Response to RC2:

This work aims to provide insight into different solver choices for a particular full-Stokes ice sheet model (ISSM) by testing a range of iterative solver choices from the widely available PETSc solver library on a specific test problem (ISMIP-HOM, experiment F) and contrasting with their native/default direct-solve approach (which uses MUMPS). They conclude that switching to the PETSc iterative solvers generally improves time to-solution and scaling as the problem size (number of elements in the finite-element mesh) increases, and are able to provide some suggestions as to which solvers appear to be better suited to their needs. In my opinion, this is a useful contribution to the literature, and I found it to be well-written and well-organized. I do have a few suggestions which I think would greatly increase the usefulness of this work.

My primary criticisms, if you can call them that, are regarding the choice of benchmark problem. While I think that the choice of ISMIP-HOM problem F is a reasonable choice for representing a fully- or mostly-grounded ice sheet (like the Greenland Ice Sheet), I wonder how extendable the results and conclusions are to systems with fast-flowing ice streams and large dynamic ice shelves as are found in Antarctica, represented, for example, by the MISMIP family of benchmarks. In our experience (admittedly not with a full-Stokes model), marine ice sheets are often much more challenging for the linear solvers due to the mathematical nature of the floating ice shelves.
My larger objection is that I strongly disagree with the choice of a linear (constant viscosity) rheology for these experiments. In our experience (again admittedly not with a full-Stokes model), one of the hardest things for many solvers to handle is the large range of viscosities produced by the normal nonlinear rheology. We've often had the case where solvers which perform perfectly well with constant viscosities perform poorly (or fail to converge) when the nonlinear rheology is turned on. I suspect you're getting an incomplete and possibly misleading view of solver performance for "real" ice sheet problems in this case. Is there a compelling reason not to use a "standard" nonlinear rheology for these tests?

Thank you for your review and comments. We used the ISMIP-HOM experiment F test since it involved a transient simulation and is a commonly used benchmark test. The intent of our study was to promote the use of iterative methods over linear solvers using a simplified model, which could then be refined in future work using a real-world simulation. As you mention, the specification of linear viscosity in this benchmark test is a limiting feature in relation to real-world problems. To address the impact of nonlinear rheology on solving the stress balance equations, we are including results from applying the same solvers on another benchmark test (ISMIP-HOM experiment A) that uses a nonlinear viscosity model for ice. We updated the manuscript to highlight the limitations of the transient benchmark test (experiment F) and will include the results from this new study.

Minor points –
1. line 95 – please cite some examples of the full-Stokes solver work that you're referring to

We added additional references to the manuscript.

2. line 102 – "well know" -> "well known"

Fixed typo.

3. line 108 – FS isn't a requirement for active GL dynamics, e.g. MISMIP(1,3d,++). In fact, the authors of this work routinely use SSA for GL problems...

We updated the text to avoid implying that full Stokes is required for modeling grounding line dynamics.

4. line 123 – "suit" -> "suite"

Fixed typo.

5. line 145 – "period" -> "periodic"

Fixed typo.

6. line 171 – please elaborate on or clarify what you mean by "methods that naturally fit the ISSM framework"

We updated the manuscript to be clearer on the point that our intention was to use solvers that did not require customization or specialization of the solver routine within ISSM so that the conclusions based on our results could be used by other models as well.

7. line 172 – using only the default settings for the PETSc components is likely too limiting of a choice. I understand why you'd do that (putting yourself in the shoes of a model-user who doesn't want to fiddle with solver parameters or explore all of the options available). However, we've found that there are cases where minor changes in options result in major improvements in solver performance and robustness. I'd suggest that since the goal of this work is to be a reference for ISSM (and other ISM) users, you should make some effort (maybe by asking the PETSc developers or another linear solver expert for some advice) to make the solvers perform as well as possible. I think this work will have a much larger impact in that case. The other point, of course, is that "default" options can change. I'd suggest presenting two sets of results – one with the "default" settings, and one after some attempt has been made to tune the solver parameters. (it is, of course, possible that the default parameters *do* produce the best performance). Of course, then, you would also need to document the particular solver options you used.

As you summarized, our intention was to highlight strong performance gains that can be attained using iterative solvers in PETSc with limited intervention on the part of modelers (i.e. using default values). In this context, we avoided the large number of options that can be tuned for each combination of iterative scheme with a particular preconditioner and treated each solver with the same level of attention. Also, in light of the simplifications underlying the benchmark test there is no guarantee that speed-ups based on customization of the components would straight forwardly correlate to real-world models. Future work, aimed at refining the results presented in this work, will be based on more realistic models and address the impact of customizing individual components, as you suggested. We updated the manuscript to note that significant performance gains are attainable by customizing the options of the PETSc components for a preferred solver.

8. Conclusion – To give a bit of extra weight to your conclusions, it might be useful to embed your conclusion in the larger context of what many have found to be the case in other scientific computation fields – one suggestion would be to add a statement along the lines of "the conclusion that scalable iterative methods are better suited than direct methods for solving large linear systems echoes the experience of many other researchers across a wide range of scientific disciplines".

Indeed this was the main conclusion of this study. We adopted your suggestion and updated the manuscript to emphasize this conclusion.

9. line 301 – The acknowledgments end with a stray "(" after Jed's name. Perhaps they got cut off?

Fixed typo.

10. Figure 2 – I am impressed with Figures 2-4 – they do a good job of conveying a lot of information

clearly. I'd suggest replacing "horizontal labels" and "vertical labels" with "horizontal rows" and "vertical columns" for clarity in the caption.

Thank you for your comments. We updated the figure caption for clarity as you suggested.

11. Figure 6 – It would be helpful to include an "ideal scaling" line in this plot for comparison. You mention the slopes in the text, but including it on the graph itself can make things easier for the reader.

As suggested, we updated the figures and captions to denote ideal scaling.

12. Figure 6 – More numbers than a single "10" on the horizontal axis would also be useful.

As suggested, we updated the axis bounds.

13. Figures 5 and 6 – If I read these plots correctly (not completely assured due to the lack of x-axis labeling in Figure 6), it appears that the number of elements per processor used for weak scaling in Figure 5 (~250) corresponds to the far-right data points (most processors/fewest elements per processor) in the strong-scaling plot in figure 6. In both of the examples, this is where it appears that you start to see a degradation in your solver scaling, which might imply that you're being a bit too aggressive when you generated figure 5 since you seem to have stepped out of your ideal scaling regime. In other words, it might be the case that if you took a look at weak scaling with more elements/processor (500, perhaps), your MUMPS weak scaling might look better.

While using 250 elements per processor provided the fastest results for iterative methods applied to all but the largest model, your assessment that the scaling deteriorates at larger model sizes, especially for the linear solver is correct. Also, figure 5 had an error in color scale, which misrepresented the results. We fixed this error in this figure and used 500 elements/CPU for presenting weak scaling, as you suggested.

14. Figure 6 – it would be nice to have one more data point for your strong scaling plots, since it appears that your scaling is just beginning to tail off for MUMPS at the largest number of processors. I also realize that it may be a point too far...

The number of points used for scaling was chosen to be consistent with the tests that were performed for all solvers and plotted in Figures 2-4.