Response to Reviewer #1 (T. Zwinger)

March 13, 2015

1 General comments

The paper provides an outline of a new apparently modular built ice-sheet code deploying the so-called Blatter/Pattyn or first order (FO) approximation to the Stokes equations. As many other new generation ice-sheet codes, the linear algebra part is left to state-of-the-art HPC suites for solution of sparse matrices, in this case, Trilinos. Also the Albany code-base seems to be a well established platform providing a good interoperability with external libraries and built-in functionality for data assimilation. This appears to be a good example of high level computational science being deployed in the field of applied sciences - in this case numerical Glaciology. FO models themselves are not a novelty, neither are inverse methods/data assimilation nor high performance computing in the field. Nevertheless, Albany/FELIX combines those aspects in a new way. In that sense, this article is well suited for publication in GMD. My general impression of the article is, that it is well structured and consistent. The language for me as a non-native speaker is without errors. I have one or two scientific concerns (or perhaps issues that demand deeper explanations), which I will discuss in detail in the following. If these points are addressed, I would recommend publication.

We thank the reviewer for his thoughtful feedback, which has been addressed in the revised manuscript, and has helped us to improve our paper. All changes to the original manuscript are marked in blue in the revision. You will find also our responses to your comments below, in blue.

2 Main points of criticism

My main concern is about the manufactured solutions you present for testing a 2-D FO model. Your manufactured solutions, which you claim to be tailored for testing FO codes, apply on two-dimensions, so I would have expected to see flow-line problems. The essential feature of Blatter/Pattyn (a.k.a. FO) is the hydrostatic assumption (Greve and Blatter, 2009). This eliminates the pressure out of the equations and as you correctly state releases you from the pain to solve a saddle point problem. It also further eliminates the third velocity component and the longitudinal vertical stresses out of the system, thus rendering the problem to be 2-D-ish. Nevertheless, there would be still vertical, z-derivatives due to the shear stress components in the equation system. And that exactly raises suspicion from my side that you are not really testing the 2-D FO problem with your setup (as you claim to do). From your equation (22) I conclude that you basically reduce your 3-D problem (2) to the horizontal plane, i.e., setting all vertical derivatives (i.e., the ones in z-direction) that still would exist in 1,2 to zero. You actually never explicitly define the two-dimensional effective shear rate, $\dot{\epsilon}_{2-D}$, but from the term in brackets in equation (23) I conclude that it is the original 3-D version stripped of all z-dependencies, which would underline my suspicion.

The reviewer is correct in how it is defined. We have made the notation more clear in the revision.

To cut a long story short: your 2-D problem setup does not reflect a flow-line FO problem (which from your words I would expect) but rather a single layer horizontal solution with no vertical shear, which reminds me rather to a Shallow Shelf (SSA) and not a FO problem. But perhaps it is your intention to proceed like that and I missed the point. What I would ask to get from you is a good argument that these cases are still a proper verification of the FO code in 2-D, as you claim in the text, or else some additional paragraph that sheds light on why you construct your solutions like that.

The reviewer is correct that our MMS solutions were not to the 3D FO Stokes equations and lacked the vertical shear term. The equations were obtained by neglecting the $\partial/\partial z$ terms from the FO Stokes equations. The test cases were not intended to verify the 3D FO Stokes equations: rather, they were intended to be used as part of a multi-stage code verification that includes also verification of the full 3D FO Stokes equations using code-to-code comparisons and mesh convergence studies on realistic geometries (Sections 5 and 6 in the paper). We have attempted to make this more clear

in the paper, and also have made it clear that our MMS problems are for a 2D version of the FO Stokes equations, not the full 3D equations.

We feel despite being simplified, our MMS test cases are nonetheless useful. The task of deriving source terms for an MMS study for the 3D FO Stokes equations is cumbersome, if not intractable. In contrast, our MMS problems are simple enough to be implemented by anyone simply by referring to the expressions in our paper.

We do agree with the reviewer that a test case based on equations which include vertical shear would be worthwhile. To address this point, we have derived an MMS test case for the FO Stokes equations in the x–z plane (obtained by neglecting the v and $\partial/\partial y$ terms in the equations), and showed some mesh convergence results for our code on this test case. Please see Section 4.2 of the revised manuscript.

My second point is about that you actually never provide information on whether Albany/FELIX is capable of doing prognostic runs, which I think is essential for an ice-sheet model, in particular if one wants, as you claim, to couple it to Earth System Models (ESM). That, in consequence, would include a discussion of the thickness evolution equation, which can be numerically quite tricky. Further it would impose the difficulty of dealing with in time changing meshes. If you could shed some light on the prognostic capabilities of your platform, this would be valuable information for the reader.

How we plan to couple to ESMs and do prognostic solves is a good question. The reviewer is correct that *Albany/FELIX* can only be used for the diagnostic stress-velocity solve, and does not discretize the ice temperature and thickness evolution equations. To do prognostic runs and couple our code to ESMs, we have interfaced *Albany/FELIX* to two other land ice codes: the *CISM* (Community Ice Sheet Model) and *MPAS-LI* (Model for Prediction Across Scales - Land Ice) codes. In the resulting dynamic cores (dycores), termed *CISM-Albany* and *MPAS-Albany* respectively, the steady state stress velocity solve occurs in *Albany/FELIX* and the ice sheet evolution (including updating the ice extent, if the ice has advanced or retreated) is calculated in *CISM* or *MPAS*. The details of these dycores will be presented in a subsequent paper, so we had intentionally omitted the discussion of *CISM-Albany* and *MPAS-Albany* from this paper. We agree with the reviewer that some mention of these dycores is worthwhile so the reader can see how we will do prognostic solves and couple to ESMs. It has been added to the "Conclusions" section of the paper. If the reviewer would like to learn more about *CISM-Albany* and *MPAS-Albany* and see some results from prognostic runs

produced by the code, there are some presentations on the subject available at: `http://www.scidac.gov/PISCEES/presentations.html`.

Finally, if I correctly recall the recommendations of GMD and also in view of perhaps your own interest that the code rises attention in the wider community you should include information on the license(s) the code and its components are published under and - if so - how it is accessible for other scientists. This might be clear for the separate components (like Trilinos), but even if parts of codes are open source, there can be combination of licenses (even open source flavours, like GPL, MIT, BSD) that might impose issues. And gathering this information in this paper is an asset.

We have added a "Code Availability" section to the paper that describes the availability of the codes. Both *Trilinos* and *Albany* have BSD-type licenses and can be cloned from their `git` repositories by anyone interested in looking at the code. The *Albany/FELIX* solver described in this paper is not publicly available at the present time. A public release of the code as part of *Albany* is planned for 2015.

3 Technicalities (sorted by their occurrence)

Line numbers refer to the printer-friendly version of the text downloaded from `http://www.geosci-model-dev-discuss.net/7/8079/2014`.

1. page 8080, line 15: ". . . discretized using structured and unstructured meshes." The wording structured might be misleading, as in FD/FV community this is synonymous to i,j,k indexed points/cells. FEM inherently is unstructured by its method (or are you taking advantage of the structure in your assembly of matrices?). I would– and this occurs a few times in the text– rather use the terminus layered mesh.

   The reviewer is correct that the FEM is inherently unstructured. No, we are not taking advantage of the structure in the matrix assembly. We do take some advantage of structure in the z–direction, e.g., in creating the decomposition for a parallel run (described in Section 6.1). We have replaced "unstructured and structured" with "tetrahedral and hexahedral" where this terminology appeared in the paper to prevent confusion. We did not change to the term "layered mesh" as all the meshes we consider (both structured and unstructured) are structured (layered) in the z–direction.

2. page 8081, line 7: opinions on what model is part of the "new generation" club are somewhat subjective and naturally have a tendency to not include those models from scientists that are in lesser proximity to oneself. But for instance (and there might be even other examples) SICOPOLIS, despite being around for some while, has been used as an adjoint model (Heimbach and Bugnion, 2009) as well as been coupled to climate models (Vizcano et al., 2009) and similar to PISM also has a hybrid approach between SIA and SSA (Sato and Greve, 2012) and has a community around it (it is open source with multiple contributors).

   We thank the reviewer for pointing us to these additional references. We have added some of these to the bibliography, but not all. Our list of references was not meant to be exhaustive or all inclusive, but merely representative. In an attempt to clarify this, we have added "e.g." at the beginning of the list of references.

3. page 8081, line 14: You might include the early work of Pattyn (2008) in this list.

   We have added this reference.

4. page 8081, line 10: the abbreviation HPC (High Performance Computing) is not explicitly given - minor thing, but readers not familiar with it might wonder.

   We agree that the abbreviation HPC should be defined prior to its first use. This has been done in the revision of the paper.

5. page 8082, line 1: Gille-Chaulet $\rightarrow$ Gillet-Chaulet

   This was a typographical error, which we have remedied.

6. page 8082, line 1: missing references: (Jay-Allemand et al., 2011) and (Favier et al., 2014) (see in References)

   As we stated in our response to 2., our list of references was meant to be representative, rather than all inclusive. To clarify this, we have added "e.g." at the beginning of the list of references. We feel our list of references is sufficiently representative.

7. page 8082, line 16: minor technicality - "HPC computing platforms" would read as "High Performance Computing computing platforms"; so, perhaps change to "HPC platforms".

We agree, and have modified the text accordingly.

8. page 8084, line 17: ". . . and the assumption that the normal vectors to the ice sheets upper and lower surfaces, $\mathbf{n} \in \mathbb{R}^3$, are nearly vertical:" I would say that this is not an additional assumption but rather a consequence of your initial shallowness assumption, as the gradients of your surfaces scale with $\mathcal{O}(\delta)$, as you correctly state in equation (1) in combination with the fact that $\mathbf{n}$ occurs in terms of $\mathcal{O}(\delta)$ or larger.

   If a *shallow* domain is defined as a domain obtained by a δ-scaling of the z–coordinates of a "regular" domain, then we agree with the referee that the gradient of its top/lower surface will scale with $O(\delta)$. However, in the paper we are simply considering a domain with small aspect ratio H/L, i.e. a thin domain, that in principle can have very rough surfaces' topographies with large slope gradients, so an additional assumption on the surfaces' slopes is needed. We refer to Dukowicz et al, *J. of Glaciology 2010*, where it is stated that FO model "is valid only when both the low-aspect-ratio and small-basal-slope assumptions are valid". Since an ice-sheet lower surface can be pretty rough we prefer to point out the assumption on $\mathbf{n}$ as it may not be always satisfied.

9. page 8085, line 5-14: I would suggest to introduce the strain-rate tensor by name.

   We have added the definition of the strain-rate tensor.

10. page 8086, line 5 and 7: It is rather the temperature relative to pressure melting point than the normal ice temperature that enters the Arrhenius factor, as you describe it. In ice sheets this is not negligible, as we are talking of a shift of about 0.87 K per kilometre ice thickness.

    We agree and have added this detail to the text.

11. page 8087, equation (16) and (17): Why are there curly brackets in front of these equations?

    We agree with the reviewer that the curly brackets were unnecessary. They have been removed.

12. page 8087, equation (17): if your z-coordinate is negative for values below sea level ($z = 0$), then the right-hand side should rather read as:

$\rho_w g \min(z, 0)\mathbf{n}$. If it needs a positive or negative sign depends on the orientation of your surface.

Thank you for pointing out this typographical error. We have fixed it in the revised manuscript.

13. page 8089, line 11: "Note that in our weak formulation Eq. (19), the source terms in Eq. (2) have not been integrated by parts." I do not get the point of this statement, as I would not see how the divergence theorem would apply to single directional derivatives, like $\partial s/\partial x$.

    In principle, one could replace $\int_\Omega \rho g \frac{\partial s}{\partial x}\phi_1 d\Omega$ with

$$-\int_\Omega \rho g s \frac{\partial \phi_1}{\partial x} d\Omega + \int_{\partial\Omega} s\phi_1 \mathbf{n}_1,$$

    using (a corollary of) divergence theorem. However, we agree with the referee that the sentence is misleading and not needed, therefore we removed it.

14. page 8090, line 15: ". . . , then splitting each prism into three tetrahedra (Fig. 17)." Out of curiosity: why do you not use wedge-type prisms but rather split them and by this give away the possibility to keep a low aspect ratio of the element?

    Mostly for historical reasons. We wanted to compare our results with another code (*LifeV*) that uses tetrahedra, and so we used the same elements/meshes. We are now exploring the possibility of using prisms, but we did not try yet with realistic geometries. As a side comment, we remark that a nice consequence of using linear FE on tetrahedra is that a one-point quadrature rule is enough to get exact integrals for the diffusive term (if the flow factor is constant on each element). We added a sentence in the paper mentioning the possibility of using wedge elements on prisms.

15. page 8092, line 9: You let the continuation parameter, α, pop up in the middle of this sentence, but the only place where it occurs else is the table showing the algorithm. This is a little bit confusing.

    To prevent confusion, we have added the phrase "(defined in Algorithm 1)" following the first appearance of α in the text.

16. page 8093: missing reference in the discussion of multi-grid methods in ice-sheet modeling: (Jouvet and Grser, 2013)

    We have added this reference.

17. page 8098, line 4, equation (23): As mentioned in my main points of critics, I do not think that the expression in brackets represents the 2-D version of the effective strain-rate for a FO problem.

    Our x–y 2D FO Stokes equations are intended to be the mathematical version of the 3D FO Stokes equations obtained by neglecting the $\partial/\partial y$ terms in the full equations. We agree that the physical interpretation of the terms becomes blurred with this definition. We have removed references to $\dot{\epsilon}_{e,2D}$ as the effective strain rate.

18. page 8100, line 2: I have issues understanding how there can be an x in the $3\pi x$ term which should be a derivative of (25) please verify.

    The reviewer is correct: there was a typographical error in equation (25). The $3\pi x$ term should be 3π. The error has been corrected in the revised manuscript. We thank the reviewer for calling the mistake to our attention.

19. page 8116, line 7: "In general, glaciers and ice sheets are modeled as an incompressible fluid in a low Reynolds number flow with a power-law viscous rheology, as described by the Stokes flow equations." In general you are right. Nevertheless (also in order to end up with an expansion of the equations with respect to the aspect ratio), in the context of creeping shallow flows one introduces scales for the typical stresses not in terms of a viscosity, but rather scaling with the hydrostatic pressure. Consequently, the resulting equations in ice sheet flow usually are presented as the double limit of a small aspect ration and Froude numbers (see Greve and Blatter, 2009).

    While we appreciate this insight from the reviewer, we've decided to leave this level of detail out of the current paper, which is why we currently refer to Dukowicz et al. (2010) and Schoof and Hindmarsh (2010) for additional discussion on how the first-order approximation is derived from the Stokes equations. If the editor thinks it necessary and appropriate, we would be happy to add an additional reference to Greve and Blatter (2009).

20. page 8138, Fig. 8: I would say that the figure in that form is not very informative and as well can be skipped.

    We agree, and have removed the figure from the paper.

21. page 8139, Fig. 9: Out of curiosity: Can you explain me, why there is a small, yet asymmetric deviation between two solvers solving a symmetric problem (mesh, partitioning, algorithm).

    The error is on the order of machine precision so truncation errors are probably the leading errors here. Part of the asymmetry could be generated by the fact that the algorithm used to split prisms into tetrahedra does not preserve symmetry. We realize that this figure may be misleading, therefore we prefer to remove it and just mention that with the two codes we get the same result up to machine precision.

22. page 8143, Fig. 13: I would skip that figure. The explanation in the text suffices.

    We agree, and have removed the figure from the paper.

23. page 8145, Fig. 15: Out of curiosity: Can you perhaps explain why the "full Newton" method does not diverge immediately (as I am used to see from my applications with full Stokes solutions), but starts to go out of the window only after a few iterations?

    Our answer is speculative, and should not be added to the publication. Our intuition is that the full Newton algorithm sees a nearly-infinite viscosity, which leads to no practical progress in the nonlinear solver, but also does not lead to divergence. In a full Stokes code, the deviatoric pressure is probably what diverges, trying to compensate for stresses weighed by the nearly-infinite viscosity, while our hydrostatic formulation does not have this term.

24. page 8148, Fig 18: I think it would be more informative to include the error between observation and the (I guess based on inverse basal friction coefficients) computed solution and perhaps explain the deviations (which there seem to be in the fast flowing areas).

    The deviations are likely due to the fact that in the optimization we also try to be in equilibrium with a given surface mass balance, which can be in competition with matching the surface velocity observations.

The inversion details and related discussion are in the referenced JGR paper by Perego, Price and Stadler. The focus of this section is mainly on demonstrating the capability of *Albany/FELIX* to efficiently solve ice-sheets steady problems and that is the reason why we used a "reasonably" realistic basal friction coefficient. The validation and discussion of the results of the inversion is out of the scope of the paper, therefore we prefer not to add additional details on this as it would distract the reader from the main focus of the paper. We added a sentence in the paper where we acknowledge the deviations and refer to the JGR paper for discussion.

4 References

We have added many but not all of these references. We feel the addition of all these references is not necessary. In our referencing, we tried to be representative, not exhaustive.

- Greve, R., and Blatter, H., Dynamics of Ice Sheets and Glaciers, Advances in Geophysical and Environmental Mechanics and Mathematics, Springer, 2009 Heimbach, P. and V. Bugnion., Greenland ice-sheet volume sensitivity to basal, surface and initial conditions derived from an adjoint model, Ann. Glaciol. 50 (52), 67-80, 2009

- Vizcano, M., U. Mikolajewicz, M. Grger, E. Maier-Reimer, G. Schurgers and A. M. E. Winguth, Long-term ice sheet-climate interactions under anthro- pogenic greenhouse forcing simulated with a complex Earth System Model, Clim. Dyn. 31 (6), 665-690, 2008

- Sato, T. and R. Greve, Sensitivity experiments for the Antarctic ice sheet with varied sub-ice-shelf melting rates, Ann. Glaciol. 53 (60), 221-228, 2012

- Pattyn, F.: Investigating the stability of subglacial lakes with a full Stokes ice- sheet model, J. Glaciol., 54, 353 - 361, doi:10.3189/002214308784886171, 2008

- Jay-Allemand M., F. Gillet-Chaulet, O. Gagliardini and M. Nodet, Investigat- ing changes in basal conditions of Variegated Glacier prior

to and during its 19821983 surge, The Cryosphere, 5, p. 659-672, 2011, doi:10.5194/tc-5-659- 2011

- Favier, L., G. Durand, S. L. Cornford, G. H. Gudmundsson, O. Gagliardini, F. Gillet-Chaulet, T. Zwinger, A. J. Payne and A. M. Le Brocq, Retreat of Pine Is- land Glacier controlled by marine ice-sheet instability, Nature Climate Change, 2014, doi:10.1038/nclimate2094

- Jouvet, G., C. Grser, An adaptive Newton multigrid method for a model of marine ice sheets, Journal of Computational Physics, Vol. 252, 2013

Response to Reviewer #2

March 13, 2015

1 General comments

Modular software libraries are used to solve partial differential equations (PDEs) modeling ice sheets. The libraries are from the Trilinos project for solution of linear and non-linear systems of equations, for discretization of PDEs with the finite element method, and for parallelization on thousands of cores. This is a good idea and extensions of the code to include inverse modeling and sensitivity estimation are possible using other parts of the libraries. The PDEs chosen here to model the ice are the Blatter-Pattyn equations. Under simplifying assumptions they can be derived from the full Stokes equations. The accuracy of the implementation is evaluated using manufactured solutions and comparisons with another code. The solution in the Greenland ice sheet is computed in parallel in a test with a realistic geometry. The parallel scalability for the whole code is good and the convergence is as expected by theory when the mesh is refined. The paper is worth publishing if the comments below are taken into account in a revised version.

We thank the reviewer for his/her thoughtful feedback, which has been addressed in the revised manuscript, and has helped us to improve our paper. All changes to the original manuscript are marked in blue in the revision. Responses to your comments can be found below, also in blue.

2 Specific comments

1. A major contribution in the paper is the use of software libraries to build the solver. Is it possible to estimate how much time and effort

that has been saved by using these libraries? Such an estimate could encourage also similar developments for many other applications.

The FO Stokes PDEs and relevant boundary conditions were implemented and verified in *Albany* (Sections 1-5 in the paper) in approximately 6 months, with one staff member working on this task approximately half-time (0.25 FTE, where FTE stands for "full-time equivalent"). It is estimated that all the work presented in the paper (including development of the AMG preconditioner based on semi-coarsening) took approximately 1.5 FTEs worth of work. These numbers have been noted in the paper, in Section 3.2. The fact that, with software libraries, the verification, extension and maintenance of the libraries is amortized over many projects by subject-matter experts, has saved us a lot of development time, and will continue to save us development time in the future.

2. For prediction of ice flow, the Stokes equation has to be coupled to an equation for the motion of the surface and for long time intervals also for the bedrock. This coupling and the moving mesh or the ice surface cutting through a stationary background mesh are not discussed.

   The moving mesh scenario would only be encountered in a transient simulation, which goes beyond the scope of this paper. We have added to the paper (the "Conclusions" section) some discussion of how it is possible to do prognostic (transient) simulations using the *CISM-Albany* and *MPAS-Albany* dycores, which we are developing but which are not discussed specifically in the paper. The former dycore has the ability to solve an equation for the motion of the bedrock for long time intervals; however, we are in general not concerned with this scenario as we plan to use the model mainly for decadal and century scale predictions, and we expect the bedrock motion to be a small factor over those time scales.

3. If a time dependent problem is considered, the difficulty with the nonlinear solver to find an initial guess is resolved by taking the solution from the previous time step (as remarked on p 15). Then the homotopy method may be needed only in the very first step.

   The reviewer is correct that for a transient simulation, the homotopy in general would only be needed in the first time step (diagnostic solve).

Although transient runs go beyond the scope of this paper, we had actually noted this in Section 3.1.1.

4. A better explanation of the choice of manufactured solutions in ch 4 is needed. Are the analytical solutions typical for the ice solutions in the interior and at the boundaries? For a fair evaluation, they should have some relation to what can be expected from the Stokes equations.

   Our MMS test cases in Chapter 4 were based on equations obtained by neglecting the $\partial/\partial z$ terms from the FO Stokes equations. The test cases were not intended to verify the 3D FO Stokes equations: rather, they were intended to be used as part of a multi-stage code verification that includes also verification of the 3D FO Stokes equations using code-to-code comparisons and mesh convergence studies on realistic geometries (Sections 5 and 6 in the paper). We have attempted to make this more clear in the paper, and also have made it clear that our MMS problems are for a 2D version of the FO Stokes equations, not the 3D equations.

   We feel despite being simplified, our MMS test cases are nonetheless useful. The task of deriving source terms for an MMS study for the 3D FO Stokes equations is cumbersome, if not intractable. In contrast, our MMS problems are simple enough to be implemented by anyone simply by referring to the expressions in our paper.

   We do agree with the reviewer that an MMS test case whose solution is related to what is expected from the 3D FO Stokes equations is worthwhile. To address this point, we have derived an MMS test case for the FO Stokes equations in the x–z plane (obtained by neglecting the v and $\partial/\partial y$ terms in the equations), and showed some mesh convergence results for our code on this test case (see Section 4.2). The analytic solution to this test case is the sum of shallow ice approximation (SIA) and shallow shelf approximation (SSA) analytic solutions (given the geometry and boundary conditions) and therefore physical in nature.

5. On p 28, line 6, the difference between the present code and another code for the FO equations is of O(1e-10) which is remarkably small considering that at least two different finite element discretizations are used (p 25). Is there an explanation to the small difference?

   The two codes used the same meshes and same basis function with the same discretization (FEM). Therefore it is expected that the difference

between solutions obtained in these codes is close to machine precision.

6. The resolution in the z-direction is studied in experiments in ch 6.2. How many layers to use depends on the required accuracy. One percent relative error in Fig 14 is probably enough (considering all the modeling and data errors). Then 10 layers should suffice. A graded or a uniform mesh does not seem to matter. An explanation could be that there is a boundary layer also at the ice surface which is not resolved by the graded mesh (see e.g. Schoof, C. and Hindmarsh, R.: Thin-film flows with wall slip: an asymptotic analysis of higher order glacier flow models, Quart. J. Mech. Appl. Math., 63, 73114, 2010). A relative accuracy less than 1e-4 can never be necessary.

   We realized after submitting the paper that the results from our z–convergence study reported originally did not give the complete picture. To give a complete picture, we have added results showing the effect of z–refinement compared to the effect of horizontal refinement (Tables 4–5). Previously, we were reporting only the last rows of these tables and the data there are somewhat misleading as the solutions are close to the reference solution. For problems with a coarser horizontal mesh resolution, there is some benefit in refining vertically over horizontally but, it appears, only up to a point. We have modified the discussion in Section 6 to explain this. Our recommendation given the new data is that it is more worthwhile to refine from a 1 km GIS mesh with 10 vertical layers to a 1 km GIS mesh with 20 vertical layers, over refining to a 500 m GIS mesh with 10 vertical layers (if one is forced to choose between refining vertically or horizontally only). We leave it up to the reader to select his/her desired level of accuracy from our data.

   We agree that there is not much value in using a graded spacing, but include those data for completeness.

7. Appendix A may be removed. The full Stokes equations can be found in many references.

   We have decided to keep Appendix A in the paper, to keep the paper self-contained.

Response to Editor

March 13, 2015

Dear authors,

In my role as Executive editor of GMD, I would like to bring to your attention our Editorial:

http://www.geoscientific-model-development.net/gmd_journal_white_paper.pdf

http://www.geosci-model-dev.net/6/1233/2013/gmd-6-1233-2013.html

This highlights some requirements of papers published in GMD, which is also available on the GMD website in the "Manuscript Types" section:

http://www.geoscientific-model-development.net/submission/manuscript_types.html

In particular, please note that for your paper, the following requirements have not been met in the Discussions paper please correct this in your revised submission to GMD.

"- The paper must be accompanied by the code, or means of accessing the code, for the purpose of peer-review. If the code is normally distributed in a way which could compromise the anonymity of the referees, then the code must be made available to the editor. The referee/editor is not required to review the code in any way, but they may do so if they so wish. "

"- All papers must include a section at the end of the paper entitled 'Code availability'. In this section, instructions for obtaining the code (e.g. from a supplement, or from a website) should be included; alternatively, contact information should be given where the code can be obtained on request, or the reasons why the code is not available should be clearly stated. "

" - All papers must include a model name and **version number** (or other unique identifier) in the title. "

We have added a section to the paper called "Code availability" where we give the location, as well as licensing and versioning information, of the *Trilinos* libraries and the *Albany* framework. The *Albany/FELIX* solver described in this paper is not publicly available at the present time. A public release of the code as part of *Albany* is planned for 2015.

Yours,

Dan Lunt

Response to Reviewer #3

March 13, 2015

1 Summary

The first order, or Blatter-Pattyn equations describing the diagnostic solution to ice flow are discretized using the finite element method and solved using modular and highly scalable software libraries. The libraries provide built-in facilities for UQ (uncertainty quantification) and optimization via AD (automatic differentiation). Similar models have been constructed in numerous recent publications (see references in this paper), and its completion is not novel enough to merit publication. However, several interesting applications of the model are performed to explore its numerical properties, making this an interesting paper. The applications are: 1) the homotopy, or a systematic method of reducing a viscosity regularization parameter which improves convergence, 2) the manufactured solutions which allow for verification of the conservation of momentum solutions, 3) application of a multi-level algebraic preconditioner, and 4) exploration of the vertical mesh spacing requirements of the model. Each of the 4 applications have been performed in some way by others, but I am not aware of another paper featuring as systematic and rigorous approach to model testing. Because of this, ice-sheet modelers should find the results interesting and worthwhile to compare their own models to.

We thank the reviewer for his/her thoughtful feedback, which has been addressed in the revised manuscript, and has helped us to improve our paper. All changes to the original manuscript are marked in blue in the revision. Responses to your comments (including those above) can be found below, also in blue.

2 Review

The utility of this paper is found in its applications of the new momentum solver. I will treat the novelty and utility of each of the applications, in turn.

Homotopy is cool, but it looks like the stopping value of the regularization parameter is close to the value used by many (10^{-10}), which makes it seem more like an additional burden faced by this model, rather than a feature that improves numerical performance, particularly in the numerous models that do not have (or perhaps fail to report on) difficulties reaching convergence. I might find the treatment more satisfying if the sequence of alphas had been continued to make the regularization exactly zero, assuring the choice of parameter does not influence results. I am not clear on how the sequence of values for alpha are selected, this should be clarified.

The sequence of values of α's were calculated using a natural continuation algorithm implemented in the *LOCA* package of *Trilinos*. Please see our response to your comment #1 in Section 3 below for more details.

AMG is nice, but somewhat lacking in terms of details of how it is implemented. The paper provides a good general overview of the method, as well as insight into the particular challenges posed by ice-sheet modeling; low aspect ratios and their impact semi-coarsening techniques. However, the discussion was not useful to a modeler interested in how AMG might be applied to a particular matrix generated by Stokes equations. This comes back to the editor's note about code release and versioning; for the material on AMG to be useful, it needs to be (minimally) accessible in the form of source code, and preferably accompanied with a better explanation. The authors do state that a second paper dealing with AMG is in preparation, and I think that some details can be delayed, but more implementation specifics should be provided now.

It was brought to our attention by the editor of *Geophysical Model Development*, as well as several reviewers, that all papers submitted to this journal must include a section entitled "Code availability" that contains instructions for obtaining the code described in the paper and licensing information. We have added such a section to the paper. In it are paths to the currently released versions of *Trilinos* and *Albany*. The AMG algorithm is available in the *ML* package of *Trilinos* (version 11.12 or later). An addendum to (Gee, 2014) describes how the multi-grid semi-coarsening algorithm is specified from a user's perspective (Tuminaro, 2014).

MMF Equations 22-23 are the shallow shelf approximation (SSA) equa-

tions, right? Where are the vertical shear terms (du/dz, dv/dz, see eqns 2-8)? Are these zero in the case where the surface slope is zero? That makes sense, but I wonder if the system has been simplified to the point where verifications is done on a specialized case of the FO equations (SSA) and not the true FO system? In this case, you've lost the most interesting part of the FO eqns; the ability to estimate BOTH vertical shear AND membrane stresses. This is probably the biggest issue with the paper as it is now written.

Equations (22)–(23) were obtained from the 3D FO Stokes equations by neglecting the $\partial/\partial z$ terms. This is why the vertical shear terms are missing from the equations. The test cases were not intended to verify the 3D FO Stokes equations: rather, they were intended to be used as part of a multi-stage code verification that includes also verification of the 3D FO Stokes equations using code-to-code comparisons and mesh convergence studies on realistic geometries (Sections 5 and 6 in the paper). We have attempted to make this more clear in the paper, and also have made it clear that our MMS problems are for a 2D version of the FO Stokes equations, not the 3D equations.

We feel despite being simplified, our MMS test cases are nonetheless useful. The task of deriving source terms for an MMS study for the 3D FO Stokes equations is cumbersome, if not intractable. In contrast, our MMS problems are simple enough to be implemented by anyone simply by referring to the expressions in our paper.

We do agree with the reviewer that a test case based on equations that include vertical shear would be worthwhile. To address this point, we have derived an MMS test case for the FO Stokes equations in the x–z plane (obtained by neglecting the v and $\partial/\partial y$ terms in the equations), and showed some mesh convergence results for our code on this test case. Please see Section 4.2 of the revised manuscript.

The exploration of the convergence for different mesh resolutions is sensible and provides clear results. It's a little difficult to relate reported errors to something practical, like; how many layers are needed to eliminate errors in prognostic runs of several hundreds of years. That is probably not an easy question to answer, but if the paper is going to have significant impact, making a clear statement about how important the errors are would be helpful.

3 Summary Statement

This is a long and important paper; providing a new means of solving a complex set of equations, and giving significantly expanded means of verifying the results are correct. There are some matters that have to be addressed before it is publishable. They are:

1. Minor changes in the section dealing with homotopy to explain the sequence of values that are used for alpha. It would be really nice to know if homotopy is needed in time dependent problems, but it seems beyond the scope of the problem. Could homotopy be used when moving from coarser to finer grid resolutions to see if it reduces Newton iterations?

   For time-dependent problems, we would recommend using the homotopy only in the first time step (diagnostic solve), since the converged solution at time t_n is likely to be a reasonable initial guess for Newton's method in time t_{n+1}, for $n \in \mathbb{N}$. We have noted this in Section 3.1.1, but do not provide numerical results, as time-dependent simulations are beyond the scope of this paper.

   We had stated in the paper that the sequence of α's are calculated using a step size control algorithm. We have added a sentence to add more information on this algorithm, and added a reference to the LOCA manual (Salinger et al., 2002) where the detailed algorithm is published. The Allgower reference is appropriate as well.

   Moving from coarser to finer meshes as a method to converge the nonlinear system on the finest mesh more rapidly (often called mesh sequencing) would be a valid approach, and could likely reduce computation time for convergence on the finest mesh. Homotopy would not be appropriate to step through mesh resolution because mesh size is not a continuous parameter. For this same reason, the mesh sequencing approach might be fragile in that there is no easy path to recovery from a failed step. Furthermore, this would need to be implemented with scripts that create and interpolate between meshes, a process that does not match HPC platform usage (batch queuing systems) very well. We have not implemented a mesh sequencing approach.

2. Major changes should be made to the section on AMG, providing better insight into the implementation details. This might be most easily done

by, Release source code, identify where it can be accessed, and provide versioning, as per the requirements of GMD.

We have added the "Code Availability" section describing how the codes can be obtained, in particular *Trilinos*, which contains the package where the AMG preconditioner is implemented, called *ML*. We have also added a reference to a document (Tuminaro, 2014) that describes how the multi-grid semi-coarsening algorithm is specified from a user's perspective (Section 3.1.2).

3. Reconsider the approach to manufactured solutions, which in the present form do not appear to be verifying the solution of the FO equations.

   We did reconsider the approach, as explained above. We have attempted to make it more clear that our MMS problems are for a 2D version of the FO Stokes equations, not the 3D equations. We have also extended our MMS test cases to include one for the x–z FO Stokes equations (obtained by neglecting the v and $\partial/\partial y$ terms in the equations). These provide a test in which the equations have vertical shear terms, and in which the solution is more physical (the sum of SIA and SSA solutions). The new MMS test case and some convergence results for this problem can be found in Section 4.2.

4. Provide a roadmap detailing how this advancement in a single component of an ice-sheet model will integrate with other components to be more comprehensive and useful. How will the energy balance be computed? What about ice transport? Integration with climate models? I understand that this effort is coming from the national laboratories, where there is significant activity on all of these components, but the authors need to demonstrate that this is destined to become more than 'just' another momentum solver with no capacity for prognostic ice sheet modeling.

   Please see our response to Comment 2 below.

All of these can be addressed in the context of minor revisions. Unless I misunderstand, the MMF will require serious reconsideration, but there is enough in this paper that without the MMF the paper is still interesting.

Specific, but less significant issues in the text.

Abstract:

- Why is "Template-Based Generic Programming" capitalized?

  We agree that it need not be capitalized, and have changed it in the paper.

Introduction:

1. "A primary development focus has been on improving the representation of the momentum balance equations over the 'shallow ice' (SIA; Hutter, 1983) and 'shallow-shelf' (SSA; Morland, 1987) approximations through the inclusion of membrane stresses over the entire model domain."

   Careful, membrane stresses are supported over the entire domain when using the SSA. Wouldn't it be more appropriate to say through inclusion of BOTH vertical shear AND membrane stresses over the entire model domain (CAPS just for emphasis). allowing for a quantifiably optimal match between modeled and observed velocities There is no guarantee that this match is globally optimal.

   We have made some changes to this portion of the text along the lines suggested by the reviewer in the first part of this comment. We do not follow entirely the second part of the comment about optimal matching between observed and modeled velocities. We do not make a claim that the match between these two quantities is globally optimal.

2. p 8083, line 1-5: There is mention of integration into ESMs. It is possibly true that the model can be easily integrated into ESMs, but if how that is done is not discussed in the paper, then it really shouldn't be mentioned in the introduction because the issues is complex enough that in a journal like GMD, such claims can not be made without some supporting documentation.

   How we plan to couple to ESMs and do prognostic model runs is a good question. The reviewer is correct that *Albany/FELIX* can only be used for the diagnostic stress-velocity solve, and does not discretize the ice temperature and thickness evolution equations. To do prognostic runs and couple our code to ESMs, we have coupled *Albany/FELIX* to two

other land ice models: the *CISM* (Community Ice Sheet Model) and the *MPAS-LI* (Model for Prediction Across Scales - Land Ice) codes. In the resulting dynamical cores (dycores), termed *CISM-Albany* and *MPAS-Albany* respectively, the steady state stress velocity solve occurs in *Albany/FELIX* and the ice sheet thermal and geometric evolution is calculated in *CISM* or *MPAS-LI*. The details of these ice sheet modeling frameworks will be presented in a subsequent paper, so we had intentionally omitted the discussion of these codes here. We agree with the reviewer that some mention of these dycores is worthwhile so the reader can see how we will do prognostic solves and couple to ESMs. It has been added to the "Conclusions" section of the paper. If the reviewer would like to learn more about *CISM-Albany* and *MPAS-Albany* and see some results from prognostic runs produced by these codes, there are some presentations on the subject available at: `http://www.scidac.gov/PISCEES/presentations.html`.

3. p 8087, line 5 not an energy balance but a conservation of energy model.

   We agree that "conservation of energy" is more appropriate of a term than "energy balance", and have changed the phrase.

4. p 8090, line 11 "horizonal" $\rightarrow$ "horizontal" x2

   This was a typographical error, which we have corrected.

5. p 8090, line 26 How is the domain decomposition performed?

   We partition in a uniform contiguous domain fashion a 2D cross-section of the full 3D mesh, so that in the 3D mesh decomposition, all elements having the same x and y coordinates are on the same processor (see Section 6.1). The partitioning was done using the decomposition utility (called `decomp`) available as a part of *Sandia Engineering Analysis Code Access System (SEACAS)* database of *Trilinos* (see the "Code Availability" section). This utility was used to create a linear decomposition of the 2D mesh. These details have been added to the text.

6. p 8091, line 5. Is the viscosity differentiated too, or just the strain rates? That is to say, its the so called "incomplete adjoint" (Goldberg and Sergienko 2010), or complete adjoint?

   In our implementation, the viscosity is differentiated, not just the strain rates. Since the problem is self-adjoint, the Jacobian is equivalent to

the adjoint matrix. We would like to note that this is only relevant for optimization problems, which go beyond the scope of this paper.

7. p 8093 line 21-23. OK, I don't really know much here, but I thought the basic idea of multigrid techniques was to use multiple resolutions to speed the rate of information transfer across the domain. It's more difficult for me to understand error capture from a high level.

   There are different views of how multigrid works. The basic principle is to utilize multiple resolution versions of the original problem to accelerate the iterative solution procedure. The key idea is that smooth error components (in the current solution approximation) can be efficiently damped by applying a simple iterative process to a coarse resolution version of the problem. This coarse version essentially facilitates the propagation of long range information across the domain. This discussion has been added to the paper for clarity (beginning of Section 3.1.2).

8. Eqn 26 contains the strain rates dot_epsilon_1 and dot_epsilon_2. These should be updated to clarify their 2D counterparts.

   We agree, and have added the definitions of the 2D versions of these quantities.

Manuscript prepared for Geoscientific Model Development (GMD)
with version 5.0 of the LaTeX class copernicus.cls.
Date: 17 March 2015

Albany/FELIX: a parallel, scalable and robust, finite element, first-order Stokes approximation ice sheet solver built for advanced analysis

Irina K. Tezaur[1], Mauro Perego[2], Andrew G. Salinger[2], Raymond S. Tuminaro[2], and Stephen F. Price[3]

[1]Quantitative Modeling and Analysis Department, Sandia National Laboratories, P.O. Box 969, MS 9159, Livermore, CA 94551, USA.
[2]Computational Mathematics Department, Sandia National Laboratories, P.O. Box 5800, MS 1320, Albuquerque, NM 87185, USA.
[3]Fluid Dynamics and Solid Mechanics Group, Los Alamos National Laboratory, P.O. Box 1663, MS B216, Los Alamos, NM, 87545, USA.

Correspondence to: Irina K. Tezaur, ikalash@sandia.gov, (248)470-9203.

Abstract. This paper describes a new parallel, scalable and robust finite-element based solver for the first-order Stokes momentum balance equations for ice flow. The solver, known as *Albany/FELIX*, is constructed using the component-based approach to building application codes, in which mature, modular libraries developed as a part of the *Trilinos* project are combined using abstract interfaces and template-based generic programming, resulting in a final code with access to dozens of algorithmic and advanced analysis capabilities. Following an overview of the relevant partial differential equations and boundary conditions, the numerical methods chosen to discretize the ice flow equations are described, along with their implementation. The results of several verification studies of the model accuracy are presented using: (1) new test cases for simplified 2D versions of the governing equations derived using the method of manufactured solutions, and (2) canonical ice sheet modeling benchmarks. Model accuracy and convergence with respect to mesh resolution is then studied on problems involving a realistic Greenland ice sheet geometry discretized using hexahedral and tetrahedral meshes. Also explored as a part of this study is the effect of vertical mesh resolution on the solution accuracy and solver performance. The robustness and scalability of our solver on these problems is demonstrated. Lastly, we show that good scalability can be achieved by preconditioning the iterative linear solver using a new algebraic multilevel preconditioner, constructed based on the idea of semi-coarsening.

1 Introduction

In its fourth assessment report (AR4), the Intergovernmental Panel on Climate Change (IPCC) declined to include estimates of future sea-level rise from ice sheet dynamics due to the inability of ice sheet models to mimic or explain observed dynamic behaviors, such as the acceleration and thinning then occurring on several of Greenland's large outlet glaciers (IPCC, 2007). Since the AR4, increased support from United States, United Kingdom, and European Union funding agencies has enabled concerted efforts towards improving the representation of ice dynamics in ice sheet models and towards their coupling to other components of Earth System Models (ESMs) (Little et al., 2007; Lipscomb et al., 2008; van der Veen et al., 2010). Thanks to this support, there has recently been tremendous progress in the development of "next generation" community-supported ice sheet models (e.g., Bueler and Brown (2009); Rutt et al. (2009); Larour et al. (2012b); Gagliardini et al. (2013); Brinkerhoff and Johnson (2013); Lipscomb et al. (2013)) able to perform realistic, high-resolution, continental scale simulations. These models run on high-performance, massively parallel high-performance computing (HPC) architectures using 10^2-10^4 processes and employ modern, well-supported solver libraries (e.g., *PETSC* (Balay et al., 2008) and *Trilinos* (Heroux et al., 2005)). A primary development focus has been on improving the representation of the momentum balance equations over the "shallow ice" (SIA; (Hutter, 1983)) and "shallow-shelf" (SSA; (Morland, 1987)) approximations through the inclusion of both vertical shear and membrane stresses over the entire model domain (e.g., Pattyn (2002)). These approaches include "hybrid" models (a combination of SIA and SSA (Bueler and Brown, 2009; Pollard and Deconto, 2009; Goldberg and Sergienko, 2011)), so-called "higher-order" models (Pattyn, 2003), "full" Stokes models (Larour et al., 2012b; Leng et al., 2012a; Gagliardini et al., 2013)), and combinations of a range of approximations up to and including full Stokes (Seroussi et al., 2012). By accounting for both vertical and horizontal stress gradients, the aforementioned models allow for more realistic and accurate simulations of outlet glaciers, ice streams, and ice shelves, as well as modeling of the transfer of perturbations from marginal to inland regions.

Other significant improvements in ice sheet modeling frameworks include the integration of unstructured (Larour et al., 2012b; Gagliardini et al., 2013; Brinkerhoff and Johnson, 2013) or adaptive meshes (Cornford et al., 2013), which allows the focusing of resolution and computational power in regions of dynamic complexity. Also becoming standard is the use of formal optimization and data assimilation techniques for generating realistic model initial conditions. Surface observations are used to infer poorly known ice properties or parameters, such as the friction coefficient at the ice-bedrock interface (e.g., Morlighem et al. (2010); Larour et al. (2012b); Gillet-Chaulet et al. (2012); Brinkerhoff and Johnson (2013)) or the rheology of floating ice shelves (Khazendar et al., 2009), allowing for a quantifiably "optimal" match between modeled and observed velocities. Recently, these approaches have been extended to simultaneously optimize both model parameter fields and uncertain initial condition fields, while also accounting for forcing from climate models in order

to minimize transient shocks when coupling to climate forcing (Perego et al., 2014). Other recent and noteworthy optimization improvements include the assimilation of time dependent observations (e.g., Goldberg and Heimbach (2013)) and the estimation of formal uncertainties for optimized parameter fields (Petra et al., 2014).

The latter capability – the characterization of parameter uncertainties – represents a critical first step towards formal uncertainty quantification ("UQ") of ice sheet model output quantities of interests, such as estimates of future sea-level rise. For this process to be computationally tractable during both the inverse (parameter estimation and uncertainty assignment) and forward propagation steps, it is critical to have robust, efficient, and scalable solves on HPC platforms (Isaac et al., 2014). This, in turn, requires advanced dynamical core capabilities, such as access to model derivatives (e.g., the Jacobian matrix), and advanced algorithms for the solution of the nonlinear and linear equations. These same requirements of robustness, efficiency, and scalability hold for the inclusion of ice sheet models as fully coupled components of large-scale, high-resolution ESMs.

In this paper, we introduce and focus on a new momentum balance solver for land ice simulations based on the first-order approximation of the nonlinear Stokes flow model for glaciers and ice sheets. This new solver, *Albany/FELIX* (Finite Elements for Land Ice eXperiments, described in more detail below), either already includes many of the capabilities discussed above or is designed to allow for their easy implementation at later stages of development. Here we present algorithms and software that lead to a robust nonlinear solution procedure (including the use of automatic differentiation (AD) technologies), scalable linear algebra, and the ability to use unstructured and highly refined grids.

The remainder of this paper is organized as follows. In Section 2, we describe in detail our mathematical model for glaciers and ice sheets, giving the relevant assumptions, partial differential equations, boundary conditions, and parameter values. Our numerical methods for discretizing this model and their implementation in *Albany/FELIX* are summarized in Section 3. In Section 4, which focuses on verification of the *Albany/FELIX* code using the method of manufactured solutions, two new test cases are derived for simplified 2D versions of the FO Stokes equations and used in a convergence verification study involving several types and orders of finite elements. In Section 5, further verification of the accuracy of solutions computed with our solver is performed using canonical ice sheet modeling test cases. The results of a mesh convergence study on a realistic Greenland ice sheet geometry are then discussed in Section 6. This study provides insight into the effects of the parallel domain decomposition on solver convergence, and the effect of the vertical mesh resolution on solution accuracy. We then describe our robust, nonlinear solver, which uses homotopy continuation with respect to the regularization parameter in the calculation of the ice effective viscosity. The solver's robustness and scalability is demonstrated on various Greenland ice sheet geometries, discretized using tetrahedral and hexahedral meshes. Finally, we show that improved scalability of our code can be achieved by preconditioning the iterative linear solver using an algebraic multi-

level preconditioner, constructed based on the idea of semi-coarsening. A concluding summary is offered in Section 7. Here, we also touch briefly on the larger ice sheet modeling frameworks that *Albany/FELIX* is being incorporated into for treating the conservation of mass and energy and for performing prognostic runs in both standalone mode and as coupled components of ESMs.

One objective of this paper is to introduce a new parallel, scalable and robust finite element first-order Stokes solver for ice flow, namely *Albany/FELIX*, to the land ice and climate modeling communities. The article also contains several new contributions to the field of ice sheet modeling, which are most notably:

- The derivation of several new test cases based on the method of manufactured solutions for simplified 2D forms of the first-order Stokes equations, which can be used to verify convergence to an exact solution for parts of the governing PDEs in any ice sheet code that discretizes these equations.
- The description of a homotopy continuation algorithm with respect to a regularization parameter in the ice effective viscosity expression, which greatly improves the robustness of a Newton nonlinear solver, especially in the absence of a good initial guess.
- Insights into the effects of the parallel decomposition and vertical mesh spacing on solver performance and solution accuracy for ice sheet simulations.
- A new algebraic multilevel preconditioner, constructed based on the idea of semi-coarsening and ideal for meshes structured in the vertical direction, that delivers a scalable linear solve when combined with a preconditioned iterative method.

2 First-order Stokes approximation mathematical model

We consider a power-law viscous, incompressible fluid in a low Reynolds number flow, described by the first-order approximation to the nonlinear Stokes flow equations for glaciers and ice sheets (Dukowicz et al., 2010; Schoof and Hindmarsh, 2010). The first-order (FO) approximation, also referred to as the "Blatter-Pattyn" model (Pattyn, 2003; Blatter, 1995), follows from assumptions of a small geometric aspect ratio, $\delta = H/L$ (where H and L are characteristic length scales for the vertical and horizontal dimensions, respectively, and $H \ll L$), and the assumption that the normal vectors to the ice sheet's upper and lower surfaces, $\mathbf{n} \in \mathbb{R}^3$, are nearly vertical:

$$\mathbf{n}^T \approx \Big(\mathcal{O}(\delta), \mathcal{O}(\delta), \pm 1 + \mathcal{O}(\delta^2) \Big). \tag{1}$$

Effectively, the FO approximation is derived by neglecting $\mathcal{O}(\delta^2)$ terms in the Stokes equations and respective boundary conditions (discussed in more detail in Appendix A). Numerical discretization of the FO Stokes equations gives rise to a much smaller discrete system than numerical discretization of the full Stokes equations. Moreover, discretization of the FO Stokes system gives rise to a "nice"

elliptic coercive problem, in contrast to the notoriously difficult saddle-point problem obtained when discretizing the full Stokes system.

Let u and v denote the x and y components of the ice velocity vector $\mathbf{u} \equiv \Big(u,\, v \Big)^T \in \mathbb{R}^2$, respectively. The FO approximation consists of the following system of partial differential equations (PDEs):

$$\begin{cases} -\nabla \cdot (2\mu \dot{\boldsymbol{\epsilon}}_1) + \rho g \frac{\partial s}{\partial x} = 0, \\ -\nabla \cdot (2\mu \dot{\boldsymbol{\epsilon}}_2) + \rho g \frac{\partial s}{\partial y} = 0, \end{cases} \tag{2}$$

where g denotes the gravitational acceleration, ρ denotes the ice density, and $s \equiv s(x, y)$ denotes the upper surface boundary:

$$\Gamma_s \equiv \{(x, y, z) \in \mathbb{R}^3 | z = s(x, y)\}. \tag{3}$$

In the most general, three-dimensional (3D) case of the FO approximation, the strain-rate tensor

$$\dot{\boldsymbol{\epsilon}} \equiv \Big(\dot{\boldsymbol{\epsilon}}_1,\, \dot{\boldsymbol{\epsilon}}_2 \Big) \in \mathbb{R}^{3\times 2}, \tag{4}$$

is given by the following components

$$\dot{\boldsymbol{\epsilon}}_1^T = \Big(2\dot{\epsilon}_{xx} + \dot{\epsilon}_{yy},\, \dot{\epsilon}_{xy},\, \dot{\epsilon}_{xz} \Big) \in \mathbb{R}^3, \tag{5}$$

and

$$\dot{\boldsymbol{\epsilon}}_2^T = \Big(\dot{\epsilon}_{xy},\, \dot{\epsilon}_{xx} + 2\dot{\epsilon}_{yy},\, \dot{\epsilon}_{yz} \Big) \in \mathbb{R}^3, \tag{6}$$

where

$$\dot{\epsilon}_{xx} = \frac{\partial u}{\partial x}, \quad \dot{\epsilon}_{yy} = \frac{\partial v}{\partial y}, \quad \dot{\epsilon}_{xy} = \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right), \quad \dot{\epsilon}_{xz} = \frac{1}{2}\frac{\partial u}{\partial z}, \quad \dot{\epsilon}_{yz} = \frac{1}{2}\frac{\partial v}{\partial z}. \tag{7}$$

The effective viscosity μ can be derived using Glen's flow law (Cuffey et al., 2010; Nye, 1957) as:

$$\mu = \frac{1}{2} A^{-\frac{1}{n}} \dot{\epsilon}_e^{\frac{1}{n}-1}, \tag{8}$$

where $\dot{\epsilon}_e$ is the effective strain rate, given by:

$$\dot{\epsilon}_e^2 \equiv \dot{\epsilon}_{xx}^2 + \dot{\epsilon}_{yy}^2 + \dot{\epsilon}_{xx}\dot{\epsilon}_{yy} + \dot{\epsilon}_{xy}^2 + \dot{\epsilon}_{xz}^2 + \dot{\epsilon}_{yz}^2. \tag{9}$$

In (8), A is the flow rate factor and n is the Glen's (power) law exponent, typically taken equal to 3 for ice sheets. Hence, μ (8) is a nonlinear expression, and the system (2) is a nonlinear, elliptic system of PDEs. The flow law rate factor A is strongly temperature-dependent, and can be described through the Arrhenius relation,

$$A(T) = A_0 \exp\left(-\frac{Q}{RT^*} \right), \tag{10}$$

where A_0 denotes a constant of proportionality, Q denotes the activation energy for ice creep, T^* denotes the ice temperature in Kelvin (K) corrected for the pressure melting point dependence, and R denotes the universal gas constant. For more details involving the relation between the flow factor and temperature (10), the reader is referred to Cuffey et al. (2010). For completeness, the expressions for the Cauchy stress tensor $\boldsymbol{\sigma}$ and the pressure p in the FO approximation are provided:

$$\boldsymbol{\sigma} = 2\mu \left(\dot{\boldsymbol{\epsilon}}_1, \dot{\boldsymbol{\epsilon}}_2, \mathbf{0} \right)^T - \rho g(s-z)\mathbf{I}, \qquad p = \rho g(s-z) - 2\mu(\dot{\epsilon}_{xx} + \dot{\epsilon}_{yy}), \tag{11}$$

where $\mathbf{0} = \left(0, 0, 0 \right)^T$ and $\mathbf{I}$ is the 3×3 identity tensor. The equations (2) are specified on a bounded 3D domain, denoted by Ω, with boundary

$$\Gamma \equiv \Gamma_s \cup \Gamma_b \cup \Gamma_l. \tag{12}$$

Here, Γ_s is the upper surface boundary (3), and

$$\Gamma_b = \{(x,y,z) \in \mathbb{R}^3 | z = b(x,y)\}, \tag{13}$$

$$\Gamma_l = \{(x,y,z) \in \mathbb{R}^3 | l(x,y) = 0\}, \tag{14}$$

are the lower and (vertical) lateral surface boundaries, respectively. The relevant boundary conditions on Γ are:

(a) A stress-free (homogeneous Neumann) boundary condition on the upper surface boundary

$$\dot{\boldsymbol{\epsilon}}_1 \cdot \mathbf{n} = \dot{\boldsymbol{\epsilon}}_2 \cdot \mathbf{n} = 0, \qquad \text{on } \Gamma_s. \tag{15}$$

(b) Either a no-slip or a sliding boundary condition on the lower surface:

$$\begin{cases} u = v = 0, & \text{on } \Gamma_0 \\ 2\mu\dot{\boldsymbol{\epsilon}}_1 \cdot \mathbf{n} + \beta u = 0, \;\; 2\mu\dot{\boldsymbol{\epsilon}}_2 \cdot \mathbf{n} + \beta v = 0, & \text{on } \Gamma_\beta, \end{cases} \tag{16}$$

where Γ_b is partitioned as $\Gamma_b = \Gamma_0 \cup \Gamma_\beta$ with $\Gamma_0 \cap \Gamma_\beta = \emptyset$, and $\beta \equiv \beta(x,y) \geq 0$ is the basal sliding coefficient. Note that we assume the partitioning of Γ_b is known *a priori*. In practice, this would be specified (through a conservation of energy equation) by locating regions of the bed for which the temperature is at the pressure melting point. It is often more practical to enforce a quasi-no-slip Robin boundary condition on Γ_0 by setting β to a large value and always using the equation on the second line of (16) (e.g., $\beta = 10^7$ kPa a m^{-1}).

(c) On the lateral boundaries, one of two boundary conditions is applied: either a kinematic (Dirichlet) boundary condition

$$u = u_l, \;\; v = v_l, \text{ on } \Gamma_l, \tag{17}$$

where u_l and v_l are prescribed values of the ice velocities on the lateral boundary, or a dynamic (Neumann) boundary condition

$$2\mu\dot{\boldsymbol{\epsilon}}_i \cdot \mathbf{n} - \rho g(s-z)\mathbf{n} = \rho_w g \min(z,0)\mathbf{n}, \text{ on } \Gamma_l, \tag{18}$$

for $i = 1, 2$, where ρ_w denotes the density of water. In (18), it has been assumed that the coordinate system has been oriented such that z is strictly elevation (that is, $z = 0$ at sea level and values of z increase for higher elevations) (MacAyeal et al., 1996). The boundary condition (18) is derived by assuming that the ice shelf is in hydrostatic equilibrium with the air/water that surrounds it and is often referred to as an "open-ocean" boundary condition, as it takes into account the pressure exerted on the ice shelf by neighboring ocean. For some canonical benchmark experiments performed here (see Section 5.1), periodic lateral boundary conditions are prescribed as well.

The values of the parameters that appear in the first-order Stokes equations and the boundary conditions described above and used herein are summarized in Table 1. From this point forward, the new first-order Stokes approximation momentum balance solver will be referred to "*Albany/FELIX*". In this code, the numerical discretization of (2) uses *Trilinos*, a suite of modular software libraries (described in detail in (Heroux et al., 2005)).

Table 1. Physical parameter values for first-order Stokes equations and boundary conditions*

Name	Value	Units	Description
A	10^{-4}	$\text{k}^{-(n+1)}\ \text{Pa}^{-n}\ \text{a}^{-1}$	Flow rate factor
n	3	–	Glen's flow law exponent
g	9.8	m s^{-2}	Gravitational constant
ρ	910	kg m^{-3}	Ice density
ρ_w	1025	kg m^{-3}	Ocean water density
R	8.314	$\text{J K}^{-1}\ \text{mol}^{-1}$	Universal gas constant
A_0	$\begin{cases} 1.30 \times 10^7, & \text{if } T < 263 \text{ K}, \\ 6.22 \times 10^{22}, & \text{if } T \geq 263 \text{ K} \end{cases}$	$\text{k}^{-(n+1)}\ \text{Pa}^{-n}\ \text{s}^{-1}$	Arrhenius constant of proportionality
Q	$\begin{cases} 6.00 \times 10^4, & \text{if } T < 263 \text{ K}, \\ 1.39 \times 10^5, & \text{if } T \geq 263 \text{ K}, \end{cases}$	J mol^{-1}	Activation energy for ice creep

*The symbol k in the table denotes km/m, i.e., k=km/m= 10^3.

3 Numerical discretization and implementation

The model described in Section 2 is discretized and solved using a collection of algorithms and software implementations that were selected for accuracy, flexibility, robustness, and scalability. The following brief discussion of the methods presumes prior knowledge of Galerkin finite element approaches and Newton-Krylov based nonlinear solvers (Strang and Fix, 1973; Pawlowski et al., 2006).

3.1 Numerical methods

The PDEs for the FO Stokes model defined by (2) and the associated boundary conditions are discretized using the classical Galerkin finite element method (FEM) (Hughes, 2000).

Let $\mathcal{V}$ denote the Hilbert space given by:

$$\mathcal{V} \equiv \mathcal{V}(\Omega) = \left\{ \phi \in H^1(\Omega) : \phi|_{\Gamma_0} = 0 \right\}, \tag{19}$$

where $H^1(\Omega)$ denotes the space of square-integrable functions whose first derivatives are also square integrable. Following classical Galerkin FEM methodology, the weak form of the problem is obtained by projecting each of the equations in (2) onto a test function in $\mathcal{V}$ (19) in the continuous L^2 inner product and integrating the second order terms by parts. Toward this effect, the weak formulation of (2), for grounded ice, reads: find $u, v \in \mathcal{V}$ such that

$$\begin{cases} \int_\Omega 2\mu\dot{\boldsymbol{\epsilon}}_1(u,v) \cdot \nabla\phi_1 d\Omega + \int_{\Gamma_\beta} \beta u \phi_1 d\Gamma + \int_\Omega \rho g \frac{\partial s}{\partial x} \phi_1 d\Omega = 0, \\ \int_\Omega 2\mu\dot{\boldsymbol{\epsilon}}_2(u,v) \cdot \nabla\phi_2 d\Omega + \int_{\Gamma_\beta} \beta v \phi_2 d\Gamma + \int_\Omega \rho g \frac{\partial s}{\partial y} \phi_2 d\Omega = 0, \end{cases} \tag{20}$$

for all $\phi_1, \phi_2 \in \mathcal{V}(\Omega)$. The surface integral along the boundary appearing in (20) arises from integration by parts of the stress term in the variational form of the PDEs. This approach leads to a weak enforcement of the basal surface boundary condition (16) for the tangential stress, and straightforward implementation of the basal boundary conditions as an integrated boundary condition. (We believe, but have not rigorously shown, that the Gelerkin finite element approach for implementing the basal surface boundary condition enables one to circumvent robustness issues stemming from the discretization that were previously seen in our work with a finite difference discretization (Lemieux, 2011).)

Letting $\mathcal{F}(u, v; \phi_1, \phi_2)$ denote the operator defining the left hand side of (20), the problem defined by (20) is equivalent to finding the roots $u, v \in \mathcal{V}$ of the following nonlinear equation:

$$\mathcal{F}(u, v; \phi_1, \phi_2) = 0, \qquad \forall \phi_1, \phi_2 \in \mathcal{V}. \tag{21}$$

Equation (21) is an infinite-dimensional problem; a finite-dimensional analog of (21) is obtained by replacing the infinite-dimensional space $\mathcal{V}$ by a finite-dimensional finite element space, $\mathcal{V}_h$, where h is a length scale associated with a triangulation of the domain Ω into a set of disjoint finite elements Ω_e ($\Omega = \cup_{e=1}^{n_{el}} \Omega_e$, where $n_{el} \in \mathbb{N}$ is the number of finite elements in the triangulation).

Our implementation (a detailed discussion of which is given in Section 3.2) allows for tetrahedral (with either trilinear or triquadratic basis functions) or hexahedral elements (with bilinear or biquadratic basis functions) for 3D problems. One reason a finite element approach was selected was for its flexibility in using unstructured grids with non-uniform mesh density to increase the resolution in areas of large velocity gradients, such as in the vicinity of outlet glaciers, while retaining relatively coarse meshes in the more static interior regions. In this paper, we present results on three different types of grids:

(i) Structured uniform hexahedral grids,

(ii) Unstructured uniform tetrahedral grids,

(iii) Unstructured non-uniform tetrahedral grids.

The structured hexahedral meshes are generated by creating a uniform quadrilateral grid of a two-dimensional (2D) horizontal cross-section of a geometry Ω, and extruding it in a uniform fashion as hexahedra in the vertical direction. Similarly, the uniform tetrahedral meshes are created by meshing a 2D horizontal cross-section of Ω using a uniform triangular mesh, extruding it in the vertical direction as prisms, then splitting each prism into three tetrahedra (Figure 14)[1]. For the unstructured tetrahedral grids, an unstructured Delaunay triangle mesh of a 2D cross-section of Ω is generated based on some kind of refinement criteria (e.g., a static refinement based on the gradient of the velocity) using a meshing software (e.g., *Triangle*, a Delaunay triangulation mesh (Shewchuk et al., 1996)), and extruded in the vertical direction in the same way as a structured triangular grid. More details on these meshes are provided in Sections 5 and 6. Note that although all the meshes employed for the ice sheet application considered here were extruded (structured) in the vertical direction, our code base allows for completely unstructured grids.

A domain decomposition approach is used to compute the solution to the discretized nonlinear problem on distributed memory parallel computers. As a pre-processing step, the elements of the mesh are partitioned into one contiguous domain per processor to provide nearly equal work per processor. To do the partitioning, we used the decomposition utility (called `decomp`) available as a part of *Sandia Engineering Analysis Code Access System (SEACAS)* database of *Trilinos* to create a linear decomposition of the 2D mesh. Additional discussion of the parallel decompositions employed can be found in Section 6.

The result of the discretization process is a large, sparse system of nonlinear algebraic equations for the two components of horizontal velocity at the nodes of the mesh (the discrete counterpart of (21)). Our approach to solving this fully-coupled, nonlinear system is Newton's method. An analytic Jacobian matrix is computed at each iteration of Newton's method using automatic differentiation (AD). The integration of AD into the Albany code base, both for Jacobians and for parameter derivatives for sensitivity analysis and UQ, has been a significant advantage of developing a new model in this framework. The matrix is stored in sparse form, with rows of the matrix distributed across the processors of the machine.

The resulting linear system is solved using a preconditioned iterative method. For the largest problems, we use multilevel preconditioning (described in Section 3.1.2) to achieve scalability, while incomplete LU (ILU) additive Schwartz preconditioners work well for modest problem sizes and processor counts. Since the model is symmetric, the Conjugate Gradient (CG) iterative linear solver is employed.

[1] Another possibility, which we have not fully explored yet, is to use wedge elements on prisms.

Because of the singularity in the viscosity formulation for stress-free solutions, such as when computing the nonlinear solution from a trivial initial guess, the Newton iteration does not reliably converge. To achieve a robust nonlinear solution procedure, we formulated and implemented a homotopy continuation approach that steps to the final solution by solving a series of nonlinear problems that reliably converge. The details of this algorithm are given in Section 3.1.1.

3.1.1 Homotopy continuation algorithm

Although the stress tensor $\boldsymbol{\sigma}$ (11) is well-defined for any differentiable function $\mathbf{u}$, the Glen's law effective viscosity (8) is not defined when $\mathbf{u}$ is a rigid movement or exactly 0 (because n is typically taken to be greater than 1; see e.g., Schoof (2010); Chen et al. (2013)). This can pose a problem for nonlinear solvers as the initial guess for $\mathbf{u}$ is often taken as uniform or 0. To circumvent this difficulty, a regularization parameter $\gamma > 0$, $\gamma \ll 1$ is added to the sum of the strain rates in the effective strain rate term of the effective viscosity (8), yielding what we refer to as μ_γ:

$$\mu_\gamma = \frac{1}{2} A^{-\frac{1}{n}} \left(\dot{\epsilon}_e^2 + \gamma\right)^{\left(\frac{1}{2n} - \frac{1}{2}\right)}, \qquad \text{where } \lim_{\gamma \to 0} \mu_\gamma = \mu. \tag{22}$$

One common practice is to define $\mu = \mu_\gamma$ in (8) using some small, fixed value for γ, e.g., $\gamma = 10^{-10}$. Here, noting that the nonlinear solver often struggles to converge initially when using Newton's method, we use a variable γ as the continuation parameter in a homotopy method (Algorithm 1). In this approach, a sequence of problems (2) is solved for a sequence of effective viscosities $\{\mu_{\gamma_i}\}$ for $i = 1, 2, ...$, with $0 < \gamma_{i+1} < \gamma_i$, until γ reaches its target value. We use a natural continuation procedure, where the final solution at one value of the continuation parameter α (defined in Algorithm 1) is used as the initial guess for the subsequent nonlinear problem. The continuation algorithm has adaptive step size control, and will backtrack and attempt a smaller parameter step if the nonlinear solve at some step fails to converge (Allgower et al., 2003). The step size increase is in part based on the number of Newton iterations that were required to converge the previous step, so a relatively easy nonlinear solve requiring just a handful of Newton iterations will lead to a more aggressive parameter step (see Salinger et al. (2002) for the detailed algorithm). We have found that starting with $\alpha_0 = 0$ leads to a system that will reliably converge from a trivial initial guess, that an initial step size of 0.1 is a good initial step, and that $\alpha_\infty = 1$ provides an adequate stopping value.

In general, the homotopy continuation approach leads to many fewer nonlinear solves than when the regularization parameter γ in (22) is fixed to some small value, e.g., $\gamma = 10^{-10}$, especially for problems where a "good" initial guess for Newton's method is unavailable. Moreover, with the homotopy continuation approach, it is found that a full step can often be employed in the Newton's method line search algorithm, without the need for backtracking (i.e., iteratively reducing the step size in the line search algorithm).

We note that the homotopy continuation approach is in general effective when the initial guess is *not* close to the solution (in which case μ_γ is very small). Similarly, a good initial guess for $\mathbf{u}$

Algorithm 1 Homotopy continuation on regularization parameter γ in μ_γ

Set $\alpha = \alpha_0$, $\mathbf{u}^0 = \mathbf{u}_0$ and $i = 0$.

while $\alpha \leq \alpha_\infty$ **do**

Set $\gamma = 10^{-10\alpha}$ and define μ_γ by the formula (22).

Set $\mu = \mu_\gamma$ in (8).

Set $i = i + 1$.

Solve (2) with initial guess $\mathbf{u}^{i-1}$ using Newton's method, to obtain $\mathbf{u}^i$.

Increase α using a homotopy continuation method (e.g., natural continuation).

end while

may not be a good initial guess when using continuation because the initial viscosity μ_{γ_0} for the continuation algorithm is generally far from the real viscosity μ. When solving transient problems, it may be better to simply use a standard Newton method (without homotopy continuation), taking the solution at the previous time step as the initial guess, and using homotopy continuation only if the Newton solver has difficulties converging. A different approach, which may be used as an alternative to homotopy continuation, is to perform a few iterations using the Picard method and then switch to the Newton method once the nonlinear iterations starts to converge (e.g., (Leng et al., 2014)). The robustness and efficiency of the Newton solver with the homotopy continuation approach summarized in Algorithm 1 is studied numerically in Section 6.1.1.

3.1.2 Multilevel preconditioning

Multigrid preconditioners are among the most efficient and scalable linear solution techniques for resolving matrix equations associated with elliptic operators. The basic idea is to utilize multiple resolution versions of the original problem to accelerate the iterative solution procedure. Toward this effect, smooth error components (in the current solution approximation) can be efficiently damped by applying a simple iterative process to a coarse resolution version of the problem. This coarse version essentially facilitates the propagation of long range information across the domain. Oscillatory components are effectively reduced through a simple iterative procedure, while smooth components are tackled using auxiliary lower resolution versions of the problem. Different geometric multigrid methods have been successfully applied to the linear systems arising from ice sheet modeling simulations, e.g., Brown et al. (2013); Cornford et al. (2013); Isaac et al. (2014).

For our capability, we prefer algebraic multigrid (AMG) methods due to the potentially unstructured nature of the mesh in the horizontal plane. AMG methods have the advantage that the lower resolution versions of the multigrid hierarchy are constructed automatically using only the matrix coefficient entries. Unfortunately, solution of the underlying linear systems is problematic due to the strong anisotropic nature of the discrete equations. This is essentially a consequence of the disparate scales in the horizontal and vertical directions and the associated large mesh aspect ratios. At the

discrete level, these aspect ratios give rise to matrices where entries representing vertical coupling are generally much larger than entries representing horizontal coupling. Anisotropic phenomena within ice sheets and fairly different types of multigrid methods have been considered in recent prior works (Brown et al., 2013; Isaac et al., 2014; Jouvet et al., 2013).

From a multigrid perspective, reducing oscillatory errors in the horizontal direction is much more difficult than in the vertical direction. Further, accurately capturing horizontal coupling on coarse levels can be challenging due to the relatively small size of the corresponding matrix entries (which are effectively averaged to generate the low resolution versions). To avoid these difficulties, we have developed a hybrid structure/unstructured AMG multigrid capability that leverages the fact that our meshes, though unstructured in the horizontal plane, are structured in the vertical direction. That is, our 3D meshes can be viewed as extrusions of unstructured 2D meshes, allowing for varying vertical mesh spacing. A paper is in preparation to further describe the details of this hybrid algorithm. Here, we briefly describe its essence.

The basic concept behind the hybrid structured/unstructured AMG method is to first apply operator dependent multigrid semi-coarsening to initially coarsen the mesh and construct the first few levels of the multigrid hierarchy. Semi-coarsening and operator dependent multigrid both have a long history on structured grid problems (Dendy et al., 2010; Schaffer, 1998; Brown et al., 2000). Semi-coarsening refers to only coarsening in some subset of coordinate directions and is often advocated to address anisotropic problems. Essentially, one only coarsens in directions where oscillatory errors are easily reduced. Operator dependent multigrid refers to family of algorithms that intimately take advantage of structure. They can be viewed as idealized or "perfect" grid transfers for one dimensional simplifications of the higher dimensional problem. In this way, several coarse level meshes are effectively constructed, each containing the same number of points within all horizontal planes. When it is no longer possible to further coarsen vertically (as there is just a single horizontal layer), a standard smoothed aggregation AMG method is applied to this horizontal problem creating additional levels in the hierarchy. Thus, finer levels of the hierarchy are created via semi-coarsening and operator dependent multigrid (leveraging grid structure). Coarser levels are constructed via AMG, which is applied after the anisotropic behavior is no longer present (as there is just a single horizontal layer). To complete this brief description, we note that line Jacobi is used as the simple iterative scheme to damp oscillatory errors on the finer levels. It allows for aggressive semi-coarsening (i.e., reduction factors greater than three in the linear system dimension as one proceeds to progressively coarser levels). Polynomial smoothing is used on the levels associated with standard AMG.

The algebraic multilevel preconditioner described above has been implemented in and is available through the (open-source) *ML* package of *Trilinos* (Heroux et al., 2005), in *Trilinos 11.12* or later (see the "Code Availability" section at the end of this paper). The linear solver can be employed with or without the *Albany* and *Albany/FELIX* codes used to perform the ice sheet simulations described

herein. The general ML User's guide[2] Gee et al. (2007) contains a detailed description of how to exercise the multigrid solver. Numerous example applications are included in the *Trilinos* release demonstrating how the multigrid solver can be used in different situations. An addendum to (Gee et al., 2007) explaining how to invoke the particular software feature used in this paper (Tuminaro, 2014) describes how the multigrid semi-coarsening algorithm is specified from a user perspective. A paper is underway describing more algorithm details (Tuminaro et al., 2015).

3.2 Software implementation

The numerical methods described above are implemented in the *Albany* code base, an open-source[3], multi-physics code/analysis package developed at Sandia National Laboratories. A full description of *Albany* can be found in a separate publication (Salinger et al., 2014). Briefly, *Albany* is a finite element code base for the solution and analysis of models of coupled PDEs using parallel, unstructured-grid, implicit algorithms. It makes use of numerous computational mathematics libraries from the *Trilinos* suite (Heroux et al., 2005), and has been previously used in other applications domains such as quantum device modeling (Gao et al., 2013) and computational mechanics (Sun et al., 2013).

The software stack in *Albany* involves dozens of libraries that are delivered through *Trilinos* as independent software packages developed by small teams of domain experts. The *Sierra ToolKit* (*STK*) package is used for mesh database structures and mesh I/O. The *Epetra* package is used for distributed memory, parallel data structures for vectors and sparse matrices, which greatly simplify parallel operations such as halo exchanges for synchronizing data between processors. The *Intrepid* (Bochev et al., 2012) package provides flexible finite element discretization algorithms and general integration kernels. The PDE equations are described by a set of evaluation kernels, whose evaluation is managed by the *Phalanx* package.

One of the main distinguishing characteristics of the *Albany* code base is the use of the template-based generic programming (TBGP) approach (Pawlowski et al., 2012a,b). With this methodology, all that is required to implement a new set of physics in *Albany* is to code the residual of the PDE equations. Given this residual, *Albany* automatically computes and assembles the sparse Jacobian matrix and sensitivity vectors without any additional code development. TBGP makes extensive use of the *Sacado* package (Phipps et al., 2012) for automatic differentiation, which employs C++ expression templates with operator overloading, and has been closely integrated with the *Phalanx* and *Intrepid* packages.

The Newton-based nonlinear system solver and homotopy continuation algorithm are implemented in the *NOX* (Pawlowski et al., 2006) and *LOCA* (Salinger et al., 2005) packages, respectively. These solvers can additionally perform sensitivity analysis using the analytic sensitivity vectors computed with automatic differentiation with respect to model parameters. Within the solvers, we have

[2] Available online at http://www.trilinos.org/oldsite/packages/ml/mlguide5.pdf.

[3] The *Albany* framework can be obtained from its public `github` repository by the interested reader: https://github.com/gahansen/Albany.

full runtime access to all the *Trilinos* preconditioners (ILU and algebraic multilevel preconditioners, from the *Ifpack* and *ML* software packages, respectively) and linear solvers by specification in an input file. For the bulk of the computations in this paper, the *ML* package was employed for algebraic multilevel preconditioners (Tuminaro, 2014), and the *Belos* package was employed for iterative solvers (CG or GMRES) (Bavier et al., 2012).

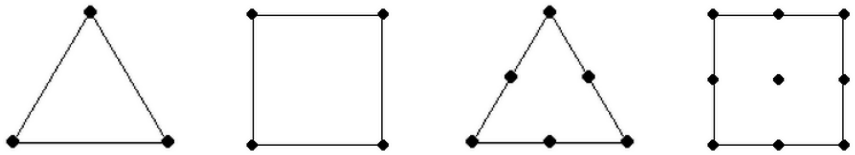
Albany is also coupled to the *Dakota* framework (Adams et al., 2013) of sampling-based optimization and UQ algorithms, which will play a significant role in model initialization, calibration, and projections. Although the application of optimization and UQ algorithms go beyond the scope of this paper, we emphasize that the component-based approach for building this application code leads to the rapid incorporation of many sophisticated capabilities.

To give the reader an idea of how much time can be saved in writing a solver using modular packages or libraries, it is noted that it took one staff member working half-time for approximately six months to write the *Albany/FELIX* solver and to verify the code on the test cases presented in Sections 4–5. It is estimated that all the work presented in the paper (including development of the AMG preconditioner based on semi-coarsening, described in Section 3.1.2) took approximately 1.5 FTEs (full-time equivalent units) worth of work.

4 Verification using the method of manufactured solutions (MMS)

We first conduct formal verification of the new *Albany/FELIX* code described in Section 3 through the method of manufactured solutions (MMS), using test cases derived here explicitly for this purpose. A survey of the literature reveals that past work has focused on deriving MMS benchmarks for the "shallow ice" and nonlinear Stokes models (e.g., **?**, respectively) rather than the FO approximation (2). The lack of MMS solutions for the FO Stokes equations in the literature is likely due to the complexity of these equations, which makes deriving source terms for a given manufactured solution difficult, if not, intractable. Here, we derive some new MMS benchmarks for simplified versions of the FO Stokes equations (2) in 2D. These equations are obtained by neglecting gradients in one of the coordinate directions, first z (Section 4.1), then y (Section 4.2), and allow us to look at the convergence of our computed solution to an exact solution for parts of the governing PDEs. The terms appearing in our test cases are simple enough to be implemented by anyone simply by referring to the expressions in this paper. It is emphasized that the test cases are intended to be used as part of a multi-stage code verification that includes also verification of the 3D FO Stokes equations using code-to-code comparisons and mesh convergence studies on realistic geometries (Sections 5 and 6, respectively).

Here, we use the *Albany/FELIX* code and these new MMS benchmarks to verify (i) that the dynamics have been implemented correctly, and (ii) that the type of finite elements employed show convergence at their expected theoretical rates.



(a) (b) (c) (d)

Fig. 1. 2D finite elements evaluated in the manufactured solution test cases. (a) Tri 3, (b) Quad 4, (c) Tri 6, (d) Quad 9

We consider four different finite element types in our numerical convergence study: three node triangles (denoted by "Tri 3"), four node quadrilaterals (denoted by "Quad 4"), six node triangles (denoted by "Tri 6"), and nine node quadrilaterals (denoted by "Quad 9") (Figure 1). Convergence is evaluated in the discrete l^2 norm. In particular, the relative error in a computed solution, denoted by $\mathcal{E}_{rel}^{disc}$, is calculated from

$$\mathcal{E}_{rel}^{disc} = \frac{||\mathbf{u}_n - \mathbf{u}||_2}{||\mathbf{u}||_2}, \tag{23}$$

where $||\cdot||_2$ denotes the discrete l^2 norm, $\mathbf{u}^T \equiv \Big(u,\, v \Big)$ is the exact solution to (24), and $\mathbf{u}_n$ is the numerically computed solution to (24). It is well-known from classical finite element theory (Hughes, 2000) that the theoretical convergence rate in the norm considered is two for the Tri 3 and Quad 4 elements, and three for the Quad 6 and Quad 9 elements. Hence, the first two elements are referred to as first-order finite elements and the second two elements are referred to as second-order finite elements. Note that the quadrilateral elements are expected to deliver a more accurate solution than their triangular counterparts of the same order.

4.1 x–y MMS test case

Consider the FO Stokes equations (2) in 2D, that is, (2) with all the $\frac{\partial}{\partial z}$ terms neglected. Assume these equations are posed on a domain whose sides are aligned with the x– and y– axes in a Cartesian reference frame, so that $\frac{\partial s}{\partial x} = \frac{\partial s}{\partial y} = 0$. Let $\mathbf{f}^T \equiv \Big(f_1,\, f_2 \Big)$ be a source term for the equations (2), to be determined such that a given manufactured solution satisfies these equations. Under these assumptions, the FO Stokes system (2) has the following form:

$$\begin{cases} -\frac{\partial}{\partial x}\left(4\mu_{2D,xy}\frac{\partial u}{\partial x} + 2\mu_{2D,xy}\frac{\partial v}{\partial y}\right) - \frac{\partial}{\partial y}\left(\mu_{2D,xy}\frac{\partial u}{\partial y} + \mu_{2D,xy}\frac{\partial v}{\partial x}\right) + f_1 = 0, \\ -\frac{\partial}{\partial x}\left(\mu_{2D,xy}\frac{\partial u}{\partial x} + \mu_{2D,xy}\frac{\partial v}{\partial y}\right) - \frac{\partial}{\partial y}\left(2\mu_{2D,xy}\frac{\partial u}{\partial x} + 4\mu_{2D,xy}\frac{\partial v}{\partial y}\right) + f_2 = 0, \end{cases} \tag{24}$$

where the viscosity $\mu_{2D,xy}$ is given by the 2D version of (8):

$$\mu_{2D,xy} = \frac{1}{2}A^{-\frac{1}{n}}\left\{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \frac{\partial u}{\partial x}\frac{\partial v}{\partial y} + \frac{1}{4}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2\right\}^{\left(\frac{1}{2n}-\frac{1}{2}\right)}. \tag{25}$$

We note that the x–y FO Stokes equations (24) can be viewed as a test for ice shelves, stress gradients in the x–z plane are negligible compared to those in the x–y plane.

The x–y MMS first test case is posed on a box domain, namely $\Omega = (0,1)\times(0,1)$ with Robin boundary conditions on $\partial\Omega$. The source term in (24) is derived such that the exact solution to this system is given by the following expression:

$$\begin{aligned} u &= e^x \sin(2\pi y), \\ v &= e^x \cos(2\pi y). \end{aligned} \tag{26}$$

(Figure 2). Substituting (26) into (24), the source terms f_1 and f_2 are obtained:

$$f_1 = 2\mu_{2D,xy}e^x\sin(2\pi y)\left[2-3\pi-2\pi^2\right] + A^{-\frac{1}{n}}\left(\frac{1}{n}-1\right)\dot{\epsilon}_{e,2D}^{\frac{1}{n}-2}\left(\frac{\partial\dot{\epsilon}_{e,2D}}{\partial x}(2\dot{\epsilon}_{xx}+\dot{\epsilon}_{yy}) + \frac{\partial\dot{\epsilon}_{e,2D}}{\partial y}\dot{\epsilon}_{xy}\right), \tag{27}$$

$$f_2 = 2\mu_{2D,xy}e^x\cos(2\pi y)\left[3\pi+\frac{1}{2}-8\pi^2\right] + A^{-\frac{1}{n}}\left(\frac{1}{n}-1\right)\dot{\epsilon}_{e,2D}^{\frac{1}{n}-2}\left(\frac{\partial\dot{\epsilon}_{e,2D}}{\partial x}\dot{\epsilon}_{xy} + \frac{\partial\dot{\epsilon}_{e,2D}}{\partial y}(\dot{\epsilon}_{xx}+2\dot{\epsilon}_{yy})\right), \tag{28}$$

where

$$\begin{aligned} \dot{\epsilon}_{e,2D} &\equiv \sqrt{\left(\tfrac{\partial u}{\partial x}\right)^2 + \left(\tfrac{\partial v}{\partial y}\right)^2 + \tfrac{\partial u}{\partial x}\tfrac{\partial v}{\partial y} + \tfrac{1}{4}\left(\tfrac{\partial u}{\partial y}+\tfrac{\partial v}{\partial x}\right)^2} \\ &= e^x\sqrt{(1+4\pi^2-2\pi)\sin^2(2\pi y) + \tfrac{1}{4}(2\pi+1)^2\cos^2(2\pi y)}, \end{aligned} \tag{29}$$

and $\mu_{2D,xy}$ is given by (25). The solution (26) implies the following boundary conditions on the boundary of Ω:

$$\begin{aligned} &\dot{\boldsymbol{\epsilon}}_1\cdot\mathbf{n} = 2(\pi-1)u, \quad \dot{\boldsymbol{\epsilon}}_2\cdot\mathbf{n} = -\left(\pi+\tfrac{1}{2}\right)v, && \text{at } x=0, \\ &\dot{\boldsymbol{\epsilon}}_1\cdot\mathbf{n} = -2(\pi-1)u, \; \dot{\boldsymbol{\epsilon}}_2\cdot\mathbf{n} = \left(\pi+\tfrac{1}{2}\right)v, && \text{at } x=1, \\ &u = 0, \qquad \dot{\epsilon}_2\cdot\mathbf{n} = 0, && \text{at } y=0 \text{ and } y=1, \\ &v = 0, && \text{at } (x,y)=(0,0), \end{aligned} \tag{30}$$

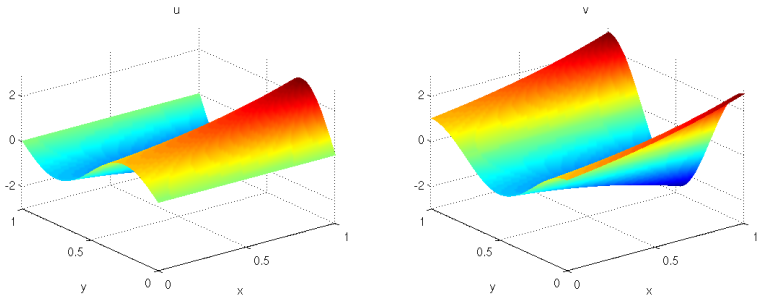
where $\mathbf{n}$ denotes the outward unit normal vector to a given boundary and where

$$\dot{\boldsymbol{\epsilon}}_1^T = \left(2\dot{\epsilon}_{xx}+\dot{\epsilon}_{yy},\, \dot{\epsilon}_{xy}\right) \in \mathbb{R}^2, \tag{31}$$

and

$$\dot{\boldsymbol{\epsilon}}_2^T = \left(\dot{\epsilon}_{xy},\, \dot{\epsilon}_{xx}+2\dot{\epsilon}_{yy}\right) \in \mathbb{R}^2. \tag{32}$$

The last condition on (30) is imposed to guarantee uniqueness of the v component of the velocity vector.



(a) (b)

Fig. 2. Plots of exact solutions to the x–y MMS test case: (a) u, (b) v

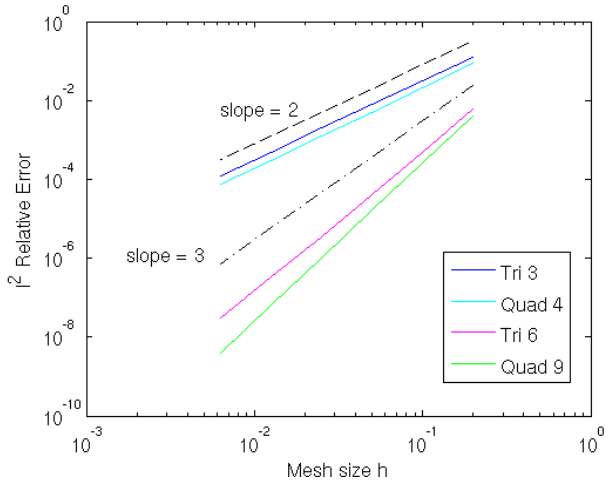


Fig. 3. Convergence rates for x–y MMS test case in the discrete l^2 norm (23)

For the x–y MMS test case considered here, the values of the flow rate factor and Glen's flow law exponent were taken to be $A = 1$ and $n = 3$, respectively. The relative errors (23) as a function of the mesh size h for the x–y MMS test case are plotted on a log-log plot in Figure 3. The two lowest-order finite elements (Tri 3 and Quad 4) converge at their theoretical rates of two, whereas the higher-order finite elements (Tri 6 and Quad 9) exhibit a slight superconvergece over their theoretical convergence rate of three. As expected, the quadrilateral elements deliver a more accurate solution than their triangular counterparts.

4.2 x–z MMS test case

The 2D FO Stokes equations in the $x-z$ variables are obtained from (2) by neglecting the y-component of the velocity (v) and all the $\frac{\partial}{\partial y}$ terms. The vector $\dot{\boldsymbol{\epsilon}}_1^T$ reads

$$\dot{\boldsymbol{\epsilon}}_1^T = \left(2\dot{\epsilon}_{xx},\, \dot{\epsilon}_{xz}\right), \tag{33}$$

and the FO Stokes equations reduce to the following 2D equation in the $x-z$ plane:

$$-\frac{\partial}{\partial x}\left(4\mu_{2D,xz}\frac{\partial u}{\partial x}\right) - \frac{\partial}{\partial z}\left(\mu_{2D,xz}\frac{\partial u}{\partial z}\right) + f_1 = 0, \tag{34}$$

where

$$\mu_{2D,xz} = \frac{1}{2}A^{-1/n}\left(\dot{\epsilon}_{xx}^2 + \dot{\epsilon}_{xz}^2\right)^{\frac{1}{2n}-\frac{1}{2}}, \qquad f_1 = \rho g\frac{\partial s}{\partial x}. \tag{35}$$

We consider the following approximate solution of (34):

$$u = \frac{2A\rho^n g^n}{n+1}\left((s-z)^{n+1} - H^{n+1}\right)\left|\frac{\partial s}{\partial x}\right|^{n-1}\frac{\partial s}{\partial x} - \frac{\rho g}{\beta}H\frac{\partial s}{\partial x}. \tag{36}$$

The first term of u is the solution of the SIA with no-slip at the bedrock interface, whereas the second term is the solution of the SSA equation when H and β are constant and s is quadratic in x. We now use the solution u as our manufactured solution, and we modify the forcing term f_1 of (34) so that the FO equation is exactly satisfied by u. In particular, we set $n=3$, and consider the geometry defined by $s = s_0 - \alpha x^2$, $b = s - H$, $x \in (-L, L)$, with constants s_0, α, H, β. Then,

$$f_1 = \frac{16}{3}A\mu_{2D,xz}^4\left(-2\phi_4^2\phi_5 + 24\phi_3\phi_4(\phi_1 + 2\alpha x^2) - 6x^3\phi_1^3\phi_2\phi_3 - 18x^2\phi_1^2\phi_2\phi_4^2 - 6x\phi_1\phi_3\phi_5\right), \tag{37}$$

where

$$\begin{aligned}
&\phi_1 = z - s, \quad \phi_2 = 4A\alpha^3\rho^3 g^3 x, \quad \phi_3 = 4x^3\phi_1^5\phi_2^2,\\
&\phi_4 = 8\alpha x^3\phi_1^3\phi_2 - \tfrac{2H\alpha\rho g}{\beta} + 3x\phi_2(\phi_1^4 - H^4),\\
&\phi_5 = 56\alpha x^2\phi_1^3\phi_2 + 48\alpha^2 x^4\phi_1^2\phi_2 + 6\phi_2(\phi_1^4 - H^4),\\
&\mu_{2D,xz} = \tfrac{1}{2}(A\phi_4^2 + Ax\phi_1\phi_3)^{-\frac{1}{3}}.
\end{aligned} \tag{38}$$

Boundary conditions are of the Robin and Neumann type, and are given by

$$\begin{aligned}
\dot{\boldsymbol{\epsilon}}_1\cdot\mathbf{n} &= f_{top}, \quad \text{at } z = s(x),\\
\dot{\boldsymbol{\epsilon}}_1\cdot\mathbf{n} + \beta u &= f_{bed}, \quad \text{at } z = b(x),\\
\dot{\boldsymbol{\epsilon}}_1\cdot\mathbf{n} &= f_{lat}, \quad \text{at } x = L,\\
\dot{\boldsymbol{\epsilon}}_1\cdot\mathbf{n} &= -f_{lat}, \quad \text{at } x = -L,
\end{aligned} \tag{39}$$

where

$$\begin{aligned}
f_{lat} &= -4\phi_4\mu_{2D,xz},\\
f_{top} &= -4\phi_4\mu_{2D,xz}n_x^{top} - 4\phi_2 x^2\phi_1^3\mu_{2D,xz}n_y^{top},\\
f_{bed} &= -4\phi_4\mu_{2D,xz}n_x^{bed} - 4\phi_2 x^2\phi_1^3\mu_{2D,xz}n_y^{bed} + 2H\alpha\rho g x - \beta x^2\phi_2(\phi_1^4 - H^4).
\end{aligned} \tag{40}$$

Here, the components of the normal to the top and bedrock surfaces read $n_x^{top} = \frac{2\alpha x}{\sqrt{4\alpha^2 x^2+1}}$, $n_z^{top} = \frac{1}{\sqrt{4\alpha^2 x^2+1}}$ and $n_x^{top} = -n_x^{bed}$, $n_z^{top} = -n_z^{bed}$.

Figure 4 shows a contour plot of the exact solution to the x–z MMS problem (36) and the domain Ω on this which problem is posed.

Reasonable values for the constants defining the x–z MMS test case, and the ones used here, are: $L = 50$ km, $s_0 = 2$ km, $H = 1$ km, $\alpha = 4e-5$ km^{-1}, $\beta = 1$ kPa yr/m and $A = 10^{-4}$ k^{-4}Pa^{-3}a^{-1}.

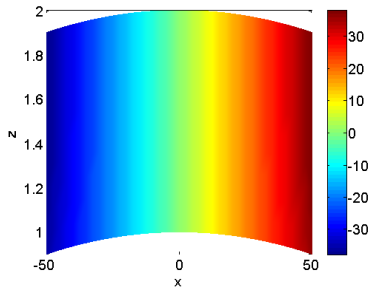


Fig. 4. Contour plot of exact solution to the x–z MMS test case

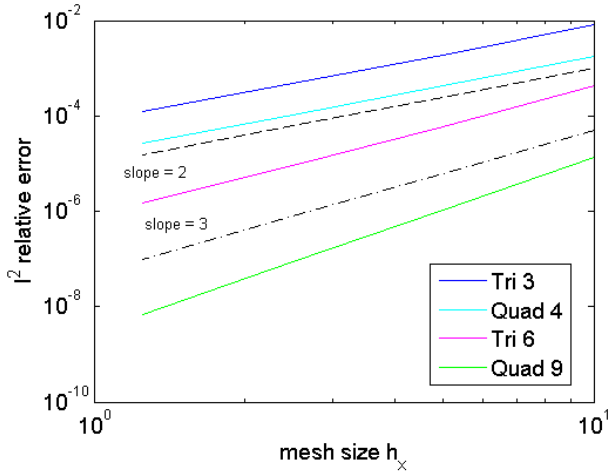


Fig. 5. Convergence rates for x–z MMS test case in the discrete l^2 norm (23)

Figure 5 plots the relative errors (23) on a log-log scale as a function of the horizontal mesh resolution h_x for the x–z MMS test case. The x and z resolutions considered are such that $n = 5$, 10, 20, and 40, where n denotes the number of elements in each spatial direction. The two first order elements, Tri 3 and Quad 4, converge at a rate of two, their theoretical convergence rate. The convergence rate of the Tri 6 element is close to its theoretical convergence rate of three. The Quad

9 element exhibits a slight superconvergence overs its theoretical convergence rate of three. As for the x–y MMS test case considered in Section 4.1 and as expected, the quadrilateral elements deliver a more accurate solution than their triangular analogs.

5 Intercomparison with other codes and benchmarks

In this section we discuss further (informal) verification of results for *Albany/FELIX* using some canonical ice sheet benchmarks, namely the ISMIP-HOM tests A and C (Section 5.1), and the confined shelf test case (Section 5.2) (Rommelaere, 1996). For these problems, the exact solution is not known in closed analytic form and our quasi-verification consists of code-to-code comparisons between the solution computed in *Albany/FELIX*, the results from other models participating in the original benchmark experiments, and the FO approximation, finite element code of (Perego et al., 2012).

The values of the physical parameters used in the two test cases considered are summarized in Table 1. We note that the units employed in our implementation are m a^{-1} for the ice velocities u and v (where "a" denotes years) and km for the length scale (e.g., the mesh dimensions). Our units are the same as in (Perego et al., 2012) but differ from other implementations, which often use a length scale of meters (m). Our units give rise to matrices with smaller differences in scale (which may be better scaled), as there is in general a smaller difference in scale in the relevant parameter values (e.g., $A = 10^{-4}$ k$^{-(n+1)}$ Pa^{-n} a^{-1} when the mesh is in km versus $A = 10^{-16}$ Pa^{-n} a^{-1} when the mesh is in m, where k=km/m=10^3).

5.1 ISMIP-HOM benchmarks

The ISMIP-HOM test cases (Pattyn et al., 2008) are a canonical set of benchmark experiments for so-called "higher-order" ice sheet models. Here, we consider tests A and C, both of which are specified on a horizontal, periodic domain with a unit length of L km. The bedrock surface, Γ_b, is given by a continuous function $z = b(x,y) \in \mathbb{R}^2$ and the upper surface, Γ_s, is given by a continuous function $z = s(x,y) \in \mathbb{R}^2$. The geometries are generated from a uniform hexahedral mesh of the unit cube $(0,1)^3 \in \mathbb{R}^3$ via the following transformation:

$$x = LX, \qquad y = LY, \qquad z = s(x,y)Z + b(x,y)(1-Z), \tag{41}$$

where X, Y, Z are the coordinates of the unit cube (in km), and $L \in \mathbb{N}$ is given. That is, a uniform mesh of $n_x \times n_y \times n_z$ elements is first generated of $(0,1)^3$, to yield the nodal coordinates X, Y, and Z, then the transformation (41) is applied. The following domain sizes are considered: $L = 5, 10, 20, 40, 80$ and 160 km. Each domain is discretized using an $80 \times 80 \times 20$ mesh of hexahedral elements. As a part of the quasi-verification, the *Albany/FELIX* solution is compared with the solution computed in the finite element code of (Perego et al., 2012) at the upper surface along the line $y = L/4$. Table 2 shows the relative difference between the *Albany/FELIX* and (Perego et

al., 2012) solutions in the l^2 norm along this line, calculated from the formula (23) with the (Perego et al., 2012) solution taken as the reference solution. Differences in the solutions are likely due to the different finite elements used: trilinear finite elements on hexahedra are used in *Albany/FELIX*, whereas linear finite elements on tetrahedra are used in the code of (Perego et al., 2012).

5.1.1 ISMIP-HOM test A

The first ISMIP-HOM benchmark considered is test A. For this problem, the upper ice surface boundary (Γ_s) is given by the following linear function

$$s(x,y) = -x\tan\alpha, \tag{42}$$

and the bedrock boundary (Γ_b) is given by the following trigonometric function

$$b(x,y) = s(x,y) - 1 + \frac{1}{2}\sin\left(\frac{2\pi}{L}x\right)\sin\left(\frac{2\pi}{L}y\right), \tag{43}$$

with $\alpha = 0.5^\circ$. The geometry is thus that of a uniformly sloping slab along the x coordinate direction with a doubly periodic, "egg crate" shaped bed. A no-slip boundary condition is prescribed on Γ_b (with $\Gamma_0 \equiv \Gamma_b$ and $\Gamma_\beta = \emptyset$), stress-free boundary conditions are prescribed on the upper surface Γ_s, and periodic boundary conditions are prescribed on the lateral boundaries Γ_l.

Figure 6 compares the solution computed within the *Albany/FELIX* code for ISMIP-HOM test A with the solution computed by the code of (Perego et al., 2012) (denoted by MP12 in this figure). The agreement between the two is excellent. The second column of Table 2 reports the relative difference between these two solutions in the l^2 norm (23). The relative difference is at most 0.1% for $L = 180$ and on the order of 0.001% for $L = 5, 10, 20, 40$.

Figure 6 also includes the mean and standard deviation of solutions computed by other models participating in the original set of benchmark experiments. For a detailed description of these models the reader is referred to (Pattyn et al., 2008). For all values of L considered, the *Albany/FELIX* solution is within one standard deviation of the mean of the other FO models considered in the original set of experiments. In Figure 6, the solutions labeled "Full Stokes" were calculated using the (more expensive but more physically realistic) full Stokes model for ice sheet flow (detailed in Appendix A). Comparing a FO Stokes solution to the full Stokes solution reveals how well the FO Stokes physics approximate the full Stokes model. The reader can observe by examining Figure 6 that agreement between the FO Stokes and the full Stokes solutions improves with increasing L.

5.1.2 ISMIP-HOM test C

For ISMIP-HOM test C, the upper and bedrock surfaces (Γ_s and Γ_b, respectively) are given by the following linear functions:

$$s(x,y) = -x\tan\alpha, \qquad b(x,y) = s(x,y) - 1, \tag{44}$$

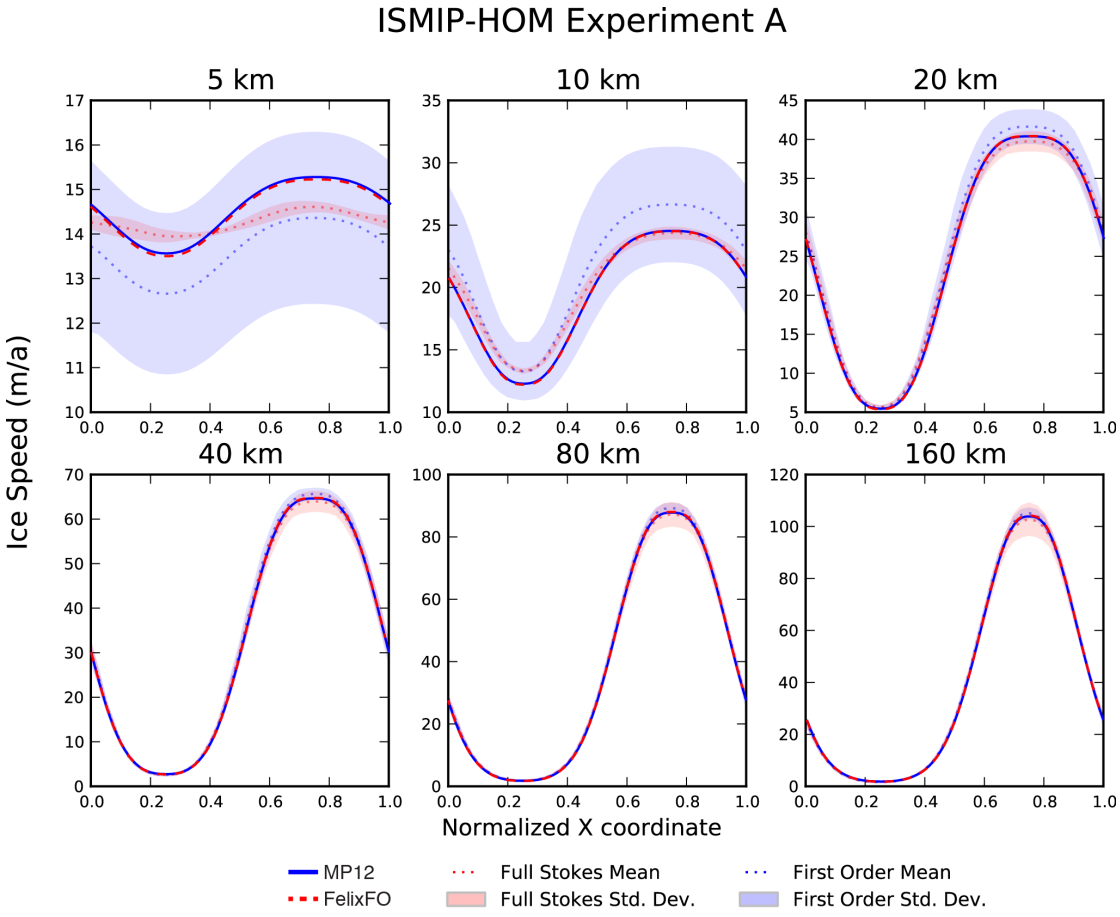


Fig. 6. ISMIP-HOM test A: surface velocity component u as a function of x at $y = L/4$ for each L considered. The blue solid line (MP12) represents results from (Perego et al., 2012) and the red-dashed line (labeled FelixFO) represents results from the current solver.

with $\alpha = 0.1°$. In addition to having a different geometry than test A, test C also differs in the boundary conditions. Unlike test A, sliding boundary conditions are prescribed on the bedrock ($\Gamma_\beta \equiv \Gamma_b$ and $\Gamma_0 \equiv \emptyset$), with the basal sliding coefficient given by

$$\beta(x,y) = 1 + \sin\left(\frac{2\pi}{L}x\right)\sin\left(\frac{2\pi}{L}y\right). \tag{45}$$

The boundary conditions at the upper and lateral boundaries (Γ_s and Γ_l respectively) are the same as for test A, namely stress-free and periodic, respectively. The geometry is thus that of a constant thickness, uniformly sloping slab along the x coordinate direction with a doubly periodic, "egg crate" spatial pattern for the basal friction parameter β.

The test case solution computed in *Albany/FELIX* is shown in Figure 7, along with the solution

computed using the solver of (Perego et al., 2012). For every L considered, the relative difference between *Albany/FELIX* and the solver of (Perego et al., 2012) (denoted, as before, by MP12 in Figure 7) is less than 1% (Table 2). Moreover, as for ISMIP-HOM test A, the *Albany/FELIX* solution is within one standard deviation of the model means for each value of L. As for ISMIP-HOM test A, Figure 7 illustrates also how well the FO Stokes model compares to the (more expensive but more accurate) full Stokes model. As for test A, the two models agree better for larger L.

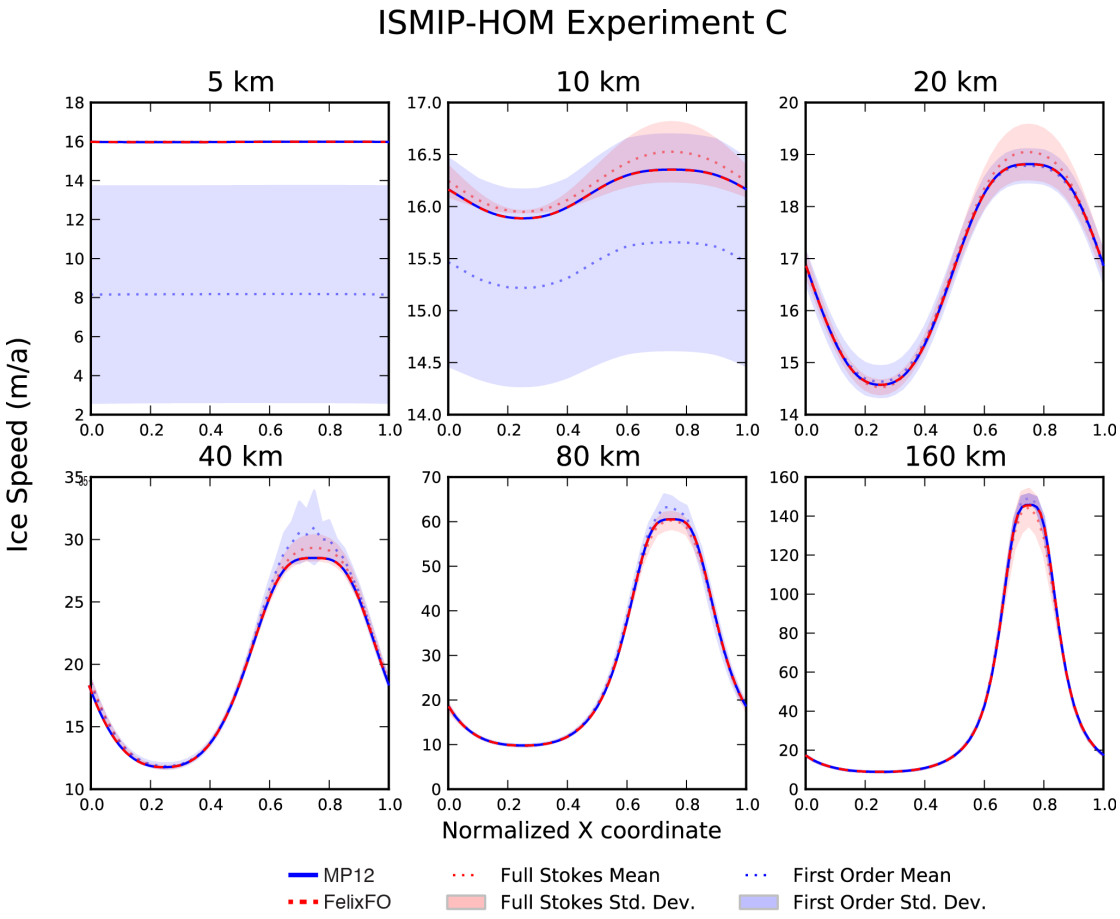


Fig. 7. ISMIP-HOM test C: surface velocity component u as a function of x at $y = L/4$ for each L considered. The blue solid line (MP12) represents results from (Perego et al., 2012) and the red-dashed line (labeled FelixFO) represents results from the current solver. Note that for the 5 km test, the MP12 and FelixFO results directly overly the results for the full Stokes models participating in the original intercomparison.

Table 2. Relative differences between *Albany/FELIX* and (Perego et al., 2012) solutions for ISMIP-HOM tests A and C

L (km)	Test A	Test C
5	0.00735%	0.386%
10	0.00629%	0.248%
20	0.00132%	0.176%
40	0.00408%	0.213%
80	0.0407%	0.277%
160	0.127%	0.320%

5.2 Confined shelf benchmark

We next consider an idealized ice shelf test case, referred to here as the "confined shelf" test case, which is a slightly modified version of test 3 from the Ice Shelf Model Intercomparison exercise (Rommelaere, 1996). The geometry is that of a 500 m thick slab of ice with equal extents of 200 km along the x– and y–dimensions, floating in hydrostatic equilibrium. A stress-free boundary condition is applied at the upper and basal boundaries ($z = s$ and $z = b$ respectively) and homogeneous Dirichlet boundary conditions ($u = v = 0$) are applied on three of the four lateral boundaries (the east $x = 200$, west $x = 0$ and north $y = 200$ boundaries). The south ($y = 0$) lateral boundary is open to the ocean and subject to the open ocean Neumann boundary condition described in Section 2 (boundary condition (c)). The values of the parameters that appear in (18) can be found in Table 1.

The confined shelf geometry is discretized using a structured tetrahedral mesh of 41×41 nodes in the $x - y$ plane with 10 vertical levels. As with the ISMIP-HOM test cases, the solution for the confined shelf test case computed in our code, *Albany/FELIX*, is compared to the solution computed by the solver of (Perego et al., 2012) on the same mesh. Figure 8 shows the solution calculated in *Albany/FELIX*, which is visually identical to the solution computed by the solver of (Perego et al., 2012). The difference between the *Albany/FELIX* and (Perego et al., 2012) solutions was found to be on the order of $\mathcal{O}(10^{-10})$ at all grid points.

6 Convergence study using realistic geometry

The final results presented herein are the results of a numerical convergence and performance study using a realistic, 1 km spatial resolution Greenland Ice Sheet (GIS) geometry (i.e., surface and bed topography from (Bamber et al., 2013)).

First, we present results from a 3D mesh convergence study in which a set of uniform quadrilateral meshes of different horizontal and vertical resolutions were considered. We began by generating

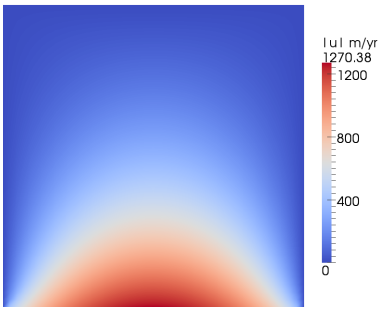


Fig. 8. *Albany/FELIX* solution to confined-shelf test case (indistinguishable from solution obtained by the solver of Perego et al. (2012)).

a quadrilateral mesh having an 8 km horizontal resolution. We then refined this coarse mesh uniformly in the horizontal direction (by splitting each quadrilateral finite element into four smaller quadrilaterals) four times to yield meshes with resolutions of 4 km, 2 km, 1 km and 500 m. The horizontal meshes were then extruded into 3D hexahedral meshes having uniform or graded spacing between the vertical layers. In the graded vertical spacing case, a transformation is performed such that a mesh having n_z vertical layers is finer near the bedrock boundary Γ_b and becomes progressively coarser moving up, towards the surface boundary Γ_s. The formulas[4] for the coordinate of the i^{th} vertical layer, z_i (for $i = 0, ..., n_z$, where n_z is the number of vertical layers), for each of these two spacings is given in Table 3. The number of layers considered in our study ranges from 5 to

Table 3. Formulas for different vertical mesh-spacing strategies (uniform vs. graded), for $i = 0, ..., n_z$.

z–spacing	z_i
Uniform	$\frac{i}{n_z}$
Graded	$1 - \frac{4}{3}\left[1 - \left(\frac{n_z}{2n_z - i}\right)^2\right]$

80. Realistic basal friction coefficient (β) fields were calculated by solving a deterministic inversion problem that minimizes simultaneously the discrepancy between modeled and observed surface velocities, modeled and observed bed topography, and between a specified surface mass balance field and the modeled flux divergence (see Perego et al. (2014) for more details). A realistic, 3D temperature field, originally calculated using *CISM* for the study in Shannon et al. (2013), was included

[4]The formula for the graded z–spacing is available in the *CISM* documentation, available at http://oceans11.lanl.gov/cism.

as an initial condition in order to provide realistic values for the flow-law rate factor (10). Prior to being interpolated onto the meshes at hand, the original topography, surface height, basal friction and temperature data were smoothed by convolution with a 2D Gaussian filter (having a standard deviation of 5 km). This smoothing filter reduces the small-scale variations of the original fields, so that it is reasonable to consider meshes from 8 km to 500 m for our convergence study. Using directly the non-smoothed data, we would have needed to consider much finer meshes in order to obtain asymptotic convergence.

The purpose of our GIS mesh convergence study is three-fold:

(i) To show a theoretical convergence rate for the finite elements evaluated with respect to refinement in all three coordinate directions.

(ii) To determine in a rigorous fashion for a GIS problem with a fixed horizontal mesh resolution how many vertical layers are required to achieve a solution having a desired accuracy,

(iii) To investigate whether the performance of our linear and nonlinear solvers changes with the number of vertical layers.

From finite element theory, theoretical convergence rates are expected for a problem in which the data is fixed on all meshes considered, so better-resolved data are intentionally not introduced on the coarser meshes that were part of our convergence study in this section. A high-resolution GIS problem, with real, high-resolution data is considered in Section 6.1.3.

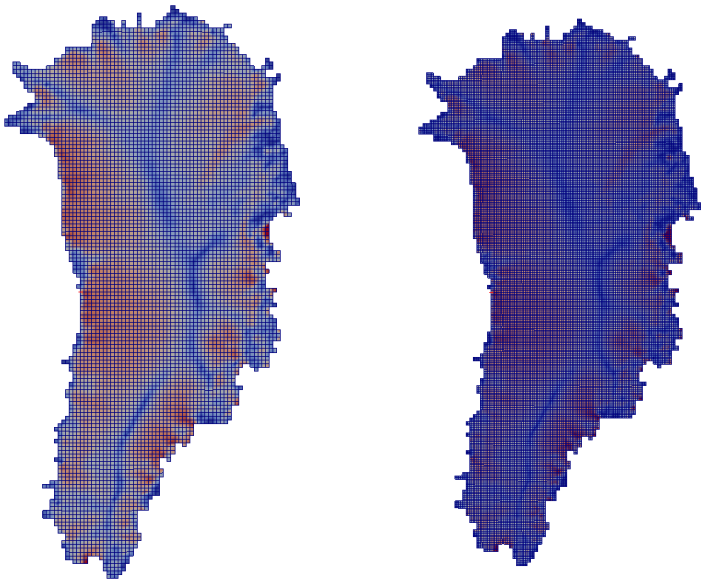
The FO equations (2) with basal sliding at the bedrock (16) and stress-free boundary conditions (15) on the remaining boundaries were solved on the base 8 km resolution mesh and the four successively refined meshes. Model runs were performed in parallel on *Titan*[5], a Cray XK6 operated by the Oak Ridge Leadership Computing Facility (OLCF). Note that the parallel decompositions employed in the runs were 2D only; all elements with the same x and y coordinates were on the same processor (convergence difficulties were encountered when splitting vertical columns in the mesh across processors). A parallel decomposition for 16 cores is illustrated in Figure 10.

Tables 4 and 5 report the relative errors in the computed solution for each mesh resolution considered with uniform and graded vertical mesh spacings (respectively). The convergence metric employed was the continuous L^2 norm. The relative error in each solution was calculated according to the following formula:

$$\mathcal{E}_{rel}^{cont} \equiv \sqrt{\frac{\int_\Omega ||\mathbf{u}_n - \mathbf{u}_{ref}||_2^2 d\Omega}{\int_\Omega ||\mathbf{u}_{ref}||_2^2 d\Omega}}. \tag{46}$$

In (46), $||\cdot||_2$ denotes the L^2 norm, $\mathbf{u}_n$ denotes the computed solution and $\mathbf{u}_{ref}$ denotes the reference solution, which here we take as the solution computed for the finest resolution mesh, the 500 m mesh with 80 vertical layers and graded vertical spacing (for this quasi-realistic problem, there is no exact

[5]More information on *Titan* can be found at www.olcf.ornl.gov/titan.



(a) (b)

Fig. 9. Examples of uniform mesh refinement: (a) No refinement (8 km GIS), (b) 1 level of refinement (4 km GIS)

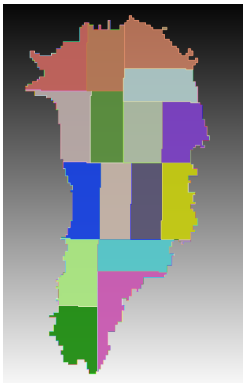


Fig. 10. GIS domain decomposition for 16 core, parallel run, with different colors representing portions of the domain owned by different cores.

solution available in closed analytic form). This finest mesh had 1.12 billion dofs. The integrals in (46) were calculated exactly using a sufficiently accurate numerical quadrature rule.

Table 4. Relative errors for GIS mesh convergence study with uniform vertical spacing.

Horizontal resolution \ Vertical layers	5	10	20	40	80
8 km	2.0×10^{-1}				
4 km	9.0×10^{-2}	7.8×10^{-2}			
2 km	4.6×10^{-2}	2.4×10^{-2}	2.3×10^{-2}		
1 km	3.8×10^{-2}	8.9×10^{-3}	5.5×10^{-3}	5.1×10^{-3}	
500 m	3.7×10^{-2}	6.7×10^{-3}	1.7×10^{-3}	3.9×10^{-4}	8.1×10^{-5}

Table 5. Relative errors for GIS mesh convergence study with graded vertical spacing.

Horizontal resolution \ Vertical layers	5	10	20	40	80
8 km	2.0×10^{-1}				
4 km	8.3×10^{-2}	7.8×10^{-2}			
2 km	3.3×10^{-2}	2.4×10^{-2}	2.3×10^{-2}		
1 km	2.2×10^{-2}	7.3×10^{-3}	5.3×10^{-3}	5.1×10^{-3}	
500 m	2.1×10^{-2}	4.7×10^{-3}	1.2×10^{-3}	2.6×10^{-4}	–

Below, we provide some discussion of the data summarized in Tables 4 and 5, as well as some conclusion drawn from these results.

Mesh convergence

Figure 11 shows the relative error (46) as a function of the horizontal mesh spacing (8 km, 4 km, 2 km, 1 km) on a log-log plot (blue line). The numerical values of the relative error $\mathcal{E}_{rel}$ plotted are the diagonal entries of Tables 4 and 5 (which were identical for the two tables). The asymptotic convergence rate (the slope of the blue line in Figure 11 disregarding the coarsest mesh data point, as it is not in the region of asymptotic convergence) is 1.97. This compares very well with the theoretical convergence rate of two, for the bilinear hexahedral elements considered in this norm (black-dashed line in Figure 11).

Effect of partitioning on mesh convergence

As noted in the discussion of the full 3D mesh convergence study described in Section 6, our study revealed that 2D parallel decompositions of the meshes (i.e., decompositions in which all elements

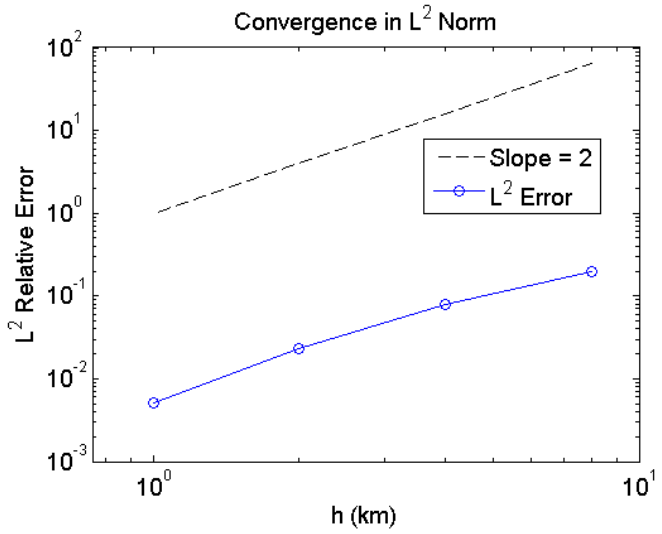


Fig. 11. Convergence in the continuous L^2 norm (46) for the realistic GIS problem with full 3D refinement.

with the same x and y coordinates were on the same processor, as shown in Figure 10) led to out-of-the-box convergence of our linear and nonlinear solves. In contrast, convergence difficulties were encountered when splitting vertical columns in the mesh across processors. The 2D parallel decomposition is therefore recommended over a full 3D parallel decomposition, especially for problems on meshes having a finer vertical resolution.

Uniform vs. graded vertical spacing

The reader may observe in comparing Table 4 and 5 that there is a not a significant difference between the errors in the solutions on the meshes with a uniform vertical resolution and those on meshes with a graded vertical resolution. Nonetheless, there is some value (at no additional computational cost) in using a graded mesh over a uniform mesh for some mesh resolutions.

Practical recommendations on mesh resolution

The data in Tables 4 and 5 suggest that, depending on the selected mesh resolution, there can be more value in refining vertically than horizontally. For example, for the uniform mesh spacing (Table 4), the solution on a 2 km resolution mesh with 10 vertical layers is more accurate than that on a 1 km resolution mesh with 5 vertical layers. Similarly, the solution on a 1 km resolution mesh with 20 vertical layers is more accurate than that on a 500 m resolution mesh with 10 vertical layers. For the graded mesh spacing scenario (Table 5), a solution refined 2 times in z is comparable in accuracy to the solution refined 2 times in the horizontal direction; however, the former problem is smaller,

as refining 2 times vertically leads to a doubling of the number of dofs whereas refining 2 times horizontally leads to a quadrupling of the number of dofs.

Although there can be value in refining vertically, this is true only up to some level of refinement. For each horizontal resolution in Tables 4 and 5 except the finest, the errors plateau beyond a certain vertical resolution. The data in the last row of the tables should be considered very cautiously as here we are using the same horizontal resolution as the reference solution.

The results in Tables 4 and 5 can be used by readers to determine the horizontal and vertical mesh resolution required to attain a desired convergence rate. We note that the errors are for a controlled study, where the ice geometries are not changing with the horizontal resolution and the fields are smoother than in reality. When both the mesh resolution and the geometric data resolution increase simultaneously, horizontal refinement will likely be more important than Tables 4 and 5 suggest.

6.1 Code performance and scalability

Having demonstrated the numerical convergence of our code on a realistic, large-scale ice sheet problem we now study the code's robustness, performance and scalability.

6.1.1 Robustness

In Section 3.1.1, we described our approach for improving the robustness of the nonlinear solver using a homotopy continuation of the regularization parameter (denoted by γ) appearing in the effective viscosity law expression (22). Here, we perform a numerical study of the relative robustness of Newton's method with and without the use of this continuation procedure on a realistic, 5 km resolution Greenland ice sheet problem. Three approaches are considered:

(a) Full Newton with no homotopy continuation.

(b) Newton with backtracking but no homotopy continuation.

(c) Full Newton with homotopy continuation.

For all three methods, a uniform velocity field is specified as the initial guess for Newton's method. To prevent the effective viscosity (8) from evaluating to "not-a-number" for this initial guess, we replace μ by μ_γ in (2), where μ_γ is given by (22) and $\gamma = 10^{-10}$ for the first two approaches. The third approach implements Algorithm 1, in which we use a natural continuation algorithm to reach $\gamma = 10^{-10}$ starting with $\alpha_0 = 0.1$.

Figure 12 illustrates the performance of Newton's method for the three approaches considered by plotting the norm of the residual as a function of the total number of Newton iterations. The reader can observe that full Newton with no homotopy continuation diverges. If backtracking is employed, the algorithm converges to a tolerance of 10^{-4} in 43 nonlinear iterations. With the use of homotopy continuation, the number of nonlinear iterations is cut almost in half, to 24 nonlinear iterations. The natural continuation method leads to four homotopy steps.

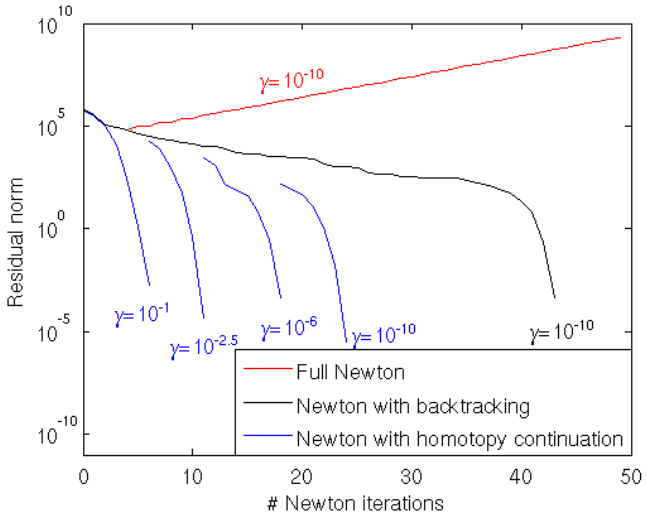


Fig. 12. Robustness of Newton's method nonlinear solves with homotopy continuation

It is well-known that for Newton's method to converge to the root of a nonlinear function (i.e., the solution to the discrete counterpart of (21)), it must start with an initial guess which is reasonably close to the sought-after solution. The proposed homotopy continuation method is particularly useful in the case when no "good" initial guess is available for Newton's method, in which case the nonlinear solver may fail to converge (see Section 3.1.1 and Algorithm 1). Homotopy continuation may not be needed for robust convergence in the case that a "good" initial guess *is* available (e.g., from observations or from a previously converged model time step).

6.1.2 Controlled weak scalability study on successively refined meshes with coarse mesh data

First, we report results for a controlled weak scalability study. For this experiment, the 8 km GIS mesh with 5 vertical layers described in Section 6 was scaled up to a 500 m GIS mesh with 80 vertical layers using the uniform 3D mesh refinement discussed earlier. A total of five meshes were generated, as summarized in Table 6. The term "controlled" refers to the fact that the resolution of the data describing the ice sheet geometry used for initial conditions was held fixed for all the grids considered and equal to the polygonal boundary determined by the coarsest 8km mesh. Moreover, topography, surface height, basal friction and temperature data have been smoothed and then interpolated as described in Section 6. Each resolution problem was run in parallel on the *Hopper*[6] Cray XE6 supercomputer at the National Energy Research Scientific Computing (NERSC) Center. The

[6]More information on the *Hopper* machine can be found here: http://www.nersc.gov/users/computational-systems/hopper.

Table 6. Meshes used in the GIS controlled weak scalability study.

horizontal resolution	# vertical layers	# dofs	# cores
8 km	5	3.34K	4
4 km	10	2.43M	32
2 km	20	18.4M	256
1 km	40	143M	2048
500 m	80	1.12B	16,384

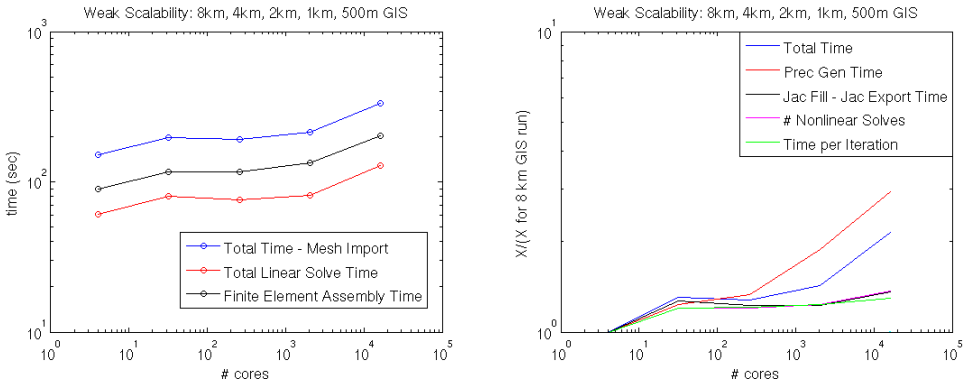
number of cores for each run (third column of Table 6) was calculated so that for each size problem, each core had approximately the same number of dofs ($\approx 70 - 80$K dofs/core). For a detailed discussion of the numerical methods employed, the reader is referred to Section 3. In particular, recall that the linear solver employed is based on the preconditioned CG iterative method. The preconditioner employed is the algebraic multilevel preconditioner based on the idea of semi-coarsening that was described in Section 3.1.2. This preconditioner is available through the *ML* package of *Trilinos* (Heroux et al., 2005).

Figure 13(a) reports the total linear solver time, the finite element (FE) assembly time and the total time (in seconds) for each resolution problem considered, as a function of the number of cores. Figure 13(b) shows more detailed timing information, namely:

- The normalized preconditioner generation time ("Prec Gen Time").
- The normalized Jacobian fill time, not including the Jacobian export time[7] ("Jac Fill - Jac Export Time").
- The normalized number of nonlinear solves ("# Nonlin Solves").
- The normalized average number of linear iterations ("Avg # Lin Iter").
- The normalized total time not including I/O ("Total Time - IO").

The run times and iteration counts have been normalized by the run time and iteration count (respectively) for the smallest run (8 km GIS with 5 vertical layers, run on 4 cores). Figure 13 reveals that the run times and iteration times scale well, albeit not perfectly, in a weak sense.

[7] "Jacobian export time" refers to the time required to transfer ("export") data from an element-based decomposition, which can be formed with no communication, to a node-based decomposition, where rows of the matrix are uniquely owned by a single processor.



(a) (b)

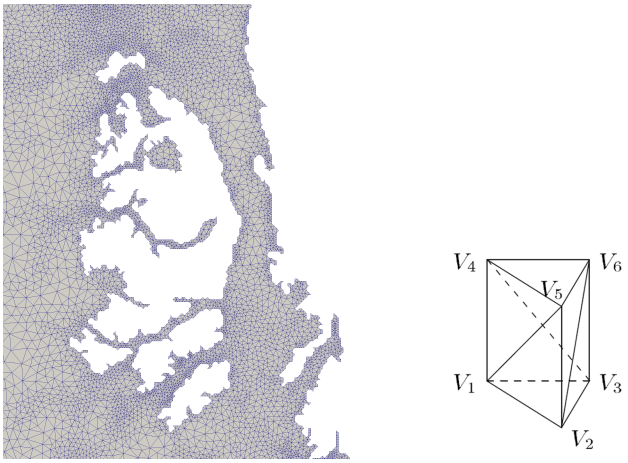
Fig. 13. Controlled, weak scalability study on *Hopper*: (a) Total linear solve, finite element assembly, and total run times in seconds, (b) Additional timing information (X = time or # iterations).

6.1.3 Strong scalability for realistic Greenland initial conditions on a variable-resolution mesh

For the performance study described in the previous paragraph, the data has been smoothed and the lateral boundary was determined by the coarsest (8 km resolution) mesh. We now perform a scalability study for the GIS directly interpolating the original datasets into the mesh considered. This results in better resolved topography, basal friction and temperature fields temperature fields in regions of the domain with higher resolution. As before, the surface topography and temperature fields are from Bamber et al. (2013) and were generated as a part of the Ice2Sea project (Ice2sea, 2014); the optimized basal friction coefficient (β) field is from Perego et al. (2014).

We consider a tetrahedral mesh with a variable resolution of between 1 km and 7 km and having approximately 14.4 million elements, leading to approximately 5.5 million dofs (Figure 14(a)). The mesh was created by first meshing the base of the GIS using the 2D meshing software *Triangle* (Shewchuk et al., 1996). The 2D mesh generated using *Triangle* was a nonuniform Delaunay triangulation in which the areas of the triangles were constrained to be roughly proportional to the norm of the gradient of the surface velocity data. This yields meshes with better resolutions in places where the solution has larger variations. The 2D mesh is then extruded in the z–direction as prisms and each prism is divided into three tetrahedra (Figure 14(b)).

First, we verify that velocities computed on the 1–7 km variable resolution tetrahedral mesh, shown in Figure 15(a), agree with observations to within expectations. The reader can observe that there is generally good agreement between the modeled velocities and those from the target field observations, shown in Figure 15(b) (from Joughin et al. (2010)). Differences between the modeled



(a) (b)

Fig. 14. (a) Close-up of variable-resolution 1–7 km GIS mesh, (b) Subdivision of hexahedral finite element into three tetrahedra.

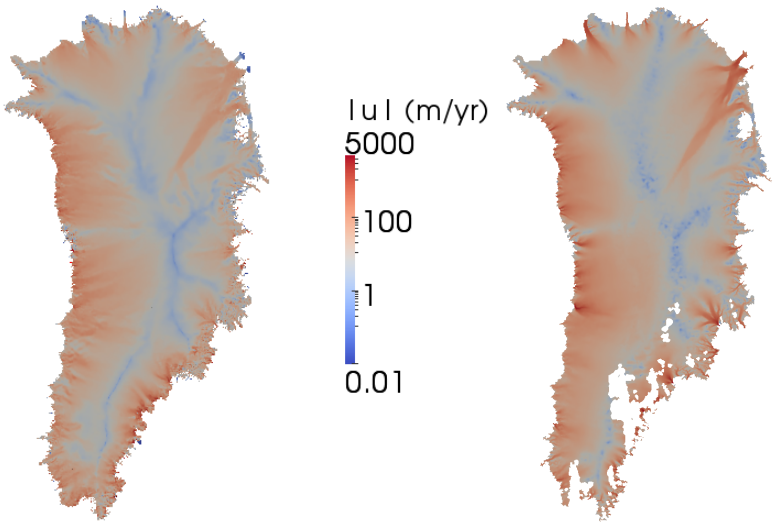
and observed velocities occur as a result of the objective function used during model optimization, which takes into account factors other than just the velocity mismatch [8].

Next, a strong scaling study on the 1–7 km variable resolution GIS problem is performed. The problem is run on different numbers of cores on *Hopper*, from 64 to 512. The total solve, linear solve and finite element assembly times for each of the runs are reported (in seconds) in Table 7. The speed-up relative to the smallest (64 core) run is plotted as a function of the number of cores in Figure 16. Good strong scalability is obtained: a 3.75 times speed-up is observed with 4 times the number of cores (up to and including 256 cores), and a 6.64 times speed-up is observed with 8 times the number of cores (up to and including 512 cores). In these results, the linear solver employed was the preconditioned CG iterative method, with the aforementioned algebraic multilevel preconditioner based on the idea of semi-coarsening (see Section 3.1.2).

7 Conclusions

In this paper, we have presented a new, parallel, finite element solver for the first-order accurate, non-linear Stokes ice sheet model. This solver, *Albany/FELIX*, has been written using a component-based approach to building application codes. The components comprising the code are modular *Trilinos*

[8] The optimization procedure, discussed in more detail in Perego et al. (2014), minimizes the difference between modeled and observed velocities and between the modeled flux divergence and a target surface mass balance field. The latter constraint, which is introduced so that the optimized model is in quasi-steady state with climate model forcing, results in the small differences between the modeled and observed velocities observed in Figure 15.



(a) (b)

Fig. 15. Solution magnitude $|\mathbf{u}|$ in meters per year: (a) *Albany/FELIX* solution (surface speed) on the variable resolution (1–7 km) tetrahedral mesh, (b) observed surface speeds (from Joughin et al. (2010)).

Table 7. Total, linear solve and finite element assembly times (sec) for variable resolution 1–7 km resolution GIS problems as a function of # cores of *Hopper*

# cores	Total Solve Time	Linear Solve Time	Finite Element Assembly Time
64	268.1	119.9	148.3
128	139.9	63.12	76.78
256	78.41	37.92	40.49
512	56.83	33.81	23.02

libraries, which are put together using abstract interfaces and template-based generic programming. Several verifications of the code's accuracy and convergence are carried out. First, a mesh convergence study is performed on several new method of manufactured solutions test cases derived for simplified 2D versions of the first-order Stokes equations. All finite elements tested exhibit their theoretical rate of convergence. Next, code-to-code comparisons are made on several canonical ice sheet benchmarks between the *Albany/FELIX* code and the finite element solver of Perego et al. (2012). The solutions are shown to agree to within machine precision. As a final verification, a mesh convergence study on a realistic Greenland geometry is performed. The purpose of this test is two-fold: (1) to demonstrate that the solution converges at the theoretical rate with mesh refinement,

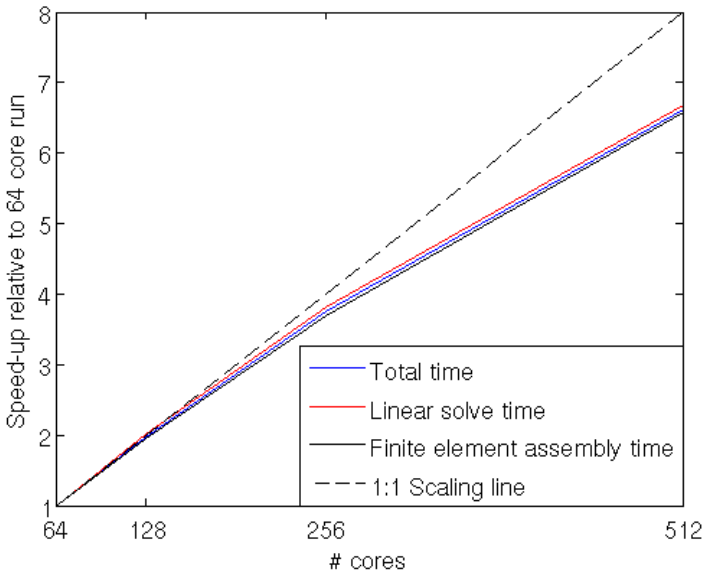


Fig. 16. Strong scalability for 1–7km resolution GIS problem: speed-up relative to 64 core run.

and (2) to determine how many vertical layers are required to accurately resolve the solution with a fixed x–y resolution, when using (low-order) trilinear finite elements. It is found that the parallel decomposition of a mesh has some effect on the linear and nonlinear solver convergence: better performance is observed on the finer meshes if a horizontal decomposition (i.e., a decomposition in which all nodes having the same x and y coordinates are on the same processor) is employed for parallel runs. Further performance studies reveal that a robust nonlinear solver is obtained through the use of homotopy continuation with respect to a regularization parameter in the effective viscosity in the governing equations, and that good weak scalability can be achieved by preconditioning the iterative linear solver using an algebraic multilevel preconditioner constructed based on the idea of semi-coarsening.

Finally, we note that the ultimate purpose for developing *Albany/FELIX* is to integrate it into more complete land ice modeling frameworks so that it can be used for prognostic simulations, both in standalone mode and as a coupled component of ESMs. In addition to the conservation of linear momentum being solved by *Albany/FELIX*, a complete prognostic land ice model must also solve discretized PDEs for the conservation of mass and energy, in addition to treating other physical processes such as lithospheric heat exchange and isostatic bedrock adjustment. To enable prognostic runs, we have written interfaces for coupling *Albany/FELIX* to two larger land ice modeling frameworks, which discretize and solve the equations for mass and energy conservation: *The Community*

Ice Sheet Model version 2.0 (*CISM2*) (Price et al. , 2014) and *The Model for Prediction Across Scales - Land Ice* (*MPAS-LI*) (Hoffman, 2014). We refer to the resulting complete land ice models as *CISM-Albany* and *MPAS-Albany* respectively. Prognostic runs using these dycores are iterative in nature, with a diagnostic solve for the velocity field occurring in *Albany/FELIX*, followed by solutions for the geometry and temperature evolution occurring in *CISM2* or *MPAS-LI*. Further discussion of *CISM-Albany* and *MPAS-Albany* and ongoing work involving their coupling to ESMs will be deferred to subsequent papers. Similarly, these combined codes will be publicly released at a later point in time.

Appendix A: Nonlinear Stokes model for glaciers and ice sheets

The model considered here, referred to as the first-order (FO) Stokes approximation, or the "Blatter-Pattyn" model (Blatter, 1995; Pattyn, 2003), is an approximation of the nonlinear Stokes model for glacier and ice sheet flow. In general, glaciers and ice sheets are modeled as an incompressible fluid in a low Reynolds number flow with a power-law viscous rheology, as described by the Stokes flow equations. The equations are quasi-static, as the inertial and advective terms can be neglected due to the slow movement of the ice.

Let $\boldsymbol{\sigma}$ denote the Cauchy stress tensor, given by

$$\boldsymbol{\sigma} = 2\mu\dot{\boldsymbol{\epsilon}} - p\mathbf{I} \in \mathbb{R}^{3\times 3}, \tag{47}$$

where μ denotes the "effective" ice viscosity, p the ice pressure, $\mathbf{I}$ the identity tensor, and $\dot{\boldsymbol{\epsilon}}$ the strain-rate tensor:

$$\dot{\epsilon}_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \tag{48}$$

for $i, j \in \{1, 2, 3\}$. The effective viscosity is given by Glen's law (Nye, 1957; Cuffey et al., 2010):

$$\mu = \frac{1}{2}A^{-\frac{1}{n}}\dot{\epsilon}_e^{\left(\frac{1}{n}-1\right)}, \tag{49}$$

where

$$\dot{\epsilon}_e = \sqrt{\frac{1}{2}\sum_{ij}\dot{\epsilon}_{ij}^2}, \tag{50}$$

denotes the effective strain rate, given by the second invariant of the strain-rate tensor. A denotes the flow rate factor (which is strongly dependent on the ice temperature), and n denotes the power law exponent (generally taken equal to 3). The nonlinear Stokes equations for glacier and ice sheet flow can then be written as follows:

$$\begin{cases} -\nabla\cdot\boldsymbol{\sigma} = \rho\mathbf{g} \\ \nabla\cdot\mathbf{u} = 0. \end{cases} \tag{51}$$

Here, ρ denotes the ice density, and $\mathbf{g}$ the gravitational acceleration vector, i.e., $\mathbf{g}^T = \Big(0,\, 0,\, -g\Big)$, with g denoting the gravitational acceleration. The values of the parameters that appear in the expressions above are given in Table 1. A stress-free boundary condition is prescribed on the upper surface:

$$\boldsymbol{\sigma}\mathbf{n} = \mathbf{0},\ \text{on}\ \Gamma_s. \tag{52}$$

On the lower surface, the relevant boundary condition is the no-slip or basal sliding boundary condition:

$$\begin{cases} \mathbf{u} = \mathbf{0}, & \text{on}\ \Gamma_0, \\ \mathbf{u}\cdot\mathbf{n} = 0 \text{ and } (\boldsymbol{\sigma}\mathbf{n} + \beta\mathbf{u})_{||} = \mathbf{0}, & \text{on}\ \Gamma_\beta, \end{cases} \tag{53}$$

assuming $\Gamma_b = \Gamma_0 \cup \Gamma_\beta$ with $\Gamma_0 \cap \Gamma_\beta = \emptyset$, where $\beta \equiv \beta(x,y) \geq 0$. The operator $(\cdot)_{||}$ in (53) performs the tangential projection onto a given surface.

Code Availability

The *Albany* framework is an open-source development project available for download on GitHub (http://gahansen.github.io/Albany). *Albany*, currently on its 2.0 release, is released under a publicly available designation with a three-term BSD license.

The *Albany* framework was written using many libraries available through *Trilinos*, also publicly available (http://trilinos.org). At the time this journal article was written, *Trilinos* was on its 11.12 release. The multigrid algorithm presented in this paper (Section 3.1.2) is implemented within the *ML* package of *Trilinos* and is available for use with *Trilinos 11.12* or later.

The *Albany/FELIX* solver described in this paper is not publicly available at the present time. A public release of the code as part of *Albany* is planned for 2015. The *CISM* and *MPAS* ice sheet models with supported interfaces to *Albany/FELIX* will also be made publicly available at a later point in time.

Acknowledgements

Support for all authors was provided through the Scientific Discovery through Advanced Computing (SciDAC) program funded by the U.S. Department of Energy (DOE), Office of Science, Advanced Scientific Computing Research and Biological and Environmental Research. This research used resources of the National Energy Research Scientific Computing Center (NERSC; supported by the Office of Science of the U.S. Department of Energy under Contract DE-AC02-05CH11231) and the Oak Ridge Leadership Computing Facility (OLCF; supported by the DOE Office of Science under Contracts DE-AC02-05CH11231 and DE-AC05-00OR22725). The authors thank M. Norman of Oak Ridge National Laboratory for generation of the Greenland geometry datasets, J. Johnson (and

students) of the University of Montana for initial development of the ISMIP-HOM plotting scripts, and M. Hoffman and B. Lipscomb at Los Alamos National Laboratory for useful discussions that led to some of the ideas and results presented in this paper.

References

B. Adams, L. Bauman, W. Bohnhoff, K. Dalbey, M. Ebeida, J. Eddy, M. Eldred, P. Hough, K. Hu, J. Jakeman, L. Swiler, and D. Vigil, DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 5.4 User's Manual, *Sandia Technical Report SAND2010-2183*, December 2009. Updated April 2013.

E. Allgower, and K. Georg, Introduction to Numerical Continuation Methods, *SIAM Classics in Applied Mathematics*, **45** (2003).

S. Balay, K. Buschelman, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. McInnes, B. Smith, H. Zhang, PETSc Users Manual, *Technical Report ANL-95/11, Revision 3.0.0*, Argonne National Laboratory, Lemont, IL (2008).

J. Bamber, J Griggs, R. Hurkmans, J. Dowdeswell, S. Gogineni, I. Howat, J. Mouginot, J. Paden, S. Palmer, E. Rignot, and D. Steinhage, A new bed elevation dataset for Greenland, *Cryosphere*, **7**(2), 499–510 (2013).

E. Bavier. M. Hoemmen, S. Rajamanickam, and H. Thornquist, Amesos2 and Belos: Direct and iterative solvers for large sparse linear systems, *Sci. Program.*, **20**, 241–255 (2012).

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

P. Bochev, C. Edwards, R. Kirby, K. Peterson, and D. Ridzal, Solving PDEs with Intrepid, *Sci. Program.*, **20**, 151–180 (2012).

D. Brinkerhoff, and J. Johnson, Data assimilation and prognostic whole ice sheet modelling with the variationally derived, higher order, open source, and fully parallel ice sheet model VarGlaS, *Cryosphere*, **7** 1161–1184 (2013) doi:10.5194/tc-7-1161-2013.

British Antarctic Survey, ice2sea FP7 EU Project Homepage, March 2014, URL: http://www.ice2sea.eu.

P. Brown, R. Falgout, and J. Jones, Semi-coarsening Multigrid on Distributed Memory Machines *SIAM J. Sci. Comput.*, **21**(5), 1823–1834 (2000).

J. Brown, B. Smith, and A. Ahmadia, Achieving textbook multigrid efficiency for hydrostatic ice sheet flow, *SIAM J. Sci. Comput.*, **35**(2), B359–B375 (2013).

E. Bueler, J. Brown, and C. Lingle, Exact solutions to the thermomechanically coupled shallow-ice approximation: effective tools for verification, *J. Glaciol.*, **53**, 499–516 (2007).

E. Bueler, and J. Brown, Shallow shelf approximation as a 'sliding law' in a thermomechanically coupled ice sheet model, *J. Geophys. Res.*, **114**, 1–21 (2009), doi:10.1029/2008JF001179.

Q. Chen, M. Gunzburger, and M. Perego, Well-Posedness Results for a Nonlinear Stokes Problem Arising in Glaciology, *SIAM J. Math. Anal.*, **45**(5), 2710–2733 (2013).

S. Cornford, D. Martin, D. Graves, D. Ranken, A. Le Brocq, R. Gladstone, A. Payne, J. Antony, E. Ng, and W. Lipscomb, Adaptive mesh, finite volume modeling of marine ice sheets, *J. Comput. Phys.*, **232**, 529–549 (2013), doi:10.1016/j.jcp.2012.08.037.

K. Cuffey, and W. Paterson, *The physics of glaciers*, 4th edition (Butterworth-Heinemann, Oxford, 2010).

J. Dendy and J. Moulton, Black box multigrid with coarsening by a factor of three, *Numer. Lin. Algebra Appl.*, **17**(2-3), 577–598 (2010).

J. Dukowicz, S. Price, and W. Lipscomb, Consistent approximations and boundary conditions for ice-sheet dynamics from a principle of least action, *J. Glaciol.*, **56**(197), 480–496 (2010).

J. Dukowicz, S. Price, and W. Lipscomb, Incorporating arbitrary basal topography in the variational formulation of ice-sheet models, *J. Glaciol.*, **57**(203), 461–467 (2011).

O. Gagliardini, T. Zwinger, F. Gillet-Chaulet, G. Durand, L. Favier, B. de Fleurian, R. Greve, M. Mallnen, C. Martin, P. Raback, J. Ruokolainen, M. Sacchettini, M. Schafer, H. Seddik, and J. Thies, Capabilities and performance of Elmer-Ice, a new-generation ice sheet model, *Geosci. Model Dev.*, **6**, 1299–1318 (2013), doi:10.5194/gmd-6-1299-2013.

X. Gao, E. Nielsen, R. Muller, R. Young, A. Salinger, N. Bishop, M. Lilly, and M. Carroll, Quantum computer aided design simulation and optimization of semiconductor quantum dots, *J. Appl. Phys.,* **114**(164302) 1–19 (2013).

M. Gee, C. Siefert, J. Hu, R.Tuminaro, and M.Sala, *ML* 5.0 smoothed aggregation user's guide, *Sandia National Laboratories Report, SAND2006-2649*, Sandia National Laboratories, Albuquerque, NM (2007).

F. Gillet-Chaulet, O. Gagliardini, H. Seddik, M. Nodet, G. Durand, C. Ritz, T. Zwinger, R. Greve, and D.G. Vaughan, Greenland ice sheet contribution to sea-level rise from a new-generation ice-sheet model, *Cryosphere*, **6**, 1561–1576 (2012), doi:10.5194/tc-6-1561-2012.

D. Goldberg, and O.Sergienko, Data assimilation using a hybrid ice flow model, *Cryosphere*, **5**, 315–327 (2011), doi:10.5194/tc-5-315-2011.

D. Goldberg, and P. Heimbach, Parameter and state estimation with a time-dependent ajdoint marine ice sheet model, *Cryosphere*, **6**, 1659–1678 (2013).

M. Hagdorn, I. Rutt, T. Payne, and F. Hebeler, Glimmer 1.5.1 Documentation, April 2010, URL: http://www.cesm.ucar.edu/models/cesm1.0/cism/docs/glimmer.pdf.

Hoffman, M. MPAS-Land Ice Model User's Guide Version: 3.0. (2013), available at: http://oceans11.lanl.gov/mpas_data/mpas_landice/users_guide/release_3.0/mpas_landice_users_guide_3.0.pdf.

M. Heroux, R. Bartlett, V. Howle, R. Hoekstra, J. Hu, T. Kolda, R. Lehoucq, K. Long, R. Pawlowski, E. Phipps, A. Salinger, H. Thornquist, R. Tuminaro, J. Willenbring, A. Williams, and K. Stanley, An Overview of the Trilinos Project, *ACM Trans. Math. Softw.*, **31**(3), 397–423 (2005).

T. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis (Dover edition, 2000).

J. Hurrell, M. Holland, P. Gent, S. Ghan, J. Kay, P. Kushner, J.-F. Lamarque, W. Large, D. Lawrence, K. Lindsay, W. Lipscomb, M. Long, N. Mahowald, D. Marsh, R. Neale, P. Rasch, S. Vavrus, M. Vertenstein, D. Bader, W. Collins, J. Hack, J. Kiehl, and S. Marshall, The Community Earth System Model: A Framework for Collaborative Research, *B. Am. Meteorol. Soc.*, **94**, 1339–1360 (2013).

K. Hutter, Theoretical glaciology: material science of ice and the mechanics of glaciers and ice sheets (Springer, 1983).

T. Isaac, G. Stadler, and O. Ghattas, Solution of nonlinear Stokes equations discretized by high-order finite elements on nonconforming and anisotropic meshes, with application to ice sheet dynamics, *submitted for publication and currently available at*: http://arxiv.org/abs/1406.6573.

I. Joughin, B. Smith, I. Howat, T. Scambos, and T. Moon, Greenland flow variability from ice-sheet-wide velocity mapping, *J. Glaciol.*, **56**, 415–430 (2010).

G. Jouvet, and C. Graser, An adaptive Newton multigrid method for a model of marine ice sheets, *J. Computat. Phys.*, **252**, 419–437 (2013), doi:10.1016/j.jcp.2013.06.032.

A. Khazendar, E. Rignot, and E. Larour, Roles of marine ice, rheology, and fracture in the flow and stability of the Brunt/Stancomb-Wills Ice Shelf, *J. Geophys. Res.* **114** (2009), doi:10.1029/2008JF001124.

E. Larour, H. Seroussi, M. Morlighen, and E. Rignot, Continental scale higher order, high spatial resolution, ice sheet modeling using the Ice Sheet System Model (ISSM), *J. Geophys. Res.*, **117**(FO1022), 1–20 (2012).

E. Larour, H. Seroussi, M. Morlighem, and E. Rignot, Continental scale, high order, high spatial resolution, ice sheet modeling using the Ice Sheet System Model (ISSM), *J. Geophys. Res.*, **117**(F01022), 1–20 (2012), doi:10.1029/2011JF002140.

J.-F. Lemieux, S. Price, K. Evans, D. Knoll, A. Salinger, D. Holland, and A. Payne, Implementation of the Jacobian-free Newton-Krylov method for solving first-order ice sheet momentum balance, *J. Comput. Phys.* **230**, 6531–6545 (2011).

W. Leng, L. Ju, M. Gunzburger, S. Price, and T. Ringler, A Parallel Higher-Order Accurate Finite Element Nonlinear Stokes Ice Sheet Model and Benchmark Experiments, *J. Geophys. Res.*, **117**(F1), 1–24 (2012).

W. Leng, L. Ju, M. Gunzburger, and S. Price, Manufactured solutions and the numerical verification of isothermal, nonlinear, three-dimensional Stokes ice sheet models, *Cryosphere*, **6** 2689–2714 (2012).

W. Leng, L. Ju, M. Gunzburger, and S. Price, A Parallel Computational Model for Three-Dimensional, Thermo-Mechanical Stokes Flow Simulations of Glaciers and Ice Sheets, *Commun. Comput. Phys.*, *in press and currently available at*: http://www.global-sci.com/galley/CiCP-272.pdf.

W. Lipscomb, R. Bindschadler, S. Price, E. Bueler, J. Johnson, and D. Holland, A community ice sheet model for sea level prediction, *Eos. Trans. AGU* **90**(3), 23, (2008).

W. Lipscomb, J. Fyke, M. Vizcaino, W. Sacks, J. Wolfe, M. Vertenstein, A. Craig, E. Kluzek, and D. Lawrence, Implementation and Initial Evaluation of the Glimmer Community Ice Sheet Model in the Community Earth System Model, *J. Climate*, **26**, 7352–7371 (2013) doi:10.1175/JCLI-D-12-00557.1.

C. Little, Toward a new generation of ice sheet models, *Eos. Trans. AGU*, **88**(52) 578–579 (2007).

D. Macayeal, Large-Scale Ice Flow Over a Viscous Basal Sediment - Theory and Application to Ice Stream-B, Antarctica, *J. Geophys. Res.*, **94**, 4071–4087 (1989).

D. MacAyeal, V. Rommelaere, P. Huybrechts, C. Hulbe, J. Datemann, and C. Ritz, An ice-shelf model test based on the Ross Ice Shelf, Antarctica, *Ann. Glaciol.*, **23**, 46–51 (1996).

L. Morland, Unconfined ice-shelf flow, *In Dynamics of the West Antarctic Ice Sheet Proc. Workshop held in Utrecht*, May 6–8, 1985 (ed. C. van der Veen and J. Oerlemans), 99–116 (1987).

M. Morlighem, E. Rignot, H. Seroussi, E. Larour, H. Ben Dhia, and D. Aubry, Spatial patterns of basal drag inferred using control methods from a full-Stokes and simpler models for Pine Island Glacier, West Antarctica, *Geophys. Res. Lett.*, **37**(L14502), 1–6 (2010), doi:10.1029/2010GL043853.

J. Nye, The Distribution of Stress and Velocity in Glaciers and Ice-Sheets, *Proc. R. Soc. London, Ser. A*, **239**(1216), 113–133 (1957).

Pattyn, F., 2002: Transient glacier response with a higher-order numerical ice-flow model. *J Glaciol.*, **48**, 467?477.

F. Pattyn, A new three-dimensional higher-order thermomechanical ice-sheet model: basic sensitivity, ice stream development, and ice flow across subglacial lakes, *J. Geophys. Res.*, **108**(B8, 2382) 1–15 (2003).

F. Pattyn, L. Perichon, A. Aschwanden, B. Breuer, B. de Smedt, O. Gagliardini, G. Gudmundsson, R. Hindmarsh, A. Hubbard, J. Johnson, T. Kleiner, Y. Konovalov, C. Martin, A. Payne, D. Pollard, S. Price, M.

Ruckamp, F. Saito, O. Soucek, S. Sugiyama, and T. Zwinger, Benchmark experiments for higher-order and full-Stokes ice sheet models (ISMIP-HOM), *Cryosphere*, **2**(2), 95–108 (2008).

R. Pawlowski, J. Shadid, J. Simonis, and H. Walker, Globalization techniques for Newton-Krylov methods and applications to the fully coupled solution of the Navier-Stokes equations, *SIAM Rev.*, **48**(4), 700–721 (2006).

R. Pawlowski, E. Phipps, and A. Salinger, Automating embedded analysis capabilities and managing software complexity in multiphysics simulation, Part I: Template-based generic programming, *Sci. Program.*, **20**, 197–219 (2012).

R. Pawlowski, E. Phipps, A. Salinger, S. Owne, C. Siefert, and M. Staten, Automating embedded analysis capabilities and managing software complexity in multiphysics simulation, Part II: Application to partial differential equations, *Sci. Program.*, **20**, 327–345 (2012).

M. Perego, M. Gunzburger, and J. Burkardt, Parallel finite-element implementation for higher-order ice-sheet models, *J. Glaciol.*, **58**(207), 76–88 (2012).

M. Perego, S. Price, and G. Stadler, Optimal Initial Conditions for Coupling Ice Sheet Models to Earth System Models, *J. Geophys. Res.* (2014), *published online and available at:* doi:10.1002/2014JF003181.

N. Petra, J. Martin, G. Stadler, and O. Ghattas, A computational framework for infinite-dimensional Bayesian inverse problems: Part II. Stochastic Newton MCMC with application to ice sheet inverse problems, *SIAM J. Sci. Comput.*, *in press and currently available at:* http://arxiv.org/abs/1308.6221.

E. Phipps, and R. Pawlowski, Efficient Expression Templates for Operator Overloading-based Automatic Differentiation, in *Recent Advances in Algorithmic Differentiation*, eds. S. Forth, P. Hovland, E. Phipps, J. Utke, and A. Walther (Springer, 2012).

D. Pollard, and R. Deconto, Modelling West Antarctic ice sheet growth and collapse through the past five million years, *Nature*, **458**, 329–332 (2009), doi:10.1038/nature07809.

S. Price, E. Waddington, and H. Conway, A full-stress, thermomechanical flow band model using the finite volume method, *J. Geophys. Res.* **112**(F03020), 1–17 (2007).

Price, S., W. Lipscomb, M. Hoffman, M. Hagdorn, I. Rutt, A.J. Payne, and F. Hebeler. CISM 2.0.0 Documentation (2014), available at: http://oceans11.lanl.gov/trac/CISM/documentation.html.

V. Rommelaere, Ice Shelf Models Intercomparison: setup of the experiments, unpublished document, 1996, URL: http://homepages.vub.ac.be/~phuybrec/eismint/shelf-descr.pdf.

I. Rutt, M. Hagdorn, N. Hulton, and A. Payne, The Glimmer community ice sheet model, *J. Geophys. Res.*, **114**(F02004), 1–22 (2009).

A. Salinger, N. Bou-Rabee, E. Burroughs, R. Lehoucq, R. Pawlowski, L. Romero, and E. Wilkes, LOCA 1.0: Theory and Implementation Manual, *Sandia Labs Technical Report*, SAND2002-0396, (2002).

A. Salinger, E. Burroughs, R. Pawlowski, E. Phipps, and L. Romero, Bifurcation tracking algorithms and software for large scale applications, *Int. J. Bifurcat. Chaos*, **3**(15), 1015–1032 (2005).

A. Salinger, E. Phipps, G. Hansen, I. Kalashnikova, J. Ostien, W. Sun, Q. Chen, A. Mota, R.Muller, E. Nielsen, and X. Gao, Albany: A Component-Based Partial Differential Equation Code Built on Trilinos, *Sandia Labs Technical Report*, SAND2013-8430, (2013).

S. Schaffer, A semi-coarsening multigrid method for elliptic partial differential equations with highly discontinuous and anisotropic coefficients, *SIAM J. Sci. Comput.*, **20**(1), 228–242 (1998).

C. Schoof, Coulomb friction and other sliding laws in a higher-order glacier flow model, *Math. Mod. Meth.*

Appl. S., **20**(1), 157–189 (2010).

C. Schoof, and R. Hindmarsh, Thin-Film Flows with Wall Slip: An Asymptotic Analysis of Higher Order Glacier Flow Models, *Q. J. Mech. Appl. Math.*, **63**, 73–114 (2010).

H. Seroussi, H. Ben Dhia, M. Morlighem, E. Larour, E. Rignot, and D. Aubry, Coupling ice flow models of varying orders of complexity with the Tiling method, *J. Glaciol.*, **58**, 776–786 (2012), doi:10.3189/2012JoG11J195.

S. Shannon, A. Payne, I. Bartholomew, M. van der Broeke, T. Edwards, X. Fettweis, O. Gagliardini, F. Gillet-Chaulet, H. Goelzer, M. Hoffman, P. Huybrechts, D. Mair, P. Nienow, M. Perego, S. Price, P. Smeets, A. Sole, R. van de Wal, and T. Zwinger, Enhanced basal lubrication and the contribution of the Greenland ice sheet to future sea-level rise, *Proc. Natl. Acad. Sci. USA*, **110**, 14156–14161 (2013), doi:10.1073/pnas.1212647110.

J. Shewchuk, Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, in *Applied Computational Geometry: Towards Geometric Engineering*, **1148**, *Lecture Notes in Compute Science*, 203–222 (Springer-Verlag, 1996).

S. Solomon, D. Qin, M. Manning, Z. Chen, M. Marquis, K. Averyt, M. Tignor, and H. Miller, Climate change 2007: The physical science basis. *Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change* (Cambridge Univ Press, Cambridge, UK, 2007).

W. Strang, and G. Fix, Analysis of the finite element method, (Prentice-Hall series in automatic computation, 1973).

W. Sun, J. Ostien, and A. Salinger, A stabilized assumed deformation gradient finite element formulation for strongly coupled poromechanical simulations at finite strain, *Int. J. Numer. Anal. Met.*, **37**(16), 2755–2788 (2013).

R. Tuminaro, ML's Semi-Coarsening Feature, *Addition to ML 5.0 Smoothed Aggregation User's Guide, Sandia National Laboratories Report, SAND2006-2649*, Sandia National Laboratories, Albuquerque, NM (2014), available at: http://www.trilinos.org/wordpress/wp-content/uploads/2014/08/SemiCoarsen.pdf.

R.Tuminaro, I. Kalashnikova, M. Perego, and A. Salinger, *A hybrid operator dependent multi-grid/algebraic multi-grid approach: Application to ice sheet modeling*, in preparation for submission to *SIAM J. Sci. Comput.* (2015).

University Corporation for Atmospheric Research (UCAR), CESM 1.0: Community Earth System Model, 2013, URL: http://www.cesm.ucar.edu/models/cesm1.0.

C. van der Veen, and ISMASS (Ice Sheet Mass Balance and Sea Level) Working Group, Ice Sheet Mass Balance and Sea Level: A Science Plan, *Scientific Committee on Antarctic Research (SCAR) Report No. 38*, ISSN 1755–9030 (2010).