The Big Thaw: Simulating Greenland's Future

John Burkardt
Graduate Student Seminar
Department of Scientific Computing
Florida State University

....

http://people.sc.fsu.edu/∼jburkardt/presentations/...
climate_2011_fsu.pdf

3pm, 08 April 2011

Introduction

In recent decades, a slight but steady increase in average world-wide temperature was noticed. This was attributed to the effects of industrialization, reduction in tree cover, increased burning of fossil fuels, the role of carbon dioxide in absorbing greater amounts of solar energy.

Even if the mechanisms are still just plausible hypotheses, the slight but steady rise in temperature has continued. It is natural to ask the straightforward question:

Is anything important likely to happen to the Earth if the temperature continues to increase?

Introduction

The question being asked is a *modeling question*. In other words, we are not asking *if* the temperature is rising a little each year, or by how much.

We are assuming a certain yearly rise in temperature, and asking if we can produce a plausible climate model which can exhibit the expected results.

Obviously, we would be interested in checking several different estimates for the increase, and if we model the climate, we'd like the actual yearly rise to vary in some statistical way about the estimated average rise.

Modeling the global climate is really hard. While we understand the properties of air, ocean, land and sunshine in general, we have limited understanding of common things such as clouds and of big, slowly moving things (glaciers and ice sheets).

Introduction: Climate is Complicated!

We know that many glaciers around the world have retreated or even melted away. That suggests that we might need to be concerned about the really enormous ice sheets in Greenland and Antarctica.

It would be easy to make a first guess that, if the global temperature is rising, then these sheets also will melt away.

But these easy guesses are worthless, because the Earth's climate is a strangely complicated thing. Initial investigations have already suggested that, at least in the beginning, the overall warming of the Earth would result in increased snowfall in Greenland, so that the ice sheet would get thicker.

Which also means that if we had chosen to look at the thickness of Greenland's ice as an indicator of the climate, we'd wrongly conclude that things are getting better!

Introduction: Don't Argue, Predict!

The important thing to do, then, is to take practical, accurate, scientific models, implement them as well as we can, and use them to make predictions for the near and long-term future.

We can then try to answer the question of whether climate modeling is believable by comparing the short-term predictions to the actual observed behavior over the next decade.

If these predictions are reasonably accurate, then the longer-term predictions (whatever they may be) will be more acceptable.

Failures of prediction are also useful in helping to improve the model (and to restart the prediction process.)

Introduction: Calving from an Ice Sheet

Introduction

It's hard to say whether the melting of a single glacier has happened because the climate has changed, or that this melting could, in turn, have an effect on the climate.

But an ice sheet can be the size of a small continent (Greenland and Antarctica, in particular); the ice can be two miles thick. The Greenland ice alone could raise the ocean 20 feet.

In other words, changes to the Greenland ice sheet are likely to represent important climate effects, and to cause further changes.

The ice in Greenland accumulates and moves slowly towards the ocean. The movement is resisted by a strong frictional force where the ice sheet rests on bedrock. This is usually a "dry" contact, but in some places, geothermal heat overcomes the enormous pressure, and the ice sheet slides quickly over a wet contact.

Introduction: Surface Melt

Introduction: Simulating the Ice Sheets

The goal of a simulation is to take some model of the earth's temperature profile and to simulate the resulting climate over a period of 100 years.

There are already computer programs available which attempt this task, but in order to make any computations, they've had to make numerous short cuts and estimates, and to simply admit that certain parts of the model are not well understood.

Because climate is so complex, many climate models are constructed out of separate programs, each of which models a particular feature. Each program outputs its current status at regular points, and can accept input from other programs or else use "canned" or approximate data in a stand-alone run.

Climate programs have a limited understanding of the ice sheets of Greenland and Antarctica.

Introduction: CESM

For example, there is a system called **CESM** or the Community Earth System Model. It comprises five physical models, implemented as computer programs by separate teams, including:

- **ATM**, atmosphere (clouds, vapor, radiation, pollutants);
- **LND**, land surface;
- **OCN**, ocean;
- **ICE**, sea ice;
- **GLC**, ice sheets (Greenland/Antarctica);

The ice sheet model is a relatively new addition to **CESM**.

http://www.cesm.ucar.edu

Introduction: New Ice Sheet Model

Since 1990, the Intergovernmental Panel on Climate Change or **IPCC**, has issued a report every five to seven years, summarizing the state of knowledge and proposing areas where a better understanding is needed.

The Fourth Report was issued in 2007, and declared that current climate models were not able to properly analyze the continental-sized ice sheets covering Greenland and Antarctica.

The panel complained of:

- insufficient data of topography, snow fall, ice depth, velocity;
- low accuracy models of ice sheet physics;
- insufficient resolution in time and space;
- computer programs unable to interface with programs modeling the ocean and atmosphere;

Introduction: New Ice Sheet Model

One of the most controversial topics in climate modeling involves predictions of the future level of the ocean.

Because the ice sheets lock up an enormous amount of water, a credible prediction of changes in sea level must be based on confidence in the predictions of the ice sheets.

The 2007 panel refused to make a prediction on sea level changes because of the uncertainty over ice sheet behavior.

The Fifth Report will be issued in 2014, and the panel demanded that by that time researchers must produce credible, detailed ice sheet data needed to predict ocean levels.

http://www.ipcc.ch/

Introduction: How CESM Changes are Made and Approved

CESM is one of about 12 separate computer models of the earth's climate developed by various research groups.

CESM is led by researchers at **NCAR**, the National Center for Atmospheric Research, in Boulder, Colorado, which has a long history of work in weather and climate. However, the component programs have come from a variety of universities and organizations, and are supported and developed by somewhat autonomous teams of scientists.

Significant changes to a CESM component program must be approved by a larger group before they are accepted into a new release of that program.

Several teams have assembled to make improvements to Glimmer-CISM, the Community Ice Sheet Model that CESM uses to model Greenland and Antarctica.

Bill Lipscomb, at Los Alamos National Laboratory, is in overall charge of an effort funded by the Department of Energy. He has a team at LANL trying to implement more realistic physical models for the ice sheet; Kate Evans, at Oak Ridge National Laboratory, is directing a team looking at improving the solver. There is another team at Lawrence Berkeley Lab, under Esmond Ng.

Bill Lipscomb decided to seek research and programming support from Max Gunzburger (FSU) and Lili Ju (SC), who have formed a small group consisting of postdoctoral students Mauro Perego, Tao Cui, and Wei Leng, slightly assisted by me.

Introduction: GLIMMER

The **Glimmer** program has been used for over thirty years to model ice sheets.

That's good, because the program has accumulated a lot of knowledge; but it also means that the program still has many features that are becoming brittle and out of date.

It is a large program, written in FORTRAN, uses a rectangular equally-spaced grid, a relatively simple physics model, and an iterative solver that can be slow to converge.

Needless to say, **Glimmer** is not a parallel code.

Introduction: GLIMMER

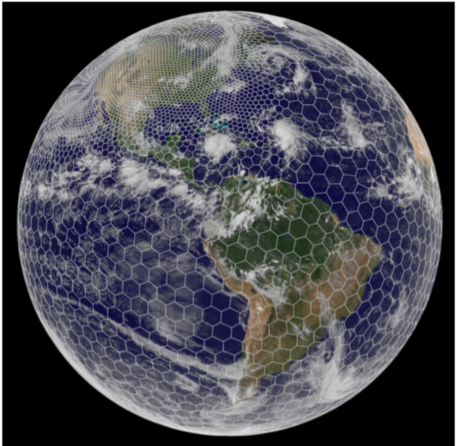
Because of its history and widespread acceptance among ice-sheet modelers, **Glimmer** was chosen to join the CESM system as the "Community Ice Sheet Model". The version of **Glimmer** being adapted for this purpose is now called **Glimmer-CISM**.

We are working with researchers at Oak Ridge, who are trying to replace the old solver with a more sophisticated iteration, and to parallelize the solution procedure, while still (currently) working on a rectangular grid.

https://glimmer-cism.berlios.de/

Introduction: MPAS

The Los Alamos research group has focussed on the mathematical model of the physics, including the question of replacing the uniform rectangular grid, using a program called **MPAS** - “Multiple Predictions Across Scales”.



The **MPAS** program is very general; it is designed to work with a sophisticated polygonal grid over all or some of the Earth's surface. The user places some variables at the centers of cells; fluxes are associated with the interfaces between cells and certain other quantities are assigned at the vertices of the cells. Discretized versions of the state equations are used to update the values over time in a way that obeys conservation laws.

The Los Alamos researchers are especially interested in formulating the physical equations of the ice sheet model in a way that **MPAS** can handle, and to generate a CVT-style mesh over Greenland that has high resolution in areas where the ice sheet is moving fast.

http://mpas.sourceforge.net

BIG PICTURE: How to Find Greenland

Greenland has an area of about 2,000,000 km^2.

Ice covers 85% of the surface, to a maximum depth of 3 km.

BIG PICTURE: Greenland and Climate

The purpose of the investigation is to “plug” Greenland into programs that model global climate.

The primary effects that Greenland exerts on the external climate (ocean and atmosphere) include:

- the temperature of the surface of the ice sheet;
- the reflectance of solar radiation from surfaces covered by ice;
- the rate at which ice and meltwater flows into the ocean;
- where the land-based ice sheet extends into the ocean, the ocean and atmosphere become "decoupled";

The external climate's effect on Greenland can be summarized by

- solar radiation, moderated by cloud cover;
- heat transfer with the atmosphere;
- snowfall, which builds the ice sheet.
- a geothermal flux through the bottom of the ice sheet.

BIG PICTURE: State Variables

The behavior of the ice sheet can be understood in terms of the following state variables:

- the temperature $T(x, y, z)$;
- the pressure $P(x, y, z)$;
- the velocity $\vec{V}(x, y, z)$;
- the ice thickness $H(x, y)$;
- the elevation, or position of the top of the ice sheet, $Q(x, y)$.

We assume that density is essentially constant.

Because we are modeling such a large region, we do not try to study localized features such as cracks and faults in the ice.

BIG PICTURE: Current Resolution 5km x 5km

Current simulations produce values of the state variables on a spatial grid that is horizontally uniform (5km x 5km). This corresponds to a rectangular grid of 301 (East to West) by 561 (North to South) cells by 11 layers $\rightarrow$ 1.5 million nodes.

The needed resolution is 1km x 1km, at least in areas of high ice-sheet velocity; this could be 25 times as many nodes.

BIG PICTURE: Rectangular Grid

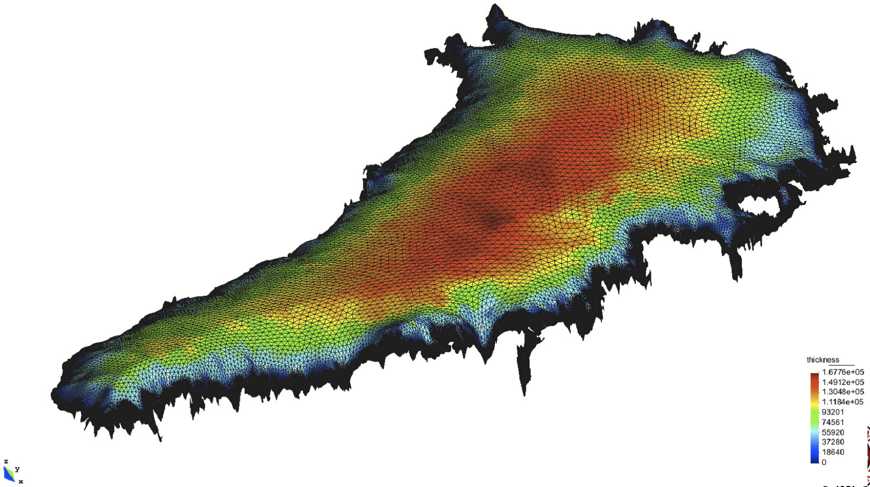
When we ask for a resolution of 1km x 1km, we are really just asking for "five times" the linear resolution of the 5km x 5km. There are vast parts of Greenland where nothing is happening, the ice is not moving, and no finer resolution is ever needed. But we do need to follow the areas of rapid ice movement carefully.

The current version of Glimmer-CISM uses a uniform rectangular grid. That means that to see a single point at high resolution requires using that same high resolution everywhere.

Since Greenland is not a rectangle, it means that many grid cells are placed in the ocean, where they do nothing (but cost us storage and time).

Greenland's coast is not straight, but it will be modeled as though it were constructed out of horizontal and vertical lines.

BIG PICTURE: Ice Thickness

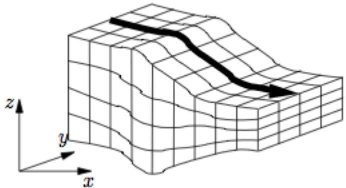


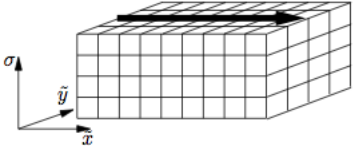
BIG PICTURE: Normalizing the Ice Thickness

In the vertical direction, the grid might use 11 proportionally spaced levels from bedrock to ice surface;

In other versions of the ice sheet model (but not ours!) the vertical direction is rescaled to a variable σ in the range [0,1].

This seems to make life easier, since the top of the ice is now always at $\sigma = 1$, but since the ice thickness varies over space and time, this means that the (x, y, σ) grid is not orthogonal. Equations with spatial derivatives suddenly inherit geometric corrections that look like fictitious forces...**so we don't do this!**





BIG PICTURE: Long Time Interval

To perform simulations as desired, the model must be able to compute data over a time span of 100 years, using a time step that may be on the order of 1 year.

Despite the fact that we are dealing with ice, a time step of a year means we must worry about the CFL (Courant-Friedrichs-Levy) condition, which essentially says that a flow particle mustn't cross more than one spatial cell in a time step.

$$\Delta t \leq C * u * \Delta x$$

The velocity of the ice sheet varies from between 1 to 400 m/year; the current grid has a resolution of about 5 km, while the goal is to provide a resolution of 1 km.

This applies for temperature and thickness, but, surprisingly, not for velocity, because of a quasi-static assumption!

BIG PICTURE: Tight Accuracy Requirement

The physical system satisfies conservation of energy and mass.

The equations used in the model should satisfy these conditions exactly, that is, to within machine roundoff.

In particular, the mass of ice at the beginning, plus all snowfall, minus all calving, must equal the mass at the end of the century, to 16 decimal places, with a similar requirement on energy.

- Introduction
- The Big Picture
- **Mathematical Modeling**
- The Finite Volume Model
- Moving to Finite Elements
- Conclusion

MODEL: The Thickness Equation

The thickness $H(x, y)$ of the ice changes with snowfall and motion of the ice sheet:

$$\frac{\partial H}{\partial t} = H_{\text{flux}} - \nabla \cdot \int_z \begin{bmatrix} u \\ v \end{bmatrix} dz$$

- H_{flux} is the flux due to snowfall;

Assuming the ice always flows outward, the boundary condition for thickness can be taken as a homogeneous Neumann condition.

MODEL: The Temperature Equation

The evolution equation for temperature T:

$$\rho c \frac{\partial T}{\partial t} = \frac{\partial}{\partial z}(\kappa \frac{\partial T}{\partial z}) - \rho c(\vec{u} \cdot \nabla T) + 2\dot{\varepsilon} :: \sigma$$

- c is the heat capacity;
- ρ the (constant) density;
- κ is the thermal conductance;
- $\dot{\varepsilon}$ is the strain rate tensor;
- σ is the stress tensor.

MODEL: The Temperature Boundary Conditions

Ice at the surface is assigned the mean annual surface temperature.

Ice at the bedrock boundary is subject to geothermal flux G and heat generated by sliding friction:

$$-k\nabla T \cdot \vec{n} = G + \vec{t} \cdot \vec{u}$$

MODEL: Stokes Equations for Velocity

The ice sheet can be regarded as a very viscous liquid with a tiny Reynolds number. A quasi-static assumption means that we can also drop the time derivative. What follows is a form of Stokes equation for velocity $\vec{U}$ with a variable viscosity:

$$
\begin{aligned}
-\nabla \cdot\ B(T)\,|\dot{\varepsilon_e}|^{\frac{1-n}{n}}\,(\nabla U + (\nabla U)'))/2 + \nabla P &= \begin{bmatrix} 0 \\ 0 \\ -\rho g \end{bmatrix} \\
\nabla \cdot U &= 0
\end{aligned}
$$

- ρ is the density, g the unit gravitational force;
- $|\dot{\varepsilon_e}|$ is the norm of the strain rate tensor;
- $n \approx 3$;
- $B(T)$ is a constitutive coefficient.

MODEL: Simplified Velocity Equations

Because the ice flow tends to be predominantly in the horizontal plane, it is possible to simply the velocity state equations even further. Simplified models produce smaller sets of equations that can be solved faster.

Three levels of simplification include:

- *The 1st order model*: simplifying the momentum balance in the **Z** direction, and incorporating this assumption into the **X** and **Y** force balance equations;
- *"Shallow Ice Equations"*: assuming that pressure is strictly a function of height, and has the form $P(x, y, z) = \rho g(Q(x, y) - z)$ where $Q(x, y)$ is the position of the top of the ice sheet; it is suitable for slow-sliding regions;
- *"Shallow Shelf Equations"* are a simplification of the 1st order model suitable for fast-sliding regions.

- Introduction
- The Big Picture
- Mathematical Modeling
- **The Finite Volume Model**
- Moving to Finite Elements
- Conclusion

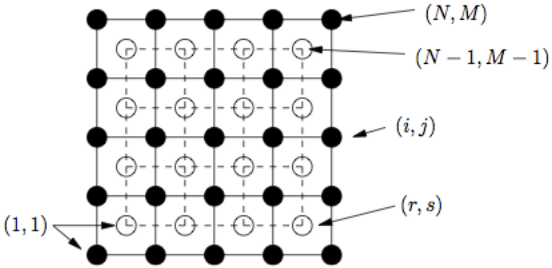
Finite Volumes: A Staggered Grid

The horizontal domain is approximated by a pair of grids, sometimes referred to as the (i, j) and (r, s) grids.

Horizontal velocities U and V are assigned to (r, s) nodes.

Vertical velocities W, ice thickness H, and temperature T, are assigned to (i, j) nodes.

Copies of these 2D grids are generated for each ice sheet layer.



Finite Volumes: Gradients

Gradients of (r, s) objects are assigned to (i, j) nodes:

$$(\frac{\partial u}{\partial x})_{i,j} \approx \frac{u_{r,s} + u_{r,s-1} - u_{r-1,s} - u_{r-1,s-1}}{2\Delta x}$$

A conservation law, written using this kind of scheme for gradients, will correctly conserve the quantity of interest.

Finite Volume: Forming Discrete System

Using the staggered meshes to form approximations to derivatives, and a constant stepsize in time, it is possible to discretize the equations for temperature, ice thickness, and velocity.

Starting from some initial condition, the solution is advanced by timesteps.

Because the number of variables is so large, and the system of equations includes nonlinearities, a direct solution is not attempted. Instead, an iterative scheme is employed. At each step of the nonlinear iteration, several linear systems must be solved. Each of these systems is also solved iteratively, using the **SLAP** sparse linear algebra package.

http://www.netlib.org/slap

Finite Volume: Solving Discrete System

The system size is reduced by uncoupling the state variable equations, and by uncoupling the ice sheet layers.

In particular, the iteration proceeds as follows:

- update the horizontal velocities using the first order equations;
- back out the vertical velocities, to get $\vec{U}(x, y, z)$;
- for each point on the ice sheet bottom layer, integrate the ice thickness equation to update $H(x, y)$;
- by ignoring the effects of horizontal dissipation, integrate the temperature equation from bottom layer to top to get $T(x, y, z)$;

Although information seems to flow only upwards, coupling coefficients relate adjacent ice sheet layers, so information also travels downwards as the iteration proceeds.

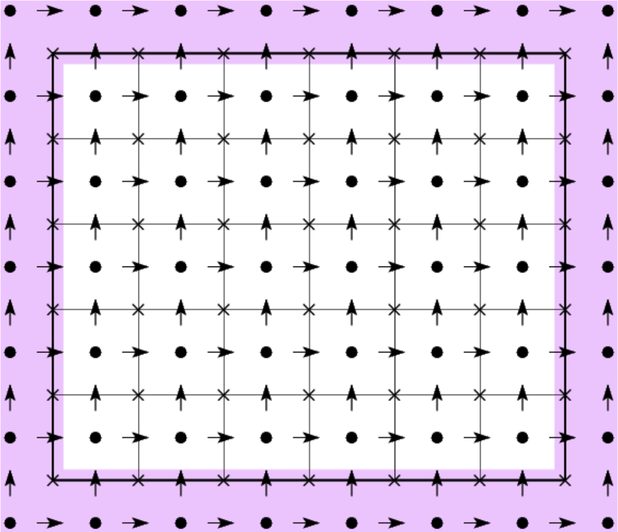
Finite Volume: Parallelism

For each subsystem being solved, the variables are laid out on a horizontal mesh.

To evaluate the equations defined at a node, it is typically necessary to access data at neighbor nodes to the east, west, north and south, as well as the lower and upper layers.

A parallel implementation which divides up the rectangular grid into subrectangles must enable each subrectangle to obtain some information from adjacent subrectangles. This is done by augmenting each partial grid with a layer of "ghost cells" or "halo cells", which are available as information, and do not need to be updated.

Finite Volume: Ghost Cells



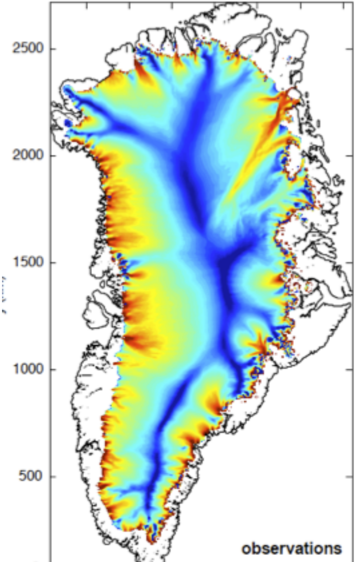
Finite Elements: Advantages

The contribution of our group has been to implement a revised treatment of the calculation using a grid adapted to the geometry and known ice behavior (Lili Ju), and a reformulation of the state equations using finite elements (Mauro Perego).

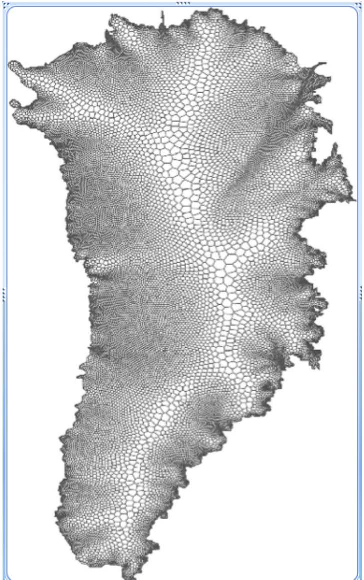
By abandoning the rectangular grid:

- we no longer waste time modeling bits of the ocean;
- we can more accurately follow the coastline of Greenland, and any other geometric objects;
- we can provide a refined mesh in areas of Greenland where the ice sheet velocity is known to be high;
- we can use a grid that smoothly interfaces with grids employed by global climate modeling programs, so that data from one program can be used by the other.

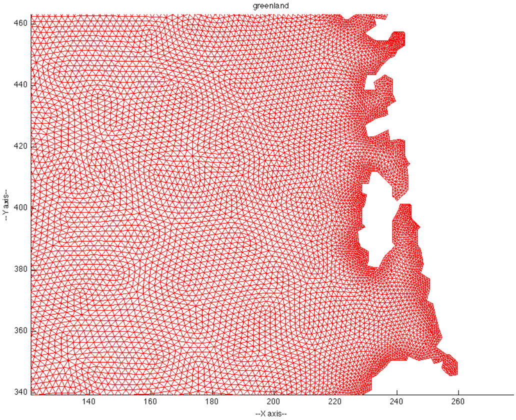
Finite Elements: Observed Ice Sheet Velocity



Finite Elements: a Sample Grid Adapted to Velocity

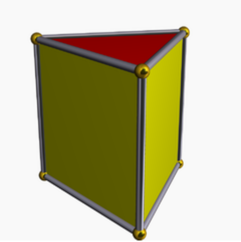


Finite Elements: Detail of Coastline Grid



Finite Elements: Using Triangular Prisms for 3D Elements

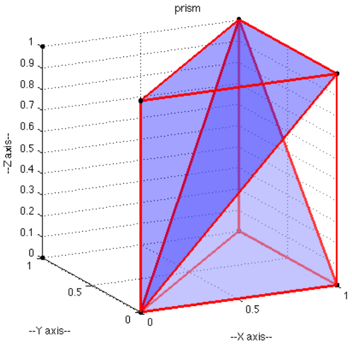
Starting from a 2D triangular grid that meshes a "flattened" version of Greenland, we can build layers in the **z** direction by constructing triangular prisms.



For our model, which involves 11 layers from bedrock to surface, we essentially form a stack of 10 such prisms on every triangle of the original 2D mesh.

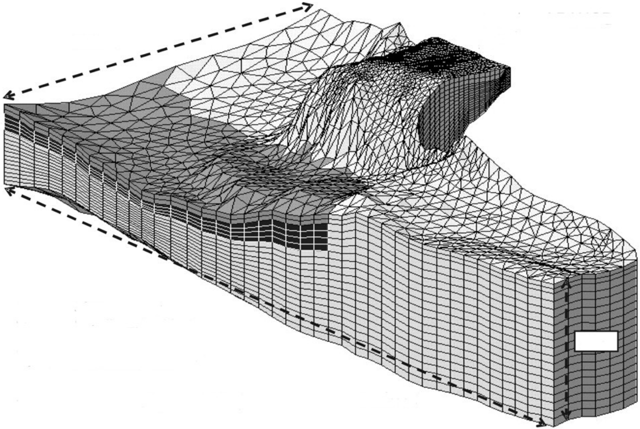
Finite Elements: 3 Tetrahedrons in each Prism

In 3D, the tetrahedron plays the role of fundamental finite element shape. So once we have set up our prisms, we decompose each into 3 tetrahedrons, inside of which we can do our usual finite element computations.



Finite Elements: Example of a Triangular Prism Grid

This is *not* the grid for an ice sheet, but it suggests the layered nature of the triangular prism grid.



Finite Elements: Example of a Triangular Prism Grid

All the state equations can be discretized using the same mesh.

The parallel assembly of the system matrix does not require any communication between processors at all.

The parallel solution of the linear systems arising in the nonlinear iteration at each time step only requires the use of an appropriate library solver.

In fact, the solution of the entire nonlinear system can also be done in parallel, using an off-the-shelf library solver.

We use Sandia National Laboratory's **Trilinos** package.

http://trilinos.sandia.gov/

Finite Elements: Trilinos NOX Nonlinear Solver

NOX is an object-oriented C++ library for large nonlinear systems. It implements Newton-based globalization techniques including line search and trust region algortihms. NOX defines interfaces to user codes through the abstract group and vector pure virtual classes.

The user can supply the underlying linear algebra solver needed by Trilinos to carry out the iterations involved in solving the nonlinear system.

To improve performance, the user can supply preconditioning or jacobian information.

http://trilinos.sandia.gov/packages/nox/

Finite Elements: Trilinos EPETRA Parallel Linear Algebra

Instead of supplying the linear algebra solver, the user can take advantage of the Trilinos's **Epetra** package. With **Epetra** the user only needs to evaluate the residual equation $F(x) = 0$ for a given x.

Epetra contains classes for distributed sparse and dense matrices and vectors; It provides a flexible and powerful data redistribution capability for load balancing and scalability of linear algebra algorithms without the user needing any special knowledge about distributed object.

Epetra provides a parallel machine interface that allows users to write generate parallel functionality without specifically using any particular parallel library.

http://trilinos.sandia.gov/packages/epetra/

Finite Elements: LIFEV Finite Element Library

To increase the order of approximation of a finite difference or finite volume code can require an extensive rewrite.

For a finite element code, the order of approximation and even the form of the equations are easily changed in a way that does not obviously affect the main user code. Instead, these choices are implemented in separate code.

LifeV is a C++ package for finite element calculations such as those involving fluids, heat transfer, structures, porous media. It makes extensive use of the modern features of C++, allowing a developer to rapidly model a physical system.

Mauro Perego is a developer of the **LifeV** package, and has been able to transfer the ice sheet model to this framework and solve benchmark problems with it.

https://sites.google.com/site/lifevproject

Conclusion: A Better Grid

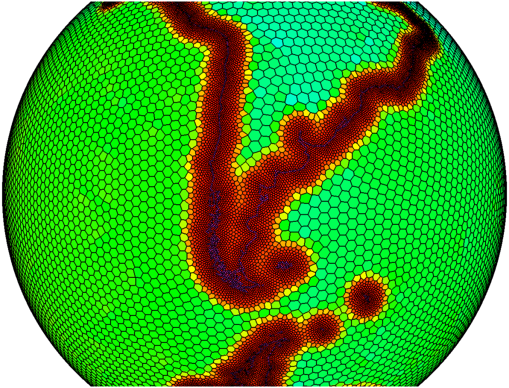
By dropping the uniform rectangular grid and moving to an adaptive mesh:

- we reduce the number of wasted cells;
- we can better conform to the geometry of the region;
- we can choose a weighting function (such as observed ice velocity) to vary the fineness of the mesh;
- we can refine the mesh near the coasts for better interaction with other simulation packages.

This allows us to achieve the desired 1 km x 1km resolution in areas of high ice sheet velocity.

Conclusion: Grid interfaces

The grid is flexible, and can be refined at the coastline so it can exchange more detailed information with a separate program modeling the ocean.

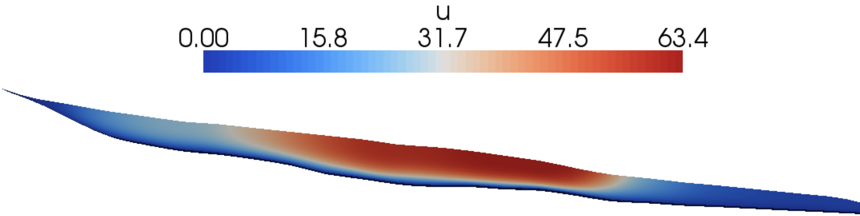


Conclusion: Using Finite Elements

By formulating the problem using finite elements:

- we are able to use the adaptive mesh;
- increasing the approximation power only requires changing a parameter;
- approximated state variables can be evaluated anywhere;
- the nonlinear solution can be handed off to an external library;
- the parallelism can be handed off to an external library.

Conclusion: Arolla Glacier Test Case

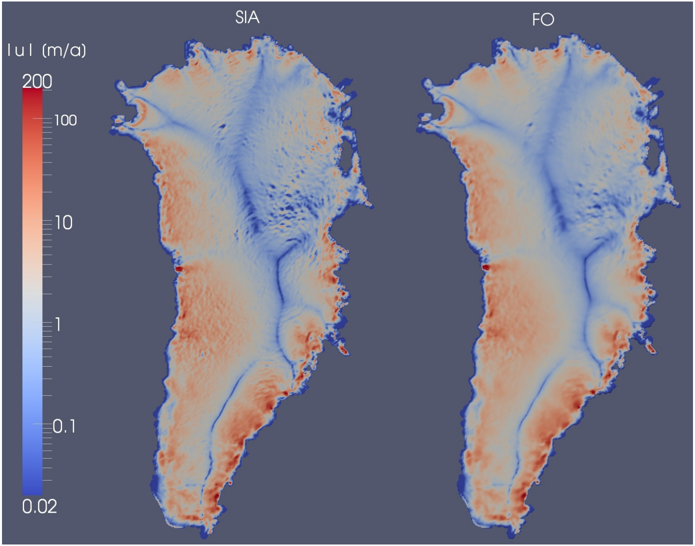


Conclusion: Sample Greenland Calculations

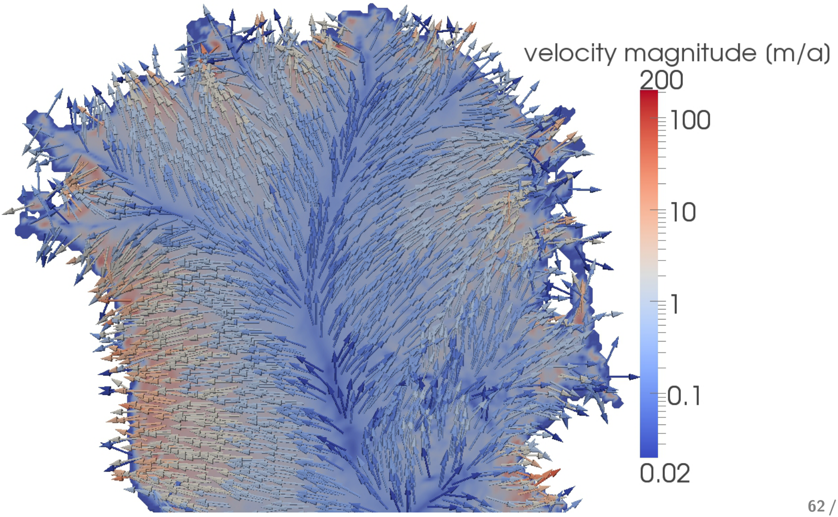
Using the finite element model that Mauro Perego has developed, we are able to run mathematical models of increasing accuracy (Shallow Ice, Shallow Shelf, First Order, L1L0) on grids of the desired resolution.

Mauro used 10 layers, for a total of 3.9 million tetrahedra and 1.4 million unknowns. The temperature field is given and ranges between 250 to 273 K. No slip boundary conditions were always used at the bedrock, because sliding data was not available.

Conclusion: SIA Model versus First Order



Conclusion: Velocity Vector Closeup



Conclusion: Using TRILINOS Package

By using TRILINOS and EPETRA for solving $F(X) = 0$ and $A * x = b$:

- we are no longer responsible for the housekeeping details required when implementing a parallel code;
- we guarantee good parallel performance on a wide range of configurations
- we have access to a variety of high-quality linear and nonlinear solvers with a uniform interface.

Conclusion

Our challenges include:

- implementing the temperature and thickness equations in finite element form (right now, we are only doing the velocities this way);
- converting the routines in LifeV from C++ to FORTRAN90, because the climate community insists on a uniform language;
- matching the very tight tolerances ($\sim$ machine precision) on conservation of mass and energy over the 100 year simulation cycles;
- finishing the process of verification, documentation, and publication before the end of 2012, after which no new input will be accepted for the Intergovernmental Panel on Climate Change (IPCC) Assessment Report 5 to be published in 2014.

Conclusion: Colleagues and Reference

- FSU/SC: Max Gunzburger, Lili Ju, Tao Cui, Wei Leng, Mauro Perego (gridding and finite elements);
- NYU: Jean-Francois Lemieux (solver);
- Oak Ridge; Kate Evans, Jeff Nichols, Pat Worley;
- Los Alamos; Bill Lipscomb, Steve Price, Todd Ringler, Xyler Asay-Davis, Dana Knoll;
- Sandia: Andy Salinger (Trilinos);

Mauro Perego, Max Gunzburger, John Burkardt,
Implementation and comparison of linear and quadratic finite element methods for higher-order ice-sheet models,
submitted, Journal of Glaciology.

Conclusion: Follow Up Seminar Begins at 5:30++

The Mellow Mushroom Institute