Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2019-249-AC1, 2019

Geoscientific Model Development Discussions

Interactive comment on "Modelling thermomechanical ice deformation using a GPU-based implicit pseudo-transient method (FastICE v1.0)" *by* Ludovic Räss et al.

Ludovic Räss et al.

lraess@stanford.edu

Received and published: 6 December 2019

Referee's comment 1

The manuscript is well-written, and the governing equations are clearly presented. Overall the manuscript was enjoyable to read, and I learned a lot. The tests were also convincing – as convincing as visual comparisons of results can be. In general, the structure of the manuscript is traditional: Introduce new model, explain the basic principles and the implementation, and test the model against other models. This is fine, and it provides a convenient reference for later work. However, as a reader I would have liked to see a demonstration of what the new model can really do – just

a sneak peek into the suite of problems that the authors hope to address with this new model. There are so many ice models being presented, but it is unfortunately surprisingly rare that we see ice-sheet models applied in ways that make us wiser. So, if possible, I encourage the authors to include a demonstration of the model toward the end of the manuscript – something that is visually, and intellectually, more appealing than the benchmark tests.

Author's reply 1

Thank you for your encouraging feedback. We agree that there is an increasing number of ice models and that it is not always clear what the specific contribution of these models to ice dynamics is. The motivation behind developing this model is to develop a process-based model that affords the necessary 3D resolution to capture englacial strain localisation. This process may be of critical importance in the boundaries of fast flow like the basal interface, grounding zones and shear margins. It is also a subtle component of the overall ice dynamics and requires a careful assessment of when and why it becomes relevant and which locations on our ice sheets might serve as test sites for the model predictions. We are currently working on two follow-up manuscripts applying this code to the flow-to-sliding transition and to shear margin stability. As you mention, developing sophisticated models and advancing our understanding of ice dynamics are two distinct challenges. The first is a necessary but not a sufficient condition for the latter. To do justice to both, we prefer to focus on the numerical methods, benchmarking and performance evaluation for this manuscript and leverage this code for advancing our understanding of ice dynamics in a separate manuscript that we will submit to a glaciological journal. While we agree that an actual application case is more appealing and interesting than benchmarks, we believe that this code can help us make progress on important, fundamental questions in glaciology and we prefer to develop this potential fully in our separate contributions. We are happy to make preliminary results available to you to demonstrate the value of the code for these problem. For this manuscript, we have included a more detailed motivation for

this kind of code and more extensive reference to the problems for which it is relevant.

Referee's comment 2

Line 36: The GPU-acceleration is very interesting and, as far as I know, rather new in ice-sheet models. However, a quick search leads to Brædstrup et al. (2014) "Ice-sheet modelling accelerated by graphic cards" in Computers & Geosciences 72, 210-220. This paper is not cited here, although it covers some of the same challenges and principles of GPU-acceleration.

Author's reply 2

Thank you for pointing this out. We indeed overlooked the citation of the work from Brædstrup et al. (2014).

Changes in the manuscript 2

We added reference to this work in the revised manuscript at line 61: "We tailor our numerical method to optimally exploit the massive parallelism of GPU hardware, taking inspiration from recent successful GPU-based implementations of viscous and coupled flow problems (Brædstrup et al., 2014; Omlin, 2017; Räss et al., 2018; Duretz et al., 2019; Räss et al., 2019a)."

Referee's comment 3

line 42: Also, regarding GPU-acceleration, it would be good to see reference to other flow problems that have successfully been GPU accelerated. What problems and models have inspired the authors?

Author's reply 3

We rephrased in a more explicit way the source of inspiration of the GPU-based FastICE implementation (line 61).

Changes in the manuscript 3

Line 61: "We tailor our numerical method to optimally exploit the massive parallelism of GPU hardware, taking inspiration from recent successful GPU-based implementations of viscous and coupled flow problems (Brædstrup et al., 2014; Omlin, 2017; Räss et al., 2018; Duretz et al., 2019; Räss et al., 2019a)."

Referee's comment 4

line 122: The comment on single-precision calculations leaves me confused. Are the GPU-calculations single precision? Or does it depend on the specific GPU architecture? Please clarify.

Author's reply 4

The benchmarks and calculations in this study are performed using double precision arithmetic if not specified otherwise. We reported single precision efficiency to show the potential performance gain from reducing the arithmetic precision of the calculations. Until recently, it was commonly admitted and implicitly assumed that scientific calculations are (and should be) performed using double precision floating point arithmetic. This choice goes back a couple of decades ago when hardware was computation-bounded; double precision would provide enhanced convergence, thus more efficient calculations, since less floating operations were needed. However, we nowadays observe a shift towards memory-bounded hardware and software where transferring memory (numbers) is more limiting compared to performing arithmetic operations. Thus single or half precision calculation may become interesting as the numbers take twice or four time less amount of memory - which results in factor 2 or 4 performance increase. Alternatively, similar performance can be observed for a two or four-times increase in the numerical grid resolution. Future work may address whether performing calculations using lower arithmetic precision but increased numerical grid resolutions can outperform well-established double precision calculations. A detailed assessment of the issue may deserve separate publication.

Changes in the manuscript 4

Line 257: “The computations in CUDA C shown in the remainder of the paper were performed using double-precision arithmetic, if not specified otherwise.”

Referee’s comment 5

line 163: Braedstrup et al has a nice description of staggered grids and GPU acceleration – must be cited here.

Author’s reply 5

Although we do not question the accurate description of the staggered grid from Braestrup et al., they use a Gauss-Seidel solver in their study, which shows some limitations in terms of parallel implementation. The solve they use requires information from neighbouring cells at each iteration which may, when executed in parallel, lead to read/write conflicts. Our PT solver relies on a fully parallel iteration strategy, which inherently takes care of updating the entire field of old values with updated ones thus circumventing the neighbouring cell read/write issues and avoiding to rely on a “red-black” type of scheme. We are now citing the suggested work, just not with specific reference to the staggered grid setup.

Changes in the manuscript 5

–

Referee’s comment 6

line 175: Even up-wind advection schemes are going to suffer from numerical diffusion – and high numerical resolution is just making it worse. Please discuss this here.

Author’s reply 6

True, upwind scheme also suffer from numerical diffusion. To ensure that our numerical results are not confounded by numerical diffusion, we set the numerical resolution such that the Grid Peclet number is smaller than the physical Peclet number, i.e. n_x>L_x*v_x/2. Limiting numerical diffusion is one motivation for using high numerical

resolution in our computations.

Changes in the manuscript 6

We have added the following clarification to the paragraph on line 183: “To ensure that our numerical results are not confounded by numerical diffusion, the Grid Peclet number must be smaller than the physical Peclet number. Limiting numerical diffusion is one motivation for using high numerical resolution in our computations.”

Referee’s comment 7

line 182: The matrix-free solver using pseudo-time is nicely explained. However, it would be good to see exactly how the residuals propagate in the grid. Many similar matrix-free relaxation schemes use multi-grid setups to make the residuals decay faster – these could be discussed.

Author’s reply 7

An excellent point, thanks for bringing it up. We have included an additional figure in section 5.5 displaying the decay of the residual as function of the damping parameter. Multi-Grid configuration are an alternative solution improving residual decay. However, MG methods may generate quite some overhead by the addition of multiple grid levels and may hinder performance by restriction and prolongation operators. Also, coarser grid may not saturate the GPU and result in a drop of efficiency.

Changes in the manuscript 7

Line 247: “The iteration count increases with the numerical problem size for second-order PT solvers scales close to ideal multi-grid implementations. However, the main advantage of the PT approach is its conciseness and the fact that only one additional read/write operation needs to be included - keeping additional memory transfers to the strict minimum.”

Referee’s comment 8

Eqn. 15: I believe that theta < 1 is often referred to as under-relaxation.

Author's reply 8

The variable is a scalar we use to select the fraction of a given nonlinear quantity to be updated each iteration. When theta=0, we would always use the initial guess, while theta=1, we would take 100% of the current nonlinear quantity. We usually define theta to be in the range of 1e-2 - 1e-1 in order to account for some time to fully relax the nonlinear quantities as the nonlinear problem may not be sufficiently converged at the beginning of the iterations. This approach is in a way similar to an under-relaxation scheme.

Changes in the manuscript 8

Line 204-209: "We use the scalar [. . .] to select the fraction of a given nonlinear quantity, here the effective viscosity [. . .], to be updated each iteration. When =0, we would always use the initial guess, while =1, we would take 100% of the current nonlinear quantity. We usually define theta to be in the range of [. . .] in order to account for some time to fully relax the nonlinear viscosity as the nonlinear problem may not be sufficiently converged at the beginning of the iterations. This approach is in a way similar to an under-relaxation scheme and was successfully implemented in the ice sheet model development by Tezaur (2015), for example."

Referee's comment 9

Eqn. 19: Again, I miss information on how the residuals decay in the grid – particularly when using this stabilizing scheme. Also, I could not find previous reference to alpha, but I might have missed it.

Author's reply 9

Thank you for pointing out the missing alpha definition. We no longer use alpha in the manuscript, replacing it explicitly for enhanced clarity in Eqn. 19. We have also added a Figure 16 in the new Section 5.5 displaying a) the residuals' convergence

history for a 2-D simulation and b) the impact of the "stabilising" scheme as function of the damping parameter nu in terms of the total number of iteration count to reach convergence threshold.

Changes in the manuscript 9

Line 453-458 and Figure 16.

Referee's comment 10

line 299: I can see how the non-dimensional equation makes the implementation simpler, but is it necessary to present results in the non-dimensional form? It just makes the output harder to understand.

Author's reply 10

As you point out, presenting results in a non-dimensional form has advantages and drawbacks. Dimensional results are more intuitive and easier to compare to observations, but non-dimensional results are more general and can be scaled back easily using the scales provided in Eqns 7 and 9 to various configurations without having to re-run the model. Here, we prefer the generality of non-dimensionality since we are looking at generic benchmark cases instead of applying our model to a particular field site or comparing against specific field measurements.

Changes in the manuscript 10

–

Referee's comment 11

Section 5: There is some repetition of captions in the text. "In Figure 4, we plot. . ."; "Figure 5 shows. . ."; "Figure 6 shows. . ." etc. This could be skipped to make the text smoother.

Author's reply 11

Thank you for pointing this out.

Changes in the manuscript 11

We re-phrased Section 5.1 and 5.2 avoiding the figure caption repetition for better clarity. Please refer to the revised text in Section 5 for updates.

Referee's comment 12

line 308: Why are the benchmark tests performed at different resolutions? Does the GPU-model require order-of-magnitude more DOFs to yield the same accuracy as the FEM model? The comparisons give leave me with that impression, and then what is the advantage of the PT setup?

Author's reply 12

Thank you for raising this important point. The benchmark tests where originally run at higher resolutions with the FastICE GPU code since we can afford it. The Elmer/Ice results are obtained on the largest available single-core/direct solver resolution (or robust iterative solver for the 3D case). The latest results for the benchmark of experiment 2 show the good agreement among FastICE and Elmer/Ice at comparable resolutions. However, discrepancy between low and high numerical grid resolutions suggest that although the two different solution strategies match, they both may not fully capture the physics with accuracy at low resolutions in some cases, such as the 3D benchmark of Experiment 2. We report this issue in a new Figure 15 in the Section 5.5, showing the convergence of the numerical implementation among grid refinement.

Changes in the manuscript 12

Lines 441-458: We added a new Section "5.5: Validation of the FastICE numerical implementation" to discuss this topic and a related Figure 15.

Referee's comment 13 line 314: "numerical resolution grid resolution"

Author's reply 13

Thank you for pointing this out. We corrected the sentence. Which now reads:

Changes in the manuscript 13

Line 329: ". . .and used a numerical grid resolution. . ."

Referee's comment 14

line 333: The authors are right to address the discrepancies between the model results – but why not follow up on the idea to pin nodes in the FEM mesh?

Author's reply 14

We re-evaluated the benchmark test case using a comparable numerical grid resolution for our FastICE GPU solver and for Elmer/Ice. The result now agree for a particular numerical grid resolution. However, discrepancy with previous results suggest that the numerical resolution used to compare the two software may not be sufficient to resolve the physical process. To address this second limitation, we provide one additional figure showing the convergence of our method with and increase in numerical grid resolution and comparing the results to a high-resolution "reference" simulation.

Changes in the manuscript 14

We updated the Figure 7 with the latest benchmark test results at similar numerical grid resolutions between FastICE and Elmer/Ice and adapted the text from Section 5.2. Lines 441-458: We added a new Section "5.5: Validation of the FastICE numerical implementation" to discuss this topic and a related new Figure 15.

Referee's comment 15

Fig. 15: The performance diagrams are very convincing – however, the use of widely different DOFs for the FEM and PT models in the benchmark tests makes we wonder if the speedup is real?

Author's reply 15

The purpose of these graphs is not to report speed-up versus single-core Matlab or Elmer/ice, but to inform the reader about the potential and the scaling of the iterative and matrix-free PT approach to handle large number of grid points representative of high-resolutions simulations. In terms of high-performance “desktop” computing - what certainly majority of the researcher still rely on - it is fair to compare the range of affordable DOF for the FEM and PT implementations. Finally, high resolution calculations affordable with the PT approach may become necessary when resolving internal deformation localising into self-consistent formation of boundary layers prone to a sliding-like behaviour.

Changes in the manuscript 15

–

Sincerely yours,

Ludovic Räss, on behalf of the authors.

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2019-249, 2019.