Imperial College London

Doctoral Thesis

Simulating Ice Processes using the Finite Element, Unstructured, Adaptive Model Fluidity

Author:
Simon Mouradian

Supervisor:
Dr. Matthew Piggott

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

in the

Applied Modelling and Computation Group
Department of Earth Science and Engineering

February 2015

Declaration of Authorship

I, Simon MOURADIAN, declare that this thesis titled, 'Simulating Ice Processes using the Finite Element, Unstructured, Adaptive Model Fluidity' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"When you set sail for Ithaca,
wish for the road to be long,
full of adventures, full of knowledge."

C.P. Cavafy

Abstract

The cryosphere impacts global climate in various ways. Snow and ice have a higher albedo than land or the open ocean and therefore affect the total reflectivity of the earth. Sea ice forms an insulating layer over the polar oceans controlling both heat and water vapour fluxes between the atmosphere and polar ocean. Ice sheets hold around 77% of Earth's freshwater reserves, and recent increases in ice loss from the Earth's ice sheets are cause for concern. This thesis develops numerical tools that can be used for the study of various ice processes such as ocean – sea ice interaction, ice sheet and glacier dynamics.

A coupled ocean – sea ice model is developed, using the open source, unstructured, adaptive ocean model Fluidity and a finite element sea ice model developed at the Alfred Wegener Institute of Polar and Marine Research, FESIM. The tightly coupled model is verified and validated through a series of tests, demonstrating its dynamical capabilities. The sea ice dynamics are a model of Elastic–Viscous–Plastic rheology, as described in Hunke and Dukowicz [1997]. The thermodynamic parameterisation is similar to the 1D simplest model of Parkinson and Washington [1979] which is based on the zero–layer approach of Semtner [1976].

Furthermore, a new computational framework for carrying out ice sheet simulations is presented. A thermo–mechanical, non–linear, full–Stokes model is used to carry out the exercises of the Ice Sheet Model Intercomparison Project for higher–order models (ISMIP–HOM). The results presented here show that Fluidity compares favourably with other ice sheet models. Further tests are performed to demonstrate the use of dynamic adaptive remeshing in lowering the computational cost of models compared to their structured, fixed–mesh counterparts, by focusing resolution only where and when required. Finally, initial simulations of the full Greenland ice sheet are performed demonstrating the potential utility of adaptive meshes for large–scale, full–Stokes modelling.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Dedicated to my grandparents Anni, Bedros, Costas and Yiannoula.

CHAPTER 1

INTRODUCTION

1.1 The Cryosphere

The cryosphere is an important and interesting part of the Earth.

Definition 1.1 (Cryosphere)**.** from the Greek cryos meaning *cold* and sphaira, meaning *globe*, describes the parts of the Earth where water is in solid form, frozen into ice or snow.

The term encompasses sea ice, lake ice, snow cover, glaciers, frozen ground, ice caps and ice sheets. Its effect on climate and weather, its inhabitants and the ever increasing worry of the effects of a changing climate make it an important component of the Earth to study.

In this thesis, we discuss the modelling of sea ice, glaciers and ice sheets.

1.1.1 Where is the Cryosphere?

The cryosphere can largely be found in the Arctic, Antarctic and high elevation regions.

1.1.1.1 The Arctic

A large part of the cryosphere can be found in the Arctic. The Arctic Ocean is cold enough such that the surface freezes into solid sea ice. Sea ice grows in the winter and shrinks during the summer months. The lands surrounding the Arctic Ocean, such as Alaska, the Canadian Arctic Archipelago, northern Siberia and Greenland, are also part of the cryosphere as the soil is cryotic (has temperatures below 0°, such soil is called *permafrost*) and is usually covered in ice or snow. One such land is Greenland, home to one of two ice sheets on Earth. The Greenland ice sheet is simulated in this thesis (see chapter 4).

1.1.1.2 The Antarctic

The Antarctic is a land mass covered by an ice sheet, surrounded by an ocean that is covered by sea ice. Shelves of floating ice (ice shelves) extend from the ice sheet into the ocean. Some portions of these break off (a process known as *calving*), forming icebergs which float in the ocean and melt as they drift into warmer waters. Although the Antarctic ice sheet is not simulated in this thesis, techniques developed here could be applied to the modelling of the Antarctic ice sheet, or the sea ice covered Southern Ocean.

1.1.1.3 High elevations

Glaciers and snow covered regions atop of mountain ranges across the world, also form part of the cryosphere. One such place, modelled in this report, is the Arolla Glacier in the Pennine Alps of Valais in Switzerland.

1.2 Modelling and the Finite Element Method

1.2.1 Importance of Modelling

Numerical models are of paramount importance in the study of the cryosphere. The various spatial and temporal scales of geophysical phenomena means that reproducing the dynamics in a laboratory setting is very challenging. Observations are limited and often hard to obtain. Measured data is sometimes only available at scattered points. Numerical modelling may be able to fill in missing data, aid observation and interpretation. In addition, it may be used to predict paleo or future states of the cryosphere.

Furthermore, numerical modelling may be able to isolate specific dynamics to provide insight and understanding of particular processes separate from the biases of the complete dynamical, geophysical system. For example, a model with only the ocean and sea ice may be able to provide insight into the dynamical coupling between the two components without the feedbacks of a changing atmosphere. Although such simplified models are not necessarily realistic, they are, potentially, able to aid the scientific understanding of specific, dynamical processes.

Simplifying and modifying particular parameters of an experiment is easier in a numerical study than the comparable experimental study. For example, deep water gravity currents are thought to be highly affected by the slope of the bathymetry of the domain in which they propagate. Simplifying, modifying and comparing different bathymetries to analyse the sensitivity of currents to such parameters is potentially much easier in a numerical experiment. In addition to changes to the bathymetry, sensitivities to other parameters may be investigated with numerical studies, such as the bottom friction coefficient.

Scientists make use of various tools to study a large range of scales of sea ice processes. These range from brine pocket formation at a microscopic scale, to the evolution of sea ice extent, concentration and thickness monitored via

satellite imaging at a macroscopic scale. The use of a relatively modern tool, that of numerical simulation, has a very large and important role to play in the analysis and prediction of this key component of the Earth's system. Numerical simulations are not bidding to replace traditional means of scientific study, but are undertaken in addition to theoretical, experimental and observational studies. The aim is for numerical studies to complement, rather than replace, such scientific endeavours.

1.2.2 The Finite Element Method and Unstructured meshes

The finite element method is a discretisation technique used to transform continuous differential equations with boundary constraints into a linear system of equations (see example in section 1.2.2.1). The linear system can then be solved either directly or iteratively (most common for large systems). The solver strategies employed for solving the linear systems obtained when solving the problems described in this thesis are outlined in each chapter, but not discussed in detail. Fluidity makes use of the PETSc library for solving linear systems of equations, and does not employ solvers of its own.

When modelling a system numerically, a domain decomposition is required that separates the domain into non-overlapping subdomains, called elements (or cells). This decomposition is called a mesh and may be structured or unstructured.

A structured mesh is one that has regular connectivity, and the neighbourhood relationship is known implicitly. This is in contrast to an unstructured mesh, where the element connectivity needs to be explicitly provided. An unstructured mesh is usually stored in a vector, as a two– or three–dimensional array is not generally suitable.

Unstructured meshes offer significant advantages over their structured counterpart in regards to representing irregular features such as coastlines and

bathymetry, as well as any sharp or anisotropic features of the solution fields. For example, an unstructured mesh may be able to better represent the edge of a sea ice covered region by smoothly increasing the resolution in the regions where the sea ice boundaries occur. Structured meshes can also have varying resolutions by nesting grids, but special treatment of such regions is usually required to maintain the correctness of the solution fields. In return, unstructured meshes come with a significant computational overhead, due to the indirect addressing of memory. It has been shown that varying the mesh resolution through the domain, either in a fixed or adaptive simulation, as well as running in parallel, can compensate for the computational overhead associated with such meshes.

1.2.2.1 Example: The Steady Advection Diffusion equation

Let us consider the transport (advection) and diffusion of a scalar property $c = c(\mathbf{x})$, in a domain Ω with smooth boundary Γ. The boundary Γ is composed of a part Γ_D, where the value of $c(\mathbf{x})$ is prescribed as in (1.2), and a complementary part Γ_N, where the flux of $c(\mathbf{x})$, $\frac{dc}{dn}$, is prescribed, as in (1.3). The conditions specified on Γ_D and Γ_N are called Dirichlet and Neumann boundary conditions, respectively.

The equations associated with the above advection diffusion problem and boundary constraints are the following:

$$\frac{\partial c}{\partial t} = \mathbf{u} \cdot \nabla c - \nabla \cdot (\nu \nabla c) + s' \qquad \text{in } \Omega, \tag{1.1}$$

$$c = c_D \qquad \text{on } \Gamma_D, \tag{1.2}$$

$$\nu \frac{\partial c}{\partial \mathbf{n}} = h \qquad \text{on } \Gamma_N, \tag{1.3}$$

where $\mathbf{u}(\mathbf{x})$ is the prescribed advection velocity, $\nu > 0$ is the coefficient of diffusivity and $s'(\mathbf{x}) = -s(\mathbf{x})$ is a sink term.

For simplicity, with no loss in appropriateness of the example, we assume steady state ($\frac{\partial c}{\partial t} = 0$) and unit diffusivity ($\nu = 1$). (1.1) hence reduces to:

$$\underbrace{\mathbf{u} \cdot \nabla c}_{\text{advection term}} - \underbrace{\nabla^2 c}_{\text{diffusion term}} = \underbrace{s}_{\text{source term}} . \tag{1.4}$$

The finite element discretisation begins by identifying the associated weak form (or variational form) of (1.4). This is obtained by multiplying by a *test* function, w, and integrating over the domain of interest:

$$\int_\Omega w \left(\mathbf{u} \cdot \nabla c\right) \, d\Omega - \int_\Omega w \nabla^2 c \, d\Omega = \int_\Omega w \, s \, d\Omega. \tag{1.5}$$

Using the divergence theorem on the diffusion term and specifying $w = 0$ on the Dirichlet portion of the boundary allows us to write the above in the following form:

$$\int_\Omega w \left(\mathbf{u} \cdot \nabla c\right) \, d\Omega + \int_\Omega \nabla w \cdot \nabla c \, d\Omega = \int_\Omega w \, s \, d\Omega + \int_{\Gamma_N} w \frac{\partial c}{\partial \mathbf{n}} \, d\Gamma_N. \tag{1.6}$$

Note that having used the divergence theorem allows us to apply the Neumann boundary condition (1.3) in a very natural way:

$$\int_{\Gamma_N} w \frac{\partial c}{\partial \mathbf{n}} \, d\Gamma_N = \int_{\Gamma_N} w \, h \, d\Gamma_N. \tag{1.7}$$

The continuous weak form (1.6) forms the basis of the finite element discretisation for this example problem.

Before proceeding with the Galerkin method of discretising the above system, we must discretise the domain Ω into elements Ω^e, $1 \leq e \leq n_{el}$, where n_{el} is the number of elements (non-overlapping sub-domains) in Ω. This domain discretisation is called a mesh. The mesh is then equipped with a *finite element interpolation space*, which associates *nodes* and *shape functions* to each element of the mesh. Common spaces include the $\mathcal{P}_m$ space, which associates linear polynomials of order $\leq m$ to the reference triangle (in two dimensions; tetrahedron in three dimensions), and space $\mathcal{Q}_m$ which associates bilinear polynomials of order $\leq m$ to the reference square (in two dimensions; cube in three dimensions).

For example, consider the $\mathcal{P}_1$ finite element interpolation space on a one dimensional element, illustrated in 1.1. This places two nodes, n_1 and n_2, on each corner of the element and each node is associated with one *basis function*, $N_i(\mathbf{x})$, and one *nodal unknown*, ϕ_i.

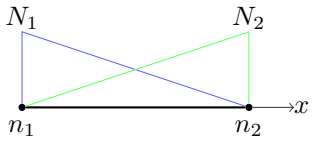


FIGURE 1.1: The $\mathcal{P}_1$ interpolation space on a one dimensional element.

The basis functions are chosen such that they have a value of unity at the nodes for which they are defined, and a value of zero at all other nodes. An unknown, $\phi(x)$, thus takes the value of ϕ_i at the coordinates of the i^{th} node, and can be reconstructed, given the nodal unknowns, for any location in the domain Ω by:

$$\phi^h(\mathbf{x}) = \sum_{A \in n_{el}} N_A(\mathbf{x})\phi_A \tag{1.8}$$

We use the notation superscript h, to indicate that this is a finite element approximation to the continuous unknown, ϕ.

Now that we have derived the continuous weak form (1.11) and have discussed the process of Galerkin approximation to functions, we have all the necessary tools to proceed and discretise the advection diffusion system introduced above. We begin by approximating the unknown function $c(\mathbf{x})$:

$$c^h(\mathbf{x}) = \sum_{A \in \eta \setminus \eta_D} N_A(\mathbf{x}) c_A + \sum_{A \in \eta_D} N_A(\mathbf{x}) c_D(x_A), \tag{1.9}$$

where η is the set of all nodes and η_D is the set of nodes on the Dirichlet portion of the boundary, Γ_D. We not that c_D represents the prescribed Dirichlet boundary condition. The *test* function w can be approximated in a similar fashion:

$$w^h(\mathbf{x}) = \sum_{B \in \eta \setminus \eta_D} N_B(\mathbf{x}) w_B, \tag{1.10}$$

noting that $w_B = 0$ on the Dirichlet boundary. Replacing the unknowns $c(\mathbf{x})$ and $w(\mathbf{x})$ in (1.11) with their corresponding Galerkin approximations we have:

$$\int_\Omega w^h \left(\mathbf{u} \cdot \nabla c^h \right) \, d\Omega + \int_\Omega \nabla w^h \cdot \nabla c^h \, d\Omega = \int_\Omega w^h \, s \, d\Omega + \int_{\Gamma_N} w^h \, h \, d\Gamma_N \tag{1.11}$$

and after substituting (1.9) and (1.10):

$$\sum_i \left\{ \int_\Omega N_A \left(\mathbf{u} \cdot \nabla N_B \right) c_i + \int_\Omega \nabla N_A \cdot \nabla N_B c_i \right\} =$$
$$\sum_i \left\{ \int_\Omega N_A \; s_i + \int_{\Gamma_N} N_A h_i \right.$$
$$\left. - \int_{\Gamma_D} N_A (u \cdot \nabla N_B) c_D - \int_{\Gamma_D} \nabla N_A \cdot \nabla N_B \; c_D \right\}. \qquad (1.12)$$

In matrix form, we can write (1.12) as:

$$\left(\mathbf{A}(\mathbf{u}) + \mathbf{K} \right) \mathbf{c} = \mathbf{f}, \qquad (1.13)$$

where $\mathbf{A}(\mathbf{u})$ is the advection matrix, $\mathbf{K}$ the diffusion matrix, $\mathbf{c}$ the vector of nodal unknowns (c_i), and $\mathbf{f}$ the *r.h.s vector* containing source and boundary forcing terms. ($\mathbf{A}(\mathbf{u})$+$\mathbf{K}$) is sometimes collectively named the stiffness matrix.

This concludes the introduction to the finite element discretisation, which is the method with which all equations studied in this thesis are discretised. In the rest of this thesis, equation discretisation will be described assuming the level of knowledge described in this Introductory chapter.

1.2.3 Dynamic Adaptivity of Unstructured meshes

Dynamic adaptivity is the process of adapting the mesh during the simulation, to obtain a new mesh that is numerically more optimal, by placing resolution where the simulated field of interest requires it. In Fluidity, the `libmda2d` library is used for 2D mesh adaptivity [Y.Vasilevskii and K.Lipnikov, 1999], and the `libadaptivity` library is used for 3D mesh adaptivity [Pain et al., 2001].

The adaptivity process takes place by first calculating element quality. The mesh is then refined with the use of various topological processes, before the solution fields are interpolated from the original mesh to the new, refined mesh. Each of these processes are briefly outlined below:

Measure element quality. In 3D, the quality of each element in a mesh is obtained using the quality functional Q_e:

$$Q_e|_{\mathbf{M}} = \left(\frac{1}{2\sqrt{6} r_e|_{\mathbf{M}}} - 1 \right)^2 + \frac{1}{2} \sum_k \left(l_{e,k}|_{\mathbf{M}} - 1 \right), \tag{1.14}$$

where the notation $|_{\mathbf{M}}$ indicates that the parameter is measured with respect to a metric tensor M. r_e is the radius of the largest sphere that can occupy a tetrahedral element e. k indexes each edge of the element and $l_{e,k}$ represents the length of each edge k. This quality functional is minimal ($Q_e = 0$) when the tetrahedron has edge lengths of unity in the metric space. The metric, M, is computed using the Hessian matrix of a field, $|H|$ and a user defined weighting, ϵ_u:

$$M = \frac{1}{\epsilon_u} |H| . \tag{1.15}$$

This is the form of the metric M, used for the adaptive simulations in this thesis, and is named the absolute metric. In Fluidity, an alternative form, named the p–metric, is also available:

$$M = \frac{1}{\epsilon_u} \det |H|^{-\frac{1}{2p+n}} |H| . \tag{1.16}$$

Both metrics have been shown to be robust. The absolute metric tends to focus resolution on the dynamics with the largest magnitude, whilst the p–metric does a better job of resolving dynamics of lesser magnitude [AMCG, 2014].

Refine mesh. Once the Hessian, metric and finally the quality functional are computed, the mesh refinement operation can begin. The process used in Fluidity involves looping over all elements (starting from the element with the worst quality) and performing the following topological operations in the locality of the element:

- Edge splitting. The edge of a tetrahedron is split and new elements that share this vertex are created. This operation increases mesh resolution.
- Edge collapse. An edge collapses to a vertex at its midpoint, and the associated zero–volume tetrahedrons are removed. This operation decreases mesh resolution.
- Face–edge and edge–face swapping. This operation manipulates edge connections between adjacent elements, potentially increasing or decreasing the resolution.
- Node movement. A vertex can be moved within the space spanned by the elements which contain it. This operation can increase or decrease the resolution.

The operations which cause an improvement to element quality are accepted and a new mesh is obtained.

Transfer field data to new mesh. Once a new mesh is obtained, transferring data from the old mesh to the new mesh is required. This process is performed through interpolation. Fluidity includes various interpolation methods, the two most commonly used are *consistent interpolation* and *Galerkin projection*. Consistent interpolation involves computing the value of a field, c, at the location of the nodes of the new mesh, using the basis functions, N_i, of the old mesh:

$$c_{new}(\mathbf{x}) = \sum_i c_i N_i(\mathbf{x}), \tag{1.17}$$

where i indexes all nodes of the element containing the new node, c_i is the known nodal values on the old mesh and N_i the basis functions on the old mesh. Consistent interpolation is bounded [Farrell et al., 2009] and computationally inexpensive [Hiester, 2011], but is not conservative and can be dissipative. It is, unless otherwise stated, the interpolation used for the problems in this thesis. Galerkin projection [Farrell, 2009], on the

other hand, is a conservative interpolation, but is non-bounded. Fluidity also offers a bounded–Galerkin projection algorithm, which diffuses the projection until boundedness is achieved [Farrell et al., 2009].

1.3 Statement of contributions

Great care has been taken in the writing of this thesis to ensure that work done by others has been marked as such and acknowledgements were rightly attributed. This section lists my main achievements, which are described in detail in the remaining chapters of this thesis.

A feasibility study of providing Fluidity (a model developed by the Applied Modelling and Computation Group and its partners) with a coupling interface was undertaken. The use of the OASIS coupler was first investigated, by performing three tasks:

- Coupling sample models: short programs (30 lines) that investigate the use of the OASIS API.
- Coupling festa and OASIS: Model initialisation and appropriate send/receive calls.
- Coupling Fluidity and OASIS:
  - On–the–fly exchange mesh generation.
  - Consistent interpolation between unstructured Fluidity mesh and structured exchange mesh before and after data exchange.

The next contribution was the coupling of Fluidity to the FESIM sea ice model. This involved:

- Converting stand–alone FESIM to a module.

- Extracting a two dimensional surface mesh from Fluidity's three dimensional mesh on the sphere. The surface mesh is then transformed from the Cartesian to the spherical coordinate system.
- Implement atmospheric forcing reader and NCAR fluxes calculator.
- Interpolate data from the ocean mesh to the sea ice mesh and the other way around.

The coupled model was then subjected to a series of tests. A sea ice test was devised using the method of manufactured solutions, and further tests found in the literature were used. Where available, the results were compared to those obtained by other models or with analytical solutions.

Finally, simulations for the modelling of ice sheets and glaciers were set up. This involved prescribing the interaction between the viscosity and velocity (and temperature, where required) terms, done via the interface provided in diamond. No further development work was required or performed. Fluidity's ability to solve the full–Stokes equations was demonstrated in a test devised using the method of manufactured solutions. The usefulness of unstructured, dynamically adaptive meshes for modelling glaciers was shown in a test of an idealised ice stream. Tests of the ice sheet modelling intercomparison study were also performed and results were compared and contrasted with those of other models. Finally, a preliminary simulation of Greenland was carried out, indicating that Fluidity is able to solve problems on realistic domains of continental scale.

CHAPTER 2

SEA ICE MODELLING

2.1 Introduction

2.1.1 Motivation

Sea ice is frozen seawater, which grows at the surface of the oceans as temperatures drop below approximately $-1.8°C$, the salinity–dependent freezing temperature typical in the polar oceans. As the surface waters of the ocean cool, small ice crystals (called *frazil ice*) form that float to the surface. As these crystals accumulate, sea ice is formed. Figure 2.1 shows an aerial view of sea ice.

Sea ice covers up to 7% of the Earth's oceans. It acts as an insulating layer, notably altering the transfer of heat, mass and momentum between the ocean and the atmosphere. Furthermore, sea ice affects the reflectivity of the region to solar radiation. The proportion of radiation reflected by a surface is termed albedo. Sea ice has a higher albedo than the open ocean.

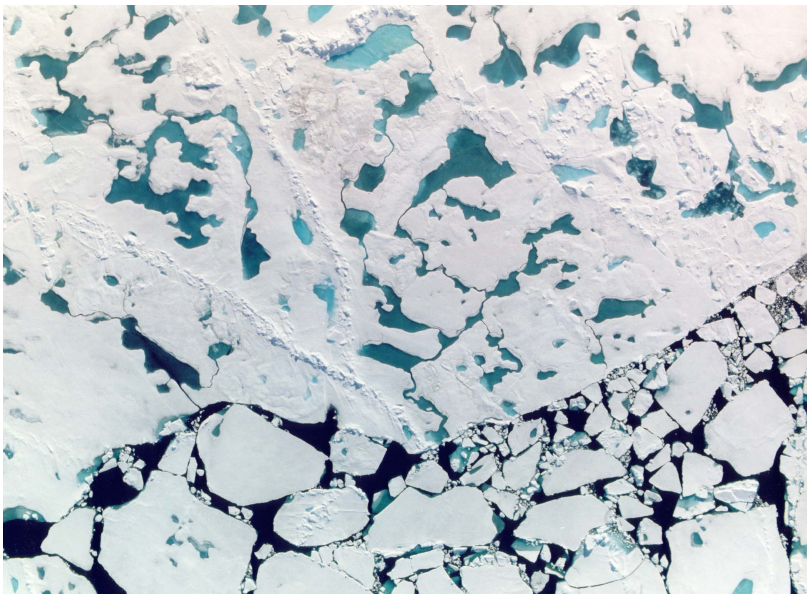


FIGURE 2.1: Aerial view of summer sea ice, with ponds of meltwater (turquoise) covering the Arctic ocean (dark blue). (Credit: Don Perovich)

Surface	**Albedo**
Open ocean	0.06
Bare sea ice	0.5–0.7
Snow covered sea ice	0.9

TABLE 2.1: Typical albedo values for various surfaces.

Typical albedo values are given in Table 2.1. As indicated in the table, sea ice can reflect up to 90% of incident solar radiation, preventing the cool surface waters of the ocean from warming. However, once increased temperatures lead to a decrease in sea ice cover (and hence a decrease in albedo), more solar radiation can be absorbed by the oceans, an effect that amplifies warming and further reduces sea ice cover. This is known as the ice–albedo effect.

Sea ice also acts as a thin blanket over the surface of the winter ocean. During the winter months, temperatures in the overlying atmosphere fall below those of the warmer ocean. The presence of this insulating layer prevents the large heat fluxes one would expect from the ocean to the atmosphere, thus ensuring the atmosphere is not warmed by the ocean.

Global ocean circulation is also affected by the behaviour of sea ice. The global "conveyor belt" (depicted in figure 2.2) is driven by the difference in

density between the various water masses.

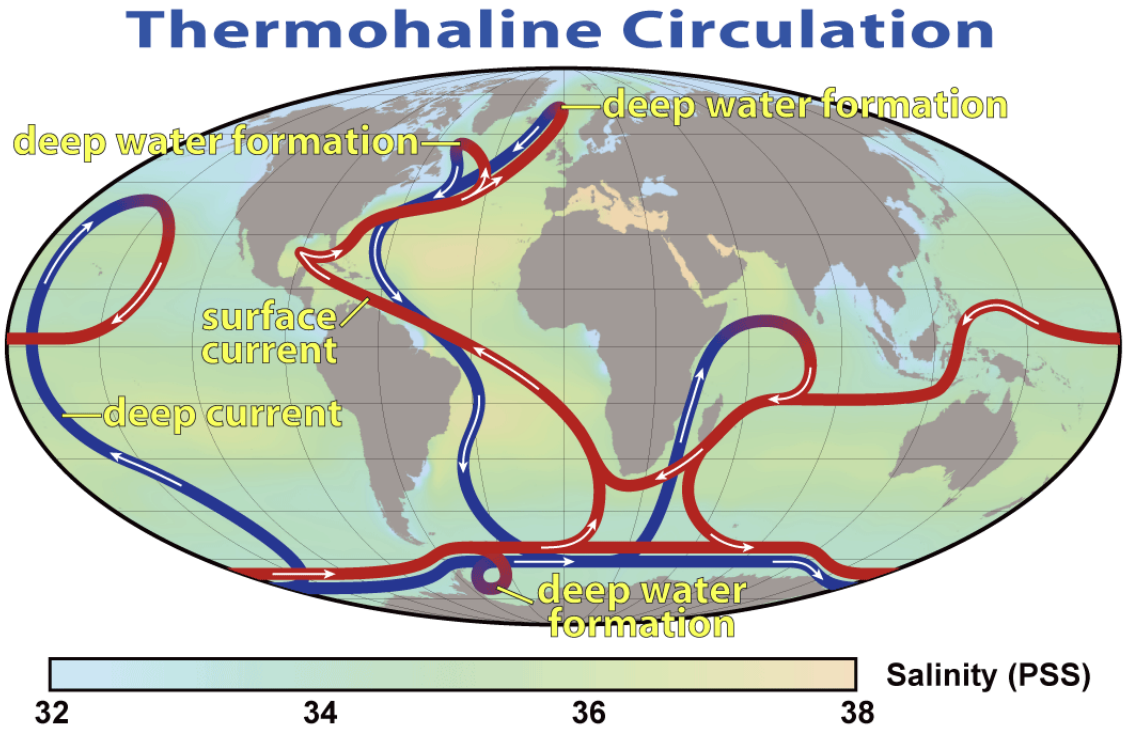


FIGURE 2.2: A highly idealised schematic of the global thermohaline circulation. Blue paths represent cooler, deep water currents, red paths represent warmer surface currents.

The thermohaline circulation is driven by the temperature (from the Greek heat, *thermos*) and salinity (Greek salt, *hals*) of the water. Sea ice formation and melt both contribute to the salinity of the underlying ocean. During the formation of frazil ice, the salinity of the surface layers of the ocean is increased. Despite the ice crystals not containing any salt, salty water (*brine*) is trapped inside the lattice of the sea ice, which is then discharged (a process known as *desalination*) through various processes. The primary processes of desalination are gravity draining (surface cooling causing convective overturning which draws ocean water from the underlying ocean to replace the brine inside the sea ice) and flushing (surface meltwater flushing through the ice lattice, expelling the brine).

The surface waters, now cooler and denser than their underlying water, sink to form bottom waters. These are illustrated in figure 2.2 by paths changing from red to blue, and marked with the label *"deep water formation"*. It is

believed that changes in sea ice cover and the amount of sea ice that forms each winter will have significant effects on the global thermohaline circulation. Such changes have the potential to impact the weather and climate in regions of the Earth which currently enjoy more temperate climates.

In summary, sea ice is an important component of the global climate system, due to its effect on solar reflectivity, its properties as an insulator of heat and mass exchange between the ocean and atmosphere and its importance in the formation of bottom waters.

2.1.1.1 Changing Arctic

In recent years, headlines of accelerated melting rates and decreasing sea ice thicknesses have led to the belief that rapid climate change is underway. This alone has led to an increased interest in the study of ocean processes in high latitude regions and their relative impact to the global climate.

Figure 2.3 shows the 30 year daily average (years 1981–2010) sea ice extent of the Arctic, as well as daily Arctic sea ice extent for years 2012 and 2014. This data was provided by the National Snow and Ice Data Center, which compiles data from three satellite sources. In September 2012, the sea ice reached its lowest ever recorded extent, at 3.41 million square kilometres, confirming the downward trend in sea ice extent.

2.1.1.2 Effects on ecosystems

The changes in sea ice cover also have consequences on the ecosystem of polar regions. For example, Adélie penguins in Antarctica rely on sea ice for migration [Ballard et al., 2010] as well as for their main food source. In the Arctic, polar bears rely on the sea ice to hunt and breed, and a changing Arctic will have consequences on polar bear population [Durner et al., 2009]. The changes of animal habits also affects the indigenous people of the Arctic, who rely on traditional hunting methods for survival.

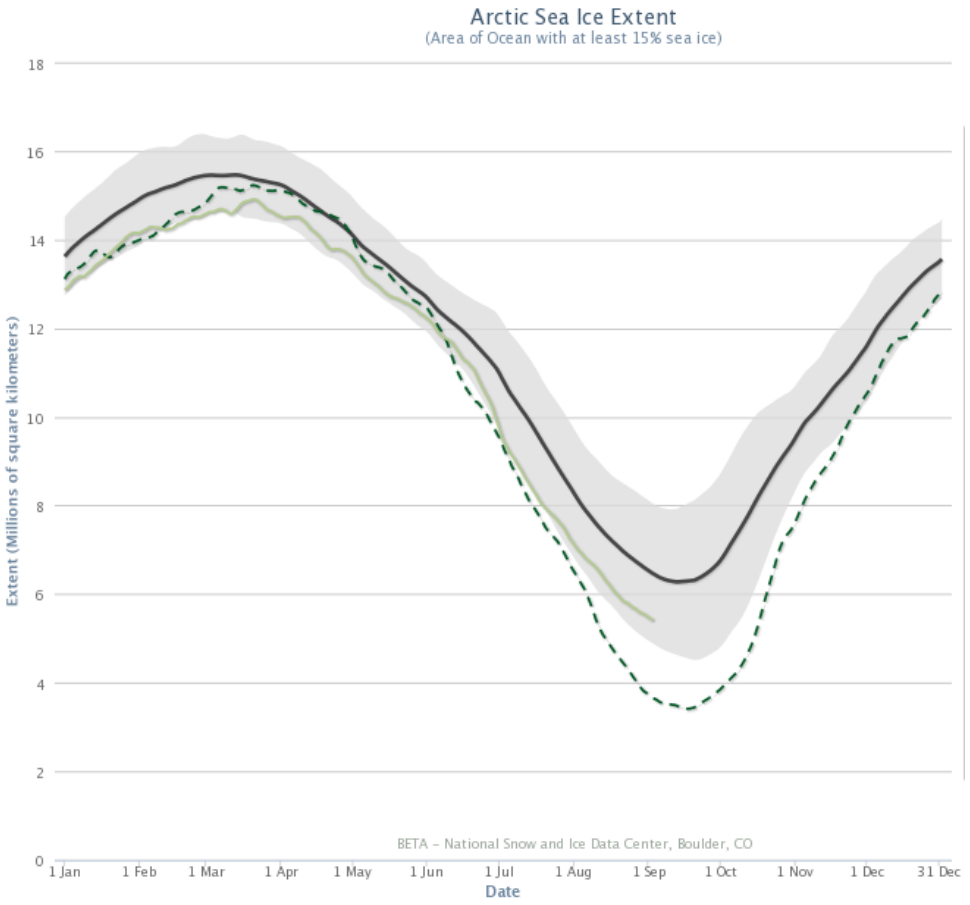


FIGURE 2.3: The daily average Arctic sea ice extent from years 1981–2010 is plotted in solid black. The shaded area denotes ±2 standard deviations. Sea ice extent for years 2014 (solid green line) and 2012 (dashed green line) is also plotted. Data obtained from the Sea Ice Index [Fetterer et al., 2002, updated daily].

2.1.1.3 Rewards of a warming north

A warming north also has the potential to have a large economic impact on the world. The opening of shipping routes along the Canadian Arctic Archipelago (ice–free for the first time in memory in 2005) cuts the shipping distance between western Europe and east Asia by one third. Shipping companies in China and other east Asian countries are already investing in ice–capable freighter ships.

In addition, countries in the region of the Arctic are looking to claim the potentially large oil and gas reserves that may be exploited when the sea ice has receded. For example, Russia's president was recorded as saying *"Offshore*

fields, especially in the Arctic, are without any exaggeration our strategic reserve for the 21st century". Shortly afterwards, a deal worth $500 billion was signed between ExxonMobil and Russia's Rosneft, to invest in developing offshore reserves [Economist, 2002].

2.1.1.4 Current modelling effort

Several modelling efforts have been made to forecast future sea ice extent. Although all climate models are in agreement that sea ice levels will decline in the 21^{st} century [Zhang and Walsh, 2006], none have been able to capture the extent of the current rate of decline [Stroeve et al., 2007]. This is shown in Figure 2.4, where the inset shows clearly that the mean of all models (labelled the ensemble mean and plotted using a solid black line) severely underestimate the rate of decline in the two decades preceding the end of observations (solid red line). This suggests that there is currently no model capable of accurately predicting future sea ice trends.

2.2 Notes on Model Coupling

2.2.1 Motivation

Model coupling enables specialised knowledge that has been channelled into component models to be used in a collaborative modelling effort. Coupled models are potentially better suited to addressing a wide variety of complex, multi–physics problems more effectively than standalone component models.

Model coupling has been used successfully in many fields, including nuclear transport, chemical engineering and CFD. It is also a key component of weather or climate forecasting and Earth system modelling, whereby it facilitates the building of more complete general circulation models.

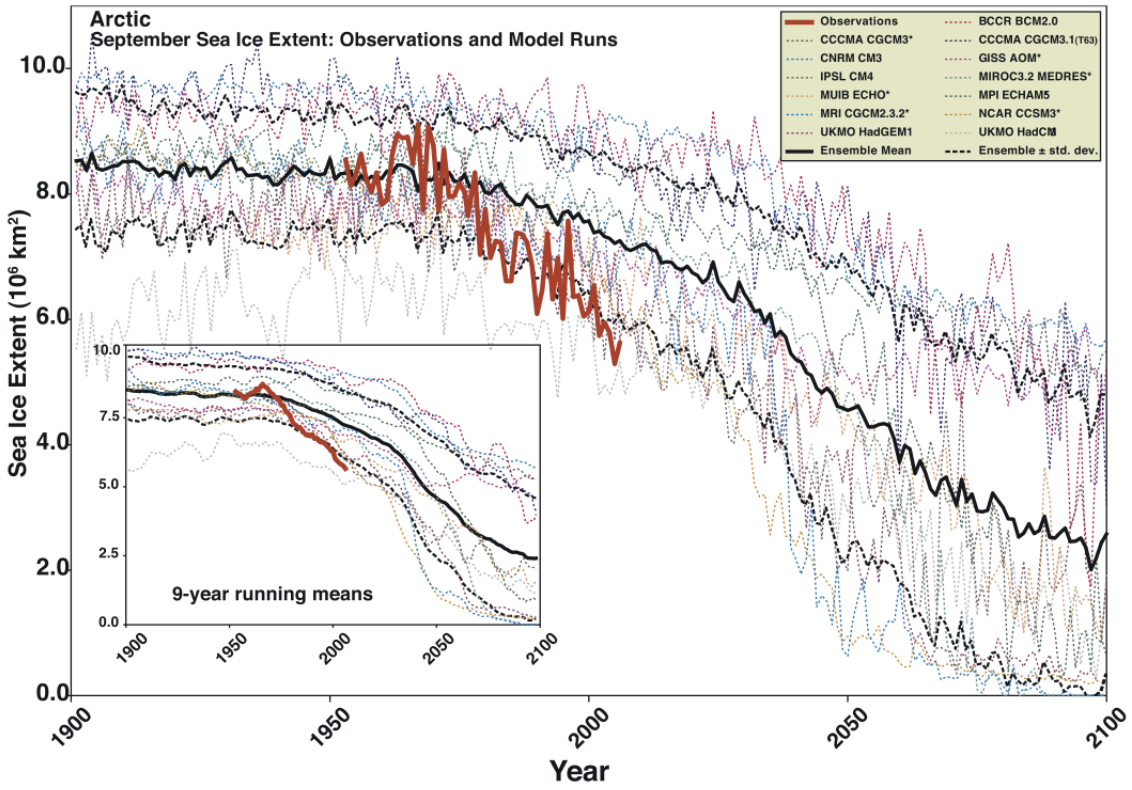


FIGURE 2.4: Arctic September sea ice extent from observations (thick red line) and 13 IPCC AR4 climate models, together with the multi-model ensemble mean (solid black line) and standard deviation (dotted black line). Inset shows 9-year running means. Taken from [Stroeve et al., 2007]

2.2.2 What is model coupling?

Coupling refers to the process of making two originally independent component models interact. Two models are *coupled* when data calculated from one model is used as forcing data or boundary constraints on another model. In its simplest form, a coupler involves data transfer between the two components. Couplers can also be responsible for:

Data restructure. Very often component models use different array structures such that fields are stored differently in memory. Given that each component model should not concern itself with the component model that it is being coupled to, it is the responsibility of the coupler to restructure data for the exchange. For example, one model may store data in an $m \times n$ 2–dimensional array, whilst the component model being coupled to it stores the field data in a 1–dimensional array of length

N, where $N = m \times n$. The coupling layer is responsible for restructuring data for the exchange.

Interpolation. Very often, component models will use different meshes and/or function spaces, that best suit the particular physics of the model. The process of transferring data from one mesh to another is called interpolation and is the responsibility of the coupling layer. Said interpolation often needs to adhere to several constraints: for example, it may need to preserve maxima and minima constraints, or to be volume or mass conserving. The constraints placed on the interpolation depend on the desired physics of the coupled model, as well as numerical method limitations.

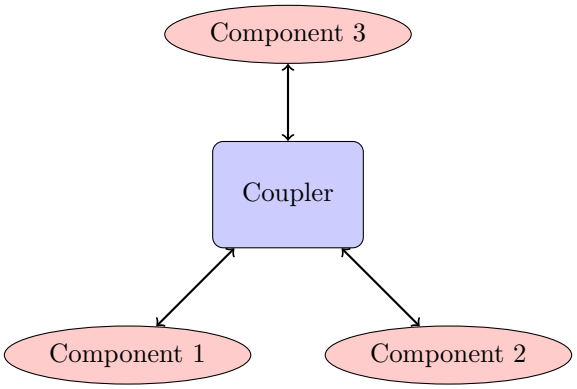
Time management and synchronisation. The coupler will be responsible for synchronising the two component models, by blocking component models from processing whilst computation from the other component model is completed. In addition, as different models solve for different physics, it is likely that different temporal and spatial scales are involved, requiring the use of different timesteps. The coupler may need to integrate quantities over time and exchange data only at the required frequency.

Calculation. The coupler may be responsible for various calculations, ranging from simple unit conversion to more complicated flux calculations. An example of this would be to convert wind speeds from an atmosphere component to surface drag on the ocean.

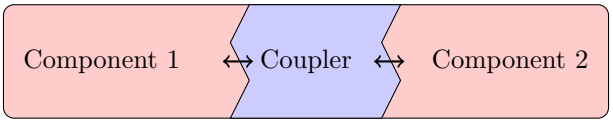
2.2.3 Coupling architecture

Coupling can generally be achieved in two different ways. One is the *multiple executable* architecture, in which component models maintain their modularity and a separate piece of software, the coupler, is used to channel communication across the components. In the multiple executable framework, data flow

is usually organised in a spoke and wheel fashion, as demonstrated in Figure 2.5a. In this scenario, data is communicated between the models using the message passing interface (MPI). Alternatively, the *integrated* (also known as *bespoke*) approach can be used, where the component models are integrated in a common executable and memory is readily shared between the models (see Figure 2.5b). In large coupled simulations, a hybrid of both architecture types may be used.



(A) Mutiple executable, spoke and wheel framework, where communication is controlled by the coupler and each component is a separate executable.



(B) Bespoke model coupling, where a single executable exists with a coupling layer controlling component communication.

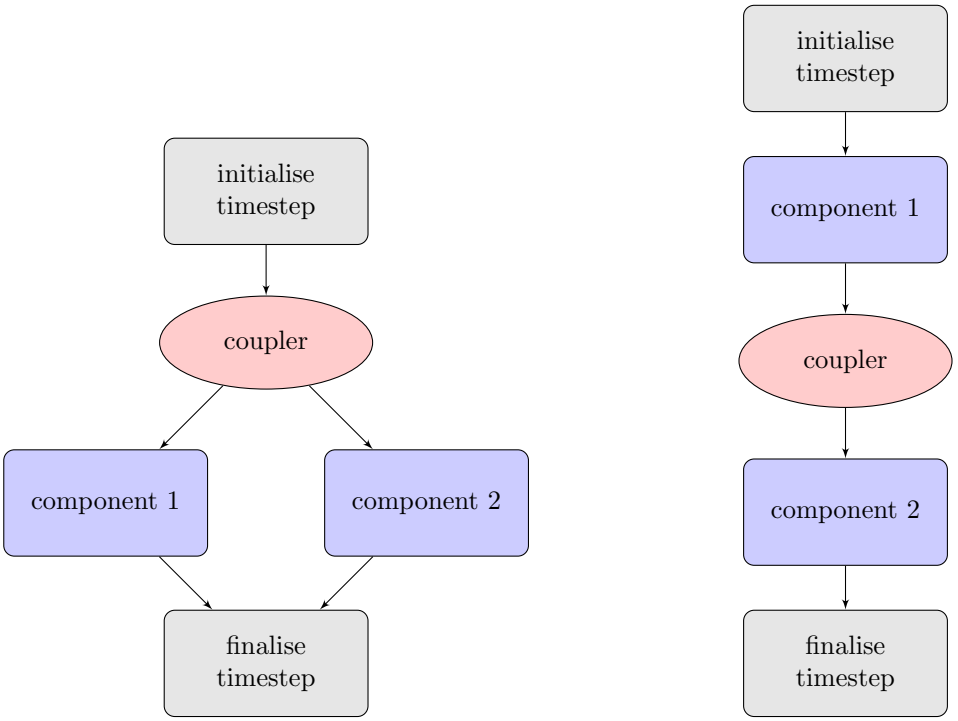
FIGURE 2.5: Coupling architectures

The multiple executable architecture is more appropriate for the coupling of two distinct component model codes, where development is carried out by different groups and the coupling infrastructure abides by a protocol that the coupling software stipulates. In theory, minimal code invasion would be necessary for this type of coupling, with just `send` and `receive` commands placed as necessary, in addition to the initialisation routine. Ideally, this coupling framework would be implemented in such a way that an array of other component models can be used in a *plug'n'play* fashion, without needing

a priori knowledge of the details of the component model with which each model will couple.

2.2.3.1 Parallel vs sequential processing

When coupling two component models, a framework decision that needs to be made is whether the component models will perform timesteps in parallel, or in sequence (see Figure 2.6). This has the potential to alter the overall runtime of a coupled model.



(A) Parallel. (B) Sequential.

FIGURE 2.6: Parallel and sequential execution models.

Component models that are timestepped in parallel exchange information at the end of each timestep. They are usually parallelisedwith the use of MPI. The component model that completes its operation first will need to pause and wait until processing in the slower of the two component models is completed. This difference in processing times for timesteps can sometimes be offset by running the more costly model on more processors. In contrast, component models that are run sequentially do not require calls to MPI for between model

communication. From a model development perspective, sequential processing is more straightforward to develop while parallel operation requires the set up of specific MPI communicators for each component model.

As a general rule, it is suggested that models with similar runtimes are timestepped in parallel, whilst models that vary greatly in runtime may be run in sequence.

2.2.3.2 Tight vs loose coupling

Two models are said to be tightly coupled when they share the same mesh and timestep. Tight coupling only requires one set of inputs from the user (mesh, timestep) when setting up a simulation. However, this particular setup should only be applied to component models that exhibit dynamics of similar spatial and temporal scales, where sharing the same mesh and timestep is a reasonable compromise.

Loose coupling, on the other hand, refers to coupling of component models in which each component model has its own mesh and timestep. In this setup, more care is required at the time of data exchange, in particular ensuring that data that has been integrated over several timesteps is exchanged at a common, appropriate timestep and that the data is interpolated from the mesh of one component model to the other.

2.2.4 Fluidity and FESIM

2.2.4.1 Initial investigation

The initial plan for this work was to implement a multiple executable framework, where both the ocean and sea ice model would be driven by the coupler. Ideally, the coupling framework would be at a level of abstraction such that the Fluidity ocean model would be able to couple to different sea ice models with

little to no changes required by the user. The coupler of choice was OASIS version 4.0 [Valcke, 2013], henceforth only referred to as OASIS[1]. Theoretically, and ideally, the coupling layer would be abstract and component models could be chosen in a *'plug-n-play'* fashion, allowing simulations to be run with a variety of other models that have also coupled to OASIS, or to seamlessly switch from a coupled simulation to a forced simulation, where data is read from a file. The initial investigation had three stages:

Sample models. Short, sample codes were tested that transferred data across OASIS. These small codes were written in order to learn and better understand the OASIS protocol interface.

festa – OASIS. A finite element model, festa, was coupled to OASIS.

Fluidity – OASIS. The initial investigation led to a prototype version of Fluidity coupled to OASIS.

2.2.4.2 Festa – OASIS

Festa, a finite element model written by Dr Candy at Imperial College, was coupled to OASIS for the purpose of investigating the coupling capabilities of OASIS. Festa was chosen due to its similarity with Fludity in terms of common data–structures and order of operation. Once coupled to OASIS, a forcing file was used that contained inflow boundary condition data. The OASIS coupler was responsible to transfer the forcing data from the file to festa. These tests were successful.

2.2.4.3 Fluidity – OASIS: The exchange mesh

During the development of the coupled Fluidity–OASIS two main limitations of OASIS were encountered. Firstly, the mesh over which the exchange would

[1]It must be noted, the OASIS coupler has been rewritten and released as version OASIS-MCT. The comments made in this thesis are not applicable to the new release, but rather refer specifically to OASIS version 4.0, which was the latest version available at the time of this work.

take place had to be defined as a structured grid (either regular, rectilinear or curvilinear) and secondly the mesh had to be defined at the initialisation stage of the simulation. The support for unstructured meshes was not implemented in OASIS and this feature did not look like it would have been delivered within the timescale of this project. For users of Fluidity, this meant that coupled simulations could not be run on unstructured meshes or adapt during the simulation, two key features of Fluidity.

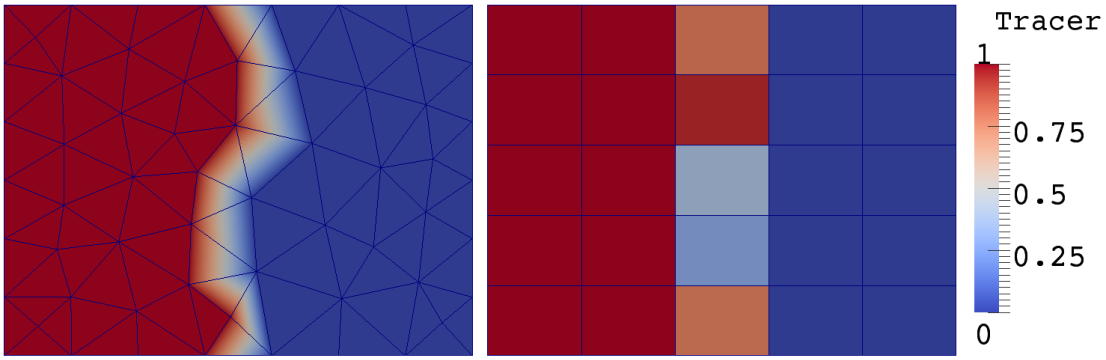


FIGURE 2.7: Consistent interpolation of a test tracer field from Fluidity's $\mathcal{P}1$ mesh (left) to the automatically generated *exchange mesh* (right), a $\mathcal{Q}0$ grid.

The solution to the above two constraints was the introduction of the *exchange mesh*, a structured mesh automatically generated at initialisation to which the Fluidity fields were interpolated on, before they are communicated with the OASIS coupler. The algorithm for the automatically generated mesh is given in appendix A. This allows Fluidity to solve for dynamics on meshes that do not comply with the OASIS coupler, but could still be communicated with other models through the exchange mesh. Figure 2.7 shows the generated exchange mesh being used in a test simulation, for the transfer of a tracer field defined as a step function. In this simulation, Fluidity was successfully coupled to OASIS, which retrieved the tracer field and dumped it to a file. In addition, Fluidity could be forced from a file, via the OASIS coupler.

After the initial investigation, it was decided to discontinue working with the OASIS coupler. The exchange mesh was a feasible workaround for Fluidity

coupling to OASIS, however introducing such an interpolation step in the coupling process can only be justified when coupling to a structured model, in which case the interpolation performed by Fluidity is only replacing the interpolation that would have been performed by the coupler. Given that the final objective of this project was to also couple with unstructured models, such as FESIM, the interpolation step to the exchange mesh introduced errors that could be avoided if the exchange occured over a more suitable, unstructured mesh.

2.2.4.4 How is Fluidity coupled to FESIM?

Fluidity and FESIM are instead coupled using a bespoke coupling architecture. The two models are restricted here to share the same topological mesh and timestep. The component models are run sequentially, with the sea ice model timestepping before the ocean model. Figure 2.8 describes the order of processes.

With reference to the processes of Figure 2.8, we make the following clarification on the purpose of each stage:

Fluidity initialisation. During Fluidity's initialisation, Fluidity reads in the Fluidity configuration file (FLML file) and the mesh generated by the user, it allocates fields that belong to Fluidity and applies any prescribed data, such as boundary conditions or prescribed field data. The mesh supplied by the user must be a 3–D mesh of the ocean domain, or a 2–D surface mesh with an extrusion to a given bathymetry specified in the Fluidity config file. The coupled ocean – sea ice model does not support a 2–D ocean mesh.

FESIM Initialisation. During FESIM initialisation, the coupler extracts the 2–D surface mesh from Fluidity and transforms it from the Cartesian coordinate system $[x, y, z]$ used by Fluidity to the spherical coordinate

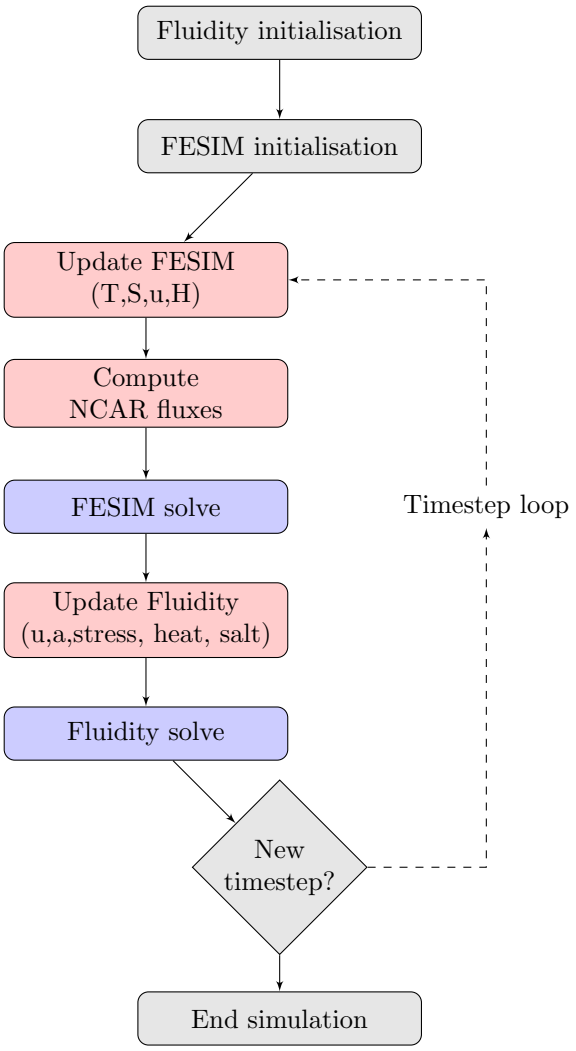


FIGURE 2.8: Flowchart outlining order of operations of the coupled Fluidity–FESIM model.

system used by FESIM $[\phi, \theta]$:

$$\phi = \arctan\left(\frac{x}{y}\right), \tag{2.1}$$

$$\theta = \arccos\left(\frac{z}{\sqrt{x^2 + y^2 + z^2}}\right), \tag{2.2}$$

where ϕ and θ are the longitude and latitude, respectively. The coupler then prescribes this as a mesh for the sea ice model. The meshes are topologically similar, but may vary in degrees of freedom as the user has the ability to choose the function space in which the ocean model solution is obtained. The sea ice model is restricted to a piecewise linear ($\mathcal{P}1$)

function space, whilst the ocean model allows for more flexibility. For example, the user may choose to solve for the ocean using a $\mathcal{P}1_{DG}\mathcal{P}2$ mixed finite-element function space. Note that the use of the above transformation for specifying the sea ice model mesh, constrains the ocean model mesh to be on a sphere. Other transformation functions can be implemented which would alleviate this constraint.

Update FESIM. During this step, the coupler populates fields required by the sea ice model. In particular, the surface temperature T, surface salinity S, sea surface ocean velocity $\mathbf{u} = [u, v]$ and sea surface elevation H are transferred to the sea ice model. On the first timestep, values are obtained from the model's prescribed boundary and initial conditions, thereafter they are prescribed from the output of Fluidity.

Update atmospheric forcing. In addition to the oceanic forcing, the sea ice model requires atmospheric forcing, which needs to be prescribed. Prescribed fields are the wind velocities $[ms^{-1}]$, shortwave and long-wave radiation $[Wm^{-2}]$, the atmospheric temperature $[C]$ and pressure $[hPa = 100Pa]$, dew point temperature $[C]$, precipitation rate $[ms^{-1}]$, cloudiness and specific humidity. This data can be obtained, for example, from reanalysis data such as ERA-40. Details of how this data is used to compute atmospheric forcing can be found in section 2.3.1.2.

FESIM solve. The FESIM solve is where the sea ice model is stepped forward in time. This process can be separated into three main stages:

1. **Dynamics computation**. In this stage of the FESIM solve, we compute for the sea ice velocity, in directions parallel to lines of constant latitude (zonal) and longitude (zonal). An overview of the equations and discretisation used is located in section 2.3.1.
2. **Scalar advection**. Given the sea ice velocity from the previous stage, the sea ice mass, concentration and snow mass are advected. These scalars are advected using a flux–corrected transport scheme, as described in Löhner et al. [1987]

3. **Thermodynamics computation**. The thermodynamics stage is where the thermodynamic source and sink terms are accounted for by increasing or decreasing the mass of sea ice and snow. The freshwater flux to the ocean is also computed at this stage. More information can be found in section 2.3.2.

Update Fluidity. Once the sea ice model FESIM is stepped forward in time, the coupler is responsible for updating Fluidity with the newly computed data. In addition to any data that is used for forcing, fields of the sea ice model (such as sea ice velocities and concentration) are also transferred to Fluidity, for I/O and visualisation purposes. The overhead of storing the sea ice data in memory twice is considered a small price for being able to output both model's results in a consistent format.

Fluidity solve. During this stage, Fluidity is stepped forward in time, solving for the temperature, salinity, and velocity of the ocean.

End simulation. The simulation ends when the simulation time has reached the end time or a steady state criterion is reached.

2.3 Equations of Motion and Thermodynamics

2.3.1 Dynamic model

In reality, the ice pack consists of discrete, rigid plates of ice called ice floes which are either closely packed together or drift freely in open waters. To make the modelling of such a system more tractable on a large scale, we consider the ice pack as a two–dimensional continuum which lies on the surface of the ocean, consisting of continuous fields of ice mass and concentration. We begin by considering the momentum equation:

$$F = m\frac{D\mathbf{u}}{Dt}, \tag{2.3}$$

where m represents the total mass of sea ice and snow, $\mathbf{u}$ is the velocity of the ice pack and $\frac{D}{Dt}$ is the material derivative, defined as:

$$\frac{D\phi}{Dt} = \frac{\partial\phi}{\partial t} + \mathbf{u}\cdot\nabla\phi. \tag{2.4}$$

F represents the forces that are applied to the ice pack, and consists of the following terms:

Wind and ocean stresses. The wind stress, τ_a, is the main driving force in the sea ice momentum equation. An analogous term is the ocean stress, τ_w, that originates from the shear of the underlying oceanic currents. Both these terms are calculated using the bulk formula:

$$\tau = \rho\, C_d\, |\mathbf{u}_{\text{rel}}|\, \mathbf{u}_{\text{rel}}, \tag{2.5}$$

where ρ is the density of the atmosphere or ocean, C_d is a representative drag coefficient and $\mathbf{u}_{\text{rel}}$ is the relative velocity of the sea ice and atmosphere or ocean. It must be noted that each stress computation is normalised by the sea ice concentration, which has been demonstrated by Connolley et al. [2004] to be more accurate in regions of low sea ice concentrations. The applied stress is therefore the product of the stress as computed above, multiplied by the sea ice concentration.

Coriolis effect. This is a pseudo force applied due to the rotation of the Earth.

Surface elevation. The ice pack moves from regions of high elevation to regions of low elevation, and this term accounts for that tendency. It is proportional to the slope of the ocean surface elevation.

Ice interaction. The ice interaction term, F_i, represents the stresses from within the ice itself.

The sea ice momentum equation can thus be written as:

$$\underbrace{\rho\frac{\partial u_i}{\partial t}}_{\text{acceleration}} = \underbrace{\frac{\partial \sigma_{ij}}{\partial x_j}}_{\text{ice interaction}} + \underbrace{\tau_{ai} + \tau_{wi}}_{\text{wind/ocean stress}} + \underbrace{\epsilon_{ij3} m f u_j}_{\text{Coriolis}} - \underbrace{mg\frac{\partial H_0}{\partial x_i}}_{\text{tilt}}, \tag{2.6}$$

where the terms are as described above, and ϵ_{ij3} is the Levi–Civita symbol (in this case $\epsilon_{ij3} = 0$ if $i = j$, 1 if $i = 1, j = 2$ and -1 if $i = 2, j = 1$). Unlike in Hibler [1979], but in line with other studies [Oberhuber, 1993] we have neglected the non–linear term $\mathbf{u} \cdot \nabla \mathbf{u}$. It can be shown that this term is an order of magnitude smaller than typical wind–stresses and therefore has negligible contribution.

The dynamics of a sea ice model are highly dependent on the rheology model chosen, that is, the choice of constitutive expression for evaluating the *internal stress* term, F_i. The rheology model determines the deformation and flow of sea ice under various load conditions. As related to the physical world, the rheology model controls the buildup of ice and ridging or the formation of leads, for example.

2.3.1.1 On the choice of Sea Ice Rheology

The evaluation of the stress tensor, σ, is given by the constitutive expression. In this section, we look at a representative sample of such constitutive expressions relevant to climate simulations:

- Viscous Plastic model (VP)

- Cavitating Fluid model (CF)
- Compressible Newtonian Fluid model (CNF)
- Free drift (with velocity correction) model (FDC)

A more thorough analysis of the above constitutive models was presented in Kreyscher et al. [2000], as part of the sea ice model intercomparison project, for the purpose of demonstrating that using a physically founded rhelogy (such as the VP model) gives more realistic results that agree better with observations.

The viscous–plastic (VP) model, based on Hibler [1979] is shown to be the most suitable for modelling sea ice, and is given by:

$$\sigma_{ij} = 2\eta\dot{\epsilon_{ij}} + (\zeta + \eta)\,\dot{\epsilon}_{kk}\delta_{ij} - \frac{P}{2}\delta_{ij}, \tag{2.7}$$

where the non–linear shear and bulk viscosities (ζ and η) are given by:

$$\zeta = \frac{P}{2\Delta}, \tag{2.8}$$

$$\eta = \frac{P}{2\Delta e^2}, \tag{2.9}$$

where e is the eccentricity of the yield surface of sea ice, and Δ is given by:

$$\Delta = \left[\left(\dot{\epsilon}_{11}^2 + \dot{\epsilon}_{22}^2\right)\left(1 + e^{-2}\right) + 4e^{-2}\dot{\epsilon}_{12}^2 + 2\dot{\epsilon}_{11}\dot{\epsilon}_{22}\left(1 - e^{-2}\right)\right]^{\frac{1}{2}}. \tag{2.10}$$

where $\dot{\epsilon_{ij}} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$.

P is a pressure, a measure of sea ice strength that depends on both thickness and concentration, and is calculated using:

$$P = P^* c H e^{-c^*(1-c)}, \tag{2.11}$$

where P^* and c^* are empirically calibrated constants. The viscosities increase with pressure, but decrease with increasing strain rates, ϵ. At the limit of zero strain rate, the viscosities tend to infinity. This behaviour is regulated by setting limits to the maximum value of the viscosity parameters.

The cavitating fluid model allows pack ice to diverge freely but deform plastically in compression. The internal ice force, F, is given by:

$$F = -\nabla p$$

where the internal ice pressure, p, is given by:

$$\begin{aligned} p = P \text{ for } \nabla u < 0, \\ p \in [0, P] \text{ for } \nabla u = 0, \\ p = 0 \text{ for } \nabla u > 0. \end{aligned}$$

P , also used in the Viscous Plastic model, is a measure of ice strength and is given by (2.11).

The compressible Newtonian fluid model is included in the discussion as a linear–viscous approach. In this model, the sea ice behaves as a film of "honey", whereby convergent and divergent forces are resisted by equal internal stresses, linearly proportional to the deformation rate. This behaviour is achieved by removing the dependence of the bulk and shear viscosities (ζ and η) on the strain rate.

The free drift with velocity correction model, the simplest of the rheology models, sets the internal stress to zero, $F = 0$. In regions of convergent forces, however, excessive ice thickness buildup is observed and thus a "correction" is applied to compensate for this; ice velocities are set to zero where the ice

thickness exceeds a specified maximum or where ice would flow from a lower to a higher thickness.

The non–linear viscous–plastic rheology suggested by Hibler [1979] has become the standard sea ice dynamics model due to its ability to accurately represent sea ice dynamics. However, it does exhibit some unfortunate numerical properties. The viscoplastic rheology becomes singular when the strain rate is zero. To overcome this, Hibler suggested a bounding value for the non–linear bulk and shear viscosities, ζ and η. Stability analysis of explicit discretisations of the VP rheology, even with bounded viscosities, leads to very stringent stability criteria, such that, for example, for a resolution of about 10km the timestep would need to be on the order of a hundredth of a second. Many improvements were suggested to address this, including the use of semi-implicit discretisation schemes.

A significant improvement was presented by Hunke and Dukowicz [1997], in the elastic–viscous–plastic (EVP) rheology model. The elastic term controls the behaviour as the bulk and shear viscosities get larger and approach the singular point. The elastic term vanishes in steady state and the VP rheology is recovered. The EVP formulation does not suffer from the stringent stability conditions of the VP model, while maintaining the viscous-plastic balance. The EVP model has been shown to be more efficient [Bouillon et al., 2009] and more accurate [Hunke and Zhang, 1999].

2.3.1.2 FESIM rheology model

The finite element sea ice model (FESIM) was developed at the Alfred Wegener Institute for Polar and Marine Research (AWI) and is used as part of the FESOM global circulation model [Timmermann et al., 2009]. Both the VP and EVP rheology models have been implemented in FESIM but only the EVP model is available in the coupled Fluidity–FESIM model.

The force balance implemented in the FESIM sea ice model is given by:

$$\rho \frac{\partial u_i}{\partial t} = F_i + \tau_{ai} + \tau_{wi} + \epsilon_{ij3} m f u_j - mg \frac{\partial H_0}{\partial x_i}, \tag{2.12}$$

where the main component is the wind stress, τ_a. The ocean stress, τ_w, and the effect of sea surface elevation, H_0, on sea ice velocity, $\boldsymbol{u}$, have been shown to be significant over long time scales. ρ is the density of ice and snow, and f and g are the Coriolis parameter and the magnitude of the acceleration due to gravity, respectively. The ice interaction term is essential in balancing the stress terms. For modelling the ice interaction term, the EVP constitutive law is used, which is composed of the viscous–plastic formulation given by:

$$\frac{1}{2\eta}\sigma_{ij} + \frac{\eta - \zeta}{4\eta\zeta}\sigma_{kk}\delta_{ij} + \frac{P}{4\zeta}\delta_{ij} = \dot{\epsilon}_{ij}. \tag{2.13}$$

modified by introducing an elastic term, proposed by Hunke and Dukowicz [1997]:

$$\frac{1}{E}\frac{\partial \sigma_{ij}}{\partial t} = \dot{\epsilon}_{ij} \tag{2.14}$$

where E corresponds to the Young's modulus. In combination with (2.13), we have the EVP constitutive model of FESIM:

$$\frac{1}{E}\frac{\partial \sigma_{ij}}{\partial t} + \frac{1}{2\eta}\sigma_{ij} + \frac{\eta - \zeta}{4\eta\zeta}\sigma_{kk}\delta_{ij} + \frac{P}{4\zeta}\delta_{ij} = \dot{\epsilon}_{ij}, \tag{2.15}$$

where η and ζ are as defined previously.

2.3.2 Thermodynamic model

Sea ice models vary in complexity. State–of–the–art models, such as the structured mesh CICE [Hunke and Lipscomb, 2010] for example, or the LIM [Vancoppenolle et al., 2009] model, are capable of dealing with several important parameters such as age, layering and growth rate of sea ice that influence the thermodynamic properties of sea ice. The thermodynamic component deals with the exchange of heat and salt at the ocean surface in response to atmospheric and oceanic forcing, as well as sea ice growth and melt.

The thermodynamic parameterisation used in Fluidity–FESIM is similar to the simplest model of Parkinson and Washington [1979]. It is based on energy balance principles, described below, with prognostic snow and ice layers.

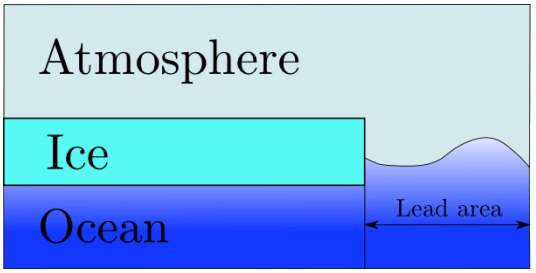


FIGURE 2.9: Schematic diagram showing the major divisions within an element of the sea ice model in the 3 layer configuration. A 4 layer configuration would include snow over the ice, whilst a 2 layer configuration would consist of the atmosphere and ocean only.

Figure 2.9 shows the major divisions within an element of the sea ice model. A layer of sea ice is situated between the atmosphere and ocean (3 layers). The model has the capability of dealing with situations where there is snow (4 layers) covering the sea ice or no snow and no ice (2 layers).

In the case of no ice, the net energy flux to the sea ice element is given by:

$$Q_{\text{net}} = H\downarrow + LE\downarrow + \epsilon_w LW\downarrow + (1-\alpha_w)SW\downarrow + F_w\uparrow - \epsilon_w \sigma T_w^4 \qquad (2.16)$$

where the arrows indicate whether the flux is atmospheric (down arrow) or oceanic (up arrow). Each term is explained in Table 2.2.

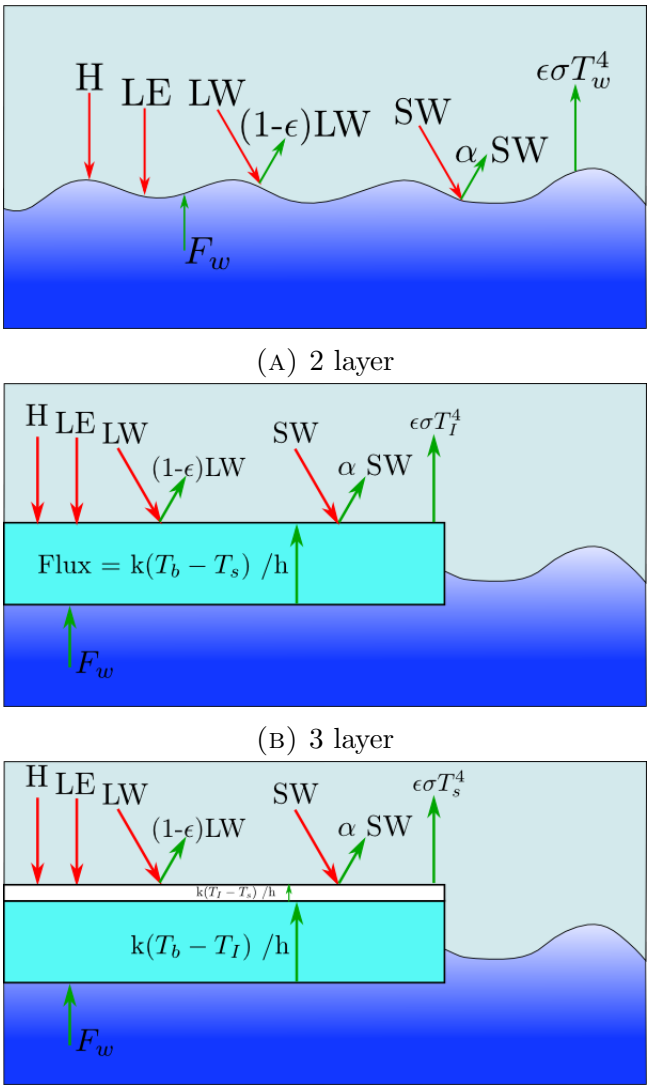
Term	Name	Description
$H\downarrow$	Sensible Heat	Sensible heat is energy exchanged due to a change in temperature. It is given by the formula $H\downarrow = \rho_a c_p C_H V_{wg}(T_a - T_{sfc})$. ρ_a is the atmospheric density, c_p is the specific heat of air, C_H the transfer coefficient for sensible heat, V_{wg} the magnitude of sea level winds. $(T_a - T_{sfc})$ is the temperature difference between the atmospheric and surface temperatures.
$LE\downarrow$	Latent Heat	Latent heat is energy exchanged due to a change of phase. It is given by the formula $LE\downarrow = \rho_a L C_E V_{wg}(q_{10m} - q_s)$. C_E is a latent heat transfer coefficient and $(q_{10m} - q_s)$ is the difference in specific humidities at 10m elevation and sea level.
$LW\downarrow$	Longwave Radiation	Longwave radiation is the energy exchanged from infrared radiation. In the model, it is parameterised by $F\downarrow = \sigma T_a^4 \{1 - 0.261 \exp[-7.77 \times 10^{-4}(273 - T_a)^2]\}$, modified by a cloud factor. In (2.16) it is also modified by a longwave emissivity factor, ϵ_w.
$SW\downarrow$	Solar Radiation	Solar radiation is parameterised according to latitude, solar time and cloud cover. In (2.16) it is also modified by the surface albedo coefficient, a_w.
$F_w\uparrow$	Upward Heat Flux	The upward heat flux has been shown to be directly proportional to the temperature difference of the water and ice. In this model, it is considered a constant.

TABLE 2.2: Table describing terms of (2.16)

The temperature balance determines the change in ocean temperature

$$\frac{dI}{dt} = Q_{\text{net}} \tag{2.17}$$

where I is the ocean's internal energy, and $I \propto T$. Given the ocean's heat capacity, the change of temperature can be calculated. If the temperature falls below $271.2K(-1.95^oC)$, ice forms. The ice thickness is chosen to be 0.01m and the extent is calculated using the heat of fusion coefficient, such that the temperature of the water remains at freezing. (2.16) is illustrated in Figure 2.10a. Figures 2.10b and 2.10c illustrate the 3-layer and 4-layer thermodynamics.



(A) 2 layer

(B) 3 layer

(C) 4 layer

FIGURE 2.10: Illustration of sea ice model thermodynamics, for cases of no ice, ice only and ice with snow, from [Parkinson and Washington, 1979]

2.4 Model verification and validation

2.4.1 Method of Manufactured Solutions

The method of manufactured solutions [Roache, 2001] is used here to verify correctness of the sea ice dynamics numerical model implementation in Fluidity–FESIM. This is done by manufacturing an analytical solution to the problem to which the model output can be compared. Once an error metric is obtained for each simulation, a convergence analysis can be performed, a

check on whether the error decreases at the expected rate as the resolution of the simulation increases. This is a very strong check of the correctness in the implementation of a model.

We begin by considering the sea ice momentum equation:

$$\rho \frac{\partial u_i}{\partial t} = \frac{\partial \sigma_{ij}}{\partial x_j} + \tau_a + \tau_o + \epsilon_{ij3} m f u_j - mg \frac{\partial H_0}{\partial x_i}, \tag{2.18}$$

and proceed by making the following simplifications:

- Coriolis parameter, $f = 0$;
- Ice–ocean drag coefficient, $C_{d_o} = 0$, therefore $\tau_o = 0$;
- Sea surface elevation, $H_0 = 0$;
- The wind–forcing, τ_a will be explicitly prescribed.

These simplifications, although not necessary, allow us to concentrate on the ice–interaction term. Considering, temporarily, the wind–forcing $\tau_a = 0$, the momentum equation reduces to:

$$\rho \frac{\partial \mathbf{u}}{\partial t} = \nabla \cdot \sigma, \tag{2.19}$$

which can be cast in terms of a linear operator $\mathcal{L}(\mathbf{u}) = 0$, where:

$$\mathcal{L}(\mathbf{u}) = \rho \frac{\partial \mathbf{u}}{\partial t} - \nabla \cdot \sigma. \tag{2.20}$$

We then choose a *manufactured* solution $\mathbf{u} = \mathbf{U_m}$, such that:

$$\mathcal{L}(\mathbf{U_m}) = \rho\frac{\partial \mathbf{U_m}}{\partial t} - \nabla \cdot \sigma = \mathbf{Q_m}, \tag{2.21}$$

where $\mathbf{Q_m}$ is the remainder. We expect $\mathbf{Q_m} = \mathbf{0}$ when $\mathbf{U_m}$ is a solution to (2.19). Since our manufactured, arbitrary choice of $\mathbf{U_m}$ is not based on any physical consideration of the dynamics of sea ice, it is unlikely that we have $\mathbf{Q_m} = \mathbf{0}$, but rather we expect $\mathbf{U_m}$ to be a solution to the modified problem statement:

$$\rho\frac{\partial \mathbf{u}}{\partial t} = \nabla \cdot \sigma + \mathbf{Q_m}. \tag{2.22}$$

where $\mathbf{Q_m} = \mathcal{L}(\mathbf{U_m})$ and can be prescribed in the model by setting $\tau_\mathbf{a} = \mathbf{Q_m}$. For this test, we choose the manufactured solution:

$$\mathbf{U_m} = \begin{pmatrix} \sin(4x) \\ 0 \end{pmatrix}, \tag{2.23}$$

which gives us:

$$Q_{m_1} = \rho\frac{\partial(\sin(4x))}{\partial t} - \frac{\partial \sigma_{11}}{\partial x} - \frac{\partial \sigma_{12}}{\partial y} \tag{2.24}$$

$$= -\frac{\partial \sigma_{11}}{\partial x} - \frac{\partial \sigma_{12}}{\partial y}, \tag{2.25}$$

and

$$Q_{m_2} = \rho \frac{\partial(0)}{\partial t} - \frac{\partial \sigma_{12}}{\partial x} - \frac{\partial \sigma_{22}}{\partial y} \tag{2.26}$$

$$= -\frac{\partial \sigma_{12}}{\partial x} - \frac{\partial \sigma_{22}}{\partial y}, \tag{2.27}$$

where subscripts 1 and 2 refer to the x and y directions respectively, and the wind forcing, τ_{a_i}, can be used to apply the source term Q_{m_i} in the model setup. The components of the stress tensor σ (from 2.15) can be calculated, as in the CICE model [Hunke and Lipscomb, 2010], by:

$$\frac{\partial \sigma_D}{\partial t} + \frac{\sigma_D}{2T_{EVP}} + \frac{P}{2T_{EVP}} = \frac{P}{2T_{EVP}\Delta}\epsilon_D, \tag{2.28}$$

$$\frac{\partial \sigma_T}{\partial t} + e^2 \frac{\sigma_T}{2T_{EVP}} = \frac{P}{2T_{EVP}\Delta}\epsilon_T, \tag{2.29}$$

$$\frac{\partial \sigma_{12}}{\partial t} + e^2 \frac{\sigma_{12}}{2T_{EVP}} = \frac{P}{2T_{EVP}\Delta}\epsilon_{12}, \tag{2.30}$$

where we have defined:

$$\sigma_D = \sigma_{11} + \sigma_{22}, \tag{2.31}$$

$$\sigma_T = \sigma_{11} - \sigma_{22}, \tag{2.32}$$

and similarly for the strain rate, $\epsilon_D = \epsilon_{11} + \epsilon_{22}$ and $\epsilon_T = \epsilon_{11} - \epsilon_{22}$. In this testcase, we only consider the limited scenario of $\Delta = \Delta_{\min}$ is a constant.

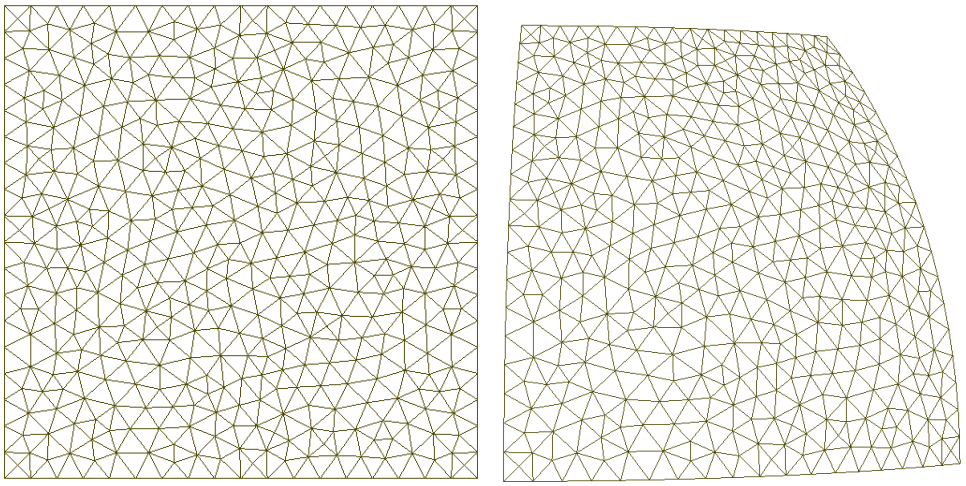
The simulation covers the region $[0, 45°] \times [0, 45°]$ in longitude, latitude, and the spherical mesh is produced by first generating a flat, two–dimensional mesh with limits $[0, 45] \times [0, 45]$. This mesh is generated using Gmsh [Geuzaine and Remacle, 2009]. This is then fed through a script that applies the following

projection:

$$x = R\cos(\phi)\cos(\lambda), \tag{2.33}$$

$$y = R\sin(\phi)\cos(\lambda), \tag{2.34}$$

$$z = R\sin(\lambda), \tag{2.35}$$



(A) Mesh before projection, in spherical coordinates of dimension $[0, 45°] \times [0, 45°]$.

(B) Mesh after projection, in 3–D Cartesian coordinates within $[2, 637100] \times [0, 4504984] \times [0, 4504984]$

FIGURE 2.11: Preprocessing step: Mesh projection for sea ice simulations.

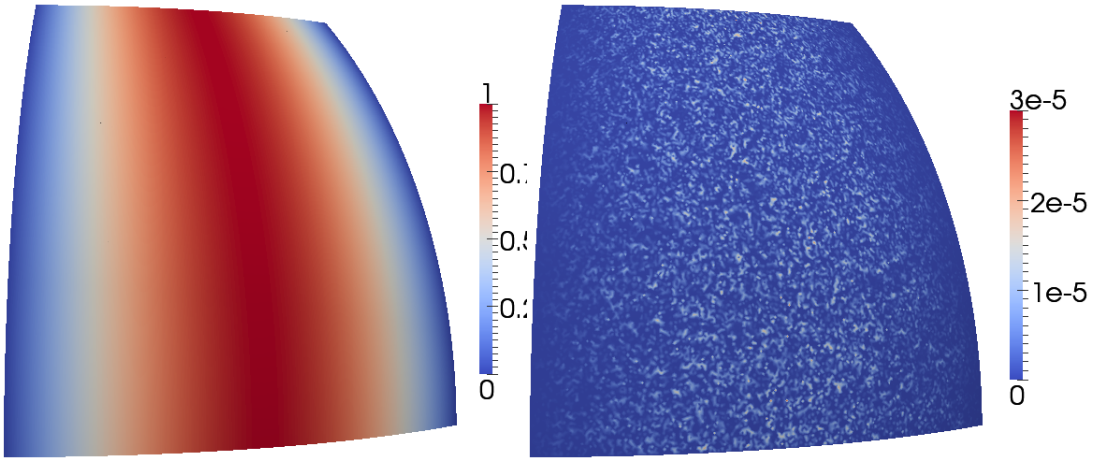
where R is the radius of the Earth and ϕ, λ are the longitude and latitude, respectively. We have thus transformed the mesh from the spherical coordinate system to the Cartesian coordinate system of a two–dimensional mesh on the sphere. The two–dimensional spherical mesh in Cartesian coordinates is used in Fluidity and extruded to an appropriate ocean depth with a single layer.

Five simulations are performed on meshes of progressively higher resolution, ranging from 4° node separation in the lowest resolution run, to 0.25° node separation in the highest resolution, doubling the resolution for each simulation.

The result of each simulation is compared with the analytical manufactured solution in the infinity–norm:

$$\text{Error} = \|u_i - U_{m_i}\|_\infty = \max_i (u_i - U_{m_i}), \tag{2.36}$$

where the subscript i indexes each node location and u and U_m represent the calculated and analytical solutions, respectively.



(A) Chosen manufactured velocity field, U_m, plot of (2.23).

(B) Absolute error in velocity field, for highest resolution mesh.

FIGURE 2.12: MMS simulation results, on a spherical domain with limits $[0, 45°] \times [0, 45°]$.

A plot of the absolute error, $|u_i - U_{m_i}|$, in the sea ice velocity field (Figure 2.12b) shows the error is spread across the domain and not dominated by the boundaries or any particular feature of the flow.

The error is plotted against the characteristic element edge length in Figure 2.13. The sea ice model is discretised on a $\mathcal{P}_1$ function space, and thus we expect second–order error convergence. The plot shows successful second–order convergence and we can thus safely deduce that the Fluidity–FESIM model correctly implements and solves the sea ice dynamical equations.

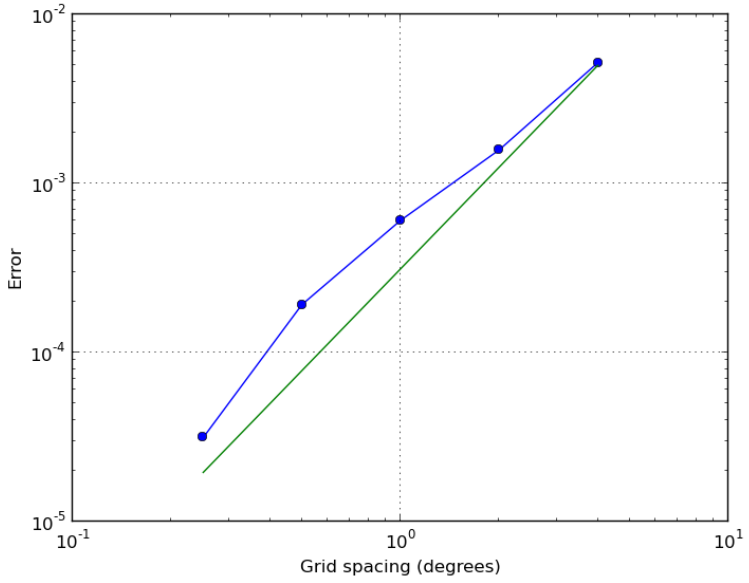


FIGURE 2.13: Error in the infinity norm (blue line) for the MMS test described in section 2.4.1. A line (green) showing second–order convergence is also plotted for comparison.

2.4.2 Pseudo one–dimensional test

We proceed with the model testing by considering a one–dimensional domain of length L, with solid boundaries on either side, as illustrated in Figure 2.14. This testcase was first proposed by Hunke and Dukowicz [1997] for the purpose of comparison between the viscous–plastic and elastic–viscous–plastic rheology models. Further, it was used by Gao et al. [2011] in the validation of their finite–volume, unstructured grid sea ice model ug–CICE.

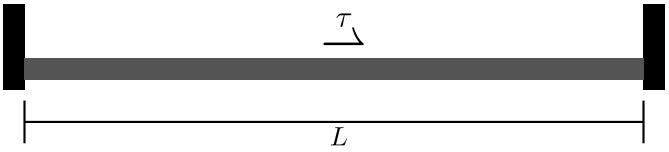


FIGURE 2.14: Illustration of pseudo one–dimensional testcase. Sea ice of constant thickness and concentration in a domain of length L, subjected to a stress τ.

The sea ice is subjected to a constant total stress τ. The solid boundaries do not allow flow of ice across the domain boundaries.

Considering equations (2.12) and (2.15) in their one–dimensional form, and lumping all external forcing in a single term, τ, we have:

$$m\frac{\partial u}{\partial t} = \frac{\partial \sigma_{11}}{\partial x} + \tau \tag{2.37}$$

and

$$\frac{1}{E}\frac{\partial \sigma_{11}}{\partial t} + \frac{\sigma_{11}}{2\eta} + \frac{\eta - \zeta}{4\eta\zeta}(\sigma_{11} + \sigma_{22}) + \frac{P}{4\zeta} = \frac{\partial u}{\partial x}, \tag{2.38}$$

$$\frac{1}{E}\frac{\partial \sigma_{12}}{\partial t} + \frac{\sigma_{12}}{2\eta} = 0, \tag{2.39}$$

$$\frac{1}{E}\frac{\partial \sigma_{22}}{\partial t} + \frac{\sigma_{22}}{2\eta} + \frac{\eta - \zeta}{4\eta\zeta}(\sigma_{11} + \sigma_{22}) + \frac{P}{4\zeta} = 0. \tag{2.40}$$

Considering the equations during steady state, and solving the system of simultaneous equations for σ_{ij}, we are left with:

$$\frac{\partial \sigma_{11}}{\partial x} = -\tau, \tag{2.41}$$

and

$$\sigma_{11} = (\zeta + \eta)\frac{\partial u}{\partial x} - \frac{P}{2}, \tag{2.42}$$

$$\sigma_{12} = 0, \tag{2.43}$$

$$\sigma_{22} = (\zeta - \eta)\frac{\partial u}{\partial x} - \frac{P}{2}. \tag{2.44}$$

Substituting (2.42) into (2.41) and noting that $\eta = \frac{\zeta}{e^2} = \frac{\zeta}{4}$ for an eccentricity $e = 2$ and P is constant, we have:

$$\frac{\partial}{\partial x}\left(\frac{5\zeta}{4}\frac{\partial u}{\partial x} + \tau x\right) = 0, \tag{2.45}$$

where $\zeta = \frac{P}{2\Delta}$ and $\Delta = \left[(\epsilon_{11}^2 + \epsilon_{22}^2)\left(1 + e^{-2}\right) + 4e^{-2}\epsilon_{12}^2 + 2\epsilon_{11}\epsilon_{22}\left(1 - e^{-2}\right)\right]^{\frac{1}{2}}$.

Considering that only $\epsilon_{11} = \partial u/\partial x$ is non–zero in one–dimension, we have:

$$\Delta = \frac{\sqrt{5}}{2}\left|\frac{\partial u}{\partial x}\right|. \tag{2.46}$$

The viscosity ζ can therefore have one of three values, $\zeta_{\min}$, $\zeta_{\max}$ or:

$$\zeta = \frac{P}{\sqrt{5}\left|\frac{\partial u}{\partial x}\right|}. \tag{2.47}$$

For the cases where $\zeta = \zeta_{\min}$ or $\zeta_{\max}$, integrating (2.45) gives:

$$\frac{5\zeta}{4}\frac{\partial u}{\partial x} + \tau x = C', \tag{2.48}$$

where C' is a constant of integration. We can rearrange for $\frac{\partial u}{\partial x}$:

$$\frac{\partial u}{\partial x} = \frac{4}{5\zeta}\left(C' - \tau x\right). \tag{2.49}$$

Given τ is a constant, we can define a new constant C such that:

$$\frac{\partial u}{\partial x} = \frac{4\tau}{5\zeta}\left(C - x\right). \tag{2.50}$$

Given the symmetry of the problem, we know that $\frac{\partial u}{\partial x}\big|_{x=L/2} = 0$ and therefore $C = L/2$:

$$\frac{\partial u}{\partial x} = \frac{4\tau}{5\zeta}\left(\frac{L}{2} - x\right), \quad \text{for } \zeta = \zeta_{\min} \text{ or } \zeta = \zeta_{\max} \tag{2.51}$$

We now consider the case when $\zeta = \frac{P}{\sqrt{5}\left|\frac{\partial u}{\partial x}\right|}$, from (2.51):

$$\frac{\partial u}{\partial x} = \frac{4\tau\left|\frac{\partial u}{\partial x}\right|}{\sqrt{5}P}\left(\frac{L}{2} - x\right), \tag{2.52}$$

i.e. we now have an implicit relationship for $\frac{\partial u}{\partial x}$.

Defining $P' = P/\sqrt{5}$ and $G = \frac{4\tau}{\sqrt{5}P}\left(\frac{L}{2} - x\right)$, we can write:

$$\frac{\partial u}{\partial x} = \begin{cases} GP'/\zeta_{\max} & \text{if} \quad \zeta > \zeta_{\max} \\ G\left|\frac{\partial u}{\partial x}\right| & \text{if} \quad \zeta_{\min} < \zeta < \zeta_{\max} \\ GP'/\zeta_{\min} & \text{if} \quad \zeta < \zeta_{\min} \end{cases} \tag{2.53}$$

Since the above is an implicit relationship, we can solve using a fixed–point iteration:

$$\frac{\partial u^{n+1}}{\partial x} = \begin{cases} GP'/\zeta_{\max} & \text{if} \quad \zeta > \zeta_{\max} \\ G\left|\frac{\partial u^n}{\partial x}\right| & \text{if} \quad \zeta_{\min} < \zeta < \zeta_{\max} \\ GP'/\zeta_{\min} & \text{if} \quad \zeta < \zeta_{\min} \end{cases} \tag{2.54}$$

where n represents the iteration level in an iteration to convergence.

The 'analytical' expression (2.54) can be integrated using a simple finite difference script, to produce a solution to (2.41), under the one–dimensional, steady state conditions prescribed in this testcase. $\frac{\partial u}{\partial x}$ can then be numerically integrated to yield the solution u. The analytical solution is plotted in green in figure 2.15a.

For this simulation, a full three–dimensional domain is used, spanning 12° in longitude and latitude (in the region $[6°, 14°]$), and $1km$ depth (in a single layer). This equates to approximately $508km$ in the x–direction. A characteristic element edge length of $12.7km$ is used, equivalent to approximately 40 elements across, in line with other studies. The sea ice mesh is extracted from the surface of the ocean. Note that the boundaries in the y–direction are far enough to have minimal effect on the result. Increasing or decreasing this distance showed insignificant changes to the result obtained. A constant forcing $\tau = 0.9$ is applied to sea ice of thickness $0.6m$ and concentration 0.1. The forcing is applied from the wind forcing term. $\zeta_{\min}$ and $\zeta_{\max}$ are set to

4×10^8 and $4 \times 10^6 P$, respectively. P is calculated as in (2.11) using constants $P^* = 27500$ and $c^* = 20$. The solution to the problem is visualised in figure 2.16a. For comparison, the viscous–plastic result from Hunke and Dukowicz [1997] is also shown.

The sea ice velocity across the centreline is extracted and plotted (blue line) in figure 2.15a. The solution is inherently two–dimensional and this accounts for the discrepancy in the magnitude of the sea ice in the middle section. Furthermore, the simulation is unable to capture the sharp changes of the analytical result that are present when changing from regions where $\zeta = \zeta_{\min}$ and $\zeta = \zeta_{\min}$, as the simulation changes slowly from one region to the other. The result obtained with Fluidity–FESIM is in line with other studies [Gao et al., 2011, Hunke and Dukowicz, 1997]. In particular, the Fluidity–FESIM result matches very closely with the results obtained in the original paper of Hunke and Dukowicz [1997] (shown in figure 2.15b for comparison). It must be noted that the dotted line marked '1D' is equivalent to the line marked 'Analytical' in figure 2.15a, however, the result plotted is from (2.54) integrated up to iteration level 30, before the iteration reaches steady state as is suggested (which requires approximately 140 iterations). The two analytical results are therefore different. However, we note that the results simulated with Fluidity–FESIM reach the same maximal value as the results in Hunke and Dukowicz [1997], and also deviate from the analytical solution at approximately the same location (distance index 8, velocity magnitude 37.5). We therefore consider this a successful comparison and are confident that the sea ice model implemented in Fluidity–FESIM is up to the standard of sea ice models used in climate studies.

2.4.3 Current – Ice interaction test.

As has been alluded to previously (section 2.3.1.1), one feature of a realistic rheological model is to allow the ice to drift freely in regions of open water, yet introduce internal stresses that resist such motion in regions of high sea

ice concentration, due to interactions of the ice pack with itself. This test is designed to investigate the interaction between currents and sea ice in regions of different sea ice concentrations.

We consider a square domain of length $1.28 \times 10^3 km$, and specify sea ice concentration as varying linearly from 0.0 at the western boundary to 1.0 at the eastern boundary. The origin (coordinates [0,0]) is at the centre of the domain. The ocean velocity is prescribed by:

$$u_w = 0.1 \left(\frac{y}{R} \right), \tag{2.55}$$

$$v_w = -0.1 \left(\frac{x}{R} \right), \tag{2.56}$$

where $R = \sqrt{x^2 + y^2}$, is the distance from the origin. This produces concentric ocean currents, centred around the origin and of magnitude linearly proportional to the distance from the origin (figure 2.17). The wind is constant throughout the simulation and is set to blow from a southwestern direction, with $u_a = v_a = 5ms^{-1}$.

The sea ice momentum equations can be written as:

$$m \frac{\partial u}{\partial t} = \frac{\partial \sigma_{1j}}{\partial x_j} + \tau_x, \tag{2.57}$$

$$m \frac{\partial v}{\partial t} = \frac{\partial \sigma_{2j}}{\partial x_j} + \tau_y, \tag{2.58}$$

$$\tag{2.59}$$

where τ represents the ocean (τ_w) and wind (τ_a) forcing terms. These terms are given by:

$$\tau_{ax} = cC_a \rho_a \left| \mathbf{u_a} \right| \left(u_a \cos\phi - v_a \sin\phi \right), \tag{2.60}$$

$$\tau_{ay} = cC_a \rho_a \left| \mathbf{u_a} \right| \left(u_a \sin\phi + v_a \cos\phi \right), \tag{2.61}$$

and

$$\tau_{wx} = C_w \rho_w \left|\mathbf{u_w} - \mathbf{u}\right| \left[(u_w - u)\cos\theta - (v_w - v)\sin\theta\right], \tag{2.62}$$

$$\tau_{wy} = C_w \rho_w \left|\mathbf{u_w} - \mathbf{u}\right| \left[(v_w - v)\cos\theta + (u_w - u)\sin\theta\right], \tag{2.63}$$

where $\phi = \theta = 25°$ are the wind and ocean turning angles.

Under steady–state ($\frac{\partial u}{\partial t} = 0$) and free–drift ($\frac{\partial \sigma_{ij}}{\partial x_j} = 0$) conditions, the sea ice velocity, $\mathbf{u}$, is balanced only by the wind and ocean stresses. We can therefore solve for it by considering:

$$\tau_{ax} + \tau_{wx} = 0, \tag{2.64}$$

$$\tau_{ay} + \tau_{wy} = 0. \tag{2.65}$$

Manipulating the above system allows us to write an analytical expression for the steady–state, free–drift sea ice velocity as:

$$\begin{aligned} u &= u_w + \frac{1}{V}\left[\tau_{ax}\cos\theta + \tau_{ay}\sin\theta\right], \\ v &= v_w - \frac{1}{V}\left[\tau_{ax}\sin\theta - \tau_{ay}\cos\theta\right], \end{aligned} \tag{2.66}$$

where $V^2 = C_w \rho_w \left(\tau_{ax}^2 + \tau_{ay}^2\right)^{1/2}$.

Given the ocean and wind forcing conditions specified above and the parameters specified in table 2.3 the steady–state, free-drift ocean velocity computed using (2.66) is illustrated in subfigures A, C and E of figure 2.18.

On the right hand side of figure 2.18 the result of the problem simulated using Fluidity–FESIM is shown. The analytical result is computed on a grid of $320\times$

Parameter	Value
C_a	5.0×10^{-4}
ρ_a	1.2
C_w	5.5×10^{-3}
ρ_w	1026.0

TABLE 2.3: Parameters used in simulation of testcase described in section 2.4.3.

320 equidistant points. The Fluidity–FESIM simulation was on a triangular mesh with a characteristic element edge length of $12km$, but the result is probed at the same points. Note that within Fluidity–FESIM, we solve the full system of equations and have not made the assumption of free–drift flow. The difference between the two results is therefore proportional to the discrepancy between the result of the full system and the free–drift solution. Figure 2.19 shows the difference in the x–component of the two solutions. The large error in the bottom left of the figure is due to the boundary condition affecting the result. In the rest of the domain, we observe that the discrepancy is minimal towards the western boundary and increases towards the eastern boundary. This indicates that the model exhibits free–drift behaviour in regions of low sea ice concentration, but deviates from free–drift as the concentration increases, as is expected from a realistic sea ice rheology model.

2.5 Conclusions

The coupled Fluidity–FESIM model was developed as part of this thesis. Results of using this model on idealised, small–scale problems has been demonstrated. The model is shown to be capable of performing such coupled ocean – sea ice simulations.

It must be noted that despite the two–way coupling interface being fully developed, all tests described in this thesis have been run with the oceanic state being fully prescribed. As a consequence, dynamical feedback from the ice

to the ocean has not been shown and therefore effectively only the one–way coupling functionality has been fully demonstrated.

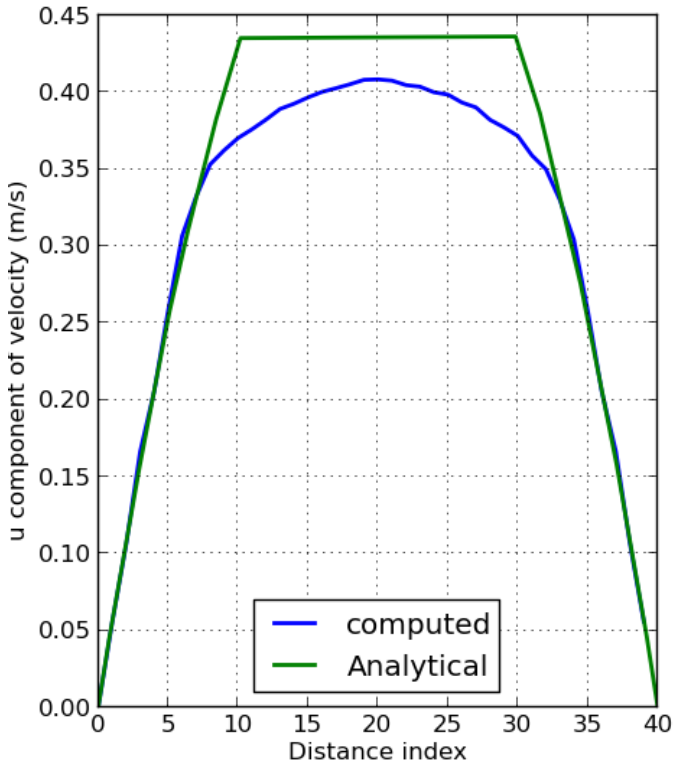
As part of verifying the model, two tests were performed. The first was a test using the method of manufactured solutions. The test showed that the error in our simulations decreases at second–order, matching the theoretical order of convergence of the $\mathcal{P}1$ discretisation used in the sea ice model, indicating that the model is correctly implemented.

The sea ice rheology was further validated using a pseudo–one dimensional test. Despite an analytical solution to the full, non–linear, sea ice momentum equations not being available, a solution may be obtained for one–dimensionalised simplifications of these equations. One such solution, described above (section 2.4.2) and originally proposed by Hunke and Dukowicz [1997] is used here to verify the implementation of the rheological model in Fluidity–FESIM. The model exhibits good agreement with the analytical solution, to the degree expected given the simplifications in deriving the analytical expression. Furthermore, Fluidity–FESIM's result is in line with other models, such as CICE [Hunke and Lipscomb, 2010] and ug–CICE [Gao et al., 2011].

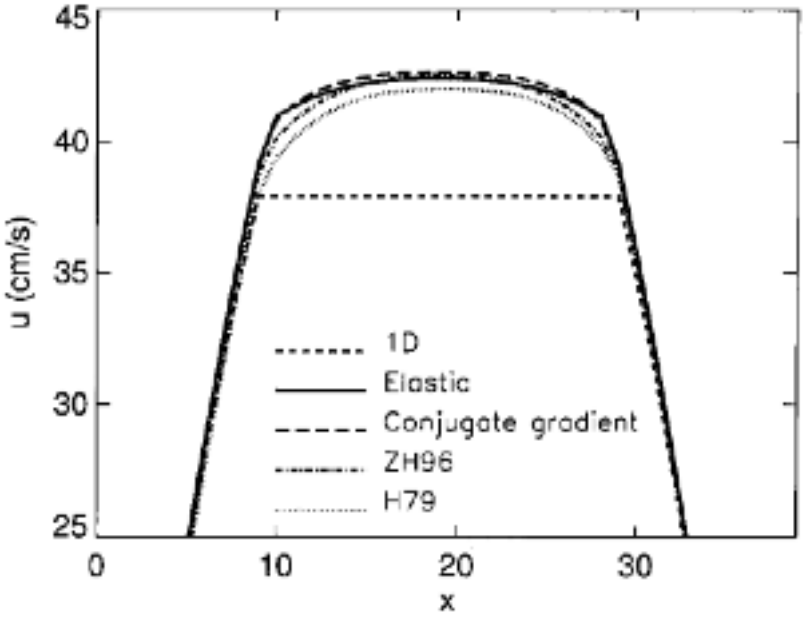
Finally, a validation testcase is implemented, showing the interaction between the sea ice and ocean/wind currents in regions with different sea ice concentrations. In this model, we demonstrate that the sea ice exhibits free–drift behaviour in regions of low sea ice concentration, while the internal stresses resist free–drift behaviour in regions of higher sea ice concentration.

The extent of testing the coupled model is only as described in this thesis, although the development is complete. It is necessary that more tests are put in place to investigate other aspects of the coupled model before Fluidity–FESIM can be used for investigations of ocean – sea ice dynamics, or forecast studies. In particular, the implemented sea ice thermodynamics need to be tested, as well as the fluxes of heat and salt between the sea ice and ocean components. Despite the need for further testing, it has been demonstrated

here that the foundations have been laid towards building a fully–capable, coupled ocean – sea ice model.

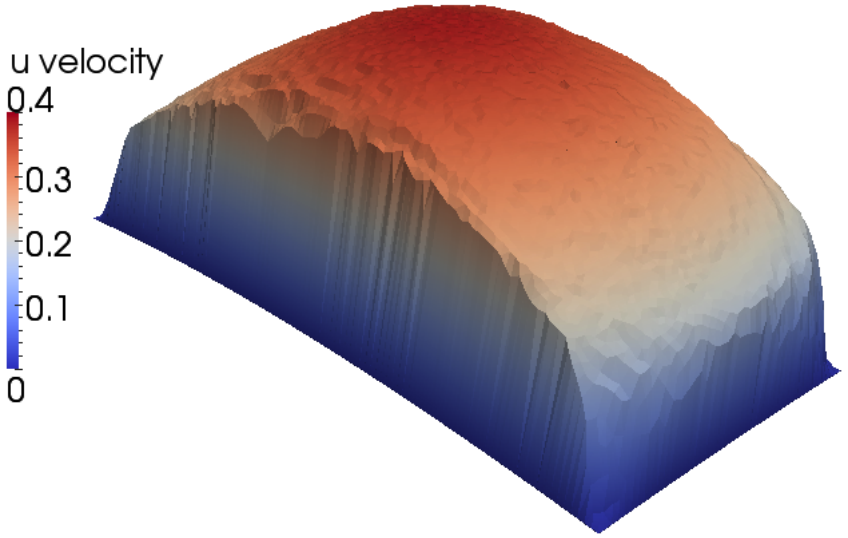


(A) Analytical expression (green line) computed from (2.54) and simulated sea ice velocity (blue line) in the x–direction, simulated using Fluidity–FESIM. The sea ice velocity plotted is along the centreline of the domain.

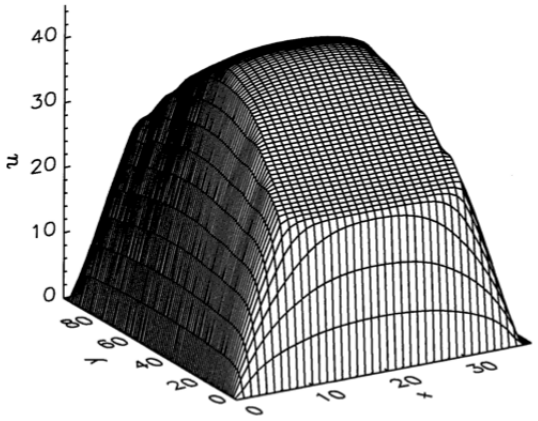


(B) This is figure 4 of Hunke and Dukowicz [1997], comparing the various rheological models.

FIGURE 2.15: Results of test problem described in section 2.4.2

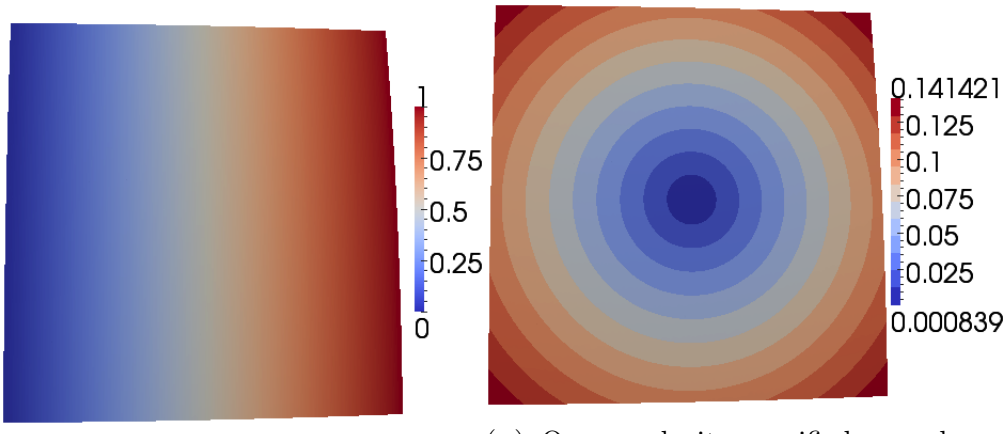


(A) Visualisation of steady–state solution for the pseudo 1–D problem described in section 2.4.2. The colours indicate the magnitude of the u–component of velocity. For ease of visualisation, and for better comparison with 2.16b, the domain has been warped according to the magnitude of the sea ice velocity component



(B) Figure 5 of Hunke and Dukowicz [1997]

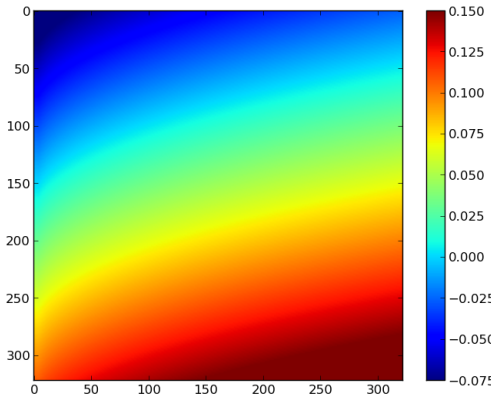
FIGURE 2.16: x–component of sea ice velocity at steady–state.



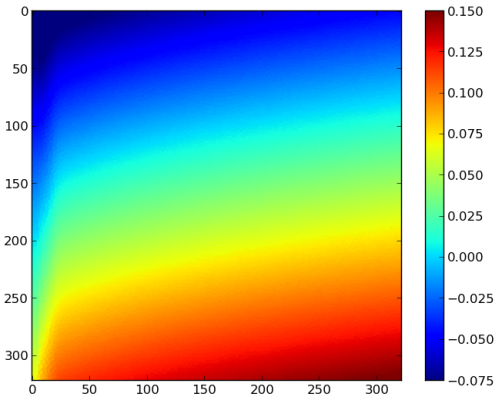
(A) Specified sea ice concentration

(B) Ocean velocity specified according to (2.56).

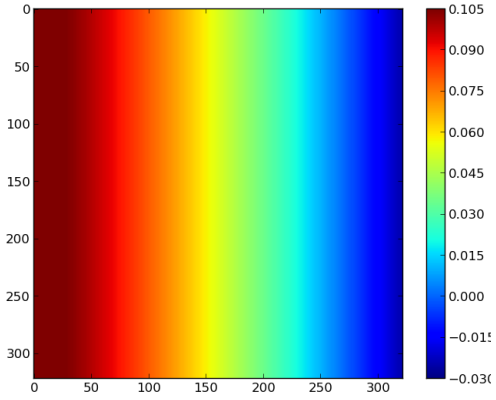
FIGURE 2.17: Figures illustrating the setup of test problem described in section 2.4.3



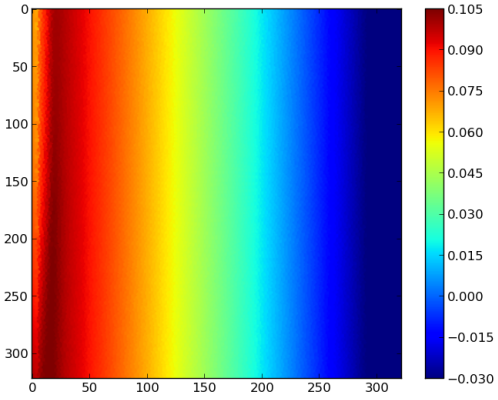
(A) Analytical x–component.



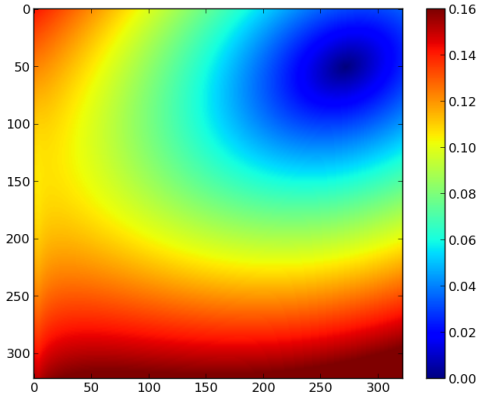
(B) Simulated x–component.



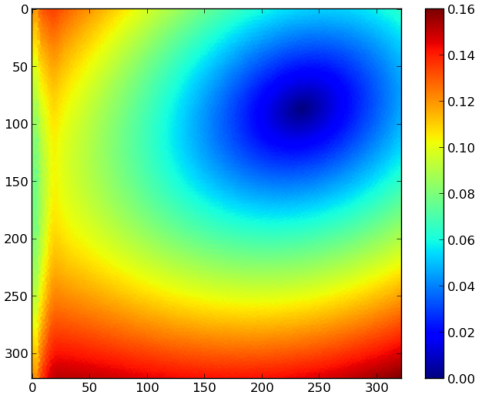
(C) Analytical y–component.



(D) Simulated y–component.



(E) Magnitude of analytical sea ice velocity.



(F) Magnitude of simulated sea ice velocity.

FIGURE 2.18: Results of test problem described in section 2.4.3. The left hand side presents the analytical results of the steady–state, free–drift sea ice velocity computed using (2.66). On the right hand side, the steady–state sea ice velocity simulated using Fluidity–FESIM is presented.

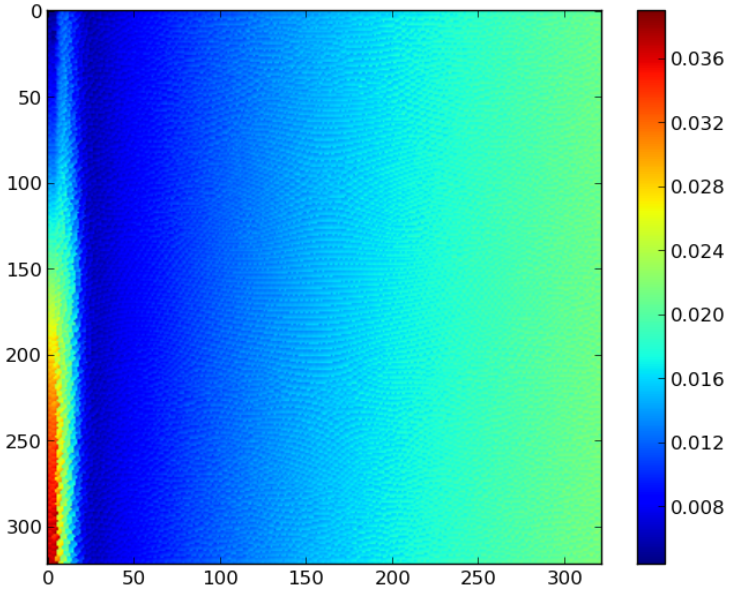


FIGURE 2.19: Difference in the x–component of sea ice velocity between the simulated and analytical solutions of problem 2.4.3, measured using $|u_{\text{simulated}} - u_{\text{analytical}}|$.

CHAPTER 3

ICE SHEET MODELLING

3.1 Introduction

Ice sheets are masses of glacial ice, defined as covering an area greater than 50,000 square kilometres. The two ice sheets currently on the Earth are the Greenland and Antarctic ice sheets, which hold a volume of water equivalent to more than 65 metres of sea level rise. Considering the magnitude of their possible contributions to sea level rise in a changing climate, and the recently observed acceleration of mass loss rate, a thorough understanding of ice sheet processes and feedback mechanisms is required.

The three–dimensional, non–linear, non–Newtonian Stokes (Stokes) equations have been shown [Cuffey and Paterson, 2010] to provide the most accurate and complete description of momentum balance for modelling the flow of land ice.

Due to the large computational cost of solving the full Stokes equations in large, three–dimensional domains, several approximations have previously

been employed. These approximations primarily seek to exploit the large aspect ratio of the domain. One such approximation is the shallow ice approximation (SIA). More recently, depth-integrated approximations of higher–order have been used, such as the Blatter–Pattyn model [Blatter, 1995, Pattyn, 2003] or the so–called L1L1 and L1L2 models [Hindmarsh, 2004]. These models neglect the longitudinal deviatoric stress components from the stress matrix and thus their validity breaks down in regions where this component is important, such as regions of steep topography [Meur et al., 2004]. More recently, an unstructured mesh, full Stokes, finite element model has been used to model the Greenland ice-sheet for a period of a century [Seddik et al., 2012].

The finite element method lends itself naturally to the use of unstructured meshes which can more accurately represent bottom topography, as well as take advantage of recent advances in spatial resolution of ice sheet boundaries, as compared to structured meshes.

This thesis presents a new computational framework for carrying out ice sheet simulations. We use Fluidity, a state–of–the–art finite element model that offers important potential benefits for the modelling of ice sheets:

- It uses unstructured meshes, which allows us to represent the complicated bedrock topography of ice sheets more accurately than structured models, as well as focus resolution, and hence computational resources, in regions where the dynamics of the ice sheet are more complicated.
- It offers dynamically adaptive, anisotropic mesh refinement, allowing the model to focus resolution in regions of dynamic importance and hence optimise the use of computational resources, without the user's a priori knowledge of where these regions may be.

We concentrate in this chapter on verifying and validating the isothermal dynamics of the model. The model is first verified by the method of manufactured solutions [Roache, 2001]. The anisotropic adaptive meshing technology

is then demonstrated in a test of the formation of an idealised ice stream. Finally, we validate the model by carrying out the experiments laid out in the Ice Sheet Model Intercomparison Project for Higher–Order ice sheet Models [Pattyn et al., 2008] (ISMIP–HOM). The performance of this model is analysed and discussed.

Following this, in the next chapter preliminary results of isothermal simulations of Greenland are presented, which attempt to make use of mesh adaptivity to minimise the error due to spatial discretisation and to optimise use of computational resources.

3.2 Equations and numerical methods

3.2.1 Governing equations

Ice sheets may be modelled using the non–linear, non–Newtonian, incompressible Stokes equations:

$$\nabla \cdot \sigma + \rho \mathbf{g} = 0, \tag{3.1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{3.2}$$

where $\mathbf{u}$ is the 3–D velocity vector of the ice sheet, σ is the stress tensor, ρ is the ice sheet density, and $\mathbf{g}$ is the acceleration due to gravity.

3.2.2 Constitutive law

We begin by decomposing the full stress tensor σ in terms of the deviatoric stress τ and the isotropic pressure p:

$$\sigma = \tau - p\mathbf{I}, \tag{3.3}$$

where $p = -\frac{1}{3}\text{tr}(\sigma)$ and $\text{tr}(\sigma)$ is the sum of the elements on the main diagonal of σ. $\mathbf{I}$ is the identity tensor.

Glen's flow law relates the deviatoric stress τ [Cuffey and Paterson, 2010], to the strain rate $\dot{\epsilon}_{\mathbf{u}}$. This states that:

$$\tau = 2\eta_{\mathbf{u}}\dot{\epsilon}_{\mathbf{u}}, \tag{3.4}$$

where the strain rate tensor is given by:

$$(\dot{\epsilon}_{\mathbf{u}})_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \tag{3.5}$$

and the effective viscosity $\eta_{\mathbf{u}}$ is defined as:

$$\eta_{\mathbf{u}} = \frac{1}{2}A^{-1/n}\dot{\epsilon}_e^{(1-n)/n}, \tag{3.6}$$

where n is the empirically determined Glen's flow law exponent that varies with temperature, crystal orientation and debris content, but is typically prescribed as a constant $n = 3$. $\dot{\epsilon}_e$ is the effective strain rate given by the second invariant of the strain rate:

$$\dot{\epsilon}_e = \sqrt{\frac{1}{2}\dot{\epsilon}_{\mathbf{u}} : \dot{\epsilon}_{\mathbf{u}}} \tag{3.7}$$

$$= \sqrt{\frac{1}{2}\left((\dot{\epsilon}_{\mathbf{u}})_{11}^2 + (\dot{\epsilon}_{\mathbf{u}})_{22}^2 + (\dot{\epsilon}_{\mathbf{u}})_{33}^2 + 2\,(\dot{\epsilon}_{\mathbf{u}})_{12}^2 + 2\,(\dot{\epsilon}_{\mathbf{u}})_{23}^2 + 2\,(\dot{\epsilon}_{\mathbf{u}})_{31}^2\right)}, \tag{3.8}$$

and A, the deformation rate factor, is strongly dependent on temperature and is given by an Arrhenius relation:

$$A = A_0 e^{\left(\frac{-Q}{RT}\right)}, \tag{3.9}$$

where A_0 is termed the pre–exponential factor, Q an activation energy, R is the gas constant and T the temperature relative to the melting point of ice at depth. In the simulations in this chapter, a constant deformation rate factor, A, is used, as all simulations are considered for simplicity to have a spatially and temporally invariant temperature. The full Arrhenius expression is used for the simulations of Greenland in the following chapter.

3.2.3 Boundary conditions

3.2.3.1 Surface boundary

At the top surface (Γ_s) of the domain, the normal stress component is balanced by atmospheric pressure:

$$\sigma \cdot \mathbf{n} = -p_{\text{atm}} \cdot \mathbf{n} \qquad \text{on } \Gamma_s. \tag{3.10}$$

Since atmospheric pressure is very small compared to the stresses of the ice mass, we make the common assumption $p_{\text{atm}} = 0$, thus applying:

$$\sigma \cdot \mathbf{n} = 0. \tag{3.11}$$

This is commonly referred to as the *natural* boundary condition.

3.2.3.2 Free Surface

In the cases where the response of the surface elevation is of interest, a prognostic free-surface algorithm can be used, as described in Kramer et al. [2012]. This boundary condition is not used in the work in this thesis, but would form an important aspect of potential future work.

3.2.3.3 Bottom boundary

Two types of boundary conditions may be applied to the ice–bedrock interface. The no–slip boundary condition is applied on the fixed portion of the boundary, $\Gamma_{\text{b,fix}}$,

$$\mathbf{u} = \mathbf{0} \quad \text{on } \Gamma_{\text{b,fix}}, \tag{3.12}$$

or the linear Rayleigh friction law prescribed on the portion of the boundary where the ice can slide, $\Gamma_{\text{b,sld}}$

$$\begin{aligned} \mathbf{u} \cdot \mathbf{n} &= 0 \quad \text{and} \\ \mathbf{n} \cdot \sigma \cdot \mathbf{t} &= -\beta^2 \mathbf{u} \cdot \mathbf{t} \quad \text{on } \Gamma_{\text{b,sld}} \end{aligned} \tag{3.13}$$

where β^2 is the friction or sliding coefficient and $\mathbf{t}$ denotes a tangential direction.

3.2.3.4 Lateral boundaries

The lateral boundaries can be prescribed using the *natural* boundary condition (3.11) or the no–slip boundary condition (3.12). In cases where periodicity is required, as for example in the exercises of the ice sheet model intercomparison study, periodic boundary conditions can also be used. As will be seen later, care needs to be taken when periodic boundary conditions are prescribed.

The no–slip condition applied to the lateral boundary physically represents ice that is fastened to the side wall, for example a glacier that is constrained laterally by mountains. On the other hand, the natural boundary condition can physically be understood as a *calving front*, where any ice that flows out from the boundary is assumed to calve off or melt. The periodic boundary conditions are unnatural, but depending on usage, can emulate a domain that can be extended to infinity in the direction of periodicity, where the domain simulated forms the 'unit cell' which can be tiled in either direction.

3.2.4 Finite element formulation

Considering the momentum, (3.1) and continuity, (3.2) equations, now in Einstein notation:

$$\partial_i \left[\eta_{\mathbf{u}} \left(\partial_j u_i + \partial_i u_j\right)\right] - \partial_i p + \rho g_i = 0, \tag{3.14}$$

$$\partial_j u_j = 0, \tag{3.15}$$

the Galerkin finite element discretisation begins by taking the weak form of the equations, where we use the vector valued function $\mathbf{w}$, and scalar function m to test the momentum and continuity equations, respectively:

$$\int_\Omega \left[w_i \ \partial_i \left(\eta \left(\partial_j u_i + \partial_i u_j \right) \right) - \partial_i p + \rho g_i \right] \ d\Omega = 0, \tag{3.16}$$

$$\int_\Omega m \ \partial_j u_j \ d\Omega = 0, \tag{3.17}$$

where we have integrated over the domain Ω. We discretise the test functions:

$$\int_\Omega N_i \ \partial_i \left(\eta \left(\partial_j u_i + \partial_i u_j \right) \right) \ d\Omega - \int_\Omega N_i \ \partial_i p \ d\Omega + \int_\Omega N_i \ \rho g_i \ d\Omega = 0, \tag{3.18}$$

$$\int_\Omega M \ \partial_j u_j \ d\Omega = 0. \tag{3.19}$$

The viscosity term in (3.18) can be integrated by parts:

$$\begin{aligned} \int_\Omega N_i \ \partial_i \left(\eta \left(\partial_j u_i + \partial_i u_j \right) \right) \ d\Omega = & - \int_\Omega \partial_i N_i \eta \left(\partial_j u_i + \partial_i u_j \right) \ d\Omega + \\ & + \int_\Gamma N_i \eta \left(\partial_j u_i + \partial_i u_j \right) n_j \ d\Gamma, \end{aligned} \tag{3.20}$$

where n represents the outward normal. The continuity equation can also be integrated by parts:

$$\int_\Omega M \ \partial_j u_j \ d\Omega = - \int_\Omega \left(\partial_j M \right) u_j \ d\Omega + \int_\Gamma M u_j n_j \ d\Gamma, \tag{3.21}$$

Substituting (3.20) into (3.18) now yields:

$$\int_\Omega \partial_i N_i \eta \left(\partial_j u_i + \partial_i u_j\right)\ d\Omega + \int_\Omega N_i\ \partial_i p\ d\Omega = \int_\Gamma N_i \eta \left(\partial_j u_i + \partial_i u_j\right)\ d\Gamma + \\ + \int_\Omega N_i\ \rho g_i\ d\Omega, \tag{3.22}$$

$$\int_\Omega \left(\partial_j M\right) u_j\ d\Omega = 0, \tag{3.23}$$

where we have neglected the boundary integral of the continuity equation:

$$\int_\Gamma M u_j n_j\ d\Gamma. \tag{3.24}$$

For no–slip boundary conditions we set both normal and tangential velocity components to zero, thus the term (3.24) disappears. Similarly, if a drag boundary condition is applied, this is coupled with a no–normal flow condition and thus $u_j n_j = 0$, thus (3.24) again disappears.

The above system of equations can be written in matrix form:

$$\begin{pmatrix} K & G \\ G^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} \tag{3.25}$$

where:

$$K_{b_i c_j} = \int_\Omega \left(\partial_j N_b\right) \eta \left(\partial_i N_c\right) + \left(\partial_k N_b\right) \eta \left(\partial_k N_c\right) \delta_{ij} \tag{3.26}$$

$$G_{ab_j} = \int_\Omega N_a \partial_j N_b \tag{3.27}$$

$$f = \int_\Omega N_i \rho g_i \quad + \text{potential boundary forcing terms} \tag{3.28}$$

3.2.5 Solving the discretised system

The system (3.25) is indefinite (i.e. has both positive and negative eigenvalues) and thus requires specialised solution algorithms (for an overview of various such methods, see May and Moresi [2008]). The method used in Fluidity for solving the discretised system is based upon a full–projection / pressure–correction approach [AMCG, 2014, Davies et al., 2011], which is equivalent to the Schur Complement Reduction method described in the study by May and Moresi [2008].

We note that the discretised system of equations (3.25) is only defined implicitly, given that the viscosity $\eta = \eta(\mathbf{u})$ is a function of velocity. At each moment, the viscosity is constructed using the current best estimate of velocity:

$$\begin{pmatrix} K(\mathbf{u}^{n-1}) & G \\ G^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}^n \\ \mathbf{p}^n \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} \tag{3.29}$$

This creates a fixed–point iteration problem, which is iterated to convergence. For steady–state problems in Fluidity, the simulation can be pseudo–timestepped to convergence. For time–varying problems (such as the case with a prognostic free–surface or temperature evolution), inner non–linear iterations may be used, to solve the system before moving forward in time.

3.3 Verification and performance

3.3.1 Method of Manufactured Solutions

Model verification is the process of error evaluation based on a known solution. This is a mathematical analysis that does not require consideration of

ice sheet physical laws. Model verification is a test of the implementation, in contrast to model validation which validates the model's ability to simulate reality. In this thesis, we present model verification conducted via the method of manufactured solutions [Roache, 2001] (MMS). Using the MMS, an analytical solution to a slightly altered problem is constructed and used to calculate a simulation error. The manufactured solution is an analytical solution to the full–Stokes equations modified by a source term. A convergence analysis was performed, indicating the rate at which the error decreases with increasing resolution. A convergence analysis is an excellent indicator of model *correctness*, that is, the model is able to correctly solve the system described by the equations. This is in contrast to *accuracy*, which depends on numerous other factors, including domain discretisation convergence and how well the equations model the real system.

Symbol	**Description**	**Value**	**Units**
A	Deformation rate factor	$3.16887646154 \times 10^{-4}$	$Pa^{-n}\ s^{-1}$
n	Glen's flow exponent	3	none
ρ	Ice sheet density	910	$kg\ m^{-3}$
g	Gravity acceleration	0.0	$m\ s^{-2}$

TABLE 3.1: Constants used for the MMS convergence test.

The simulation parameters used in this verification exercise are detailed in table 3.1. Note that, for simplicity, non–realistic deformation rate factor and gravity values are used for this experiment, but with no loss of appropriateness of the test.

We choose a manufactured solution $(\mathbf{u}, p) = (\mathbf{U_m}, P_m)$ where:

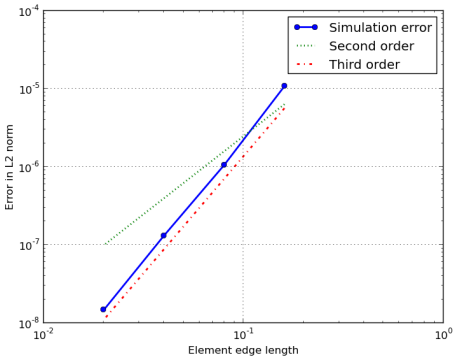
$$\mathbf{U_m} = (\sin(x)\cos(y), -\cos(x)\sin(y))\,, \tag{3.30}$$

$$p = \cos(x)\cos(y). \tag{3.31}$$

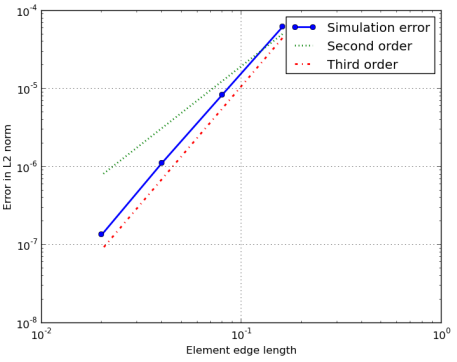
Note that although the chosen solution, $\mathbf{U_m}$, need not take ice-sheet physics into account, we ensure here that it satisfies $\nabla \cdot \mathbf{U_m} = 0$, i.e. is divergence free. This is to avoid the need for adding a source term to the continuity equation.

The domain was a square of edge–length $1m$ with $0.0 \leq x \leq 1.0$ and $0.0 \leq y \leq 1.0$. The analysis was conducted on meshes with characteristic element edge lengths varying from $0.16m$ down to $0.02m$. The manufactured solution was applied as a Dirichlet boundary condition for velocity on all sides of the domain.

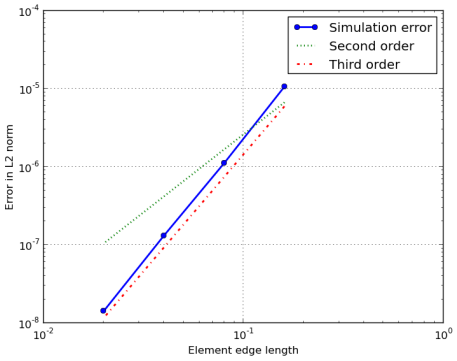
Two separate tests were performed. A *linearised* case where the viscosity is prescribed ($\eta = 1$) and a *non–linear* case where the viscosity is diagnosed as a function of the second invariant of the strain rate of velocity, according to Glen's flow law, as in (3.6). The results of the convergence analysis are shown in Figure 3.1. The finite element pair used in these simulations is $\mathcal{P}2\mathcal{P}1$ (i.e. we use a piecewise continuous quadratic representation for velocity and a piecewise continuous linear representation for pressure) and therefore we expect 3^{rd}–order convergence for velocity and 2^{nd}–order for pressure. The results presented in figure 3.1 indicate model correctness and the expected orders of convergence.



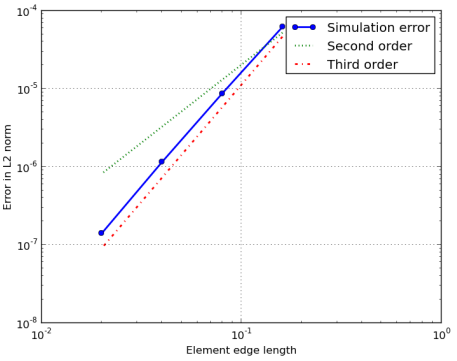
(A) Linearised case: Velocity x-direction



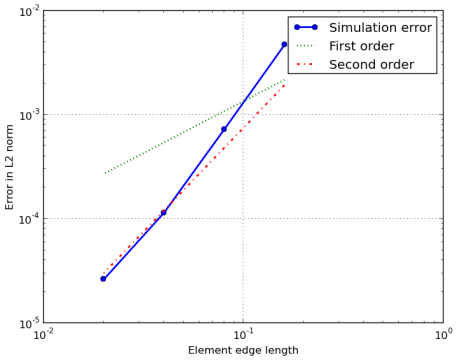
(B) Non linear case: Velocity x-direction



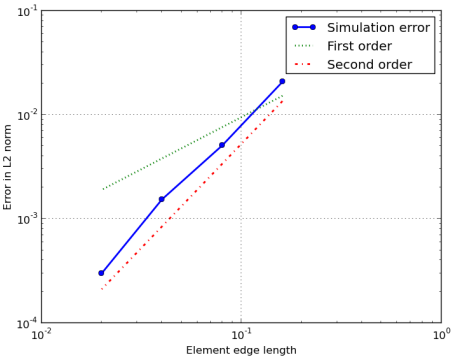
(C) Linearised case: Velocity y-direction



(D) Non linear case: Velocity y-direction



(E) Linearised case: Pressure



(F) Non linear case: Pressure

FIGURE 3.1: Convergence plots for linearised (left column) and non–linear (right column) Stokes, for x–direction (top row), y–direction (second row) velocity components and pressure(bottom row).

3.4 Adaptive modelling of ice streams

Finite element modelling of the full Stokes system is, numerically, very costly. Unstructured meshes have recently been used to mitigate these costs by focusing resolution only in regions where the dynamics require it. However, generating such unstructured meshes requires a priori knowledge of where the regions of important dynamics are located. Dynamically adaptive, unstructured meshes offer the potential of optimally adjusting the domain discretisation in response to evolving solution fields, such that higher resolution is placed in regions where it is required, without the user's a priori knowledge of where these regions are located.

This test uses dynamic adaptivity to model an idealised ice stream. In particular, emphasis is placed on showing that the simulations converge with spatial resolution and that the adaptive simulations are superior to their fixed, structured mesh counterparts.

Consider an ice sheet of 1 km thickness and size 160 $km \times 160$ km in a domain that is periodic in the x–direction. The ice sheet is placed on a bed inclined in the x–direction, with the bedrock elevation, b, given by:

$$b = -x \sin(\theta), \tag{3.32}$$

where $\theta = 0.5°$. A linear bottom friction is applied, where the friction coefficient, C, is given by:

$$C(x, y) = C_0 \left\{ 1 + \epsilon + \sin \left[\frac{2\pi y}{R} + m \sin \left(\frac{2\pi x}{R} \right) \right] \right\}, \tag{3.33}$$

with parameters $C_0 = 1.0 \times 10^5$ Pa m^{-1} a, $\epsilon = 5.0 \times 10^{-5}$, $R = 160$ km and $m = 0.25$. These parameters are in line with Cornford et al. [2013], who also

demonstrated their structured adaptive mesh refinement technology on this testcase.

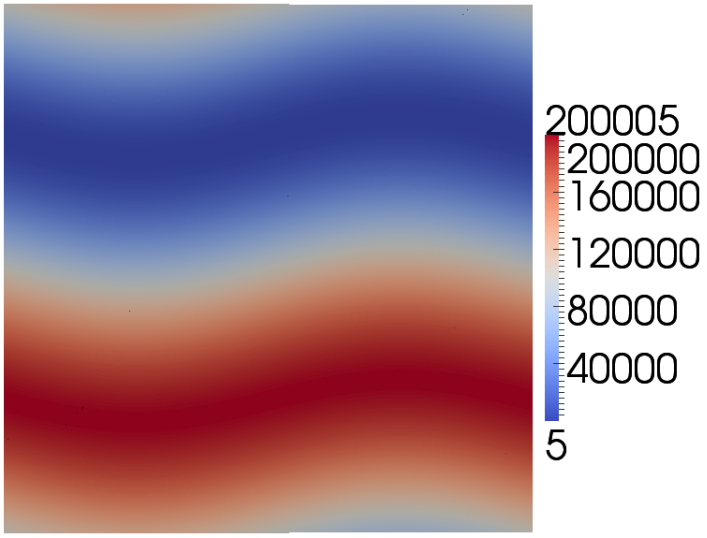


FIGURE 3.2: Bedrock friction coefficient for idealised ice stream setup, given by (3.33).

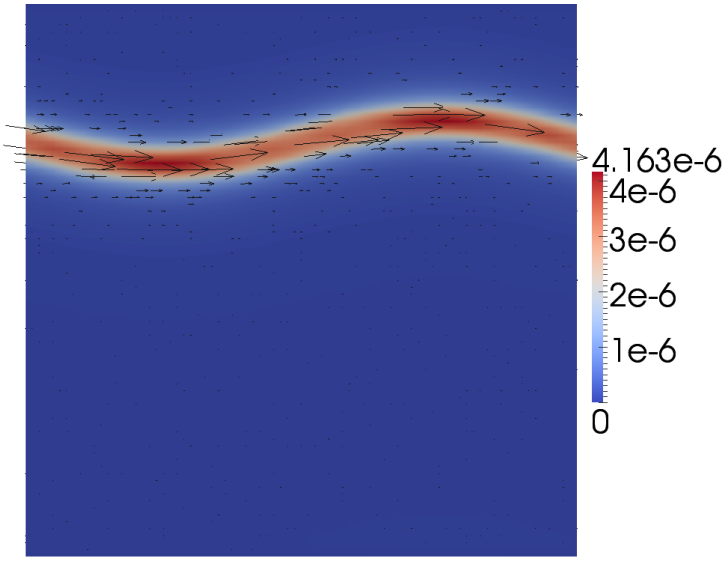
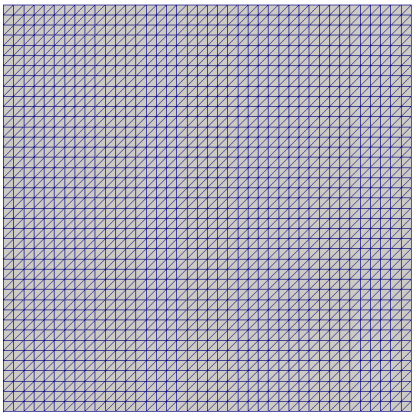


FIGURE 3.3: Result of simulation described in section 3.4. The result shown is from the fixed mesh of 4 km resolution. The colourmap indicates the magnitude of velocity vectors in ms^{-1} and the arrows the velocity direction. Arrows are proportional in size to the magnitude of velocity. Small arrows away from the ice stream indicate the whole ice sheet is drifting along the domain. A sinusoidal region of higher velocity ice flow indicates the formation of an ice stream.

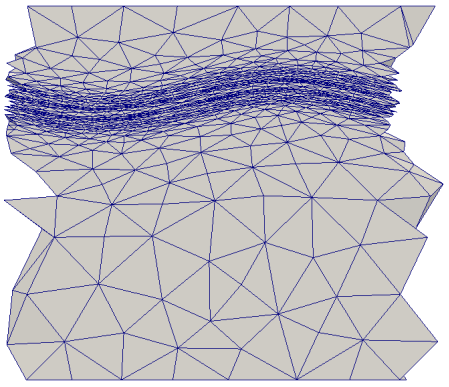
Equation (3.33) produces the bottom friction illustrated in figure 3.2. It produces a sinusoidal region of lower friction closer to the top of the domain, in

which the ice sheet is allowed to accelerate and form an idealised ice stream. The simulation was performed on both fixed and adaptive meshes of varying resolutions. An example result is shown in figure 3.3. At steady–state, the full ice stream drifts along the x–direction of the domain, whilst a higher velocity sinusoidal stream forms in the region of lower bottom friction coefficient.

The problem was run on both fixed and dynamically adaptive meshes. The fixed, tetrahedral meshes were structured, with element edge lengths varying from 16 km (resulting in 600 elements in the domain) down to 0.25 km (approximately 2.5 million elements) (figure 3.4a), with a mesh *refinement ratio* of 2, i.e. the element edge lengths were halved for each structured grid generated. The vertical resolution remained constant at a single $1km$ layer, for all simulations. The element edge lengths controlled were the short–edges of the triangular face aligned in the horizontal. For the adaptive meshes, the number of nodes was controlled using the interpolation error bound parameter, as introduced in section 1.2.3, with minimum edge length in the vertical direction set to $1km$, to maintain a single $1km$ layer of elements in the vertical.



(A) Fixed mesh with 4 km edge length. Total number of nodes: 96,000



(B) Adaptive mesh. Total number of nodes: 38,030

FIGURE 3.4: Meshes for idealised ice stream test problem. Notice that far smaller (and larger) element sizes are used in (B) while overall the number of nodes is far less than (A).

The order of convergence is best estimated by comparing to the exact solution, as shown in the previous test using the method of manufactured solutions. However, in the case of the idealised ice stream, an exact solution is not available. Therefore, following Roache [1998], we estimate the order of convergence by considering the error, E:

$$E = f(h) - f_{\text{exact}} = Ch^p + \text{higher–order terms}, \tag{3.34}$$

where $f(h)$ and f_{exact} are approximate and exact solutions to the problem. h is a measure of node distance such as the characteristic element edge length, C a constant and p the order of convergence. A model which exhibits second–order convergence would therefore have $p = 2$.

If we neglect higher–order terms and take the logarithm of (3.34) we get:

$$\log(E) = \log(C) + p\log(h). \tag{3.35}$$

We note that an approximation for the order of convergence, p, can be obtained from the slope of the curve $\log(E)$ versus $\log(h)$. As stated above we do not know f_{exact} and hence also E, for this problem. However, given three approximations to the solution on three consecutive meshes with constant refinement ratio, we can estimate the order of convergence using:

$$p = \frac{\ln\left(\frac{f_3 - f_2}{f_2 - f_1}\right)}{\ln(r)}, \tag{3.36}$$

where f_1, f_2 and f_3 are a functional representing the result of each simulation performed on consecutively finer resolution, and r represents the refinement

ratio between the meshes. For each simulation, the function f was evaluated as the l^2–norm of the x–component of velocity, u:

$$f = |u| = \sqrt{\sum_{k=1}^{n} x_k^2}, \tag{3.37}$$

where k indexes nodes of a uniform, regular grid on which the solution of all fixed mesh simulations is interpolated on. This results in the functional f being a consistent measure of the root–mean–square error of each simulation.

The theoretical order of convergence of velocity in the full–Stokes model, using the $\mathcal{P}2\mathcal{P}1$ element pair, is third–order, as seen in section 3.3.1. In practice, however, boundary conditions, computer round–off errors and iterative convergence errors result in a value that is often smaller. Using the method described above (equation 3.36) and calculating f for three fixed resolution meshes, we calculate $p = 1.7$ as the convergence order, indicating model convergence at a good order.

For the adaptive simulations the mesh adapts every 20 iterations, and the simulation is run to steady–state, checked by computing the relative change of the velocity field in the l^∞ norm, evaluated as:

$$\frac{|\mathbf{u}^{n+1} - \mathbf{u}^n|_\infty}{\frac{1}{2}\left(|\mathbf{u}^{n+1}|_\infty + |\mathbf{u}^n|_\infty\right)} < \epsilon, \tag{3.38}$$

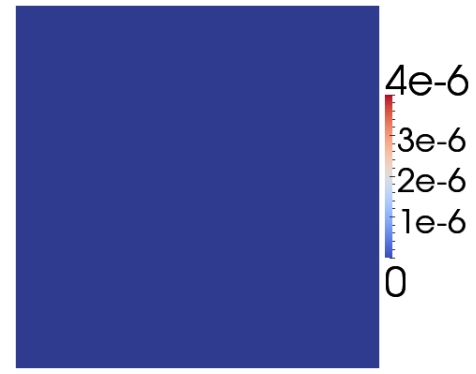
where $\epsilon = 1 \times 10^{-9}$, is a constant $\epsilon \ll 1$.

Figure 3.5 shows the evolution of the solution and mesh during a dynamically adaptive simulation. The initial input mesh is a medium resolution (8 km edge length) structured mesh made up of tetrahedra. The medium resolution mesh was chosen to demonstrate Fluidity's capability of both mesh refinement as

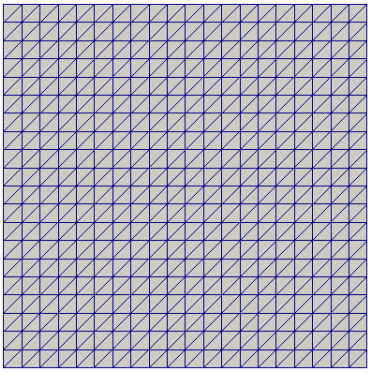
well as coarsening. At the first adapt, Fluidity increases the resolution in the locality of the stream. As the simulation progresses and the stream accelerates, the node density along the stream increases even further. In contrast, outside of the ice stream where the ice sheet is slowly drifting and no particularly interesting dynamics are observed, the resolution is decreased.

To compare the adaptive and fixed mesh results, we plot the velocity on a line within the domain at $x = 80km$, traversing the domain laterally from the points $y = 110km$ to $y = 130km$, at a depth of $12m$. This section was chosen as it crosses the ice stream in the transverse direction and extracts what is considered to be the most significant information from the simulation, the maximum velocity of the ice stream. The choice of depth was arbitrary, and results at different depths or at the surface produce equivalent figures. These results are shown in figure 3.6. The cyan line is the result obtained using the highest resolution fixed mesh, with a $0.25km$ element edge length. The other solid lines (red, yellow and green) are progressively lower resolution and produce a progressively lower peak velocity. At the lowest resolutions, spurious behaviour can also be observed. The dashed lines represent adaptive simulation results. It is important to note that both the adaptive simulations (with approximately 16 and 80 thousand elements) produce results that are better than the $1km$ resolution structured mesh (with $\sim$ 150 thousand elements). The 80,000 element adaptive simulation produces a result nearly as good as the highest resolution fixed mesh which uses 30 times more elements.

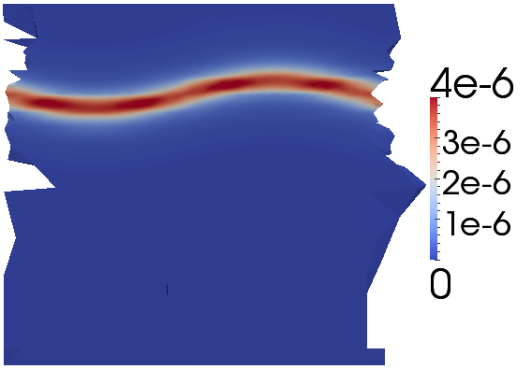
A further interesting feature of the adapted mesh, is the region along the midsection of the ice stream where a lower node density is observed. This can be seen if one looks closely at figures 3.5f and 3.7a. As can be seen in the velocity profile plotted in figure 3.6, a 'flat top' is observed in the midsection of the ice stream. In this region, fewer nodes are required to capture the dynamics and therefore resolution is placed only on either side of this flat section. This is a result of the metric being generated based on the curvature of the solution field at the time of the adapt.



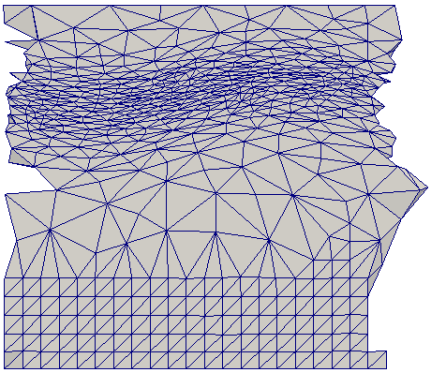
(A) Initial condition of velocity, $\mathbf{u} = 0$.



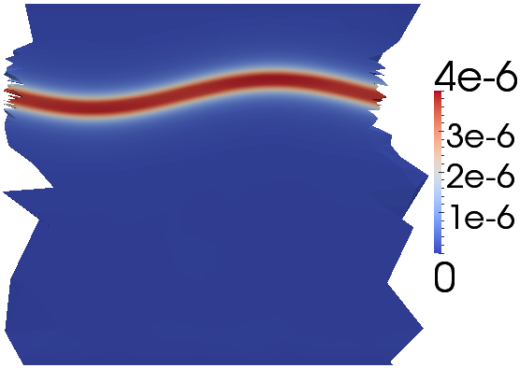
(B) Initial mesh.



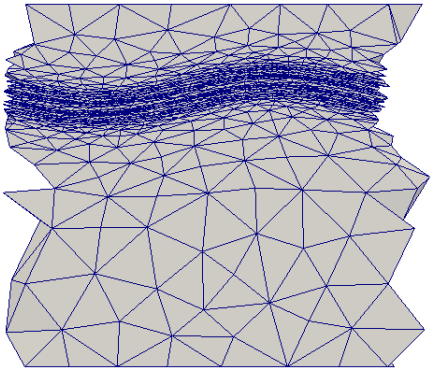
(C) Magnitude of velocity $\mathbf{u}$, $t = 20$.



(D) Mesh after first adapt.



(E) Magnitude of velocity $\mathbf{u}$, $t = 200$



(F) Mesh after tenth adapt.

FIGURE 3.5: Simulation of ice stream problem (section 3.4) using adaptive mesh technology. Each row of images corresponds to a particular iteration level (timesteps 0, 20 and 200, respectively from top to bottom). The images on the left show the solution of the velocity field, whilst the images on the right correspond to the mesh at each iteration level.

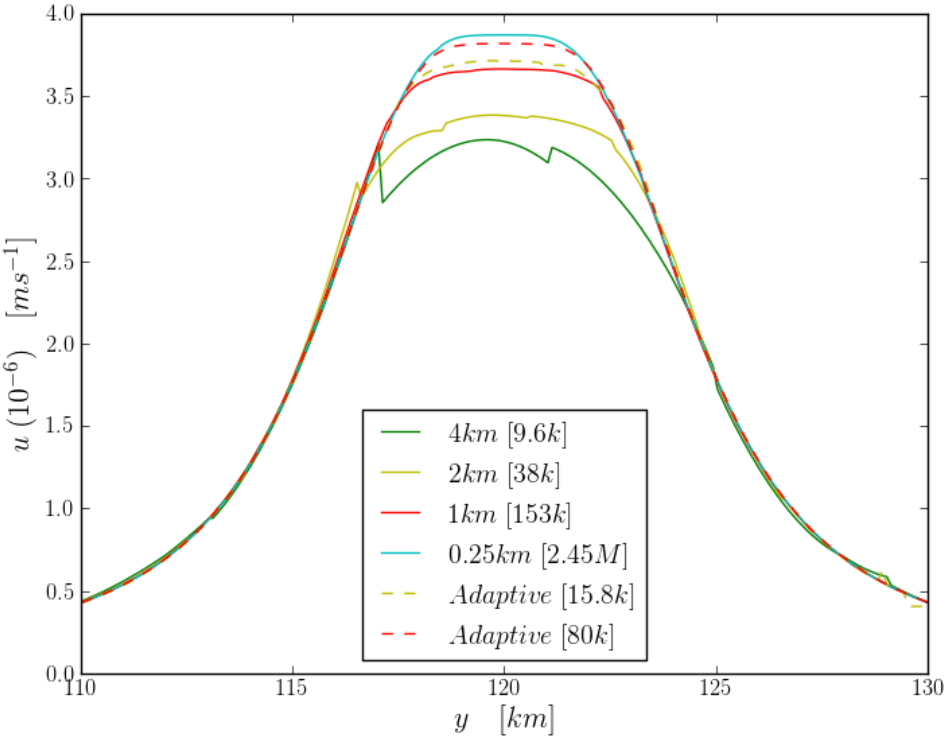


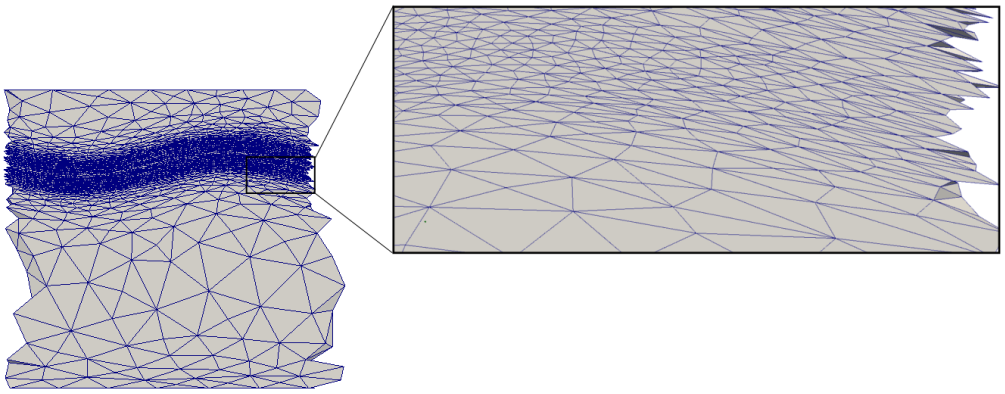
FIGURE 3.6: x–component of velocity for various mesh resolutions, plotted along a line at $x = 80km$, traversing the domain laterally from $y = 110km$ to $y = 130km$, at a depth of $12m$. The legend also specifies the number of elements at the end of each simulation, in square brackets.

Previous studies of dynamically adaptive meshes applied to ice streams [Cornford et al., 2013] were performed using the BISICLES model. The BISICLES model uses adaptive mesh refinement (AMR) technology on structured rectangular grids and solves the vertically–integrated momentum equations of [Schoof and Hindmarsh, 2010]. The AMR technology of BISICLES adapts meshes based on a scalar metric, limiting the adapted meshes to isotropic elements. Fluidity, however, is able to produce anisotropic meshes because the metric produced is in the form of a three dimensional tensor, which has been shown [Piggott et al., 2009] to be even more efficient in the use of computational resources. Fluidity offers an advantage over previous studies, in this regard.

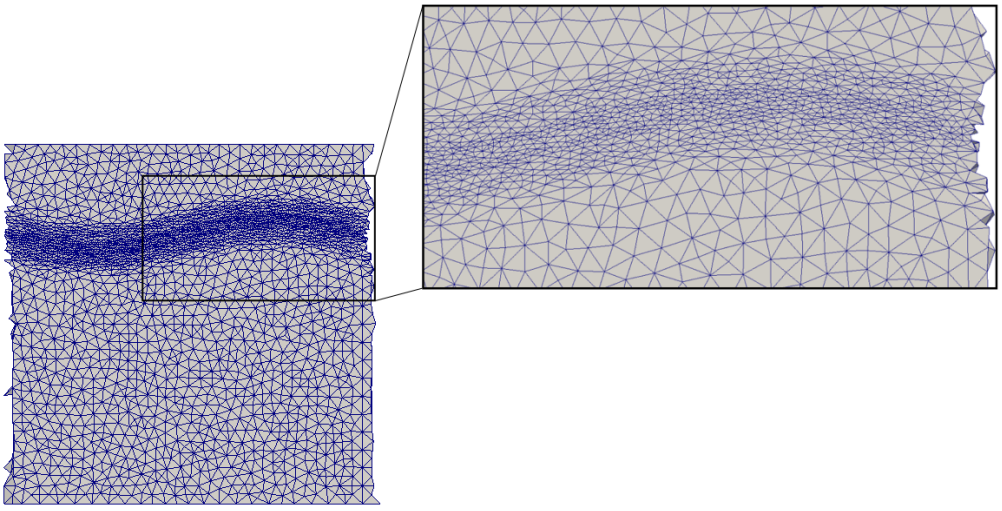
This thesis is the first to present anisotropic, dynamically adaptive, unstructured mesh refinement of the full–Stokes equations for the modelling of ice

sheets.

An example isotropic mesh, obtained using Fluidity, is shown in figure 3.7b. This mesh is obtained by restricting the permissible element aspect ratio. We can see that with isotropic meshes, a larger node density is required for the same effective transverse resolution. It should be noted that the overall number of nodes in the two examples shown in figure 3.7 can not be directly compared, because the isotropy is controlled in all three dimensions with this constraint. Given the high aspect ratio of the domain ($160km$ wide, $1km$ deep), the extent of an element in the horizontal plane is severely restricted by engaging isotropic adaptivity. In a deeper domain, or a two dimensional example, the isotropic example of figure 3.7b would be coarser in the regions away from the ice stream.



(A) Anisotropic.



(B) Isotropic.

FIGURE 3.7: Meshes of adaptive simulations showing the difference between isotropic and anisotropic meshes. The isotropic meshes are obtained by restricting the maximum permissible element aspect ratio to 2.

3.5 Scaling

The simulations in this thesis were performed on a mixture of single and multi–processor configurations. The highly idealised, two–dimensional simulations were usually run on a single core, whilst the more numerically demanding cases were run on multiple nodes on the Imperial College clusters [HPC]. Therefore, reliable and comparable timing data is not readily available for most of the simulations presented in this thesis. However, a limited scaling analysis has been performed to demonstrate a) Fluidity's ability to solve the full–Stokes equations on multiple cores and b) the scaling performance of said model.

The simulation used for the scaling analysis was the three–dimensional ISMIP–HOM experiment A (see section 3.6.1 for more details on the model setup), with aspect ratio 40 and with a resolution of $N_x = N_y = 80$, $N_z = 10$. This resolution was chosen, as it provides a number of nodes that can be used for both the serial and 32–core parallel configuration. The choice of resolution is important because choosing a resolution that is too fine will result in not being able to run the problem in a reasonable amount of time with the smaller core counts, and choosing too coarse a resolution will result in too few nodes per processor being available in the parallelised case, making the communication overhead of parallelisation even greater. The case used has a total of 38,400 tetrahedral elements.

Cores	**Time** $(hh:mm)$	**Speedup**	**Factor**
1	03 : 58	1	-
2	02 : 50	1.39	1.39
4	01 : 31	2.59	1.86
8	00 : 59	3.97	1.5
16	00 : 40	5.85	1.48
32	00 : 21	11.25	1.92

TABLE 3.2: Timing and speedup for scaling analysis described in section 3.5. The *Speedup* column represents the speedup with respect to the serial run, whilst the *Factor* column is the speedup obtained from doubling the number of cores used for each simulation.

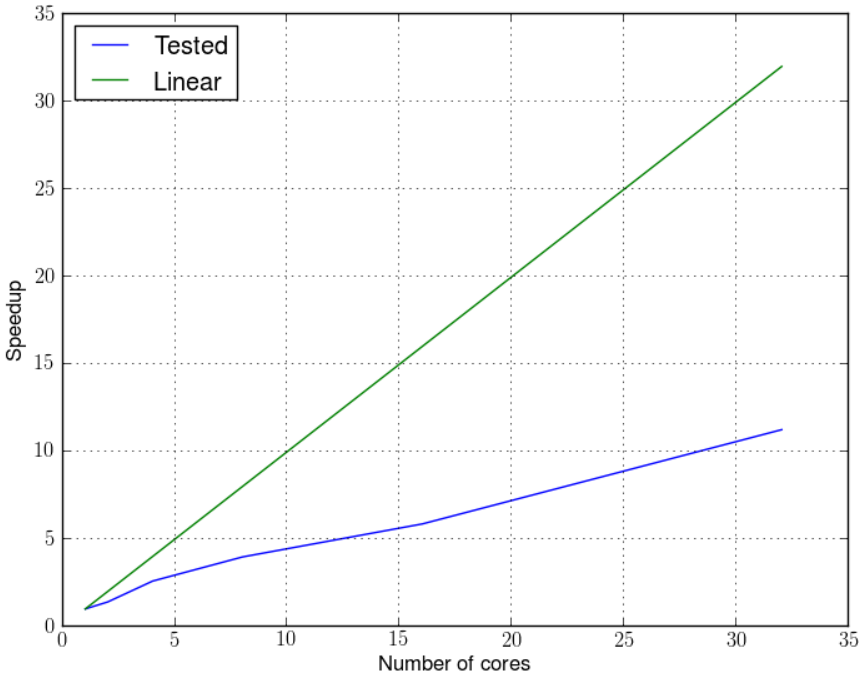


FIGURE 3.8: Scaling plot of Fluidity running simulation described in section 3.5.

The results of the scaling analysis are plotted in figure 3.8 and also tabulated in table 3.2. The table tabulates the *time*, calculated by taking the average time per timestep of ten timesteps for each simulation. The *speedup* is the time of each parallel simulation normalised by the time of the serial simulation and is computed by T_i/T_1, where T_i represents the time of each parallel simulation and T_1 the time of the serial run. The *factor* parameter is the speedup obtained by doubling the number of cores and was calculated using $T_i/T_{i/2}$, where T_i represents the time of each parallel simulation and $T_{i/2}$ represents the time of the parallel simulation run on half the number of cores. This number is significant because it is a measure of the efficiency of doubling the number of cores.

The results of this analysis are only indicative of the Stokes model's scaling performance. For a thorough analysis, several factors need to be considered including a) memory bandwidth, b) processor affinity, c) multi–threading, d) I/O performance and e) other users also on the cluster. Given as there was

no scope for controlling the above factors in this study, the results are not presented as conclusive, but rather as an indication of approximate scaling performance.

3.6 Ice sheet model intercomparison project

The ice sheet model intercomparison project for higher–order models, hereafter simply ISMIP–HOM, represents a suite of test experiments that form a benchmark for model development [Pattyn et al., 2008]. Fluidity is used to carry out these experiments and the results are reported in an attempt to document its performance and suitability for ice sheet modelling.

ISMIP–HOM is formed of six testcases, named *Experiment A* to *Experiment F*. All testcases are isothermal with no effect of temperature on the viscosity. All except one of the problems are solved on simplified, idealised geometries. *Experiment E* is the exception, a model of Haut Glacier d'Arolla, and forms the one and only realistic testcase.

3.6.1 Experiment A – ice flow over a bumpy bed

This experiment simulates a slab of ice over a sloped, bumpy bed. The slope is maximal in the x–direction and zero in the y–direction. The top surface elevation, z_s, is defined as:

$$z_s(x, y) = -x \tan(\alpha), \tag{3.39}$$

where $\alpha = 0.5°$ is the slope angle.

The slab of ice has a mean ice thickness of $1000m$. The topography is a series of sinusoidal bumps with an amplitude of $500m$, resulting in an ice sheet that

varies in thickness from $500m$ at its thinnest to $1500m$ at maximal depth. The bottom topography, z_b, for Experiment A, is defined as:

$$z_b(x, y) = z_s - 1000 + 500 \sin(\omega x) \sin(\omega y). \tag{3.40}$$

The frequency of the basal bumps is $\omega = 2\pi/L$ with $0 \leq x, y \leq L$, where L represents the length of the domain in both the longitudinal and transverse directions. The experiments are run for horizontal domain sizes ranging from $5km$ to $160km$, with $L = 5km, 10km, 20km, 40km, 80km$ and $160km$.

The thickness of the ice slab does not increase with different values of L. The aspect ratio of the domain thus changes from 5 to 160 as the domain length, L, is increased. This allows an investigation on how different models behave in regions of different aspect–ratios, as the aspect–ratio is thought to be a control of ice sheet dynamics. The domain is illustrated in Figure 3.9.

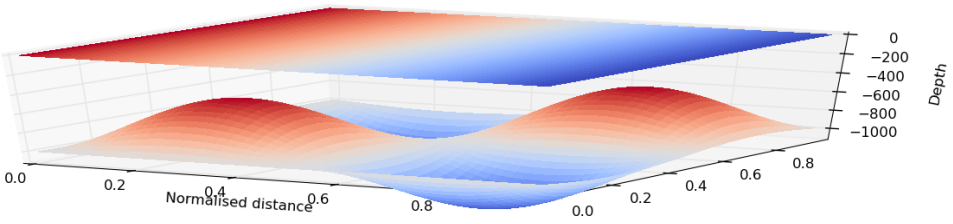


FIGURE 3.9: Illustration of the top and bottom surfaces of the domain for experiment A, with normalised longitudinal and transverse directions. Colouring represents surface depth, indicating the domain is inclined.

Ice is considered frozen at the base, so a no–slip boundary condition, $\mathbf{u}(z_b) = \mathbf{0}$, is prescribed at the bottom. The lateral boundaries are periodic in both the longitudinal (x) and transverse (y) directions. Physically, this is equivalent to simulating an ice sheet that is infinitely long and wide resting on a sloping plane whose topography is repeated (tiled) in the same fashion as the topography of the domain we are simulating. One requirement of periodic boundary conditions is that the solution fields, and therefore the bounding and forcing constraints applied, are continuous along the periodic edges. For Fluidity to

run with periodic boundary conditions, a mapping must exist between corresponding nodes on each periodic face. This imposes two restrictions on the periodic faces. a) The number of nodes on the original and aliased face must be equal and b) there must exist a mapping function for each node. These restrictions need to be considered in the preparation stage of the simulation, because the mesh needs to be designed appropriately. All periodic ISMIP–HOM examples run with Fluidity have been run with faces that share the same mesh (i.e. the mesh of the boundary along the plane $x = L$ is identical to the mesh of the boundary on the plane $x = 0$, and similarly for $y = 0$ and $y = L$).

The periodic mapping for experiment A in the longitudinal direction is given by:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} x - L \\ y + L\tan(\alpha) \\ z \end{pmatrix} \tag{3.41}$$

A similar mapping, that excludes the jump term $\tan(\alpha)$, is used for the periodicity in the transverse direction.

The parameters for experiment A are given in the ISMIP–HOM specifications [Pattyn and Payne, 2006], and are shown in table 3.3. It is noted that for all experiments in this thesis, Fluidity is set up to follow the international system of units (SI), despite Fluidity as a model being unit agnostic and the user having the freedom to choose the system in which to solve the equations. For example, although the deformation rate factor is commonly given in units of $Pa\ a^{-1}$ in glaciology, this is converted to $Pa\ s^{-1}$ as the input to Fluidity. Correspondingly, the timestep is also set in seconds, rather than years.

The results of experiment A are given in two forms:

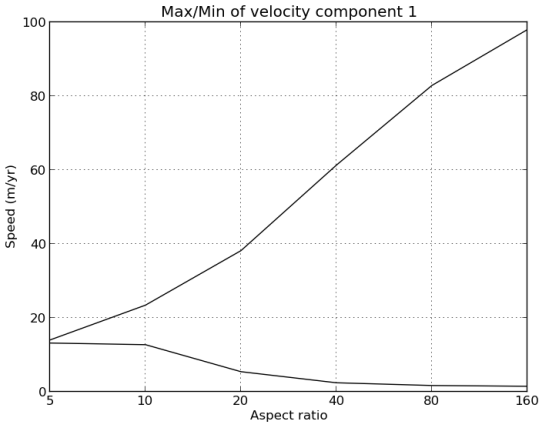
- Maximum and minimum values of surface velocities for domains with different aspect ratios are plotted against the aspect ratio, figure 3.10;

Symbol	Value	Units
A	$3.16887646154 \times 10^{-24}$	$Pa^{-n}\ s^{-1}$
n	3	none
ρ	910	$kg\ m^{-3}$
g	9.81	$m\ s^{-2}$

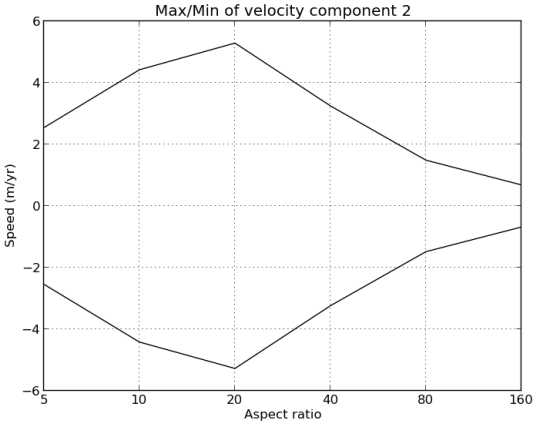
TABLE 3.3: Parameters used for experiment A of ISMIP–HOM.

- Contours of surface velocity are drawn as a function of normalised horizontal displacements x and y, figures 3.11, 3.12 and 3.13.

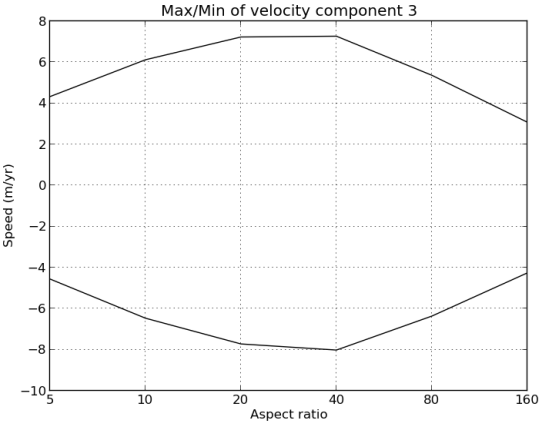
The results are presented here and discussion of these results is deferred to the end of the chapter (section 3.7).



(A) x–component of velocity

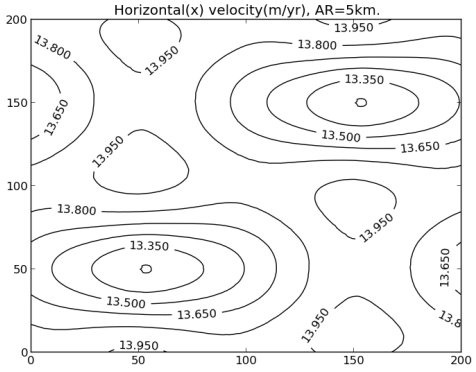


(B) y–component of velocity

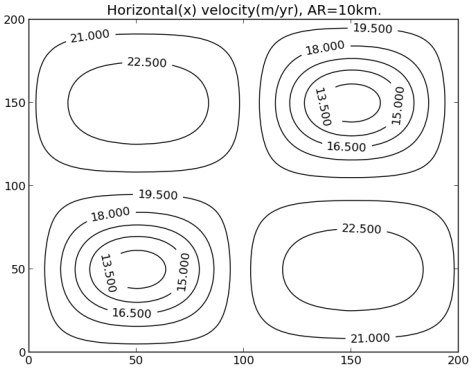


(C) z–component of velocity

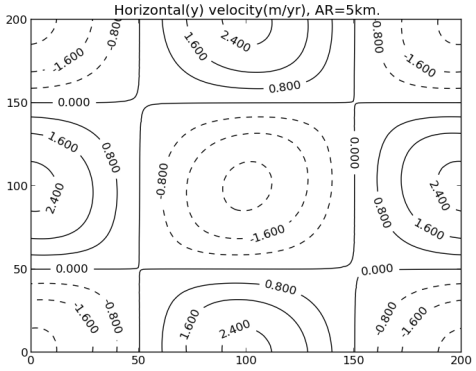
FIGURE 3.10: Maximum and minimum values of surface velocity components as computed with Fluidity, in metres per year, plotted against domain aspect ratio, for ISMIP–HOM experiment A.



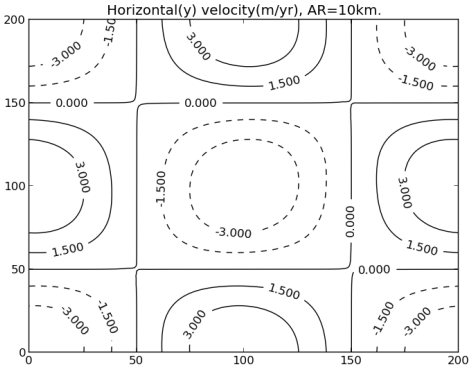
(A) x–component, aspect ratio 5.



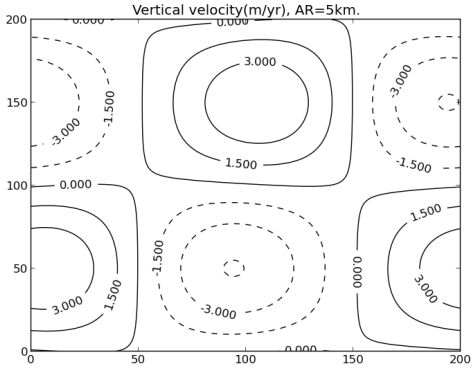
(B) x–component, aspect ratio 10.



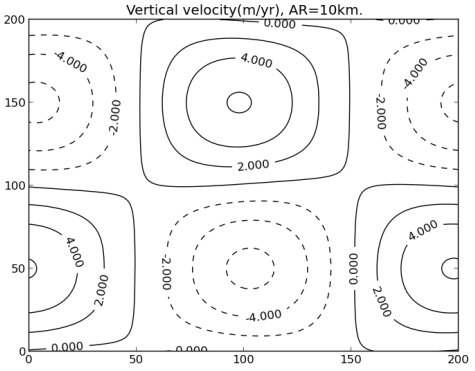
(C) y–component, aspect ratio 5.



(D) y–component, aspect ratio 10.

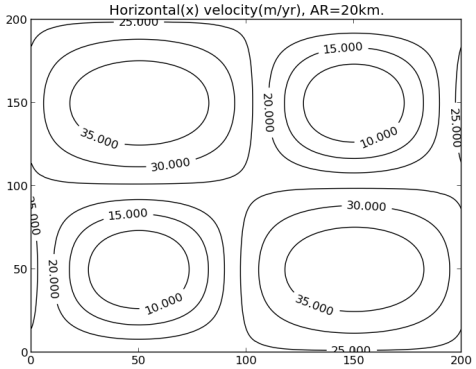


(E) z–component, aspect ratio 5.

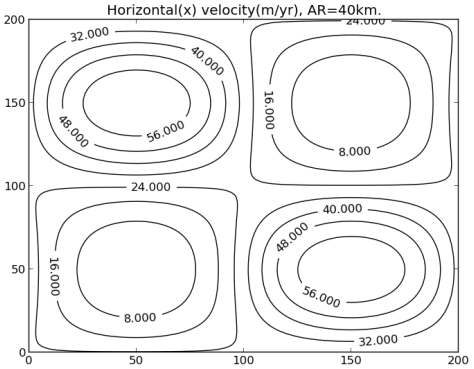


(F) z–component, aspect ratio 10.

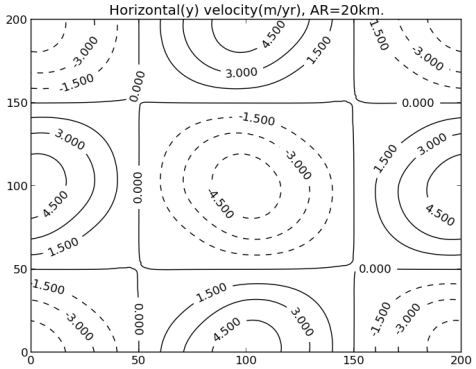
FIGURE 3.11: Contour plots of velocity components for domain with aspect ratios 5 (left column) and 10 (right column), for ISMIP–HOM experiment A, as computed with Fluidity. The indices 0 to 200 represent the normalised distance $x, y = 0$ and L respectively.



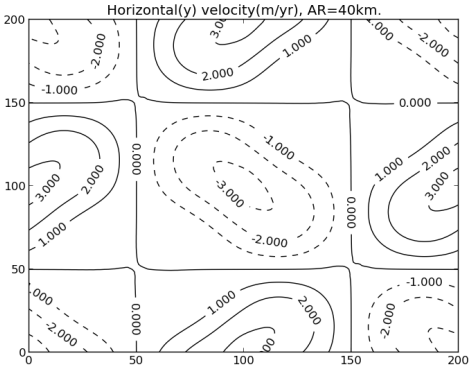
(A) x–component, aspect ratio 20.



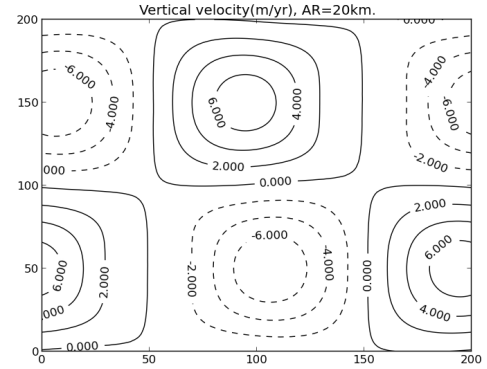
(B) x–component, aspect ratio 40.



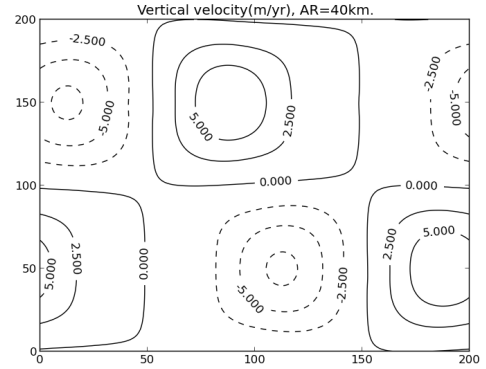
(C) y–component, aspect ratio 20.



(D) y–component, aspect ratio 40.

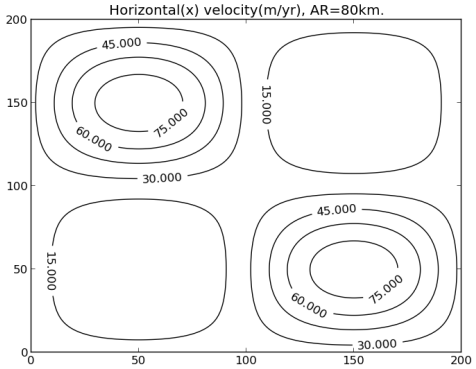


(E) z–component, aspect ratio 20.

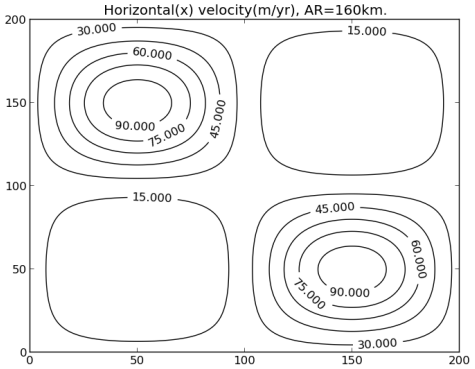


(F) z–component, aspect ratio 40.

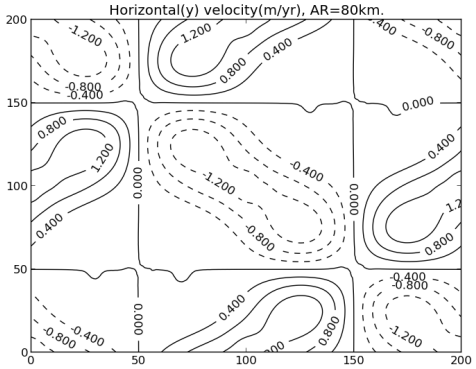
FIGURE 3.12: Contour plots of velocity components for domain with aspect ratios 20 (left column) and 40 (right column), for ISMIP–HOM experiment A, as computed with Fluidity. The indices 0 to 200 represent the normalised distance $x, y = 0$ and L respectively.



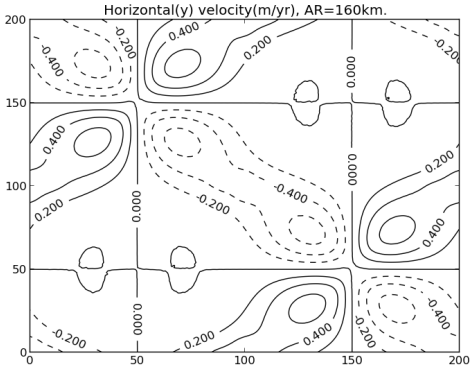
(A) x–component, aspect ratio 80.



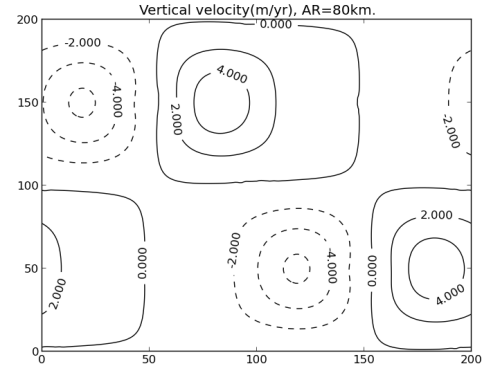
(B) x–component, aspect ratio 160.



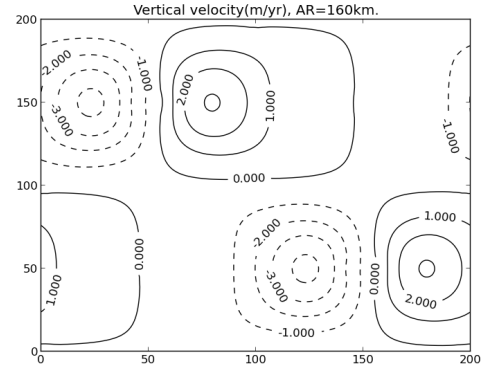
(C) y–component, aspect ratio 80.



(D) y–component, aspect ratio 160.



(E) z–component, aspect ratio 80.



(F) z–component, aspect ratio 160.

FIGURE 3.13: Contour plots of velocity components for domain with aspect ratios 80 (left column) and 160 (right column), for ISMIP–HOM experiment A, as computed with Fluidity. The indices 0 to 200 represent the normalised distance $x, y = 0$ and L respectively.

3.6.2 Experiment B – ice flow over a rippled bed

Experiment B varies from A in the form of the bottom topography used. The topography of this testcase does not vary in the transverse direction, it is given by (3.42) and illustrated in Figure 3.14.

$$z_b(x, y) = z_s - 1000 + 500 \sin(\omega x). \tag{3.42}$$

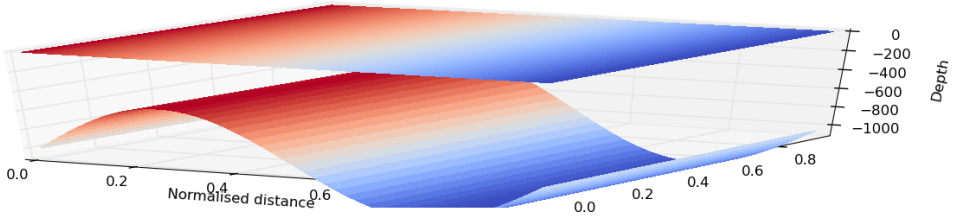
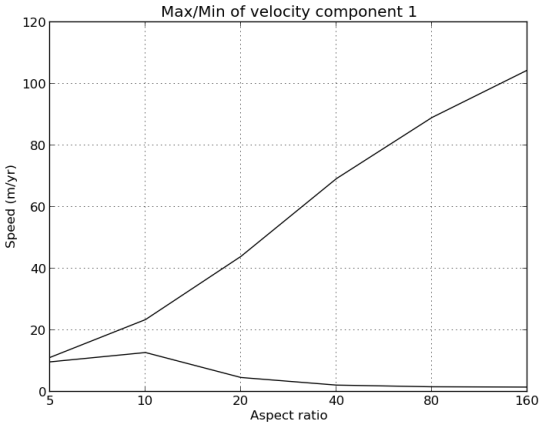
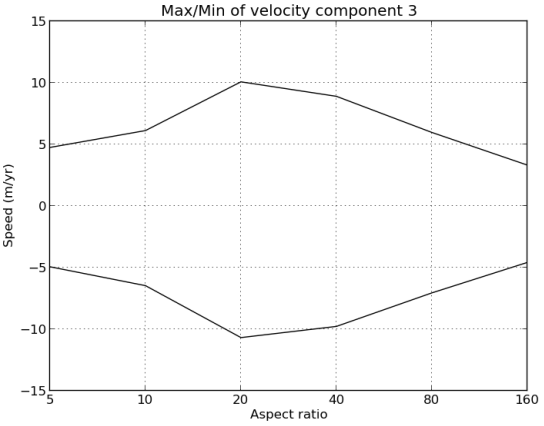


FIGURE 3.14: Illustration of the three–dimensional domain for experiment B.

Given the homogeneity of the domain in the transverse direction, this simulation was run in two–dimensions. Results are shown in figures 3.15, 3.16 and 3.17. Discussion of results is deferred until section 3.7.

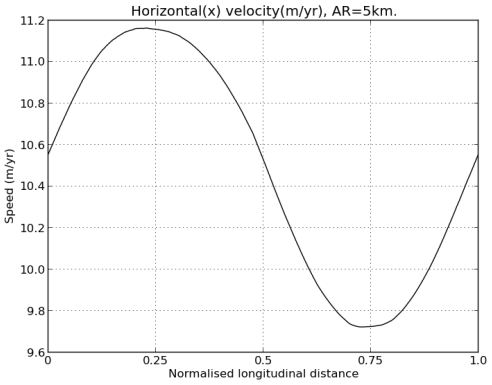


(A) Horizontal component of velocity

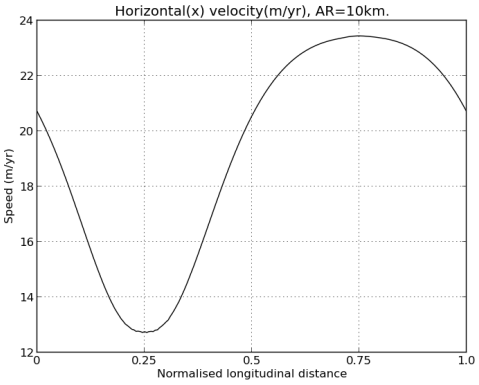


(B) Vertical component of velocity

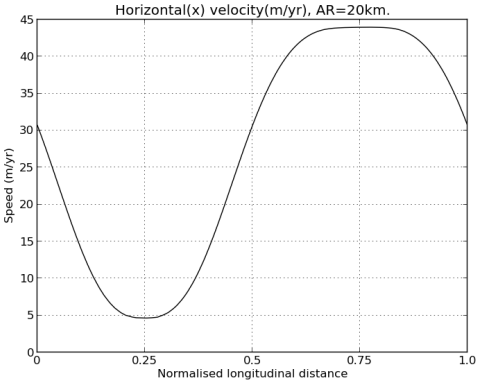
FIGURE 3.15: Maximum and minimum values of surface velocity components as computed with Fluidity, in metres per year, plotted against domain aspect ratio, for ISMIP–HOM experiment B.



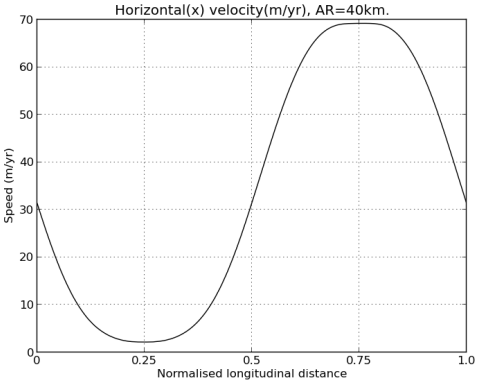
(A) Aspect ratio 5.



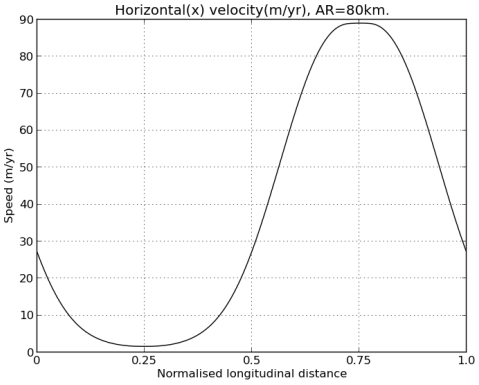
(B) Aspect ratio 10.



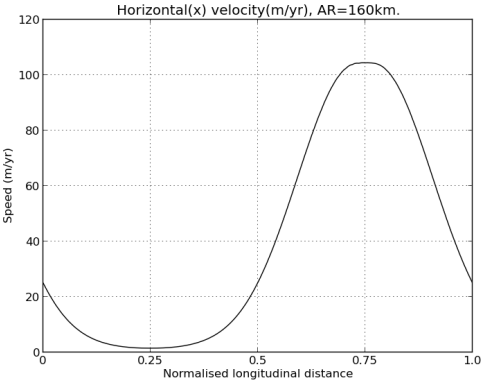
(C) Aspect ratio 20.



(D) Aspect ratio 40.

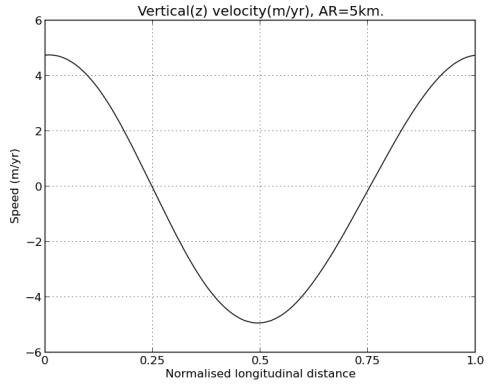


(E) Aspect ratio 80.

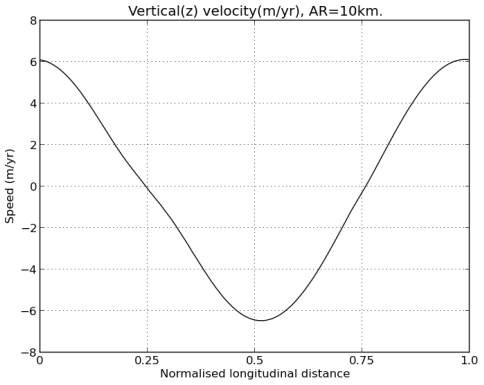


(F) Aspect ratio 160.

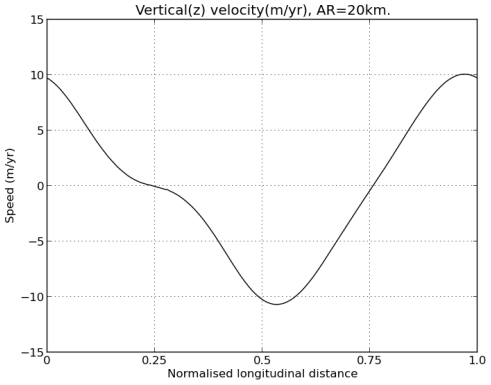
FIGURE 3.16: Profile of horizontal surface velocity, plotted against normalised longitudinal distance, for simulations with aspect ratios from 5 to 160, for ISMIP–HOM experiment B, as computed with Fluidity.



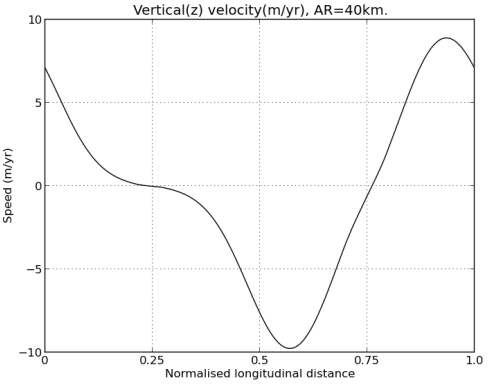
(A) Aspect ratio 5.



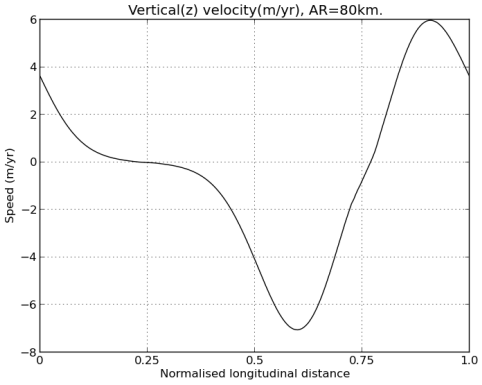
(B) Aspect ratio 10.



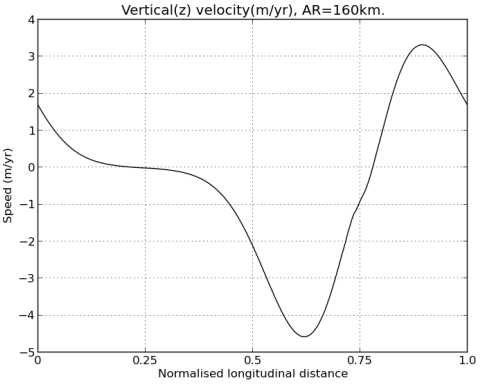
(C) Aspect ratio 20.



(D) Aspect ratio 40.



(E) Aspect ratio 80.



(F) Aspect ratio 160.

FIGURE 3.17: Profile of vertical surface velocity, plotted against normalised longitudinal distance, for simulations with aspect ratios from 5 to 160, for ISMIP–HOM experiment B, as computed with Fluidity.

3.6.3 Experiments C and D – ice stream flow

For experiments C and D, consider a slab of glacial ice, with a thickness of $1km$ on an inclined plane. These experiment vary from experiment A in two significant ways:

- The bottom topography is flat, rather than bumpy;
- The ice is no longer fastened to the bottom, but allowed to slide.

A linear Rayleigh friction is applied at the bottom boundary (as described by equations (3.13)). For experiment C, the friction coefficient, β^2, is specified as:

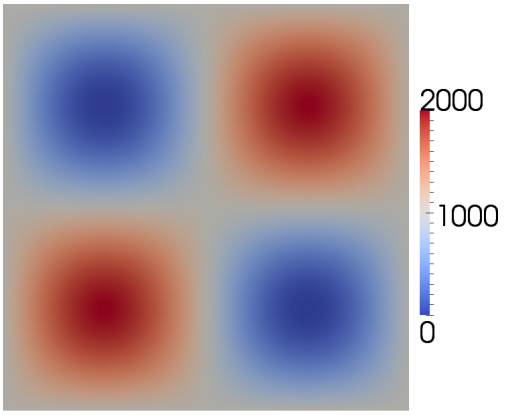
$$\beta^2(x, y) = 1000 + 1000 \sin(\omega x) \sin(\omega y), \tag{3.43}$$

where $\omega = \frac{2\pi}{L}$ is the frequency of the basal bumps and L is the length of the domain, which varies similarly to experiment A, such that domains of aspect ratios 5 to 160 are used. Experiment D is similar but removes the bottom friction variation in the transverse direction, the bottom friction thus specified as:

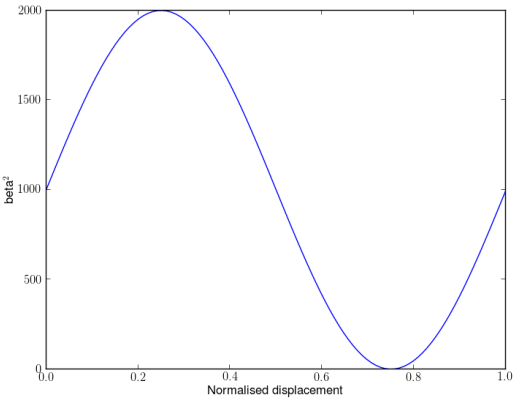
$$\beta^2(x, y) = 1000 + 1000 \sin(\omega x). \tag{3.44}$$

Figure 3.18 shows the friction coefficient for experiments C and D.

Given the similarity between experiments C and D, only the two–dimensional experiment D was performed, the results to which are presented in figures 3.19, 3.20 and 3.21, and discussed in section 3.7.

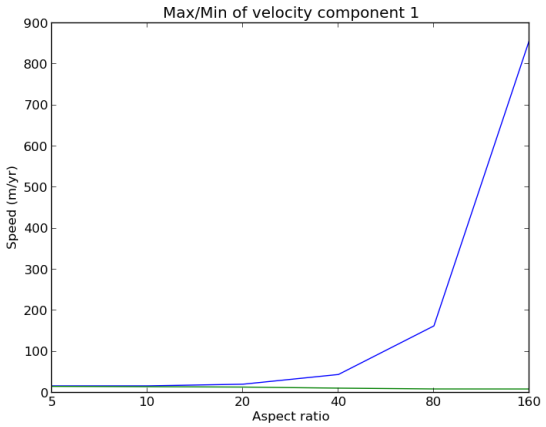


(A) Experiment C.

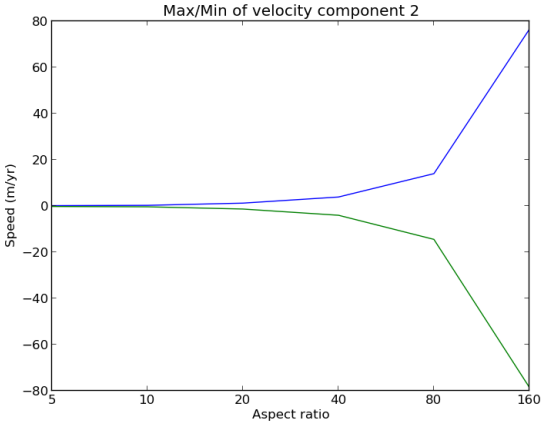


(B) Experiment D.

FIGURE 3.18: Bottom friction coefficient, β^2, for linear Rayleigh friction law, as used in ISMIP–HOM experiments C and D.

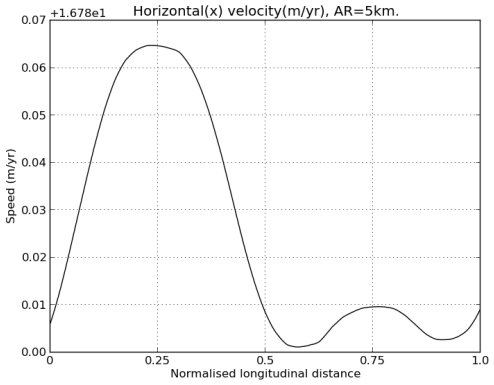


(A) Horizontal component of velocity

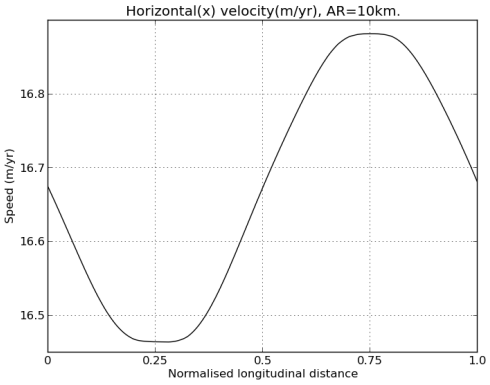


(B) Vertical component of velocity

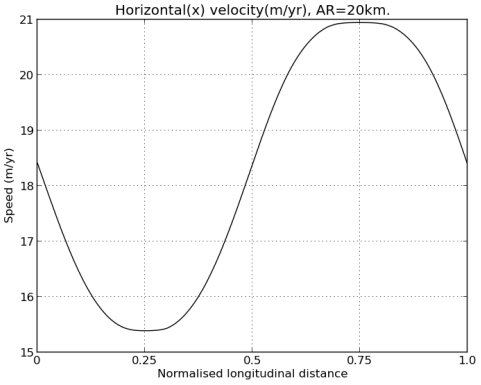
FIGURE 3.19: Maximum and minimum values of surface velocity components as computed with Fluidity, in metres per year, plotted against domain aspect ratio, for ISMIP–HOM experiment D.



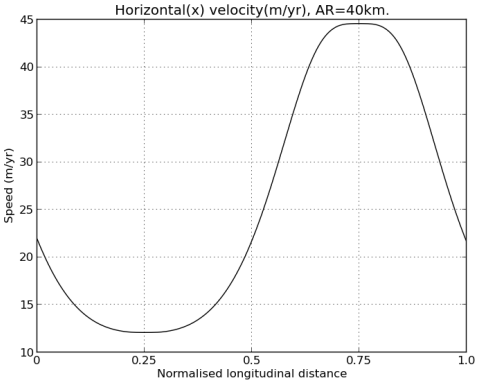
(A) Aspect ratio 5.



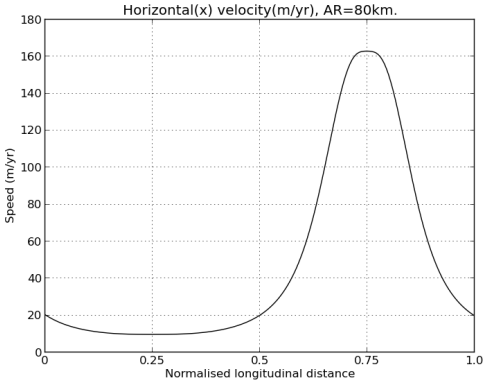
(B) Aspect ratio 10.



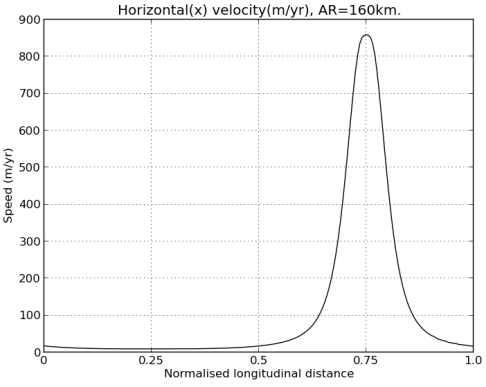
(C) Aspect ratio 20.



(D) Aspect ratio 40.

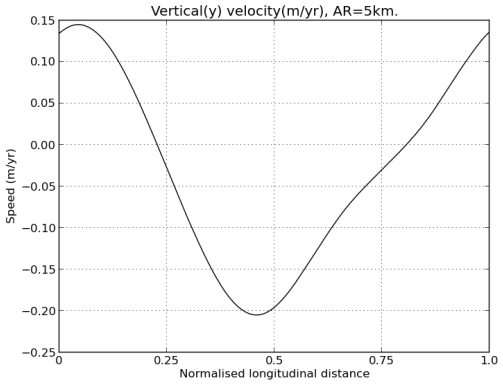


(E) Aspect ratio 80.

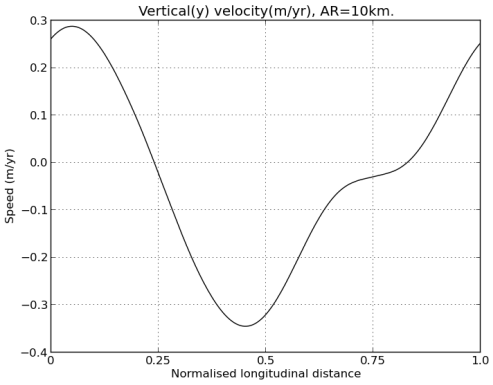


(F) Aspect ratio 160.

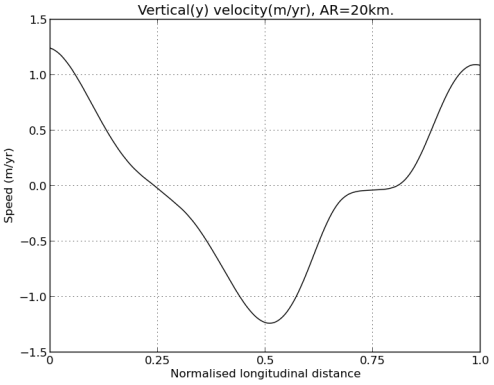
FIGURE 3.20: Profile of horizontal surface velocity, plotted against normalised longitudinal distance, for simulations with aspect ratios from 5 to 160, for ISMIP–HOM experiment D, as computed with Fluidity.



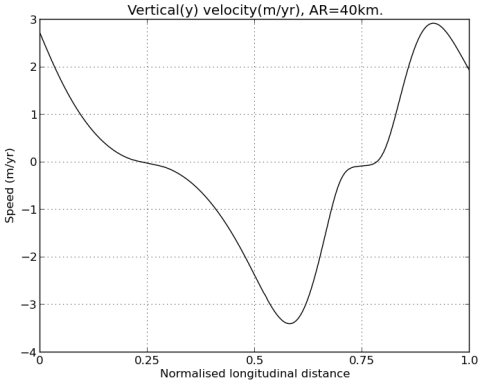
(A) Aspect ratio 5.



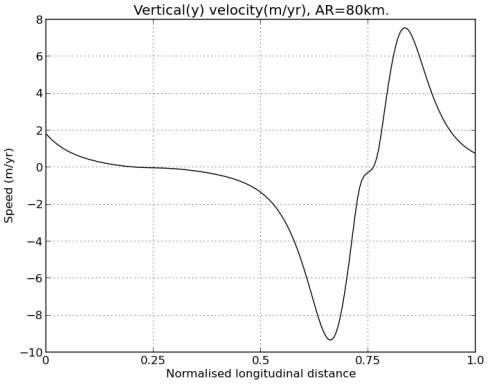
(B) Aspect ratio 10.



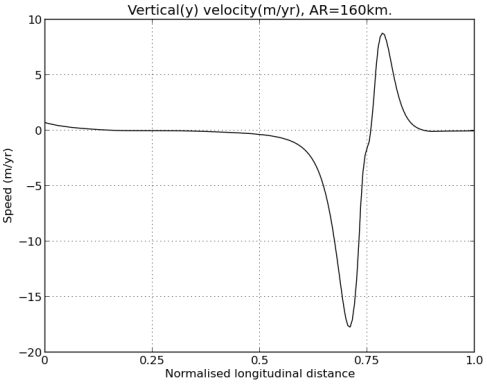
(C) Aspect ratio 20.



(D) Aspect ratio 40.



(E) Aspect ratio 80.



(F) Aspect ratio 160.

FIGURE 3.21: Profile of vertical surface velocity, plotted against normalised longitudinal distance, for simulations with aspect ratios from 5 to 160, for ISMIP–HOM experiment D, as computed with Fluidity.

3.6.4 Experiment E – Haut Glacier d'Arolla

Experiment E forms the only realistic example in the ISMIP–HOM suite of experiments. It is an experiment along the central flowline of a glacier in the Swiss Alps, the "Haut glacier d'Arolla", shown in figure 3.22.

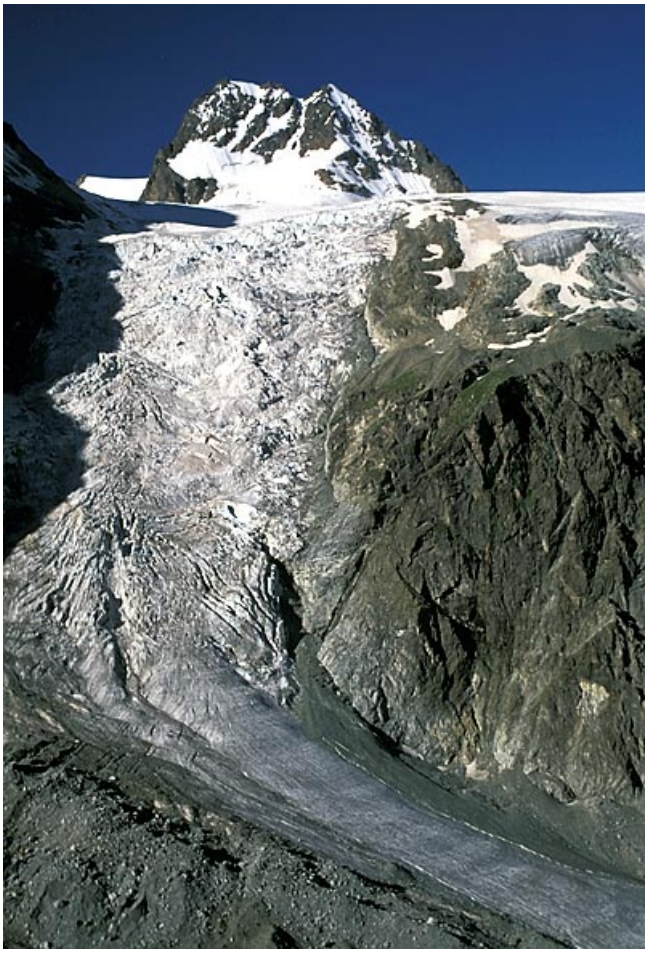


FIGURE 3.22: The d'Arolla glacier in Switzerland. A section along the centre of this glacier was used as the domain for the ISMIP–HOM experiment E. ["Arollagletscher". Licensed under Creative Commons Attribution-Share Alike 3.0 via Wikimedia Commons]

A digital elevation model (DEM) with elevations for the top and bottom surfaces was made available by the organisers of the ISMIP–HOM testcases [Pattyn and Payne, 2006]. The DEM provided elevations to the nearest centimetre, but had a horizontal resolution of $100m$. The total extent of the glacier in the longitudinal direction is $5km$. A python script was written to read in the DEM and construct the geometry of the domain in the Gmsh geometry

format, which was then meshed using the Gmsh meshing algorithm [Geuzaine and Remacle, 2009]. One such mesh, in relatively low resolution ($40m$ element edge length) and scaled in the vertical, is shown in figure 3.23.

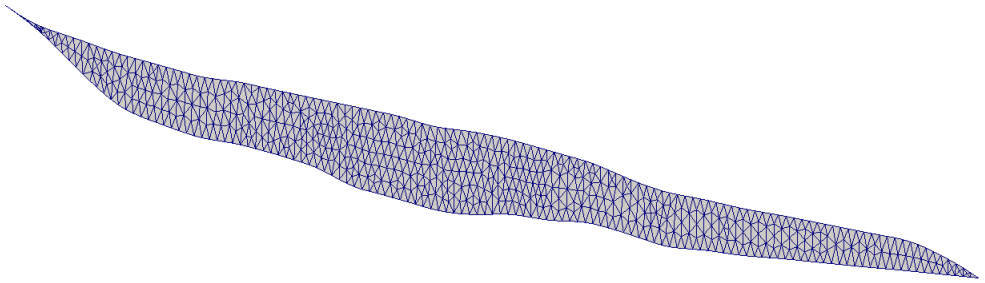


FIGURE 3.23: Low resolution ($40m$ edge length) mesh of the domain for ISMIP–HOM experiment E, generated from digital elevation map. The vertical dimension has been scaled by a factor of 2, for illustration purposes only.

The same parameters were used for this simulation as the simulation of experiment A, and are tabulated in table 3.3. The glacier is considered frozen to the ground and therefore the no–slip boundary condition (3.12) is applied.

Results for experiment E are presented in figures 3.24, 3.25.

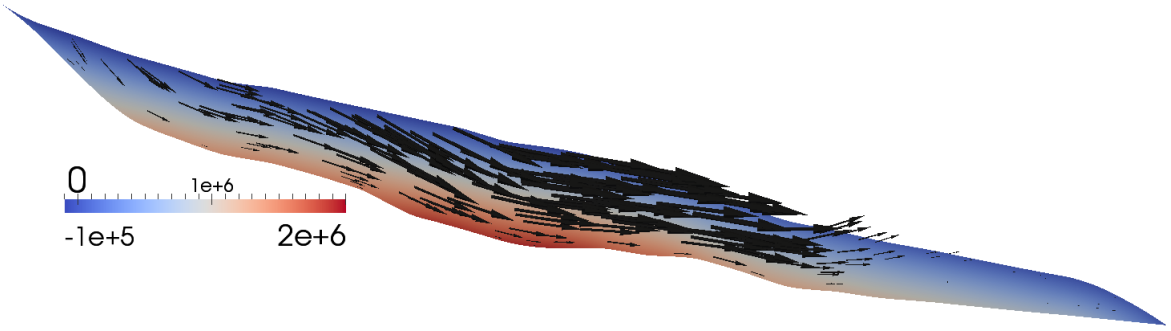
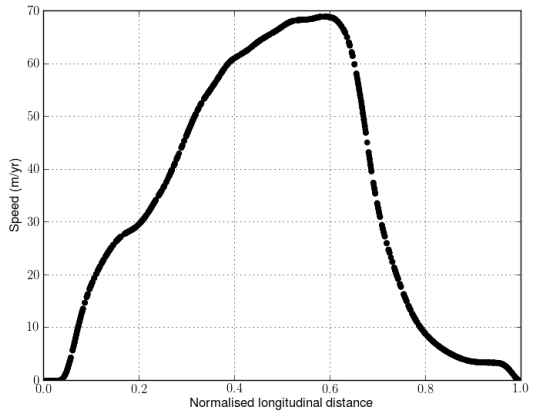


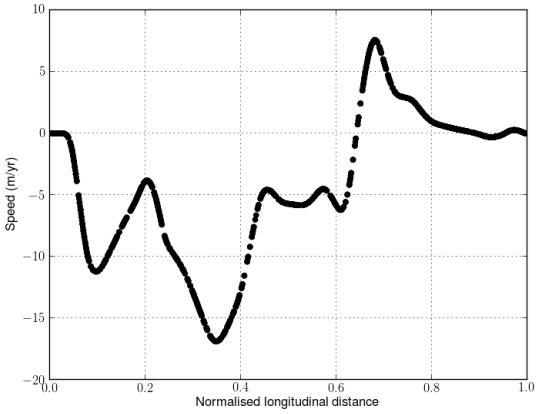
FIGURE 3.24: Results of ISMIP–HOM experiment E, computed with Fluidity, along the long profile of glacier Haut d'Arolla. The glacier colourmap represents the pressure, whilst the arrows represent the glacier velocity at steady state. The magnitude of the arrows represent the magnitude of velocity and the direction of the arrowhead the velocity direction.

3.6.5 Experiment F – Free surface relaxation

The final experiment of ISMIP–HOM, experiment F, requires a prognostic, periodic free–surface, a capability that Fluidity does not currently have. This



(A) Horizontal component of velocity



(B) Vertical component of velocity

FIGURE 3.25: Surface velocity of the glacier Haut d'Arolla as computed with Fluidity (ISMIP–HOM experiment E) at steady–state, with no–slip bottom boundary condition. Figures shown velocity component, in metres per year, plotted against the normalised longitudinal distance.

experiment could therefore not be undertaken.

3.6.6 Mesh generation

The preparation phase of running the ice sheet models described in the previous section, required the generation of appropriate meshes. For an illustration of the process of producing the meshes for experiments A to D, we will use the example of a 2–D mesh for exercise B.

The domain to be modelled is shown in figure 3.26:

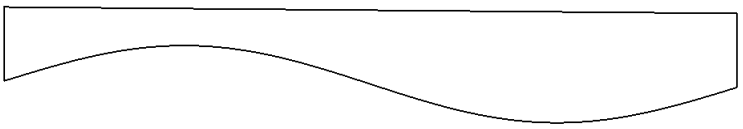


FIGURE 3.26: Domain for experiment B. The domain is inclined at an angle α and has a sinusoidal bottom topography.

The desired geometry can be obtained with a series of extrusions and transformations, beginning from a single node:

Extrude node in y–direction. This operation produces a line of length 1.

Extrude line in x–direction. The operation produces a square of length 1.

Mesh the square domain. The square domain is passed to Gmsh [Geuzaine and Remacle, 2009], which produces a mesh of the unit square. This is illustrated in figure 3.27.

Transform the domain. This lengthens the domain and produces the sinusoidal topography. While the above described extrusions are performed using the Gmsh geometry module, this step is performed with a python script.

Skew or rotate the domain. The domain is inclined at an angle α by skewing the nodes, or by applying a rotation. The result of this is shown in figure 3.28.

A more detailed description of how the layered and unstructured meshes were obtained can be found in appendix B.

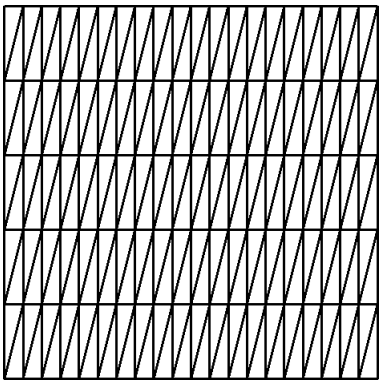


FIGURE 3.27: Structured 2–D mesh, an intermediate step in the production of the mesh of the domain shown in figure 3.26.

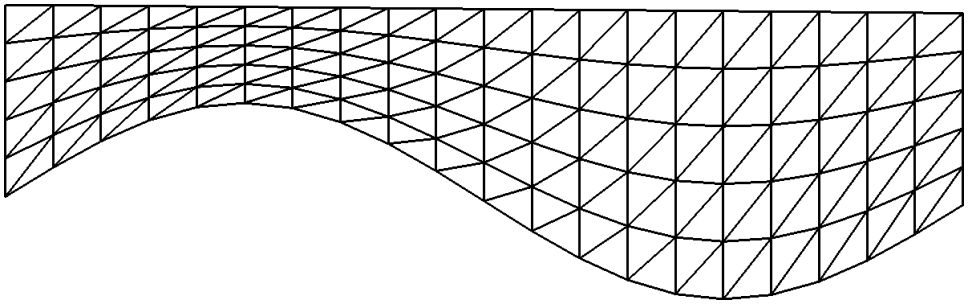


FIGURE 3.28: Transformed and skewed mesh of desired geometry.

We note that the mesh obtained in the above scenario has the following features:

- Constant number of σ–layers, resulting in a higher effective resolution in regions of thinner ice sheet depth and lower resolution in the regions of maximum depth.
- The leftmost and rightmost boundaries have been scaled by the same factor and thus can be related by a simple translation, ensuring there are no issues introducing periodic boundaries in Fluidity.
- The vertically aligned nodes of the original cube have been preserved.

It must be noted that Fluidity is not restricted to domains that are vertically aligned or that have the structure of the mesh in figure 3.28. In particular, unstructured meshes can also be produced, as in figure 3.29. The mesh illustrated in figure 3.29 is produced by altering the sequence of operations of

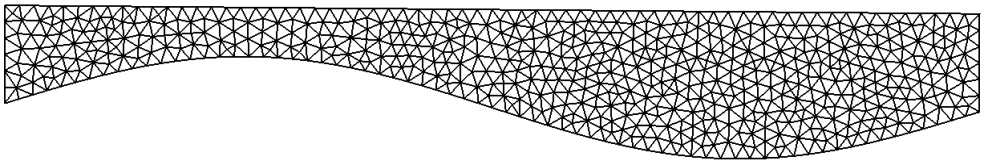


FIGURE 3.29: Illustration of unstructured mesh for ISMIP–HOM experiment B.

producing a mesh, in particular the geometry of the mesh is obtained before being passed to the Gmsh library for triangulation. Alternatively, an unstructured mesh can be produced and then transformed. This results in a mesh that is unstructured similar to the mesh in figure 3.29 but has the resolution varying in the same manner as the mesh in figure 3.28 (i.e. node density proportional to depth). Finally, Fluidity's adaptivity library can be used to produce meshes that conform to the solution field. Experiment B has been run on all the different meshes that have been described in this section, with results consistent with the results presented in section 3.6.2.

Experiment E, the only realistic case described in this chapter, was produced by parsing a digital elevation model which contained the elevation of the top and bottom boundaries along the centerline of the glacier. The mesh produced was shown previously in figure 3.23. It is noteworthy that due to the unstructured nature of the mesh, Fluidity does not suffer from issues due to layers of the mesh converging in regions of very shallow glacier depth. 'Pinched' regions can therefore easily be represented and modelled using Fluidity.

3.6.6.1 Mesh resolution for presented results

The results presented in this thesis for the ISMIP–HOM experiments A to D were all run on meshes of the form shown in figure 3.28.

Meshes were generated by transforming a square (or cubic) domain. The square is initially meshed by splitting it into $N_x(\times N_y) \times N_z$ layers and further splitting it into triangles (or tetrahedra). Table 3.4 describes the number of layers for each aspect ratio:

AR	N_x, N_y	N_z
5	40	10
10	40	10
20	40	10
40	60	10
80	90	10
160	135	10

TABLE 3.4: Table of number of layers for simulations of ISMIP–HOM exercises.

The number of layers for each aspect ratio was chosen such that the two conditions were met:

- The solution was accurate yet the computational cost was not prohibitive. Mesh convergence was ensured by checking that increasing the mesh resolution did not result in significantly different results.
- The element aspect ratio did not exceed approximately 15. It was observed during initial runs that meshes with element aspect ratios exceeding approximately 20 result in poorly conditioned matrices that were harder to solve, and thus more computationally expensive.

For the glacier Haut d'Arolla, results presented were produced on a mesh with $10m$ resolution.

3.7 Discussion

The aim of this chapter is to describe and demonstrate the methods with which Fluidity solves the three–dimensional, non–linear, non–Newtonian, full–Stokes equations, using Glen's flow law. The equations and their discretisation as implemented in Fluidity are described in section 3.2, along with a brief description of the solvers employed in Fluidity. The model is subjected to a vigorous verification and validation process, by first performing a test using the method of manufactured solutions, in section 3.3.1. This demonstrates

that the Stokes model for ice sheet modelling is correctly implemented within Fluidity, as both velocity and pressure converge at their theoretically expected order.

Furthermore, the use of Fluidity's dynamic, adaptive remeshing technology is presented in a simulation of an idealised ice stream. This demonstrates that Fluidity is capable of dynamically optimising the mesh by increasing the resolution in regions of interesting dynamics, whilst lowering the resolution in areas where it is not needed. Furthermore, it was shown that the anisotropy of the adapted mesh offers the potential for even further reduction in overall number of degrees of freedom compared to isotropic, unstructured meshes.

A scaling analysis was performed (section 3.5), which indicates that Fluidity can successfully be run in parallel. The scaling performance demonstrated that the scaling performance, although good, is far from ideal and falls below the near–linear speedup expected of a state–of–the–art ice sheet model. We note, however, a very promising result when doubling the core–count from 16 to 32, where a near double speedup was realised. Several reasons have been identified for why the scaling analysis may have fallen short of expected performance, primarily related to the non–exclusive use of the cluster on which the analysis was performed.

Following this, a rigorous study was undertaken that presents Fluidity's results to the exercises of the Ice Sheet Model Intercomparison Study. Figure 3.30 offers one such comparison, showing the results of Fluidity in tandem with the equivalent plot of the Elmer/Ice ice sheet model [Gagliardini and Zwinger, 2008]. The Elmer/Ice model is chosen, as it is a full–Stokes, unstructured mesh, finite element model that took part in the original study. The use of a similar discretisation method makes the comparison with Elmer/Ice the most appropriate.

In figure 3.30 we observe that Fluidity exhibits both the same trends as Elmer/Ice, as well as good quantitative agreement for experiment A. Similarly

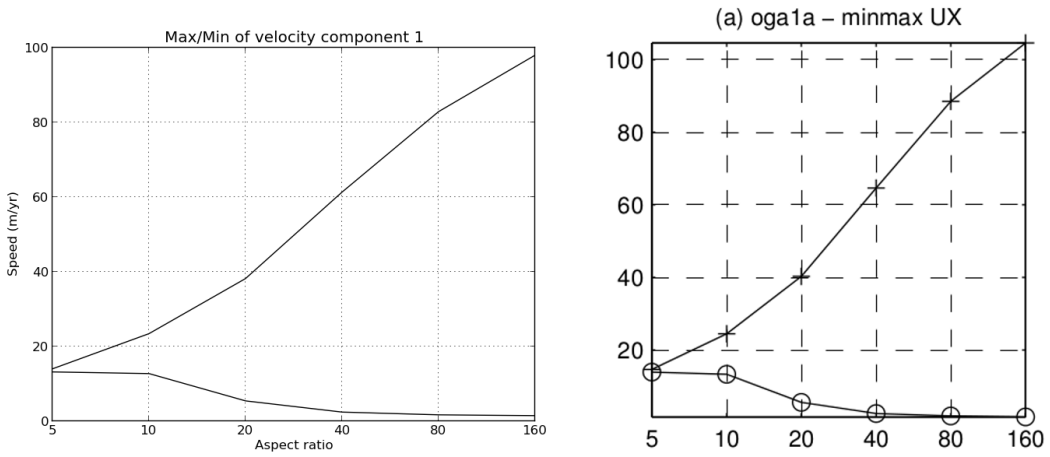


FIGURE 3.30: Comparison of Fluidity (left plot) with Elmer/Ice (right plot) for maximum and minimum values of velocity x–component plotted against the aspect ratios used in ISMIP–HOM experiment A. The right plot is used here for comparison purposes and was taken from Gagliardini and Zwinger [2008]

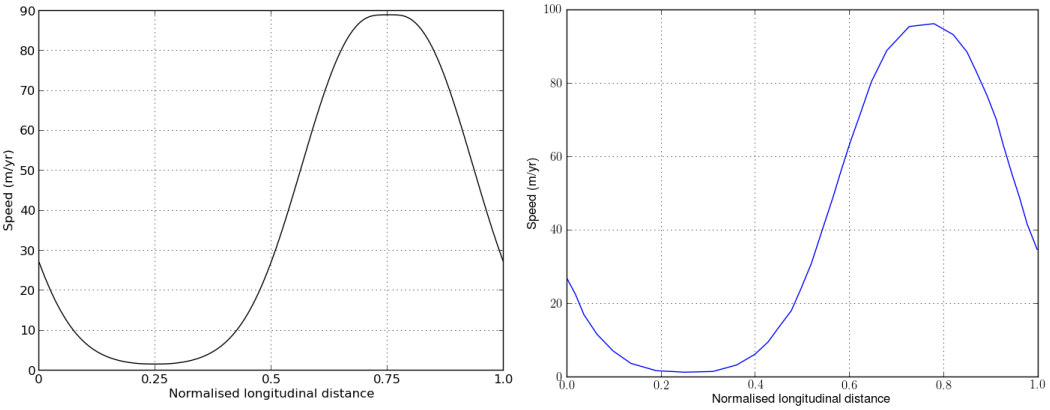


FIGURE 3.31: Comparison of Fluidity (left plot) with mean value of models that took part in ISMIP–HOM (right plot), for surface velocity x–component plotted against the normalised longitudinal distance for experiment B aspect ratio 80. The data for the plot on the right was extracted from Pattyn et al. [2008].

good agreement is observed in the other tests (see figures 3.31 and 3.32), particularly in terms of trends. We do, however, make note the following caveat: some of the high aspect ratio domains (with $L = 80$ and $L = 160$) are not fully converged, based on the convergence criterion (3.38) with $\epsilon = 1 \times 10^{-9}$. This is due to the increasing computational cost for the higher aspect ratio domains. The cost is increased due to the increasing number of degrees of freedom as well as the increase in element aspect ratio. It has been observed that the larger the element aspect ratio, the harder the system is to solve (the

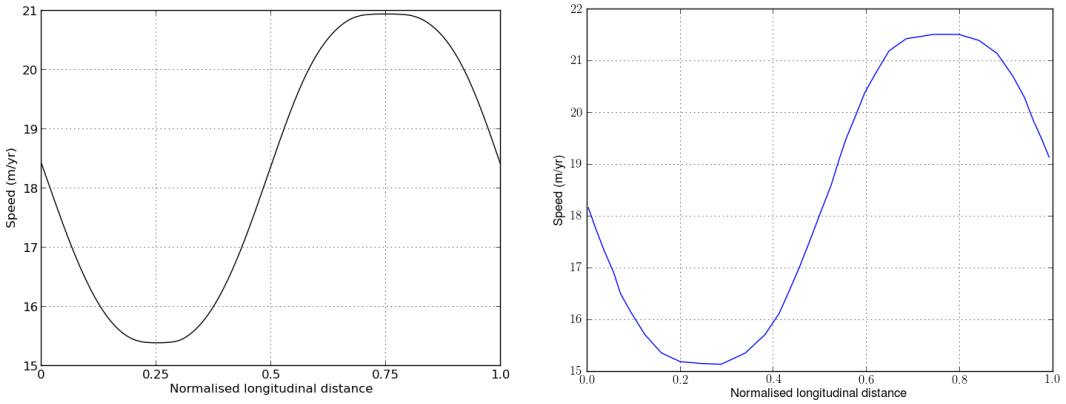


FIGURE 3.32: Comparison of Fluidity (left plot) with mean value of models that took part in ISMIP–HOM (right plot), for surface velocity x–component plotted against the normalised longitudinal distance for experiment D aspect ratio 20. The data for the plot on the right was extracted from Pattyn et al. [2008].

matrix is less well–conditioned) and thus the more computationally expensive. In general, however, very good agreement was observed between Fluidity and other ISMIP–HOM models.

A notable observation from a closer inspection at the results of experiments A and B, is the influence of the aspect ratio on the maximum horizontal (x–component) velocity (see figures 3.10 and 3.15). The larger the aspect ratio (L/H) of the domain, the higher the maximum horizontal velocity at the surface is. As the aspect ratio of the domain gets closer to unity (towards the $AR = 5$ side of the abovementioned figures), the velocity profile along the flowline appears smoothened. This is illustrated more clearly in figure 3.33. The longitudinal profile of these experiments is such that the shallow ice approximation (SIA) predicts a surface velocity independent of the longitudinal span of the domain [Pattyn et al., 2008] (and hence independent of the aspect ratio). The surface velocity, according to SIA theory, is given by:

$$v_x(z_s) = v_x(z_b) + \frac{2A}{n+1} (\rho g \tan \alpha)^n H^{n+1}, \tag{3.45}$$

where $v_x(z_b)$ is the velocity in the x–direction at the bottom of the domain

(0 in this experiment), A is the ice–flow parameter, $n = 3$ is the Glen's flow exponent, α the domain inclination and H the height along the domain. The surface velocity obtained by SIA theory is plotted in figure 3.33, and reaches a maximal velocity of approximately $119ms^{-1}$. We note that all of the higher–order models, including Fluidity, showed results that were dependent on the aspect ratio of the domain. This behaviour is caused by the longitudinal stress gradients arising from topographic features. These stress gradients have the effect of 'smoothing' the velocity profile, with the maximum velocity ranging from around $100ms^{-1}$ at the highest aspect ratios to $10ms^{-1}$ at the lower aspect ratios (where longitudinal stress terms become even more dominant). At the lowest aspect ratio ($AR = 5$), the profile is almost flat, suggesting that topographic features have minimal effect on the maximal velocity along the domain.

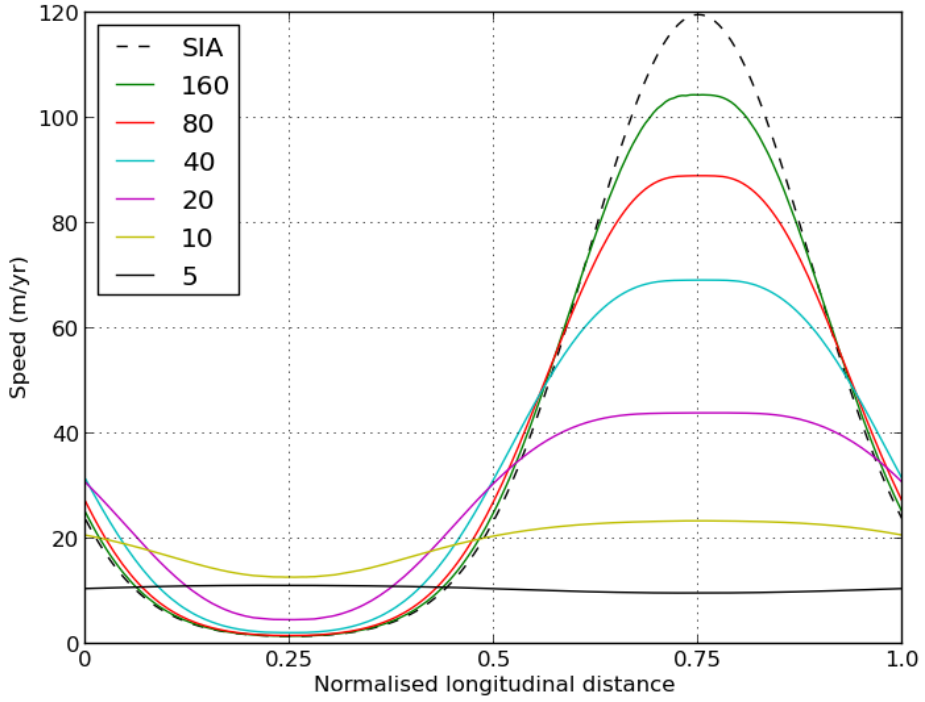
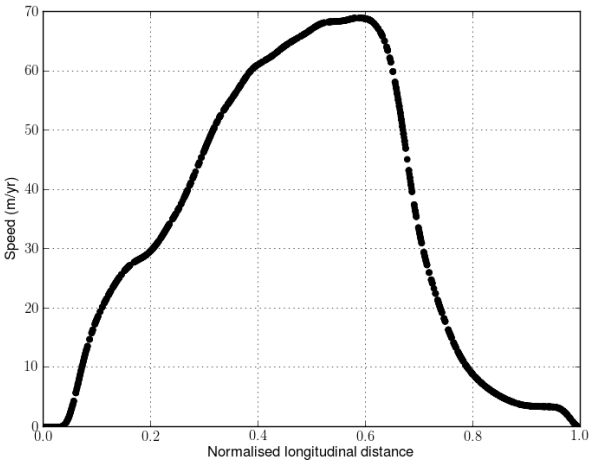


FIGURE 3.33: Maximum surface velocity (x–component) plotted against normalised longitudinal distance for experiment B. The solid lines indicates the numerically obtained profile for the simulations of varying domain aspect ratio (5, 10, 20, 40, 80, 160). The dashed line indicates the result obtained by shallow ice approximation (SIA) theory.

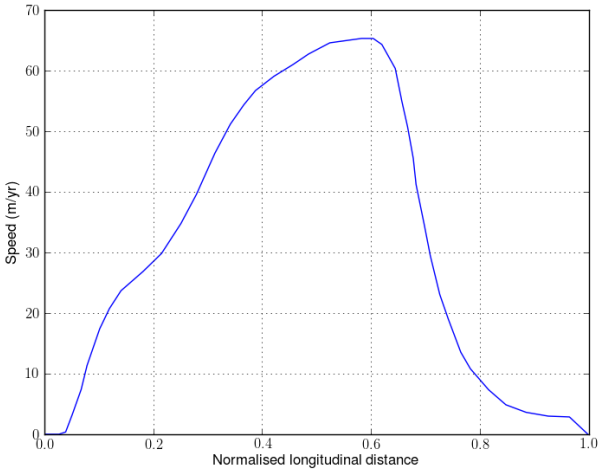
A second notable feature of the results of experiment B is the difference in the tendency observed between the velocity profiles of $AR = 5$ and the rest of the simulations. In $AR = 10$ to 160 the ice sheet decelerates when passing over the bump and accelerates through the topographic trough; the opposite behaviour is observed in the domain of $AR = 5$. The difference is easier to see in figure 3.16 rather than 3.33, due to the range of the vertical axis. As described in Pattyn et al. [2008], this difference is explained by the principle of conservation of mass. Pattyn et al. [2008] argues the horizontal ice flux can not be balanced by the vertical flux at the free surface, as the required vertical velocity would be too large for the given depth. It must be noted that this behaviour is an artefact of the diagnostic nature of these experiments, and the behaviour would not be evident if the free–surface was allowed to respond. Also notable is the fact that in the three–dimensional experiment A, where the topography is rippled in the transverse, the flow is able to accelerate around the bump, thus allowing the $AR = 5$ simulation to behave similarly to the simulations in domains with higher aspect ratios. We remind the reader that *all* other full–Stokes models participating in the ice–sheet model intercomparison study exhibited the same behaviour as Fluidity.

As a further example of good agreement between Fluidity and the benchmark results, figure 3.34 shows a result obtained with Fluidity for the glacier Haut d'Arolla, next to the equivalent result derived using the Elmer/Ice model, from Gagliardini and Zwinger [2008].

A further observation regarding the use of Fluidity for ice sheet modelling is the ability to choose between different meshing strategies for ice sheet simulations. We note that results for experiment B of ISMIP–HOM have been obtained with a) fixed, layered, vertically aligned meshes (as those presented for the ISMIP–HOM), b) fixed, unstructured meshes as well as c) dynamically adaptive, unstructured meshes. Figure 3.35 shows a comparison of representative results from simulations with two different meshing setups. We can see that despite the two simulations being run on different meshes, the result is indistinguishable. We can therefore argue that for the purpose of modelling



(A) Fluidity



(B) Elmer/Ice

FIGURE 3.34: Comparison of Fluidity result for ISMIP–HOM experiment E with Elmer/Ice.

idealised ice sheets and glaciers, layering and/or vertically aligned nodes are not essential, but rather that fully unstructured meshes can be used successfully.

The experiments described in this chapter lay the foundations for simulations of more complicated, realistic simulations of Greenland that can be used for glaciological purposes.

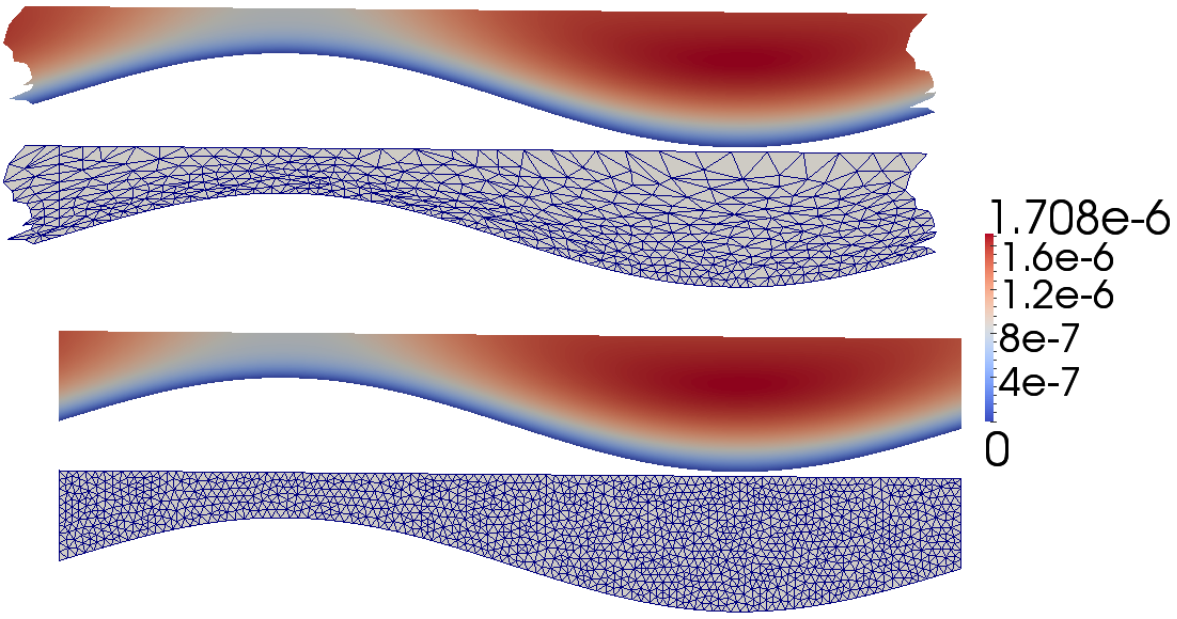


FIGURE 3.35: Fluidity results of ISMIP–HOM experiment B using dynamically adaptive (top two) and fixed, unstructured meshes (bottom two). The colourmap relates to the magnitude of the x–component of velocity, in metres per second. The colourplots, as well as their corresponding meshes are from the 10th timestep of a simulation initialised with an ice sheet at rest.

CHAPTER 4

MODELLING OF THE GREENLAND ICE SHEET

4.1 Introduction

Numerical modelling of ice sheets is a subject of growing interest due to the potential role ice sheets could have on sea level rise in a changing climate [Gregory and Huybrechts, 2006]. The full–Stokes equations have been shown to offer the most accurate and true representation of ice sheet flow [Meur et al., 2004]. The fourth assessment report of the Intergovernmental Panel on Climate Change made a statement that a poor understanding of the importance of dynamic changes has limited the ability to put an upper bound on the contribution of ice sheets to sea level rise by the year 2100 [IPCC, 2007]. It is therefore essential that further investigations are undertaken in understanding the dynamics and thermodynamics of these changes. Numerical modelling has an important role in furthering our understanding of ice sheet dynamics.

This chapter furthers the development of the full–Stokes modelling framework described in chapter 3, by attempting preliminary idealised simulations of Greenland, on a realistic domain.

4.2 Model setup

The three–dimensional, non–linear, non–Newtonian Stokes equations, discretised with the finite element method as described in chapter 3 are used for modelling the Greenland ice sheet.

A range of simulations of increasing complexity are performed, beginning with a highly idealised simulation of a domain with the outline of Greenland and a flat surface and bottom topography, and progressively moving to a realistic domain of Greenland. Beginning from the flat domain, we introduce a realistic hypsometry for the bottom surface only, and introduce the full Greenland domain (with realistic top and bottom surfaces) after that. The meshes used are described in section 4.3.

The model uses Glen's flow law (3.4), as introduced in the previous chapter, with $\eta_{\mathbf{u}}$ given by:

$$\eta_{\mathbf{u}} = \frac{1}{2} A^{-1/n} \dot{\epsilon}_e^{(1-n)/n}, \tag{4.1}$$

where unlike previously (where a constant deformation rate factor was used) we now use the temperature–dependant deformation rate factor:

$$A = A(T) = A_0 e^{\left(\frac{-Q}{R(T_0+T')}\right)}, \tag{4.2}$$

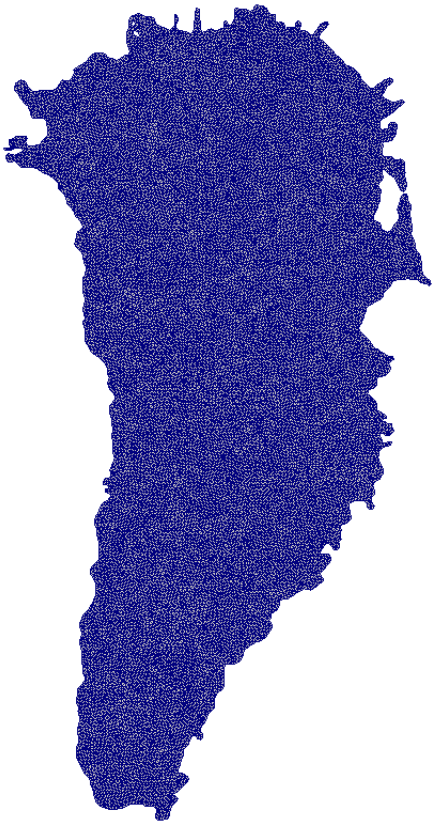
where T_0 is the melting temperature at low pressure ($273.16K$) and T' is the temperature relative to the melting temperature and is given by $T' = T - T_m$. T is the absolute temperature, and the melting temperature, T_m, is given by the relationship $T_m = T_0 - \beta p$, where β is the Clausius–Clapeyron constant and p the pressure. We note that Fluidity only uses the hydrostatic pressure for the computation of the pressure melting temperature, a simplification justified by the fact that the hydrostatic component forms the largest part of the pressure. The other parameters (A_0, Q, R) are as introduced previously. Parameters used for the simulations of Greenland in this chapter are detailed in table 4.1. We also make note that the pre–exponential factor is commonly found in literature multiplied by a constant flow enhancement factor, E. This term has been absorbed in the pre–exponential factor, as presented here.

Symbol	**Value**	**Units**
ρ , Density of ice	910	$kg\ m^{-3}$
g , Gravitational acceleration	9.81	$m\ s^{-2}$
n , Power law exponent	3	
A_0 , Pre–exponential factor	1.2×10^{-12}, if $T' \leq -10°C$ 5.748×10^{3}, if $T' > -10°C$	$s^{-1}\ Pa^{-3}$
Q , Activation energy	60, if $T' \leq -10°C$ 139, if $T' > -10°C$	$kJ\ mol^{-1}$
β , Clausius–Clapeyron constant	9.8×10^{-8}	$K\ Pa^{-1}$
R , Universal gas constant	8.314	$J\ mol^{-1}\ K^{-1}$

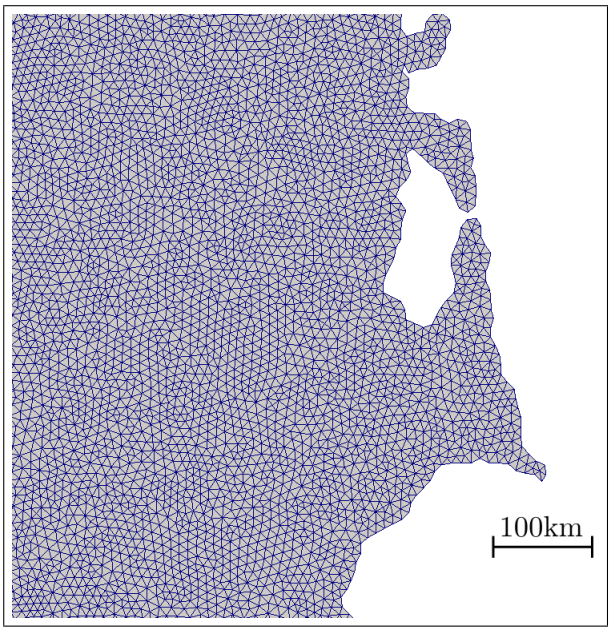
TABLE 4.1: Physical parameters used for Greenland simulations described in chapter 4.

4.3 Mesh generation

Full three-dimensional meshes of Greenland were generated, with a realistic surface and bottom topography. The Shingle [Candy, 2014] library was used to generate these meshes. The process of mesh generation requires a netCDF file of the bathymetry of the region. Here the RTopo [Timmermann et al., 2010] global bathymetry dataset has been used. A contour that traces the coastline can be obtained by calculating the contour at elevation zero, within Shingle. From the contour, a representation of the coastline is obtained. The



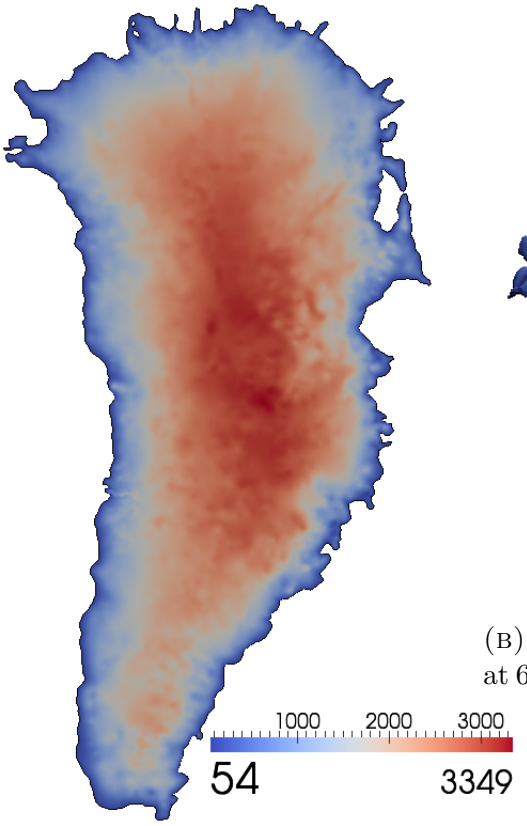
(A) Plan–view of a three–dimensional mesh of Greenland. The elements have a characteristic edge length of $10km$.



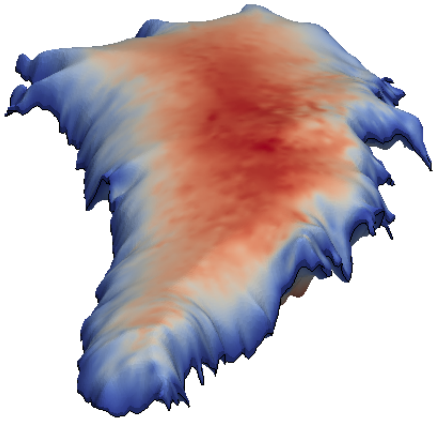
(B) Magnified region of the Greenland mesh. The scale indicates a distance of $100km$.

FIGURE 4.1: Mesh of realistic Greenland domain, at $10km$ resolution. Meshes provided by Dr. Adam Candy.

coastline representation is then converted to a Gmsh geometry file outlining the boundary of the region to be meshed. This geometrical boundary is in a UTM projection of the Greenland boundary. From this, a two–dimensional flat mesh is produced in the shape of Greenland. A uniform resolution mesh is obtained (at $5km$, $10km$ and $20km$ resolutions), with no refining near the boundaries or based on the depth, although such meshes could also be generated. The final step in 3–D mesh generation involves extruding the mesh to the top and bottom surfaces, to produce the meshes shown in figure 4.1. These meshes have vertically aligned nodes, terrain following layers (σ–layers) and are composed of three elements in the vertical. The realistic meshes of 5, 10 and $20km$ resolution of Greenland were generated with the support of Dr. Adam Candy, of Imperial College London.



(A) Mesh of Greenland, coloured by the thickness of the ice sheet.



(B) Same as image on the left, pitched at 60° and vertically scaled by a factor of 100.

FIGURE 4.2: Mesh of a realistic Greenland domain, vertically scaled by a factor of 100.

These meshes are fully–three dimensional and align with the surface of the ice sheet and the Greenland topography. This is illustrated in figure 4.2, which colours the domain according to ice sheet depth, and shows the domain at an incline.

Prior to modelling the ice sheet on a realistic Greenland domain, simplified domains with a flat top surface and progressively less smoothed topography representation were used. One such mesh is shown in figure 4.3. The 2–D flat mesh for these domains was obtained in a similar fashion to the technique described above, but using the GMT library. The extrusion to the bottom was performed with Fluidity's extrusion functionality and the topography was smoothed using the multi–dimensional image processing library `ndimage`, which is used to apply a Gaussian filter. The script that performs the smoothing and provides Fluidity with the depth is described in appendix B.3.

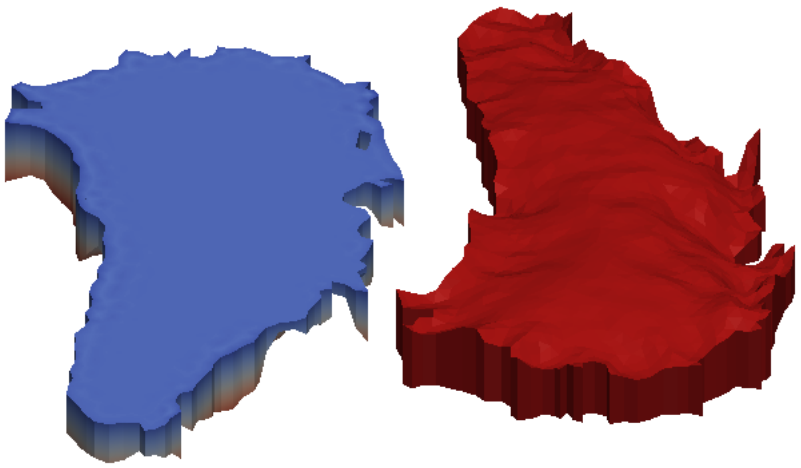


FIGURE 4.3: Domain with flat top (left) and smoothed Greenland topography (right). The top surface is shown in blue and the bottom surface in red. The topography was smoothed by applying a Gaussian filter with a standard deviation, $\sigma = 2.5$. The meshes are scaled in the vertical by a factor of 100.

4.3.1 Initialisation.

Due to the numerical cost of running a full–Stokes model in 3–D, the spin–up period of a simulation the scale of Greenland is prohibitively expensive. Initialising the model from rest (i.e. $\mathbf{u} = 0$) results in transient waves that could take up to 100 thousand years to be removed (one glacial cycle). This presents a problem for modellers of ice sheets solving the full–Stokes system. There are generally two accepted workarounds:

- Establish an initial condition by assimilating available information of the present state.
- Perform a spin–up cycle using a model that is less computationally expensive, for example a model that uses the Shallow Ice Approximation. The result of the spun–up simulation can then be used as the initial condition for the more expensive model.

For Fluidity, we obtain the result of a 125 thousand year spin–up cycle, performed using the SICOPOLIS [Greve, 1997] model. SICOPOLIS is based on the Shallow Ice Approximation (SIA) and is therefore numerically much more

efficient. The results of the spin–up were kindly provided by Dr. Hakkime Seddik, postdoctoral researcher at the Glacier and Ice Sheet research group of Hokkaido university. The data was provided in netCDF format and contained both the temperature and velocity distribution of the ice sheet, at 200 years before present.

4.3.2 Boundary conditions

The natural boundary condition (3.11) is applied to both the lateral and top surfaces of the ice sheet. For the lateral sides, this is equivalent to a calving front, where any ice that flows past the domain boundary is disregarded and assumed to have calved off. This means that no special consideration is taken for boundaries that extend into the ocean (ice shelves). For the top boundary, this is equivalent to atmospheric pressure being applied to the top of the ice sheet, where the pressure of the atmosphere is considered negligible compared to the internal pressure of the ice. The prognostic free–surface, although available in the Fluidity codebase, is not used in these simulations.

4.3.2.1 Bedrock boundary condition

A no–slip, or Rayleigh drag law is applied to the bottom boundary. In the case of no–slip, the ice sheet is assumed frozen and fastened at the bottom. With the Rayleigh drag law, the ice sheet is allowed to slip.

The drag applied to the ice sheet by the bedrock is a strong controlling factor of the velocity of the ice sheet. The friction coefficient can vary by several orders of magnitude in different regions of the ice sheet, depending on bedrock roughness and basal lubrication [Jay-Allemand et al., 2011]. Recent studies have used inverse methods to determine bedrock friction coefficients from surface velocities [Gillet-Chaulet et al., 2012, Jay-Allemand et al., 2011]. This capability is not available in Fluidity, we therefore use a constant friction coefficient for the ice sheet.

The effects of isostatic compensation are also neglected, i.e. the bedrock hypsometry is static and rigid throughout the simulations.

4.3.3 Temperature

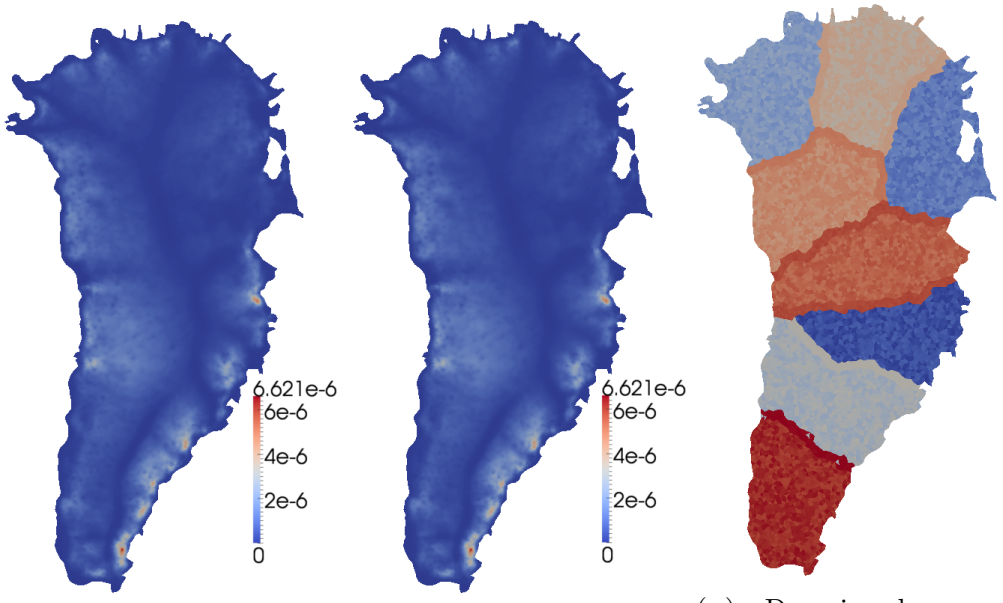
The temperature is initialised from the netCDF data containing the spun–up data from SICOPOLIS. As we attempt to establish the velocity field of the current state of the ice sheet, we do not solve for temperature evolution during our simulations i.e. it is kept fixed. There is also, therefore, no geothermal heat flux specified at the bottom boundary, where the temperature is prescribed directly from the initialisation dataset. The temperature field is important in our simulations, however, as it affects the non–linear viscosity term through the temperature dependent deformation rate factor.

4.4 Model runs

The simulations began by running scenarios with a flat top surface and a bedrock topography of progressively higher granularity. This was achieved by progressively lowering the stencil of the Gaussian kernel in the filtered topography. These simulations were necessary in order to investigate whether there was a limit to the detail of the topography that could be represented in the domain before the solvers were unable to solve the matrices. The results are visually and glaciologically uninteresting as the ice sheet remains primarily static, apart from ice at the lateral boundaries which is forced to flow outwards. It is, however, a useful step in building up to the simulation of Greenland using a realistic domain.

Following the simulations with a flat top, a realistic top surface was introduced and simulations were run on the realistic meshes. These meshes were illustrated in figure 4.1. The results are presented in figure 4.4c. The results are visually indistinguishable, and the maximum value of velocity magnitude

are the same. The simulations were run with a no–slip boundary condition at the bedrock and on the fixed, uniform mesh with resolution of $10km$.



(A) Serial. (B) Parallel. (C) Domain decomposition for parallel run.

FIGURE 4.4: Comparison of serial (left) and parallel (center) runs, coloured by the magnitude of the computed ice sheet velocity. The image on the right shows the domain decomposition of the parallel run into 8 subdomains.

Following the comparison of serial and parallel runs, we proceed to run simulations of Greenland, with both fixed, uniform resolution meshes as well as with mesh adaptivity.

The fixed mesh simulations were performed with various Rayleigh friction drag coefficients. One such result is shown in figure 4.5. As was mentioned earlier, the use of a constant, uniform bottom friction coefficient is unrealistic and highly idealised. Studies have previously shown that simple parameterisations of basal friction of ice sheets do not produce realistic velocity fields [Schäfer et al., 2012]. We are therefore not able to perform a reasonable quantitative comparison of the obtained results with field measured data. However, we are able to comment on the trends of the solution. Considering figure 4.5 we are able to capture the Greenland discharge network, where the ice streams (high velocity flows shown with a red colouring) are prevalent. These are very encouraging first results as the flow pattern is mostly accurate and all the

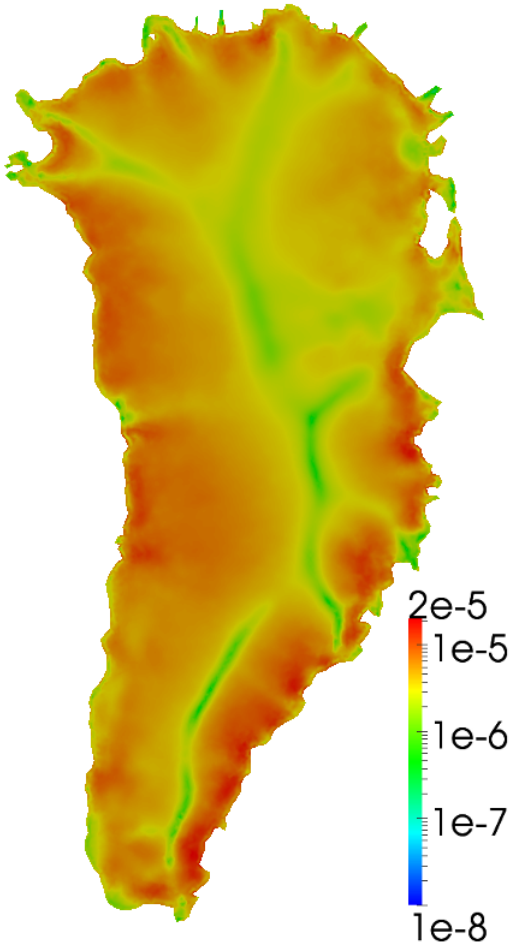


FIGURE 4.5: Preliminary result of Fluidity on fixed, uniform, $10km$ resolution mesh. The figure is coloured by the magnitude of velocity, with a logarithmic scale (in units of $[m\ s^{-1}]$). Linear Rayleigh friction applied to the bottom boundary with friction coefficient 2×10^8.

major features of Greenland are captured. The results show that reducing basal lubrication (by increasing the bedrock friction coefficient) results in the ice sheet flowing more slowly. Initial results also potentially suggest that at these resolutions, the solution is mesh dependant. This is due to the $20km$ resolution result varying from the $10km$ resolution. However, we can not state conclusively that the solution is not mesh–converged, as further analysis would be required. This could be performed by also running on the $5km$ resolution meshes.

Finally, in an attempt to capture the dynamics of Greenland more accurately, adaptive simulations were performed. A mesh of the simulation, after an adapt, is shown in figure 4.6. We emphasise that the mesh shown in figure 4.6

was obtained solely by Fluidity's adaptivity library and based on the solution after a few iterations on a uniform, $10km$ resolution mesh. Many of the interesting regions of Greenland were captured and are now highly resolved, whilst no user interaction was necessary to do so, and no prior knowledge of where these regions may be.

Labelled on the mesh, are some selected Greenland glaciers. In particular, we note that the Jackobshavn Isbrae (glacier), on the west of the domain marked with the initials JG, one of Greenland's fastest flowing outlet glaciers and responsible for up to 6.5% of Greenland drainage to the ocean [Joughin et al., 2004], has been identified as a region of interest and has therefore an increased node density. East of JG, two other glaciers are noted. First, the Helheim glacier (marked HG), which was subject to scientific interest in recent years due to it's rapid acceleration [Joughin et al., 2008] and also the Kangerlugssuaq glacier (marked KG) have had resolution increased in their vicinity.

On the north eastern front, the Zacharia Isstrom (ice stream, marked ZI) is not particularly well captured. Although we notice a small increase in node density near the edge of the front, the high velocity ice stream is not captured, as is evident in figure 4.5, where the ice sheet seems to slow down, rather than accelerate and form an ice stream. This is similarly the case for the icefjord Nioghalvfjersbrae (marked N). Although the reasons for this should be further investigated, a possible explanation is that the surface and bedrock hypsometry necessary for these two regions to be resolved, may have been filtered and smoothed in the initial, $10km$ resolution mesh. In addition, it may be that the friction coefficient used for the drag on the ice base is too high to capture the ice stream.

4.5 Conclusions

Fluidity has been applied to idealised simulations of Greenland in a realistic domain. Although further development is required before a realistic simulation

can be undertaken with Fluidity (see section 5.1), several key model features have been presented that can improve the state of current ice sheet modelling efforts. These features include:

- The use of unstructured meshes that can better represent bedrock topography and surface elevation, as well as the lateral boundaries of the ice sheet.
- Dynamically adaptive meshing technology, which increases node density in dynamically relevant areas, allowing for accurate, efficient simulations across multiple scales.

Ice streams are narrow bands of ice within an ice sheet that flow much faster than their surroundings. For most (but not all) ice sheets, ice streams are responsible for discharging the greatest amount of ice and sediment from within. Their occurrence, but also their spatial and temporal stability, is central to the dynamic behaviour of past, present and future ice sheets [Bennett, 2003]. Accurately resolving these critical ice streams requires resolutions higher than those required for the rest of the domain, a problem to which unstructured meshes offer the greatest benefit. Furthermore, as many significant ice streams are not constrained by topographic features (these ice streams are also called *pure* ice streams in the literature) predicting their location prior to a simulation is difficult. The ability, offered by dynamic adaptivity, to increase resolution in the locality of ice streams whose location is unknown, is also potentially significant. Pure ice streams, which are often the largest contributors to ice and sediment flux, usually exhibit patterns of instability, by either shifting location (spatial) or undergoing cycles of flow activity (temporal). It is with these dynamical processes in mind that we emphasise the potential benefits of dynamic adaptivity to long–term ice sheet modelling.

One overlying question remains: Is Fluidity ready to be used for ice sheet modelling from a glaciologists perspective? To answer this, a balanced overview of the Fluidity model is required. Chapters 3 and 4 have presented a preliminary

application of the Fluidity framework for modelling ice sheets, which has the ability to generate and solve on realistic meshes. Its solvers are capable of running in parallel and solving the matrices that are obtained from discretising continental scale glacial ice, whilst adapting the domain to ensure accuracy and computational efficiency. The ability of unstructured, adaptive meshes to accurately resolve ice streams has been demonstrated, and an overview of the potential utility of such technology in large scale, long–term ice sheet modelling has been outlined. However, several drawbacks hinder the confident adoption of Fluidity as an ice sheet model. The Fluidity model solves the full–Stokes equations, which although have been shown to be the most accurate in determining the flow of ice sheets, are also the most computationally expensive. For ice sheet scale simulations where the dynamics of interest occur at time scales of a full glacial period (thousands of years), a model that solves an approximation of the full–Stokes is likely to be more appropriate. Fluidity is still in its early stages of development as an ice sheet model, and thus lacks some other important features for ice sheet modelling. Namely, the choices for lateral and basal boundary conditions are limited. For the lateral boundaries, the ice sheet can either be considered constrained by a wall (with either a fixed or friction boundary condition) or it can be considered as a calving–front. In either case, Fluidity does not currently have the ability for lateral boundaries to grow and retreat in response to forcing. Finally, Fluidity does not have any inversion methods by which friction coefficients can be determined by sensitivity studies. A technique that is becoming increasingly popular in ice sheet modelling lately, is to use satellite inferred surface velocities to determine the basal friction coefficients. This technique is not currently available in Fluidity, where either a rough approximation of basal friction has to be used, or the values of basal friction need to be obtained from a separate source. An overview of the work required for addressing these drawbacks of Fluidity is presented in section 5.1, which details the future work of this development work.

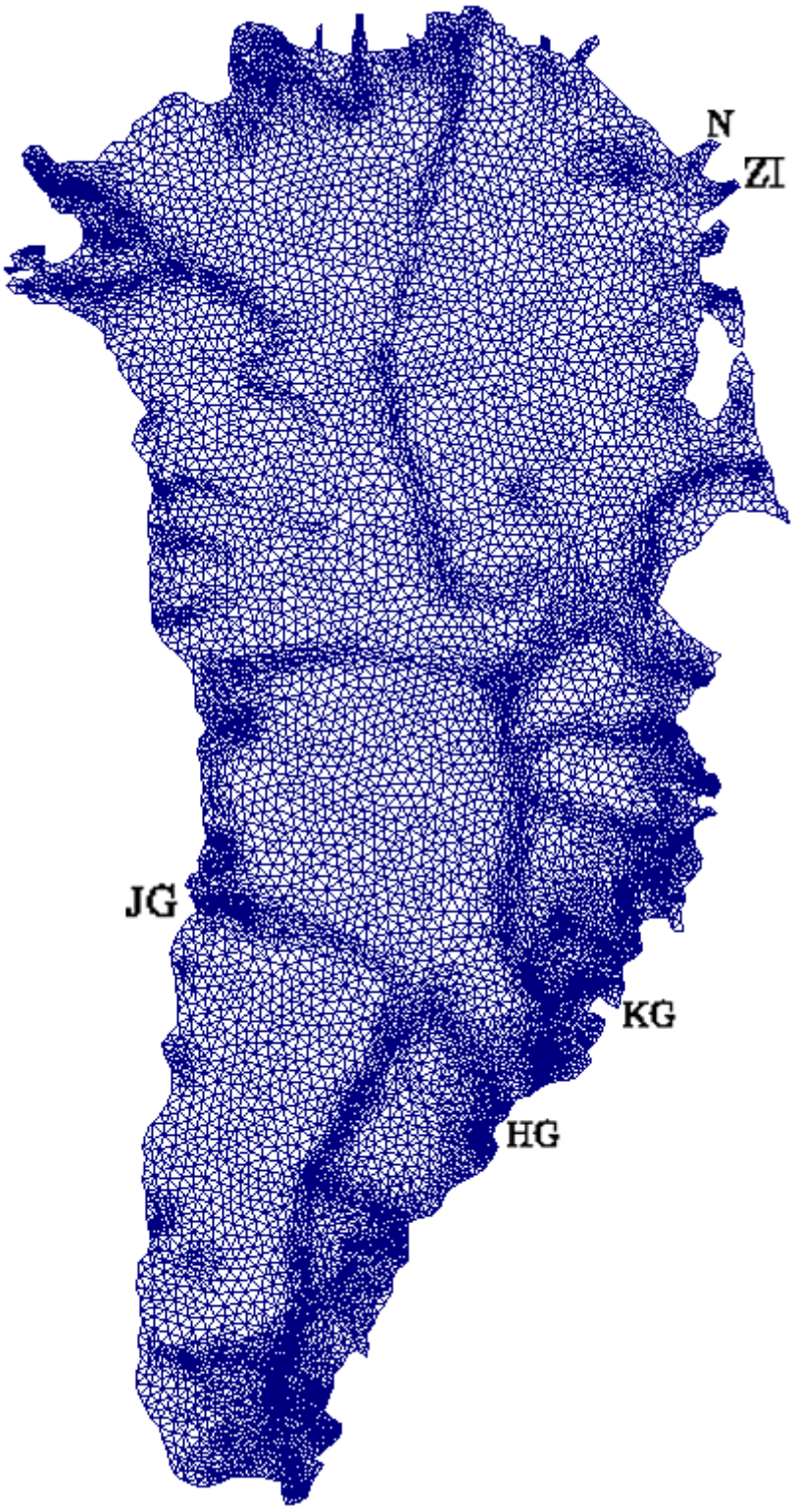


FIGURE 4.6: Mesh of Greenland after an adapt of the mesh in Fluidity. The pre–adapt mesh was a uniform, 10km resolution mesh, as illustrated in figure 4.1. Labelled are some notable glaciers of the Greenland ice sheet. Clockwise from the topmost label: N, Nioghalvfjersbrae; ZI, Zacharia Isbrae; KG, Kangerlugssuaq glacier; HG, Helheim Gletscher; JG, Jackobshavn isbrae.

CHAPTER 5

CONCLUSIONS

This thesis has described the development of coupling strategies for the ocean – sea ice Fluidity–FESIM model (chapter 2). The development process first involved a feasibility study where the use of the OASIS coupler was investigated by coupling to festa, a simple FE model, prior to a prototype version of Fluidity coupled using OASIS via an automatically generated exchange mesh (section 2.2.4.3). Finally, Fluidity was coupled to FESIM using a bespoke architecture, where the coupling subroutines were developed in–house.

The coupled model was then subjected to a series of verification and validation tests, in which successful second–order convergence was shown using the method of manufactured solutions (section 2.4.1) and good agreement with other models was found in a pseudo one–dimensional stress test (section 2.4.2). Finally, in a test of the interaction of sea ice with currents in regions of differing sea ice concentrations, we demonstrate the realistic behaviour of the coupled model. The development of the coupled model thus far, has laid the foundations for further testing and development to produce a model that is potentially as good as current state–of–the–art unstructured models.

In chapter 3 we describe and demonstrate Fluidity's ability to solve the full–Stokes equations with non–linear viscosity, as applied to the modelling of ice sheets and glaciers. The linearised and non–linear equations are shown to converge at the correct order (section 3.3.1) and a simulation of an idealised ice–stream (section 3.4) is used to demonstrate the usefulness of anisotropic adaptivity in maintaining the accuracy of a simulation while substantially decreasing the number of degrees of freedom and hence computational cost. Fluidity's results of performing a series of exercises taken from the ice sheet modelling intercomparison project (ISMIP–HOM) were also presented (section 3.6), showing good agreement with established ice sheet models. Idealised simulations in realistic domains of Greenland are carried out in chapter 4. We demonstrate the feasibility for Fluidity to undertake continental scale ice sheet simulations and produce reasonable results where the trends of observed surface velocities are in agreement.

Fluidity is very different to most current ice sheet models, as those models typically employ lower order approximations of the Stokes and are solved on structured grids. Although the full Stokes system is harder to solve, and thus computationally more expensive, Fluidity's dynamic, anisotropic mesh adaptivity can potentially go a long way to offset these costs.

5.1 Future Work

5.1.1 Sea ice modelling

The coupling interface of the ocean – sea ice coupled Fluidity–FESIM model is fully developed, yet the coupled model lacks the *refinement* of an operational coupled model.

In particular, only one map projection is applied, restricting the meshes that can be used for the ocean model to a three–dimensional domain on the sphere. A potentially useful extension would include a generalisation of the sea ice

model mesh definition algorithm to include ocean domains on flat surfaces, by implementing other spherical projections.

Furthermore, the sea ice thermodynamics of the model remain largely untested. As part of developing a coupled ocean – sea ice model, it is required for the thermodynamics modules, including heat fluxes between the ocean and sea ice, to be tested and validated. Only then can Fluidity–FESIM be released to the scientific community for simulations that may lead to understanding of coupled ocean – sea ice dynamics, or forecast studies.

5.1.2 Ice sheet modelling

The model described in chapter 4 uses a temperature–dependent viscosity term. However, the temperature field is prescribed, rather than prognostic, and thus heat is not advected or diffused across the domain. For the steady–state simulations undertaken in this thesis, this is an acceptable compromise. It is also a reasonable compromise for forecast studies on decadal scales, as the temperature does not change rapidly enough to have significant impact on the flowfield in a few decades, and the numerical cost of solving for temperature advection is not always justified. Longer forecast studies, however, on the scale of centuries and longer, require a prognostic temperature field and geothermal heat fluxes specified at the bedrock. Although Fluidity has the ability to implement a prognostic temperature field, the method would require validating first.

Furthermore, all testcases demonstrated in this thesis have been with a static, non–moving free surface. Future work should concentrate on testing the implicit, prognostic free–surface methods described in Kramer et al. [2012], already implemented in Fluidity.

The flow of ice sheets is strongly controlled by the boundary conditions being applied. Correctly estimating bottom friction coefficients is therefore a crucial component of successful studies of Greenland. One method recently

used for estimating bottom friction coefficients involves an inversion study, where an optimisation method is used to calculate the bottom friction coefficients [Gillet-Chaulet et al., 2012, Jay-Allemand et al., 2011]. Currently, Fluidity does not have the ability to perform these inversion methods, and thus only simplistic approximations of bottom friction coefficients can be used.

Further to correctly measuring the friction coefficients, various bottom friction laws can be used to model the sliding of ice sheets. Currently only three options are available in Fluidity: a) no–slip boundary condition where the ice is fastened to the ground, b) linear Rayleigh friction law, which was applied to all simulations described in this thesis that required a friction law and c) quadratic Rayleigh friction law. A Weertman type friction law can also be implemented and compared to other drag laws. A Weertman type friction law is given in the form:

$$\tau = C u^{m-1} \cdot u, \tag{5.1}$$

with $m = \frac{1}{3}$ being the Weertman exponent. This friction law is commonly used for modelling the flow of ice sheets and glaciers.

BIBLIOGRAPHY

Imperial College London AMCG. Fluidity manual v4.1.11. 04 2014. URL http://dx.doi.org/10.6084/m9.figshare.995912.

G. Ballard, V. Toniolo, D. G. Ainley, C. L. Parkinson, K. R. Arrigo, and P. N. Trathan. Responding to climate change: Adélie Penguins confront astronomical and ocean boundaries. *Ecology*, 91(7):2056–2069, Jul 2010.

Matthew R. Bennett. Ice streams as the arteries of an ice sheet: their mechanics, stability and significance. *Earth-Science Reviews*, 61(3–4):309 – 339, 2003. ISSN 0012-8252. doi: http://dx.doi.org/10.1016/S0012-8252(02)00130-7. URL http://www.sciencedirect.com/science/article/pii/S0012825202001307.

H. Blatter. Velocity and stress fields in grounded glaciers: a simple algorithm for including deviatoric stress gradients. *Journal of Glaciology*, 41:333–344, 1995.

Sylvain Bouillon, Miguel Ángel Morales Maqueda, Vincent Legat, and Thierry Fichefet. An elastic–viscous–plastic sea ice model formulated on arakawa b and c grids. *Ocean Modelling*, 27(3–4):174 – 184, 2009. ISSN 1463-5003. doi: http://dx.doi.org/10.1016/j.ocemod.2009.01.004. URL http://www.sciencedirect.com/science/article/pii/S1463500309000043.

A.S. Candy. Shingle: Generation of boundary representation from arbitrary geophysical fields. Unpublished, 2014. URL http://shingleproject.org/.

W. M. Connolley, J. M. Gregory, E. Hunke, and A. J. McLaren. On the consistent scaling of terms in the sea-ice dynamics equation. *J. Phys. Oceanogr.*, 34(7):1776–1780, July 2004. ISSN 0022-3670. URL `http://dx.doi.org/10.1175/1520-0485(2004)034<1776:OTCSOT>2.0.CO;2`.

Stephen L. Cornford, 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(1):529 – 549, 2013. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/j.jcp.2012.08.037. URL `http://www.sciencedirect.com/science/article/pii/S0021999112005050`.

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

D. Rhodri Davies, Cian R. Wilson, and Stephan C. Kramer. Fluidity: A fully unstructured anisotropic adaptive mesh computational modeling framework for geodynamics. *Geochemistry, Geophysics, Geosystems*, 12(6):n/a–n/a, 2011. ISSN 1525-2027. doi: 10.1029/2011GC003551. URL `http://dx.doi.org/10.1029/2011GC003551`.

George M. Durner, David C. Douglas, Ryan M. Nielson, Steven C. Amstrup, Trent L. McDonald, Ian Stirling, Mette Mauritzen, Erik W. Born, Øystein Wiig, Eric DeWeaver, Mark C. Serreze, Stanislav E. Belikov, Marika M. Holland, James Maslanik, Jon Aars, David A. Bailey, and Andrew E. Derocher. Predicting 21st-century polar bear habitat distribution from global climate models. *Ecological Monographs*, 79(1):25–58, February 2009. ISSN 0012-9615. URL `http://dx.doi.org/10.1890/07-2089.1`.

The Economist. Special report: The melting north, June 2002. URL `http://www.economist.com/node/21556798`.

Patrick E. Farrell. *Galerkin projection of discrete fields via supermesh construction*. PhD thesis, Imperial College London, 2009.

P.E. Farrell, M.D. Piggott, C.C. Pain, G.J. Gorman, and C.R. Wilson. Conservative interpolation between unstructured meshes via supermesh construction. *Computer Methods in Applied Mechanics and Engineering*, 198 (33–36):2632 – 2642, 2009. ISSN 0045-7825. doi: http://dx.doi.org/10.

1016/j.cma.2009.03.004. URL http://www.sciencedirect.com/science/article/pii/S0045782509001315.

F. Fetterer, K. Knowles, W. Meier, and M. Savoie. Sea Ice Index, ASINA, National Snow and Ice Data Center, Boulder, Colorado USA, 2002, updated daily. URL http://dx.doi.org/10.7265/N5QJ7F7W.

O. Gagliardini and T. Zwinger. The ismip-hom benchmark experiments performed using the finite-element code elmer. *The Cryosphere*, 2(1):67–76, 2008. doi: 10.5194/tc-2-67-2008. URL http://www.the-cryosphere.net/2/67/2008/.

Guoping Gao, Changsheng Chen, Jianhua Qi, and Robert C. Beardsley. An unstructured-grid, finite-volume sea ice model: Development, validation, and application. *Journal of Geophysical Research: Oceans*, 116(C8):n/a–n/a, 2011. ISSN 2156-2202. doi: 10.1029/2010JC006688. URL http://dx.doi.org/10.1029/2010JC006688.

Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009. ISSN 1097-0207. doi: 10.1002/nme.2579. URL http://dx.doi.org/10.1002/nme.2579.

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

J.M Gregory and P Huybrechts. Ice-sheet contributions to future sea-level change. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 364(1844):1709–1732, 2006. doi: 10.1098/rsta.2006.1796. URL http://rsta.royalsocietypublishing.org/content/364/1844/1709.abstract.

Ralf Greve. Application of a polythermal three-dimensional ice sheet model to the greenland ice sheet: Response to steady-state and transient climate scenarios. *J. Climate*, 10(5):901–918, May 1997. ISSN 0894-8755. URL http://dx.doi.org/10.1175/1520-0442(1997)010<0901:AOAPTD>2.0.CO;2.

W. D. Hibler. A dynamic thermodynamic sea ice model. *Journal of Physical Oceanography*, 9(4):815–846, 2014/08/09 1979. doi: 10.1175/1520-0485(1979)009⟨0815:ADTSIM⟩2.0.CO;2. URL http://dx.doi.org/10.1175/1520-0485(1979)009<0815:ADTSIM>2.0.CO;2.

Hannah R. Hiester. *The application of adaptive mesh techniques to numerical simulations of gravity current flows.* PhD thesis, Imperial College London, 2011.

R. C. A. Hindmarsh. A numerical comparison of approximations to the stokes equations used in ice sheet and glacier modeling. *Journal of Geophysical Research: Earth Surface*, 109(F1):n/a–n/a, 2004. ISSN 2156-2202. doi: 10.1029/2003JF000065. URL http://dx.doi.org/10.1029/2003JF000065.

HPC. Imperial college high performance computing service. Unpublished. URL http://www.imperial.ac.uk/ict/services/teachingandresearchservices/highperformancecomputing.

E. C. Hunke and J. K. Dukowicz. An elastic–viscous–plastic model for sea ice dynamics. *Journal of Physical Oceanography*, 27(9):1849–1867, 2014/08/09 1997. doi: 10.1175/1520-0485(1997)027⟨1849:AEVPMF⟩2.0.CO;2. URL http://dx.doi.org/10.1175/1520-0485(1997)027<1849:AEVPMF>2.0.CO;2.

E. C. Hunke and W. H. Lipscomb. CICE: The Los Alamos Sea Ice Model, Documentation and Software User's Manual, Version 4.1. Technical Report LA-CC- 06-012, Los Alamos National Laboratory, Los Alamos, New Mexico, 2010. available at http://oceans11.lanl.gov/trac/CICE (last access: 6 February 2014).

E. C. Hunke and Y. Zhang. A comparison of sea ice dynamics models at high resolution. *Mon. Wea. Rev.*, 127(3):396–408, March 1999. ISSN 0027-0644. URL http://dx.doi.org/10.1175/1520-0493(1999)127<0396:ACOSID>2.0.CO;2.

IPCC. *Climate Change 2007: The Physical Science Basis. Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change.* Cambridge University Press, Cambridge, United Kingdom and New York, USA, 2007.

M. Jay-Allemand, F. Gillet-Chaulet, O. Gagliardini, and M. Nodet. Investigating changes in basal conditions of variegated glacier prior to and during its 1982–1983 surge. *The Cryosphere*, 5(3):659–672, 2011. doi: 10.5194/tc-5-659-2011. URL http://www.the-cryosphere.net/5/659/2011/.

Ian Joughin, Waleed Abdalati, and Mark Fahnestock. Large fluctuations in speed on greenland's jakobshavn isbrae glacier. *Nature*, 432(7017):608–610, December 2004. ISSN 0028-0836. URL http://dx.doi.org/10.1038/nature03130.

Ian Joughin, Ian Howat, Richard B. Alley, Goran Ekstrom, Mark Fahnestock, Twila Moon, Meredith Nettles, Martin Truffer, and Victor C. Tsai. Ice-front variation and tidewater behavior on helheim and kangerdlugssuaq glaciers, greenland. *Journal of Geophysical Research: Earth Surface*, 113(F1):n/a–n/a, 2008. ISSN 2156-2202. doi: 10.1029/2007JF000837. URL http://dx.doi.org/10.1029/2007JF000837.

Stephan C. Kramer, Cian R. Wilson, and D. Rhodri Davies. An implicit free surface algorithm for geodynamical simulations. *Physics of the Earth and Planetary Interiors*, 194–195(0):25 – 37, 2012. ISSN 0031-9201. doi: http://dx.doi.org/10.1016/j.pepi.2012.01.001. URL http://www.sciencedirect.com/science/article/pii/S0031920112000027.

Martin Kreyscher, Markus Harder, Peter Lemke, and Gregory M. Flato. Results of the sea ice model intercomparison project: Evaluation of sea ice rheology schemes for use in climate simulations. *Journal of Geophysical Research: Oceans*, 105(C5):11299–11320, 2000. ISSN 2156-2202. doi: 10.1029/1999JC000016. URL http://dx.doi.org/10.1029/1999JC000016.

Rainald Löhner, Ken Morgan, Jaime Peraire, and Mehdi Vahdati. Finite element flux-corrected transport (fem–fct) for the euler and navier–stokes equations. *International Journal for Numerical Methods in Fluids*, 7(10):1093–1109, 1987. ISSN 1097-0363. doi: 10.1002/fld.1650071007. URL http://dx.doi.org/10.1002/fld.1650071007.

Dave A. May and Louis Moresi. Preconditioned iterative methods for stokes flow problems arising in computational geodynamics. *Physics of the Earth and Planetary Interiors*, 171(1–4):33 – 47, 2008. ISSN 0031-9201. doi: http://dx.doi.org/10.1016/j.pepi.2008.07.036. URL http://www.

sciencedirect.com/science/article/pii/S003192010800191X. Recent Advances in Computational Geodynamics: Theory, Numerics and Applications.

Emmanuel Le Meur, Olivier Gagliardini, Thomas Zwinger, and Juha Ruokolainen. Glacier flow modelling: a comparison of the shallow ice approximation and the full-stokes solution. *Comptes Rendus Physique*, 5(7): 709 – 722, 2004. ISSN 1631-0705. doi: http://dx.doi.org/10.1016/j.crhy.2004.10.001. URL http://www.sciencedirect.com/science/article/pii/S1631070504001732. Ice: from dislocations to icy satellites.

Josef M. Oberhuber. Simulation of the atlantic circulation with a coupled sea ice-mixed layer-isopycnal general circulation model. part i: Model description. *Journal of Physical Oceanography*, 23(5):808–829, 2014/08/09 1993. doi: 10.1175/1520-0485(1993)023⟨0808:SOTACW⟩2.0.CO;2. URL http://dx.doi.org/10.1175/1520-0485(1993)023<0808:SOTACW>2.0.CO;2.

C.C. Pain, A.P. Umpleby, C.R.E. de Oliveira, and A.J.H. Goddard. Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations. *Computer Methods in Applied Mechanics and Engineering*, 190(29–30):3771 – 3796, 2001. ISSN 0045-7825. doi: http://dx.doi.org/10.1016/S0045-7825(00)00294-2. URL http://www.sciencedirect.com/science/article/pii/S0045782500002942.

Claire L. Parkinson and Warren M. Washington. A large-scale numerical model of sea ice. *Journal of Geophysical Research: Oceans*, 84(C1):311–337, 1979. ISSN 2156-2202. doi: 10.1029/JC084iC01p00311. URL http://dx.doi.org/10.1029/JC084iC01p00311.

F. Pattyn and T. Payne. Ismip–hom, benchmark experiments for numerical higher-order ice-sheet models. February 2006. URL http://homepages.ulb.ac.be/~fpattyn/ismip/ismiphom.pdf.

F. Pattyn, L. Perichon, A. Aschwanden, B. Breuer, B. de Smedt, O. Gagliardini, G. H. Gudmundsson, R. C. A. Hindmarsh, A. Hubbard, J. V. Johnson, T. Kleiner, Y. Konovalov, C. Martin, A. J. Payne, D. Pollard, S. Price, M. Rückamp, F. Saito, O. Souček, S. Sugiyama, and T. Zwinger. Benchmark experiments for higher-order and full-stokes ice sheet models (ISMIP–HOM). *The Cryosphere*, 2(2):95–108, 2008. doi: 10.5194/tc-2-95-2008. URL http://www.the-cryosphere.net/2/95/2008/.

Frank Pattyn. A new three-dimensional higher-order thermomechanical ice sheet model: Basic sensitivity, ice stream development, and ice flow across subglacial lakes. *Journal of Geophysical Research: Solid Earth*, 108(B8): n/a–n/a, 2003. ISSN 2156-2202. doi: 10.1029/2002JB002329. URL `http://dx.doi.org/10.1029/2002JB002329`.

M. D. Piggott, P. E. Farrell, C. R. Wilson, G. J. Gorman, and C. C. Pain. Anisotropic mesh adaptivity for multi-scale ocean modelling. *Philosophical Transactions of the Royal Society A: Mathematical,Physical and Engineering Sciences*, 367(1907):4591–4611, 2009. doi: 10.1098/rsta.2009.0155. URL `http://rsta.royalsocietypublishing.org/content/367/1907/4591.abstract`.

Patrick J. Roache. Code verification by the method of manufactured solutions. *Journal of Fluids Engineering*, 124(1):4–10, 11 2001. URL `http://dx.doi.org/10.1115/1.1436090`.

P.J. Roache. *Verification and Validation in Computational Science and Engineering*. Hermosa, 1998. ISBN 9780913478080. URL `http://books.google.co.uk/books?id=ENRlQgAACAAJ`.

M. Schäfer, T. Zwinger, P. Christoffersen, F. Gillet-Chaulet, K. Laakso, R. Pettersson, V. A. Pohjola, T. Strozzi, and J. C. Moore. Sensitivity of basal conditions in an inverse model: Vestfonna ice cap, nordaustlandet/svalbard. *The Cryosphere*, 6(4):771–783, 2012. doi: 10.5194/tc-6-771-2012. URL `http://www.the-cryosphere.net/6/771/2012/`.

Christian Schoof and Richard C. A. Hindmarsh. Thin-film flows with wall slip: An asymptotic analysis of higher order glacier flow models. *The Quarterly Journal of Mechanics and Applied Mathematics*, 2010. doi: 10.1093/qjmam/hbp025. URL `http://qjmam.oxfordjournals.org/content/early/2010/01/25/qjmam.hbp025.abstract`.

Hakime Seddik, Ralf Greve, Thomas Zwinger, Fabien Gillet-Chaulet, and Olivier Gagliardini. Simulations of the greenland ice sheet 100 years into the future with the full stokes model elmer/ice. *Journal of Glaciology*, 58(209):427–440, 2012. doi: doi:10.3189/2012JoG11J177. URL `http://www.ingentaconnect.com/content/igsoc/jog/2012/00000058/00000209/art00001`.

Albert J. Semtner. A model for the thermodynamic growth of sea ice in numerical investigations of climate. *Journal of Physical Oceanography*, 6(3):379–389, 2014/08/09 1976. doi: 10.1175/1520-0485(1976)006⟨0379:AMFTTG⟩2.0.CO;2. URL http://dx.doi.org/10.1175/1520-0485(1976)006<0379:AMFTTG>2.0.CO;2.

Julienne Stroeve, Marika M. Holland, Walt Meier, Ted Scambos, and Mark Serreze. Arctic sea ice decline: Faster than forecast. *Geophysical Research Letters*, 34(9):n/a–n/a, 2007. ISSN 1944-8007. doi: 10.1029/2007GL029703. URL http://dx.doi.org/10.1029/2007GL029703.

Ralph Timmermann, Sergey Danilov, Jens Schröter, Carmen Böning, Dmitry Sidorenko, and Katja Rollenhagen. Ocean circulation and sea ice distribution in a finite element global sea ice–ocean model. *Ocean Modelling*, 27 (3–4):114 – 129, 2009. ISSN 1463-5003. doi: http://dx.doi.org/10.1016/j.ocemod.2008.10.009. URL http://www.sciencedirect.com/science/article/pii/S1463500308001571.

Ralph Timmermann, Anne M Le Brocq, Tara J Deen, Eugene W Domack, Pierre Dutrieux, Ben Galton-Fenzi, Hartmut H Hellmer, Angelika Humbert, Daniela Jansen, Adrian Jenkins, Astrid Lambrecht, Keith Makinson, Fred Niederjasper, Frank-Oliver Nitsche, Ole Anders N?st, Lars Henrik Smedsrud, and Walter Smith. Antarctic ice sheet topography, cavity geometry, and global bathymetry (RTopo 1.0.5-beta), 2010. URL http://doi.pangaea.de/10.1594/PANGAEA.741917. Supplement to: Timmermann, Ralph; Le Brocq, Anne M; Deen, Tara J; Domack, Eugene W; Dutrieux, Pierre; Galton-Fenzi, Ben; Hellmer, Hartmut H; Humbert, Angelika; Jansen, Daniela; Jenkins, Adrian; Lambrecht, Astrid; Makinson, Keith; Niederjasper, Fred; Nitsche, Frank-Oliver; N?st, Ole Anders; Smedsrud, Lars Henrik; Smith, Walter (2010): A consistent dataset of Antarctic ice sheet topography, cavity geometry, and global bathymetry. Earth System Science Data, 2(2), 261-273, doi:10.5194/essd-2-261-2010.

S. Valcke. The oasis3 coupler: a european climate modelling community software. *Geoscientific Model Development*, 6(2):373–388, 2013. doi: 10.5194/gmd-6-373-2013. URL http://www.geosci-model-dev.net/6/373/2013/.

Martin Vancoppenolle, Thierry Fichefet, Hugues Goosse, Sylvain Bouillon, Gurvan Madec, and Miguel Angel Morales Maqueda. Simulating the mass balance and salinity of arctic and antarctic sea ice. 1. model description and validation. *Ocean Modelling*, 27(1–2):33 – 53, 2009. ISSN 1463-5003. doi: http://dx.doi.org/10.1016/j.ocemod.2008.10.005. URL `http://www.sciencedirect.com/science/article/pii/S1463500308001613`.

Y.Vasilevskii and K.Lipnikov. An adaptive algorithm for quasioptimal mesh generation. *Computational Mathematics and Mathematical Physics*, 39: 1468–1486, 1999.

Xiangdong Zhang and John E. Walsh. Toward a seasonally ice-covered arctic ocean: Scenarios from the ipcc ar4 model simulations. *Journal of Climate*, 19(9):1730–1747, 2014/08/05 2006. doi: 10.1175/JCLI3767.1. URL `http://dx.doi.org/10.1175/JCLI3767.1`.

APPENDIX A

STRUCTURED GRID GENERATION

In the OASIS configuration file (CML file), information regarding the resolution (number of elements in each direction, `num_x, num_y`) and extent (minimum and maximum) of longitude and latitude (`lon_min, lon_max, lat_min, lat_max`) of the coupled domain are requested from the user. This information is enough to fully define a structured grid and thus we eliminate the need for the user to supply a structured mesh generated in gmsh. Instead, the structured grid is generated automatically by Fluidity. This is done in the `gen_def_grid` subroutine of `<trunk>/main/test_coupler.F90` of the coupling development branch.

The algorithm is outlined and explained below:

1. Create Q1 element shape.

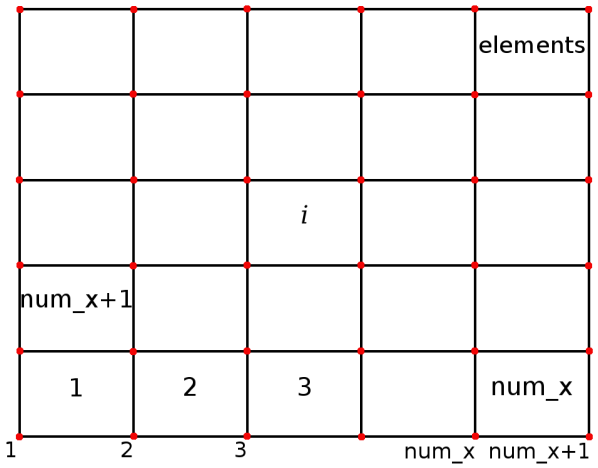
   Each element is a quadrilateral (4 vertices, 2 dimensions) with linear (degree=1) shape functions. The `quadrature_type` associated with this element is thus also to be accurate for degree one basis functions.

   ```
   shape = make_element_shape(4,2,1,quadrature, stat=stat)
   ```

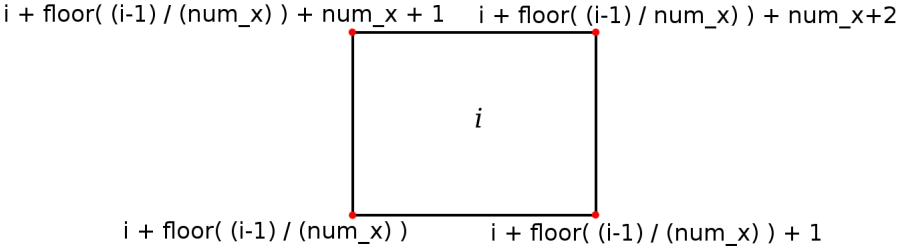
2. Allocate Q1 mesh.

   From the extent information, the number of nodes and number of elements is known:

   ```
   nodes = (num_lon+1) * (num_lat+1)
   elements = num_lon * num_lat
   ```



(A) Element and node numbering convention.



(B) Node numbering for i^{th} element.

FIGURE A.1: Element and node numbering convention for automatically generated structured grid.

Along with the shape generated in the previous step, there is enough information to allocate a `mesh_type`. In steps 3 and 4 we set the connectivity and coordinate locations for the mesh.

3. Set element nodes.

   For each element in the mesh, we need to specify the node numbers that consitute it's vertices. This is done by looping over each element and specifying it's corresponding node numbers. For each element, the corresponding node numbers are as illustrated in Figure A.1b.

   Having set the element nodes, we have specified the mesh topology, but the mesh geometry is yet undefined.

4. Set node locations.

   To set the mesh geometry, the node locations (coordinates) need to be specified. This is done by looping over each node, and setting its coordinates.

   This is done with the following code algorithm:

   - Displacement between nodes is calculated:

```
disp1 = (lon_max - lon_min) / num_x
disp2 = (lat_max - lat_min) / num_x
```

- Bottom left corner longitude value of each node in ascending longitude order is stored in corners_lon(:,1), and bottom left corner latitude value of each node in ascending latitude order is stored in corners_lat(:,1):

```
do i = 1, num_x
   corners_lon(i,1) = lon_min + disp1 * (i-1)
enddo

do i = 1, num_y
   corners_lat(i,1) = lat_min + disp2 * (i-1)
enddo
```

- Bottom right corner longitude value of each node in ascending longitude order is stored in corners_lon(:,2), and top left corner longitude value of each node in ascending latitude order is stored in corners_lat(:,2):

```
corners_lon(:,2) = corners_lon(:,1) + disp1
corners_lat(:,2) = corners_lat(:,1) + disp2
```

- The coordinates of each ith node can therefore be specified by:

```
do i = 1, nodes
   i1 = mod(i,(num_x+1))
   i4 = 1
   if (i1 == 0) then
      i1 = num_x
      i4 = 2
   end if
   i2 = floor((i-1) / (num_x+1) ) + 1
   i5 = 1
   if (i2 == num_y+1) then
      i2 = num_y
      i5 = 2
   end if
   call set(coord,i,(/ corners_lon(i1,i4), corners_lat(i2,i5) /) )
end do
```

5. Extract Q0 mesh from Q1 mesh.

   To generate a Q0 femtools `mesh_type`, the Q0 mesh needs to be extracted from the Q1 mesh (defined above). This is done using:

```
shape = make_element_shape(4,2,degree=0,quadrature, stat=stat)

Mesh_Q0 = make_mesh(Mesh_Q1 , shape=shape, &
  continuity=-1, name=''StructuredMeshQ0'')
```

   where this time we specify a different element shape, that of discontinuous elements (degree = 0).

6. Specify node locations of Q0 mesh, at the center of the elements.

   This is done similarly to step 4, shifted half a displacement, `disp`, such that the new node is located in the center of the element.

APPENDIX B

ICE SHEET MESH GENERATION

B.1 Layered mesh generation

This section describes in more detail the process overviewed in section 3.6.6. It describes a process of producing meshes for ISMIP–HOM experiments A to D. It produces layered meshes and can be used for both two– and three–dimensional meshes.

We first generate the two extrusions, by processing the following script with the Gmsh geometry module:

```
dy=YY;
dx=XX;
Point(1) = {0, 0, 0, 1.0};
Extrude {0, -1, 0} {
  Point{1}; Layers{dy};
}
Extrude {1, 0, 0} {
  Line{1}; Layers{dx};
}
//LEFT
Physical Line(1) = {1};
//RIGHT
Physical Line(2) = {2};
//BOTTOM
Physical Line(6) = {4};
//TOP
Physical Line(8) = {3};
//meshed surface
Physical Surface(10) = {5};
```

where `YY` and `XX` represents the number of layers in the vertical and horizontal directions, respectively. At this stage we also specify the physical boundary IDs that are used in Fluidity to specify the appropriate boundary conditions.

We can then invoke the Gmsh meshing module on the above domain to obtain the mesh illustrated in figure 3.27.

The mesh, now in Gmsh format, is tranformed to the Triangle mesh format using the `gmsh2triangle` tool, available in the Fluidity trunk.

The Triangle file that contains node locations is then read in by a python script that applies a transformation to the node locations. The section of the script that applies the transformation is included:

```
for line in cubefile:
    if counter:
        hold_S = line.strip().split()[0]
        X0,X1 = line.strip().split()[1:3]
        X = float(X0)
        Y = float(X1)
        X = domain_extent*X #Horizontal transformation
        Y = 1000.0 * Y      #Vertical transformation
        Y = -X*tan_alpha + Y - (Y/2.)*sin(omega*X)  # Skew operator
        line = hold_S +' '+str(X) +' '+ str(Y) + '\n'
        counter -= 1
    if search_nodes:
        counter = int(line.strip().split()[0])
        search_nodes = False
        print counter, 'nodes.'
    meshfile.write(line)
```

B.2 Unstructured mesh generation

This method can be used for two–dimensional meshes only, and produces meshes that are unstructured. It involves generating the geometry of the meshed region by looping around the boundary. This method was used for the two dimensional meshes of exercises B, D, as well as the realistic glacier simulation of the long profile of Haut d'Arolla, exercise E.

The method involves generating a geometry representation of the mesh, which can be passed to Gmsh Geuzaine and Remacle [2009] for meshing:

The 2–D ISMIP–HOM experiment B mesh will be used here as an example. We begin by writing points along the sinusoidal bottom boundary:

```
    while x>0:
        writepoint(x, -x*tana - 1000 + 500*math.sin(omega*x), 0.0, id_count)
        id_count += 1
        x -= sep
```

where the `sep` variable holds the point resolution and x is the horizontal displacement. The `writepoint` function writes a point in the Gmsh geometry definition format, and is defined as:

```
point_temp='''
Point(PID) = {XX,YY,ZZ,dd};'''

def write_point(X,Y,Z, pid):
    meshfile.write(point_temp
                   .replace('PID',str(pid))
```

```
                    .replace('XX',str(X))
                    .replace('YY',str(Y))
                    .replace('ZZ',str(Z)))
```

The whole domain can be traced in a similar manner. For a copy of the full scripts used to generate the unstructured meshes for ISMIP–HOM experiments B, D and E, readers are asked to contact the author.

B.3 Topography smoothing for Greenland

The process involves applying a Gaussian filter to the topography of the RTopo [Timmermann et al., 2010] netCDF file. This is applied with the appended script, where the `get_depth` function returns the smoothed depth at the coordinates $[x, y]$. Gaussian filter is applied using the multi–dimensional image processing library `ndimage`.

```
from scipy.io import netcdf
import scipy.interpolate

from scipy import ndimage

nc = netcdf.netcdf_file('grl.nc','r')
ex = nc.variables['xi'][:]
why = nc.variables['eta'][:]
values = nc.variables['zm'][:][:]

filtered_values = ndimage.filters.gaussian_filter(values, 1.0)

max_depth = filtered_values.max()

interpolator = scipy.interpolate.RectBivariateSpline(ex, why, filtered_values)

def get_depth(X):
   return interpolator(X[0], X[1]) - max_depth - 40.0
```