Summer 8-22-2016

Improving the Performance of Ice Sheet Modeling Through Embedded Simulation

Christopher G. Dufour
University of Maine, christopher.dufour@maine.edu

IMPROVING THE PERFORMANCE OF ICE SHEET MODELING THROUGH EMBEDDED SIMULATION

By

Christopher G Dufour

B.S. University of Maine, 2012

A THESIS

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

(in Computer Science)

The Graduate School

The University of Maine

August 2016

Advisory Commitee:

Phillip M. Dickens, Professor of Computing and Information Science, Advisor

James L. Fastook, Professor of Computing and Information Science

Max J. Egenhofer, Professor of Computing and Information Science

THESIS ACCEPTANCE STATEMENT

On behalf of the Graduate Committee for Christopher Dufour I affirm that this manuscript is the final and accepted thesis. Signatures of all committee members are on file with the Graduate School at the University of Maine, 42 Stodder Hall, Orono, Maine.

Phillip Dickens, Professor | August 10, 2016

LIBRARY RIGHTS STATEMENT

IMPROVING THE PERFORMANCE OF ICE SHEET MODELING

THROUGH EMBEDDED SIMULATION

By Christopher G Dufour

Thesis Advisor: Dr Phillip Dickens

An Abstract of the Thesis Presented
in Partial Fulfillment of the Requirements for the
Degree of Master of Science
(in Computer Science)
August 2016

Understanding the impact of global climate change is a critical concern for society at large. One important piece of the climate puzzle is how large-scale ice sheets, such as those covering Greenland and Antarctica, respond to a warming climate. Given such ice sheets are under constant change, developing models that can accurately capture their dynamics represents a significant challenge to researchers. The problem, however, is properly capturing the dynamics of an ice sheet model requires a high model resolution and simulating these models is intractable even for state-of-the-art supercomputers.

This thesis presents a revolutionary approach to accurately capture ice sheet dynamics using embedded modeling at a high resolution. Such an approach embeds a high-resolution ice sheet model of a region evolving rapidly within a low-resolution ice sheet model of areas evolving slowly. The embedded model approach was implemented within the Parallel Ice Sheet Model (PISM), a widely used model for the study of large scale ice sheets limited to simulating models in isolation. PISM is limited to simulating

ice sheet models in isolation and thus implementing an embedded model requires new synchronization and communication schemes. In this work we analyze the accuracy of our prototype embedded model with respect to directly observed ice velocities. We have shown a stronger correlation to directly observed values, yielding a T-test value of 0.64, compared to a non-embedded model T-test of 0.02.

ACKNOWLEDGMENTS

This thesis was supported in part through NSF Grant Number 1229716. The work in this thesis used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1053575. We acknowledge the Texas Advanced Computing Center and Louisiana State University for providing Supercomputers for use by the procedures detailed in this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Understanding the impact of global climate change is a critical concern for society at large. One important piece of the climate puzzle is how large-scale ice sheets, such as those covering Greenland and Antarctica, respond to a warming climate. Given such ice sheets are under constant change, developing models that can accurately capture their dynamics represents a significant challenge to researchers [1]. One of the critical issues that must be addressed with such models is resolution, which is the level of detail at which the physical processes are modeled. For example, ice streams, which are corridors of ice that are flowing at a much higher rate than the surrounding ice, have been identified as critical to the overall dynamics and stability of the whole ice sheet [2,31,37]. For this reason, it is critical to model their interactions. One factor that makes the modeling of ice streams difficult is that they have to be modeled at a high resolution to accurately capture their dynamics.

The current approach to understanding such interactions is to model the entire ice sheet at the resolution of the ice streams. Due to the size of the resulting data sets supercomputers must be used for efficient execution. The problems with this approach can be seen in Figure 1.1, which displays the number of model data points as a function of resolution. A resolution change from ten kilometers to half of a kilometer for the Greenland ice sheet increases the number of data points from 8.4 million to 6.7 billion. Similarly, this change in resolution causes the memory requirements to increase from sixteen gigabytes to thirteen terabytes. Simulating one model year using half a kilometer resolution on Stampede, the tenth fastest supercomputer in the world, requires a minimum of 4,096 processors and 7.5 hours of execution time [3]. This demonstrates that models of this size and complexity are nearly intractable even for state-of-the-art supercomputers.

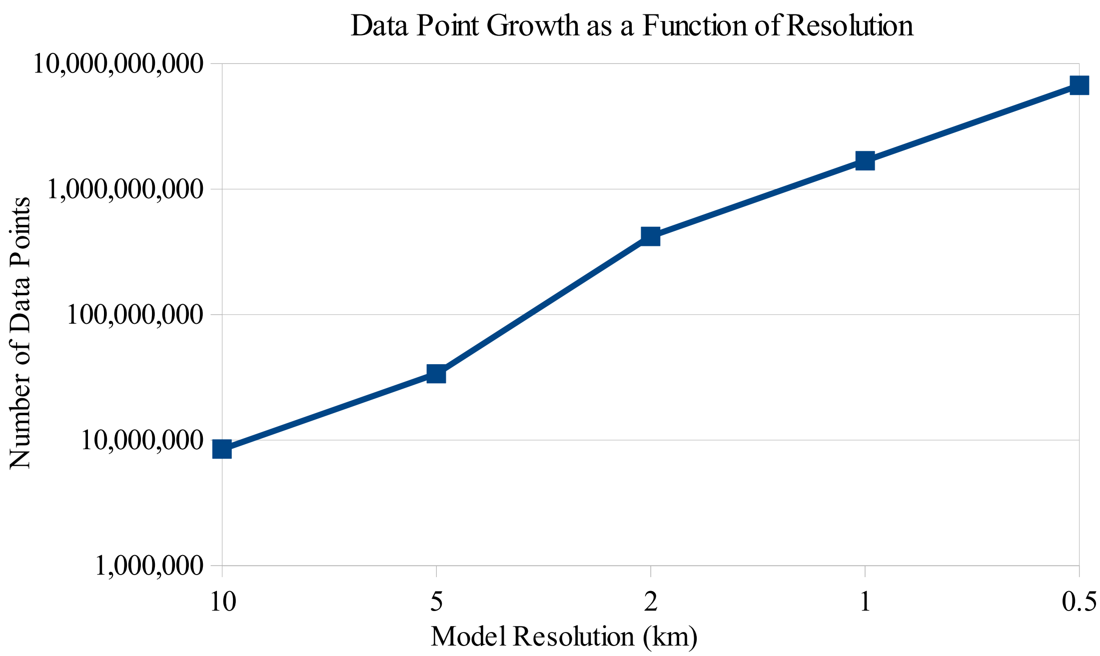

Figure 1.1: Data Point Growth as a Function of Resolution. Number of data points in an ice sheet model as a function of model resolution.

This thesis develops an alternative approach using embedded simulation to understand the interactions between ice streams and ice sheets. In this approach areas of the ice sheet undergoing rapid change are modeled at a high resolution, while areas that are changing more slowly are modeled at a lower resolution. This approach yields results comparable to a high resolution model of an entire ice sheet, but at the benefit of requiring fewer data points and, therefore, less computational resources. We base our approach on the Parallel Ice Sheet Model (PISM), a widely used model for the study of large-scale ice sheets [4,5,6,37]. While PISM does provide the opportunity to develop regional models of rapidly changing areas, such models can only be executed in isolation. Thus the important feedback between ice streams and the underlying ice sheet are lost. Our approach embeds concurrent high-resolution ice stream models within an existing low-resolution model of the entire ice sheet. The challenge of this new approach is that it requires careful synchronization between the models and the development of new mechanisms for communication between them.

The remainder of this thesis is organized as follows. In Chapter 2 represents a survey into ice sheet models and their simulation mechanisms. Chapter 3 represents a survey into ice sheet modeling in general, examining both ice dynamics and introducing PISM. In Chapter 4 we describe our challenges involved with implementing an embedded simulation in PISM, as well as our solutions to them. Chapter 5 shows our experimental setup used in testing the embedded model, while Chapter 6 compares both the overall results as well as the space and time complexity between the non-embedded PISM simulation and our embedded simulation. Finally, Chapter 7 summarizes our findings as well as describes future development.

CHAPTER 2

RELATED WORK

There are many forces acting upon ice and it is critical to model them properly. The basis for all physical laws in an ice sheet is the Stokes flow law, which models all forces affecting ice including vertical drag [7]. The Stokes flow law provides the most comprehensive model of physics on an ice sheet, but does so at high computational costs [7]. Simulating a high-resolution full ice sheet model using the Stokes flow law is intractable for even state-of-the-art supercomputers. To overcome this limitation on the Stokes flow law, researchers have removed certain forces that do not significantly impact the ice sheet. For example, the Blatter-Pattyn high-order model neglects vertical stress gradients, which do not impact the ice significantly [8]. While dropping this one force increases performance, growing error is introduced to the model making long-term simulation infeasible.

To promote long-term simulation further refinements to the Stokes flow law are needed, such as Shallow Ice Approximation (SIA), which is a common variation of the Stokes model that drops all forces except gravity [9]. This approximation only models the force of gravity against the force of friction leaving the remaining minor forces as constants [9]. This simplification assumes all ice is grounded. Since there are no longitudinal forces to represent floating ice, SIA is unsuitable to outlet glacier modeling [10]. Floating ice is critical to a complete ice sheet model and it is imperative that it be modeled properly.

The Shallow Shelf Approximation (SSA) is the answer to modeling floating ice efficiently. Rather than modeling gravitational forces, SSA focuses on longitudinal stresses, which are the driving forces on floating ice [10]. SSA does not model basal sheer friction, since gravitational forces are completely balanced by buoyancy forces from water.

SIA and SSA are both suited to long-term modeling of ice sheets, but cannot properly model all parts of the ice sheet. To solve this issue some ice sheet models combine SIA and SSA flow laws, where SSA is applied to areas with floating ice while SIA is used for grounded ice [11]. However, the SIA+SSA approach continues to suffer from propagating errors due to neglect of some forces. SIA+SSA models only the most significant forces, such as gravitational and longitudinal stressors, while having superior performance to other flow laws (Table 2.1.).

Table 2.1: Flow Law Comparison

Flow Law	**Forces Modeled**
Stokes Higher Order Model	Gravitational Driving Stress, Basal Drag, Longitudinal Stresses, Vertical Stressors, Lateral Drag
Blatter-Pattyn Higher Order Model	Gravitational Driving Stress, Basal Drag, Longitudinal Stress, Vertical Stressors.
SSA + SIA	Gravitational Driving Stress, Basal Drag, Longitudinal Stress
SSA	Longitudinal Stress
SIA	Gravitational Driving Stress, Basal Drag

Table 2.1: Flow Law Comparison. Comparison of examined flow laws in terms of the forces they model. The flow laws are listed from most computationally complex to least computationally complex.

Early ice sheet models, such as the Glimmer ice sheet model, have focused on implementing the full Stokes flow law [12]. Glimmer suffers from poor performance, because it simulates ice conditions serially and thus cannot take full advantage of modern computers. Glimmer is also limited to using the higher-order flow laws, such as Stokes or Blatter-Pattyn, which further degrades performance.

Modern ice sheet models, such as The Community Ice Sheet Model (CISM) , attempt to improve on the Glimmer model by supporting lower-order flow laws, such as SSA in addition to higher-order flow laws [13]. CISM also allows models to fully utilize modern computers by simulating models using higher-order flow laws in parallel. Lower-order flow laws, such as SSA, must still be simulated serially, however.

While most modern ice sheet models provide support for different flow laws, each approaches the modeling paradigm differently. One of the more prolific ice sheet models is the Ice Sheet System Model (ISSM), which expresses a data set as a series of data points forming triangles [14]. A model constructed from triangles forms a mesh where areas of more rapid change, such as the edges of an ice sheet, can contain more data points effectively increasing the accuracy of simulating those regions. The resolution of these models are therefore non-uniform, allowing ISSM to selectively model features at a high resolution. ISSM also supports the use of modern flow laws such as the full Stokes model and SIA+SSA approximations.

The focus of this thesis is the Parallel Ice Sheet model. PISM is a powerful parallel model that can simulate ice sheet conditions far into the future [15,16]. PISM utilizes the SIA+SSA hybrid flow law for stress balance while also providing tools to customize models to fit the needs of researchers. PISM expresses a data set as a three dimensional grid of points within a rectangular computational domain (Figure 2.1) [15,16,17]. The resolution, which is the distance between data points, is uniform in the horizontal dimensions as opposed to ISSM. The ice sheet model is divided more intuitively however, with the computational load better distributed across the model. The data points in the vertical dimension are at a higher resolution closer to the basal layer where complex physics affect the ice sheet more readily. An example of this can be seen in Figure 2.2, where there are more data points at the basal layer.

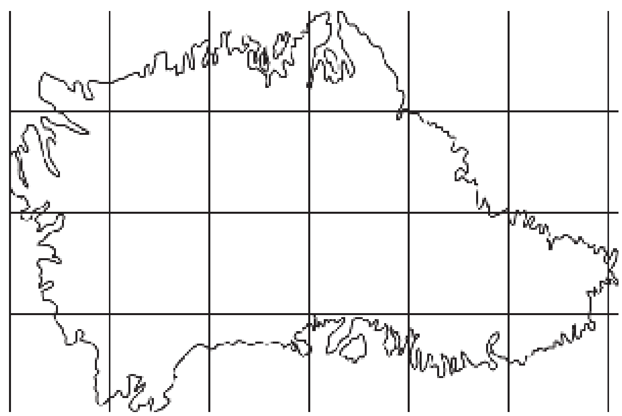

Figure 2.1: PISM Subdomain Division. An example data set of Greenland that is divided into twenty-four equally sized subdomains. Each of these subdomains is then operated on by a single PISM process.

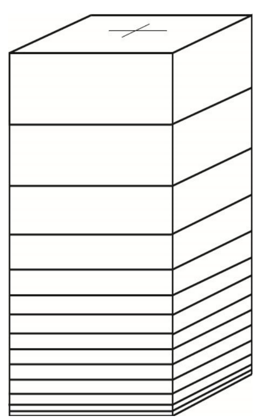

Figure 2.2: PISM Vertical Dimension. Three dimensional vertical representation of a PISM data point, where each vertical data point is separated by a black line. Note the decreasing distance between data points as we approach the basal layer at the bottom

Like ISSM, PISM uses parallel processes and divides the data set grid between each process. The differences between PISM processes and those of the other ice sheet models can be seen in Table 2.2. While PISM provides a highly scalable implementation of an ice sheet model it does so in isolation. Because each model is required to be at a single resolution this makes long-term high-resolution simulation of a whole ice sheet intractable.

Table 2.2: Ice Sheet Model Comparison

Ice Sheet Model	**Processor Parallelism**	**Data Resolution**
Glimmer	Serial only	Single resolution
CISM	Parallel with higher order flow laws only; serial otherwise	Single resolution
ISSM	Parallel	Multiple resolutions
PISM	Parallel	Single resolution

Table 2.2: Ice Sheet Model Comparison. Comparison of examined ice sheet models with special attention paid to the level of parallelism and their handling of data resolutions.

CHAPTER 3

ICE SHEET MODELING

In this chapter we go into detail on the dynamics of ice sheets as well as the challenges involved in modeling them. An analysis of PISM's methods for operating on a model and simulating it in parallel is also made here.

3.1 Ice Sheets

Ice sheets are broad layers of ice covering terrain in polar regions formed from layers of ice that are added over thousands of years [18,19]. They are under constant impact by climate conditions often resulting in melted ice migrating downhill under gravitational forces [18,19]. This interaction is important to researchers because ice discharge is a factor in changing sea levels.

The movement of ice is not uniform across the entire ice sheet and some regions are subject to significantly higher velocities than others. For example, consider ice streams, such as the Jakobshavn ice stream (Figure 3.1), which are narrow corridors of ice flowing at a rate approximately eighty times higher than the velocity of surrounding ice. The ice moving through ice streams represent 90% of all ice discharged from an ice sheet and failure to model these ice streams can lead to an inaccurate representation of ice sheet dynamics [2]. This phenomenon represents a variation of the Modifiable Areal Unit Problem (MAUP), which states that errors are created when data is grouped for analysis [35, 36]. Because the impact of ice streams on an ice sheet does not correspond to its size it is critical that data points exist within the ice stream, which is only possible at high resolutions. Therefore it is critical to develop scientific models that can properly model ice streams with respect to an entire ice sheet.

Modeling ice streams represents a challenge to researchers due to their relatively small size compared to the remainder of the ice sheet. Ice streams can be as narrow as one kilometer and data points must be present within this span to properly model their interactions. Data sets at low resolution may not contain any data points that lie within an ice stream (Figure 3.2). Developing models that can simulate ice streams at a high resolution are imperative to researchers.

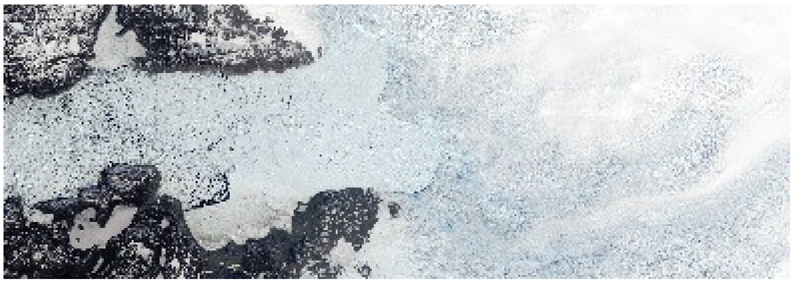

Figure 3.1: Jakobshavn Ice Stream Outlet. The Jakobshavn ice stream at the calving front where velocity is at its highest.

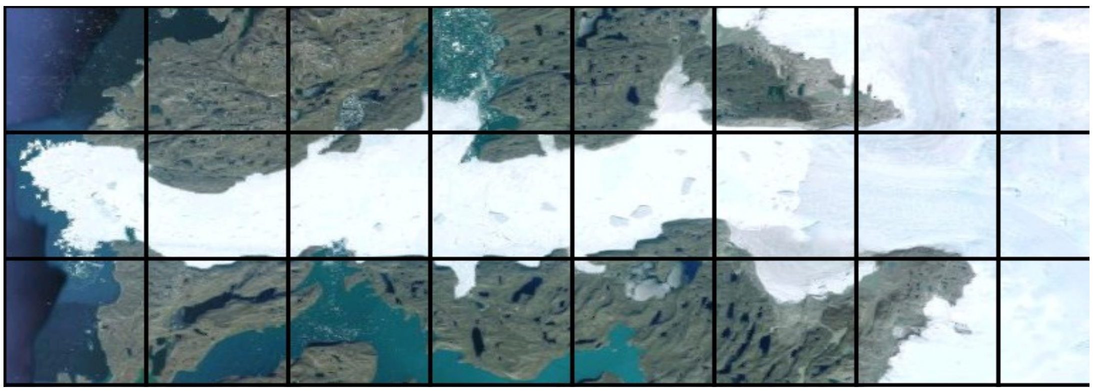

Figure 3.2: Low Resolution Jakobshavn Grid. An example configuration of a low-resolution ice sheet model of an ice stream which runs right to left through the middle of the image. Each line intersection is the location of a data point. Note that while the most significant ice velocity is found in the ice stream there are no data points lying within it, causing this important factor to be lost in simulation.

3.2 Impacts of Ice Streams

Understanding the influence an ice stream and the entire ice sheet have on each other is critical to predicting overall ice sheet conditions. PISM's models, which are performed in isolation, handle these influences through internal simulation mechanisms. In our embedded model, however, it is imperative that these impacts be well defined so that they may be communicated between ice sheet models of different spatial resolutions.

The most immediate impact of an ice stream on the entire ice sheet is the total ice discharge [2, 31, 37]. Recall that ice streams are responsible for the majority of ice discharge in an ice sheet, and their velocity can drastically impact conditions across the entire ice sheet. For example, capturing a high velocity can impact the whole ice sheet by reducing the total ice thickness throughout the ice sheet. Failure to simulate an ice stream at a high resolution results in a poor handling of this impact, and an inaccurate representation of ice discharge for the entire ice sheet [20, 37].

Ice streams are also impacted by conditions throughout the ice sheet. Changes to ice thickness throughout an ice sheet can cause frictional forces to be reduced as the glacier becomes buoyant further inland. Such a phenomenon causes ice to be discharged at a higher rate and thus increases the ice velocity in ice streams [21, 37]. Because ice thickness throughout an ice sheet is updated by velocity in an ice stream there exists a positive feedback loop between ice streams and ice sheets (Figure 3.3). Capturing all of these impacts is critical to properly modeling this feedback loop and thus properly modeling long-term conditions on an ice sheet.

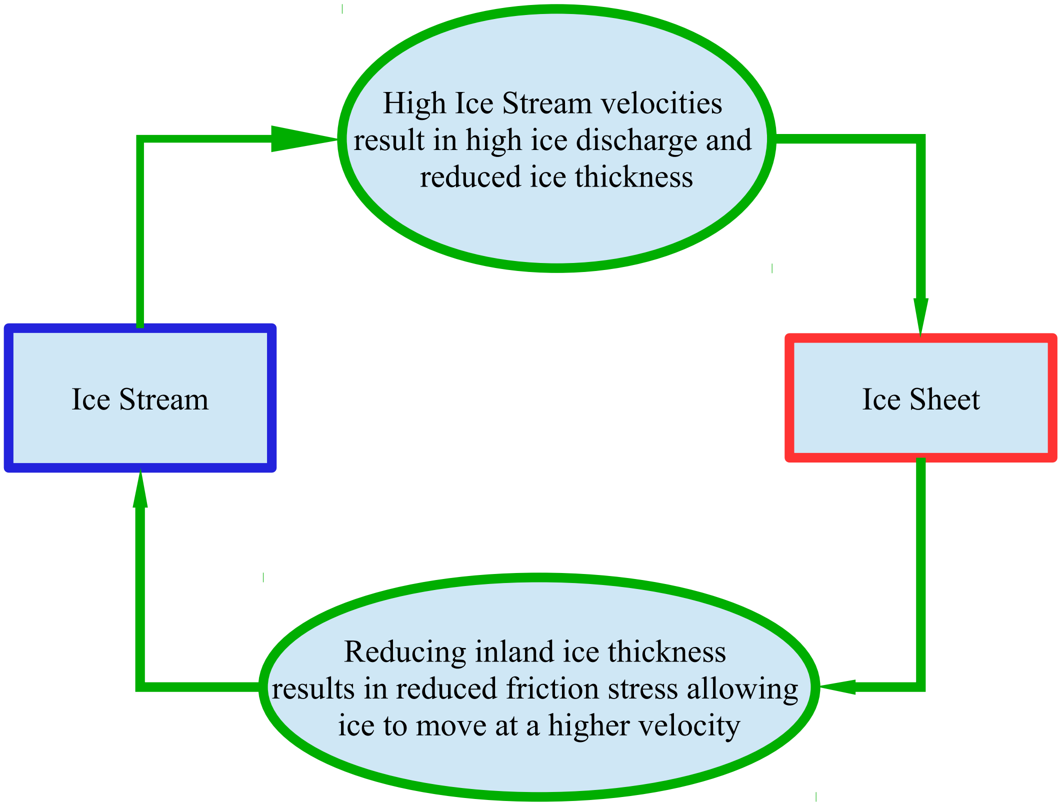

Figure 3.3: Positive Feedback Loop Visualized. The positive feedback loop affecting ice streams and the whole ice sheet.

3.3 PISM

PISM is a powerful parallel model that can simulate ice sheet conditions far into the future as well as incorporate a variety of physical processes. Within PISM a data set is expressed as a three-dimensional grid of points within a rectangular computational domain where the horizontal dimensions contain data points that are uniformly spaced. Each horizontal data point represents variables at a given location on the ice sheet, such as ice velocity. The vertical dimension has a higher resolution that is not uniform since some elevations, such as the basal layer, require more data points to properly model physics [15,16,17].

The computational domain is divided into a number of interdependent rectangular subdomains based on the number of processes used for the simulation. Each subdomain is dependent on values from its neighbors, with data points on the borders requiring updates from other subdomains making it critical that borders are as short as possible to minimize required communication. However, many combinations of input data sets and processes do not support an equal division into square subdomains. In these cases PISM will allow for an additional row or column such that no subdomain will be more than one row wider or one column deeper.

A PISM simulation is progressed in adaptive timesteps, which are the logical time between the start and end of a simulation step. The length of a time step is dependent on the maximum amount of time each subdomain can be simulated before encountering causality errors. To avoid these errors, every subdomain must completely update its own data points based on its neighbors periodically. Before modeling the ice dynamics on the ice sheet, each subdomain must update their boundaries by interpolating values from neighboring subdomains. This process is critical to the model as it allows the impacts of each subdomain to be conveyed to its neighboring subdomains and thus impact the entire model.

Although PISM provides a strong foundation for modeling an ice sheet in isolation, it is necessary to introduce a new synchronization scheme to support simultaneous interacting simulations. Subdomains on the borders of each simulation must be able to communicate with each other in order to capture interactions between the simulations. Embedding a model creates situations where the interiors of subdomains within one simulation may impact the boundaries of another and it is critical that these cases be handled (Figure 3.4).

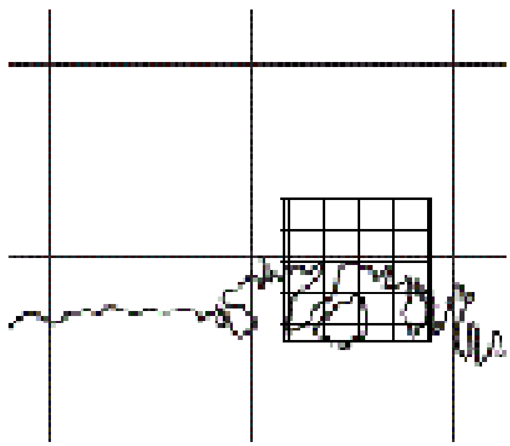

Figure 3.4: Embedded Region Related to Low Resolution Domain. An example of an embedded model where the embedded model sub-domains require input from one or more low resolution sub-domains.

CHAPTER 4

CHALLENGES

The basis of our embedded model approach involves multiple discrete simulations operating within the same time and space. This presents a challenge, because these models share the same state and thus directly impact each other during simulation. Developing a strategy to capture the impact of an ice sheet model while preserving fidelity is critical, but presents a significant challenge. To this end we model the impact through updates to each of the embedded model initial boundary conditions only, which are then propagated throughout the model by existing PISM mechanisms. It is critical that the interior portions of the embedded models are not updated directly otherwise a significant loss of fidelity will result from overwriting sensitive high-resolution regions, such as ice streams, with low-resolution data.

Such a procedure is difficult due to the differing temporal resolutions of the embedded ice sheet models and low-resolution whole ice sheet model. The embedded models require input from the low-resolution whole ice sheet model several times as they simulate up to the same logical time. This presents a challenge because that input is only available at the end of the low-resolution whole ice sheet models' simulation step. We, therefore, apply one-dimensional linear interpolation between data points at the beginning and end of a given simulation step to determine the state of the low resolution whole ice sheet model (Figure 4.1). This procedure empowers our embedded models to update their borders based on the impact by the low-resolution whole ice sheet model at any logical time, however it is only possible when the ice sheet models are synchronized.

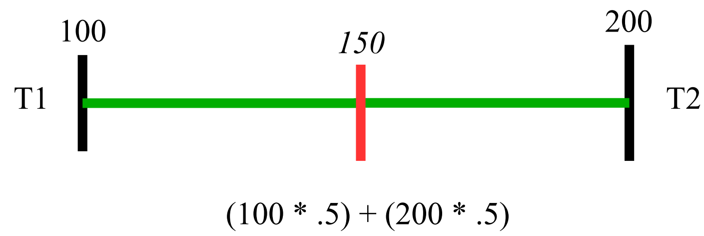

Figure 4.1: Temporal Interpolation Procedure. Depiction of an example of our temporal interpolation. The black lines at either end represent logical times T1 and T2, where the low resolution whole ice sheet model began and ended its time step. The red line and value, which is present at the midpoint between T1 and T2, represents an example embedded model time step which requires interpolation to determine the low resolution state at that time.

Synchronizing all ice sheet models is critical to conveying the impact of models but requires substantial modifications to PISM. Our approach, termed “Two-Phase Synchronization Protocol”, allows us to force the low-resolution whole ice sheet model to halt simulation while waiting for updates from the slower embedded models. In our first phase, the low-resolution whole ice sheet model, starting at logical time T1, is allowed to take a single simulation step ending at logical time T2. After this step the second phase of the protocol begins where the embedded models, using input from the low-resolution whole ice sheet model at logical times T1 and T2, is allowed to take as many simulation steps as necessary to reach logical time T2. Once all embedded models have reached logical time T2, they then convey their state back to the low-resolution whole ice sheet model, which is now allowed to continue for another simulation step, repeating the entire process until completion of the simulation (Figure 4.2).

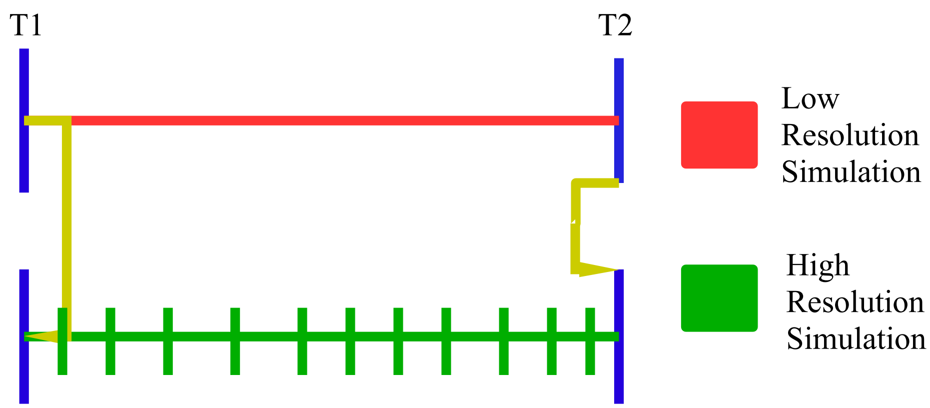

Figure 4.2: Two-Phase Synchronization Protocol. While the embedded high-resolution simulation requires updates several times between T1 and T2, communication only occurs when the models are synchronized at T2.

With a synchronization protocol implemented, we must also establish a mechanism through which the embedded model borders can be updated. The embedded models and the low-resolution whole ice sheet model are of different spatial resolutions, which presents a challenge to updating border conditions. While PISM implements linear interpolation for its models in isolation, it assumes each data point is uniformly spaced thus simplifying calculations. This assumption cannot be made when communicating data points of a different spatial resolution to a model (Figure 4.3). The border points of the embedded model cannot all have the same Cartesian coordinates as points within the low resolution whole ice sheet model making a new interpolation strategy a requirement.

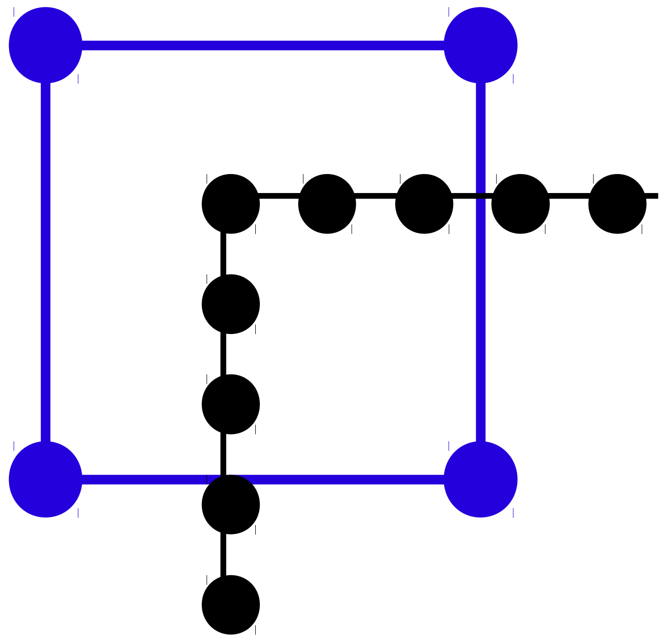

Figure 4.3: Embedded Border Data Point Neighbors. A corner of our embedded model (black) in relation to low resolution whole ice sheet model data points (blue). To update any of the embedded border points with whole ice sheet data, a new interpolation scheme is needed, because the distance of each embedded border point between its neighbors is different for each point. The differing spatial resolutions also require that we operate with the same neighbors for many different embedded border points, such as those contained within the blue rectangle.

Performing our cross-model interpolation requires the use of common Cartesian coordinates, which we can use to identify the nearest four neighboring low-resolution points. Because our models are operating at differing spatial resolutions we cannot rely on PISM's internal data structures. A new procedure for finding neighbors was developed using the Cartesian coordinates of each data point instead, where an iterative process is initially used to find each low-resolution neighbor (Figure 4.4).

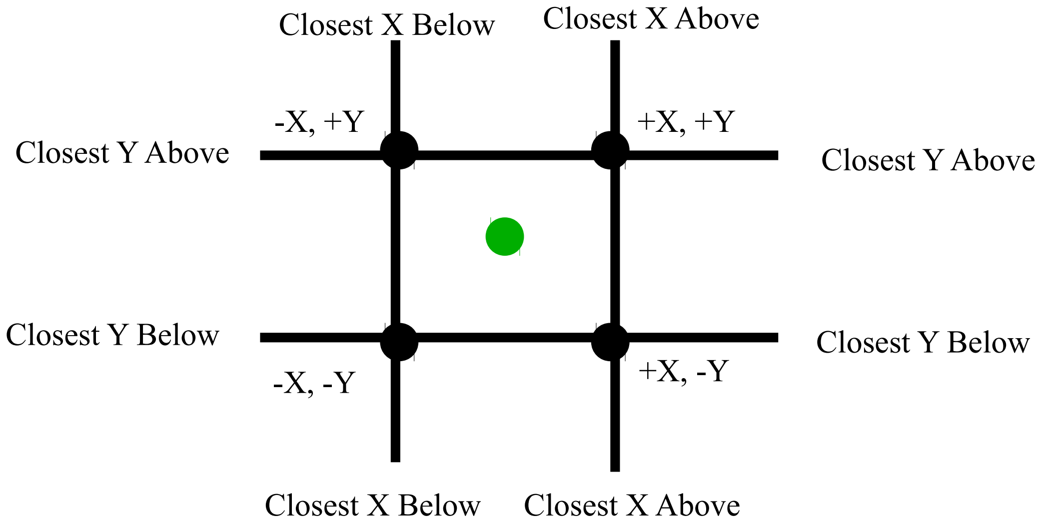

Figure 4.4: Neighbor finding scheme. Here we are searching for the low-resolution neighbors, shown as black dots, of an embedded model data point, shown as a single green dot. We first draw four lines through the model representing the closest X and Y values that are above and below the coordinates of our targeted embedded model data point. The intersections of these four lines represents the closest neighbors of the embedded model data point.

In this procedure we attempt to generate the smallest possible square around a target embedded model data point by searching for the closest low-resolution whole ice sheet data points with Cartesian coordinates above and below the target. Such a procedure, while computationally expensive, is required only at initialization, because the position of data points cannot change horizontally, thus allowing us to cache the neighbors of every border points in each embedded model for quick reference. After each neighbor is identified it is now possible to use two-dimensional linear interpolation to derive the new value of a high resolution data point based on four low-resolution neighbors. This interpolation is shown in Figure 4.5, where an embedded model border point is updated based on four low-resolution data points.

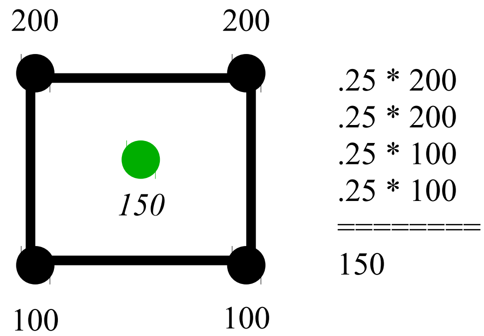

Figure 4.5: Two-Dimensional Linear Interpolation Strategy. We assume an embedded model data point exists in the exact center between each of the neighboring low-resolution whole ice sheet model data points. Because of this assumption we can weigh each neighbor point equally at 0.25, which is multiplied with their data value and summed to yield the interpolated value of the embedded data point. If the embedded data point was closer to any one neighbor its interpolation weight would grow while the others would decrease. Interpolation weights must always sum to 1.

CHAPTER 5

EXPERIMENTAL DESIGN

In this chapter we describe our modifications to PISM designed to address the challenges in utilizing an embedded model as well as the parameters we used for our simulations.

5.1 Geographic Projection

Before any ice sheet data set is ready for simulation it must first be expressed in a geographic projection, which is a conversion between latitude and longitude and a Cartesian plane. Such a conversion is critical, because ice sheet data sets must be expressed on a flat plane. Simply applying latitude and longitude pairs here would not work because the Earth is an oblate ellipsoid and the distance between each point of latitude and longitude is not uniform. It is imperative that the embedded models have the same projection as the whole ice sheet model, which presents a challenge to researchers since there are over seven thousand geographic projections in use and the probability of finding two data sets with the exact same geographic projection is small.

There are tools to convert a data set from one projection to another, but this presents new problems since the conversion process often disrupts the uniformity of the spacing between the data points (Figure 5.1). Because PISM requires all data points to be uniformly spaced, the following procedure had to be developed. We first use readily available tools, such as Proj4, to convert each data point in one projection to another individually, yielding a grid of non-uniform points [22]. We then use two-dimensional linear interpolation to generate a new, uniformly spaced grid of data points, which is embedded onto a data set in the target geographic projection. Because the conversion process generates a new grid that is tilted with respect to the target data set we use a simple cropping procedure to remove the edges of the embedded data set. The removal of the edges of the embedded data set is required to yield a rectangular high resolution grid ready for use in PISM (Figure 5.1).

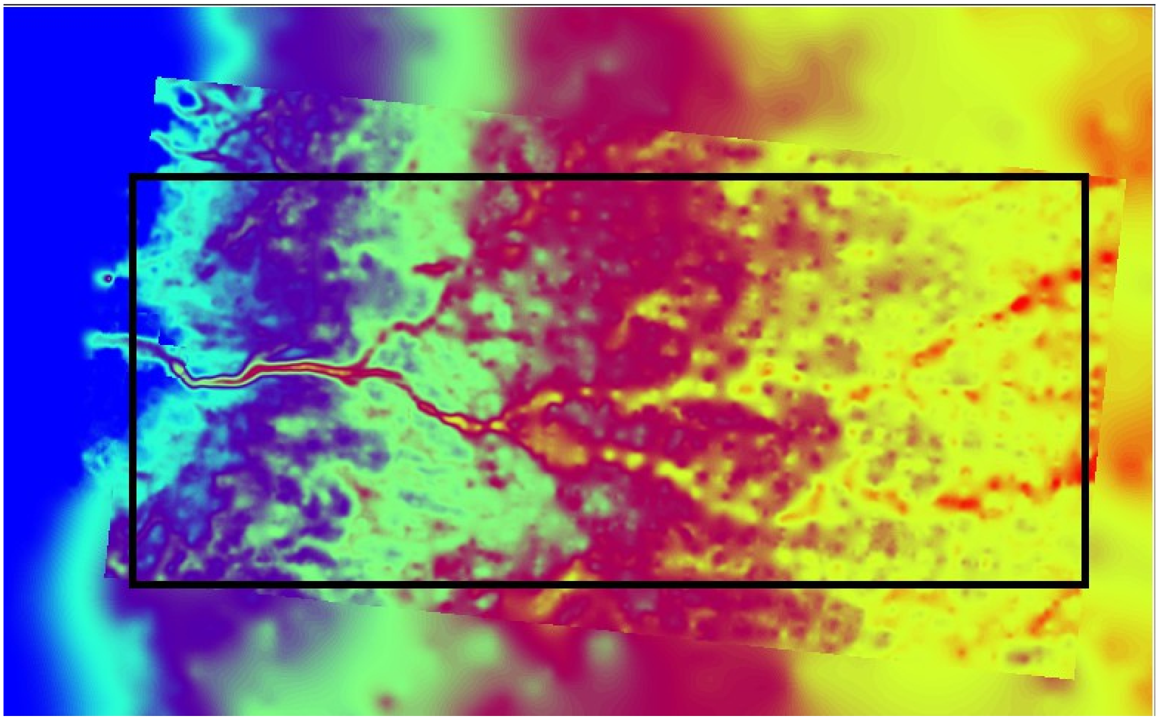

Figure 5.1: Embedded Model After Projection Conversion. A heatmap of ice thickness with our embedded portion imposed over a low-resolution data set of the same data set. The embedded portion is tilted with respect to the low-resolution data set as a result of our projection conversion mechanism. It is critical that our embedded data set does not contain any low-resolution data, therefore, our final embedded data set crops everything outside of the black box.

5.2 Message Passing Interface

PISM derives its computational scalability from the Portable Extensible Toolkit for Scientific Computation (PETSc), which is a widely used library for scientific models using partial differential equations [23,24,25]. PETSc itself utilizes the Message Passing

Interface (MPI) for spreading computations across multiple cores [26]. Recall that PISM divides an ice sheet model such that each allocated process is assigned an equal portion to simulate. Such a critical procedure is governed by MPI functionality and modifying it is necessary for the creation of an embedded model.

In order to use an embedded model it is critical that PISM be modified to operate on multiple models inside a single execution. Our goal is to instantiate multiple models for multiple different input data sets instead of instantiating a single model. This represents a significant challenge because PISM is designed to simulate ice sheet models in isolation with a single MPI communicator governing the model.

To minimize the effects on PISM's established design we opted to approach this problem by using existing MPI functionality. For each input data set read into PISM a new MPI communicator is spawned which receives an equal share of processes allocated to PISM to simulate ice sheet models. These normally discrete MPI communicators are then connected using an overarching MPI intracommunicator which will be used for synchronizing the ice sheet models. This approach is visualized in the MPI process hierarchy, where each discrete model is now joined through an intercommunicator (Figure 5.2). This crucial modification allows PISM to operate on as many interconnected ice sheet models as desired.

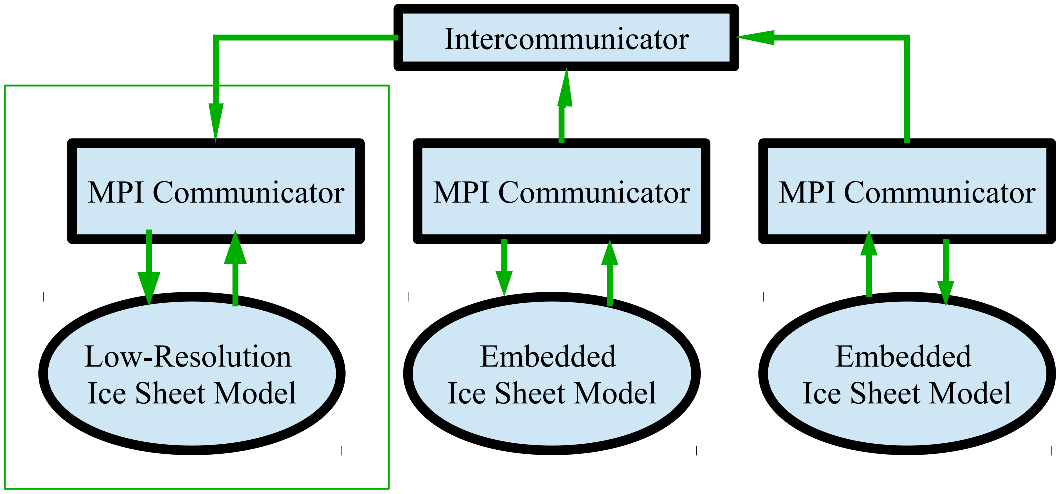

Figure 5.2: New MPI Hierarchy. This figure shows our MPI hierarchy in use by our embedded model. The green box represents the base implementation in PISM. To implement an embedded model we add multiple additional communicators and connect them all with an intracommunicator. In this case we are presenting the flow of data from the embedded model through the intracommunicator to the low resolution whole ice sheet model.

With a new MPI communication scheme in place and the ability to simulate multiple ice sheet models in parallel, our previously defined synchronization protocol can now be implemented. Recall that implementing a synchronization protocol in PISM is challenging since PISM is designed to simulate in isolation. It is, therefore, imperative that we implement synchronization using the previously mentioned MPI intracommunicator and existing MPI functionality to avoid disturbing internal PISM mechanisms.

To synchronize our ice sheet models we first force each model to be fully initialized before any modeling begins since different data sets require varying amounts of pre-processing. This is done primarily through the use of blocking calls, such as MPI_Barrier, which forces processes responsible for simulating one model to wait until every other model has caught up to the same simulation milestone. By using barriers within our intracommunicator it is possible to force faster models to halt simulation ensuring all models have passed certain milestones, such as initialization. Blocking communications, such as MPI_Recv, are also used to force one ice sheet model to halt until it has received data from another model. This occurs during the simulation and is crucial to ensuring no communication takes place unless models are at equal logical time.

It is imperative that a method to communicate ice sheet data be implemented within PISM using existing MPI functionality to avoid disrupting internal simulation mechanisms. While MPI implements methods that can communicate variables between processes it does so with limitations, such as the inability to send complex objects between processes, which is troublesome because ice sheet models within PISM are expressed as specialized two dimensional vectors [15,16,17]. To overcome this limitation it is critical to repackage ice sheet data in a manner that can be easily sent via MPI. Rather than attempting to send a two-dimensional vector object, we convert the data into a contiguous one-dimensional vector of MPI_Double values. This vector is then easily sent using MPI calls (Figure 5.3), where it is then converted back into a two-dimensional vector object to be used in interpolation.

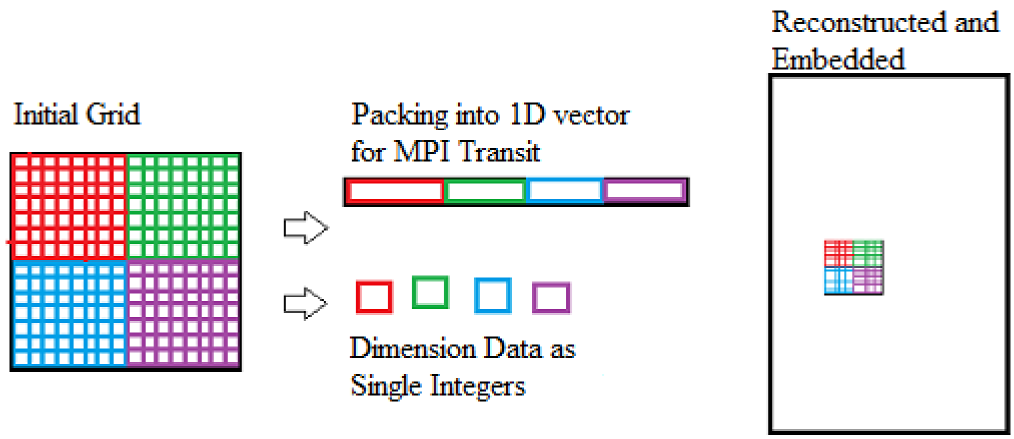

Figure 5.3 Data Communication Scheme. We gather all values of the subdomains (shown as different colors) into a one-dimensional vector, which is sent along with the dimensions of each subdomain using simple MPI functionality.

5.3 Hardware

For our prototype embedded model we utilized the SuperMic computing cluster located at the High Performance Computer Center at Louisiana State University [27,32]. SuperMic consists of 360 compute nodes, which include two 10-core Intel Xeon Ivy Bridge-EP E5-2680 processors operating at a rate of 2.8 GHz. Each of SuperMic's compute nodes also include 64GB DDR3 1866MHz Ram with a 500gb hard drive.

SuperMic uses Terascale Open-source Resource and Queue Manager (TORQUE) as part of its program job system. TORQUE is an open-source extension of the Portable Batch System (PBS), which is software that acts as a job scheduler and resource allocator for shared systems. TORQUE allows users to request full control of a number of nodes for a given task. This is advantageous for our work with PISM because it allows us to dedicate processors wholly to PISM without concern for concurrent usage by other researchers [27,28,32].

Our prototype implementation was tested using a Greenland ice sheet model at a two-kilometer resolution from the Sea-level Response to Ice Sheet Evolution (SeaRISE) group, which is a community initiative dedicated to understanding ice sheet regions under rapid change [29]. We are embedding a five-hundred-meter resolution model of the Jakobshavn outlet glacier generated from the Center for Remote Sensing of Ice Sheets (CreSIS), which is a research group focused on new methods of measuring ice sheet data [30]. Our primary goal is to provide a proof-of-concept for the embedded model approach by comparing simulation results, specifically ice velocity, with directly observed results as well as unaltered PISM results. Finally we will establish baseline performance metrics to compare performance between our embedded model and non-embedded models.

CHAPTER 6

RESULTS

In this chapter we present the results of our embedded model and compare them to results from non-embedded models. We begin by comparing the fidelity of our embedded model to an equivalent non-embedded model. A cross-section of the embedded model is taken from these results to demonstrate the accuracy of our embedded model by comparing values to directly observed data. Finally the baseline computational costs are shown for our embedded model and non-embedded model.

We first compare the whole Greenland ice sheet results between our embedded model and an unmodified version of PISM. Each model used a two-kilometer-resolution data set and was simulated one year into the future. The focus of these results is the overall level of detail in the ice velocity results of the embedded Jakobshavn outlet glacier region, where velocities as high as seven kilometers per year have been observed. These results are presented in the form of normalized heatmaps, where the Jakobshavn region is emphasized (Figure 6.1).

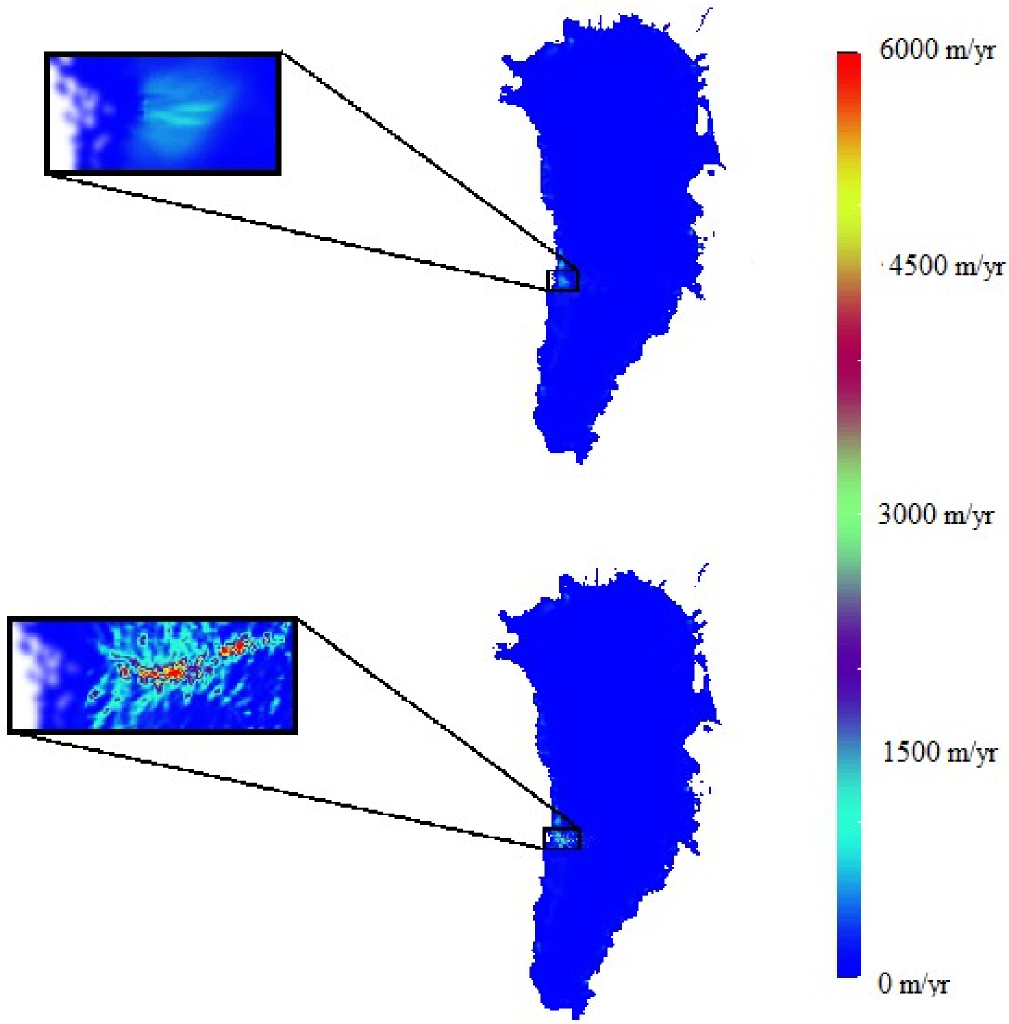

Figure 6.1: Heatmap Results of the Embedded Model. Embedded model (bottom) is compared to non-embedded (top) with color scale.

The topmost map shows the Greenland ice sheet as modeled by an unmodified version of PISM. The bottom map also shows the Greenland ice sheet but is modeled by our embedded model. Both heatmaps are accompanied by a common scale, which is critical to demonstrating their differences visually.

The ice velocities recorded by the embedded model are significantly higher than those of the unmodified model, in some areas reaching seven kilometers per year. These velocities are at their highest within the ice stream, causing the embedded model heatmap to correspond to the bedrock elevation of the ice stream. This is in stark contrast to the unmodified heatmap in which the velocities recorded do not exceed one kilometer a year and fail to match the shape of the ice stream resulting in a blurry image. Therefore these results clearly show an increase in simulation fidelity is present in our embedded model compared to the unmodified model.

While these results show an increase in fidelity over unmodified PISM models it is critical that we compare our simulated results with directly observed values provided from the National Snow and Ice Data Center [33, 34]. To do so we have extracted ten-kilometer cross-sections of ice velocity measurements in the Jakobshavn outlet glacier. These cross sections are compared to directly observed values taken of the Jakobshavn outlet glacier. All 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. These results are presented in Figure 6.2.

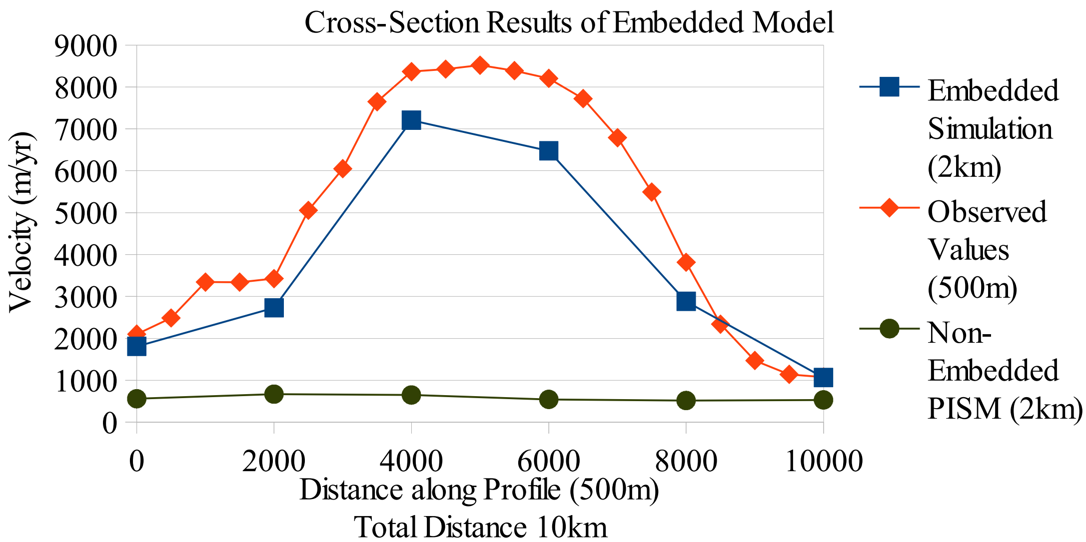

Figure 6.2: Cross Section Results of Embedded Model. Directly observed ice velocities, simulated ice velocities from the embedded model, and simulated ice velocities from the non-embedded model. There are more data points for the observed values because they are taken for every 500m, while our models operated at a resolution of 2km.

It is clear from the comparison (Figure 6.2) that the embedded model computed higher velocities than a non-embedded model. Most intriguing is that the embedded model results begin to capture the same curve as the observed results, demonstrating a feedback loop operating between the high-resolution and low-resolution models within the embedded region. Such a phenomenon does not exist for the non-embedded model, where the same velocity curve is far less pronounced. T-test values of both the embedded values (0.64) and non-embedded values (0.02) that the embedded model more closely captures the feedback loop between ice streams and the whole ice sheet.

Having established a clear improvement in model fidelity with our embedded model we now want to provide a baseline measure of computational costs. Therefore, we measured the total time required for our embedded model and a non-embedded two-kilometer resolution model to simulate one logical year. For our embedded model we want to present the amount of time spent waiting as well as the total simulation time. These results are presented in Table 6.1.

Table 6.1 Computational Results			
	500m Non-embedded	2km Non-embedded	2km with 500m Embedded
Number of Processors	4096	700	350 each
Run Time	~27000.00s	300.44s	6373.20s
Wait Time	0s	0s	1037.90s
Number of Time Steps	24930	277	8694

Table 6.1: Computational Results. Basic performance characteristics of our prototype embedded simulation compared to a non-embedded simulation.

Our embedded model took longer to simulate out to one year than a non-embedded model (Table 6.1). There are many reasons for this, the most significant of which being the total number of time steps. Our embedded model experienced a 31-fold increase in number of timesteps compared to a non-embedded model, resulting in slower overall execution of the model. This is because high resolution model data cannot be simulated as far without the possibility of causality errors, resulting in shorter overall timesteps. While an increase in the number of timesteps is to be expected when synchronizing models of different spatial and temporal resolutions, the magnitude of the increase was significantly higher than expected.

Other contributing factors of the longer simulation time are as a result of our prototype implementation. Our current synchronization protocol, due to the required blocking of each simulation, introduces additional wait times in the embedded simulation that are not seen in a non-embedded simulation. We also do not have a processor sharing scheme established, instead opting to evenly divide processors between the embedded model and low-resolution whole ice sheet model. Finally the higher-resolution ice thickness provided to the low-resolution whole ice sheet model result in more rigorous computations required for our embedded simulation compared to a non-embedded simulation using only low-resolution data.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

In this thesis we have presented our experimental design and implementation of a high-resolution ice sheet model of a region undergoing rapid change embedded within a low-resolution ice sheet model of regions evolving at a slower pace. We have described our experimental synchronization protocol used to allow communication between models of different spatial and temporal resolutions while maintaining fidelity of the simulation. We have also shown our experimental modifications to PISM to allow for such a synchronization protocol using PETSc to preserve internal simulation mechanisms.

Our embedded model prototype results show a clear feedback loop is captured in the shared spatial domain of the grids. This feedback loop is clear when the embedded model results are compared to observed results, where the shape of the velocity curve in the embedded model corresponds to the directly observed values. Given this strong correlation between the embedded model and the directly observed values we believe our prototype model was successful.

While our prototype serves as a proof of concept, there are still performance issues that remain to be solved. The most immediate improvement is introducing additional load balancing to the embedded model since currently the computational load is based on evenly dividing the processors between the models. Such a simple approach can result in cases where processors responsible for one model are forced to wait longer for the other model. The total waiting time in our prototype can be reduced by balancing the computational load between each model. We are also interested in incorporating

SuperMic's Intel Xeon Phi 7120P coprocessors into our prototype embedded model. This would grant additional computational resources that are otherwise left unused as well as increasing the number of processors for use in load balancing.

Finally we are exploring the possibility of incorporating optimistic simulation techniques to further reduce the waiting time. Such techniques will allow each model in the embedded simulation to proceed without waiting, allowing a model to rollback its state whenever error conditions are detected. Further research is needed to determine how to best use these optimistic simulation techniques and what would constitute an error condition.

REFERENCES

[1] Rignot, Eric, and Pannir Kanagaratnam. "Changes in the velocity structure of the Greenland Ice Sheet." *Science* 311, no. 5763 (2006): 986-990.

[2] Truffer, Martin, and Keith A. Echelmeyer. "Of isbrae and ice streams." *Annals of Glaciology* 36, no. 1 (2003): 66-72.

[3] “Stampede”, Texas Advanced Computing Center, accessed April 23, 2016, https://www.tacc.utexas.edu/stampede/

[4] Seguinot, J., I. Rogozhina, A. P. Stroeven, M. Margold, and J. Kleman. "Numerical simulations of the Cordilleran ice sheet through the last glacial cycle." *Cryosphere Discussions* 9, no. 4 (2015).

[5] Yan, Qing, Zhongshi Zhang, and Huijun Wang. "Investigating uncertainty in the simulation of the Antarctic ice sheet during the mid‐Piacenzian." *Journal of Geophysical Research: Atmospheres* (2016).

[6] Aschwanden, Andy, Mark A. Fahnestock, and Martin Truffer. "Complex Greenland outlet glacier flow captured." *Nature communications* 7 (2016).

[7] Pattyn, Frank, Laura Perichon, A. Aschwanden, Beate Breuer, Brigitte De Smedt, Olivier Gagliardini, G. Hilmar Gudmundsson et al. "Benchmark experiments for higher-order and full Stokes ice sheet models (ISMIP-HOM)." *The Cryosphere Discussions* 2, no. 1 (2008): 111-151.

[8] Cornford, Stephen L., Daniel F. Martin, Daniel T. Graves, Douglas F. Ranken, Anne M. Le Brocq, Rupert M. Gladstone, Antony J. Payne, Esmond G. Ng, and William H. Lipscomb. "Adaptive mesh, finite volume modeling of marine ice sheets." *Journal of Computational Physics* 232, no. 1 (2013): 529-549.

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

[10] Bueler, Ed, and Jed 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).

[11] Docquier, David, Laura Perichon, and Frank Pattyn. "Representing grounding line dynamics in numerical ice sheet models: recent advances and outlook." *Surveys in geophysics* 32, no. 4-5 (2011): 417-435.

[12] Rutt, Ian C., M. Hagdorn, N. R. J. Hulton, and A. J. Payne. "The Glimmer community ice sheet model." *Journal of Geophysical Research: Earth Surface* 114, no. F2 (2009).

[13] Lipscomb, William H., Jeremy G. Fyke, Miren Vizcaíno, William J. Sacks, Jon Wolfe, Mariana Vertenstein, Anthony Craig, Erik Kluzek, and David M. Lawrence. "Implementation and initial evaluation of the glimmer community ice sheet model in the community earth system model." *Journal of Climate* 26, no. 19 (2013): 7352-7371.

[14] 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).

[15] Winkelmann, R., M. A. Martin, M. Haseloff, T. Albrecht, E. Bueler, C. Khroulev, and A. Levermann. "The Potsdam parallel ice sheet model (PISM-PIK)–Part 1: Model description." *The Cryosphere* 5, no. 3 (2011): 715-726.

[16] Martin, M. A., R. Winkelmann, M. Haseloff, T. Albrecht, E. Bueler, C. Khroulev, and A. Levermann. "The Potsdam Parallel Ice Sheet Model (PISM-PIK)–Part 2: Dynamic equilibrium simulation of the Antarctic ice sheet." *The Cryosphere* 5, no. 3 (2011): 727-740.

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

[18] Kwok, Ronald, and Mark A. Fahnestock. "Ice sheet motion and topography from radar interferometry." *Geoscience and Remote Sensing, IEEE Transactions on* 34, no. 1 (1996): 189-200.

[19] Tedesco, M., J. E. Box, J. Cappelen, X. Fettweis, T. Mote, R. S. W. van de Wal, C. J. P. P. Smeets, and J. Wahr. "Greenland ice sheet." *www. arctic. noaa. gov/reportcard* (2014): 22.

[20] Velicogna, I., and J. Wahr. "Time‐variable gravity observations of ice sheet mass balance: Precision and limitations of the GRACE satellite data." *Geophysical Research Letters* 40, no. 12 (2013): 3055-3063.

[21] Joughin, Ian, Waleed Abdalati, and Mark Fahnestock. "Large fluctuations in speed on Greenland's Jakobshavn Isbrae glacier." *Nature* 432, no. 7017 (2004): 608-610.

[22] "PyProj 4", accessed April 17, 2016, https://pypi.python.org/pypi/pyproj

[23] 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).

[24] 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.

[25] 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.

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

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

[28] "Torque Resource Manager", accessed April 23, 2016, http://www.adaptivecomputing.com/products/open-source/torque/

[29] "SeaRISE Assessment", Sea Level Response to Ice Sheet Evolution (SeaRISE), accessed February 17, 2016, http://websrv.cs.umt.edu/isis/index.php/SeaRISE_Assessment

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

[31] "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/

[32] 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.

[33] Joughin, I., B. Smith, I. Howat, T. Scambos, and T. Moon. Greenland Flow Variability from Ice-Sheet-Wide Velocity Mapping, *Journal of Glaciology.* 56 (2007). 415-430.

[34] Joughin, I., B. Smith, I. Howat, and T. Scambos. 2015. *MEaSUREs Greenland Ice Sheet Velocity Map from InSAR Data, Version 2*. Boulder, Colorado USA. NASA National Snow and Ice Data Center Distributed Active Archive Center. http://dx.doi.org/10.5067/OC7B04ZM9G6Q. Accessed February 17, 2016.

[35] "MAUP", GIS Population Science, accessed August 8, 2016, http://gispopsci.org/maup/

[36] Dark, Shawna J., and Danielle Bram. "The modifiable areal unit problem (MAUP) in physical geography." *Progress in Physical Geography* 31, no. 5 (2007): 471-479.

[37] Dickens, Phillip M., C. Dufour, and J. Fastook. 2016. "A Prototype Implementation of an Embedded Simulation System for the Study of Large Scale Ice Sheets". *Procedings of the 2016 Winter Simulation Conference*, (2016).

BIOGRAPHY OF THE AUTHOR

Christopher Dufour was born in Fort Kent, Maine on November 21, 1988. He was raised by Glenn and Nancy Dufour in Wilton, Maine, and graduated from Mount Blue High School in 2007. Christopher graduated from the University of Maine at Orono in Fall 2012 with a Bachelors of Science in Biology and minor in Computer Science. He has since enrolled in Graduate School at the University of Maine at Orono while working as both a Teaching Assistant and Course Instructor. He is a candidate for the Master of Science degree in Computer Science from the University of Maine in August 2016.