in each computational cell and at each time step, the torque exerted by the avalanche on a tree trunk,

$$M(u, h) \approx \frac{1}{2}\rho_f u^2 D h^2, \tag{36}$$

to the resistance of the tree trunks against breaking, expressed in terms of the mean tree diameter and the Modulus of Rupture (MoR) for the tree type in question:

$$M_{b,\max} = \frac{\pi}{32}\mathrm{MoR}\, D^3 \approx (5\text{–}6\,\mathrm{MPa})\, D^3, \tag{37}$$

The MoR has typical values around 50 MPa, but can be lower or higher by a factor 2 or so; it can be specified by the user in the run set-up file. At this point, uprooting is not explicitly considered because this would require knowledge about the local soil properties.

If the torque exceeds the bending resistance of the tree trunks in a cell at some time t_1, the stand density in this cell is reduced exponentially with time,

$$n(t > t_1) = n_0 \cdot \mathrm{e}^{-\lambda(D)\cdot(t-t_1)}, \tag{38}$$

mimicking the fall of the trees, during which their braking effect gradually disappears. The decay constant λ (in s^{-1}) is taken to depend on the trunk diameter as

$$\lambda(D) = 0.15\,\mathrm{m\,s^{-1}} \,/\, D, \tag{39}$$

This formula stipulates that small trees and bushes fall in less than 1 s while a very large tree with $D = 1$ m takes about 5 s. The user can modify the coefficient in the run set-up file.

Evolving bed geometry In 2013, J.-T. Fischer contributed an addition to the code that allows the bed geometry to be modified between time steps. This option makes it possible to study self-channeling effects due to entrainment. However, since the slope and curvature of each cell in the instantaneous computational domain needs to be recomputed at every time step, simulations take significantly longer.

5 The numerical scheme: Method of Transport

The numerical scheme Method of Transport (Fey & Jeltsch, 1993; Fey & Morel, 1995; Fey *et al.*, 1997) used in MoT-Voellmy to solve the three partial differential equations (18) and (19) can be considered its most distinctive feature compared to other codes. In its essence, it is a simple first-order upwind scheme while most other codes use second-order finite-volume approaches with (approximate) Riemann solvers. A special feature is that the fluxes in the x and y-direction are not split, but advection occurs along the flow direction, including diagonal neighbor cells. While the Method of Transport was designed to capture the propagation of shocks in 2D (or any number of dimensions) as accurately as possible and without a bias along the coordinate directions, MoT-Voellmy disregards the wave structure that is found by diagonalizing the left-hand sides of Eqs. (18) and (19) but instead discretizes the pressure-gradient term $\boldsymbol{\nabla}\cdot(h\bar{p})$ explicitly. In this regard, it resembles the decoupled hydrological discretization scheme (DHD) by Cea & Bladé (2015). The latter tries to compensate for a deficit in spreading (of a perfect fluid) at low Froude number by means of an extra diffusion term in the mass balance. Such a term has not been implemented in MoT-Voellmy at present but could be tested in a future version.

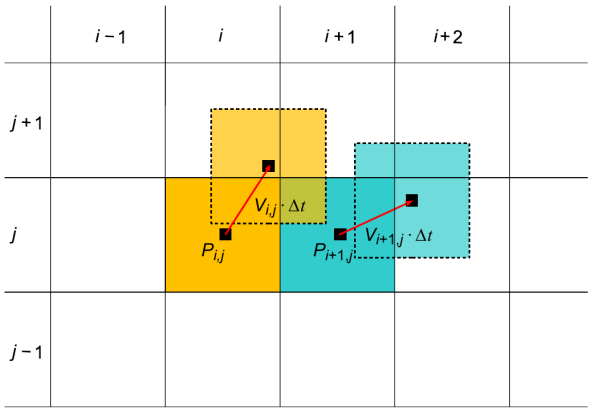

Figure 1 Schematic representation of the treatment of advection in the truncated version of the Method of Transport. $\boldsymbol{v}_{i,j}$ is the velocity vector in the cell (i, j), Δt the time step length. Both mass and momentum are advected in this way. In contrast to other numerical schemes, diagonal neighbors of (i, j) are also included.

Spatial discretization The discretized balance equations can be written compactly in terms of the vector in function space, $\boldsymbol{\Phi} = (h, hu, hv)^{\mathrm{T}}$, as

$$\boldsymbol{\Phi}_{ij}^{(n+1)} \Delta A_{ij} = \boldsymbol{\Phi}_{ij}^{(n)} \Delta A_{ij} + \left[\sum_{(k,l) \in \text{neighbors}} \left(\mathbf{F}_{kl \to ij}^{(n)} - \mathbf{F}_{ij \to kl}^{(n)} \right) + \mathbf{S}_{ij}^{(n)} \Delta A_{ij} \right] \Delta t_n. \quad (40)$$

Here, (i, j) is the cell for which the time evolution is computed, (k, l) are its eight neighbor cells including diagonal neighbors. n and $n + 1$ designate two instances in time separated by the time step Δt_n, i.e., $t_{n+1} = t_n + \Delta t_n$. ΔA_{ij} is the area of cell (i, j). $\mathbf{S}_{ij} = (S_{h,ij}, S_{hu,ij}, S_{hv,ij})^{\mathrm{T}}$ is the vector of source terms, with

$$S_{h,ij}^{(n)} = q_{e,ij}^{(n)} / \rho_f \,, \quad (41)$$

$$\boldsymbol{S}_{h\boldsymbol{u},ij}^{(n)} = h_{ij}^{(n)} \boldsymbol{g}_{\parallel,ij} - \frac{\boldsymbol{u}_{ij}^{(n)}}{\| \boldsymbol{u}_{ij}^{(n)} \|} \left[\mu_{ij}^{(n)} g_{z,ij}^{\prime(n)} + k_{ij}^{(n)} \left(\boldsymbol{u}_{ij}^{(n)} \right)^2 \right] \quad (42)$$

for the mass and momentum balances, respectively. g_z', μ and k can depend on the time: the former due to centrifugal effects, the latter two due to forest resistance.

$\mathbf{F}_{ij \to kl} = (F_{ij \to kl}^{(h)}, F_{ij \to kl}^{(hu)}, F_{ij \to kl}^{(hv)})^{\mathrm{T}}$ is the vector (in function space) of fluxes from cell (i, j) into cell (k, l). Figure 1 shows that these fluxes can be expressed as

$$\begin{aligned} \mathbf{F}_{ij \to i+1,j}^{(h)} &= R(u_{ij})(\Delta y_{i+1,j} - |v_{ij}| \mathrm{d}t) \boldsymbol{\Phi}_{ij} \mathrm{d}t, \\ \mathbf{F}_{ij \to i+1,j+1}^{(h)} &= R(u_{ij}) R(v_{ij}) \boldsymbol{\Phi}_{ij} (\mathrm{d}t)^2, \\ \mathbf{F}_{ij \to i,j+1}^{(h)} &= (\Delta x_{ij} - |u_{ij}| dt) R(v_{ij}) \boldsymbol{\Phi}_{ij} \mathrm{d}t \\ \mathbf{F}_{ij \to i-1,j+1}^{(h)} &= R(-u_{ij}) R(v_{ij}) \boldsymbol{\Phi}_{ij} (\mathrm{d}t)^2 \end{aligned} \quad (43)$$

and analogously for the remaining four directions. The ramp function is defined by $R(x) = \max(0, x)$. The contribution from the diagonal directions is formally $\mathcal{O}((\mathrm{d}t)^2)$ but need not be negligible if the CFL number is in the range 0.5–1. These fluxes are added to the receiver cells and subtracted from the donor cell (i, j) so that mass and momentum conservation is assured up to round-off errors.

The pressure gradient term, $\boldsymbol{\nabla}(h\bar{p})$, is treated separately. Early versions used central differencing involving the two neighbor cells, i.e.,

$$\partial_x(h\bar{p})_{ij} \approx \frac{g_z\, i+1,j h^2_{i+1,j} - g_z\, i-1,j h^2_{i-1,j}}{2(\Delta x_{i-1,j} + \Delta x_{ij})}, \tag{44}$$

but this formulation suffers from checkerboard oscillations because the pressure value at i drops out of the approximation. Also, at the edge of the avalanche, momentum is transferred into cells that do not contain mass yet, leading to infinite acceleration. The scheme implemented presently approximates the pressure values at the cell interfaces by the geometric mean of the cell in question and its neighbor,

$$\partial_x(h\bar{p})_{ij} \approx \rho_f \frac{\sqrt{(h\bar{p})_{i+1,j}(h\bar{p})_{ij}} - \sqrt{(h\bar{p})_{ij}(h\bar{p})_{i-1,j}}}{\Delta x_{ij}}. \tag{45}$$

In this way, checkerboard oscillations are restricted to configurations where the flow depth alternates between 0 and a finite value in a direction. Moreover, no momentum is transferred to empty cells at the edge of the avalanche.

Adaptive time stepping The time step Δt must fulfill the well-known Courant–Friedrichs–Lewy condition for stability: the flow must not traverse more than one cell during a time step. At each time step, MoT-Voellmy sets

$$\Delta t_n = \frac{\mathrm{CFL} \cdot \Delta X}{\max_{(i,j)\in\mathcal{D}} \left(\|\boldsymbol{u}_{ij}^{(n)}\| + \sqrt{h_{ij}^{(n)} g_{z,ij}^{\prime(n)}} \right)}, \tag{46}$$

where CFL isa user-selected number between 0 and 1, a typical value being 0.8. $\Delta X \leq \Delta x, \Delta y$ is the spacing of the grid when projected on a horizontal plane. In addition, the user can set upper and lower limits for Δt to reduce time discretization errors and to force the simulation to stop if numerical instabilities lead to extremely small time steps. The time looping continues until the user-selected maximum simulated time is reached or exceeded or a stopping criterion is met.

Adaptive computational domain At each time step after the first, the computational domain $\mathcal{D}$ in Eq. (46) is adjusted to the minimum rectangle comprising the flow domain plus a one-cell-wide buffer zone. The flow domain is determined by the criterion $\|\boldsymbol{u}\| > u_{\min}$, with $u_{\min}$ a user-specified speed cut-off.

Stopping criteria The model intercomparison study ISeeSnow (Wirbel *et al.*, 2024) has highlighted how important the stopping criteria applied in numerical models of the Coulomb or Voellmy type are—the simulated run-out differences may differ much under otherwise equivalent conditions. At this point, several stopping criteria have been implemented in MoT-Voellmy, but their effects have not been investigated systematically yet; instead, one has mostly relied on suitably choosing the maximum simulated time to make sure the front has (nearly) come to a stop while the slow creeping, which is characteristic of this type of models, has not masked the true result yet.

The following mechanisms are available:

- If the speed in a cell drops below a user-selected value u_{min}, that cell is stopped, but it may start moving again under given conditions (see the last criterion).
- If the velocity in a cell would change sign in a time step, the cell is stopped and cannot start moving again before the next time step.
- For a cell at rest to start moving (again), the vectorial sum of gravity and the pressure-gradient force must exceed the maximum mobilizable dry friction.
- If the sum over all cells of the absolute values of specific momentum,

$$J(t_n) := \sum_{(i.j)\in\mathcal{D}} \|\boldsymbol{u}_{ij}^{(n)}\|\, h_{ij}^{(n)} \Delta A_{ij}, \tag{47}$$

  drops below a user-selected threshold J_{min}, the avalanche is considered stopped and the simulation terminated. J_{min} should be chosen with the size of the avalanche in mind, e.g., $1\,\text{m}\,\text{s}^{-1} \cdot V_{\text{max}}$, with V_{max} either the release volume if entrainment is not simulated or the maximum expected avalanche volume otherwise.
- A run is terminated if the simulated time exceeds a user-selected value t_{max} or if the time step Δt drops below the user-selected limit Δt_{min}, indicating numerical instability. At typical grid spacings of 5–10 m, $\Delta t_{\text{min}} \sim 0.001\,\text{s}$ is usually adequate, but in difficult terrain lower values may occasionally be required.

6 Concluding remarks

MoT-Voellmy is designed to be a simple and fast model for simulating dense gravity mass flows like snow or rock avalanches. Its physical adequacy is strongly limited by the Voellmy friction law, which is, however, still the dominating modeling approach in 2025. Given the uncertainties in the calibration of the friction parameters μ and k and in the implementation of adequate stopping criteria, MoT-Voellmy emphasizes performance instead of high numerical accuracy and has therefore been chosen as the computing engine in NAKSIN, the Norwegian tool for generating country-wide avalanche hazard indication maps (Issler *et al.*, 2023).

The key simplification in MoT-Voellmy compared to most other comparable models is the neglect of resolving the shock structure. The author hopes to implement the full Method of Transport in the near future to quantitatively assess the trade-off between accuracy and speed gain in MoT-Voellmy.

The simple structure of the code makes it fairly straightforward to create variants that implement different friction laws, like MoT-muI by (Tregaskis *et al.*, 2023) or the two-layer model for mixed avalanches, MoT-PSA (Vicari & Issler, 2025), or other capabilities. Some worthwhile further developments could be

- implementing the extended NIS model (Issler & Gauer, 2008) as a quasi-3D code by adding a shear-rate dependent evolution equation for the flow density,
- implementing the depth-averaged Herschel–Bulkley rheology for modeling the flow of cohesive soils,
- options for other boundary conditions than von Neumann's to compute 1D flows efficiently or impenetrable cells inside the flow domain to model impacts against buildings, and
- adapting the instantaneous computational domain not to a rectangle but to a subset of cells following the edge of the active flow with a buffer zone to increase the computation speed further so that very large submarine flows can be simulated efficiently.

Acknowledgements

I wish to thank Maja Adamska-Szatko for the pleasant collaboration during her stay at NGI and for prompting me to create the first version of MoT-Voellmy as a faster and practice-oriented variant of the code DIMAS she developed for her PhD thesis, Jan-Thomas Fischer for contributing the option of tracking geometry changes in 2013, and Callum Tregaskis and Hervé Vicari for some code fixes in 2022/2023. Some later additions to MoT-Voellmy, code testing and making it open-source have partly been financed through the grant for avalanche research to NGI from the Norwegian Department of Petroleum and Energy, administrated by the Norwegian Directorate of Waterways and Energy (NVE).

References

Adamska-Szatko, M. (2012). *Modelowanie matematyczne powierzchniowych zjawisk dynamicznych z zastosowaniem automatów koórkowych [Mathematical Modeling of Surface Dynamic Processes with the Use of Cellular Automata]*. PhD thesis, Akademia Górniczo-Hutnicza (AGH University of Science and Technology), Dept. of Geoinformatics and Applied Computer Science, Cracow, Poland.

Barker, T., Schaeffer, D. G., Shearer, M. & Gray, J. M. N. T. (2017). Well-posed continuum equations for granular flow with compressibility and $\mu(I)$-rheology. *Proc. R. Soc. (London), Ser. A* **473**, 20160,846. doi:10.1098/rspa.2016.0846.

Bartelt, P., Bühler, Y., Christen, M., Deubelbeiss, Y., Salz, M. *et al.* (2017). *RAMMS::AVALANCHE User Manual*. URL https://ramms.slf.ch/fileadmin/user_upload/WSL/Microsite/RAMMS/Downloads/RAMMS_AVAL_Manual.pdf. Version 1.7.0.

Bartelt, P., Christen, M., Bühler, Y., Caviezel, A. & Buser, O. (2018). Snow entrainment: avalanche interaction with an erodible substrate. *In: Proc. of the Intl. Snow Science Workshop, Innsbruck, Austria, 2018*, pp. 716–720. URL https://arc.lib.montana.edu/snow-science/objects/ISSW2018_O08.6.pdf.

Bartelt, P., Salm, B. & Gruber, U. (1999). Calculating dense snow avalanche runout using a Voellmy-fluid model with active/passive longitudinal straining. *J. Glaciol.* **45**(150), 242–254. URL http://www.ingentaconnect.com/content/igsoc/jog/1999/00000045/00000150/art00007.

Bartelt, P., Vera Valero, C., Feistl, T., Christen, M. & Bühler, Y. (2015). Modelling cohesion in snow avalanche flow. *J. Glaciol.* **61**(229), 837–850. doi:10.3189/2015JoG14J126.

Bouchut, F. & Westdickenberg, M. (2004). Gravity driven shallow water models for arbitrary topography. *Comm. Math. Sci.* **2**(3), 359–389.

Cea, L. & Bladé, E. (2015). A simple and efficient unstructured finite volume scheme for solving the shallow water equations in overland flow applications. *Water Resour. Res.* **51**, 5464–5486. doi:10.1002/2014WR016547.

Chehata, D., Zenit, R. & Wassgren, C. R. (2003). Dense granular flow around an immersed cylinder. *Phys. Fluids* **15**(6), 1522–1531. doi:10.1063/1.1571826.

Christen, M., Kowalski, J. & Bartelt, P. (2010). RAMMS: Numerical simulation of dense snow avalanches in three-dimensional terrain. *Cold Regions Sci. Technol.* **63**(1–2), 1–14. doi:10.1016/j.coldregions.2010.04.005.

Discoe, B. (2007). The BT (Binary Terrain) File Format. URL http://vterrain.org/Implementation/Formats/BT.html. Last accessed 2024-10-03.

Eglit, M. E. & Demidov, K. S. (2005). Mathematical modeling of snow entrainment in avalanche motion. *Cold Regions Sci. Technol.* **43**(1–2), 10–23. doi:10.1016/j.coldregions.2005.03.005.

ESRI, I. (2020). Esri ASCII raster format. Website. URL https://desktop.arcgis.com/en/arcmap/latest/manage-data/raster-and-images/esri-ascii-raster-format.htm.

Fey, M. (1998a). Multidimensional upwinding. Part I. The Method of Transport for solving the Euler equations. *J. Comp. Phys.* **143**, 159–180. doi:10.1006/jcph.1998.5958.

Fey, M. (1998b). Multidimensional upwinding. Part II. Decomposition of the Euler equations into advection equations. *J. Comp. Phys.* **143**, 181–199. doi:10.1006/jcph.1998.5959.

Fey, M. & Jeltsch, R. (1993). A new multidimensional Euler-scheme. *In:* Donato, A. & Oliveri, F. (eds.), *Nonlinear Hyperbolic Problems: Theoretical, Applied, and Computational Aspects. Proc. Fourth Intl. Conf. Hyperbolic Problems, Taormina, Italy, April 3–8, 1992*, *Notes on Numerical Fluid Mechanics (NNFM)*, vol. 43, pp. 227–234. Vieweg+Teubner Verlag, Braunschweig/Wiesbaden, Germany. doi:10.1007/978-3-322-87871-7_27.

Fey, M., Jeltsch, R., Maurer, J. & Morel, A.-T. (1997). The method of transport for nonlinear systems of hyperbolic conservation laws in several space dimensions. Tech. Rep. 97, Seminary for Applied Mathematics, ETH Zurich, Zürich, Switzerland.

Fey, M. & Morel, A.-T. (1995). Multidimensional method of transport for the shallow water equations. resreport 95-05, ETH Zurich, Seminary of Applied Mathmatics. URL https://www.sam.math.ethz.

ch/sam_reports/reports_final/reports1995/1995-05_fp.pdf.

Fraccarollo, L. & Capart, H. (2002). Riemann wave description of erosional dam-break flows. *J. Fluid Mech.* **461**, 183–228. doi:10.1017/S0022112002008455.

Gray, J. M. N. T. & Edwards, A. N. (2014). A depth-averaged $\mu(I)$-rheology for shallow granular free-surface flows. *J. Fluid Mech.* **755**, 503–534. doi:10.1017/jfm.2014.450.

Grigorian, S. S. & Ostroumov, A. V. (2020). On a continuum model for avalanche flow and its simplified variants. *Geosci.* **10**(1), 35. doi:10.3390/geosciences10010035.

Hutter, K., Siegel, M., Savage, S. B. & Nohguchi, Y. (1993). Two-dimensional spreading of a granular avalanche down an inclined plane. Part I. Theory. *Acta Mech.* **100**, 37–68. doi:10.1007/bf01176861.

Issler, D. (2006). Curvature effects in depth-averaged flow models on arbitrary topography. NGI Report 20021048–14, Norwegian Geotechnical Institute, N–0806 Oslo, Norway.

Issler, D. (2014). Dynamically consistent entrainment laws for depth-averaged avalanche models. *J. Fluid Mech.* **759**, 701–738. doi:10.1017/jfm.2014.584.

Issler, D. (2020). Comments on"On a Continuum Model for Avalanche Flow and Its Simplified Variants" by S. S. Grigorian and A. V. Ostroumov. *Geosci.* **10**(3), 96. doi:10.3390/geosciences10030096.

Issler, D. & Gauer, P. (2008). Exploring the significance of the fluidized flow regime for avalanche hazard mapping. *Ann. Glaciol.* **49**(1), 193–198. doi:10.3189/172756408787814997.

Issler, D., Gisnås, K. G., Gauer, P., Glimsdal, S., Domaas, U. *et al.* (2023). NAKSIN –– a new approach to snow avalanche hazard indication mapping in Norway. doi:10.2139/ssrn.4530311. URL https://ssrn.com/abstract=4530311. Preprint.

Jop, P., Forterre, Y. & Pouliquen, O. (2006). A constitutive law for dense granular flows. *Nature* **441**(7094), 727–30. doi:10.1038/nature04801.

Noelle, S. (2000). The MoT-ICE: A new high-resolution wave-propagation algorithm for multidimensional systems of conservation laws based on Fey's Method of Transport. *J. Comp. Phys.* **164**(2), 283–334. doi:10.1006/jcph.2000.6598.

Oesterle, F., Tonnel, M., Wirbel, A. & Fischer, J.-T. (2021). com1DFAOrig: Original DFA-Kernel. Webpage. doi:https://zenodo.org/badge/latestdoi/281922740. URL https://docs.avaframe.org/en/latest/moduleCom1DFAOrig.html.

Parker, G., Fukushima, Y. & Pantin, H. M. (1986). Self-accelerating turbidity currents. *J. Fluid Mech.* **171**, 145–181. doi:10.1017/S0022112086001404.

Perla, R., Cheng, T. T. & McClung, D. M. (1980). A two-parameter model of snow avalanche motion. *J. Glaciol.* **26**(94), 197–207. doi:10.3189/s002214300001073x.

Peruzzetto, M., Mangeney, A., Bouchut, F., Grandjean, G., Levy, C. *et al.* (2021). Topography curvature effects in thin-layer models for gravity-driven flows without bed erosion. *J. Geophys. Res.: Earth Surf.* **126**(4), e2020JF005,657. doi:10.1029/2020JF005657.

Pitman, E. B., Nichita, C. C., Patra, A., Bauer, A., Sheridan, M. *et al.* (2003). Computing granular avalanches and landslides. *Phys. Fluids* **15**(12), 3638–3648. doi:10.1063/1.1614253.

Pudasaini, S. P. & Hutter, K. (2007). *Avalanche Dynamics*. Springer, Berlin Heidelberg, Germany.

Steinkogler, W., Gaume, J., Löwe, H., Sovilla, B. & Lehning, M. (2015). Granulation of snow: From tumbler experiments to discrete element simulations. *J. Geophys. Res.* **F120**(6), 1107–1126. doi:10.1002/2014JF003294.

Steinkogler, W., Sovilla, B. & Lehning, M. (2014). Influence of snow cover properties on avalanche dynamics. *Cold Regions Sci. Technol.* **97**, 121–131. doi:10.1016/j.coldregions.2013.10.002.

Tregaskis, C., Gauer, P. & Issler, D. (2023). A depth-averaged $\mu(I)$ model for dense snow avalanches. *In: Proc. Intl. Snow Science Workshop 2023, Bend, Oregon*, pp. 504–511. International Snow Science Workshop, University of Montana Library, Bozeman, Montana. URL https://arc.lib.montana.edu/snow-science/objects/ISSW2023_P1.65.pdf.

Vera Valero, C., Wikstroem Jones, K., Bühler, Y. & Bartelt, P. (2015). Release temperature, snow cover entrainment and the thermal flow regime of snow avalanches. *J. Glaciol.* **61**(225), 173–184. doi:10.3189/2015JoG14J117.

Vicari, H. & Issler, D. (2025). MoT-PSA: a two-layer depth-averaged model for simulation of powder snow avalanches on three-dimensional terrain. *Ann. Glaciol.* **65**, e16. doi:10.1017/aog.2024.10.

Voellmy, A. (1955). Über die Zerstörungskraft von Lawinen. *Schweiz. Bauztg.* **73**(12, 15, 17, 19), 159–165, 212–217, 246–249, 280–285.

Wassgren, C. R., Cordova, J. A., Zenit, R. & Karion, A. (2003). Dilute granular flow around an immersed cylinder. *Phys. Fluids* **15**(11). doi:10.1063/1.1608937.

Wikipedia authors (2002–2024). Curvature. Wikipedia article. URL https://en.wikipedia.org/wiki/Curvature#Second_fundamental_form.

Wirbel, A., Oesterle, F., Fischer, J.-T., Chambon, G., Faug, T. *et al.* (2024). ISeeSnow – initiating an avalanche simulation tool intercomparison. Abstract EGU24-17750. doi:10.5194/egusphere-egu24-17750. EGU General Assembly 2024, Vienna, Austria, 14–19 Apr 2024.

NGI (Norwegian Geotechnical Institute) is a leading international centre for research and consulting within the geosciences. NGI develops optimum solutions for society and offers expertise on the behaviour of soil, rock and snow and their interaction with the natural and built environment.

NGI works within the following sectors: Offshore energy – Building, Construction and Transportation – Natural Hazards – Environmental Engineering.

NGI is a private foundation with office and laboratory in Oslo, branch office in Trondheim and daughter companies in Houston, Texas, USA and in Perth, Western Australia.

www.ngi.no

NGI (Norges Geotekniske Institutt) er et internasjonalt ledende senter for forskning og rådgivning innen ingeniørrelaterte geofag. Vi tilbyr ekspertise om jord, berg og snø og deres påvirkning på miljøet, konstruksjoner og anlegg, og hvordan jord og berg kan benyttes som byggegrunn og byggemateriale.

Vi arbeider i følgende markeder: Offshore energi – Bygg, anlegg og samferdsel – Naturfare – Miljøteknologi.

NGI er en privat næringsdrivende stiftelse med kontor og laboratorier i Oslo, avdelingskontor i Trondheim og datterselskap i Houston, Texas, USA og i Perth, Western Australia.

www.ngi.no

OSLO
TRONDHEIM
HOUSTON
PERTH

NORWEGIAN GEOTECHNICAL INSTITUTE
NGI.NO

Main office
PO Box 3930 Ullevaal St.
NO-0806 Oslo
Norway

Trondheim office
PB. 5687 Torgarden
NO-7485 Trondheim
Norway

T (+47)22 02 30 00
F (+47)22 23 04 48
NGI@ngi.no

BIC NO. DNBANOKK
IBAN NO26 5096 0501 281
ORGANISATION NO.
958 254 318MVA

ISO 9001/14001
CERTIFIED BY BSI
FS 32989 / EMS 612006