Verifying PISM, once known as "COMMVNISM"

A C++ Object-oriented Multi-Modal, Verifiable Numerical Ice Sheet Model

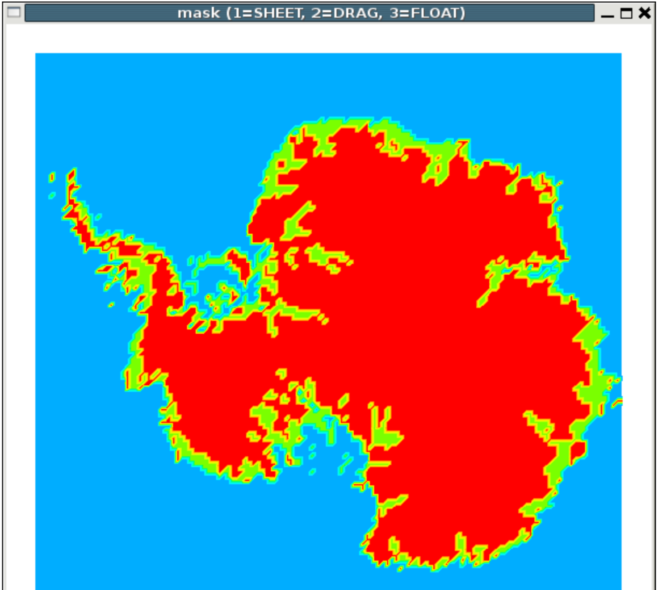
Jed Brown[1] Ed Bueler[1] Craig Lingle[2]

[1]Department of Mathematics
University of Alaska Fairbanks

[2]Snow, Ice, and Permafrost Group
University of Alaska Fairbanks

2006-08-04 / MS Project Defense / retitled by ELB 10/17/06

Multi-modal flow



Theory and Implementation

1. Physics
   - Constitutive Relations
   - Stokes Equations
   - Basal dynamics
2. Approximations
   - Shallow Ice Approximation
   - Ice Shelf and Stream Flow
3. Numerical Tools
   - Portable Extensible Toolkit for Scientific computing
   - Programming Considerations
4. Solving the equations
   - Shallow Ice Approximation
   - The Macayeal Equations

Outline

The Stress Tensor

Stress

Units of pressure (Pa).

$$\tau = \begin{bmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{bmatrix}$$

Pressure

$$P = -\frac{1}{3}(\tau_{xx} + \tau_{yy} + \tau_{zz})$$

Deviatoric stress

$$\sigma_{ij} = \tau_{ij} + P\delta_{ij}$$

Strain Rate

Strain

$$\epsilon = \frac{\Delta L}{L}$$

Strain rate

$$D_{ij} = \dot{\epsilon}_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

Relationship to stress

$$D_{ij} = F(\sigma, \dots)\sigma_{ij}$$

where σ is the second invariant of σ_{ij}

$$2\sigma^2 = \|\sigma_{ij}\|^2_{\text{Frob}} = \sigma_{ij}\sigma_{ij}$$

Simplifications

Symmetry

Deviatoric stress and strain rate are symmetric tensors.

$$\sigma_{ij} = \sigma_{ji} \qquad D_{ij} = D_{ji}$$

Trace zero

- Deviatoric stress has zero trace.
- Incompressibility implies strain rate is also trace free.

$$\sigma_{ii} = 0 \qquad D_{ii} = 0$$

Glen Flow Law

Recall

$$D_{ij} = F(\sigma, \ldots)\sigma_{ij}$$

Power Law

$$F(\sigma, T, P) = A(T^*)\sigma^{n-1}$$

in terms of homologous temperature

$$T^* = T + CP$$

Arrhenius Relation

$$A(T^*) = A_0 \exp\left(-\frac{Q}{RT}\right)$$

Goldsby-Kohlstedt Flow Law

A nontrivial combination

$$F(\sigma, T^*, P, d) = F_{\text{diff}}(\sigma, T^*, d) + F_{\text{disl}}(\sigma, T^*, P) + \left(\frac{1}{F_{\text{basal}}(\sigma, T^*)} + \frac{1}{F_{\text{gbs}}(\sigma, T^*, P, d)} \right)^{-1}$$

Each term has form similar to Glen's flow law, but

- different exponents, but all ≥ 1
- different Arrhenius terms

Monotonicity

The function $D(\sigma) = F(\sigma, T^*, P, d)\sigma$ is strictly increasing.

Outline

Stokes equations

Incompressibility

$$\nabla \cdot \boldsymbol{u} = 0$$

Force balance

$$(\text{Inertial term}) = \nabla \cdot \boldsymbol{\tau} + \boldsymbol{F}$$

Slow flow

drop the inertial term and write in terms of deviatoric stress

$$\nabla P = \nabla \cdot \boldsymbol{\sigma} + \boldsymbol{F}$$

where $\boldsymbol{F} = \rho \boldsymbol{g}$ is gravitational force

Inverting a flow law

Working with stresses is sometimes inconvenient.

A different approach

With $D^2 = \frac{1}{2} D_{ij} D_{ij}$, define $\nu(D, \dots)$ so that the scalar equation

$$\sigma = 2\nu(D, \dots) D$$

is equivalent to

$$D = F(\sigma, \dots)\sigma$$

Example

For the Glen flow law,

$$\nu(D, T^*) = \frac{B(T^*)}{2} D^{\frac{n-1}{n}}$$

Outline

Linear sliding

Note

All basal sliding is thermally activated.
If ice is frozen to the bed, there is no sliding.

Motivation

Linear viscous till

- viscosity ν
- thickness L

$$(\text{basal velocity}) = \frac{L}{\nu}(\text{basal stress})$$

Dragging

$$(\text{basal stress}) = \beta(\text{basal velocity})$$

Alternate schemes

Power law till

$$u_i = C\sigma^{n-1}\sigma_{i3}$$

$$\sigma_{i3} = \beta(\boldsymbol{u})u_i$$

Plastic till

$$\sigma_{i3} = \sigma_{\text{critical}}\frac{u_i}{|\boldsymbol{u}|}$$

Basal water models

1. Solve a nonlinear PDE for water pressure (Jesse Johnson)
2. Use bed elevation and basal melt rate in an ad-hoc scheme

Outline

1. Physics
   - Constitutive Relations
   - Stokes Equations
   - Basal dynamics
2. Approximations
   - Shallow Ice Approximation
   - Ice Shelf and Stream Flow
3. Numerical Tools
   - Portable Extensible Toolkit for Scientific computing
   - Programming Considerations
4. Solving the equations
   - Shallow Ice Approximation
   - The Macayeal Equations

Non-dimensionalization

Critical assumptions

Thickness scale $[H]$, horizontal scale $[L]$, aspect ratio $\epsilon = [H]/[L]$

$$\sigma_{13}, \sigma_{23} \sim \rho g [H] \epsilon \qquad \sigma_{ii}, \sigma_{12} \sim \rho g [H] \epsilon^2$$

$$P - \rho g (h - z) \sim \rho g [H] \epsilon^2$$

Consequences

1. Only shear parallel to bed remains σ_{13}, σ_{23}
2. Shear is proportional to depth.
3. Flow is completely determined by local quantities

Equations

The system

$$\frac{\partial h}{\partial t} = M - \nabla \cdot \boldsymbol{Q}$$

$$\boldsymbol{Q} = \overline{\boldsymbol{U}} H \quad \text{and} \quad \boldsymbol{Q} = D \nabla h$$

$$\frac{\partial \boldsymbol{U}}{\partial z} = -2F(\sigma, T, \dots) P \nabla h$$

$$\frac{\partial T}{\partial t} + \boldsymbol{u} \cdot \nabla T = K \frac{\partial^2 T}{\partial z^2} + (\text{strain heating})$$

Isothermal, Glen

$$h_t = M + \nabla \cdot \left[\Gamma H^{n+2} \left| \nabla h \right|^{n-1} \nabla h - H \boldsymbol{u}_{\text{basal}} \right]$$

Sliding in SIA regions

Example

A common algorithm

1. Compute shear stress at the bed: $\sigma_{i3} = \rho g H \frac{\partial h}{\partial x_i}$
2. if $T_{\text{bed}} = T_{\text{pmp}}$ then $u_i = \mu \sigma_{i3}$
3. else $u_i = 0$

Problems

1. Horizontal velocity is not continuous
2. Vertical velocity is unbounded
3. Poor behavior under grid refinement

Outline

Non-dimensionalization

Thickness $[H]$, thickness variation $[s]$, aspect ratio $\epsilon = [H]/[L]$

Critical assumptions (Schoof 2006)

$$\sigma_{13}, \sigma_{23} \sim \rho g [s] \epsilon \qquad \sigma_{ii}, \sigma_{12} \sim \rho g [s]$$

$$P - \rho g (h - z) \sim \rho g [s]$$

Recall SIA critical assumptions

$$\sigma_{13}, \sigma_{23} \sim \rho g [H] \epsilon \qquad \sigma_{ii}, \sigma_{12} \sim \rho g [H] \epsilon^2$$

$$P - \rho g (h - z) \sim \rho g [H] \epsilon^2$$

Consequences

1. No shear parallel to bed
2. Horizontal velocity is independent of depth

The MacAyeal Equations

Coordinate free form (Vectors in 2 dimensions)

$$\nabla \cdot (2\nu H \boldsymbol{D}) + \nabla \operatorname{tr}(2\nu H \boldsymbol{D}) - \beta \boldsymbol{u} = \rho g H \nabla h$$

$$\nu = \frac{\overline{B}}{2} \left(\frac{1}{2} \|\boldsymbol{D}\|^2_{\mathsf{Frob}} + \frac{1}{2} (\operatorname{tr} \boldsymbol{D})^2 \right)^{\frac{1-n}{2n}}$$

Usual form

$$\left[2\nu H(2u_x + v_y)\right]_x + \left[\nu H(u_y + v_x)\right]_y - \beta_1 u = \rho g H h_x$$

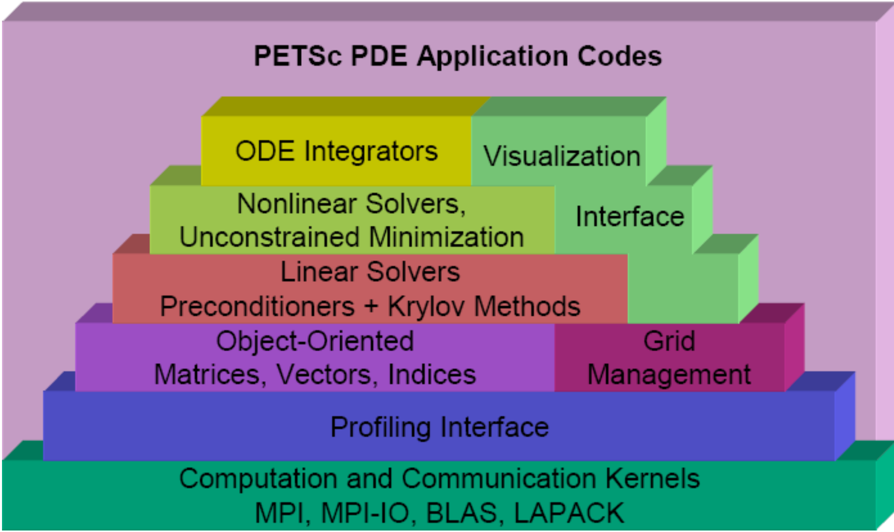
$$\left[2\nu H(2v_y + u_x)\right]_y + \left[\nu H(u_y + v_x)\right]_x - \beta_2 v = \rho g H h_y$$

$$\nu = \frac{\overline{B}}{2} \left[\frac{1}{2} u_x^2 + \frac{1}{2} v_y^2 + \frac{1}{2} (u_x + v_y)^2 + \frac{1}{4} (u_y + v_x)^2 \right]^{\frac{1-n}{2n}}$$

Outline

1. Physics
   - Constitutive Relations
   - Stokes Equations
   - Basal dynamics
2. Approximations
   - Shallow Ice Approximation
   - Ice Shelf and Stream Flow
3. **Numerical Tools**
   - **Portable Extensible Toolkit for Scientific computing**
   - Programming Considerations
4. Solving the equations
   - Shallow Ice Approximation
   - The Macayeal Equations

Overview


PETSc PDE Application Codes
ODE Integrators
Visualization
Nonlinear Solvers,
Unconstrained Minimization
Interface
Linear Solvers
Preconditioners + Krylov Methods
Object-Oriented
Matrices, Vectors, Indices
Grid
Management
Profiling Interface
Computation and Communication Kernels
MPI, MPI-IO, BLAS, LAPACK

Message Passing Interface

Advantages

1. Message passing standard
2. Portable
3. Low level
4. Fast
5. Flexible

Disadvantages

1. Low level

Distributed arrays and vectors

DA

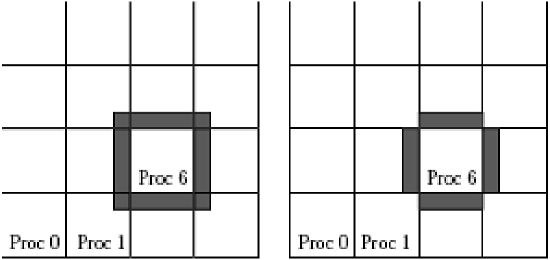
Describes parallel layout

1. Ghosted values
2. Periodicity
3. Coordinates

Vec

Holds scalar quantities

1. Can be based on DA
2. Global vs Local
3. Can be viewed
4. Algebra with matrices



Iterative linear algebra

Problems with direct linear algebra

1. Slow: $\mathcal{O}(n^3)$
2. Does not take advantage of sparsity
3. Complicated and may not parallelize well

Krylov Subspace Methods

Orthogonalize the subspace $\mathrm{span}\{b, Ab, A^2b, \ldots, A^kb\}$
Minimize norm of residual $r = b - Ax$ over subspace.

Preconditioning

Better convergence when condition number $\|A\| \, \|A^{-1}\|$ is small.
If $P^{-1}A$ is well conditioned, solve $P^{-1}Ax = P^{-1}b$.

Krylov Subspace methods

Problem

Efficiently solve $Ax = b$ where x and b are distributed and A is

1. Sparse, huge and distributed OR
2. Defined by a function

KSP Acronym Soup

1. Conjugate Gradients
2. GMRES(n)
3. Bi-CGStab
4. Transpose free QMR
5. MINRES

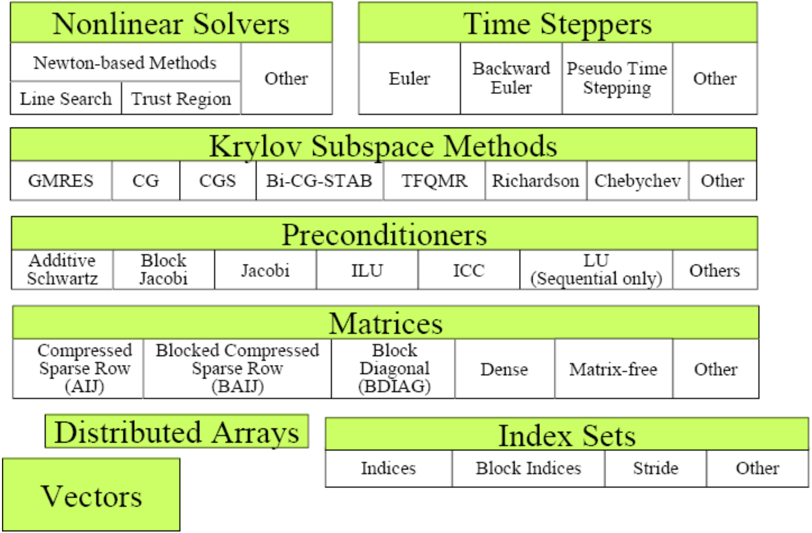
PC Acronym Soup

1. (block) Jacobi
2. SOR
3. ILU(k), ICC(k)
4. Multigrid
5. External AMG

Bonus

All KSP and PC are extremely customizable at command line.

Putting it all together



Visualization

PETSc

1. Runtime viewers
2. Real-time convergence monitoring for KSP
3. Internal diagnostics and profiling

External software

1. Matlab
2. Vis5D

Outline

Other software

Recommended open source tools

1. netCDF: A platform independent binary format
2. FFTW: Discrete Fourier Transforms
3. GSL: Integration, special functions, etc.
4. Python/Ruby: Preprocessing

Parallel considerations

Array layout

1. Our 3D work is column oriented; address as `T[i][j][k]`
2. 3D and 2D arrays should have compatible layout
3. Periodic

Message passing

1. need to communicate ghosted values before taking derivatives
2. ghosted values are small packets
3. latency is more critical than bandwidth
4. multiplexing ghosted communication would help

Scaling

1. many communications per time step
2. can saturate more processors with large grids

Outline

Class structure

IceModel

IceGrid
parallel layout

IceParam
Serializable
Parameters & history

IceType
$F(\sigma), \nu(D)$

BedrockType
Constants

OceanType
Coupling point

IceType
Abstract class

GlenIce
Isothermal

ThermoGlenIce
Split Arrhenius (P-B)

HybridIce
G-K for $F(\sigma, \dots)$
ThermoGlen for $\nu(D, \dots)$

IceCompModel
Verification

ShelfModel
Verification

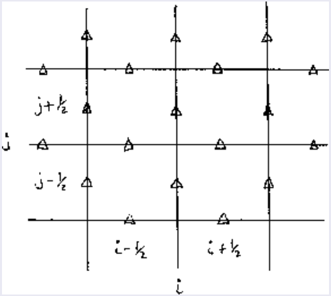
Core model

Continuity Equation

$$h_t = M - \nabla \cdot \boldsymbol{Q} \tag{1}$$

$$\boldsymbol{Q} = D\nabla h \text{ (SIA)} \qquad \boldsymbol{Q} = \overline{\boldsymbol{U}}H \text{ (MacAyeal)} \tag{2}$$

Staggered grid



Features

1. Explicit mass balance
2. Semi-implicit temperature
3. Adaptive time stepping
4. Asynchronous grain size
5. Flexible parallel regridding

Outline

The iteration

Algorithm

until converged **do**

compute effective viscosity

assemble linear system

solve linear system

end

Orderings

1. DA ordering is nice for finding neighbors
2. Matrix structure would be obnoxious in DA ordering
3. We want to solve for two vectors (u, v) simultaneously
4. We can use a different ordering for the linear system

Effective viscosity

Equation

$$\nu = \frac{\overline{B(T,\dots)}}{2}\left[\frac{1}{2}u_x^2 + \frac{1}{2}v_y^2 + \frac{1}{2}(u_x+v_y)^2 + \frac{1}{4}(u_y+v_x)^2\right]^{\frac{1-n}{2n}}$$

Observations

1. We need ghosted values to calculate ν
2. It is hard to find neighbors in KSP-ordering
3. We need to do a scatter operation during the iteration.

Effective viscosity

Equation

$$\nu = \frac{\overline{B(T,\ldots)}}{2}\left[\frac{1}{2}u_x^2 + \frac{1}{2}v_y^2 + \frac{1}{2}(u_x+v_y)^2 + \frac{1}{4}(u_y+v_x)^2\right]^{\frac{1-n}{2n}}$$

Observations

1. We need ghosted values to calculate ν
2. It is hard to find neighbors in KSP-ordering
3. We need to do a scatter operation during the iteration.

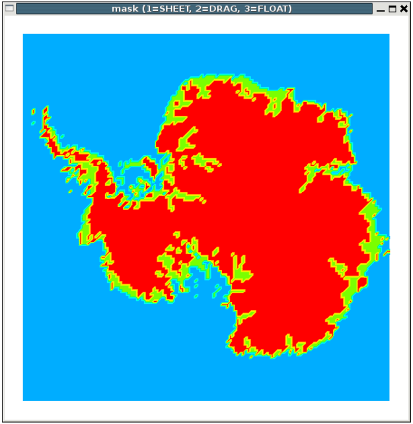
Alternatives

1. Manually communicate ghosted values in KSP-ordering
2. Create a new DA to hold 2-vectors and use for Matrix ordering

Matrix assembly

Equation

$$2\left[\nu H(u_x + v_y)\right]_x + \left[\nu H(u_y + v_x)\right]_y - \beta u = \rho g H h_x$$



SIA Region

Put 1 on diagonal and computed $\boldsymbol{u}$ on RHS

Macayeal Region

1. 13 point stencil
2. ν and β use old $\boldsymbol{u}$

Solving the linear system

Our choice

GMRES(30) with ILU preconditioning

Just as good

CGS with Block Jacobi preconditioning

half the iterations, but iterations take twice as long

Multigrid Preconditioning

1. Would be easy to implement
2. Ice streams are inherently high frequency
3. Probably a waste of time

A better scheme

Minimize the Schoof-MacAyeal Functional

$$J(\boldsymbol{v}) = \int_\Omega \frac{2BH}{p}\left[D_{ij}(\boldsymbol{v})D_{ij}(\boldsymbol{v})/2 + D_{ij}(\boldsymbol{v})^2/2\right]^{p/2} + \tau\left|\boldsymbol{v}\right| - \boldsymbol{f}\cdot\boldsymbol{v}d\Omega - \int_{\partial\Omega}\boldsymbol{F}\cdot\boldsymbol{v}d\Gamma$$

Nonlinear conjugate gradients

1. Easy to implement with PETSc SNES
2. The Jacobian is essentially the same matrix as before
3. Should converge faster and be more robust

Verification and Antarctica

5 Verification
- Shallow Ice
- Macayeal Equations

6 Antarctica
- Current data
- Tuning
- Bootstrapping

Outline

Isothermal Tests

Exact solutions

1. Moving margin similarity
2. Compensatory accumulation
3. Basal sliding with compensatory accumulation

Observations

1. Exact numerical volume conservation
2. Large errors near margin
3. Small errors in interior

Thermocoupled Tests

Exact solutions

1. Compensatory heating
2. Perturbation in anulus
3. Margin has isothermal shape

Observations

1. Convergence of coupled geometry and temperature
2. No “spokes”
3. Verified model produces spokes for EISMINT experiment F

Outline

Exact solutions

The 1D Weertman solution

$$D_{xx} = \left(\frac{\rho g h}{4\overline{B}}\right)^n$$

The 2D Weertman solution

$$D_{xx} = D_{yy} = 3^{-(n+1)/2}\left(\frac{\rho g h}{2B}\right)^n$$

Observation

All second derivatives are zero. These are boring for verification.

Finding an interesting exact solution

Compensatory drag

1. Choose nontrivial H, b, u, v
2. Compute β_1 and β_2 to satisfy Macayeal equations

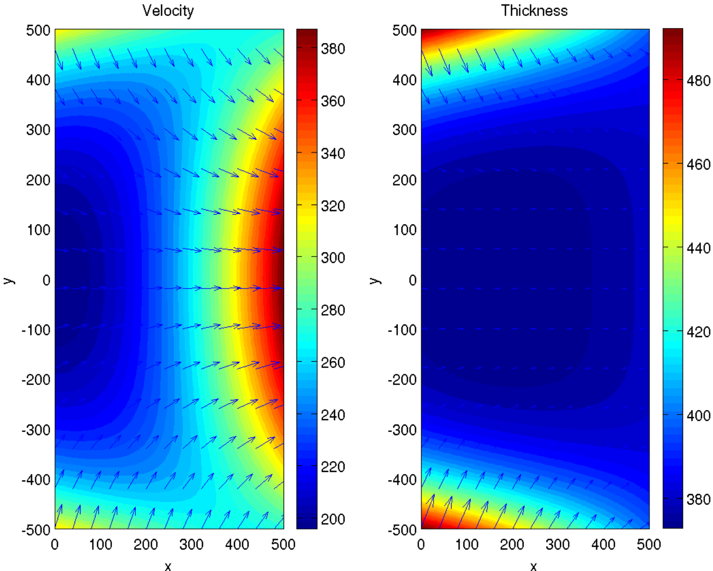
Software helps

Use Matlab or a symbolic algebra software (Maxima, Maple, Mathematica)

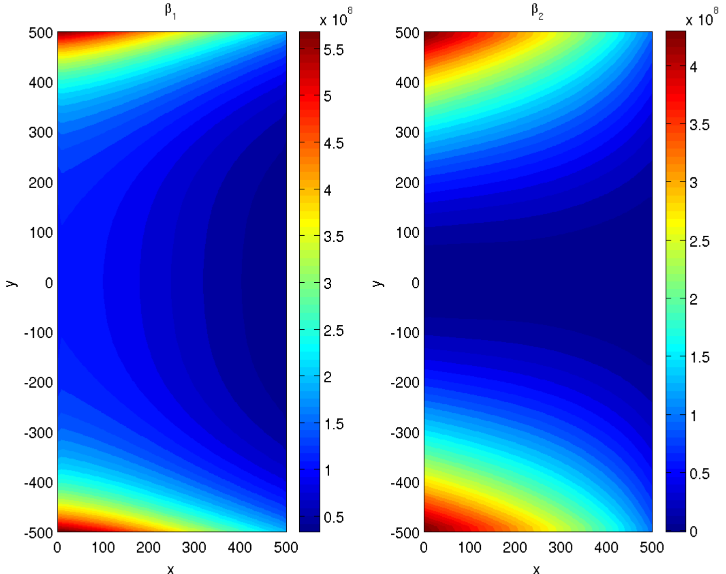
Criteria for a good compensatory solution

1. β_1 and β_2 should both be nonnegative and reasonably sized
2. β_1 and β_2 should be similar

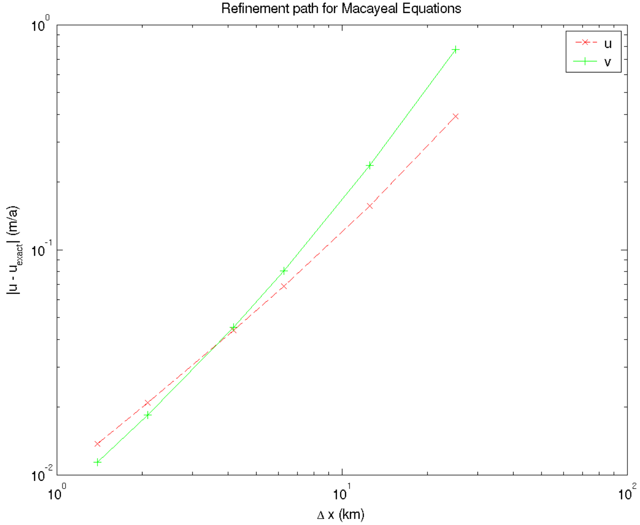
A solution


Velocity
500
400
300
200
100
0
-100
-200
-300
-400
-500
y
0
100
200
300
400
500
x
380
360
340
320
300
280
260
240
220
200
Thickness
500
400
300
200
100
0
-100
-200
-300
-400
-500
y
0
100
200
300
400
500
x
480
460
440
420
400
380

How realistic is it?



Convergence of the linearized problem



Convergence of the nonlinear problem

Problem

There may be multiple fixed points of our iteration.

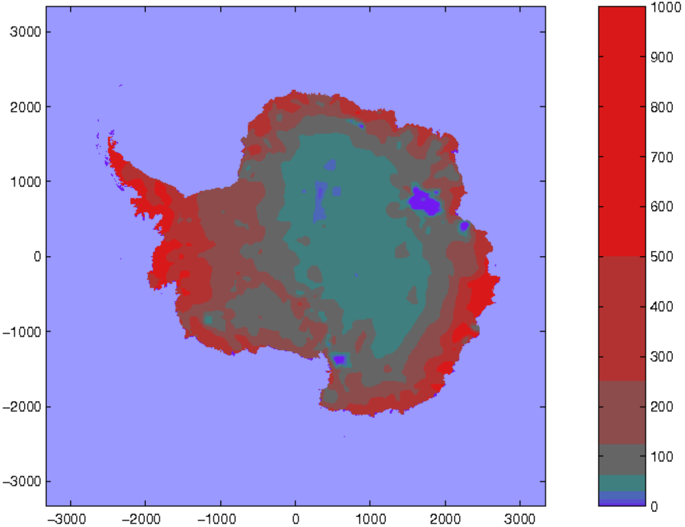
Outline

5 Verification
- Shallow Ice
- Macayeal Equations

6 Antarctica
- Current data
- Tuning
- Bootstrapping

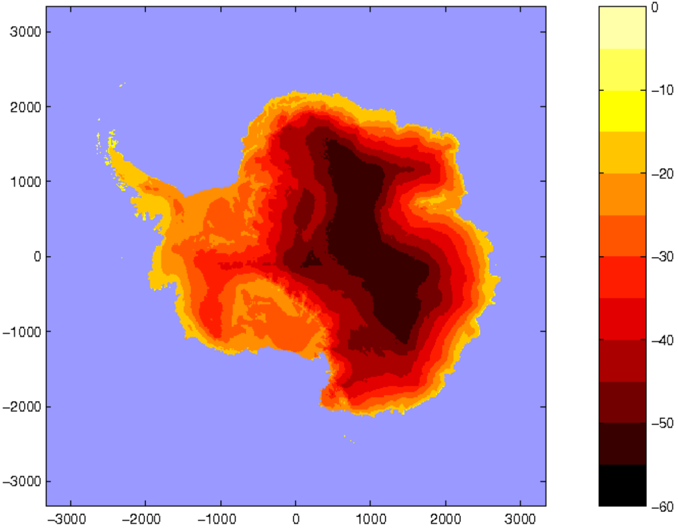
Accumulation

British Antarctic Survey 2004



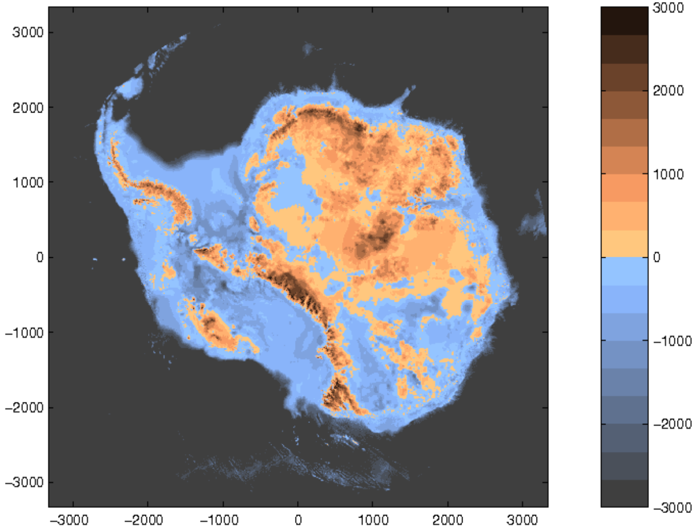
Temperature

British Antarctic Survey 2004



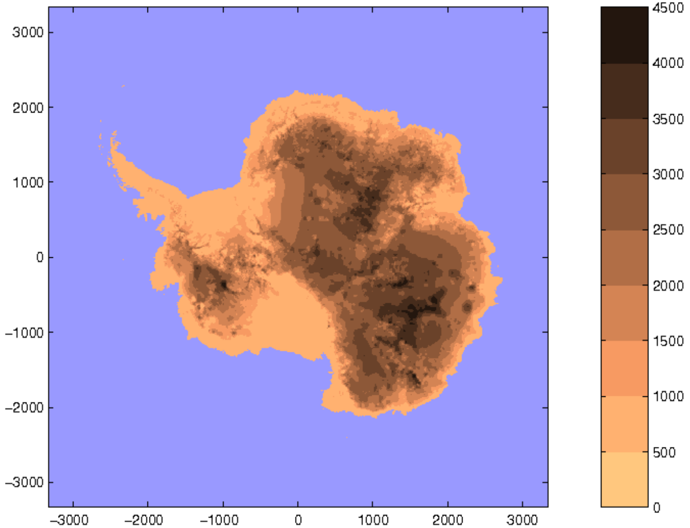
Bed elevation

British Antarctic Survey 2004



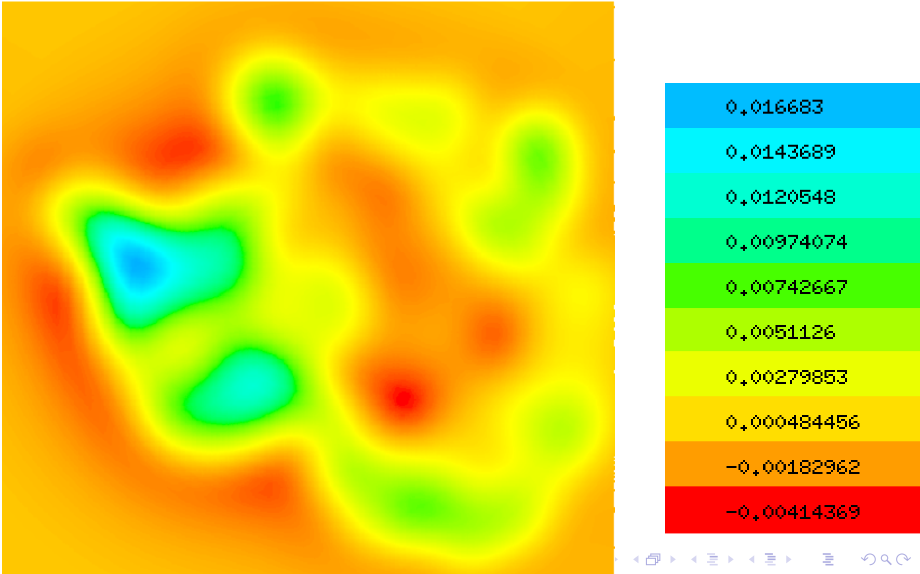
Thickness

British Antarctic Survey 2004



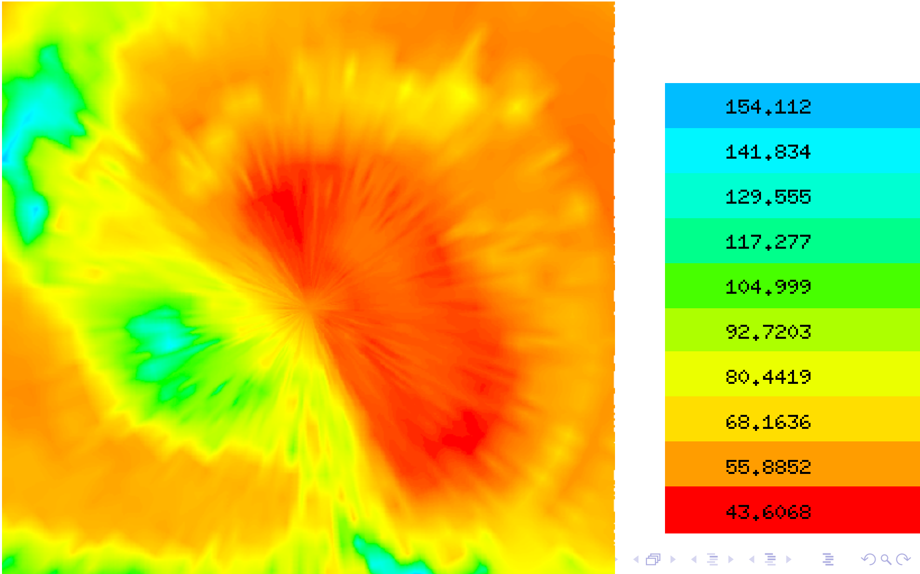
Uplift Rate

Ivins and James (1998; JGR)



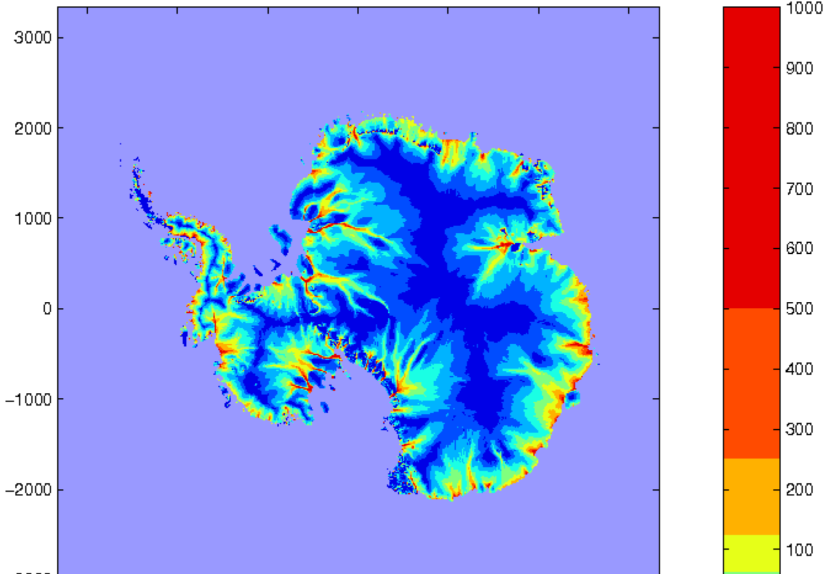
Geothermal heat flux

Shapiro & Ritzwoller (2004; Earth Planetary Sci. Let.)



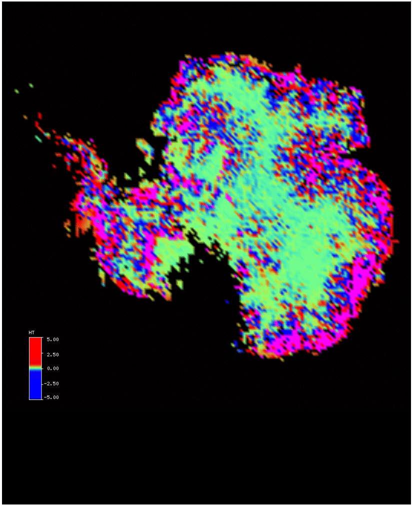
Balance velocity

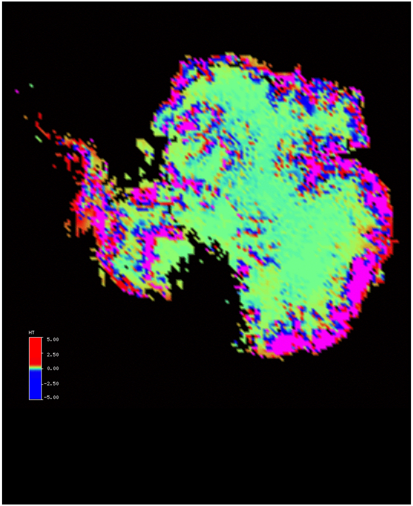
Bamber, Vaughan and Joughin (1999)



Outline

Assessing a flow law





Glen vs. Goldsby-Kohlstedt

One measure

Let F be the frozen bed region and optimize enhancement factor so

$$\int_F h_t = 0.$$

Observe $\|h_t\|_{L^1(F)}$ and $\|h_t\|_{L^2(F)}$.

Constitutive Relation	1-norm	2-norm
Goldsby-Kohlstedt	0.52 m/a	1.5 m/a
Glen	1.3 m/a	4.9 m/a

Outline

Cleaning and smoothing

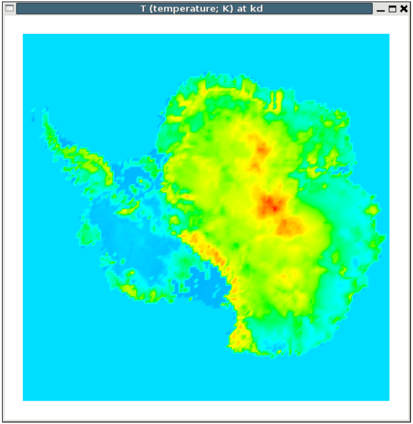
Problem

Initial data can rough, inconsistent and missing.

Process

1. Patch up missing values where there is a principle
2. Solve Laplace's equation in the remaining regions
3. Modify the mask to be compatible with new results
4. Make data consistent where appropriate
5. Run model for a short period to smooth data

Temperature and age



Problem

Temperature and grain size are needed to compute flow.

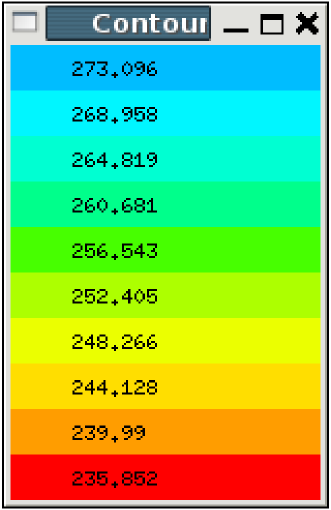
The Right Way ©

Solve the inverse problem.

The Ad-hoc Way

Hold geometry constant while running temperature evolution until it converges.

Temperature and age



Problem

Temperature and grain size are needed to compute flow.

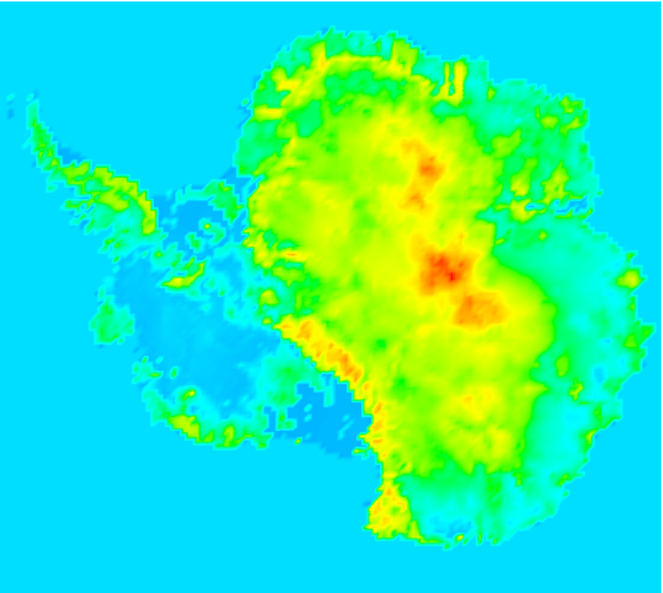
The Right Way ©

Solve the inverse problem.

The Ad-hoc Way

Hold geometry constant while running temperature evolution until it converges.

Temperature and age



needed

The Rig

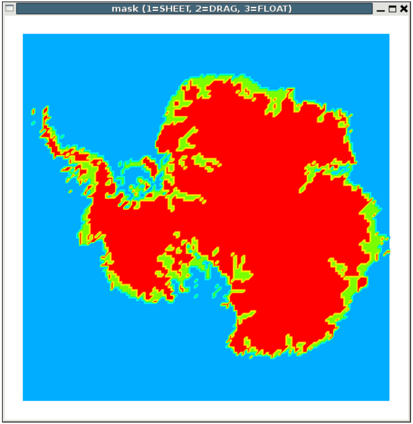
Solve th

The Ad-

Hold ge

running

Which regime is where?



Problem

Where are the streams.

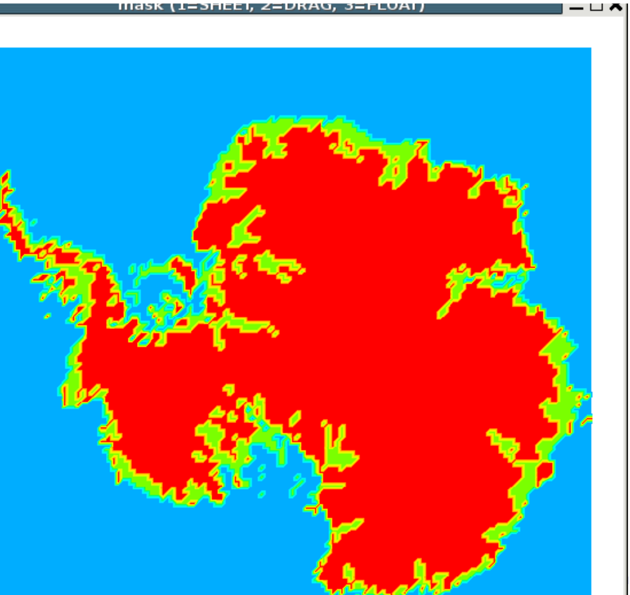
The Right Way ©

Solve the some hard variational inequality ala Schoof.

The Ad-hoc Way

1. Use balance velocity and SIA deformation to calculate u_{basal}.
2. Where $|u_{\text{basal}}|$ is large, use streams.

Which regime is where?



Where are th

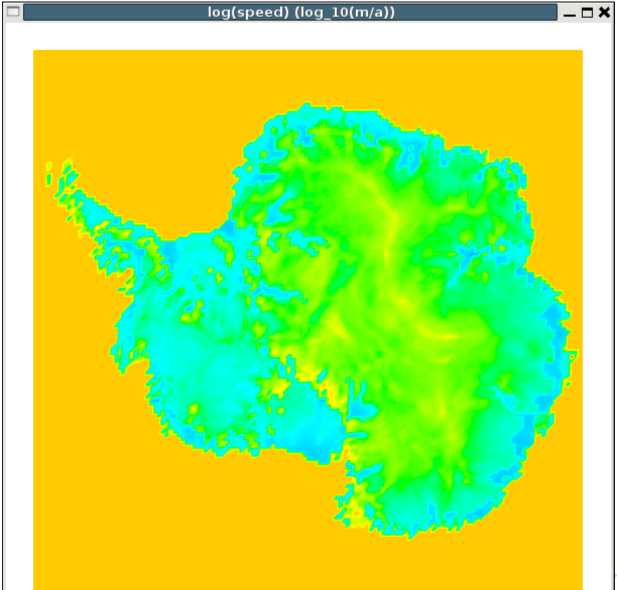
The Right W

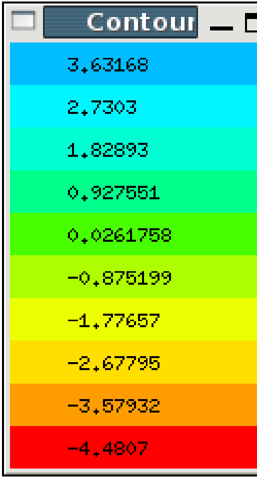
Solve the so
inequality al

The Ad-hoc

1. Use bal
SIA def

$\log_{10}$ Speed


log(speed) (log_10(m/a))


Contour
3.63168
2.7303
1.82893
0.927551
0.0261758
-0.875199
-1.77657
-2.67795
-3.57932
-4.4807

A full ice age cycle

Observations

1. Even if Antarctica is nearly in steady state now, it hasn't been forever.
2. Temperature (in ice and bedrock) reflect history.
3. Age and grain size reflect history.
4. Uplift rates reflect history.

We need

1. a well tuned model
2. a good reconstruction of climate history
3. lots of computer time for a high resolution model

Summary

Our model is

1. Multi-modal (SIA and MacAyeal)
2. Verifiable for each regime separately
3. Parallel

Further work

For the Mathematician

1. Variational Inequality approaches
2. Inverse problems
3. Existence and uniqueness for MacAyeal Equations
4. Prove anything about coupled systems

For the Physicist

1. Unified shallow model
2. Improved basal dynamics
3. Anisotropy