

Response to review #3

Thank you for this thorough review! Original comments are in blue, our responses are in black, and hyperlinks in magenta.

Summary

Icepack is an important new model, and a description paper is appropriate. Four features of Icepack stand out: its use of Firedrake/Python, the flexible action-principle design of its stress-balance solver module, its from-the-start attention to data assimilation and inverse modeling, and its design as a modeling environment and language instead of a ready-to-run model. All of these choices are addressed appropriately in this manuscript. Readers are very likely to try the model, which fullfills a major purpose of a description paper at this early-ish development stage.

However, this description paper can be improved in three significant ways:

A. Greater attention to the meaning of Icepack as a *time-dependent and geometry-evolving model*, and to related defaults.

B. Avoidance of *bad linguistic habits* inherited (mostly) from the ice sheet modeling literature.

C. As currently laid out, the paper treats the reader mostly as a potential co-developer of the model, while Icepack's *effectiveness as a simulation tool for science* is muddled.

Fully addressing concern A would require major code extensions, which are not my intention in commenting on this. Rather, the manuscript should make the reader aware of which evolution aspects are well-handled by current Icepack and which are in future development.

The above leading concerns will be addressed in more detail below, in a list which addresses specific line numbers. The associated potential improvement "A", "B", "C" is listed when appropriate.

Line-by-line comments

13, AC: Presumably all software packages can be effective in the hands of experts, and here it is suggested that Icepack stands-out as better for non-experts. My question is whether it will be effective for non-experts interested in announced goals 1 and 4 (among those listed on lines 15–18). This is hard to believe given how Icepack seems to be designed around, and the manuscript focused on, goals 2 and

3. Consider the reader who wants to simulate glacier extent for some years into the future for a mountain glacier or Greenland, who can supply a simulated climate (atmosphere/ocean), a bed topography, and a current geometry in some data files. Does this paper convince me that they need *less* expert knowledge to use Icepack for that purpose than other existing ice sheet models? Not yet. Of course a full usage answer occurs in online tutorials and examples, not just the manuscript. Nonetheless the absence of attention to transport equation boundary conditions, and to how mass/energy surface inputs are handled, gives me the impression that such a reader is on a long co-development path with the Icepack authors, requiring the development of much expert knowledge before first useful results. Said a different way, expert knowledge is required for any software that does not have an aggressive scheme for putting reasonable defaults into the hands of novices. To paraphrase a recent Firedrake paper [Farrell et al 2020], it is a mechanism-vs-policy concern. Icepack has a library of mechanisms, but an absence of apparent policy means expert knowledge is needed to recover usability for real-science applications. I rolled my eyes at the implications of the sentence on line 13, and these concerns remained after reading the whole paper.

This depends on what kind of expertise you mean. We believe that more work can be done to relieve glaciologists *who are not experts in scientific computing* from having to understand, e.g., the details of what preconditioners are used in solving linear systems, what globalization strategy is used in a Newton solver, etc. By way of an “aggressive scheme for putting reasonable defaults into the hands of novices”, we describe our policy at the end of section 6, beginning in line 695. Other software packages have (exactly as you say) all the mechanisms, but the policies on display suggest the priority is for speed above all else. For example, several of the demonstration codes in Elmer default to choices of iterative solvers and preconditioners that work on the particular problem in the example but which are likely to require hand-tuning when generalized to real data. See for example [this line](#) from Elmer’s demonstration of an inverse solver, which uses the GCR scheme and an ILU0 preconditioner to solve the momentum equation. A graduate student could easily try to take this code and use it on real data only to have convergence failures to diagnose. Other examples from Elmer do use direct solvers, for example their [demonstration code](#) to run the ISMIP test case with the SSA model uses a direct solver from UMFPACK for the momentum equation. Other packages show a similar pattern – BISICLES has hand-tuned numbers of Picard iterations and multi-grid smoothing iterations in their demonstration codes. A new user either must (1) know enough about numerical methods to diagnose the problem from bad input data, bad linear solver options, or bad nonlinear solver options, or (2) have a mentor who can. Experienced users have all the tools at hand that they need to create large simulations that will run on supercomputers, but at the same time these choices can create friction for people who may be getting their first experience of modeling by using these examples.

We do not claim to have reduced the need for expertise in glaciology itself, i.e. understanding mass and energy balance, etc. We also consider having some understanding of the different boundary conditions that might apply to the system to be part of this knowledge. The text has been expanded to describe these boundary

conditions in more detail (see comments that follow). Granted, (1) there is more demonstration work that we need to do in order to cover use cases like modeling a small mountain glacier and (2) there are technical hurdles we still need to overcome to do a better job handling the margins. Very likely your recent work on variational inequalities will factor into improvement on the second front.

We have altered the text to state more clearly that our goal is to help users who are not experts in scientific computing. We added the following to the introduction: "We have focused efforts thus far on process studies of individual glaciers or drainage basins (use cases 2 and 3 of the list above). Development of icepack is ongoing and we will broaden our efforts to encompass more use cases in future." We also added the following sentence to the conclusion: "Exactly how to apply principles from HCI to maximize usability is, nonetheless, not an exact science. Our implementation may have failed to meet this goal and changes in future versions will be guided by what users find most difficult."

21–39: I also think Python+Firedrake+PETSc is the most promising environment to build a new ice flow simulation library/model. I'm on board!

44–46, B: Let me vote to **not** maintain the "diagnostic"/"prognostic" linguistic tribalism. The world calls these equations "conservation of momentum" and "conservation of mass". (Indeed one should remind the reader that the latter describes thickness evolution because of how glacier models normally parameterize fluid geometry.) An "also known as the 'prognostic equation' [cite]" is appropriate, but another paper using this tribal language will cause yet more students to need to unlearn silly language in order to read the mainstream fluids and numerical PDE literature.

I agree with you that this is not great terminology, but this isn't a fight we could win. Many other software packages for glacier flow modeling, including PISM and ISSM, use this terminology. No change to the text.

46–47: Here! Here! It is a infinite-dimensional DAE! Good. Many readers will be unfamiliar with the concept; I cite [Ascher & Petzold 1998] for that but there may be better references.

Added a reference to Ascher and Petzold.

56, equation (1) and nearby, AB: Two concerns. First, it is later acknowledged that this is not really an advection equation (lines 376–377), so one does not need to call it that here either. The SIA is not some weird alternative universe of weak-willed modelers, it is what **all* models should produce** in the large. That model, the only clearly-understood coupled model, makes equation (1) a diffusion, as noted. Surely calling this a "transport equation" or even "thickness transport equation" is adequate. One then points-out that $q = hu$ is one way to parameterize flux, and that u comes from a coupled equation *driven by* ∇s (even in the Stokes case). It might be acknowledged that (1) does not have a PDE "type" in the classical sense. (It is a DAE, after all.) Second, equation (1) holds with what boundary conditions? This manuscript maintains the tradition of pretending not to notice. To quote [Schoof & Hewitt 2013], "A sometimes weakly perceived point in glaciology is that the model above is in fact a free-boundary problem ...", and **this applies to (1) regardless of the stress balance model**. Is this paper just going to pretend boundary conditions for the main, and first, equation don't exist? (Or pretend that all glaciers end in cliffs

of sufficient height so that the singular change of coordinates (16) is no problem?) This manuscript could even cite existing wellposedness literature for (1), especially [Calvo et al 2003] and [Jouvet & Bueler 2012], which confirm the meaningfulness of the free-boundary problem formulation and the resultant simultaneous Dirichlet ($h=0$) and Neumann ($q=0$) boundary conditions at grounded margins. And then say what Icepack will when solving (1).

We removed the statement that this is an advection equation and added the following sentences: “This problem has the apparent form of a conservative advection equation, but the velocity u is coupled to the thickness and surface slope in such a way that the whole problem is not hyperbolic. For the specific case of the shallow ice approximation (see section §2.2.1), the coupled system is parabolic. In all other cases, the problem does not have a PDE ‘type’ in the usual sense because the velocity is found through solving an elliptic PDE where the thickness and surface slope are coefficients.”

Our discussion of boundary conditions was mostly lacking. We added the following:

“We implement two types of boundary conditions for the prognostic equation. Users can specify an inflow flux value and this value becomes a source of thickness at any point along the domain boundary where the ice velocity is pointing in to the domain. The flux at the inflow boundary can change in time. Second, we impose outflow boundary conditions on any part of the domain where the ice velocity is pointing outwards. Which segments of the boundary are inflow or outflow are diagnosed automatically by calculating the sign of the dot product between the velocity and the unit outward normal vector.

The mass transport equation for ice thickness is a free boundary problem, where the free boundary is the contour between ice-covered and ice-free regions (Schoof and Hewitt 2013). A naively-implemented prognostic solver could erroneously compute negative thickness values in subsequent timesteps when there is ablation in ice-free regions. A common and ad-hoc approach to work around this issue is to truncate the thickness at zero at every timestep. The principled approach is to instead treat the free boundary problem directly as a variational inequality (Jouvet and Bueler 2012). Icepack currently lacks a principled scheme for tracking this free boundary and we instead rely on truncation. Treating this problem as a variational inequality in icepack will be the subject of future development. PETSc includes scalable solvers for variational inequalities (Bueler 2020) that are also available through Firedrake.”

We also added a new section which is now section §2.3 in the text on the boundary conditions for the diagnostic models.

68, B: It is not clear why the conservation of momentum model for a slow fluid should ever be called a "momentum transport" model. There is no d/dt for momentum! Surely "conservation of momentum equation" or "stress balance equation" suffice.

Changed from “momentum transport” to “stress balance”.

74, B: No, the Blatter-Pattyn (BP) equation is not a "hybrid". This word is perfectly descriptive when categorizing ice sheet momentum models, and I am not sure who decided to mangle the language in this way. (It precedes this manuscript.)

There are principled models, of which SIA, SSA, and BP are three, arising from small parameter arguments. There were horizontal hybrids (Ritz) and there still are vertical hybrids (Pollard, Bueler, Winkelmann). Hybrids, by definition, combine more than one principled model in some (less principled!) manner. If you are trying to say that BP is a shallow model, but less-shallow than SIA and SSA ... that's true! And if you are saying that BP can balance stresses in a plug or a shear flow, that's true too. Say those things.

This was bad terminology. I adopted it before there were many other contributors to the project and everyone who has come along afterwards has told me that it's confusing. What I had imagined is that a discretization of BP using only a single vertical layer and relatively low-degree (up to 4) polynomials in the vertical direction could be considered a kind of approximate model in its own right, similar to how oceanographers will use semi-discrete two- or three-layer ocean models. In the next release we will change the name to "Blatter-Pattyn" or "First Order" and deprecate the name "Hybrid" for removal in a future release or until we develop an actual hybrid model which is less computationally intensive than this Thing That I Should Not Have Called a Hybrid Model.

101–102, B: Regarding the sentence "The action principle can be viewed as a consequence of the Onsager reciprocity relations for systems near to equilibrium ...": I am trying to imagine a reader for whom reading that would be a useful learning experience regarding the action principle, as opposed to an irritating one.

The previous two sentences have been removed from the text but we've kept the reference to the book by De Groot and Mazur. When I've spoken to students about action principles and why they're useful, I usually highlight the numerical advantages of solving convex optimization problems as opposed to the more general problem of solving large nonlinear systems of equations. But the question invariably comes up as to why a given physics problem should have a minimization principle at all while others do not. Non-equilibrium thermodynamics provides some insight and I think we would be doing a disservice by not mentioning this at all.

108: I do not understand the intended meaning of "vastly more convenient numerically" here. The cost of evaluating the objective, versus a merit function like the square of the residual norm, is the same. The cost of evaluating a residual is the same whether or not it happens to be a gradient of an objective. Perhaps what is meant is more like: "Minimizing a convex functional provides better convergence guarantees than solving a general system of nonlinear equations ...", which is certainly true. At least a forward reference to section 4.2 would help explain the phrase.

Added a forward reference to section 4.2 and changed this sentence to "Algorithms for minimizing convex functionals have better convergence guarantees than algorithms for solving general nonlinear systems of equations while having no additional computational cost" with a reference to Nocedal and Wright.

115, equation (5), B: Now the bad linguistic habits are being inherited from the FE world. This is not "mass" in any sense other than that any matrix with entries $a_{ij} = \int \psi^i \psi^j dx$ is always called a "mass matrix" in the FE world. And the reader may not have that FE bad habit already, in which case confusion ensues because (5) has nothing to do with mass conservation! My suggestion for this term, implicitly

acknowledging how it arises in the SIA, is to call it "localization".

Changed to "localization".

115–124: The action combining (5) and (6) is already convex and coercive in L^2 . The reason to add the penalty term is (presumably) because the FE space choices want to work in H^1 . (Adding (7) with $\ell > 0$ makes J coercive in H^1 .) This penalized form is a perfectly reasonable idea, and it makes the SIA behave more like other stress balances, *and* it is one of those unprincipled things one does to get it all to work properly ... if you add another kludge you could even call yourself a "hybrid".

126: This author's name is Bueler not Beuler. (The latter is a PETSc `-ts_type`, so it is easy to get confused.)

Apologies, the text has been corrected.

137–140: There is a double negative in equations (12) and (14) which is not used in (6) and (8), respectively. Does this reflect anything important?

These modules were developed by different authors. We will get rid of the discrepancy in a future version.

148–149: It is of course true that for general boundary conditions the "shallow stream equations do not have a simple analytical solution." What is probably meant here, however, is that there is not a well-known, exactly-solvable, basal-friction-included boundary value problem suitable for testing. But that's not true, and indeed the *very first theoretical paper in glaciology* provides one, namely [Böðvarsson 1955]! See the full story, and a derived marine flow-line exact solution suitable for testing, in [Bueler 2014]. The method of manufactured solutions applies, of course, but at loss of clarity on the meaning and magnitude of the resulting numerical errors, and with great danger of testing the wrong parts of a (nonlinear) system phase space, and with loss of the history of mathematical glaciology.

Added a reference to Bueler 2014 and Böðvarsson 1955. You make a great point about testing the wrong parts of a nonlinear phase space and this was actually a bit of a struggle when creating the tests for this solver. You could put in some thickness and velocity fields and a PDE and SymPy will spit out a manufactured friction, but there's no guarantee that the result will be physically reasonable. I had to manually adjust the input parameters so that the basal shear stress values would come out to something sensible. A sentence has been added to the text to describe this part of the testing process.

158: I guess the phrase "individual fast-flowing glacier" arises because of the sense that the SSA does not handle slow flow very well, which is true, and that fast ice is separated by slow flow. Nonetheless, I would replace "features of an individual fast-flowing glacier" by "features of fast-flowing glaciers".

Done.

170–171: I am not clear on "a tensor product basis of Lagrange finite elements in the vertical and higher-order polynomials in a single vertical layer". Should the first "vertical" actually be "horizontal"? The novice reader not already used to FE stuff could use a figure or more words here, I suspect.

That was a typo, fixed.

175, equation (16), A: From here forward, in the manuscript, the lateral boundary condition for equation (1) is even less clear because this change of variables

takes the lateral free margin and blows it up. (In the grounded ice case, and where "blow up" is used in the mathematical sense.) With what consequences for simulating moving margins? Section 4.1 does not address this (i.e. how the boundary condition for the transport equation behaves as $h \rightarrow 0$; a boundary condition is not mentioned). I believe, on the other hand, that the SIA grounded-margin example in section 5.2 does not use (16), which as described applies only applying to "the hybrid flow model", but section 5.2 does not mention the margin anyway. I am concerned that positive consequences of using (16) are emphasized while negative consequences are not mentioned, and that Icepack has no path around these negative consequences.

Section 2.3 in the text describes the lateral boundary conditions for the diagnostic models and an admission that we assume the entire spatial domain is ice-covered. It's possible that I've painted us into a corner with this choice. In the $h \rightarrow 0$ limit, the only singularity in the equations is in the vertical shear stress, which – including a factor of h from the determinant of the coordinate transformation – scales like h^{-1} . A dirty hack would be to replace this term with $1 / \max_value(h_min, h)$, effectively clamping it at the reciprocal of this minimum thickness. This would still permit ice-free areas at the risk of underestimating vertical shear stresses in areas where the ice is thinner than whatever threshold value was used. Alternatively, we may have no choice but to switch to a z -coordinate model in future versions.

215: For clarity to readers not already thinking about Firedrake's extruded meshes, I suggest adding a parenthetical: "linear or quadratic elements on (horizontal) triangles".

Done.

216: Instead of "Lagrange interpolating polynomial basis", which readers accustomed to numerical analysis terminology (outside FE) might associate to the Lagrange method of finding interpolating polynomials, it is probably clearer to just say "Lagrange elements". Perhaps emphasize the nodal placement? (Legendre nodes, concentrated near endpoints, versus the equally-spaced nodes of Lagrange elements.) Orthogonality is not the only reason why Legendre is better; node location implies better approximation properties.

Done and added a reference to Szabó et al 2004.

227: "Smoke test" is an unfamiliar idiom to me. As in testing machinery or testing electronics or smoking-out bugs or what-am-I-smoking? Pity the non-native speaker.

The term comes from electronics. "Plug the circuit board in. If you see smoke coming out, unplug it. You don't need to do any more testing." Changed to "sanity test".

227: "most minimalistic" \rightarrow "minimal"

Done.

234–235: I have received the advice "write about what you have done, not what you haven't", and I think it applies here. Using two layers in this way assumes one polythermal structure and will not allow transition to the other. No need to consider or mention it.

These sentences have been removed.

253–289: This chunk of text is important, and it describes one of the best aspects of Icepack’s use of Python. I think it should be put at the start of subsection 2.3, which will give clarity that when equations (22) and (23) appear, they are primarily an example of submodel plug-ability.

283–289: I’m on board with keyword arguments!

298–299: Suggested replacement for the mushy phrase: "describing realistic glacier flows, but it is not sufficient by itself" → "parameterizing fluidity".

Done.

300–301, C: I seriously doubt that more than one user in 100 will want to substitute their own melt-fraction dependence, because no one has the data for it. So: "likely" → "possible". (This is a clear case where good defaults are more important than modularity.)

Changed to “Some users of this module may want to substitute in their own parameterization for melt fraction dependence.”

297–305, C: This paragraph reads, to me, like an argument that expert knowledge *will* be required for the Icepack user. ("users must calculate the fluidity field themselves") That is, unless Icepack includes both defaults and higher-level simulation drivers which allow the novice user to never know about these possibilities. More broadly, what does Icepack design offer, when it comes to avoiding explicit choices of all submodels, on the way to a first scientifically-useful model for the beginner?

What you’re describing is knowledge of glaciology, not scientific computing. We have stated explicitly in the first sentence of section 6 that we aim to reduce the need for users to be experts in the latter but not the former. If we were to define how some fields are coupled, we would be faced with a difficult design choice of where this is or is not appropriate. Including the effect of temperature on rheology seems fairly obvious. Some modeling papers also account for much higher fluidity for ice that accumulated during the last glacial period because of the greater dust content. Should it also be the responsibility of the package to include this effect? We did not feel that, at this stage of the development, we could also implement high-level simulation drivers that hide these details while also satisfying our other design goals. See also response to the first comment and the text added to the conclusion. We believe that our design choices will ultimately enhance usability for the intended audience and we did our best by applying principles from HCI. But we may well have missed the mark and the interface may change to respond to evolving understanding of our target audience’s needs.

318–320: Mushy sentences. Needed?

We wanted to emphasize that this model is very ad hoc and that we’re not passing it off as more accurate than it is. These sentences have been removed.

345, C: The sentence "The key classes that users interact with are flow models and solvers" is right at the heart of my concern C. Indeed, after developing PISM with much (developer group) concern about flow models and solvers, we spent the next 10 years talking to users about their data formats, parameter-study schema, and surface process models (supra, sub, and calving front). Barely had time to work on flow models and solvers ... until I quit day-to-day PISM development. The by-far largest population of scientists care about how flowing ice interacts with their

climates. They use ice flow models on the assumption that ice flow modelers know how to model ice flow! This "key classes" sentence is not describing users, it is describing co-developers.

Again I think this seems to be a disagreement about what the software package should do and what the users should be responsible for. Icepack lacks any of the high-level simulation drivers that PISM has. Giving users a relatively lower level interface to the physics solvers was a concession to the fact that we can't predict what they will want to do with an ice flow model by way of coupling to climate, oceans, solid earth, etc. Obviously you and the developers of PISM have had far more experience. We may well have misjudged what glaciologists do and don't find difficult about modeling and what type of interface best serves the needs of the most people. See response to first comment.

368: "Practicing ... may be unfamiliar with" is unnecessary. Just say what you have to say.

Changed to "Some glaciologists may be unfamiliar with the Courant-Friedrichs-Lewy (CFL) condition."

362–381, A: Three concerns about section 4.1. First, as already noted, transport equation (1) has boundary values and they are again silent here. Second, there are multiple transport problems in glacier modeling: equation (1), the advection-dominated enthalpy equation, the ice shelf damage transport equation, the age equation, ... etc. Is this paragraph covering them all, or just equation (1)?

See response to previous comment on boundary conditions. We cut some of the text in this section in response to comments from reviewer #2 and clarified that we were referring to the mass transport equation.

368–373, AC: The third concern, regarding this paragraph, is that Icepack apparently has not yet set-up effective adaptive time-stepping. (I have not gotten a runtime error from a modern explicit ODE solver for a long time! Have you? Some runs are slow cause they are stiff, indeed.) This should be addressed/acknowledged. In particular, assuming a perfectly-implemented *implicit* solver, how does Δt scale with Δx ? If the problem were really advective, so that the goal is to model influences as they travel at the characteristic speed, then the answer would be $\Delta t = O(\Delta x)$ for accuracy, even with your implicit scheme. (Better than $O(\Delta x^2)$, yes indeed ... but then explicit steppers would be fine ... but (1) is not an advection ... we must think more.) Though the problem is actually some diffusive/advective mix, some defined scaling of Δt with Δx is still needed for accuracy. This can come from an adaptive time-stepper, or be designed from scratch. Then there is the matter of the user's data time scale for surface mass balance, etc., which implies "events" in the adaptive time-stepper (when data is read). In fact, the actual questions a glacier model must answer, regarding time steps, are these: How does time-stepping change under spatial grid refinement? Does it converge in space-time refinement? Is it robust over realistic geometries and inputs? What is the user interface? This paragraph convinces me that Icepack is not yet there, which should not be completely buried.

Text has been reorganized, and we added "Icepack currently lacks an adaptive timestepping scheme. Unconditionally stable schemes allow taking long timesteps, but taking very long timesteps will give inaccurate solutions. At present, users are

still responsible for checking the accuracy of their results, for example by running at more than one resolution. Adaptive timestepping will be added in a future release.”

This is one of the areas where we don't (yet) live up to our goal of keeping glaciologists from having to worry about numerics. Our plan is to use the package `fire Drake_ts`, which offers an interface to the PETSc timestepping schemes.

363: "equation equation"

Fixed.

365, A: The "In the interest of simplicity" phrase here tells me the authors simply have not thought-through the time-evolution of glacier geometry and the needed boundary conditions. (You have a scheme for unconditional-stably generating glacier surfaces which conserves mass in the presence of ablation? Then don't hide it!) Please take the problem seriously: Address how you maintain reasonable margin shape and positive thickness in an implicit scheme. It is o.k. to admit that time steps cannot be arbitrarily long in your scheme, to get convergence; the "conditional/unconditional stable" language is an artifact of linear PDE theory, and your problem, taken seriously, is super-duper-nonlinear.

See response to previous comment. We've changed the text to say that the scheme is unconditionally stable for the advection equation with a note that the coupled system is not linear nor is it hyperbolic. We also changed a later sentence to: "Implicit schemes allow taking longer timesteps than explicit ones, but taking very long timesteps will give inaccurate solutions and, in the presence of ablation, may yield negative thickness values."

376–377, B: This sentence is a very good argument for *not* calling (1) an advection equation.

383–444, A: Sections 4.2 and 4.3 convince me that when the Icepack team takes on aspects of model design they care about then it comes out very well! These sections suggest the momentum balance ("diagnostic" ... `grrr`) solver has great defaults. Likewise with sections 4.4 and 4.5. (Now for serious attention to time-stepping, mass conservation, and the user's surface mass- and energy-flux data, and clarity on Icepack's TODO list.)

459–462: Gauss-Newton is a good choice for this purpose, as pointed out for exactly this purpose, inversion of glacier models, by [Habermann et al 2012]. Which should be cited.

Added a citation to Pratt et al 1998 and Habermann et al 2012.

463–466: Symbolic derivatives. Again I am on board with the benefits of a Firedrake-based tool chain.

469, A: Here "terminus boundary condition" refers to the momentum balance equation. There is, as far as I can tell, no regard for the "terminus boundary condition" needed when solving equation (1), i.e. the mass conservation terminus boundary condition, even in the case of a cliff, much less a grounded margin.

See response to previous comments.

474–487: Extruded meshes. On board with Firedrake. But if the model claims to have a solution of (1) then there would be some mechanism for addressing regions which become ice-free within a time step? No mention thereof.

You can use extruded meshes of variable thickness and that thickness can become 0. So the choice of extruded meshes does not force one to use terrain-

following coordinates as we did, whether or not that was a good idea. No change to the text.

493–515, A: Section 4.5 describes a nice solution to a real problem. In fact, can one consider this kind of fit to the glacier profile at a grounded margin? Not obvious how to proceed, I agree, but note that the thickness field h of a glacier has the similar continuous-but-non-differentiable character as the pressure on the ocean calving front.

The technique might be applicable in other areas but we haven't had the chance to try it yet. No change to the text.

517–522: Rich problem-aware preconditioners and solvers. On board with Fire-drake.

523, A: This honest, clear sentence is not at all expected by the reader who remembers lines 15-19 in the Introduction. Please let the reader know earlier.

We added the following to the introduction: "We have focused efforts thus far on process studies of individual glaciers or drainage basins (use cases 2 and 3 of the list above). Development of icepack is ongoing and we will broaden our efforts to encompass more use cases in future."

533–586: The MISMIP+ experiment is part of the 2008-onward tradition of running ice sheet models in rectangular boxes (MISMIP, EISMINT-HOM, ...). This is fine in a field where one also has some laboratory fluid models, i.e. actual stuff, which fit in boxes. But this tradition has a distressing impact on new model development, which is to modularize around the choices one makes between intercomparison-specified boxes and their associated boundary conditions. (Yes, that is what Icepack looks like right now.)

We included this assuming that it's a necessity to repeat at least some of the standard benchmarks from the literature in order to publish a new model. No change to the text.

588–597, A: This example would be more impressive with hybrid physics, right?, but Icepack is not there yet. (Or else it would have been applied here.) Readers of GMD should be taken to be serious people. Tell them the score, and I don't mean relative to what other models are capable of. What needs to improve?

See response to previous comments. The point of the exercise was to reproduce at least some of the results from a prior paper that used the shallow ice model exclusively.

598: Regarding "computationally cheap enough ... in a matter of minutes on a desktop": I suppose the excuse for this sentence is that other people get away with writing such stuff? The question is how run time scales with mesh resolution. The reader can handle a plot, if you want to generate one.

We avoided discussing performance or scalability considerations in this paper. See our responses to reviewer #2. These will be the subject of future work.

604–647: Nice examples! Ice shelf physics is a scope where Icepack is a convincing choice for a research project.

662–663: Regarding "The person learning ... is largely absent from the discussion", have you looked at the PISM User's Manual? (Start from page 1.) This opinion ranges from disputable to insulting, but mostly reflects not paying too much attention to unpublished prior literature (i.e. online manuals). On exactly that topic, the

online Icepack documentation is excellent.

This sentence has been removed.

666–713: Saying these HCI principles out loud is a very worthwhile aspect of the manuscript.

677–678, C: There is a *big* gap between "user interface in an interpreted language" and a "program that can only run in batch mode". C-based PETSc programs like the PISM ice sheet model, and many other command-line programs, live in the interior of this gap. For example, please consider these three HCI principles as they apply to command-line git. It is neither "a user interface in an interpreted program" nor a "batch mode" only thing. (Most command-line tools are not!) Git has a steep learning curve, because it is a DAG modeling language, but such command-line design can hit your principles too. Indeed, a good antidote to your false dichotomy is [Brown et al 2014], and addressing UI points made there would increase the credibility of section 6. (You'll see that Icepack is definitely a Brown-approved design library-wise, but doing science is not equal to designing a library API either.) Ultimately any science application of Icepack will be a map from inputs (data) to outputs (simulations or inversions), and the usability of that map (e.g. in ensemble simulations) is different from the develop-ability of it in an interactive environment. Python is a great environment for experimentation, and for ice sheet modeling, and progressive evaluation is a desirable principle, but an interactive Python session is not the only alternative to 1990s climate-model design. Presumably Icepack usage is intended to progress from all-interactive mode to parallel production/ensemble runs in batch systems anyway? Address that?

We added the statement: "Simulations that have been debugged can then easily be transformed into a single Python script, for example using the tool `nbconvert`, for production runs and parallel execution. In other words, while there is an interactive interface, there is also a faster batch mode interface as well."

681–683, C: Regarding "The API ... one step at a time": There is a *huge* amount of expert knowledge required to do science with an "ice sheet model" which does not have a policy for doing things "one step at a time". Imagine a paper that proposed a new-and-better WRF model but said something like this about the demands on the user!

682–683, C: Also, this "user" is a co-developer.

Several authors have noted the necessity of providing a path for users to become developers for the long-term health of an open-source software project. See for example Turk (2013), Scaling a code in the human dimension or Bangerth and Heister (2013), What makes computational open source software libraries successful?

693–713, C: This basic point about abstraction gradient is good. But solver/discretization components/choices is not the only such gradient. From talking to a lot of novice glacier modelers, I assert the key abstraction gradient, which an ice sheet model must finesse, regards ice flow physics. (What aspects of the full, coupled, nonlinear dynamics are the next ones that the user's constructed Icepack model should/can handle for the science goal? How to build-in that physics without unnecessary parameters? How to generate intermediate results which reveal which processes are missing, re the science goal?) Thinking on this stuff is where Icepack developers could make their next real progress. That means de-emphasizing solver compo-

nents/discretizations in the user's view, primarily by setting aggressive defaults, even as the developers must get solvers right (which is the strength of Icepack).

It's possible that icepack will evolve to have a more end-to-end simulation driver as the main interface. See changes to intro and conclusion.

721: "physics"

Fixed.

724: Having reviewed quite a few ice sheet modeling papers, every single one claimed something about its usability. Less mush, more answers please.

Changed to "Relatively few works in the computational science literature draw directly from relevant work in HCI when discussing usability; see for example Hanay et al. 2009, Harris et al. 2020."

725: I'm on board with (2), and wish for more careful analysis of Icepack's contribution here, but "quantifying" in (1) is not what you are doing in this manuscript. (Nor, probably, do you have the ability to do it.)

Expanded to: "We believe that this is because of two difficulties. First, the degree to which usability is a rate-limiting factor for scientists is hard to quantify and likely differs widely across disciplines. Second, concretely assessing what features make software tools more or less usable is highly subjective." We're making an unsubstantiated claim that usability is indeed a rate-limiting factor. We could be wrong about this. Even if we're right, the steps that we took to address this problem could have been misguided.

732-740, AC: If I saw "4. robust time-evolution and climate interaction tools", or similar, on this list then I would be more of a believer in Icepack's future. Will you be able to break out of stress-balance-and-inverse-model-solver-playground mode, and start answering some of the science questions enumerated in the Introduction?

We have changed the list to include "improved physics formulations and solvers that work in ice-free areas".

References

Ascher, U. and Petzold, L. (1998). Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations, SIAM, Philadelphia.

Böðvarsson G (1955) On the flow of ice-sheets and glaciers. *Jökull*,

5, 1-8 (find this too-little-read paper at: <https://github.com/bueler/bod-marine/blob/master/Bodvardsson1955>)

Brown, J., Knepley, M. G., & Smith, B. F. (2014). Run-time extensibility and librarization of simulation software. *Computing in Science & Engineering*, 17(1), 38-45.

Bueler, E. (2014). An exact solution for a steady, flowline marine ice sheet. *Journal of Glaciology*, 60(224), 1117-1125.

Calvo, N., Díaz, J. I., Durany, J., Schiavi, E., & Vázquez, C. (2003). On a doubly nonlinear parabolic obstacle problem modelling ice sheet dynamics. *SIAM Journal on applied mathematics*, 63(2), 683-707.

Farrell, P. E., Kirby, R. C., & Marchena-Menendez, J. (2020). Irksome: Automating Runge-Kutta time-stepping for finite element methods. arXiv preprint arXiv:2006.16282.

Habermann, M., Maxwell, D., & Truffer, M. (2012). Reconstruction of basal properties in ice sheets using iterative inverse methods. *Journal of Glaciology*, 58(210), 795-808.

Jouvet, G., & Bueler, E. (2012). Steady, shallow ice sheets as obstacle problems: well-posedness and finite element approximation. *SIAM Journal on Applied Mathematics*, 72(4), 1292-1314.

Schoof, C., & Hewitt, I. (2013). Ice-sheet dynamics. *Annual Review of Fluid Mechanics*, 45, 217-239.