We thank the reviewer for their constructive criticism of our manuscript, and would hereby like to respond to their concerns. Their comments are shown in italics, our response in regular type.

General assessment

> *I would encourage the authors to better motivate the acceptance level with respect to the Stokes or higher-order ensemble results (why is 20km still accepted, but 10km is not?)*
>
> *L145: "DIVA produces velocities that agree well with the Stokes solution down to horizontal scales for basal topographical features of about 20 km" How can "agree well" be quantified? In the mentioned references, the solution for 20km is not within the higher order or Stokes ensemble.*
>
> *L368: "The DIVA remains accurate to spatial scales of about 20 km,..."*
> *Gain, please define "accurate". What deviation from the FS ensemble is accepted? Why is the 20km still accepted but 10km not?*

The relative errors that are deemed acceptable are, ultimately, a subjective judgment. In ISMIP-HOM Experiment A with L = 20 km, the DIVA overestimates the surface velocity of the fast-flowing ice by about 25 %. In the experiment with L = 10 km, this number increases to about 40 %. Based on the inter-model spread of intercomparison exercises such as ISMIP6, and also considering the fact that ISMIP-HOM presents an extreme case (with quite dramatic subglacial topography), we found a 25 % deviation to be large but workable. Of course, so long as running simulations with the BPA is not computationally feasible, the DIVA remains the lesser of two evils, providing the least inaccurate solution (compared to the hybrid SIA/SSA, and to several other depth-integrated approximations, according to Robinson et al. (2021). We will add a few lines to the manuscript to reflect these thoughts.

> *However, DIVA can be thought of a modified SSA approximation (considering membrane stresses in the plane), in which the effective viscosity is treated in a different way (also accounting for vertical velocity gradients). I suspect that the non-linearly diffusive component of the ice sheet flow (e.g. Bueler et al., 2007), which is relevant in many (purely shear-stress-driven) parts of the ice sheets, may be underrepresented in DIVA.*
>
> *L164: Maybe it is worth mentioning the differences and similarity of SSA to DIVA.*
>
> *L186: "but starts to deviate significantly from the Stokes solution earlier than the DIVA as the length scale decreases (Berends et al., 2022; this study).*
> *"In the ISMIP-HOM experiment". Generally SIA describes a (non-linear) diffusive process that is characteristic for large parts of the ice sheet flow and not represented in SSA, and likely not (or only limited) in DIVA. Please refer to the respective section in this study.*

It can be shown that, in the absence of horizontal strain rates (so pure shear flow over a flat plane), the DIVA is identical to the SIA. We have at some point run the EISMINT-1 experiments (an idealised, roughly Greenland-sized ice sheet lying on a flat plane, achieving steady state through a simple spatially variable mass balance) with the DIVA instead of the SIA; the resulting ice sheet is nearly identical to that resulting from the SIA model (a few meters thickness difference near the ice divide and near the margin). ISMIP-HOM Experiment A has a no-slip condition at the base and so too is dominated by vertical shear, and there too the DIVA performs well. We will clarify this in the description of the DIVA in Sect. 2.2.2.

Regarding the SSA: the similarity between the linearised equations (Eq. 6 and Eq. 9) is actually a great benefit in practice, as it makes it very easy to adapt SSA models to be able to solve the DIVA. We will mention this in the manuscript.

> *Maybe, benefits of DIVA over SIA/SSA will become clearer in real world (or regional) application, as planned for the mentioned follow-up study.*

We have no such experiments planned for the Part II paper, which will instead focus mostly on model initialisation, nudging of basal friction, melt, and calving, and on comparing to Greenland & Antarctic retreat model ensembles (e.g. ISMIP6).

> *I am also missing in the description how (lateral/marginal) boundary conditions have been treated.*
>
> *L135ff: What about the lateral boundaries (margins, calving fronts)?*

UFEMISM uses the “infinite slab” approach, where the momentum balance is solved even in the ice-free part of the domain (by assigning a small – 10 cm – ice thickness to those cells), and applying a simple Neumann boundary condition to the domain boundary. We will mention this in the manuscript.

> *Apart from the DIVA solver, some model components (e.g. thermodynamics validated with EISMINT) have already been described in the v1.0 paper by Berends et al., 2021. It would be good to mention more clearly what has not changed since v1.0.*

The list of things that have not changed is quite extensive, but consists largely of model components that are not used in the idealised experiments described here (surface mass balance, climate forcing, GIA, etc.), and which were also not described in the v1.0 paper (as they were mostly adapted from earlier models). We agree that thermodynamics is an important one since it is so closely related to the ice dynamics itself, so we will mention it in the manuscript.

> *Also in the v1.0 paper, the same MISMIP-inspired experiment has been performed, but with a SSA/SIA hybrid stress balance, flux correction, an explicit first-order finite volume upwind scheme for the ice thickness evolution and much coarser resolution. Readers may*

be interested in a more quantitative comparison of the new model version compared to the older version, or to other similar models (e.g. flow-line MISMIP or MISMIP2d).

A direct comparison to v1.0 is difficult because of the many changes in the model. The change from the flux condition to the sub-grid friction scaling, the change in time stepping, and the change in resolution, all can be viewed as "mathematical", but there have also been some "physical" changes. One in particular (which was perhaps not immediately obvious from the manuscript, so we will state this explicitly in the revised version) is that v1.0 solved a simplified version of the SSA, where the gradients of the effective viscosity were neglected. Back then, we based this on (much) earlier work, but we later performed some experiments that indicated that this approach, while making the model substantially more numerically stable and thereby faster, introduced significant errors in the velocity solution, particularly in geometries with migrating grounding lines. We have therefore removed this simplification in v2.0 (which necessitated the change from defining the velocities on the grid edges, to the triangle centres, in order to achieve a numerically stable solution).

Since v1.0 has not been used in any practical applications yet (as our own focus has been more on developing v2.0), and there are therefore no earlier results that can be retroactively assessed, we do not think there is much added value in such a detailed comparison.

We also do not think performing the flowline MISMIP or MISMIP2d experiments would be of much added value. The two main model properties that these experiments investigate, namely (the lack of) path-dependency in the grounding-line position and the rate of grounding-line advance/retreat, are already assessed in the experiments we already did (path-dependency in the modified, plan-view MISMIP experiments, and retreat rate in MISMIP+).

The authors also mention that they switched for the discretization from neighbor functions to a least squares-based scheme. The v1.0 paper already uses an averaged-gradient (numerical stencils) approach similar to an unweighted least-squares approach (Syrakos et al., 2017). Is this the same as in v1.0?

In the v1.0 paper, we compared the numerical convergence of the neighbour functions approach (which could indeed be described as an averaged-gradient approach, and which is what was implemented in v1.0) to that of the least-squares approach by Syrakos et al. (2017), finding that they produced very similar results. Because of the change in the SSA/DIVA discretisation mentioned before (no longer neglecting gradients in the effective viscosity), and the subsequent change in definition of the ice velocities (triangles instead of edges), v2.0 makes a lot more use of staggering than did v1.0. As the least-squares approach can be more easily generalised to staggered grids (and also has a much more elegant derivation), we opted to remove the neighbour functions approach entirely and use the least-squares approach instead. We will clarify this in the manuscript.

I tried to install UFEMISM (without nix), but failed for the PETSc dependency, while searching for help in an installation manual or README.

We are aware that it can be tricky getting the different libraries and compilers to work on different platforms. Unfortunately, since our funding for IT support from the eScience Centre (in the person of co-author V. Azizi) has expired, we do not foresee significant improvements in this regard in the near future.

Detailed comments

> *L10: "... irreducible uncertainty in many of these processes..." Not sure what this means.*

For many of these physical processes, there is a limit to how accurately we can predict their future behaviour, based on currently available observations. Typically, this limit is determined experimentally by sampling the phase space of possible values of the parameters governing these processes, which is where the models we mentioned appear. We agree that describing this uncertainty as "irreducible" might be confusing; we will remove this word.

> *L43: "...have recently directed their efforts at creating new, more powerful ice-sheet models (e.g. Pattyn, 2017; Hoffman et al., 2018; Quiquet et al., 2018; Lipscomb et al., 2019; Robinson et al., 2020; Berends et al., 2022)" I am not sure if all models on this list had the intention to become "more powerful", as some are used for improving process understanding etc.*

We agree this was probably not their only intention, but we have deliberately referenced models that have included high-performance computing and/or user-friendliness in their model design, so we believe this phrasing to be appropriate here.

> *L47: "Version 1.0 (Berends et al., 2021) was the second ice-sheet model to use a dynamic adaptive mesh..." There should be previous experience with AMR, e.g. dos Santos, 2019 for ISSM or Gladstone et al., 2010a, just to give two examples. Better rephrase or provide a complete list.*

We will add references to Todd et al. (2018) for Elmer/ice, Durand et al. (2009), Gladstone et al. (2010), and to dos Santos et al. (2019, for ISSM) to the manuscript, and change the phrasing accordingly.

> *L62: "Part 2, which is submitted for review and publication separately (Bernales et al, in prep.)..." It would have been great to had both as companion.*

We agree this would have been great. Unfortunately, the manuscript of Part II has been delayed, as two of the authors have had to switch contracts in the meantime (a consequence of the failing system of scientific funding in the Western world).

> *L98: "...with no significant loss of accuracy"*

Hard to find quantitative numbers here or in Berends et al. (2021). If this remapping step is performed regularly also a small information loss (diffusion) can accumulate.

In the v1.0 manuscript, this was studied in the Halfar and Bueler dome experiments, where we could compare to an analytical, time-dependent solution. Both these experiments involved dozens of remeshes (over a hundred in the higher-resolution cases), and the solution was not visibly affected. In the current manuscript, probably the most solid evidence comes from the MISMIP+ experiment. In the 500 m resolution experiment, the mesh was updated about a hundred times, and yet the solution stays well within the model ensemble. We suspect that, since the flow of ice generally already has a substantial diffusive term, the small amount of numerical diffusion added by the remapping does not have a large effect (although in the MISMIP+ experiment, it might be possible that the solution is somewhat affected, which could explain why the grounding-line positions in the 1 km, 750 m and 500 m experiments differ more than expected).
We will add a few lines to the manuscript to reflect these thoughts.

L228: "... flux condition has been replaced by a sub-grid friction scaling scheme, following the approach used in PISM (Feldmann et al., 2014),".
Is this really a "replacement", as both methods may have different effects in controlling grounding line flux? I think the linear GL interpolation goes back to Gladstone et al., 2010b, and then there are 2D extension (bi-linear), e.g. Feldmann et al., 2014. Would it be possible to align the mesh with the grounding line, such that meshes are either grounded or floating/icefree?

We will add a few lines to briefly state the (dis)advantages of the two approaches.

While aligning the mesh with the grounding line is technically possible, it would imply remeshing the model in every time step. Not only would this be computationally very expensive, but in that case, our earlier claim that the numerical diffusion resulting from the remapping is not a problem, would likely not hold anymore.

L247: " an implicit scheme"
The is an interesting paper which may be cited as well (Bueler, 2023). Is the (semi-) implicit scheme still mass conservative (as it should be as a natural property of finite volume models)?

This is indeed a very interesting paper, we thank the reviewer for pointing it out and will definitely cite it.

The implicit scheme is mass-conserving within machine precision. The semi-implicit scheme introduces errors that are larger than can be explained by machine precision, although still very small (if I recall, around 10^-12 relative error).

L284: Eq. 33: I understand the notation is consistent with Robinson et al. 2022 and Cheng et al. 2017, but τ is often associated with stresses or time scales. Better use another epsilon consistent with Eq. 34. From Eq. 33 it seems that τ has dimensions of m/s. How does this fit with the default value for ε =3 m in L287? I would have expected it to be dimensionless.

We understand the possible confusion, but we much prefer to keep the notation consistent with the existing literature.

We had not noticed the mismatching dimensions before. Neither Cheng et al. (2017) nor Robinson et al. (2020) mention the units of the tolerance epsilon. As tau should indeed have units of m/s (or m/yr in UFEMISM), this suggests epsilon should too (which actually seems more plausible than simply m, as m/yr implies a tolerance in the thinning rate). We will mention this in the manuscript.

L293ff: "memory chips" and "processors"
For HPC architectures terms like "multi-core CPU nodes" are used, and the standard is rather 2x64 cores per node.

L299: "32 processors"
Do you mean CPU cores here, as mentioned in the Fig. 2 caption?

We will change "processors" to "cores" throughout the manuscript and clarify the statement regarding the typical number of processors per node.

L299: "...often accounting for more than 80 % of the total computation time of a simulation."
This very much depends on the application and the size of the computational domain. For instance, I/O can be a bottleneck for high-resolution application.

In UFEMISM, even in relatively low-resolution simulations, solving the momentum balance accounts for > 80 % of computation time, even when using the DIVA (which is the fastest to solve). I/O has never been a significant contributor in our experiments.

L307: "..., 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."
In deed, domain decomposition as well as matrix factorization and preconditioning have quite some potential for performance speed-up. Berends et al., 2021 describes in v1.0 already a load- balanced processor domain decomposition, which would be worth the refer to.

We suspect the scaling problems lies with the way we have implemented PETSc within UFEMISM. Mesh partitioning is currently done the same way as in v1.0. Where we suspect the

current performance problem lies, is with how PETSc determines (or is told) the connectivity between the vertices, and by implication, the non-zero structure of the sparse matrices it must work with. In the current implementation, in every non-linear viscosity iteration (where the sparse matrix equation must be solved), PETSc is (re)initialized, the sparse matrices are constructed, the resulting matrix equation is solved, and PETSc is finalised. While our own time measurements show that it is the solving step that accounts for 99% of the computation time (of these combined steps), we still suspect that somehow storing the non-zero structure of the sparse matrices (which remains unchanged until the next mesh update) could help with performance. The reason we suspect this is because UFEMISM v1.0 was much faster in this regard, and the only significant change in this particular part of the code is the change from shared memory to distributed memory.
We will reflect these thoughts in the Discussion section of the manuscript.

> *L322: "appropriate remapping function"*
> *You could already refer to the kind of remapping (bilinear, conservative) in Sect. ?. What input files not covering all of the computational domain, are there missing values or extrapolation applied?*

"Appropriate" here means the function appropriate for the type of source grid (x/y, lon/lat, unstructured), not the kind of remapping. Currently, nearest-neighbour extrapolation is used for input files not covering the entire model domain (although routines for assigning a user-defined missing value or doing e.g. linear or Gaussian extrapolation exist and should be easy to integrate here. We will mention this in the manuscript.

> *L329: "full list of the 100+ fields.."*
> *Does fields" imply that only 2D variables are available as diagnostic output? I guess for the current size of the applications, parallel I/O is not yet considered?*

3-D fields of e.g. velocity, temperature, and (effective) viscosity are available as well. Additionally, UFEMISM generates a NetCDF output file with domain-integrated values of e.g. mass balance components, ice volume, etc. All of this is indeed done serially, as we have not yet encountered any significant computational load from this. We will mention this in the manuscript.

> *L338: "The UFEMISM Github repository also features integration with the nix package manager..."*
> *Does also imply also other options? I found an EasyBuild example in the "templates" folder in the Git repo. Is there a general installation manual or a website with instructions? What open source license is used, I found in the "CITATIONS.cff" the entry for "Apache-2.0"?*

The nix package manager option was added as a convenience to have a reproducible build of UFEMISM2.0 and a canonical way to compile the code. However, since the nix package manager is not widely used, the main way to install UFEMISM2.0 is by using make. UFEMISM2.0 has

minimal and easily installable external library dependencies (petsc, netcdf). We will add this information in the README of the GitHub repository and also update the CITATIONS.cff with the correct license.

> *L365: "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."*
> *Please define "agree well", what is the acceptance range with respect to the FS ensemble? SIA/SSA seems to perform better than DIVA and BPA , why?*

We will change this phrasing to clarify that all results lie within the ensemble range except the BPA, which overestimates the full Stokes solution by up to 13 %.

We have spent a considerable amount of time looking into the difference between UFEMISM's BPA results, and that of the Pattyn et al. (2008) higher-order ensemble. While we cannot be sure, we think there is the possibility that UFEMISM's solution slightly more accurate than that of the ensemble. We find that, in the small-wavelength versions of Experiment C, the non-linear viscosity iteration converges extremely slowly compared to the other experiments, particularly when using the BPA (the SIA/SSA and the DIVA show the same behaviour, but to a lesser extent). Using a tolerance of e.g. a difference between subsequent velocity iterations of 1e-7 m/yr (which is very small; for most other experiments, values of about 1e-4 m/yr are sufficient) still results in a significantly different solution from using an even smaller value of 1e-9 m/yr. If the other models in the ensemble used a stop criterion for this iteration that was not strict enough, it is possible that their modelled velocities are too low, as the non-linear iteration terminated before having converged. This could also explain why the full Stokes ensemble shows higher velocities than the higher-order ensemble (as, according to Pattyn et al., the full Stokes models generally use pre-existing, general-purpose finite element packages, while the higher-order models more often used code written by the researchers themselves). As the Pattyn et al. ensemble was published one-and-a-half decades ago, there is no convenient way to check this. Other than this, the only plausible explanation is a coding error on our side – which is, of course, always possible.
We will leave it up to the editor if this consideration should be written down in the manuscript.

> *L372: "...performed an experiment along the lines of the Marine Ice-Sheet Intercomparison Project (MISMIP; Pattyn et al., 2012). The experiment describes a circular, cone-shaped island, ..."*
> *Please, better motivate how the experiment deviated from the original MISMIP (and from Berends et al., 2021, Sect. 3.3) I assume that it also covers two-dimensional aspects of the stress balance and geometry evolution. However, this prohibits a comparison to the semi-analytical solution provided for the flowline SSA case. The authors should indicate that they only considered the experiment with downward-sloping bed (EXP1+2, without overdeepening EXP3).*

The reason we have opted to extrude the 1-D geometry radially, rather than transforming the original 1-D flowline into a 2-D flowband, is that, while this means the resulting grounding-line position no longer matches the (semi-)analytical solution provided by Pattyn et al. (2012), it offers the advantage of checking the full 2-D stress balance (instead of only the x-component). This particularly allows us to check the symmetry of the grounding line. A well-known (but, as far as we are aware, never published) issue with flux condition schemes in square-grid models is the "octagonal" grounding line. A similar undesirable dependency on the grid geometry could sometimes be seen in UFEMISM v1.0 (which used a flux condition scheme), but has since been fixed with the introduction of the sub-grid friction scaling scheme in v2.0. We will mention this in the manuscript.

> *L378: "grounding-line hysteresis"*
> *In order to not confuse the reader here, the authors should name this effect "numerical path- dependency" or similar, as this has nothing to do with the (intrinsic) hysteresis associated with EXP3 in MISMIP.*

Accepted.

> *L405: "using the Schoof sliding law"*
> *How is this choice motivated?*

We chose the Schoof sliding law because, of the three options offered by the MISMIP+ protocol, we find that this one results in the best numerical stability in our model. We will mention this in the manuscript.

> *L425: "...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."*
> *I encourage the authors to provide some rough numbers, in comparison to v1.0 and maybe to other similar models? PISM with 4km resolution in Antarctica (1521 x 1521 x 221) would need for one model year about 10 wall clock minutes on one 2x64 CPU node, with 16km and 64 CPU cores (one socket) it would be about 30 wall clock hours (from Albrecht et al., in review).*

Our earlier response explains the difficulty in directly comparing the performance of v1.0 and v2.0. We will add some more detail about the "basin-scale" experiments mentioned in the manuscript, which consist of 20,000-year pan-Antarctic experiments with a 4-km grounding-line resolution in the Pine Island and Thwaites basins, which can be run in ~24 hours on a 2-core Macbook Pro M2 2023.

> *L465: "The polygon-based routine can be used to increase the mesh resolution over a certain ice- sheet section, e.g. the Pine Island Glacier drainage basin. This is illustrated in Fig. A2."*
> *Where is the polygon-based refinement in PIG illustrated in Fig. A2?*

The polygon-based refinement is not illustrated; we will clarify this in the manuscript.

> *L674: "The number of vertical layers is configurable, and is by default set to 12."*
> *But this would mean for 3 km thick ice more than 600m at the top and about 80m at the base? In order to resolve temperate ice at the base, this is quite coarse (Kleiner et al., 2015).*

This is correct.

> *L714: "matrix whose coefficients depend on the mesh geometry and the ice velocities"*
> *Is Lij in Eq. F7 related to mesh geometry?*

It is, see line 874 (right after Eq. F4): "L^ij is the length of their shared Voronoi cell boundary"

> *L8: "...man-made climate-change-caused mass loss...", also L26 better say "human-made" or "anthropogenic"*

Accepted.

> *L18: "The version control system.." you mean git? Not mentioned before.*

We will state that this is indeed git.

> *L20: "The i/o...", also L55, L334 Define and better write capital I/O.*

Accepted.

> *L23: "earth" Earth*

Accepted.

> *L29: comma before "range"*

Accepted.

> *L56: "It also includes a version control system that includes..." uses*

We do not think this suggestion makes the sentence more readable.

> *L68: " It solves an approximation of the Stokes equations..." "different approximations", or "for approximations"*

Accepted.

L94: "...thus defeating the purpose of the adaptive mesh."
Sounds a bit harsh, maybe use "offsetting the benefits of the adaptive mesh"?

Accepted.

L100: "... adapted into computer code..." "the way it has been implemented"

Accepted.

L104: "The most complete is the Blatter-Pattyn approximation..."
Maybe mention L109 earlier, "can all be derived by neglecting increasingly more terms in the Stokes equation"

Accepted.

L180: Eq. 13 and Eq. 20 use the horizontal nabla operator, should be introduced at some point, maybe in Table 1? It is defined later in Eq. E2 in the appendix.

We believe the nabla operator to be such a basic mathematical tool that we can trust the majority of readers to be familiar with it. The reason we mention its use explicitly in Appendix E is to clarify its two-dimensional (rather than three-dimensional) use in that context.

L216: "... (which is not the square of the basal friction coefficient β, but a confusingly named separate entity, which we maintain for the sake of consistency with earlier literature) for the Weertman-type part."
Then better use β or some different index (e.g. T).*

We believe the confusion arising from this terminology is to be preferred over the confusion of having different letters for the same variable across different papers.

L264: Better not start a sentence with an abbreviation.

Accepted.

L319: "square grids"
Do you mean a regular "Cartesian grids"? Can the domain be also rectangular, or is a square with Lx = Ly required?

We do indeed mean Cartesian grids; we will mention this in the manuscript. The domain can also be rectangular, but the code does require that $\Delta x = \Delta y$ (if this is not the case, it will throw an error).

L321: "...UFEMISM will automatically detect the type of grid from the dimensions of the NetCDF file"

Does this mean, the model checks if lon/lat or x/y are used as dimensions? Or do you also use projection parameters from the metadata in the HDF5 headers, proj string?

It does indeed check only the names of the dimensions (though it accepts multiple variants, e.g. 'x', 'X', 'x-coordinate', etc.). Projection parameters are not read; it assumes the provided grid has the same projection as UFEMISM (i.e. the ISMIP standard projections for Greenland and Antarctica). We will mention this in the manuscript.

L324: "The sparse matrices representing the remapping operators..." This is often called remapping "weights" (e.g. YAC based on CDO).

We will state this widespread alternative name in the manuscript.

L332: "Github Actions"
Better use "GitHub" with capital "H", and refer to the website for the GitHub Actions software development framework (https://docs.github.com/en/actions).

Accepted.

L417: " verified its performance"
What does this mean? I would associate "performance" with numerical efficiency.

We will change this to "verified it in a number of different benchmark experiments"

L436: "define regions where the ice thickness should not change."
For instance, along flow divides. What about defined boundary velocities?

Currently this is not possible, although the basic tools are there. The velocity solver has as optional input arguments a mask defining where velocities are defined, and the accompanying velocity field; right now, the field is simply zero everywhere, but it would be trivially easy to e.g. read a NetCDF with satellite-derived velocities, and provide those to the solver instead.

L488: " nV-by-nV matrix" with V the number of vertices?

nV is the number of vertices in the mesh; we will mention this in the manuscript.

Fig. 1: Maybe provide a scale as measure, or mention the length of the domain for reference. What are the used distance measures here, mentioned later as config variables?

The caption states that the domain covers the entire Antarctic continent, as is clearly visible in the figure. The caption also already states the size of the triangles in km.

Fig. 2: I would assume you used the default DIVA? What kind of architecture did you use, CPU nodes with 2x32 cores? Hence, 128 cores would be associated with 2 CPU nodes using inter communication?

We have will add some additional information about the preliminary scaling tests, concerning both the experimental set-up and the system the simulations were run on. The tests were performed on the Snellius supercomputer, on 128-core nodes. We have also added an additional test with 256 cores to the plot

Figs. 3+4: In the labels, I would expect the FS/HO mean to be the line, and in transparent the ensemble range?

Correct, the legends have the entries for the ensemble mean and ensemble range reversed. We will fix this.

Fig. 5 How does this experiment differ from Berends et al., 2021, Fig10?

Berends et al. (2021) used the hybrid SIA/SSA instead of the DIVA, the flux condition instead of the sub-grid friction scaling, and substantially coarser resolutions. We will mention this in the manuscript.

Fig. 6: The dashed lines in panel a are hard to distinguish, e.g. where is the red 5km solution? The solutions of MISMIP+ ice1r seem to converge for increasing resolution against a solution above the ensemble mean from Cornford et al., 2020. I am assuming the authors used the DIVA stress balance approximation here? This is interesting, as the 750m solution seems to be an exceptions of this convergence? What could be the reason for this resolution dependence. Fig. 11b in Cornford et al., 2020 suggests that higher resolution may provide solutions below the mean, same for the HO contributions in Fig. 9b in Cornford et al., 2020.

The lines in panel A are indeed hard to distinguish because the solutions are very close to each other. We will mention that we did indeed use the DIVA for this experiment.

We suspect it could be possible that the solutions in the higher-resolution simulations (1 km, 750 and 500 m) are starting to show some accumulating diffusion from the many remeshes (which happened more than a hundred times in the 500 m simulation). Possibly this could be solved by setting a wider band around the grounding line where the high resolution should be applied to the mesh generation, which would reduce the frequency of the mesh updates. We will mention this in the manuscript.

Fig. A2: I personally like the humor in this figure but 3 rows would be sufficient to show how line-based, point-based and polygon-based routines work. Also, the refinement seems to be more dense in the half-circle case compared to the circle? I guess this refers

to the different config parameter named in L476? If yes, this information would be helpful in the figure caption.

We agree that steps d and e, while humourous, are perhaps unnecessary; we will remove them. We will also mention that, indeed, the half-circle was given a higher resolution than the circle.