Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2018-67-AC3, 2018

Geoscientific Model Development Discussions

Open Access

EGU

Interactive comment on "faSavageHutterFOAM 1.0: Depth-integrated simulation of dense snow avalanches on natural terrain with OpenFOAM" *by* Matthias Rauter et al.

Matthias Rauter et al.

matthias.rauter@uibk.ac.at

Received and published: 23 May 2018

The manuscript faSavageHutterFOAM 1.0: Depth-integrated simulation of dense snow avalanches on natural terrain with OpenFOAM is a well-written contribution to the field of modeling and simulation of dense snow avalanches and fits into the scope of the journal. It extends the description of the OpenFOAM implementation first published by one of the authors, Matthias Rauter in (Rauter and Tukovic, 2018), to realistic topographies. The authors introduce the OpenFOAM specific workflow with a special focus on mesh generation from DEM data and GIS compatible post-processing. They demonstrate the capability of the proposed solver while analyzing a specific case

Printer-friendly version

Discussion paper

study, namely the Wolfsgruben avalanche. Simulation results of the new OpenFOAM based solver are compared to the established tool samosAT.

Dear Dr. Kowalski,

thank you very much for your detailed review. You uncovered some noteworthy aspects which we try to discuss in the following as well as in the revised manuscript.

My overall impression of this article is very positive due to the following reasons:

• First of all, by implementing the mathematical model into the mature software framework OpenFOAM, the authors actually outsource a lot of issues associated with e.g. data structure, parallelization and linear solvers. This allows a focus on the mathematical/physical model formulation itself, which is in my opinion the correct way to go.

-

• The mathematical/physical model itself is passed over to the solver in an encapsulated way as demonstrated on page 6 line 20 - 30. I find this a very big advantage of the approach. In fact this seems to be so straight forward, that one of the other reviewers even objected on the relative ease with which the model can be applied to real hazard scenarios, and the responsibility that arises from that. I certainly do have a slightly different opinion: While we as modelers do indeed have the responsibility to investigate and clearly communicate any limitations to the model, I fully support the authors approach in aiming towards modularizing software development and mathematical model representation as far as possible.

Thank you very much. Indeed, this was the main motivation of this work.

• Another aspect of the paper that I like very much is the comparison to an alternative simulation model. Although I have to admit that personally I find SamosAT a somewhat surprising choice (see my comment further down), this in principle is exactly the kind of results that help to better assess scope, potential and limitation of the various software tools that are in use.

-

• Finally, I want to emphasize the author's attempt to publish reproducible results. The paper refers to the code, which can be downloaded and tested. All in all, I am clearly in favor of publishing this paper and think it will be a valuable contribution to the field, if the following objections have been addressed.

-

• Title seems odd: Probably the title is chosen according to the OpenFOAM module - still I think it doesn't reflect the content of the paper. While the mechanical model that the authors solve is similar to the Savage-Hutter model (see also my next comment), some of the original Savage Hutter theory's defining flavors, e.g. its questionable active-passive earth-pressure term and the 'dry-friction-only' resistive force, are absent. This , however, is also OK, as it does not seem to be what the authors want to promote, compare and discuss in their paper. Straight-on question: Why not faavalancheFOAM, for example?

According to the journal policy, the main paper must give the model name and version number (or other unique identifier) in the title.

The OpenFOAM solver itself is called faSavageHutter to distinguish it from strictly two-dimensional ones with horizontal velocity (e.g. ShallowFOAM, https://github.com/mintgen/shallowFoam). The name is also owed to the historical development, as the classic Savage-Hutter model was our starting point. We kept the name because we stuck to the mechanical model of Savage & Hutter. It is true that we changed closure (or process) models to match the current practice. However, these changes do not influence solver code but solely added user-selectable submodels. Similarly, OpenFOAMs Navier-Stokes solver do not change their name when applying non-Newtonian viscosity models.

We will consider renaming the solver to 'faAvalancheFoam' in the future, however, the name of the solver in the current OpenFOAM release (OpenFOAM-v1712) can not be changed.

We will explain that the name refers to the mechanical description in the revised manuscript.

• I like how you distinguish between the 'mechanical model' on the one side and the 'process model' on the other side. Personally, I would refer to it as the kinematic description and the dynamic closure, but it essentially means the same thing. Since it is important to differentiate between the two, I suggest to have the content of the footnote on page 2 integrated into the main text. As a side remark: I had to read the footnote several times, in order to digest and understand it. I think readability would benefit from an additional comma between 'integration' and 'and'. Or maybe rephrase into two sentences. Generally, I think it would be good to explain the name faSavageHutteFOAM (if you want to stick to it) based on the difference between mechanical and process model. It seems to me that the 'mechanical model' solved in this paper is similar to the Savage-Hutter mechanical model, while the 'process model' had been altered.

Distinguishing between 'mechanical model' and 'process model' or kinematic description and dynamic closure is common practice in the OpenFOAM community.

We will move the footnote into the text and rework phrasing to increase readability in the next revision.

• Section 2.1: In the interest of the reader, it would be good to work over the structure of section 2.1.. Two concrete suggestions: 1) Include a brief explaination of equations (1)-(3) just after they have been introduced rather than stating: Variables and mathematical operators are explained later .... I find it OK to refer to Rauter and Tukovic (2018) for details, but it must be possible to understand the paper without. In that context: Is there is typo in the last term of equation (2), what's the 'e'? 2) Why not taking up the earlier thought ('mechanical model' vs 'model closure') in the structure of section 2.1, e.g a subsection mechanical model and its solution (description of the numerics, etc.), and a subsection on model closures , e.g. equations (9) - (12).

We tried to follow the structure you suggest:
P. 4, line 5 to p. 5, line 16: kinematic description,
p. 5, line 17 to p. 6, line 12: dynamic closure,
p. 6, line 13 to p. 6, line 31: numerical solution and implementation.

We will add the respective sub-sub-sections entries in the revised manuscript:
2.1 Flow model,
2.1.1 Kinematic description,
2.1.2 Dynamic closure and
2.1.3 Numerical solution.

You're right about that equations are not explained properly, for readers not familiar with the classic Savage-Hutter model. We will explain individual terms in Eqs. (1) - (3), i.e. temporal derivative, advection, entrainment, basal friction and lateral pressure gradient. Moreover, we will add an introduction into surface projections to the appendix (see answers to reviewer 1 and 2).

The 'e' is indeed a typo, which we will remove in the revised version.

• Section 2.1: page 6 equation (11). Is z here the actual altitude rather than vertical coordinate? If so please make this more explicit.

z is the actual altitude and it has to be consistent with the vertical coordinate in order for our preprocessing tool to work. We will make this more explicit in the revised manuscript.

• Section 2.1: page 6 line 20-30: I like including a code snippet. Given that the finite area aspect is new to most readers it would be very instructive to see a code snippet that has either a tangential or a normal operator in it. Does OpenFOAM have different operators for this? Or do you specifically implement their definition into the code?

Historically, the standard operators, `fam::grad(...)` and `fam:div(...)` contain the surface tangential projection if the respective result is a vector. The operators `fam::ngrad(...)` and `fam:ndiv(...)` have been added lately and contain the surface normal projection. All operators are mentioned in Rauter and Tukovic (2018) and we would prefer to refer to it in the revised manuscript.

• Section 2.2: page 6 line 14: 'Numerical diffusivity' does not 'prevent' oscillations!

Diffusive behavior is rather a feature of first-order methods, while oscillations and lower diffusivity is a feature of second order methods. Please adjust the sentence accordingly.

We understand that this paragraph is misleading and we will rework it in the revised manuscript.

• The authors talk about the model to work in 'moderately curved' topographies several times. Is it possible to quantify this somehow? What curvature values can be dealt with and what values are critical and can't be dealt with? Or from the perspective of a potential user of the code: Can you put this rather fuzzy observation in a concrete guideline and provide information for which cases it is not possible to use the code? If this is not possible, at least lay out what would have to be done to find such thresholds?

Greve et al. (1994) assumed a ratio between typical avalanche lenght L and curvature radius R of order $O(\epsilon^{1/2})$, and a ratio between flow thickness H and avalanche length of order $O(\epsilon)$, where $\epsilon \ll 1$. From this, we can deduce that $H/R = O(\epsilon^{1/2})$, i.e. that flow height is significantly smaller than the curvature radius. This allows depth-integration in surface normal direction. In our experience, this assumption is often violated in practice. At the moment, it is not possible for us to quantify the effects of strong curvature on results. A corresponding investigation would certainly be appropriate, but would go beyond the scope of this work. We will therefore explain what 'mildly curved' means and add a comment on strong curvature in the outlook of the revised manuscript.

• Section 2.3: I like the description of the opensource workflow. A simple question that I could imagine some readers (inlcuding me) would be interested in: Is it

Printer-friendly version

Discussion paper

also possible to run the OpenFOAM Solver with any arbitrary unstructured polygonial surface mesh? If yes, in which format does it have to come? Can you comment on this?

The mesh should be a valid Finite Volume / Finite Area Mesh, which enforces some constraints, e.g. mostly flat and convex faces.

Moreover, the solver expects the mesh in OpenFOAM format. However, conversion tools for many formats are available (see https://cfd.direct/openfoam/user-guide/standard-utilities/, section 3.6.3, Mesh conversion). Moreover, a custom conversion tool can be written using the OpenFOAM C++ library (see, e.g. supplementary material, file slopeMesh/slopeMesh.C, lines 307-362, where a new mesh is created with known edge points and connections). Finally, the OpenFOAM mesh format consists of ascii-files and can therefore be easily written with custom tools (see, e.g. supplementary material, file scripts/txt2mesh.py, lines 472-670). Note that the Finite Area Mesh file format just contains references to Finite Volume Mesh files, meaning that a Finite Volume Mesh has to be present. However, this can be handled by creating a dummy FVM. We will add a hint to mesh conversion tools in the revised manuscript.

• Section 4: What the authors refer to as numerical uncertainty seems to be established in Roache 1997. Please repeat its definition here in the paper as it is unclear. It should be simply a measure of the accuracy of the numerical method, right? In that case the important aspect is that it can be controlled by the scheme itself (whereas typically other sources of uncertainty can't be easily controlled). This is why I don't like the expression 'numerical uncertainty too much, personally). I do disagree with your statement on page 15 around line 5, that the numerical implementation influences the results dramatically. I rather see a tendency that carefully implemented and quality assessed numerical schemes of different type solving the same mathematical model get closer and closer. What do you mean here exactly?

The uncertainty estimation following Roache is indeed a measure of the accuracy and an estimation of the related error, based on a Richardson Extrapolation. This is mentioned in the method section, page 7, line 25.

Sod (1978) showed high differences between different methods when solving the compressible Navier-Stokes equations. Ferziger and Peric (2002), for example, report that there are substantial differences in results, even when using the same method/model: "Recent comparative studies in which the same problem was solved by different groups using different codes reveal that the differences between solutions obtained using different codes with the same models are often larger than the differences between solutions obtained using the same code and different models (Bradshaw et al., 1994; Rodi et al., 1995). These differences can only be due to numerical error or user mistakes, if the models are really the same (it is not unusual that supposedly same models turn out to be different due to different interpretation, implementation, or boundary treatment)." Therefore, we think that differences between SamosAT and our code have to be expected. However, we see that referring to Sod (1987) may be misleading at this point and we will change this reference to Ferziger and Peric (2002) in the revised manuscript.

• Section 4: I like the comparison of the OpenFOAM results with results of other codes. To me it would however seem very valuable to compare against a model that is more similar in spirit, e.g. the Voellmy Salm portion of the r.avaflow project for instance, simply because it is also implemented as a finite volume based method, and the sources for differences could be tracked down more easily. Any reasons, why you chose the samosAT path instead?

We chose SamosAT because we trust its method (Lagrangian; SPH) in terms of complex topography (see manuscript, page 7, lines 7-9) and because we have experience using it (Fischer et al. 2015; Rauter et al. 2016). In particular, r.avaflow, as described by Mergili et al. (2012), seems to be limited in terms of complex topography, since it is using the model of Gray et al. (1999).

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2018-67, 2018.