Continental-Scale, High-Order, High Spatial-Resolution Ice Flow Modeling Based on Graphics Processing Units (GPUs)

Kenneth Mosley[1], Mathieu Morlighem[2], Anjali Sandip[1]
Department of Mechanical Engineering, University of North Dakota[1]; Department of Earth Sciences, Dartmouth College[2]

Motivation

Solving time-independent stress balance equations to predict ice velocity (or flow) is the most computationally expensive part of ice-sheet simulations in terms of computer memory and execution time.

Here, we develop numerical algorithms for graphics processing unit (GPU) acceleration of the Ice-sheet and Sea-level System Model's (ISSM's) (Larour et al.) 3-D Higher Order (HO) stress balance module and compare it with a standard CPU implementation.

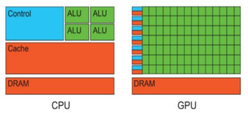


Figure 1. *Chip architecture for the modern CPU and the GPU. The unmarked blue, green, and orange boxes on the GPU chip architecture represent controlling units, cache memory, and the Arithmetic Logic Units. Picture courtesy of Rass et al.*

Model Configurations

To test the performance of the GPU implementation beyond simple idealized geometries, we apply it to two regional-scale glaciers in Antarctica: Thwaites and Pine Island Glacier, and then to the entire Antarctic Ice Sheet (AIS) with the configuration used for the Ice Sheet Model Intercomparison for CMIP6 activities (ISMIP6) (Seroussi et al.).

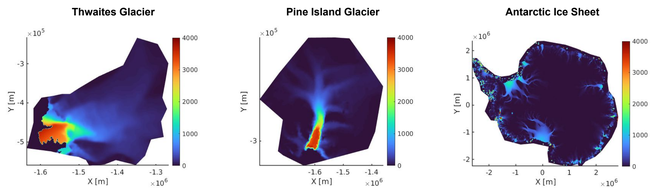


Figure 2. Glacier model configurations: observed surface velocities (in meters per year) interpolated on a uniform mesh. We apply water pressure at the ice front and nonhomogeneous Dirichlet boundary conditions on the other boundaries.

ISSM 3-D HO Stress Balance Model GPU Implementation

3-D Higher-Order (HO) stress balance approximation in the discretized form:

$$\boldsymbol{KV} = \boldsymbol{F}$$

where **K** is the stiffness matrix, ***F*** is the load vector and ***V*** is the vector of ice velocity.

Steps:

1. Transfer element metadata, basis functions, scalar material properties, global degrees of freedom, and offsets from the CPU to the GPU
2. Perform numerical integration and generate the values for **K**, and ***F*** on the GPUs
3. Assemble the matrix **K** and the vector ***F*** in the COO format suitable for GPU acceleration by implementing PETSc (Balay et al.) routines MatSetValuesCOO() and VecSetValuesCOO()
4. Solve for vector **V** on GPUs using PETSc's KSPsolve routines (CG/bjacobi) with multiple-cores enabled/Message Passing Interface (MPI)-parallelized

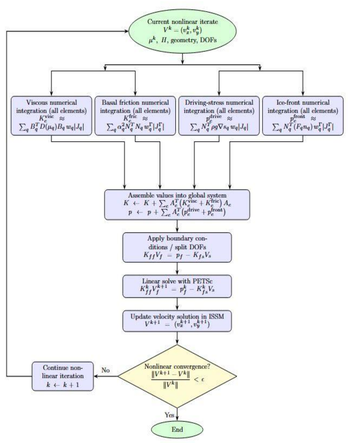


Figure 3. ISSM 3-D HO Stress Balance GPU Implementation, *for illustrative purposes only.*

Predictive Accuracy

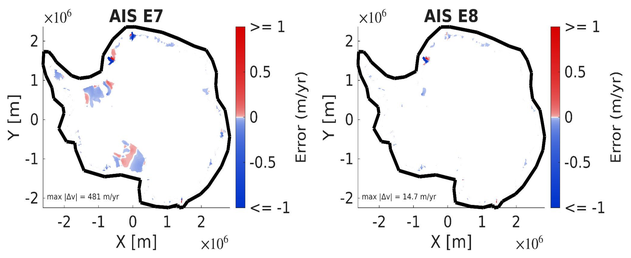


Figure 4. Performance comparison of the GPU implementation with the standard CPU implementation employing predictive accuracy for the AIS models. We use two configurations: one at the spatial resolution used for ISMIP6, with approximately 5×10^7 degrees of freedom (DoFs) (AIS E7), and a refined configuration, generated with ISSM's mesh tools, that increases the DoF count by a factor of four to approximately 2×10^8 DoFs and achieves sub-kilometer resolution at the grounding lines (AIS E8).

Speedup

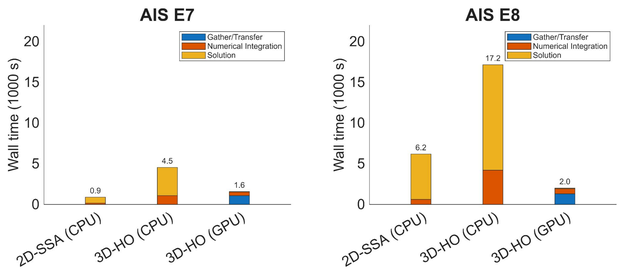


Figure 5. Performance comparison of the GPU implementation with the standard CPU implementation employing wall time for the AIS E7 and AIS E8 models, respectively. We obtain timings using NVIDIA A100 GPUs. CPU timing for both models use 1 node and 69 cores. Timings for AIS E7 and AIS E8 were obtained using 4 nodes (16 GPUs) and 16 nodes (64 GPUs) respectively. GPU device count was increased for E8 to accommodate its larger memory footprint; CPU memory was sufficient at 1 node for both problem sizes.

Scalability

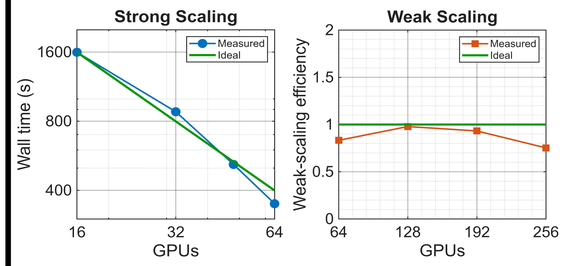


Figure 6. Strong scaling walltime and weak scaling efficiency of the GPU implementation. Strong scaling was performed using the AIS E7 model. Weak scaling was performed using the AIS E8 model, which is approximately 4x the size of the AIS E7 model, and using 4x the number of devices shown to achieve a comparable workload per device. Both experiments were performed using NVIDIA A100 GPUs.

Numerical Robustness

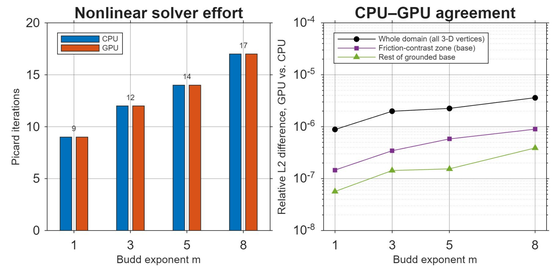


Figure 7. Numerical robustness test on Pine Island Glacier (3-D HO, diagnostic), Budd exponent m = 1, 3, 5, 8 with the friction coefficient rescaled pointwise from an m = 1 reference solution. (a) Picard iterations. (b) Relative L2 velocity difference, GPU vs. CPU: all 3-D vertices, and basal grounded vertices inside and outside the steepest friction contrasts.

Future Studies

Our next steps will include leveraging the accelerated GPU implementation to run high-fidelity simulation ensembles of ISMIP6-based projections of ocean-forced Antarctic Ice Sheet evolution. These findings have not been available due to computational costs; however, they are urgently needed, as the Antarctic Ice Sheet is losing mass at an increasing rate, and will significantly benefit process-oriented and sea-level projection studies over the coming decades.

Acknowledgements

We thank the National Science Foundation (Award # 2327095) and NVIDIA Academic Grant Program for supporting this project.

References

1. Balay, S., Abhyankar, S., Adams, M., Brown, J., Brune, P., Buschelman, K., ... & Zhang, H. (2019). PETSc users manual.
2. Larour, E., Seroussi, H., Morlighem, M., & Rignot, E. (2012). Continental Scale, High Order, High Spatial Resolution, Ice Sheet Modeling Using the Ice Sheet System Model (ISSM). *Journal of Geophysical Research: Earth Surface*, *117*(1), F01022. https://doi.org/10.1029/2011JF002140
3. Räss, L., Licul, A., Herman, F., Podladchikov, Y. Y., & Suckale, J. (2020). Modelling Thermomechanical Ice Deformation Using an Implicit Pseudo-Transient Method (Fastice V1. 0) Based on Graphical Processing Units (GPUs). *Geoscientific Model Development*, *13*(3), 955-976. https://doi.org/10.5194/gmd-13-955-2020
4. Seroussi, H., Nowicki, S., Payne, A. J., Goelzer, H., Lipscomb, W. H., Abe-Ouchi, A., Agosta, C., Albrecht, T., Asay-Davis, X., Barthel, A., Calov, R., Cullather, R., Dumas, C., Galton-Fenzi, B. K., Gladstone, R., Golledge, N. R., Gregory, J. M., Greve, R., Hattermann, T., et al. (2020). ISMIP6 Antarctica: A multi-model ensemble of the Antarctic ice sheet evolution over the 21st century. The Cryosphere, 14(9), 3033–3070. https://doi.org/10.5194/tc-14-3033-2020