Review of *The Utrecht Finite Volume Ice-Sheet Model (UFEMISM version 2.0) – part 1: description and idealised experiments*

General Impression

The manuscripts describes a new version of a finite-volume ice-flow model solving different approximations to the Stokes equations, namely, 1st order (a.k.a. Blatter-Pattyn), shallow ice approximation (SIA) in combination with shallow shelf approximation (SSA) and a higher order, vertically integrated scheme, DIVA.

The article lists in great detail the differential equations representing the aforementioned approximations, the basal sliding parametrizations and the solution of the vertically integrated equation of mass conservation. This is followed by a chapter that addresses parallel execution of the code (including aspects of performance and scalability), a recently introduced interface with the NetCDF library and the code's development ecosystem. Thereafter, attention is focused on – according to the authors – verification of the code by comparison with the ensemble outputs from several inter-comparison projects, namely, ISMIP-HOM, (an altered and reduced version of) MISMIP EXP1 and MISMIP+ experiment ice1r. After a brief conclusion chapter, a rather extensive Appendix presenting certain aspects of mesh-generation and remapping and numerical discretization-schemes concludes the article.

Topically, this article appears to be well suited for this journal. I can, to a large extend follow the concept of the manuscript. Graphs are generally well implemented not overcrowded and easy to interpret (something I value). I find the mesh-adaptivity in this new version a particular interesting feature.

Nevertheless, I need to point out a few issues I found in the manuscript. What I do not really understand, is, why a new model variant has to be presented in two different papers? In my opinion – e.g., by reducing the details on presenting Stokes approximations that can be looked up in standard literature (e.g., Greve and Blatter, 2009) and numerical concepts in the Appendix – it would be more convenient for the reader to access all information (whatever is planned for part 2) from within a single manuscript, in particular as part 2 to me seems to contain information that appears to me as essential to understand statements on performance and scalability (or the lack of the latter). Furthermore, in parts, I cannot understand motivation of altered intercomparison experiments (MISMIP) or the interpretation of results (ISMIP-HOM) in terms of model and approximation accuracy. I would see the necessity to address these points (to which I will refer in the next chapter) in details, before I would consider this article to be ready for publication.

Main points of critics

Here I summarize the main points I see necessary to be addressed in an elaborated fashion. Please, find references to them also in the part of *Detailed list of requested changes and elaborations*. The main topics I see necessary to be addressed in the manuscript are:

1 In parts inaccurate description of parallel implementation

This criticism mainly refers to section 3, and in particular sub-section 3.1. I find a few claims that I simply think are not correct or conclusions that are in my opinion unjustified, related to the

information given therein. The authors seem to have Message Passing paradigm (MPI) introduced into UFEMISM v2. I can find little information on how exactly this has been achieved. From the sentence "… data is distributed over many memory chips" I assume that they are applying something like a domain-decomposition to distribute the mesh over different tasks. I understand that the solution step (should one apply approximations that demand solution of a matrix system) is taken over by PETSc (which comes with an MPI interface), but, for instance, in SIA the algorithm will have to evaluate hydrostatic pressure gradients across domain boundaries – how is this achieved? What kind of MPI communication (blocking, non-blocking) has been implemented for exchanging data? What also confuses me, is the quote "*UFEMISM offers a set of standardised routines that interface with the OpenMPI library (Gabriel et al., 2004) to facilitate this*". MPI is a standard and there are several implementations on library-level, such as OpenMPI, MPICH or MVAPICH and also vendor specific MPI-libraries. What part in the code makes it necessary to particular utilize OpenMPI? This is in my opinion not unimportant, in particular in view of the initial claim in the abstract, that UFEMISM should be ready to be integrated into earth system models (ESM). ESM's generally (by the high demand from their atmospheric and ocean components) are run on dedicated supercomputers that in many cases restrict the usage of anything else than a vendor MPI (implied by the interconnection network in place) that not necessarily is derived from OpenMPI. I would ask to **add more information on how exactly MPI has been implemented for the different approximations, how PETSc is integrated on MPI level and why there is a suggested restriction to a single MPI implementation (OpenMPI)?**

2 In parts inaccurate description and conclusions of scaling and performance analysis

Another point I see the need to be improved is the description of performance and scalability. To me the only relevant information on performance can be derived from Figure 2, where some performance measure in terms of seconds (I guess wall clock time?) per 1000 simulated years is to be found in the annotation of the y-axis. Yet, the information there, in my opinion, is too limited to draw any clear conclusion on performance and scalability of UFEMISM v2.0. First, what model is run (1st order, hybrid SIA/SSA or DIVA) for the reported scalability test? Similar issue with the claim to run millennial Antarctic "basin-scale" simulations on a laptop – what approximation was used there and what size of problem are we looking at? Secondly, what computational platform are the tests from Fig. 2 ran on? Judging by the amount of cores, to me it appears to be a single node of a computing-cluster. From my own experience, often there are situations with memory-bound codes (and generally finite volume, finite difference, finite element fall under that category) on, e.g., AMD-EPYC systems (with versions that exactly have 2 sockets a 64 = 128 cores per node), where performance on the single node drops first at 8 cores (due to L3 cache misses) and then at 32 cores because of insufficient or insufficiently used memory bandwidth (the architecture has 4 NUMA domains) – something I also can observe in Fig. 2., except that intra-node scaling completely breaks down past 32 cores, which raises my doubts that inter-node scaling can be achieved at all. Also, intra-node performance highly depends on the implementation (e.g., Byckling et al., 2017). A single non-performing serial section in an OpenMP threaded application has the potential to destroy scalability (Amdahl, 1960). A further problem (should Fig 2 demonstrate runs confined to a single node) I have to point out that one cannot deduce inter-nodal from intra-nodal scalability – to answer this, I would ask the authors to provide scalability results run over several distributed memory nodes. Third and final point of critics is that scaling of a dimensionally reduced flow-line problem as MISMIP in my opinion is not representative of a full Antarctic setup, assuming that applications like that are the final goal to achieve scaling with. The authors must have performed MISMIP+ spinup (*ice0r*) to get a starting point for the reported melt experiment

(*ice1r*) – why not reporting performance/scaling numbers from that setup? To summarize, I would see the **necessity of elaborating the circumstances (platform, compiler, compiler-optimization flags, and most important applied approximation to Stokes) that lead to those scalability results and (provided we are looking at intra-nodal scaling here) extend to inter-nodal scalability tests to be in a position to evaluate code scalability for distributed memory applications, if possible for applications that solve (parts of) whole ice-sheets rather than flow-line setups**. Furthermore, **I ask to provide wall-clock times concerning the performance, in particular** of the most versatile approximation (**1st order/Blatter-Pattyn**) – which by ISMIP-HOM results to me appears to be the only option if sufficient accuracy is sought - **on large scale applications, such as Antarctica or MISMIP+.**

3 Missing information on constraints of applicability of the applied approximations in view of intercomparison results and code verification

The manuscript contains intercomparison results for ISMIP-HOM, MISMIP and MISMIP+ runs with the new code version of UFEMISM. The authors use these comparison of the results of different approximations (1st order, DIVA and SIA/SSA) as the means of verification of the new implementation of UFEMISM. Arguments have been brought forward that exact solutions rather than intercomparison should be used to verify a code (Bueler, 2005). But even in case of the latter, I would ask for a more detailed analysis. To me it appears that for high-frequency disturbances in ISMIP-HOM Exp. A, the SIA/SSA as well as the DIVA approximation are significantly deviating from the ensemble (both the HO and even more the Stokes) and in case of the bedrock friction experiment (ISMIP-HOM C) somewhat (to me surprisingly) in the lower row of Fig. 4 the hydrostatic first order (Blatter-Pattyn) solution – despite the authors claiming that it is contained in the ensemble-range for all domain scales. In particular, with respect to conclusions of inaccuracies arising in both, the SIA/SSA and DIVA approximation at smaller disturbance length scale and to me also the Blatter-Pattyn solution showing deviations in Exp. C, I would ask to provide **a discussion on the expected accuracy and the acclaimed verification of these approximation applied to ice-sheet simulation, also beyond synthetic intercomparison setups**. For the marine ice-sheet examples (MISMIP and MISMIP+) I could not find the information what approximation to the Stokes equation has been used to compute the results. I am confused by the output in Figure 5 (MISMIP), where a timescale of 30 kyr = 3.0×10^2 yr is depicted. If this should resemble Exp 2 in Pattyn et al. (2012), I am missing several further step-wise increases. Also, the (here only two varying) values of the factor *A*, to me appear several orders of magnitude larger than those used in the MISMIP experiments. I do not know the reasons to deviate that much from the MISMIP protocol, but I would ask the authors to **provide a reason for those to me rather strong deviations from the original MISMIP experiment**, such that the reader can correctly interpret the meaning of "*an experiment along the lines of the Marine Ice-Sheet Intercomparison Project*" not to try to make an attempt to compare with the results reported in Pattyn et al. (2012**). Or - in view of the claimed code verification - run the MISMIP Exp 1 and Exp 2 according to the protocol,** such that the reader can get a clear picture on how UFEMISM v2.0 behaves in view of the MISMIP ensemble.

4 Missing information on thermo-mechanical coupled problems

All the investigated intercomparison setups focus on pure mechanics. Yet, a changing temperature field by the Arrhenius law has a significant impact on ice-dynamics (Schoof and Hewit, 2021). The manuscript is not mentioning the inclusion or even computation of heat transfer in a coupled

thermo-mechanical context at all in the text. **I would ask the authors to add a paragraph if and how temperature (or even damage) is accounted for or included in UFEMISM?**

Detailed list of requested changes or elaborations

The list of issues is in order of the text indicated by page and line-number of the submitted manuscript. Quotes from the manuscript are kept in blue text.

page 1 – line 20: The i/o now follows the NetCDF-4 standard, including automated remapping between regular grids and irregular meshes, reducing user workload for pre- and post-processing.
What exactly do the authors mean by NetCDF standard? A certain convention, like CF?
page 1 – line 23: … and which is ready to be used for coupling within earth system models.
What constitutes the readiness for inclusion in ESM's? Does the code have coupler interfaces for online coupling to atmospheric or ocean models (e.g., Gladstone et al., 2021) implemented? From the scalability figures given in the text, I see a problem to run the code on large supercomputers, which I see as a necessity for inclusion in ESM's. The in my opinion unclear restriction with respect to MPI implementation (OpenMPI) most likely constitute a hurdle to run on Tier 0 or Tier 1 HPC facilities. From what is presented in this paper, I would not derive a readiness of UFEMISM v2.0 to be incorporated into ESM's. See my argumentation under major points #1 and #2.
page 2 – line 34 Part of this large uncertainty stems from poorly constrained physical properties and processes in the Antarctic ice sheet system, including subglacial conditions (e.g. Kazmierczak et al., 2022; Berends et al., 2023a), basal sliding (Sun et al., 2020), interactions between the ice shelf and the ocean in the sub-shelf cavity (e.g. Burgard et al., 2022; Berends et al., 2023b), and ice-dynamical processes (e.g. Rückamp et al., 2022).
What about calving-induced instabilities, like MICI (e.g. Crawford et al., 2021) - or are those subsumed under ice-dynamical processes?
page 2 – line 46: Here, we present version 2.0 of the Utrecht Finite Volume Ice-Sheet Model (UFEMISM). Version 1.0 (Berends et al., 2021) was the second ice-sheet model to use a dynamic adaptive mesh, the first being BISICLES (Cornford et al., 2013).
May I point out that in the context of calving-front computations in Greenland, Todd et al. (2018) introduced a dynamic remeshing algorithm into *Elmer/Ice*. Additionally, in the first flowline marine ice-sheet full-Stokes experiments by Durand et al. (2009) adaptive meshing around the grounding line was already introduced 15 years ago in connection to the also in this manuscript discussed MISMIP setups. Furthermore, also *ISSM* includes mesh-adaptation, according to their web-site (https://issm.jpl.nasa.gov/documentation/mesh/).
page 3 – line 80: Earlier research in ice-sheet modelling has shown that the accuracy of a numerical model is particularly resolution of the grid around the grounding line (Gladstone et al., 2012; Pattyn et al., 2012)..
For flow-line problems, mesh sensitivity of grounding line positions – even in the context of full-Stokes - was described in even earlier works by Durand et al. (2009).
page 4 – line 104: The most complete is the Blatter-Pattyn approximation (BPA; Pattyn, 2003), which is described in Sect. 2.2.1.
"Most complete" in what sense? And I would add Blatter (1995) as a citation here.
page 5 – line 112: The BPA arises from the Stokes equations …
In my opinion, if the applied approximations to the Stokes equations are discussed in such details, it would be best to write out the complete Stokes equations to relate the approximations. But in

my opinion – as I also mentioned in the *General Impression* - a reference to standard literature (e.g. Greve & Blatter, 2009) or presentation of the equations from section 2.2 in an appendix would be sufficient and significantly shorten the article.

page 5 – line 114: The BPA produces ice velocities that are generally very close to those from the Stokes equations (Pattyn et al., 2008).

The wording "generally very close" in my view is hard to interpret and easy to misinterpret. If taken by its spatial extend of validity on large ice-sheets, one could claim that also SIA is "generally" very close to Stokes, but it completely fails under ice-domes and ridges and in fast flow regions and shelves. The ISMIP-HOM reference (Pattyn et al., 2008) is a set of idealized synthetic benchmark cases and in my opinion does not justify a statement that could be interpreted that first-order approximation is a sufficient substitute to the complete Stokes solution in every situation – which does not apply, in particular where variations in vertical advection are of essence, like in thermo-mechanically coupled problems of ice-streams (Schoof and Mantelli, 2021), advection problems of tracers (Jouvet et al., 2021) and flow at ridges and domes (Seddik et al., 2011).

page 5 – line: 117, … owing to the fact that, where those approximations either parameterise or neglect vertical variations in velocities or strain rates, the BPA solves for such variations explicitly.

To me it appears that this could be interpreted that BPA solves for all vertical variations of strain-rates. My suggestion: *… where those approximations parametrise or simply ignore the in BPA not neglected vertical derivatives of the horizontal velocity components*.

page 5 – line 118: This requires the model to discretise the vertical dimension as well, whereas the DIVA and the hybrid SIA/SSA operate in the two-dimensional plane, yielding a system of linear equations that is larger by a factor of the number of vertical layers in the model.

I would understand SIA to be solving column wise quadrature on a three-dimensional mesh (e.g., Greve and Blatter, 2009), so not being confined to a plane mesh.

page 5 – line 126; Equ. (2):

This links to major points of critics #4 that the authors seem to completely neglect the thermo-mechanical aspect of ice-sheet modelling, which in my opinion is of essence (Schoof and Hewit, 2013). In my opinion, one at least should mention that the rate factor, *A(T,p)* , is a function of the temperature and the pressure (and damage, if one wants to extend to that).

page 6 – line 132: The similar zero-stress boundary condition at the ice base includes a basal friction term.

To me this sentence is a contradiction: either there is a zero stress boundary or there is friction with a resulting tangential stress applied. Suggestion: *A similar dynamical boundary condition …*

page 6 – line 144: In the Ice-Sheet Model Intercomparison Project for Higher-Order Models (ISMIP-HOM; Pattyn et al., 2008) experiments, the DIVA produces velocities that agree well with the Stokes solution down to horizontal scales for basal topographical features of about 20 km (Berends et al., 2022; Robinson et al., 2022; this study, Sect. 4.1).

This links to my major point #3. As DIVA is introduced to be the default solver in UFEMISM, does that mean in a complementary conclusion that DIVA should not be deployed to mesh sizes below this threshold, as then accuracy is compromised? I would like to see some sort of deeper discussion in the with respect to the rest of the manuscript extremely brief section 5.

page 6 – line 154: The integral term F_2, which can be thought of as the depth-integral of the square of the inverse viscosity, is defined as:

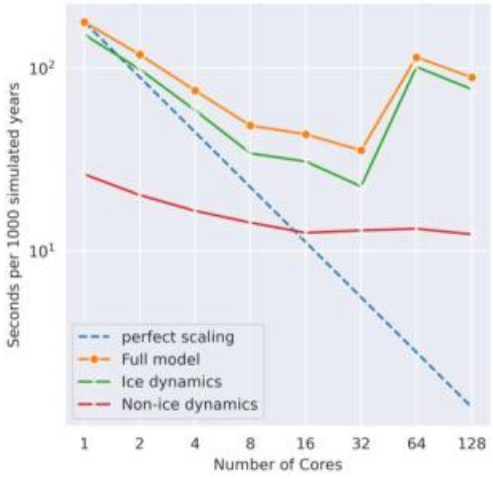
To me that does not come clear from (8) (i.e., there is no exponent *n=2* over the viscosity). Also, for consistency, the product $\beta \cdot F_2$ in (7) should be dimensionless, which to me does not work out if F_2 is proportional to the square of the inverse viscosity.

page 6 – line 156: Note that, in Eq. 7, n = 2; Eq. 8 lists the general form because elsewhere in the DIVA, F_1 appears as well.
The reader might ask themselves where that would be. Are there other equations entering the system? In view of a more complex derivation of the equations of motion, I would suggest to drop everything around eqts. (7) and (8) and directly refer to look things up in Lipscomb et al. (2019)
page 7 – line 178: Substituting Eq. 12 into Eq. 11, and assuming a stress-free boundary condition at the ice surface and a no-slip boundary condition at the ice base, leads to the following analytical solution for the vertical profile of the horizontal ice velocity:.
I would like to have the assumption of no-slip motivated. I do not even understand it in case of hybrid SIA/SSA, as to my understanding there the sliding velocity should be provided by the SSA solution (hence non-zero).
page 8 – line 187: … but starts to deviate significantly from the Stokes solution earlier than the DIVA as the length scale decreases (Berends et al., 2022; this study).
This links to major point of critics #3. I would see it necessary to quantify this in terms of grid sizes that can be addressed with hybrid SIA/SSA.
page 8 – line 205: Here, N is the effective pressure between the ice and the bedrock, which is equal to the ice overburden pressure minus the subglacial water pressure.
That leaves me (and perhaps some readers) with the question on how water pressure is determined in UFEMISM v2.0? Is there a sub-glacial hydrology model included (e.g., Gagliardini and Werder, 2018) to provide that variable?
page 9 – line 229: Therefore, in UFEMISM v2.0 the flux condition has been replaced by a sub-grid friction scaling scheme, following the approach used in PISM (Feldmann et al., 2014), CISM (Leguy et al., 2021), and IMAU-ICE (Berends et al., 2022).
May I point out that sub-grid friction parametrizations at grounding lines are wider spread in the ice-sheet model community. Also ISSM (Seroussi et al, 2014,) and Elmer/Ice (Gagliardini et al., 2016) deploy a sub-grid friction parametrization.
page 9 – line 237: Conservation of ice mass for a shallow layer of incompressible ice in the 2-D plane is expressed mathematically as:

$$\frac{\partial H}{\partial t} = -\nabla \cdot (\boldsymbol{u}H) + m$$

Equation (21) seems to be vertically integrated mass balance. Hence, $\boldsymbol{u}H$ being the vertically integrated, horizontally vector-valued, volume flux. Please, to inform the readers, add a definition of it to the text, also if/how the definition of this term differs between the available approximations to the Stokes equations.
page 10 – line 264: v2.0 uses the predictor/corrector (PC) time-stepping scheme by Robinson et al. (2020).
How does the predictor-corrector scheme link with the time-discretization schemes presented in section 2.5? I have the suspicion that the symbols Δt have different meanings in 2.5 and 2.6. Please, elaborate.
page 11 – line 191: … a shared-memory architecture, where all data is stored in the same memory chip which all processors can access ...
To me, this is gives an over-simplified picture. Even in a shared-memory machine/node, generally, it is not a single chip containing the data. And, not all processing cores (which I would use as a term rather than processors) can access a certain chip in a similar fast way (NUMA domains). A better formulation in my view would be: … *in shared memory architecture, where all parts of the memory are accessible via a common bus to all computing cores, in contrary to distributed memory*

architecture that demands communication between by memory separated computing nodes.
page 11 – line 293: The distributed-memory architecture is slower than a shared-memory program running on the same number of processors, as data frequently needs to be exchanged between the different processor.
I would understand architecture as the hardware, which cannot directly be compared to a program (software implementation). For the latter, it very much depends on how the code is implemented. It is correct that on a pure hardware-level distributed memory access across nodes is slower (how much depends also on the performance of the interconnect-network and the memory-layout of the shared memory node) than the one of shared memory, yet, even shared memory parallelism (talking again about software) obeys Amdahl's law (Amdahl, 1965) and performance mainly hangs on the serial sequences of the code (which generally exist). Suggestion: *Memory access within shared memory nodes outperforms message passing across shared-memory nodes.*
page 11 – line 298: Solving the matrix equation representing the momentum balance is currently the most computationally demanding part of the model by far, often accounting for more than 80 % of the total computation time of a simulation
To my understanding, not every approximation needs a matrix system to be solved, SIA does not. And the SSA/DIVA matrix must be significant smaller than the 1st-order system. I wonder: Does the number given apply to all discussed approximations? If not, I would ask to be more specific.
page 12 – line: 303: … UFEMISM offers a set of standardised routines that interface with the OpenMPI library (Gabriel et al., 2004) to facilitate this..
This links to #1 of major issues. Why the constraint to the OpenMPI flavour? Does this mean UFEMISM cannot compile with another MPI-standard library, like IntelMPI or vendor specific MPI implementation? If so, please explain why.
page 12 – line 306: This likely has to do with the way data communication between processes is handled by PETSc, which could be improved by paying more attention to the way the model domain is partitioned over the processes, and the way PETSc decides which data should be communicated.
This also links to #1 of major issues. From the manuscript, I do not get enough information to be in a position to understand how the MPI parallelism in UFEMISM is organized. It would be interesting to the HPC inclined reader to learn how partitioning is done and - in particular with respect to the remeshing - the load balancing is guaranteed. To my knowledge, PETSc is well tuned to perform on multi-node clusters - what in particular do the authors suggest to be changed therein?
page 12 – line 308: However, it should be noted that v2.0 in its current form is capable of performing multi-millennial simulations of the Antarctic ice sheet, using a grounding-line resolution of < 5 km across selected basin-scale regions, on a dual-core, consumer-grade laptop (Bernales et al., in prep.). Large-scale practical applications of the model are therefore already feasible even without these future improvements.
This links to #2 of major issues. Firstly, I am missing the information what approximation applied allows one to run “basin-scale” (not sure what it means in terms of grid-sizes and spatial dimensions) on a laptop. Secondly, simulations on a laptop to me have a remote relevance to parallel performance/scaling on large machines, particular on distributed memory setups, which I understand this chapter to be about if the authors refer to “Large-scale practical applications”.
page 12 – line Figure 2:



This figure is the main reason for point #2 in the list of major critics. This graph, in my opinion, needs way more explanation and discussion – also in the text, not only the caption. Like in other parts in the text, it is missing information on the approximation to the Stokes equations that is being studied here. Secondly, the informed reader might want to know on which computational platform this was run on and if we look at a single- or multiple node run. I already mentioned that a flow-line model in my view is a non-representative example for scalability if one wants to get a picture on how the code would perform and scale being applied to full ice-sheet problems. Yet, this seems to be the only place in the manuscript where the reader can get an idea on a performance baseline in terms of solved time-steps/wall clock time. I already suggested to do scalability tests with MISMIP+ if not on the full Antarctic setup. Furthermore, I would like to learn more on how the authors determine and separate ice dynamics and non-ice-dynamics parts in this figure. From a pure computational science point of view, I interpret the fact that a run on 64 and even 128 cores consumes a comparable wall-clock time as a 2 core run points to a serious issue in the parallel implementation that in my opinion prohibits production runs on compute clusters.

page 12 – line 322: … detect the type of grid from the dimensions of the NetCDF file …

Do the authors mean that there is some automatic parsing of the meta-data of the NetCDF file (CF convention?) that deduces the coordinate system? Further question: is UGRID format meant when referring to triangular meshes?

page 14 – line 365 In experiment C (Fig. 4), which concerns sliding over a bed with spatially varying roughness, all three approximations result in velocities that agree well with the ensemble.

To me, in the lower row in Fig 4 displays the 1st order results (not tremendously, yet visible) surface velocities to exceed these of the ensemble. I would ask to explain why this is the case and – since the authors do not seem to raise any concern in the text – why it can be neglected.

page 15 – line 273: The experiment describes a circular, cone-shaped island, subjected to a spatially uniform positive mass balance.

I could not find any mentioning in Pattyn et al. (2012) of a lateral circular symmetry to apply to MISMIP flowline setups.

page 15 – line 381: We performed simulations with grounding-line resolutions of 10, 8, 5, and 4 km.

Like in the caption of Figure 5 on page 16, the information on what approximation to the Stokes-equation has been used for this resolution-experiments is missing.

page 15 – line 381: We start with a 10,000-yr spin-up phase, with a uniform flow factor of A = 10−16Pa−3yr−1. We then decrease the flow factor to A = 10−17Pa−3yr−1 for a period of 10,000 years, resulting in an advance of the grounding line by about 200 km.

As mentioned in the major points of critics #3, I would ask the authors to relate parameter choice and the reduced time-span of the experiments to the original MISMIP protocol and explain – also in light of the argument of verification – this deviation.

page 16 – line 405: We have performed MISMIP+ experiment “ice1r” (100 years of increased-melt forcing) with UFEMISM v2.0, using the Schoof sliding law (Eq. 20) …

Like in the MISMIP chapter, information on the applied approximation to run the MISMIP+ experiments seems to be missing, also in the caption of Fig. 5. Please add this information.

page 17 – line 417: We have presented version 2.0 of UFEMISM and verified its performance in a number of benchmark experiments with idealised geometries.

If this is about computational performance, I have to disagree. From this paper I am lacking information to really judge the computational performance of the code. Deducing from Fig. 2, I would even conclude that there is an unresolved issue what comes to scalability of the code. If it is about code verification, I previously mentioned that the MISMIP tests to me do not provide the means to deliver on that aspect, as they deviate from the original protocol which prohibits comparison, which leaves ISMIP-HOM (with some approximation showing strong deviations) and MISMIP+, where I could not deduce what approximation has been used for intercomparison with ensemble results.

page 17 – line 423: The numerical stability and computational performance of the model have been greatly improved. This includes the new time-stepping scheme, as well as the switch from a simple successive over-relaxation scheme to PETSc for solving the matrix equations. As a result, v2.0 is much faster than v1.0, capable of either running the same simulations in a fraction of the required computation time, or running a simulation in the same amount of time as before, but at a much higher resolution.

As mentioned earlier (point #2 of major points), in my view this paper is lacking information to really judge about performance of the code, as the reader is not provided with a baseline value. In my opinion, wording like "much faster" are not conveying enough information to the reader that would enable quantification of the code's performance. To really judge about performance, the reader would need to get an idea on a performance-baseline. For instance, how much the solution of one ISMIP 6 scenario run for Antarctic ice sheet using a particular approximation (preferably the optimal one, hence BPN) needs wall clock time on one, two or even more nodes of a computing cluster. From figure 2 I would draw the conclusion that the code does not scale beyond 8 cores (of whatever platform it was run on).

page 17 – line 431: However, solving the BPA can easily require 50 times more computation time than solving the DIVA, which would be unfeasible for many practical applications.

As I mentioned before, statements like this to me are impossible to evaluate without providing a baseline value. Just the fact that some algorithm takes 50 times longer (presumably using the same amount of computational resources) in my view does not imply that it is impossible to solve it – in particular if the code is claimed to run parallel (should in theory be able to use 50 times more resources, provided it scales) and comes with mesh-adaptation scheme.

page 18 – line 435: In preparation for such an approach, the code of UFEMISM's routines for solving the ice thickness equation has been written in such a way as to easily allow the user to define regions where the ice thickness should not change.

I wonder, should there be larger thickness changes in the active region, how does the model deal with the artificially imposed hydrostatic pressure gradients at boundaries with a one-sided fixed ice-thickness?

page 18 – line : While the model is already capable of performing high-resolution (< 5 km), multi-millennial simulations of the Antarctic ice sheet (Bernales et al., in prep.), moving to even higher resolutions would currently still require the user to wait for several days for the simulation to complete.

To me this again lacks information to really get a clear picture on what type of simulations with what approximation and what accuracy can be achieved with what computational resources. There is only information on minimum resolution, but not the approximation used. Multi-day

simulations are to me nothing that renders a computational problem impossible. But to me this sentence backs my suggestion that a fusion of the two papers would be beneficial to the reader to pick information that I conclude exists in the other manuscript (Bernales et al., in prep) to be able to relate statements presented in this one.

page 19 – line Figure A1:

Looking at the distances of the polygon-points at the zoom-in over Ronne-Filchner ice-shelf, I conclude that this is far away from the acclaimed 5 km resolution. If this is really the resolution to start from, I would ask the authors to include a sentence on how the accuracy of the coastline is increased when increasing the mesh density: is it just linearly interpolated between existing points or is additional geometrical information added to the polygon?

page 20 – Figure A2:

This is a large figure. I would try to simplify the composition resulting now into entry e) and thereby reduce (entries a – e) the size of the figure. Instead, what I would welcome to see included is a visual demonstration on how the remeshing algorithm could enhance mesh densities in areas of large derivatives of the velocity field, which turned out to be essential to resolve (thermo-)dynamics of fast outlet glaciers (e.g., Zhao et al. 2018) in other applications.

page 21 – Appendix B:

This whole part appears abstract to me and without looking things up in Syrakos et al. (2017), difficult to understand. I would even suggest that, equally, a simple reference to the paper above and removing whole part B would shorten the manuscript. If the authors want to keep it, in my opinion it would help to have some figure of local grid configurations annotated with the node-indices and showing the most important features, like distances ($\Delta x_j, \Delta y_j$) and value entries (f_a^j) to better help illustrating the formulation of the stencils as presented in this appendix, also, with respect to the different discussed mesh-types and the coordination numbers of nodes/variables therein.

page 21 – Eqs B9 and B10:

In my view, it would be beneficial to the reader to explain what the third case "otherwise" means. With respect to definition in the text that indices $j \in [1, n]$ indicate all neighbours of *i*, I must miss something by interpreting that neighbours are connected by definition and "otherwise" being irrelevant for A-grids. Like mentioned before, some graphical display of a local mesh arrangement in my opinion could aid the understanding and prevent misinterpretations.

page 24 – Appendix C:

As remeshing/-mapping seems to be one of the main new features of UFEMISM v2.0, I would suggest to move this part (at least the non-mathematical) into the main section of the paper and rather take out other parts from there (I already suggested earlier).

page 25 – line 561: Let there exist two meshes that both cover the same domain Ω: a source mesh (indicated from here by the subscript s) and a destination mesh (subscript d). Suppose the source mesh is the one that existed before a mesh update, and the destination mesh is the newly generated mesh.

Like before, I find this section relatively abstract and difficult to read. Similar as before, I would be of the opinion that some graphics on the mesh-to-mesh projections showing the domain, its boundary and the two (source and destination) meshes to get a picture what this is about.

page 25 – line 566: where A_d^i are the Voronoi cells of the vertices of the destination mesh

Some readers that have not dealt with dual graphs might not know what a Voronoi cell is. Displaying this in a graph (see point above) and defining it in the text, in my view, would improve the readability of this chapter.

List of less important issues

Please, find here a list of minor things I only suggest to be changed.

page 2 – line 55: This includes a change from a shared-memory to a distributed-memory architecture (Sect. 3.1), ...
I guess the authors want to express that the code can now be run on distributed memory architecture (referring to architecture as being the hardware, rather than the program itself)? If referring to code, I would suggest to change to: *… from a shared- to a distributed memory implementation*.
page 4 – line 91: Some models solve this problem by using a mesh with a high resolution over a wider area, ...
In my opinion, a statement like this would demand references to be included.
page 6 – line 141: The DIVA, which is the default option for the momentum balance approximation in v2.0, arises by neglecting vertical variations in the membrane stresses in the BPA (i.e. $\frac{\partial}{\partial z}\left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}\right) \approx 0$) …
The brackets show the vertical derivatives of the components constituting the horizontal strain-rates (and not membrane stresses).
page 11 – line 297: processors - I would use *cores*
page 13 – line 344: These experiments describe a slab of ice on a flat, sloping bed.
One group of experiments does include flat beds (C,D,F), the other (A,B - as correctly pointed out in the sentence to follow) compute on an undulated bed and Experiment E on a glacier flowline (which is not flat, either).
page 15 – line 388: Note that all these simulations were performed with the dynamic adaptive mesh; whereas in v1.0, a mesh update would result in a small but noticeable "jump" in the grounding-line position (Berends et al., 2021, their Fig. 390 10b), improvements to the remapping scheme in v2.0 have greatly reduced this problem..
Since remapping is mentioned here, I would suggest to make a reference to Appendix C for the reader's convenience to quickly look things up.
page 19 – line 488: Here, Mx,a,a is an *nV*-by-*nV* matrix.
nV at this stage appears to me as being undefined (guess, it relates to some correlation number)
page 21 – line 505, eq. B5; page 23 eq. B13; page 24, eq B20:
In all equations the lower symmetry entries in the matrix are omitted. Despite the redundancy, in my view it is preferable to either spell things out in the equation or mention the unusual notation in the text.
page 28 – line 665: Just a suggestions, but one could elegantly use the Kronecker delta to define $D^{ij} = \delta_i^j \, f^i$ to achieve a better typesetting result in that line.

References

Amdahl, G (1967): *Validity of the single processor approach to achieving large scale computing capabilities*, In AFIPS '67 (Spring): Proceedings of the April 18-20, 1967, spring joint computer conference April 1967, pp 483–485 doi:10.1145/1465482.1465560

Blatter, H. (1995): *Velocity and stress fields in grounded glaciers: a simple algorithm for including deviatoric stress gradients*. Journal of Glaciology, 41 (138), 333–344

Bueler E, Lingle CS, Kallen-Brown JA, Covey DN, Bowman LN (2005): *Exact solutions and verification of numerical models for isothermal ice sheets*. Journal of Glaciology, 51(173):291-306. doi:10.3189/172756505781829449

Byckling, M., J. Kataja, M. Klemm, and T. Zwinger (2017): *OpenMP SIMD Vectorization and Threading of the Elmer Finite Element Software*, In: de Supinski B., Olivier S., Terboven C., Chapman B., Müller M. (eds) Scaling OpenMP for Exascale Performance and Portability. IWOMP 2017. Lecture Notes in Computer Science, vol 10468. Springer, doi: 10.1007/978-3-319-65578-9_9

Crawford, A.J., Benn, D.I., Todd, J, Åström, J.A, Bassis, J.N. and Zwinger, T. (2021): *Marine ice-cliff instability modeling shows mixed-mode ice-cliff failure and yields calving rate parameterization.* Nat Commun 12, 2701, doi:10.1038/s41467-021-23070-7.

Durand G., O. Gagliardini, T. Zwinger, E. Le Meur and R.C.A. Hindmarsh (2009): *Full Stokes modeling of marine ice sheets: influence of the grid size*, Ann. Glaciol., 50(52), 109–114.

Gagliardini, O., Brondex, J., Gillet-Chaulet, F., Tavard, L., Peyaud, V., and Durand, G. (2016): *Brief communication: Impact of mesh resolution for MISMIP and MISMIP3d experiments using Elmer/Ice*, The Cryosphere, 10, 307–312, doi:10.5194/tc-10-307-2016, 2016.

Gagliardini O., and Werder, M. (2018): *Influence of increasing surface melt over decadal timescales on land-terminating Greenland-type outlet glaciers*, Journal of Glaciology, 64(247), 700-710, doi:10.1017/jog.2018.59

Gladstone, R., Galton-Fenzi, B., Gwyther, D., Zhou, Q., Hattermann, T., Zhao, C., Jong, L., Xia, Y., Guo, X., Petrakopoulos, K., Zwinger, T., Shapero, D., and Moore, J. (2021): *The Framework For Ice Sheet–Ocean Coupling (FISOC) V1.1.* Geosci. Model Dev., 14, 889–905. doi:10.5194/gmd-14-889-2021

Greve, R. and Blatter, H (2009): *Dynamics of Ice Sheets and Glaciers*, Springer, Berlin, Heidelberg doi:10.1007/978-3-642-03415-2

Jouvet, G., Röllin, S., Sahli, H., Corcho, J., Gnägi, L., Compagno, L., Sidler, D., Schwikowski, M., Bauder, A., and Funk, M. (2021): *Mapping the age of ice of Gauligletscher combining surface radionuclide contamination and ice flow modelling*, The Cryosphere, 14, 4233–4251, doi:10.5194/tc-14-4233-2020, 2020.

Pattyn, F., Schoof, C., Perichon, L., Hindmarsh, R. C. A., Bueler, E., de Fleurian, B., Durand, G., Gagliardini, O., Gladstone, R. M., Goldberg, D., Gudmunsson, G. H., Huybrechts, P., Lee, V., Nick, F. M., Payne, A. J., Pollard, D., Rybak, O., Saito, F., and Vieli, A. (2012): *Results of the Marine Ice Sheet Model Intercomparison Project, MISMIP*, The Cryosphere 6, 573-588, doi:10.5194/tc-6-573-2012

Seddik H., R. Greve, T. Zwinger and L. Placidi (2011)**:** *A full-Stokes ice flow model for the vicinity of Dome Fuji, Antarctica, with induced anisotropy and fabric evolution*, The Cryosphere, 5, 495-508, doi:10.5194/tc-5-495-2011

Schoof, C, and Hewit, I. (2013): *Ice-Sheet Dynamics*, Ann. Rev. of Fluid Mech., 45: 217-239, doi: 10.1146/annurev-fluid-011212-140632

Schoof, C., and Mantelli, E. (2021): *The role of sliding in ice stream formation*. Proc. R. Soc. A 477: 20200870, doi:10.1098/rspa.2020.0870

Seroussi, H., Morlighem, M., Larour, E., Rignot, E., and Khazendar, A. (2014): *Hydrostatic grounding line parameterization in ice sheet models*, The Cryosphere, 8, 2075–2087, doi:10.5194/tc-8-2075-2014

Syrakos, A., Varchanis, S., Dimakopoulos, Y., Goulas, A., and Tsamopoulos, J. (2017): A critical analysis of some popular methods for the discretisation of the gradient operator in finite volume methods, Physics of Fluids 29, 127103, doi:10.1063/1.4997682

Todd, J., P. Christoffersen, T. Zwinger, P. Råback, N. Chauché, D. Benn, A. Luckman, J. Ryan, N. Toberg, D. Slater, and A. Hubbard (2018): *A Full-Stokes 3D Calving Model applied to a large Greenlandic Glacier*. Journal of Geophysical Research: Earth Surface, 123, 410–432, doi:10.1002/2017JF004349

Zhao, C., Gladstone, R. M., Warner, R. C., King, M. A., Zwinger, T., and Morlighem, M. (2018): *Basal friction of Fleming Glacier, Antarctica – Part 1*: *Sensitivity of inversion to temperature and bedrock uncertainty*, The Cryosphere, 12, 2637–2652, doi:10.5194/tc-12-2637-2018