A PROTOTYPE IMPLEMENTATION OF AN EMBEDDED SIMULATION SYSTEM FOR THE STUDY OF LARGE SCALE ICE SHEETS

Phillip Dickens
Christopher Dufour
James Fastook

School of Computing and Information Science
University of Maine
Boardman Hall
Orono, Maine 04469 USA

ABSTRACT

Understanding the impact of global climate change on the world's ecosystem is critical to society at large and represents a significant challenge to researchers in the climate community. One important piece of the climate puzzle is how the dynamics of large-scale ice sheets, such as those covering Greenland and Antarctica, will respond to a warming climate. Relatively recently, glaciologists have identified ice streams, which are corridors of ice that are flowing at a much higher rate than the surrounding ice, as being crucial to the overall dynamics and stability of the entire ice sheet. However, ice stream dynamics, and their impact on the entire ice sheet, are far from understood. It is thus critical to develop simulation models through which these important issues can be studied. In this paper, we present one novel approach to developing such simulation capabilities through the use of *embedded simulation*.

1 INTRODUCTION

Ice streams are fast-moving corridors of ice that originate in the interior of an ice sheet and drain into the sea through what are termed *outlet glaciers.* They have very different dynamics than the rest of the ice sheet, and need to be modeled at a much higher resolution to develop better insight into their behavior and impact on overall ice sheet dynamics. This represents a significant challenge for ice sheet modelers, because even small increases in resolution often result in tremendous increases in both computational costs and the sizes of the input/output data sets that must be handled by the simulation. For example, modeling the Greenland ice sheet at a resolution of 5-km requires on the order of 34 million grid points and data sets on the order of 1-GB. Increasing to the 1-km resolution, however, increases the number of grid points to over 1.6 billion, and increases the size of the data sets to over 28GB. Modeling at a resolution of 500 meters, which appears to be a minimum resolution for the study of ice streams, would increase the number of grid points in the simulation from 1.6 billion to over 6.7 billion. It would similarly increase the size of the input/output files to over 125GB. While these file sizes may be manageable if output is only written to disk at the end of the run, it is often the case that they need be saved on a much more frequent basis, such as when creating animations of ice sheet dynamics over time and when performing basic state saving for error recovery.

Given these difficult challenges, we are pursuing a different technique using *embedded simulation.* In this approach, areas of the ice sheet undergoing rapid change are modeled at the highest resolution available, while the interior portions of the ice sheet, whose dynamics are evolving much less rapidly, are modeled at lower resolutions. In fact, we believe it is somewhat more intuitive to view the modeling problem in terms of multiple physical processes, executing at different spatial and temporal resolutions, each of which can interact with and impact the behavior of the other physical processes. In this paper, we present our approach that provides the synchronization and communication mechanisms that enable such simulation modeling.

We are developing our approach using the Parallel Ice Sheet Model (Torsten Albrecht, et. al, 2015), which is a widely used parallel simulation model for the study of large-scale ice sheets. It involves implementing multiple, independent instances of a full PISM simulation, utilizing the power of the full PISM model when possible, and augmenting it with additional mechanisms to accurately model and capture the important feedback loops between them.

We believe this paper provides two important contributions to the modeling and simulation community. First, it provides a clear proof of concept for this basic approach. This is demonstrated through experimental results that document the accuracy of the overall simulation, and that provide strong indications that it is able to capture important feedback loops between the simulation processes.

Second, while the focus of this paper is on large-scale ice sheet modeling, we believe the basic approach will have much broader applicability. In particular, large scientific modeling applications, where the domain is laid out across a checkerboard grid and includes processes logically executing at different special and temporal resolutions, could derive significant benefits from using this approach.

The rest of the paper is organized as follows. In Section 2, we provide an overview of PISM and its approach to modeling ice dynamics. In Section 3, we describe our approach to the development of embedded simulation modeling. This includes a discussion of the data sets used in this research, the basic building blocks necessary to implement this approach, and the synchronization and communication between the simulation models required to correctly capture their interactions while maintaining coherence of the global simulation. In Section 4, we discuss our experimental design, and we provide our experimental results in Section 5. We discuss related research In Section 6, and provide our conclusions and future research in Section 7.

2 THE PARALLEL ICE SHEET MODEL (PISM)

PISM is a powerful and highly flexible parallel simulation model that provides researchers with a window into past, present, and future climatic events. It provides both the Shallow Ice Approximation (Le Meur et al. 2004), and the Shallow Shelf Approximation (Bueler and Brown, 2004), for the resolution of stress balance equations. It also provides the hybrid ‘SSA + SIA’ model (Winkelmann et al. 2011), which incorporates elements of both models in the solution of stress balance equations. It provides ways to customize and combine such models to fit the problem at hand, and makes available a wide range of models for marine ice physics, ocean calving, and conservation of energy. PISM also provides the ability to easily couple the simulation with external ocean and atmospheric models.

Within PISM, the model takes place in a rectangular computational box that consists of a collection of data points in three dimensions representing the space that encloses the glacier being studied. In each of the x and y dimensions, the grid points are equally spaced, and have a relatively coarse resolution. The z dimension is given a relatively finer resolution than x or y, and the spacing of grid points along this dimension may vary within a given model allowing for more detail near the base of the ice where driving forces are greatest. Each x, y pair represents a single column of ice that is parallel with the force of gravity.

PISM takes the entire computational box and divides it into n rectangular sub-grids, where n is the number of processes being used for the simulation. It attempts to make the sub-grids as square as possible in the x and y dimensions because the calculation of a new value for a given grid point often only depends on the current values of variables at adjacent grid points. Therefore, the degree to which the computation of one sub-grid depends on results from another sub-grid is proportional to the perimeter of the local grid, and the square has the minimum perimeter for any rectangle. However, for many computational box sizes and values of n, there is no way to evenly distribute the grid points to n non-intersecting squares. In such cases, PISM arranges the sub-grids into r rows and c columns with n = rc , with no row being more than one grid wider than any other row, and no column being more than one grid point taller than any other column.

A PISM simulation progresses in units of time steps, which define the logical time at which the step begins and ends. The length of the time step is determined cooperatively by all simulation components, and represents the maximum advance in simulation time that is consistent with the preservation of basic numerical and physical consistency constraints. PISM utilizes an interpolation process between neighboring grid points to define the *initial,* or *boundary conditions* for the current time step, and then models the evolution of ice dynamics from that point forward to the end of the time step. For reasons made clear below, it is important to note that the values of all simulation variables are not fully resolved until the end of a given time step, and are thus not available in any intermediate form until that point.

PISM derives its computational scalability from its use of the Portable Extensible Toolkit for Scientific Computation (PETSc, Satish et al., 2015, 2015a, 1997), which is a widely used library of data structures and routines for scientific models that require partial differential equations for their solutions. PETSc, in turn, derives its scalability by spreading its computation across multiple processing cores and using MPI (The Message Passing Interface Standard) for inter-process communication.

3 APPROACH

3.1 High Resolution Data Sets

The Jakobshavn outlet glacier, located on the western coast of Greenland, is the fastest flowing glacier in the world (Moulins, 2008), through which approximately 7% of the entire ice sheet is discharged into the ocean (Torsten Albrecht, et. al, 2015). It is also important in that it experienced a significant calving event quite recently in 2015, and has retreated over 35KM in the last 150 years. Thus the dynamics of this particular ice stream/outlet glacier are critically important to understand, but must be modeled at high-resolution to gain insight into its dynamical processes. One primary reason for this is that the Jakobshavn region has very deep trenches, which have a significant impact on ice velocities and flows, but are not wide enough to be detected at lower resolutions.

To develop a better understanding of the dynamics of such ice streams/outlet glaciers, the Center for Remote Sensing of Ice Sheets (CReSIS, https://www.cresis.ku.edu/), has been collecting and making available high-resolution data sets for such areas of the ice sheet undergoing rapid change. The most recent data sets published by CReSIS for the Jakobshavn region (https://data.cresis.ku.edu/data/), include bedrock topography at a 500-meter resolution. These are the data sets that form the basis of the high-resolution embedded simulation model used in this research.

3.2 Simulation Regions

Our prototype implementation utilizes a 500-meter resolution model of the Jackobshavn region embedded within a 2KM model of the entire ice sheet. These are instantiated as two complete, independent PISM models, both of which are fully parallelized and whose behavior is dictated by the underlying PISM system. Thus both PISM models are correct and, in isolation, correctly synchronized by their respective runtime systems.

However, the models share the region within the domain of the high-resolution embedded simulation, and have overlapping grid points in this region of the grid. It is within this area that one simulation can *directly* impact the *model state* of the other, requiring careful synchronization to correctly capture their interactions while maintaining coherence of the global simulation.

The overlapping region of the grid is partially depicted in Figure 1, where the ‘x’ characters represent high-resolution grid points, all of which are contained within the embedded region, and the ‘o’ characters represent grid points of the low-resolution model. The high-resolution grid points are fully contained within the embedded region while the low-resolution grid points cover the entire domain, including the embedded region. For the purposes of this discussion, we show only the low-resolution points within and immediately adjacent to the embedded domain.

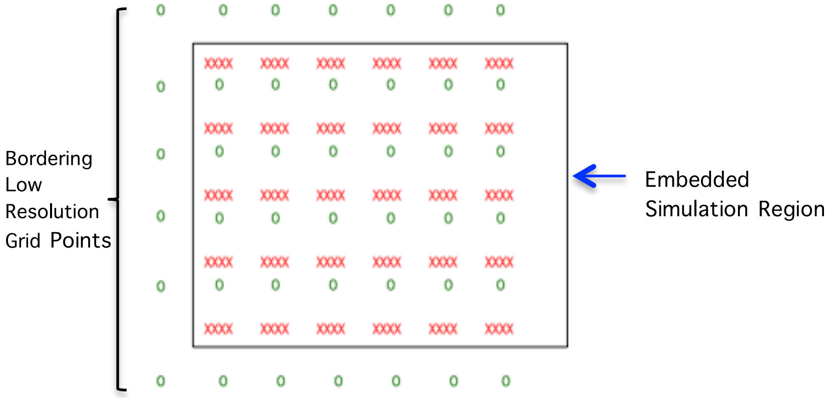

Figure 1: This figure shows the distribution of high- and low-resolution grid points within the high-resolution embedded region of the grid. The high-resolution grid points are depicted as ‘x’, and the low-resolution grid points are depicted as ‘o’.

3.3 Components of the Embedded Simulation Model

In this section, we define the building blocks of our approach that are put into their proper perspective when we discuss the two-phase synchronization protocol below.

Recall that the components of the PISM model cooperatively compute new model boundary conditions at the beginning of each time step through the application of an interpolation algorithm applied to each grid point and its nearest neighbors. We use a similar approach in the computation of the boundary conditions for both the high- and low-resolution simulations.

First, consider the high-resolution simulation. New boundary conditions are computed for the high-resolution grid points lying at the border of the embedded region. The four nearest low-resolution neighbors for each point are determined, taking two such neighbors from the grid points immediately outside of the embedded region and the two closest neighbors within the embedded region. We then compute the initial conditions using a weighted linear interpolation of values from the chosen neighbors. Thus while several high-resolution grid points may share the same set of low-resolution neighbors, the impact of each neighbor is weighted by its relative distance from the high-resolution point under consideration. This process is depicted in Figure 2.

The process of computing the initial boundary conditions for the next time step, which utilizes interpolated values from the low-resolution simulation, represents one important way in which the state of the low-resolution simulation impacts the state of the high-resolution simulation. This feedback between the simulations is propagated throughout the high-resolution domain as a consequence of PISM’s modeling of the evolution of ice dynamics from that point forward to the end of the current time step.

The initial boundary conditions for the low-resolution grid points lying within the embedded region are computed in a similar manner. In particular, each low-resolution point within the region determines it’s closest four neighboring high-resolution points, and computes its boundary conditions using the weighted linear interpolation of these neighbors. This represents one important way in which the high-resolution simulation impacts the state of the low-resolution simulation.

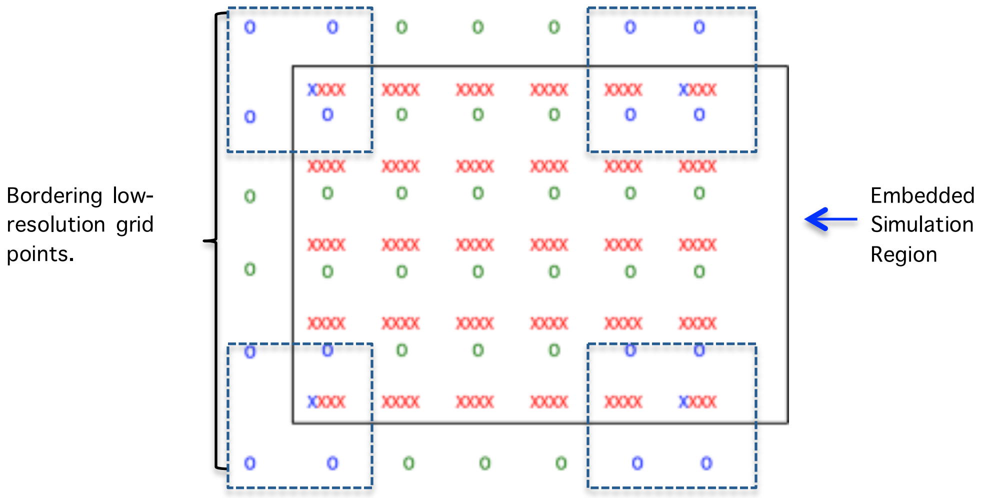

Figure 2: This figure partially depicts the process of establishing initial boundary conditions for the high-resolution grid points. The high-resolution points are depicted as 'x', and the low-resolution grid points are depicted as 'o'. For each high-resolution grid point along the border of the embedded region the four closest low-resolution points are selected. The initial boundary conditions are then established utilizing a weighted linear interpolation between these neighbors. For example, consider the four high-resolution points represented as 'x'. The figure shows the corresponding low-resolution grid points that will be used for the interpolation (represented as 'o').

3.4 Two Phase Simulation Protocol

Assume the simulations are synchronized at some logical time T1 representing the end of the current time step. At this point, the high-resolution simulation sends the current values of all of the low-resolution grid points within the embedded region to the low-resolution simulation. Similarly, the low-resolution grid nodes within and immediately bordering the embedded region are provided to the high-resolution simulation. Once this communication is completed, the two-phase simulation protocol begins.

In Phase 1 of the protocol, the grid points provided by the high-resolution simulation are (logically) dropped into the corresponding grid domain of the low-resolution simulation. After this point, the simulation is completely controlled by PISM's computational processes, which have no knowledge of a shared region of the grid, and there is no further interaction between the two simulation models until the current time step is completed. Thus PISM uses its own interpolation mechanisms to establish the boundary conditions for the next time step, and then models the evolution of the ice dynamics in the normal fashion up to the end of the time step. It is important to emphasize that while PISM is unaware of any feedback from the high-resolution simulation, such feedback is already incorporated via the values of the grid points that have been impacted by the execution of the high-resolution simulation model.

Once the low-resolution simulation reaches the end of the time step, it provides to the high-resolution simulation the most recent values of its grid points within the overlapping region.

These values are then used to establish the initial boundary conditions for the high-resolution simulation.

Phase 2 of the protocol is a bit more complex because of the differing temporal and special resolutions at which they are operating. In particular, modeling at higher resolutions requires smaller time steps to resolve the underlying dynamics. This means that the high-resolution simulation will have to execute more, smaller time steps, to reach logical time T2. However, it only has state information from the low-resolution model at logical time T1, and updated values of the low-resolution grid points will not be available until time T2. Thus it can establish boundary conditions for the first time step, but not for the subsequent time steps it will execute to reach logical time T2.

This issue is addressed in Phase 2 of the protocol, which requires that the high-resolution simulation block during the execution of the low-resolution time step. Once the low-resolution simulation reaches logical time T2, it provides its updated values to the high-resolution simulation. Given information about the state of the overlapping grid points at logical times T1 and T2, we use a weighted linear 2D interpolation between these values in the computation of boundary conditions for the subsequent time steps. This process continues until it reaches logical time T2, at which point it provides to the low-resolution simulation the values it needs to compute the boundary conditions for the next time step. This process is repeated until the end of the simulation.

To summarize, the feedback loop between the models is captured through the values of the grid points they exchange that are utilized in the computation of initial boundary conditions. The impact of this feedback on the evolution of ice dynamics is propagated throughout the respective simulations through the subsequent execution of the PISM model.

4 EXPERIMENTAL DESIGN

All of our experimental results were obtained utilizing the SuperMic cluster located at the High Performance Computing Center at Louisiana State University. It consists of 360 compute nodes, each of which includes two 10-core Intel Xeon Ivy Bridge-EP E5-2680 processors operating at a rate of 2.8 GHz. Each node also provides two Intel Xeon Phi 7120P coprocessors that have not yet been incorporated into our initial implementation. Each node is equipped with 64GB DDR3 1866MHz Ram, and a 500GB hard drive. It is attached to the rest of the system via a 56 Gigabit per second Infiniband network interface. Each node runs the Red Hat Enterprise 6 operating System.

We tested our prototype implementation using a 2KM model of the entire Greenland ice sheet with an embedded model of the Jakobshavn outlet glacier region at the 500-meter resolution. Our primary goal was to provide a proof of concept for the basic approach we are pursuing. To help answer this question, we wanted to compare ice velocities computed by our simulation with those obtained through direct observation and measurement. We also wanted to demonstrate the impact of model resolution on simulation results in this critical domain. Finally, we wanted to establish baseline performance metrics to gage the impact of various optimization techniques under development.

5 EXPERIMENTAL RESULTS

We first wanted to compare the ice velocities computed by our simulation with those obtained from direct observation and measurement (Joughin et. al, 2010). The computed and observed velocities were taken from a 10KM cross-section of the ice flow within the Jakobshavn outlet glacier. The measurements were taken between the grounding line, where ice begins to float, and the Calving Front, which is where floating ice detaches from the ice sheet. The results are shown in Figure 3.

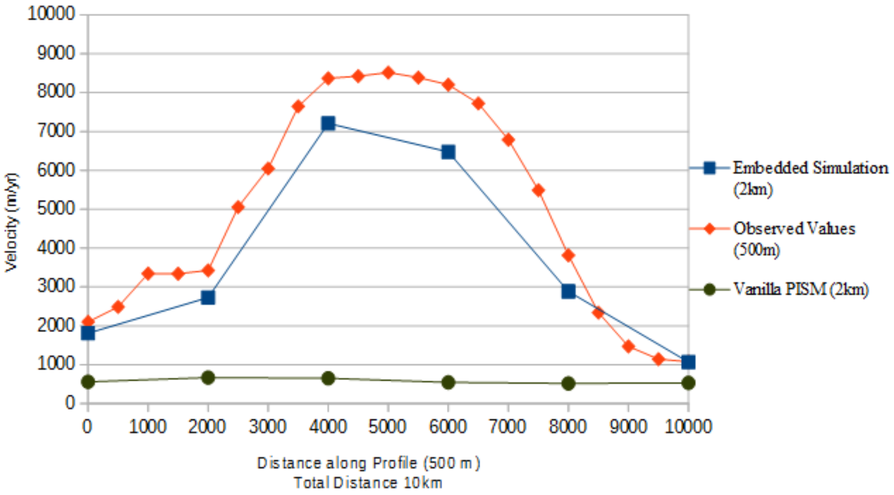

Figure 3: This graph shows the observed ice velocities (top), the simulated velocities using the embedded simulation (middle), and the velocities computed by the non-embedded simulation.

The top line in the graph represents the observed ice velocities sampled at 500-meter resolution. The next line shows the values of the *low-resolution grid points* that fell within the embedded region. For comparison purposes, we also included the simulated velocities computed by a 2KM non-embedded simulation.

As can be seen, the velocities computed using embedded simulation are reasonably close to the observed values, and even begin to capture the basic shape of the curve of observed values toward the middle of the cross section. It is very interesting to note that the results clearly demonstrate a feedback loop between the high- and low-resolution simulations operating within the embedded region of the grid. This can be seen by comparing the results provided by the embedded simulation, which represent the values of the low-resolution grid points within the embedded region, to the third set of results, which represent the same grid points as computed by the 2KM whole ice sheet model. The differences in these velocities are directly attributable to the impact of the high-resolution simulation on the values of the low-resolution grid points.

5.1 Computational Costs

We also wanted to provide a baseline measure of the computational costs incurred by the prototype implementation of our system. We measured this by executing both a non-embedded, 2KM simulation of the whole Greenland ice sheet and the same model with the embedded simulation. We executed both models for one logical year. The results are provided in Table 1.

Table 1: This table provides basic performance characteristics of our prototype embedded simulation system.

Performance Metrics	**2KM Vanilla**	**2KM with 500-meter embedded**
Number of processors	700	350 per model
Total Run time	300.44s	6373.20s
Total Wait time	0s	1037.90s
Total Number of time steps	277	8694

The most striking result is the fact that it took approximately 21 times as long to execute the 2KM embedded simulation model as it did to execute the non-embedded model. There are several contributing factors leading to the increased computational costs. One such factor is a 31-fold increase in the number of time steps required to complete the simulation. And while an increase in the number of time steps was certainly expected, the magnitude of the increase was somewhat surprising. However, this is a cost that will be incurred by any simulation model using a time stepping algorithm to resolve dynamic processes at high resolution.

The other contributing factors are largely related to the initial implementation techniques. One problem is that the system does not yet support processor sharing between simulations, and, in this example, simply divided the total number of available cores equally between them. Because of the blocking nature of the simulation protocol, this led to lengthy wait times during which one half of the computational resources were idle. We discuss our strategies for addressing these issues in Section 7.

6 RELATED WORK

The PISM developers are also investigating solutions to these same challenges. One approach that has been implemented within PISM is the ability to perform *regional* simulation models at high resolutions (Torsten Albrecht, et. al, 2015). Similar to the ideas outlined in this paper, their goal is to only simulate at high resolution in regions of the ice sheet undergoing rapid change. However, in their approach the high-resolution simulation executes in isolation, and there is no mechanism to feedback results from the high-resolution region of the grid to other regions of the ice sheet.

A recently published paper (Aschwanden, 2016), describes a whole Greenland ice sheet model, executing at a constant 600-meter resolution, which is implemented within PISM. The model was executed out 100 years, and produced simulated ice velocities, for a number of outlet glaciers across Greenland, which fit observed values quite well. Unfortunately, the computational costs of their approach, including the costs of performing I/O at such high resolutions, were not discussed in the paper. We are working on obtaining such results, but were unable to do so by the time of this writing.

Other available parallel ice sheet models with higher-order capability include the Ice Sheet System Model (ISSM, (Larour et al. 2012)) and we have pursued incorporating CReSIS high-resolution bed-maps into this model. One advantage of ISSM is its ability to generate and use a variable-resolution grid. ISSM provides interpolation tools that take regular checkerboard grids (rows and columns, uniformly spaced and aligned with the coordinate axes) such as those provided by CReSIS, and based on some external metric (usually measured velocity if available, although it can be specified manually), provide non-uniform triangular element grids that provide higher resolution where needed, and lower resolution where it is not. Similar to the paper describing the 600-meter whole ice sheet model, we have been unable to find detailed discussions of the computational costs of the ISSM approach.

While we have thus far discussed our research in terms of numerical simulation modeling, the approach we are pursuing begins to incorporate some of the basic synchronization techniques and ideas found within the parallel discrete-event simulation community. Viewed in this way, our approach is essentially an enhanced window-based synchronization protocol quite similar to the YAWNS protocol (Nicol, 1993). Given the high blocking costs incurred by our two-phase synchronization protocol, it becomes attractive to think in terms of integrating optimistic synchronization techniques, e.g., Bounded Time Warp (Turner and Xu,, 1992), into our embedded simulation system.

7 CONCLUSIONS AND FURTURE RESEARCH

In this paper, we have presented our design and prototype implementation of a high-resolution simulation for regions of an ice sheet undergoing rapid change, embedded within a lower-resolution model for regions of the ice sheet that are evolving at a much slower rate. We have described our approach to integrating the results from simulations that are operating at different spatial and temporal resolutions, while maintaining the coherence of the overall system. We demonstrated one important feedback loop within the shared region of the grid, and showed that the computed ice velocities compare quite well, although not yet perfectly, to the observed values. We believe that other, larger feedback loops will also be captured with extended simulation runs. Overall, we are pleased with the success of the prototype.

There is of course much work remaining, the most important of which is reducing the computational costs of the prototype implementation. There are three basic approaches under investigation. First, we are conducting very basic load-balancing experiments to determine the number of cores to allocate to each simulation to minimize the wait time. Second, we are in the process of incorporating the two Intel Xeon Phi 7120P coprocessors into the system. This would serve to provide additional (free) computational resources as well as more extensive opportunities for load balancing. Third, we are interested in implementing gang scheduling such that both simulations utilize the same set of computational resources. This would largely eliminate blocking, but this would come at the expense of increased context switching costs.

We are also interested in embedding multiple high-resolution simulations within a low-resolution whole ice sheet model. We believe this will represent a strait-forward extension of the current approach, and could provide a much deeper understanding of underlying physical processes being modeled.

Our longer-term goal is to utilize optimistic simulation techniques to further reduce cost of blocking that is inherent in conservative simulation approaches. However, more research is needed to determine the opportunities for optimistic execution, and to define the error conditions that would require rolling back to an earlier state.

REFERENCES

Aschwanden, Andy, M. Fahnestock, and M. Truffer. "Complex Greenland outlet glacier flow captured." *Nature communications* 7 (2016).

Bueler,Ed, and J. Brown. "Shallow shelf approximation as a “sliding law” in a thermomechanically coupled ice sheet model." *Journal of Geophysical Research: Earth Surface* 114, no. F3 (2009).

“Data Products – File Listing”, Center for Remote Sensing of Ice Sheets (CReSIS), accessed February 17, 2016, https://data.cresis.ku.edu/data/.

Joughin, I., B. Smith, I. Howat, T. Scambos, and T. Moon. 2010. Greenland Flow Variability from Ice-Sheet-Wide Velocity Mapping, *Journal of Glaciology*. 56. 415-430.

Larour, E., H. Seroussi, M. Morlighem, and E. Rignot. "Continental scale, high order, high spatial resolution, ice sheet modeling using the Ice Sheet System Model (ISSM)." *Journal of Geophysical Research: Earth Surface* 117, no. F1 (2012).

Le Meur, Emmanuel, O. Gagliardini, T. Zwinger, and J. Ruokolainen. "Glacier flow modelling:comparison of the Shallow Ice Approximation and the full-Stokes solution." *Comptes Rendus Physique* 5, no. 7 (2004): 709-722.

“The Message Passing Interface standard.”, Argonne National Laboratory (ANL), accessed April 2, 2016, http://www.mcs.anl.gov/research/projects/mpi/.

“Moulins, Calving Fronts and Greenland Outlet Glacier Acceleration”, accessed April 1, 2016, http://www.realclimate.org/index.php/archives/2008/04/moulins-calving-fronts-and-greenland-outlet-glacier-acceleration/comment-page-5/

Nicol, David M. "Global synchronization for optimistic parallel discrete event simulation." In *ACM SIGSIM Simulation Digest*, vol. 23, no. 1, pp. 27-34. ACM, 1993.

"PISM, a Parallel Ice Sheet Model", accessed June 13, 2015, http://www.pism-docs.org, 2015

"PISM: Parallel Ice Sheet Model User Manual.", accessed February 13, 2016. http://www.pism-docs.org/wiki/lib/exe/fetch.php?media=manual.pdf

Satish, Balay., S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. McInnes, K. Rupp, B. Smith, S. Zampini, and H. Zhang. "PETSc Web page." (2015).

Satish, Balay., S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. McInnes, K. Rupp, B. Smith, S. Zampini, and H. Zhang. "Petsc users manual revision 3.5. Technical report", Argonne National Laboratory (ANL), 2015.

Satish, Balay., W. Gropp, L. McInnes, and B. Smith. "Efficient management of parallelism in object- oriented numerical software libraries." In *Modern software tools for scientific computing*, pp. 163-202. Birkhäuser Boston, 1997.

"SuperMIC", Louisiana State University (LSU), accessed April 23, 2016, http://www.hpc.lsu.edu/resources/hpc/system.php?system=SuperMIC

Towns, John, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. Peterson, R. Roskies, and J. Scott. "XSEDE: accelerating scientific discovery." *Computing in Science & Engineering* 16, no. 5 (2014): 62-74.

Turner, S. and Xu, M. Performance Evaluation of the Bounded Time Warp Algorithm. In *Proceedings of the 6th Workshop on Parallel and Distributed Simulation. 1992.*