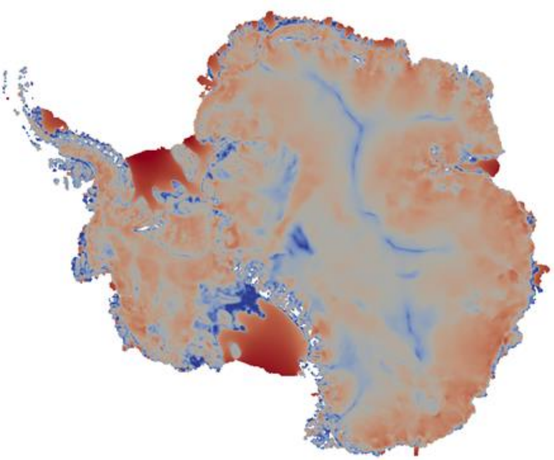
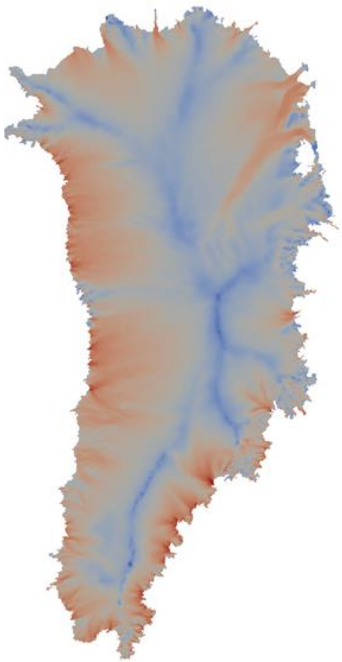
Computational Methods in Ice Sheet Modeling for Next-Generation Climate Simulations

Irina K. Tezaur
Quantitative Modeling & Analysis Department
Sandia National Laboratories
Livermore, CA



In collaboration with Andy Salinger, Mauro Perego, Ray Tuminaro, Steve Price, Matt Hoffman, Mike Eldred, John Jakeman and Irina Demeshko.



Monday, November 16, 2015
Duke University
Durham, NC

SAND2015-9389 PE

Sandia National Laboratories

Outline

- Motivation.
- The PISCEES project.
- The Ice Sheet Equations.
- The Albany/FELIX Steady Stress-Velocity Solver.
- Deterministic Inversion for Ice Sheet Initialization.
- Dynamic Simulations of Ice Sheet Evolution.
- Uncertainty Quantification.
- Summary & future work.

Outline

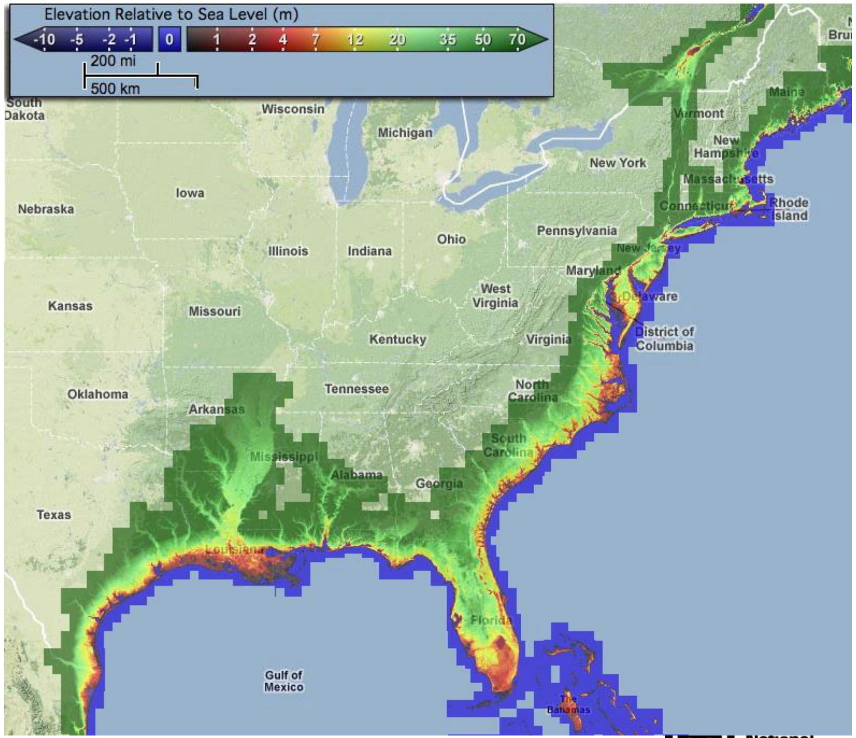
- **Motivation.**
- The PISCEES project.
- The Ice Sheet Equations.
- The Albany/FELIX Steady Stress-Velocity Solver.
- Deterministic Inversion for Ice Sheet Initialization.
- Dynamic Simulations of Ice Sheet Evolution.
- Uncertainty Quantification.
- Summary & future work.

Motivation

Sea-level rise has received a lot of media attention in recent years!

- Full deglaciation: sea level could rise up to ~65 meters*
- Potential contributions to sea level rise by ice sheet:
    - Greenland ice sheet: ~7 meters
    - East Antarctic ice sheet: ~53 meters
    - West Antarctic ice sheet: ~5 meters

*Estimates given by Prof. Richard Alley of Penn State who testified in 1999 about climate change to Al Gore.

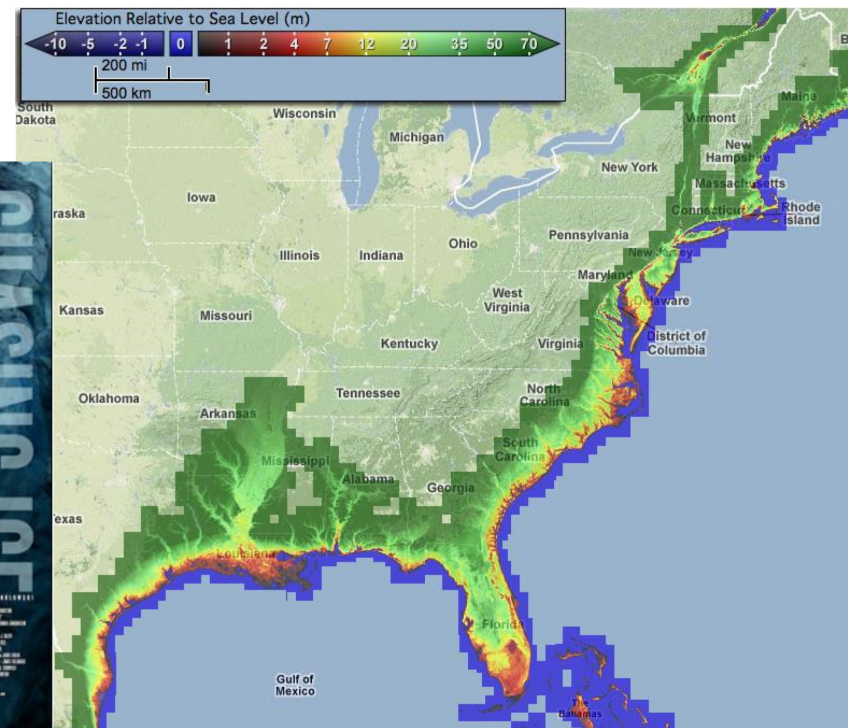


National Laboratories

Motivation

Sea-level rise has received a lot of media attention in recent years!

- 2012 Award Winning Film “**Chasing Ice**”: National Geographic photographer James Balog deploys time-lapse cameras to capture a multi-year record of the world's changing glaciers.

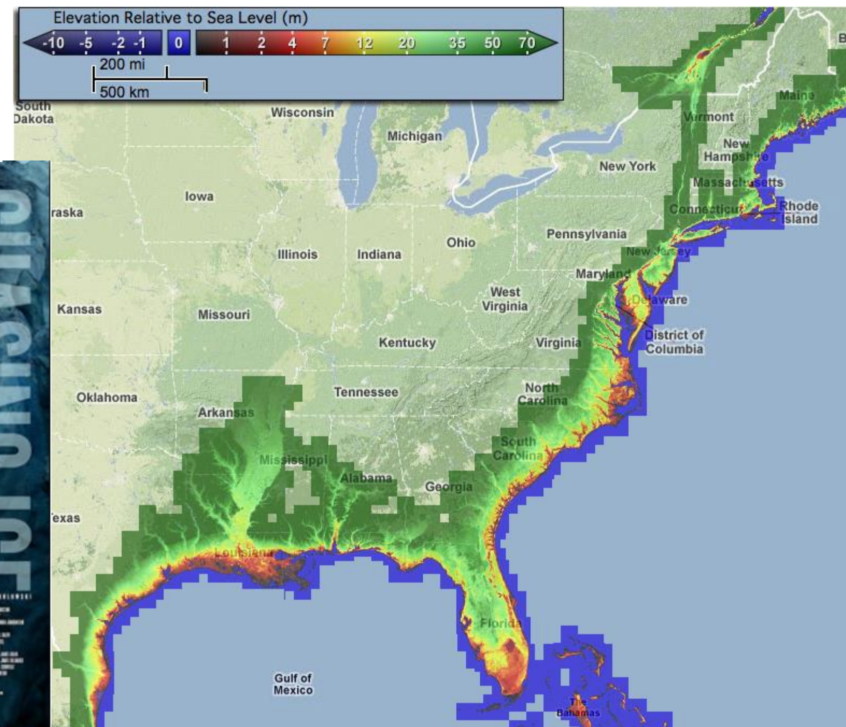


Sandia National Laboratories

Motivation

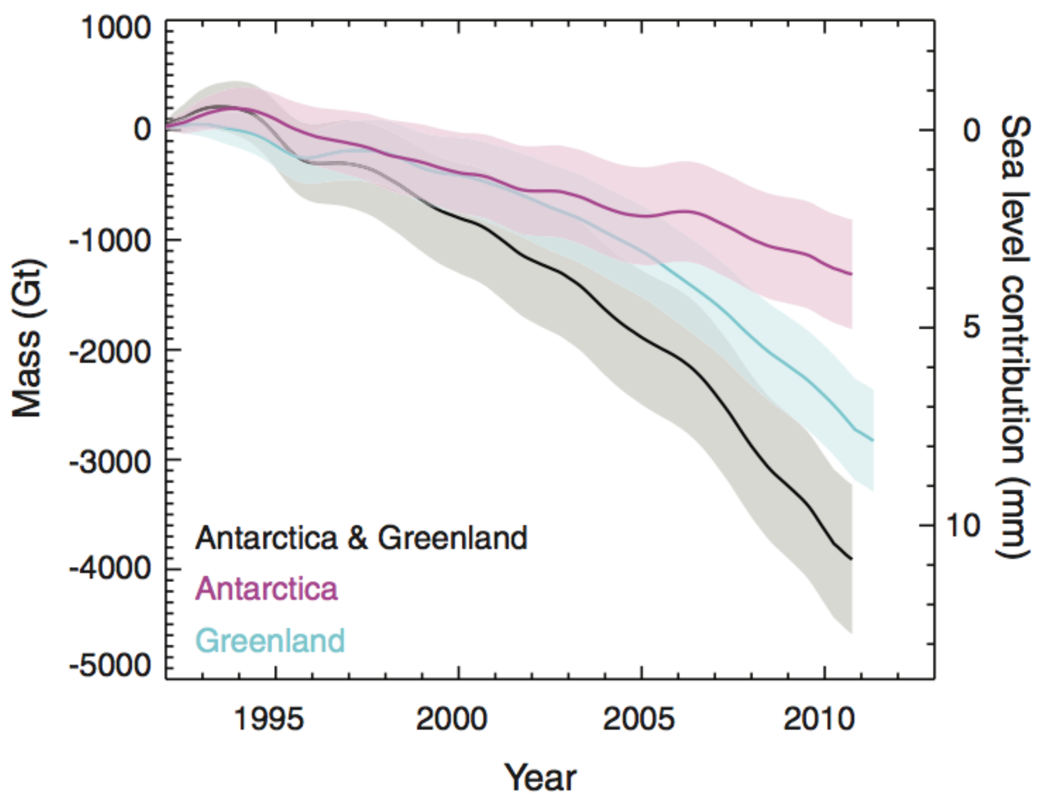
Sea-level rise has received a lot of media attention in recent years!

- **2015 Headlines:** “Former Top NASA Scientist Predicts Catastrophic Rise in Sea Levels”, “Earth’s Most Famous Climate Scientist Issues Bombshell Sea Level Warning”, “Climate Seer James Hansen Issues His Direst Forecast Yet”



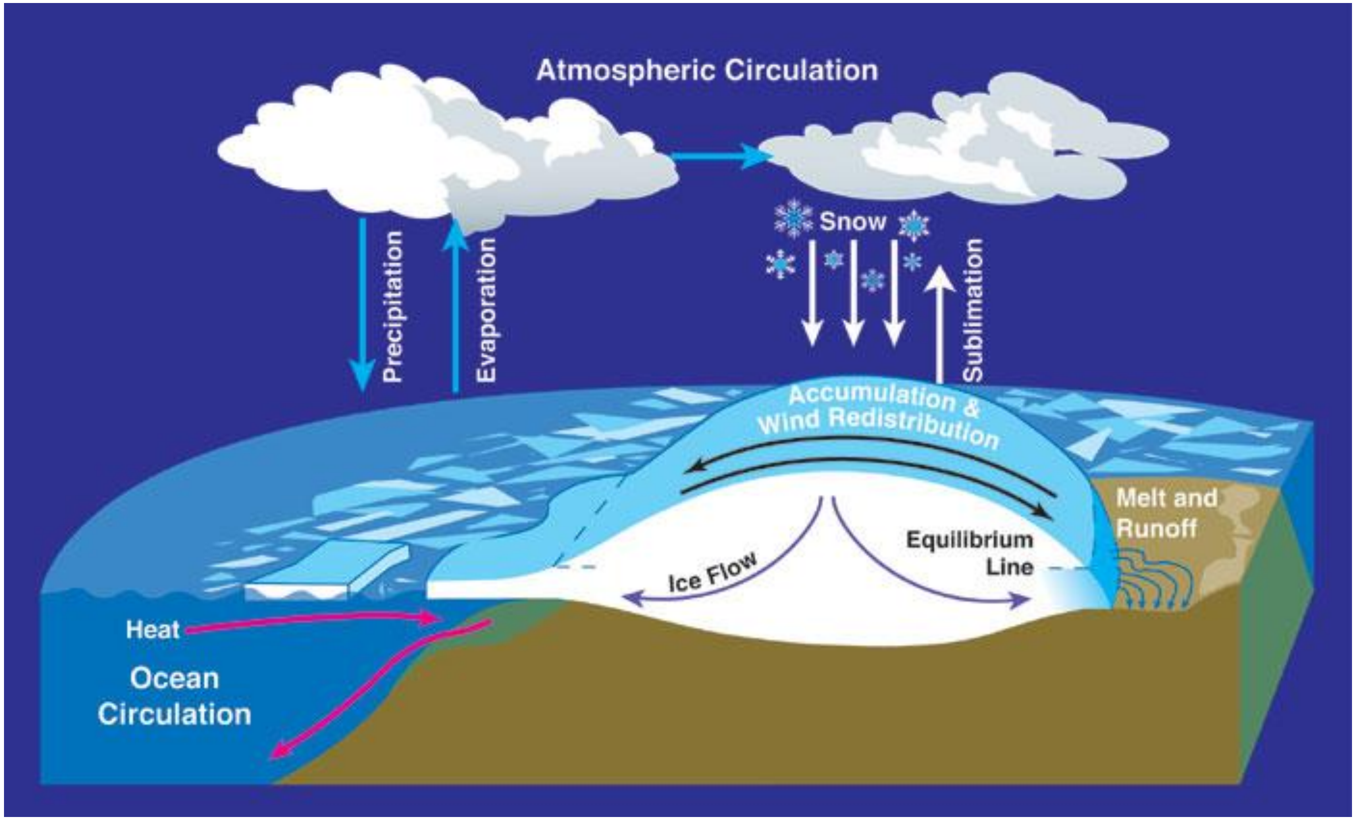
Motivation

*Mass loss from the Greenland & Antarctic ice sheets is **accelerating**!*



Earth System Models (ESMs)

Ice sheets are part of a **global** climate system.

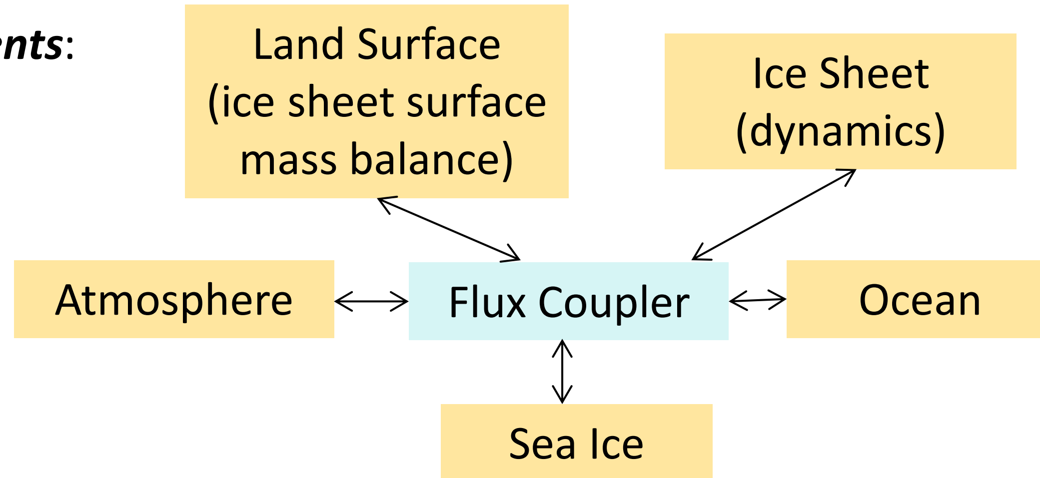


Mass balance: change in ice sheet mass = mass in – mass out

sea level change *snow fall* *melt,calving*

Earth System Models: CESM, DOE-ESM

- An ESM has ***six modular components***:
    1. Atmosphere model
    2. Ocean model
    3. Sea ice model
    4. Land ice model
    5. Land model
    6. Flux coupler



Goal of ESM: to provide actionable scientific predictions of 21st century sea-level rise (including uncertainty).

Climate Model passes:

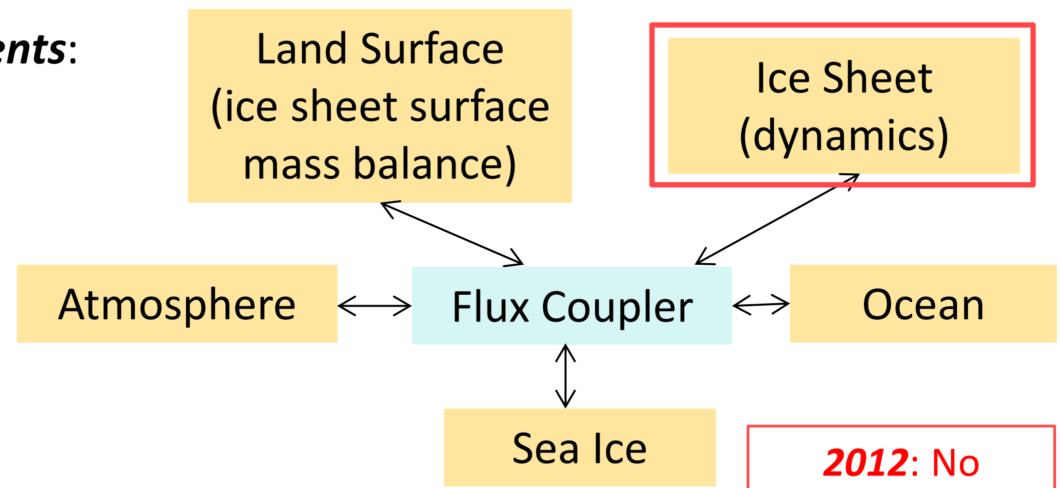
- Surface mass balance (SMB)
- Boundary temperatures
- Sub-shelf melting

Land Ice Model passes:

- Elevation
- Revised land ice distribution
- Oceanic heat and moisture fluxes (icebergs)
- Revised sub-shelf geometry

Earth System Models: CESM, DOE-ESM

- An ESM has ***six modular components***:
  1. Atmosphere model
  2. Ocean model
  3. Sea ice model
  4. Land ice model
  5. Land model
  6. Flux coupler



2012: No robust land ice model! ☹

Goal of ESM: to provide actionable scientific predictions of 21st century sea-level rise (including uncertainty).

<u>Climate Model passes:</u>

- Surface mass balance (SMB)
- Boundary temperatures
- Sub-shelf melting

<u>Land Ice Model passes:</u>

- Elevation
- Revised land ice distribution
- Oceanic heat and moisture fluxes (icebergs)
- Revised sub-shelf geometry

Outline

- Motivation.
- **The PISCEES project.**
- The Ice Sheet Equations.
- The Albany/FELIX Steady Stress-Velocity Solver.
- Deterministic Inversion for Ice Sheet Initialization.
- Dynamic Simulations of Ice Sheet Evolution.
- Uncertainty Quantification.
- Summary & future work.

The PISCEES Project

- In its fourth assessment report (AR4)* in 2007, the Intergovernmental Panel on Climate Change (IPCC) declined to include estimates of future sea-level rise from ice sheet dynamics due to the ***inability*** of ice sheet models to mimic or explain observed dynamic behaviors, e.g., the acceleration and thinning then occurring on several of Greenland's large outlet glaciers.

* Solomon, S., Qin, D., Manning, M., Chen, Z., Marquis, M., Averyt, K., Tignor, M., and Miller, H. "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 Univ. Press, Cambridge, UK, 2007.

The PISCEES Project

- In its fourth assessment report (AR4)* in 2007, the Intergovernmental Panel on Climate Change (IPCC) declined to include estimates of future sea-level rise from ice sheet dynamics due to the ***inability*** of ice sheet models to mimic or explain observed dynamic behaviors, e.g., the acceleration and thinning then occurring on several of Greenland's large outlet glaciers.
- Although ice sheet models have improved in recent years, ***much work is needed*** to make these models robust and efficient on continental scales and to quantify uncertainties in their projected outputs.

* Solomon, S., Qin, D., Manning, M., Chen, Z., Marquis, M., Averyt, K., Tignor, M., and Miller, H. "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 Univ. Press, Cambridge, UK, 2007.

The PISCEES Project

- In its fourth assessment report (AR4)* in 2007, the Intergovernmental Panel on Climate Change (IPCC) declined to include estimates of future sea-level rise from ice sheet dynamics due to the ***inability*** of ice sheet models to mimic or explain observed dynamic behaviors, e.g., the acceleration and thinning then occurring on several of Greenland's large outlet glaciers.
- Although ice sheet models have improved in recent years, ***much work is needed*** to make these models robust and efficient on continental scales and to quantify uncertainties in their projected outputs.

PISCEES (**P**redicting **I**ce **S**heet **C**limate & **E**volution at **E**xtreme **S**cales) aims to:

1. Develop/apply ***robust***, ***accurate***, ***scalable*** dynamical cores (dycores) for ice sheet modeling on structured and unstructured meshes.
2. Evaluate models using new tools and data sets for verification/validation and ***uncertainty quantification***.
3. Integrate models/tools into DOE-supported ***Earth System Models***.

* Solomon, S., Qin, D., Manning, M., Chen, Z., Marquis, M., Averyt, K., Tignor, M., and Miller, H. "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 Univ. Press, Cambridge, UK, 2007.

The PISCEES Project (cont'd)

PISCEES
SciDAC Application Partnership
(DOE's BER + ASCR divisions)
Start date: June 2012
5 years

3 land-ice dycores developed under PISCEES

FSU FELIX
FSU
Finite Element
Full Stokes Model

Albany/FELIX
SNL
Finite Element
"First Order" Stokes Model

BISICLES
LBNL
Finite Volume
L1L2 Model

Increased fidelity

PISCEES: Predicting Ice Sheet Climate & Evolution at Extreme Scales
FELIX: Finite Elements for Land Ice eXperiments
BISICLES: Berkeley Ice Sheet Initiative for Climate at Extreme Scales

The PISCEES Project (cont'd)

PISCEES
SciDAC Application Partnership
(DOE's BER + ASCR divisions)
Start date: June 2012
5 years

3 land-ice dycores developed under PISCEES

FSU FELIX
FSU
Finite Element
Full Stokes Model

Albany/FELIX
SNL
Finite Element
"First Order" Stokes Model

BISICLES
LBNL
Finite Volume
L1L2 Model

Increased fidelity

PISCEES: Predicting Ice Sheet Climate & Evolution at Extreme Scales
FELIX: Finite Elements for Land Ice eXperiments
BISICLES: Berkeley Ice Sheet Initiative for Climate at Extreme Scales

Outline

- Motivation.
- The PISCEES project.
- **The Ice Sheet Equations.**
- The Albany/FELIX Steady Stress-Velocity Solver.
- Deterministic Inversion for Ice Sheet Initialization.
- Dynamic Simulations of Ice Sheet Evolution.
- Uncertainty Quantification.
- Summary & future work.

Stokes Ice Flow Equations

Ice behaves like a very viscous shear-thinning fluid (similar to lava flow) and is modeled using nonlinear incompressible Stokes' equations.

- **<u>Nonlinear incompressible Stokes' ice flow equations</u>** <u>(momentum balance):</u>

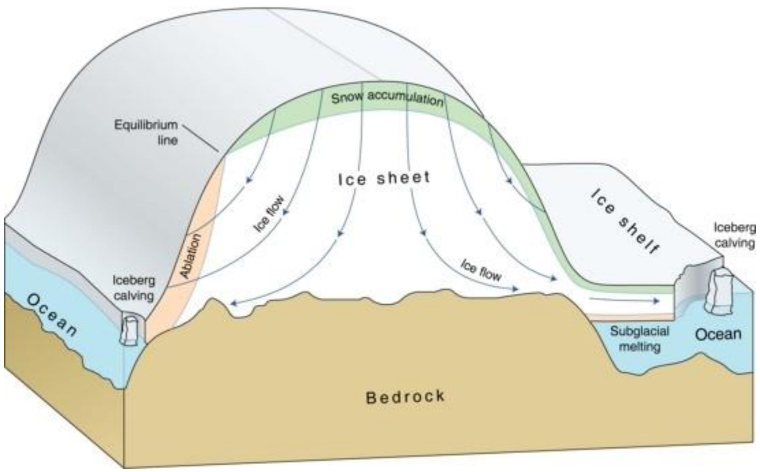
$$\begin{cases} -\nabla \cdot \boldsymbol{\sigma} = \rho \boldsymbol{g} \\ -\nabla \cdot u = 0 \end{cases}, \quad \text{in } \Omega$$

with

$$\boldsymbol{\sigma} = 2\mu\, \dot{\boldsymbol{\epsilon}} - p\boldsymbol{I}, \qquad \epsilon_{\mathrm{ij}} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

and nonlinear ***"Glen's law"*** viscosity

$$\mu = \frac{1}{2} A(T)^{-\frac{1}{n}} \left(\frac{1}{2}\sum_{ij} \boldsymbol{\epsilon}_{ij}^{\,2}\right)^{\left(\frac{1}{2n}-\frac{1}{2}\right)}, \quad n = 3.$$



Stokes Ice Flow Equations

Ice behaves like a very viscous shear-thinning fluid (similar to lava flow) and is modeled using nonlinear incompressible Stokes' equations.

- **Nonlinear incompressible Stokes' ice flow equations** (momentum balance):

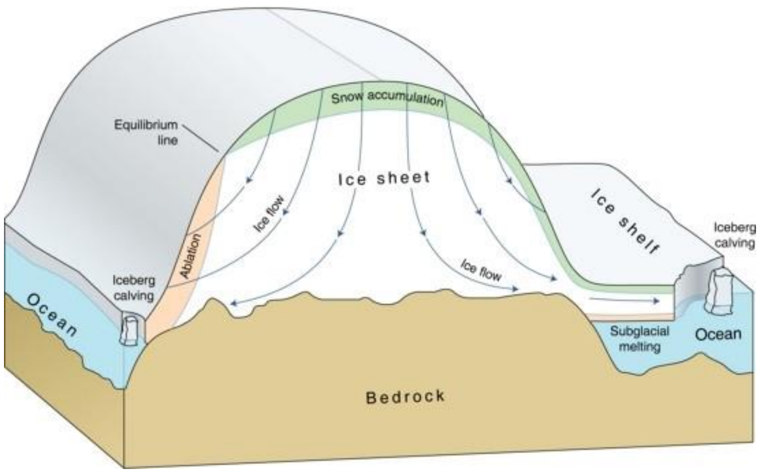
$$\begin{cases} -\nabla \cdot \boldsymbol{\sigma} = \rho \boldsymbol{g} \\ -\nabla \cdot u = 0 \end{cases}, \quad \text{in } \Omega$$

with

$$\boldsymbol{\sigma} = 2\mu \, \dot{\boldsymbol{\epsilon}} - p\boldsymbol{I}, \qquad \epsilon_{\mathrm{ij}} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

and nonlinear ***"Glen's law"*** viscosity

$$\mu = \frac{1}{2} A(T)^{-\frac{1}{n}} \left(\frac{1}{2}\sum_{ij} \boldsymbol{\epsilon}_{ij}{}^2\right)^{\left(\frac{1}{2n}-\frac{1}{2}\right)}, \quad n = 3.$$



→"nasty" saddle point problem

The First-Order Stokes Model

- In our model, ice sheet dynamics are given by ***"First-Order" Stokes PDEs***: "nice" elliptic approximation* to Stokes' flow equations.

$$\begin{cases} -\nabla \cdot (2\mu \dot{\boldsymbol{\epsilon}}_1) = -\rho g \dfrac{\partial s}{\partial x} \\ -\nabla \cdot (2\mu \dot{\boldsymbol{\epsilon}}_2) = -\rho g \dfrac{\partial s}{\partial y} \end{cases}, \quad \text{in } \Omega$$

$$\dot{\boldsymbol{\epsilon}}_1{}^T = (2\dot{\epsilon}_{11} + \dot{\epsilon}_{22}, \dot{\epsilon}_{12}, \dot{\epsilon}_{13})$$
$$\dot{\boldsymbol{\epsilon}}_2{}^T = (2\dot{\epsilon}_{12}, \dot{\epsilon}_{11} + 2\dot{\epsilon}_{22}, \dot{\epsilon}_{23})$$
$$\dot{\epsilon}_{\mathrm{ij}} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

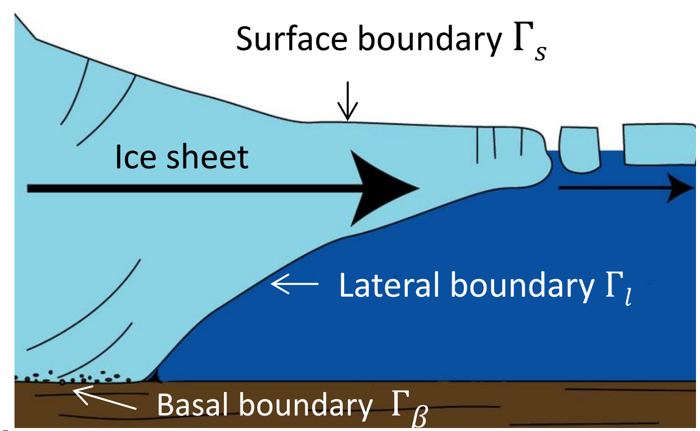
- Viscosity μ is nonlinear function given by "***Glen's law***":

$$\mu = \frac{1}{2} A(T)^{-\frac{1}{n}} \left(\frac{1}{2}\sum_{ij} \dot{\boldsymbol{\epsilon}}_{ij}{}^2\right)^{\left(\frac{1}{2n} - \frac{1}{2}\right)} \qquad (n = 3)$$

- Relevant boundary conditions:
  - ***Stress-free BC:*** $2\mu\dot{\boldsymbol{\epsilon}}_i \cdot \boldsymbol{n} = 0$, on Γ_s
  - ***Floating ice BC:***

$$2\mu\dot{\boldsymbol{\epsilon}}_i \cdot \boldsymbol{n} = \begin{cases} \rho g z \boldsymbol{n}, & \text{if } z > 0 \\ 0, & \text{if } z \le 0 \end{cases}, \text{ on } \Gamma_l$$

  - ***Basal sliding BC:*** $2\mu\dot{\boldsymbol{\epsilon}}_i \cdot \boldsymbol{n} + \beta u_i = 0$, on Γ_β



$$\beta = \text{sliding coefficient} \ge 0$$

***Assumption:** aspect ratio δ is small and normals to upper/lower surfaces are almost vertical.

Sandia National Laboratories

Importance of Boundary Conditions!

Boundary conditions have tremendous effect on ice sheet dynamics!

Importance of Boundary Conditions!

Boundary conditions have tremendous effect on ice sheet dynamics!

- ***Basal sliding BC:*** $2\mu\dot{\boldsymbol{\epsilon}}_i \cdot \boldsymbol{n} + \beta u_i = 0$, on Γ_β
  - $\beta = \beta(x, y) =$ measure of friction
    - Large $\beta \Rightarrow$ a lot of friction $\Rightarrow$ no-slip: $u_i = 0 \Rightarrow$ frozen ice (does not move).
    - Small $\beta \Rightarrow$ not much friction $\Rightarrow$ ice moves a lot!
  - Cannot be measured directly, and must be estimated from data (e.g., by solving an inverse problem).

Sandia National Laboratories

Importance of Boundary Conditions!

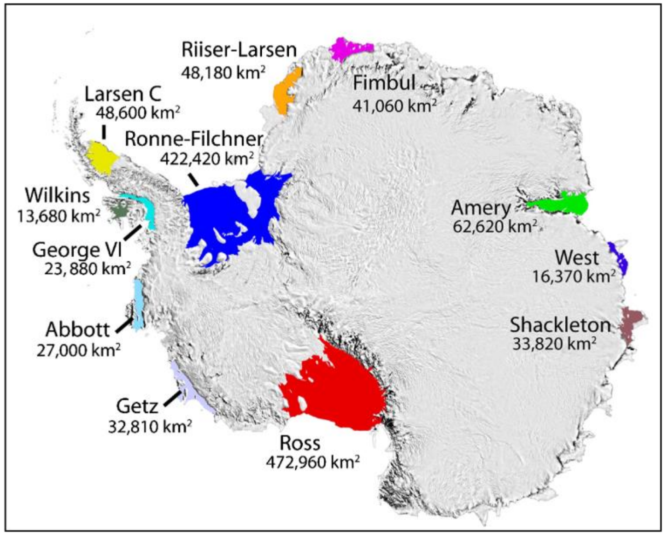
Boundary conditions have tremendous effect on ice sheet dynamics!

- ***Basal sliding BC:*** $2\mu\dot{\boldsymbol{\epsilon}}_i \cdot \boldsymbol{n} + \beta u_i = 0$, on Γ_β
  - $\beta = \beta(x, y) =$ measure of friction
    - Large $\beta \Rightarrow$ a lot of friction $\Rightarrow$ no-slip: $u_i = 0 \Rightarrow$ frozen ice (does not move).
    - Small $\beta \Rightarrow$ not much friction $\Rightarrow$ ice moves a lot!
  - Cannot be measured directly, and must be estimated from data (e.g., by solving an inverse problem).

- ***Floating ice BC:***

$$2\mu\dot{\boldsymbol{\epsilon}}_i \cdot \boldsymbol{n} = \begin{cases} \rho g z \boldsymbol{n}, & \text{if } z > 0 \\ 0, & \text{if } z \le 0 \end{cases}, \text{ on } \Gamma_l$$

  - Floating ice = "ice shelves" ⇒ Antarctica
  - Ice shelves buttress interior ice ⇒ can cause a lot of sea-level rise (SLR) in short period.



IPCC WG1 (2013): "*Based on current understanding, only the collapse of marine-based sectors of the Antarctic ice sheet, if initiated, could cause [SLR by 2100] substantially above the likely range [of ~0.5-1 m].*"

Sandia National Laboratories

Thickness & Temperature Equations

- Model for ***evolution of the boundaries*** (thickness evolution equation):

$$\frac{\partial H}{\partial t} = -\nabla \cdot (\bar{\boldsymbol{u}} H) + \dot{b}$$

where $\bar{\boldsymbol{u}}$ = vertically averaged velocity, $\dot{b}$ = surface mass balance (conservation of mass).

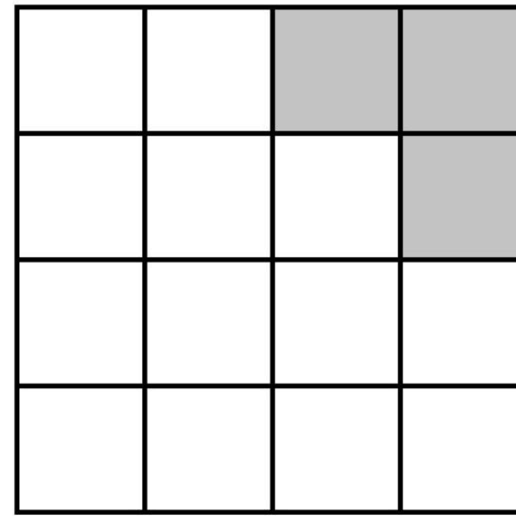
- ***Temperature equation*** (advection-diffusion):

$$\rho c \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) - \rho c \boldsymbol{u} \cdot \nabla T + 2\dot{\boldsymbol{\epsilon}} \boldsymbol{\sigma}$$

(energy balance).

- ***Flow factor*** A in Glen's law depends on temperature T: $A = A(T)$.

- Ice sheet ***grows/retreats*** depending on thickness H.



time t_0

Ice-covered ("active") cells shaded in white ($H > H_{min}$)

Sandia National Laboratories

Thickness & Temperature Equations

- Model for ***evolution of the boundaries*** (thickness evolution equation):

$$\frac{\partial H}{\partial t} = -\nabla \cdot (\bar{\boldsymbol{u}} H) + \dot{b}$$

where $\bar{\boldsymbol{u}}$ = vertically averaged velocity, $\dot{b}$ = surface mass balance (conservation of mass).

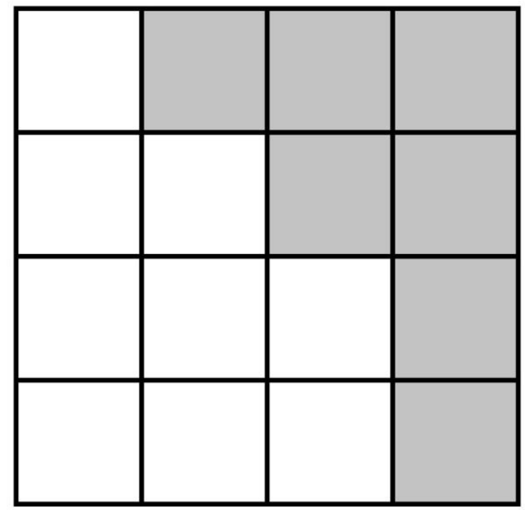
- ***Temperature equation*** (advection-diffusion):

$$\rho c \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) - \rho c \boldsymbol{u} \cdot \nabla T + 2 \dot{\boldsymbol{\epsilon}} \boldsymbol{\sigma}$$

(energy balance).

- ***Flow factor*** A in Glen's law depends on temperature T: $A = A(T)$.

- Ice sheet ***grows/retreats*** depending on thickness H.



time t_1

Ice-covered ("active") cells shaded in white ($H > H_{min}$)

Thickness & Temperature Equations

- Model for ***evolution of the boundaries*** (thickness evolution equation):

$$\frac{\partial H}{\partial t} = -\nabla \cdot (\bar{\boldsymbol{u}} H) + \dot{b}$$

where $\bar{\boldsymbol{u}}$ = vertically averaged velocity, $\dot{b}$ = surface mass balance (conservation of mass).

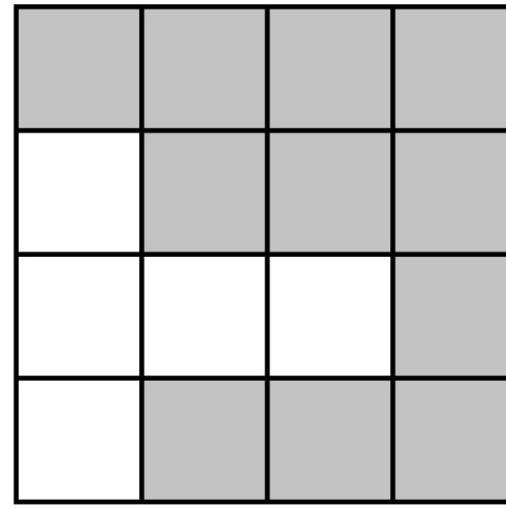
- ***Temperature equation*** (advection-diffusion):

$$\rho c \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) - \rho c \boldsymbol{u} \cdot \nabla T + 2\dot{\boldsymbol{\epsilon}} \boldsymbol{\sigma}$$

(energy balance).

- ***Flow factor*** A in Glen's law depends on temperature T: $A = A(T)$.

- Ice sheet ***grows/retreats*** depending on thickness H.



time t_2

Ice-covered ("active") cells shaded in white ($H > H_{min}$)

Outline

- Motivation.
- The PISCEES project.
- The Ice Sheet Equations.
- **The Albany/FELIX Steady Stress-Velocity Solver.**
- Deterministic Inversion for Ice Sheet Initialization.
- Dynamic Simulations of Ice Sheet Evolution.
- Uncertainty Quantification.
- Summary & future work.

Algorithmic Choices

<u>Sandia's Role in the PISCEES Project:</u> to **develop** and **support** a robust and scalable land ice solver based on the "First-Order" (FO) Stokes approximation

Sandia National Laboratories

Algorithmic Choices

<u>Sandia's Role in the PISCEES Project:</u> to **develop** and **support** a robust and scalable land ice solver based on the "First-Order" (FO) Stokes approximation

<u>Objectives:</u> to create a solver that

- Is scalable, fast, robust.
- Becomes a dynamical core (dycore) when coupled to codes that solve thickness and temperature evolution equations for integration in ESMs.
- Possesses advanced analysis capabilities (adjoint-based deterministic inversion, Bayesian calibration, UQ, sensitivity analysis).
- Is portable to new/emerging architecture machines (multi-core, many-core, GPUs).

Algorithmic Choices

<u>Sandia's Role in the PISCEES Project:</u> to **develop** and **support** a robust and scalable land ice solver based on the "First-Order" (FO) Stokes approximation

<u>Objectives:</u> to create a solver that

- Is scalable, fast, robust.
- Becomes a dynamical core (dycore) when coupled to codes that solve thickness and temperature evolution equations for integration in ESMs.
- Possesses advanced analysis capabilities (adjoint-based deterministic inversion, Bayesian calibration, UQ, sensitivity analysis).
- Is portable to new/emerging architecture machines (multi-core, many-core, GPUs).

Our algorithmic choices aligned with our objectives!

Algorithmic Choices

<u>Sandia's Role in the PISCEES Project:</u> to **develop** and **support** a robust and scalable land ice solver based on the "First-Order" (FO) Stokes approximation

<u>Objectives:</u> to create a solver that

- Is scalable, fast, robust.
- Becomes a dynamical core (dycore) when coupled to codes that solve thickness and temperature evolution equations for integration in ESMs.
- Possesses advanced analysis capabilities (adjoint-based deterministic inversion, Bayesian calibration, UQ, sensitivity analysis).
- Is portable to new/emerging architecture machines (multi-core, many-core, GPUs).

Our algorithmic choices aligned with our objectives!

<u>Algorithms:</u>

- Component-based code-development approach.
- Finite element method discretization.
- Newton nonlinear solver with automatic differentiation Jacobians.
- Preconditioned iterative methods for linear solves.

Components-Based Code Development Approach

New land-ice solver developed using ***Trilinos**** libraries for everything but PDE description.

Trilinos

Linear Algebra
- Data Structures
- Iterative Solvers
- Direct Solvers
- Eigen Solver
- Preconditioners
- Multi-Level Methods

Analysis Tools *(embedded)*
- Nonlinear Solver
- Time Integration
- Continuation
- Sensitivity Analysis
- Stability Analysis
- Optimization
- UQ Solver

DAKOTA

Analysis Tools *(black-box)*
- Optimization
- UQ (sampling)
- Parameter Studies
- Bayesian Calibration
- Reliability

Mesh Tools
- Mesh Database
- Mesh I/O
- Inline Meshing
- Partitioning
- Load Balancing
- Adaptivity
- DOF map

Discretizations
- Discretization Library
- Field Manager

Derivative Tools
- Sensitivities
- Derivatives
- Adjoints
- UQ / PCE Propagation

Utilities
- Input File Parser
- Parameter List
- Memory Management
- I/O Management
- Communicators

PostProcessing
- Visualization
- Verification
- Model Reduction

*40+ packages; 120+ libraries; www.trilinos.org.

Albany/FELIX First-Order Stokes Solver

The ***Albany*/FELIX***** First Order Stokes dycore is implemented in a Sandia (open-source) parallel C++ finite element code called…

Started by A. Salinger

Albany

Land Ice Physics Set (***Albany/FELIX code***)

Other Albany Physics Sets

"Agile Components"

- Discretizations/meshes
- Solver libraries
- Preconditioners
- Automatic differentiation
- Many others!

Trilinos

- Parameter estimation
- Uncertainty quantification
- Optimization
- Bayesian inference

DAKOTA

- Configure/build/test/documentation

*Open-source code available on github: https://github.com/gahansen/Albany.
** "FELIX" = Finite Elements for Land-Ice eXperiments.

Sandia National Laboratories

Albany/FELIX First-Order Stokes Solver

The ***Albany*/FELIX***** First Order Stokes dycore is implemented in a Sandia (open-source) parallel C++ finite element code called…

Started by A. Salinger

Albany

Land Ice Physics Set (***Albany/FELIX code***)

Other Albany Physics Sets

"Agile Components"

- Discretizations/meshes
- Solver libraries
- Preconditioners
- Automatic differentiation
- Many others!

Trilinos

- Parameter estimation
- Uncertainty quantification
- Optimization
- Bayesian inference

DAKOTA

- Configure/build/test/documentation

*Use of **Trilinos** components has enabled the **rapid** development of the **Albany/FELIX** First Order Stokes dycore (~**3 FTEs** for all of work shown!).*

*Open-source code available on github: https://github.com/gahansen/Albany.

** "FELIX" = Finite Elements for Land-Ice eXperiments.

Sandia National Laboratories

Discretization

<u>Discretization:</u> unstructured grid finite element method (FEM)

- Can handle readily complex geometries.
- Natural treatment of stress boundary conditions.
- Enables regional refinement/unstructured meshes.
- Wealth of software and algorithms.

Discretization

Discretization: unstructured grid finite element method (FEM)

- Can handle readily complex geometries.
- Natural treatment of stress boundary conditions.
- Enables regional refinement/unstructured meshes.
- Wealth of software and algorithms.

Example: finite difference discretization of basal BC via central differences

$$\mu \frac{\partial u}{\partial z} + \beta u = 0, \qquad \beta \gg 1 \qquad \Longrightarrow \qquad \begin{pmatrix} -\frac{\mu}{2h} & \beta & \frac{\mu}{2h} & 0 & 0 & 0 \\ \frac{\mu}{h^2} & -\frac{2\mu}{h^2} & \frac{\mu}{h^2} & 0 & 0 & 0 \\ 0 & \frac{\mu}{h^2} & -\frac{2\mu}{h^2} & \frac{\mu}{h^2} & 0 & 0 \\ 0 & 0 & \frac{\mu}{h^2} & -\frac{2\mu}{h^2} & \frac{\mu}{h^2} & 0 \\ 0 & 0 & 0 & \frac{\mu}{h^2} & -\frac{2\mu}{h^2} & \frac{\mu}{h^2} \end{pmatrix}$$

Discretization

<u>Discretization:</u> unstructured grid finite element method (FEM)

- Can handle readily complex geometries.
- Natural treatment of stress boundary conditions.
- Enables regional refinement/unstructured meshes.
- Wealth of software and algorithms.

<u>Example:</u> finite difference discretization of basal BC via central differences

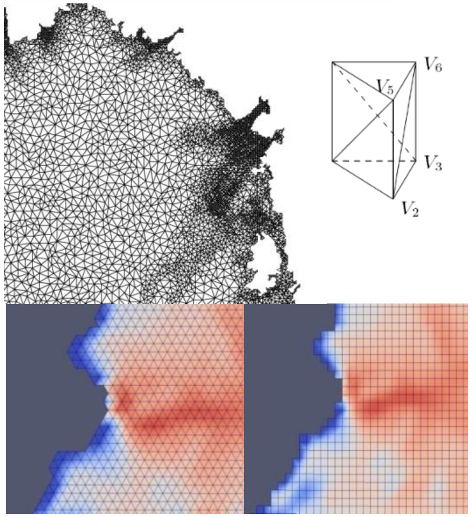
$$\mu \frac{\partial u}{\partial z} + \beta u = 0, \qquad \boxed{\beta \gg 1} \qquad \Longrightarrow \qquad \begin{pmatrix} -\frac{\mu}{2h} & \beta & \frac{\mu}{2h} & 0 & 0 & 0 \\ \frac{\mu}{h^2} & -\frac{2\mu}{h^2} & \frac{\mu}{h^2} & 0 & 0 & 0 \\ 0 & \frac{\mu}{h^2} & -\frac{2\mu}{h^2} & \frac{\mu}{h^2} & 0 & 0 \\ 0 & 0 & \frac{\mu}{h^2} & -\frac{2\mu}{h^2} & \frac{\mu}{h^2} & 0 \\ 0 & 0 & 0 & \frac{\mu}{h^2} & -\frac{2\mu}{h^2} & \frac{\mu}{h^2} \end{pmatrix}$$

Matrix has large off-diagonal entries ⇒ very difficult to solve (ill-conditioning)!

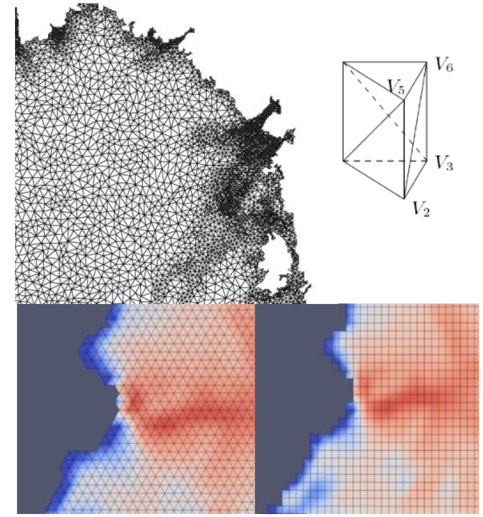
Meshes and Data

<u>**Meshes:**</u> can use any mesh but interested specifically in

- ***Structured hexahedral*** meshes (compatible with *CISM*).
- ***Tetrahedral*** meshes (compatible with *MPAS LI*)
    - ***Unstructured Delaunay triangle*** meshes with regional refinement based on gradient of surface velocity.
- All meshes are extruded (structured) in vertical direction as tetrahedra or hexahedra.

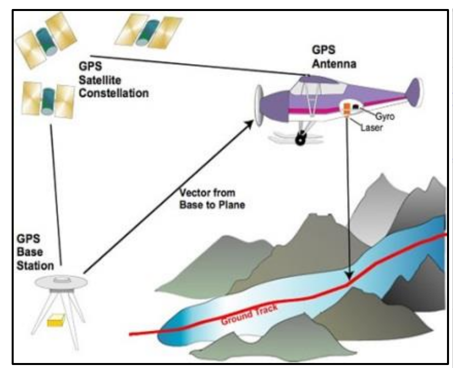


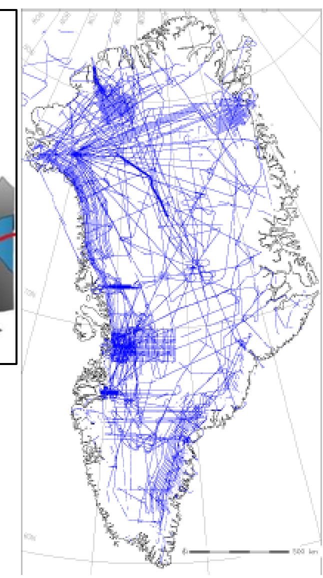
Meshes and Data



Meshes: can use any mesh but interested specifically in

- ***Structured hexahedral*** meshes (compatible with *CISM*).
- ***Tetrahedral*** meshes (compatible with *MPAS LI*)
    - ***Unstructured Delaunay triangle*** meshes with regional refinement based on gradient of surface velocity.
- All meshes are extruded (structured) in vertical direction as tetrahedra or hexahedra.





Data: needs to be imported into code to run "real" problems (Greenland, Antarctica).

- ***Surface data*** are available from measurements (satellite infrarometry, radar, altimetry): ice extent, surface topography, surface velocity, surface mass balance.
- ***Interior ice data*** (ice thickness, basal friction) cannot be measured; estimated by solving an inverse problem.

Nonlinear & Linear Solvers

Nonlinear solver: full Newton with analytic (automatic differentiation) derivatives and homotopy continuation

- Most robust and efficient for steady-state solves.
- Jacobian available for preconditioners and matrix-vector products.
- Analytic sensitivity analysis.
- Analytic gradients for inversion.

Linear solver: preconditioned iterative method

- **Solvers:** Conjugate Gradient (CG) or GMRES
- **Preconditioners:** ILU or algebraic multi-grid (AMG)

Nonlinear Solve for $\boldsymbol{f}(\boldsymbol{x}) = 0$ (Newton) → Automatic Differentiation Jacobian: $\boldsymbol{J} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}$ → Preconditioned Iterative Linear Solve (CG or GMRES): Solve $\boldsymbol{J}\boldsymbol{x} = \boldsymbol{r}$ → (back to Nonlinear Solve)

Automatic Differentiation (Sacado)

Automatic Differentiation (AD) provides exact derivatives w/o time/effort of deriving and hand-coding them!

Automatic Differentiation (Sacado)

Automatic Differentiation (AD) provides exact derivatives w/o time/effort of deriving and hand-coding them!

- How does AD work? → freshman calculus!
  - Computations are composition of simple operations (+, *, $\sin()$, etc.)
  - Derivatives computed line by line then combined via chain rule.

Automatic Differentiation Example:

$$y = \sin(e^x + x\log x), \quad x = 2$$

		$\frac{d}{dx}$
$x \leftarrow 2$	$\frac{dx}{dx} \leftarrow 1$	1.000
$t \leftarrow e^x$	$\frac{dt}{dx} \leftarrow t\frac{dx}{dx}$	7.389
$u \leftarrow \log x$	$\frac{du}{dx} \leftarrow \frac{1}{x}\frac{dx}{dx}$	0.500
$v \leftarrow xu$	$\frac{dv}{dx} \leftarrow u\frac{dx}{dx} + x\frac{du}{dx}$	1.301
$w \leftarrow t + v$	$\frac{dw}{dx} \leftarrow \frac{dt}{dx} + \frac{dv}{dx}$	8.690
$y \leftarrow \sin w$	$\frac{dy}{dx} \leftarrow \cos(w)\frac{dw}{dx}$	-1.188

Automatic Differentiation (Sacado)

Automatic Differentiation (AD) provides exact derivatives w/o time/effort of deriving and hand-coding them!

- How does AD work? → freshman calculus!
  - Computations are composition of simple operations (+, *, $\sin()$, etc.)
  - Derivatives computed line by line then combined via chain rule.
- Derivatives are as accurate as analytic computation – no finite difference truncation error!

Automatic Differentiation Example:

$$y = \sin(e^x + x\log x), \quad x = 2$$

		$\frac{d}{dx}$
$x \leftarrow 2$	$\frac{dx}{dx} \leftarrow 1$	1.000
$t \leftarrow e^x$	$\frac{dt}{dx} \leftarrow t\frac{dx}{dx}$	7.389
$u \leftarrow \log x$	$\frac{du}{dx} \leftarrow \frac{1}{x}\frac{dx}{dx}$	0.500
$v \leftarrow xu$	$\frac{dv}{dx} \leftarrow u\frac{dx}{dx} + x\frac{du}{dx}$	1.301
$w \leftarrow t + v$	$\frac{dw}{dx} \leftarrow \frac{dt}{dx} + \frac{dv}{dx}$	8.690
$y \leftarrow \sin w$	$\frac{dy}{dx} \leftarrow \cos(w)\frac{dw}{dx}$	-1.188

Sandia National Laboratories

Automatic Differentiation (Sacado)

Automatic Differentiation (AD) provides exact derivatives w/o time/effort of deriving and hand-coding them!

- How does AD work? → freshman calculus!
  - Computations are composition of simple operations (+, *, $\sin()$, etc.)
  - Derivatives computed line by line then combined via chain rule.
- Derivatives are as accurate as analytic computation – no finite difference truncation error!
- Great for multi-physics codes (e.g., many Jacobians) and advanced analysis (e.g., sensitivities)

Automatic Differentiation Example:

$$y = \sin(e^x + x \log x), \quad x = 2$$

		$\frac{d}{dx}$
$x \leftarrow 2$	$\frac{dx}{dx} \leftarrow 1$	1.000
$t \leftarrow e^x$	$\frac{dt}{dx} \leftarrow t\frac{dx}{dx}$	7.389
$u \leftarrow \log x$	$\frac{du}{dx} \leftarrow \frac{1}{x}\frac{dx}{dx}$	0.500
$v \leftarrow xu$	$\frac{dv}{dx} \leftarrow u\frac{dx}{dx} + x\frac{du}{dx}$	1.301
$w \leftarrow t + v$	$\frac{dw}{dx} \leftarrow \frac{dt}{dx} + \frac{dv}{dx}$	8.690
$y \leftarrow \sin w$	$\frac{dy}{dx} \leftarrow \cos(w)\frac{dw}{dx}$	-1.188

Automatic Differentiation (Sacado)

Automatic Differentiation (AD) provides exact derivatives w/o time/effort of deriving and hand-coding them!

- How does AD work? → freshman calculus!
  - Computations are composition of simple operations (+, *, sin(), etc.)
  - Derivatives computed line by line then combined via chain rule.
- Derivatives are as accurate as analytic computation – no finite difference truncation error!
- Great for multi-physics codes (e.g., many Jacobians) and advanced analysis (e.g., sensitivities)
- There are many AD libraries (C++, Fortran, MATLAB, etc.) that can be used (https://en.wikipedia.org/wiki/Automatic_differentiation) → we use Trilinos package Sacado.

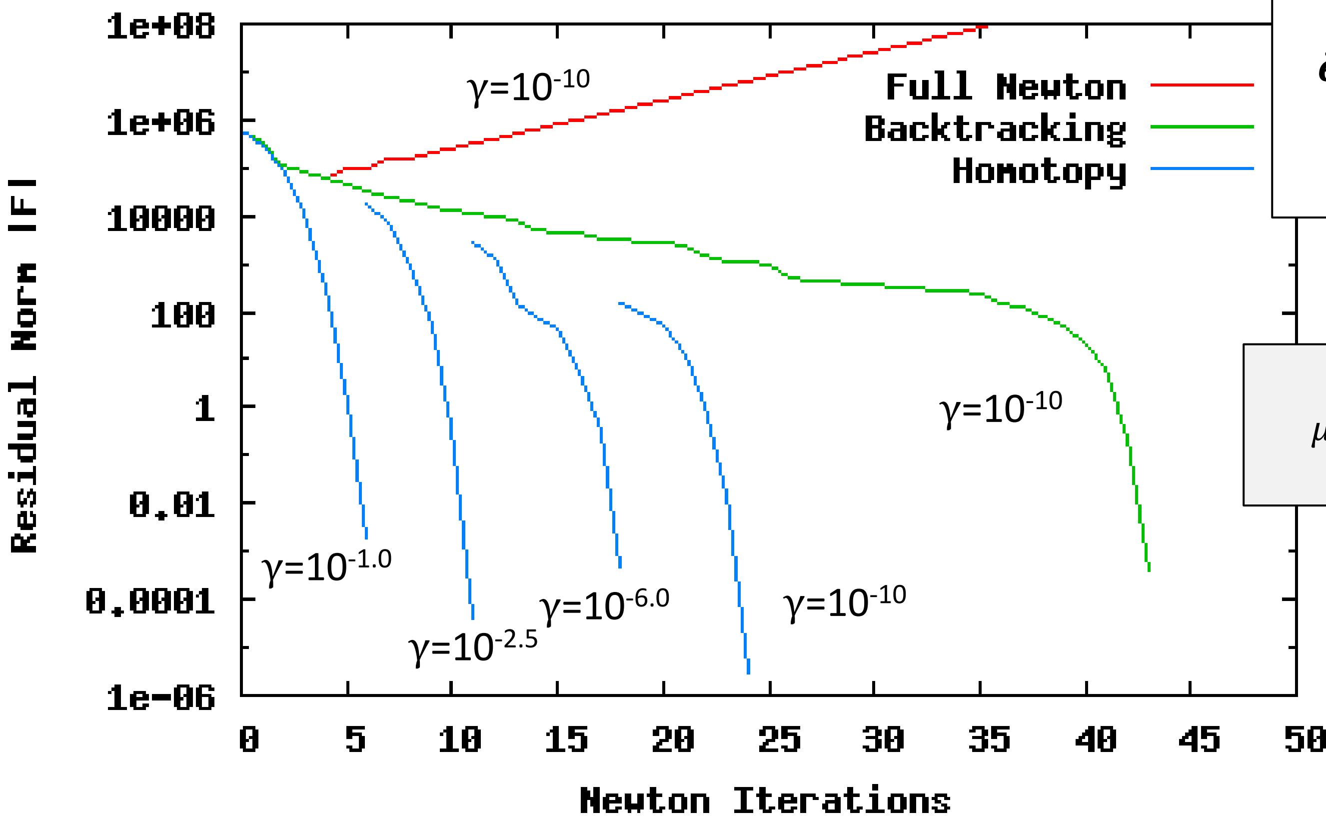
Automatic Differentiation Example:

$$y = \sin(e^x + x\log x), \quad x = 2$$

		$\frac{d}{dx}$
$x \leftarrow 2$	$\frac{dx}{dx} \leftarrow 1$	1.000
$t \leftarrow e^x$	$\frac{dt}{dx} \leftarrow t\frac{dx}{dx}$	7.389
$u \leftarrow \log x$	$\frac{du}{dx} \leftarrow \frac{1}{x}\frac{dx}{dx}$	0.500
$v \leftarrow xu$	$\frac{dv}{dx} \leftarrow u\frac{dx}{dx} + x\frac{du}{dx}$	1.301
$w \leftarrow t + v$	$\frac{dw}{dx} \leftarrow \frac{dt}{dx} + \frac{dv}{dx}$	8.690
$y \leftarrow \sin w$	$\frac{dy}{dx} \leftarrow \cos(w)\frac{dw}{dx}$	-1.188

Sandia National Laboratories

Robustness of Newton's Method via Homotopy Continuation (LOCA)



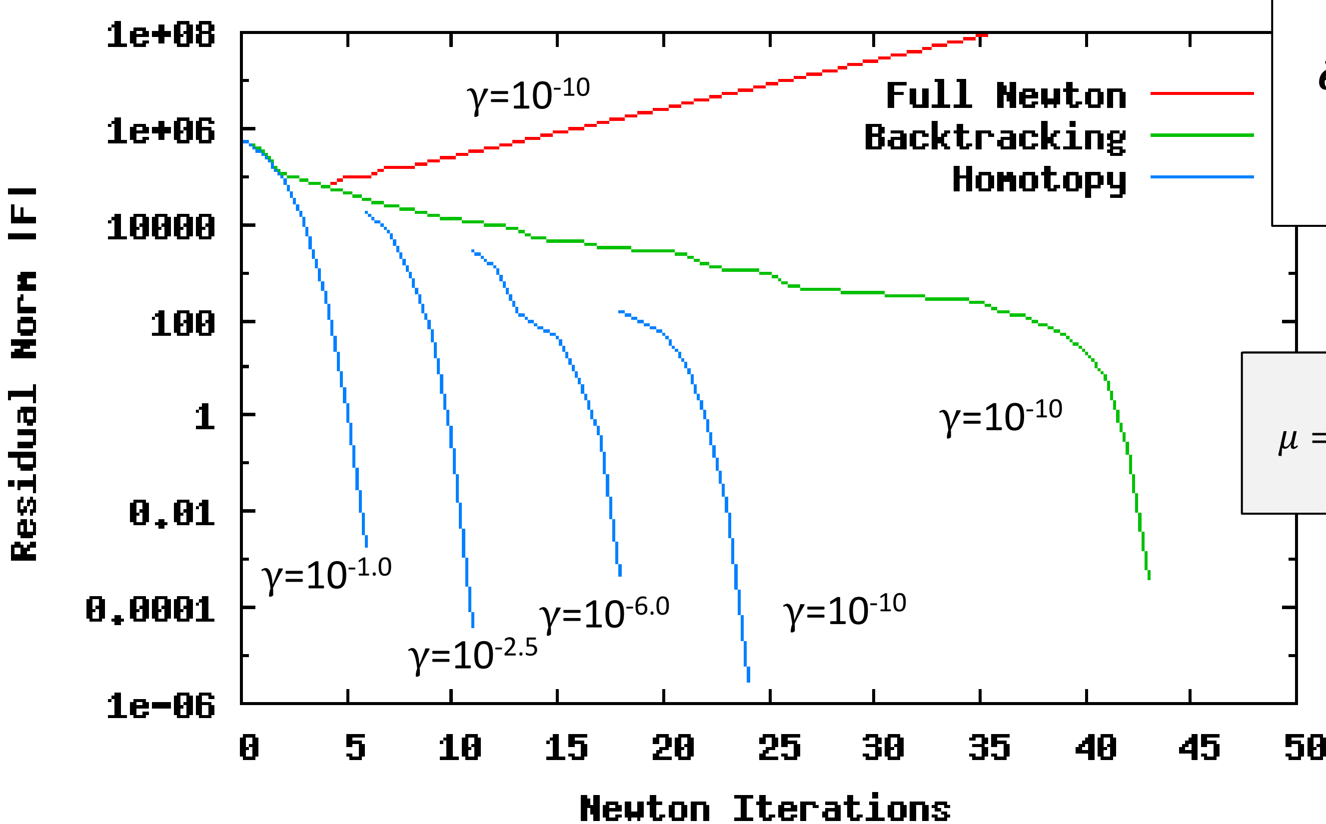
$$\dot{\boldsymbol{\epsilon}}_1{}^T = (2\dot{\epsilon}_{11} + \dot{\epsilon}_{22}, \dot{\epsilon}_{12}, \dot{\epsilon}_{13})$$
$$\dot{\boldsymbol{\epsilon}}_2{}^T = (2\dot{\epsilon}_{12}, \dot{\epsilon}_{11} + 2\dot{\epsilon}_{22}, \dot{\epsilon}_{23})$$
$$\dot{\epsilon}_{\mathrm{ij}} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

Glen's Law Viscosity:

$$\mu = \frac{1}{2} A^{-\frac{1}{n}} \left(\frac{1}{2}\sum_{ij} \dot{\boldsymbol{\epsilon}}_{ij}{}^2\right)^{\left(\frac{1}{2n}-\frac{1}{2}\right)}$$

$n = 3$
(Glen's law exponent)

Robustness of Newton's Method via Homotopy Continuation (LOCA)



$$\dot{\boldsymbol{\epsilon}}_1{}^T = (2\dot{\epsilon}_{11} + \dot{\epsilon}_{22}, \dot{\epsilon}_{12}, \dot{\epsilon}_{13})$$
$$\dot{\boldsymbol{\epsilon}}_2{}^T = (2\dot{\epsilon}_{12}, \dot{\epsilon}_{11} + 2\dot{\epsilon}_{22}, \dot{\epsilon}_{23})$$
$$\dot{\epsilon}_{\mathrm{ij}} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

Glen's Law Viscosity:

$$\mu = \frac{1}{2} A^{-\frac{1}{n}} \left(\frac{1}{2}\sum_{ij} \dot{\epsilon}_{ij}{}^2 + \gamma\right)^{\left(\frac{1}{2n}-\frac{1}{2}\right)}$$

γ = regularization parameter

$n = 3$
(Glen's law exponent)

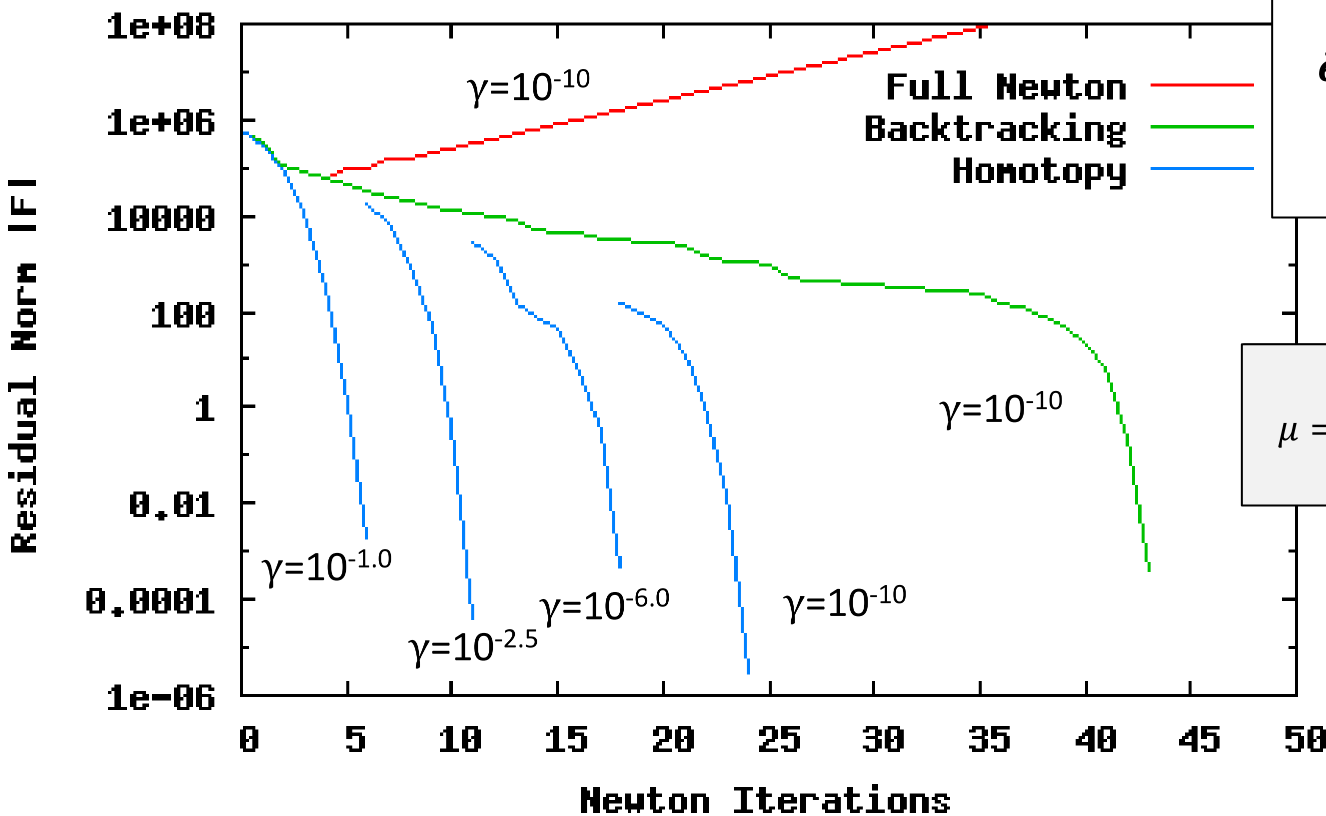
$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \Delta \boldsymbol{x}_k$$

Full step: $\alpha_k = 1$

Backtracking: line-search for α_k

Sandia National Laboratories

Robustness of Newton's Method via Homotopy Continuation (LOCA)



$$\dot{\boldsymbol{\epsilon}}_1{}^T = (2\dot{\epsilon}_{11} + \dot{\epsilon}_{22}, \dot{\epsilon}_{12}, \dot{\epsilon}_{13})$$
$$\dot{\boldsymbol{\epsilon}}_2{}^T = (2\dot{\epsilon}_{12}, \dot{\epsilon}_{11} + 2\dot{\epsilon}_{22}, \dot{\epsilon}_{23})$$
$$\dot{\epsilon}_{\text{ij}} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

Glen's Law Viscosity:

$$\mu = \frac{1}{2} A^{-\frac{1}{n}} \left(\frac{1}{2}\sum_{ij} \dot{\epsilon}_{ij}{}^2 + \gamma\right)^{\left(\frac{1}{2n} - \frac{1}{2}\right)}$$

γ = regularization parameter

$n = 3$
(Glen's law exponent)

Newton most robust with full step + homotopy continuation of $\gamma \to 10^{-10}$: converges out-of-the-box!

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \Delta \boldsymbol{x}_k$$

→ Full step: $\alpha_k = 1$

→ Backtracking: line-search for α_k

Sandia National Laboratories

Iterative Linear Solvers & Preconditioning

- In practice, large sparse linear systems $\boldsymbol{Ax} = \boldsymbol{b}$ are solved using **iterative methods**.
  - GMRES (Generalized Minimal RESidual).
  - CG (Conjugate Gradient) – for symmetric positive definite $\boldsymbol{A}$.

Sandia National Laboratories

Iterative Linear Solvers & Preconditioning

- In practice, large sparse linear systems $\boldsymbol{Ax} = \boldsymbol{b}$ are solved using **iterative methods**.
  - GMRES (Generalized Minimal RESidual).
  - CG (Conjugate Gradient) – for symmetric positive definite $\boldsymbol{A}$.
- Convergence of iterative methods for solving $\boldsymbol{Ax} = \boldsymbol{b}$ depends on **condition number** of $\boldsymbol{A}$: $\kappa(\boldsymbol{A}) = \frac{\sigma_{max}}{\sigma_{min}}$.
  - Large condition number $\Longrightarrow$ slow convergence.

Sandia National Laboratories

Iterative Linear Solvers & Preconditioning

- In practice, large sparse linear systems $\boldsymbol{Ax} = \boldsymbol{b}$ are solved using **iterative methods**.
  - GMRES (Generalized Minimal RESidual).
  - CG (Conjugate Gradient) – for symmetric positive definite $\boldsymbol{A}$.
- Convergence of iterative methods for solving $\boldsymbol{Ax} = \boldsymbol{b}$ depends on **condition number** of $\boldsymbol{A}$: $\kappa(\boldsymbol{A}) = \frac{\sigma_{max}}{\sigma_{min}}$.
  - Large condition number $\Longrightarrow$ slow convergence.
- Convergence of iterative method can be accelerated through **preconditioning**:
  - Find a matrix $\boldsymbol{P}$ such that $\boldsymbol{PA}$ has better condition number than $\boldsymbol{A}$ and solve $\boldsymbol{PAx} = \boldsymbol{Pb}$.
  - Perfect preconditioner: $\boldsymbol{P} = \boldsymbol{A}^{-1}$.

Iterative Linear Solvers & Preconditioning

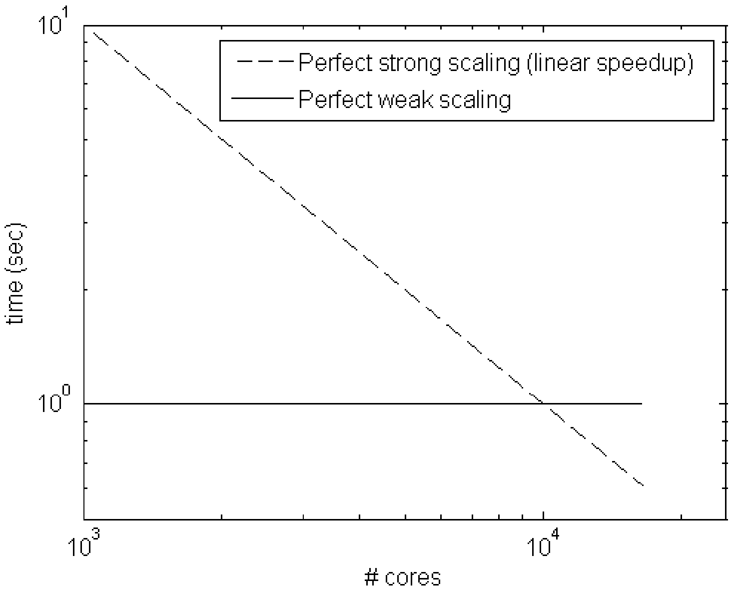
- In practice, large sparse linear systems $\boldsymbol{Ax} = \boldsymbol{b}$ are solved using **iterative methods**.
  - GMRES (Generalized Minimal RESidual).
  - CG (Conjugate Gradient) – for symmetric positive definite $\boldsymbol{A}$.
- Convergence of iterative methods for solving $\boldsymbol{Ax} = \boldsymbol{b}$ depends on **condition number** of $\boldsymbol{A}$: $\kappa(\boldsymbol{A}) = \frac{\sigma_{max}}{\sigma_{min}}$.
  - Large condition number $\Longrightarrow$ slow convergence.
- Convergence of iterative method can be accelerated through **preconditioning**:
  - Find a matrix $\boldsymbol{P}$ such that $\boldsymbol{PA}$ has better condition number than $\boldsymbol{A}$ and solve $\boldsymbol{PAx} = \boldsymbol{Pb}$.
  - Perfect preconditioner: $\boldsymbol{P} = \boldsymbol{A}^{-1}$.
- **Common preconditioners:**
  - ***Incomplete LU (ILU)*** factorization preconditioners: $\boldsymbol{A} \approx \tilde{\boldsymbol{L}}\tilde{\boldsymbol{U}}$; $\tilde{\boldsymbol{L}}, \tilde{\boldsymbol{U}}$ sparse $\boldsymbol{P}^{-1} = (\tilde{\boldsymbol{L}}\tilde{\boldsymbol{U}})^{-1}$.
  - ***Multigrid (MG)*** preconditioners: use solution to problem on coarse mesh to accelerate convergence on fine mesh → Geometric Multigrid (GMG), ***Algebraic Multigrid (AMG)***.

Definitions: Strong vs. Weak Scaling

Scalability (a.k.a. **Scaling Efficiency**) = measure of the efficiency of a code when increasing numbers of parallel processing elements (CPUs, cores, processes, threads, etc.).

- ***Strong scaling:*** how the solution time varies with the number of cores for a fixed total problem size.
  - $\Rightarrow$ Fix problem size, increase # cores.
  - **<u>Ideal:</u>** linear speed-up with increase in # cores ("hyperbolic strong scaling curve").

- ***Weak scaling:*** how the solution time varies with the number of cores for a fixed problem size per core.
  - $\Rightarrow$ Increase problem size and # cores s.t. # dofs/core is approximately constant.
  - **<u>Ideal:</u>** solution time remains constant as problem size and # cores increases.



<u>Note:</u> scalability usually declines above some threshold number of cores (Amdahl's Law, Gustafson's Law) → some operations are serial, communication is not free, etc.

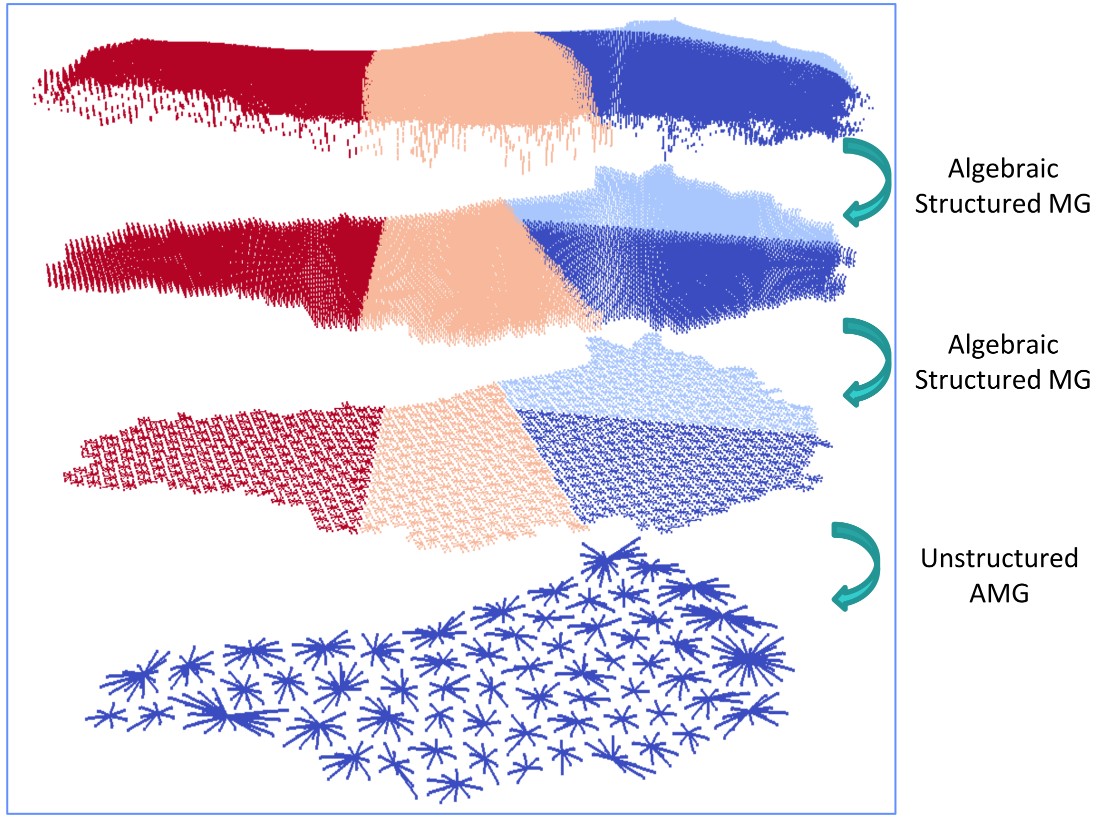
Laboratories

Scalability via Algebraic Multi-Grid Preconditioning with Semi-Coarsening

Bad aspect ratios ($dx \gg dz$) ruin classical AMG convergence rates!

- relatively small horizontal coupling terms, hard to smooth horizontal errors

$\Rightarrow$ Solvers (AMG and ILU) must take aspect ratios into account

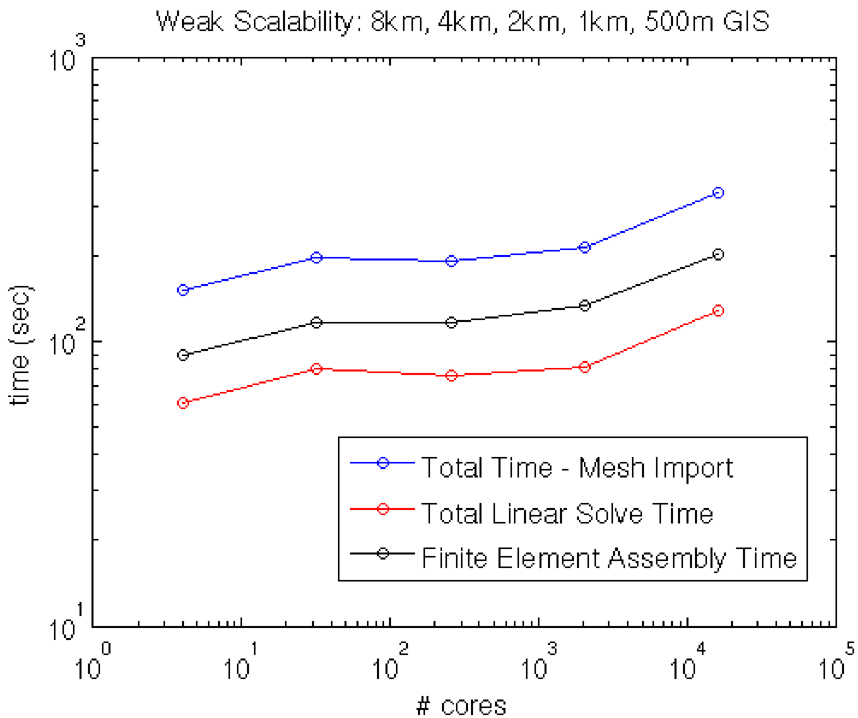


We developed a **new AMG solver** based on aggressive **semi-coarsening** (available in *ML/MueLu* packages of *Trilinos*)

See (Tuminaro, 2014), (Tezaur *et al.*, 2015), (Tuminaro *et al.*, 2015).

<u>Scaling studies (next slides):</u>
New AMG preconditioner vs. ILU

Greenland Controlled Weak Scalability Study



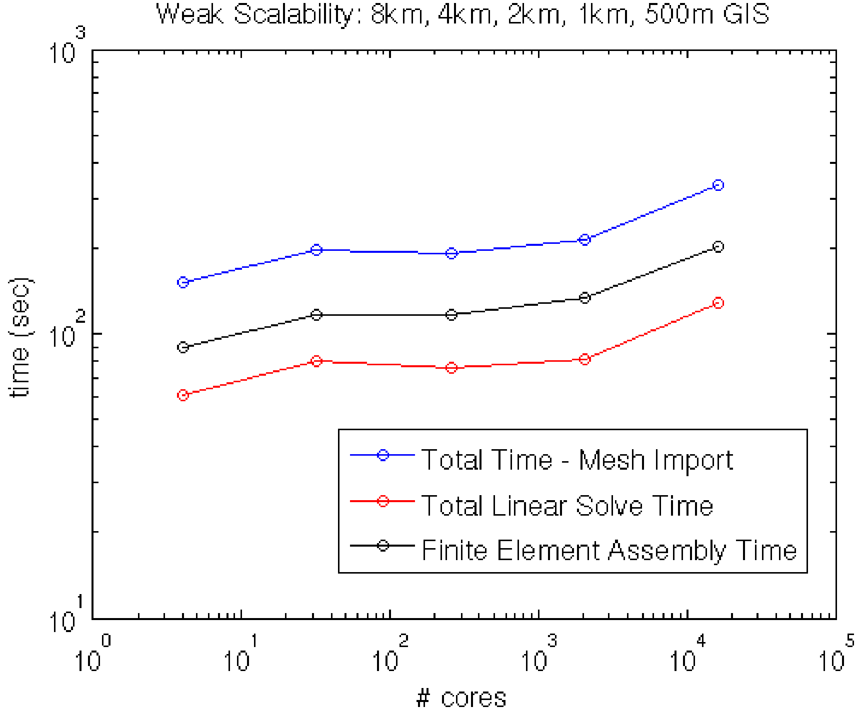
4 cores 334K dofs 8 km Greenland, 5 vertical layers	$\times 8^4$ scale up	16,384 cores **1.12B dofs(!)** 0.5 km Greenland, 80 vertical layers

- Weak scaling study with fixed dataset, 4 mesh bisections.
- ~70-80K dofs/core.
- ***Conjugate Gradient (CG) iterative method*** for linear solves (faster convergence than GMRES).
- ***New AMG preconditioner*** developed by R. Tuminaro based on ***semi-coarsening*** (coarsening in z-direction only).
- ***Significant improvement*** in scalability with new AMG preconditioner over ILU preconditioner!

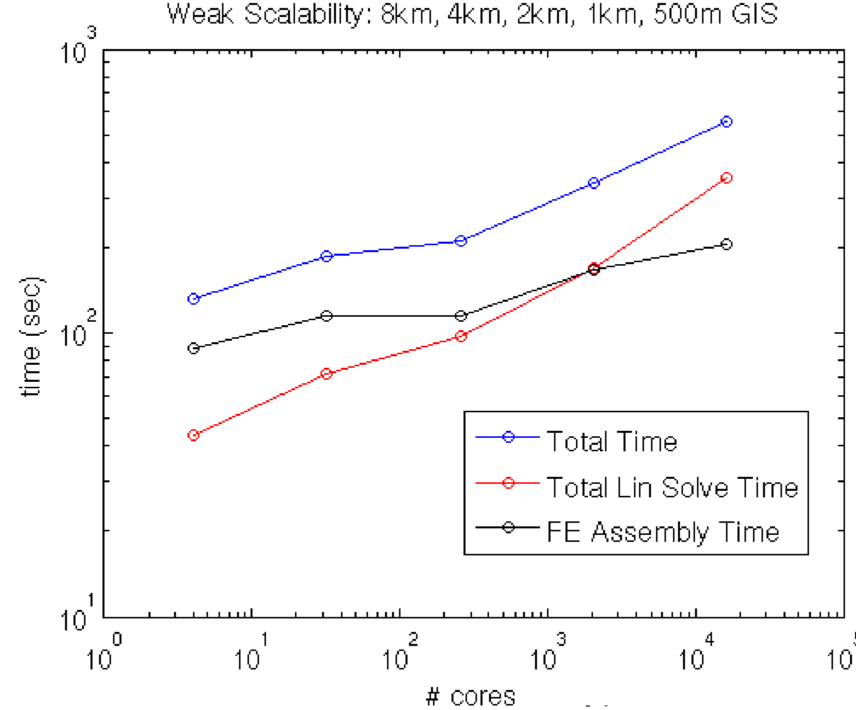
Laboratories

Greenland Controlled Weak Scalability Study

New AMG preconditioner



ILU preconditioner



4 cores
334K dofs
8 km Greenland,
5 vertical layers

$\times 8^4$
scale up

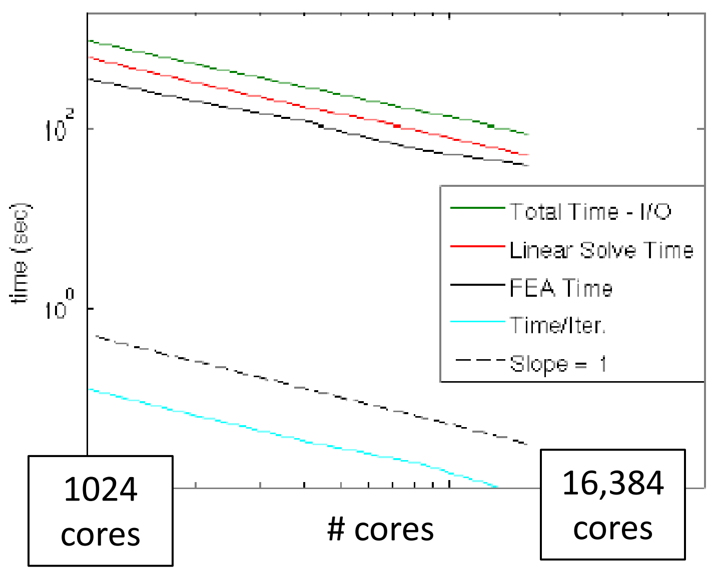
16,384 cores
1.12B dofs(!)
0.5 km Greenland,
80 vertical layers

- ***Significant improvement*** in scalability with new AMG preconditioner over ILU preconditioner!

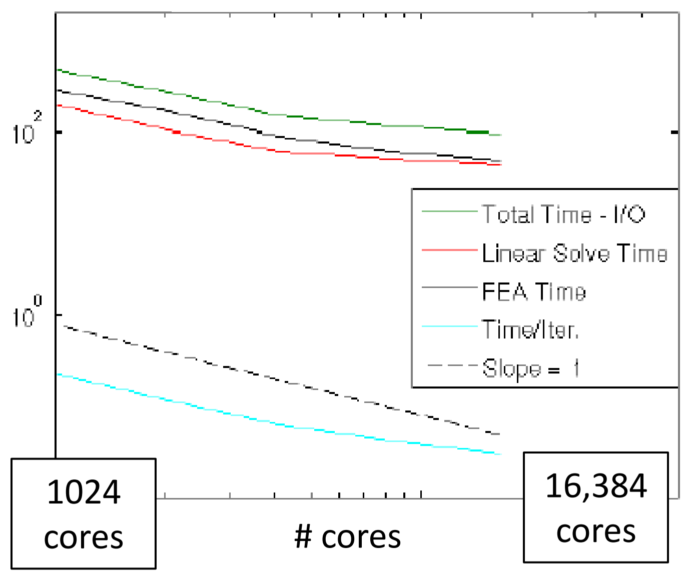
Fine-Resolution Greenland Strong Scaling Study

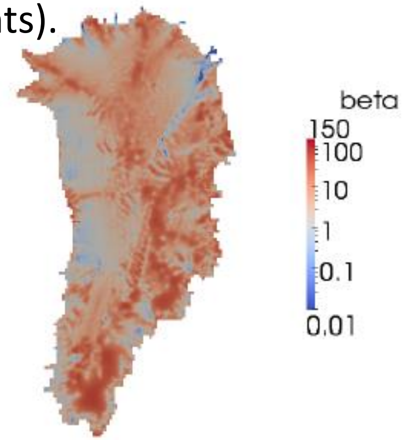
- Strong scaling on 1km Greenland with 40 vertical layers (143M dofs, hex elements).
- Initialized with realistic basal friction (from deterministic inversion) and temperature fields → interpolated from coarser to fine mesh.
- **Iterative linear solver**: CG.
- **Preconditioner**: ILU vs. new AMG (based on aggressive semi-coarsening).

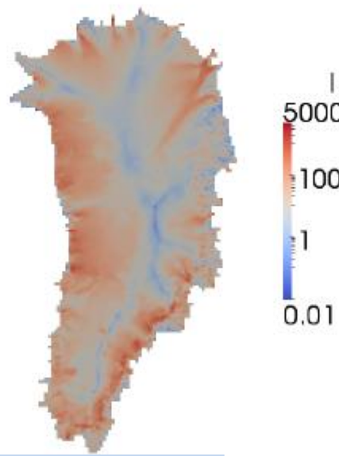
ILU



AMG





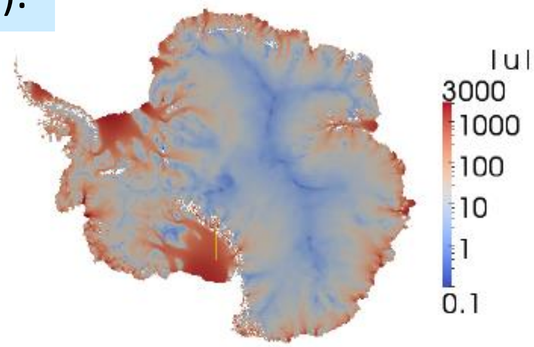


ILU solver scales better than AMG but ILU solve is slightly slower: AMG solver becomes inefficient when # unknowns/core small (expensive setup; a lot of communications).

Sandia National Laboratories

Moderate Resolution Antarctica Weak Scaling Study

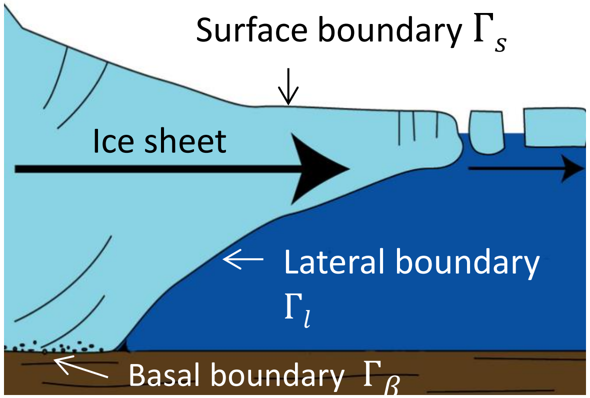
Antarctica is fundamentally different than Greenland:
AIS contains large ice shelves (floating extensions of land ice).



- ***Along ice shelf front:*** open-ocean BC (Neumann).
- ***Along ice shelf base:*** zero traction BC (Neumann).

⇒ For vertical grid lines that lie within ice shelves, top and bottom BCs resemble Neumann BCs so sub-matrix associated with one of these lines is almost* singular.

(vertical > horizontal coupling)
+
Neumann BCs
=
nearly singular submatrix associated with vertical lines



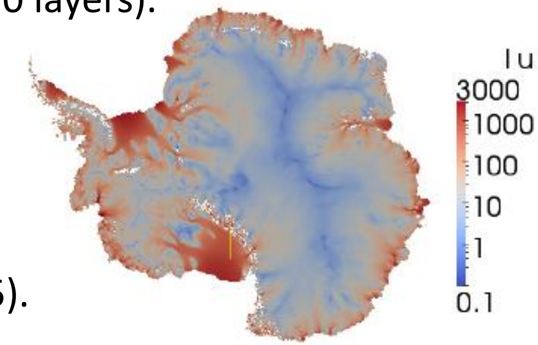
⇒ Ice shelves give rise to severe ill-conditioning of linear systems!

*Completely singular in the presence of islands and some ice tongues.

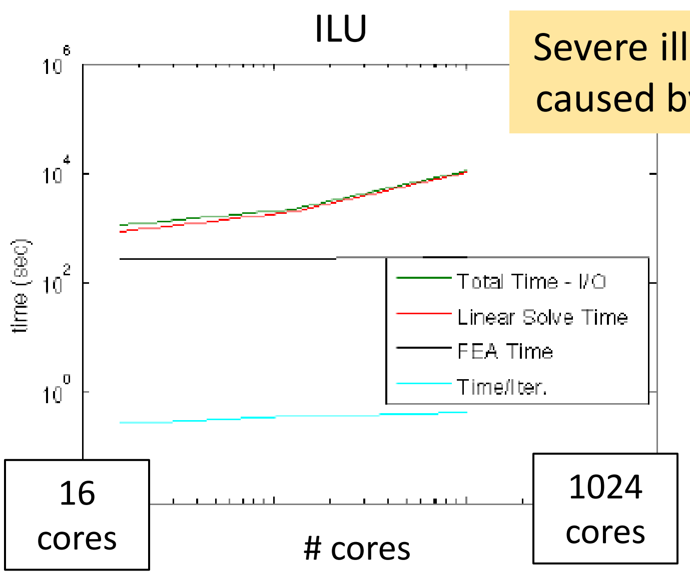
Sandia National Laboratories

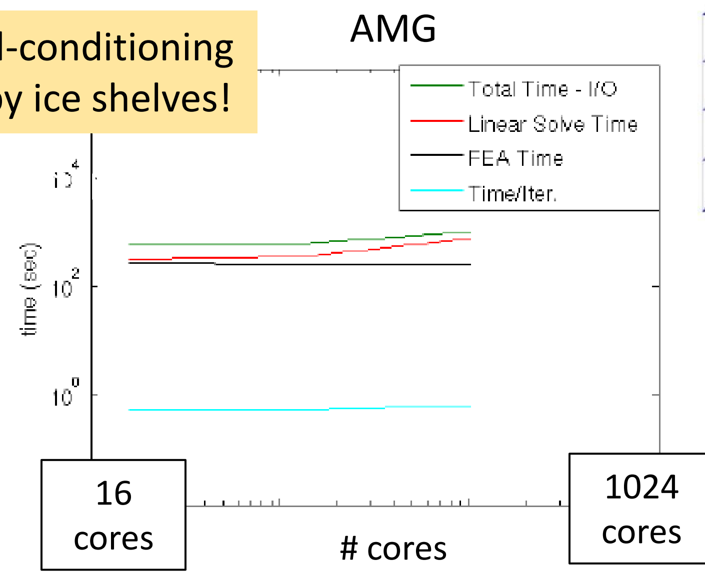
Moderate Resolution Antarctica Weak Scaling Study

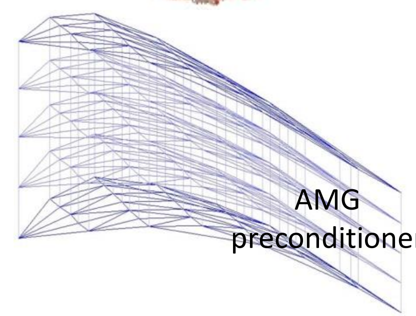
- Weak scaling study on Antarctic problem (8km w/ 5 layers → 2km with 20 layers).
- Initialized with realistic basal friction (from deterministic inversion) and temperature field from BEDMAP2.
- **Iterative linear solver**: GMRES.
- **Preconditioner**: ILU vs. new AMG based on aggressive semi-coarsening (Tezaur *et al GMD* 2014, Tezaur *et al ICCS* 2015, Tuminaro *et al SISC* 2015).



Severe ill-conditioning caused by ice shelves!







(vertical > horizontal coupling)
+
Neumann BCs
=
nearly singular submatrix associated with vertical lines

AMG preconditioner less sensitive than ILU to ill-conditioning (ice shelves → Green's function with modest horizontal decay → ILU is less effective).

Outline

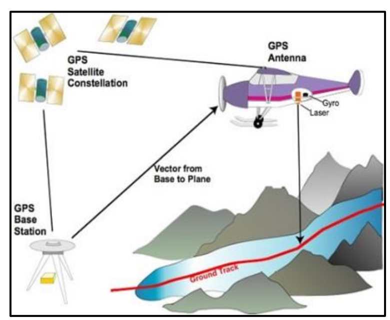
- Motivation.
- The PISCEES project.
- The Ice Sheet Equations.
- The Albany/FELIX Steady Stress-Velocity Solver.
- **Deterministic Inversion for Ice Sheet Initialization.**
- Dynamic Simulations of Ice Sheet Evolution.
- Uncertainty Quantification.
- Summary & future work.

Deterministic Inversion: Estimation of Ice Sheet Initial State

Objective: find ice sheet initial state that
- Matches observations (e.g., surface velocity, temperature, etc.)
- Matches present-day geometry (elevation, thickness).
- Is in "equilibrium" with climate forcings (SMB).

Approach: invert for unknown/uncertain ice sheet model parameters.
- Significantly reduces non-physical transients without model spin-up.

Available data/measurements:
- Ice extent and surface topography.
- Surface velocity.
- Surface mass balance (SMB).
- Ice thickness H (sparse measurements).

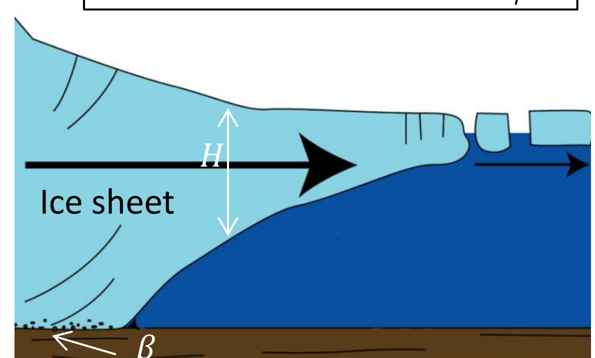


Field to be estimated:
- Basal friction β (spatially variable proxy for all basal processes).
- Ice thickness H (allowed to be weighted by observational uncertainties).

Assumptions:
- Ice flow described by FO Stokes equations.
- Ice is close to mechanical equilibrium.
- Temperature field is given.

Basal sliding BC:

$$2\mu\dot{\boldsymbol{\epsilon}}_i \cdot \boldsymbol{n} + \beta u i = 0, \text{ on } \Gamma_\beta$$



Deterministic Inversion Problem

First Order Stokes PDE Constrained Optimization Problem:

minimize $_{\beta,H}$ $J(\beta, H)$
s.t. FO Stokes PDEs

where

$$J(\beta, H) = \frac{1}{2}\alpha_v \int_{\Gamma_{top}} |\boldsymbol{u} - \boldsymbol{u}^{obs}|^2 ds + \frac{1}{2}\alpha \int_{\Gamma} |div(\boldsymbol{U}H) - SMB|^2 ds + \frac{1}{2}\alpha_H \int_{\Gamma_{top}} |H - H^{obs}|^2 ds + \mathcal{R}(\beta) + \mathcal{R}(H)$$

- Minimize difference between:
    - Computed and measured ***surface velocity ($\boldsymbol{u}^{obs}$)*** → *common*
    - Computed divergence flux and measured ***surface mass balance (SMB)*** → *novel*
    - Computed and ***reference thickness (H^{obs})*** → *novel*

- Control variables:
    - ***Basal friction (β)***.
    - ***Thickness (H)***.

Deterministic Inversion Algorithm and Software

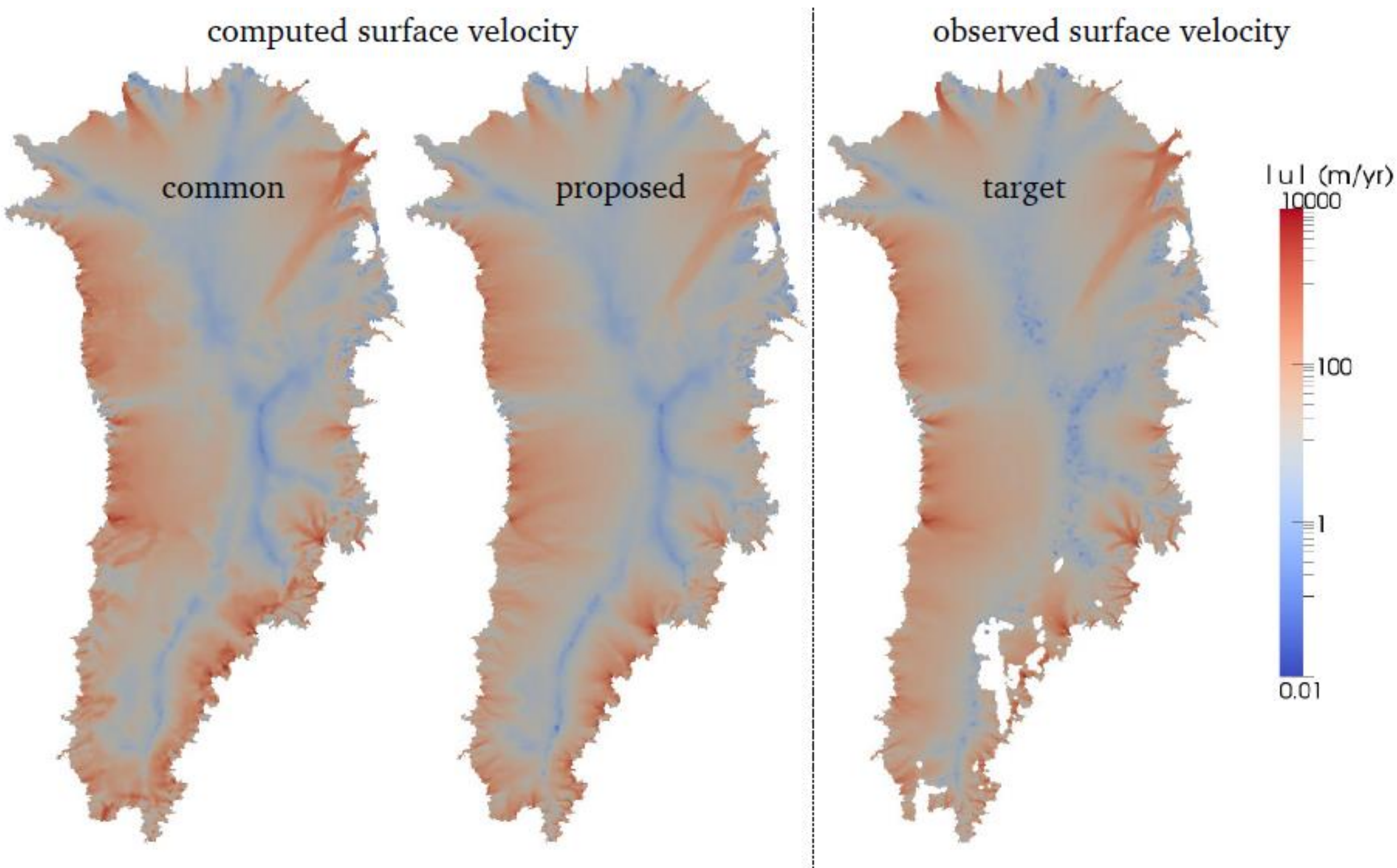
Algorithm	Software
Finite Element Method discretization	Albany
Quasi-Newton optimization (L-BFGS)	ROL
Nonlinear solver (Newton)	NOX
Krylov linear solvers	AztecOO+Ifpack/ML

Trilinos

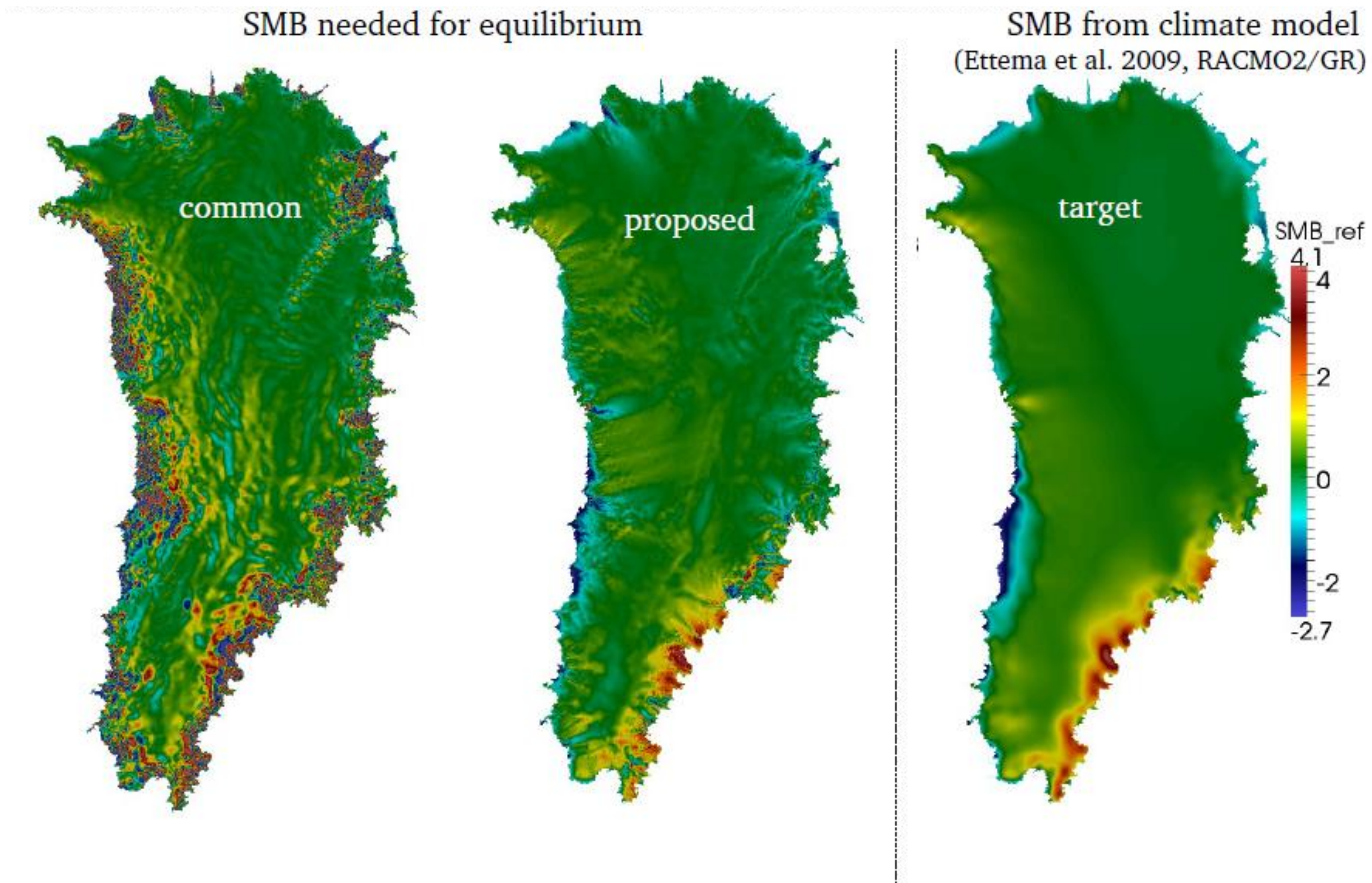
- ***Some details:***
    - Regularization: Tikhonov.
    - Total derivatives of objective functional J computed using adjoints and automatic differentiation.
    - L-BFGS initialized with Hessian of regularization terms.

Sandia National Laboratories

Deterministic Inversion: Greenland



Deterministic Inversion: Greenland (cont’d)

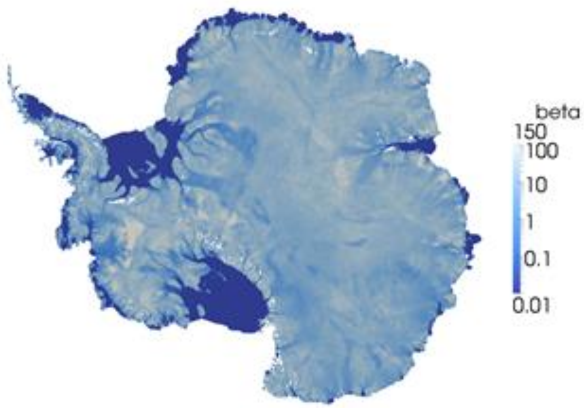


Deterministic Inversion: Antarctica (basal friction only)

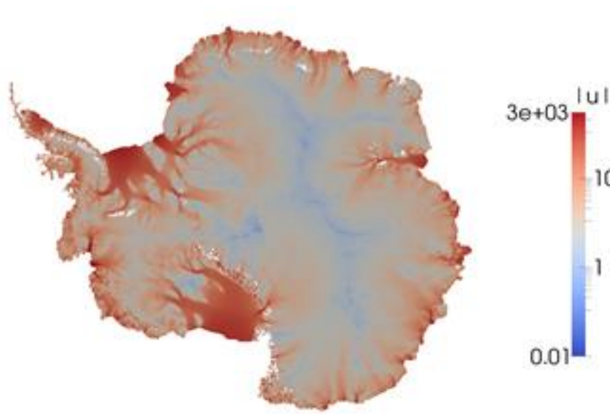
FO Stokes PDE Constrained Optimization Problem:

$$J(\beta) = \frac{1}{2}\int_{\Gamma_{top}} \alpha |\boldsymbol{u} - \boldsymbol{u}^{obs}|^2 ds + \mathcal{R}(\beta)$$

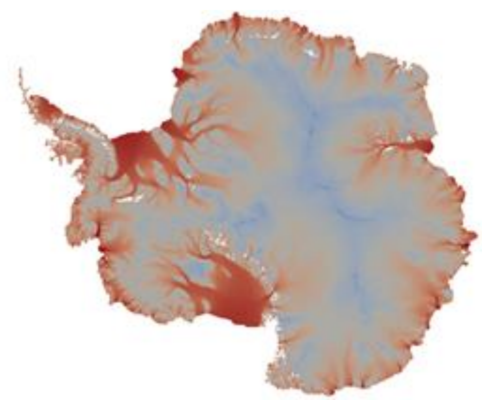
Geometry: Cornford, Martin *et al.* (in prep.)
Bedmap2: Fretwell et al., 2013
Temperature: Pattyn, 2010.



β (kPa y/m) obtained through inversion



$|\boldsymbol{u}|$ (m/yr) computed with estimated β



$|\boldsymbol{u}|$ (m/yr) for observed surface velocity

Antarctic ice sheet inversion performed on up to **1.6M** parameters

Sandia National Laboratories

Outline

- Motivation.
- The PISCEES project.
- The Ice Sheet Equations.
- The Albany/FELIX Steady Stress-Velocity Solver.
- Deterministic Inversion for Ice Sheet Initialization.
- **Dynamic Simulations of Ice Sheet Evolution.**
- Uncertainty Quantification.
- Summary & future work.

Interfaces to *CISM/MPAS LI* for Transient Simulations

CISM-Albany

cism_driver

CISM (Fortran)
Thickness evolution, temperature solve, coupling to CESM

C++/Fortran Interface, Mesh Conversion

output file

Albany/FELIX (C++)
velocity solve

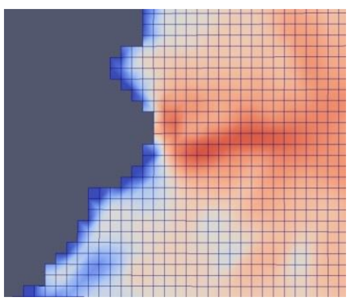
MPAS LI-Albany

LandIce_model

MPAS Land-Ice (Fortran)
Thickness evolution, temperature solve, coupling to DOE-ESM

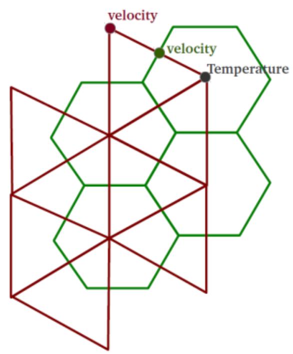
C++/Fortran Interface, Mesh Conversion

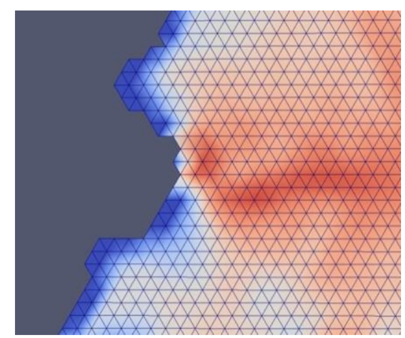
output file



- Structured hexahedral meshes (rectangles extruded to hexes).

- Tetrahedral meshes (dual of hexaganonal mesh, extruded to tets).





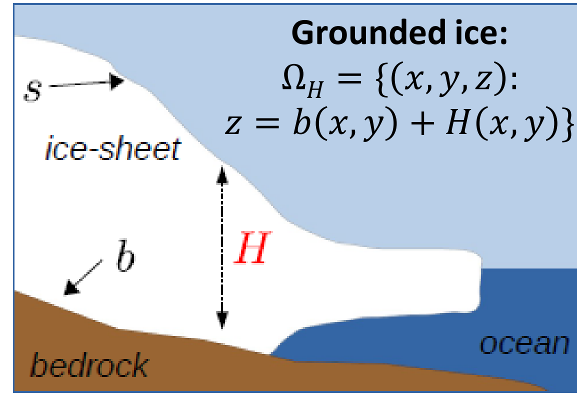
Albany/FELIX has been coupled to two land ice dycores: **C**ommunity **I**ce **S**heet **M**odel (***CISM***) and **M**odel for **P**rediction **A**cross **S**cales for **L**and Ice (***MPAS LI***)

Sandia National Laboratories

First Order Stokes-Thickness Coupling Methods

$$-2\mu\nabla\cdot\dot{\boldsymbol{\epsilon}} = -\rho g\nabla(b+H)\,, \quad \text{in } \Omega_H$$

$$\frac{\partial H}{\partial t} = -\nabla\cdot(\bar{\boldsymbol{u}}H) + \dot{b}$$



- ***<u>Sequential coupling</u>*** (common approach):
  - Given H^n, solve FO Stokes system for $\boldsymbol{u}^n$.
  - Compute $\bar{\boldsymbol{u}}$ and solve thickness evolution equation for H^{n+1}.
  - Thickness equation solved with upwind scheme + incremental remap.
  - **Upside:** fits nicely into existing codes
  - **Downside:** CFL requires tiny time steps for fine meshes.

- ***<u>Semi-Implicit coupling</u>*** (new approach):

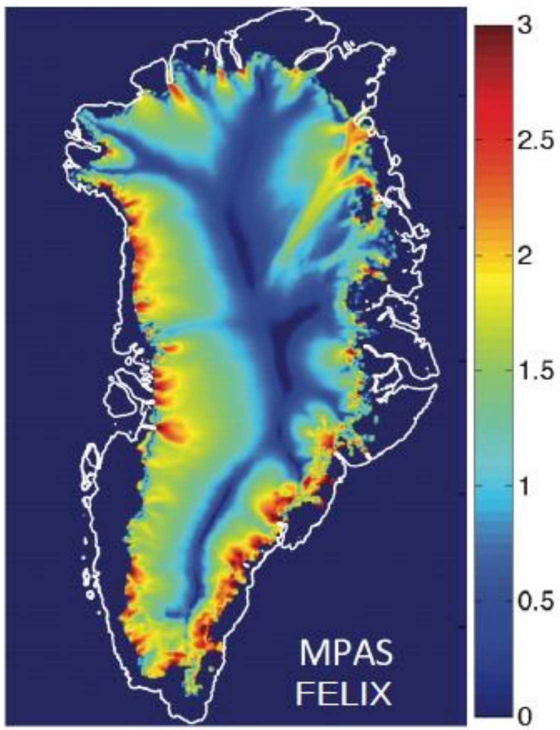
$$-2\mu\nabla\cdot\dot{\boldsymbol{\epsilon}}(\boldsymbol{u}) = -\rho g\nabla(b+H)\,, \quad \text{in } \Omega_H$$

$$\frac{H - H^n}{\Delta t} = -\nabla\cdot(\bar{\boldsymbol{u}}H^n) + \dot{b}$$

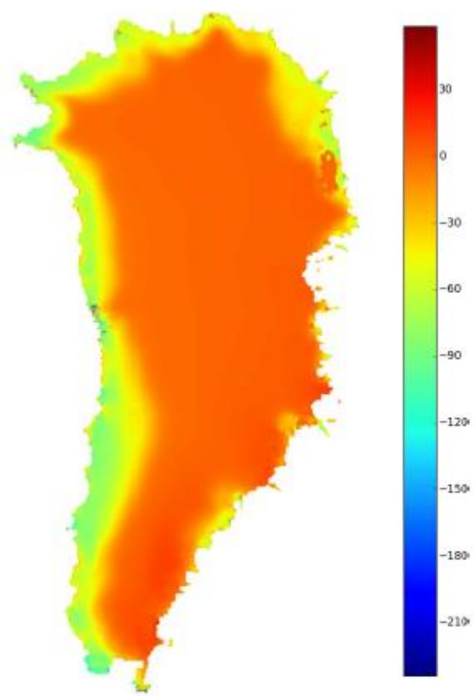
  - $\boldsymbol{u}$ computed in *Albany/FELIX* with implicit solve; *MPAS* uses velocity to march in time explicitly.
  - **Upside:** semi-implicit discretization mitigates stability issue (can use larger Δt)
  - **Downside:** more intrusive implementation; larger system; expense associated to geometry changing between iterations (use Newton to compute shape derivatives).

Sandia National Laboratories

Results Using Sequential Approach: 5km Greenland



Surface velocity magnitude [km/yr]



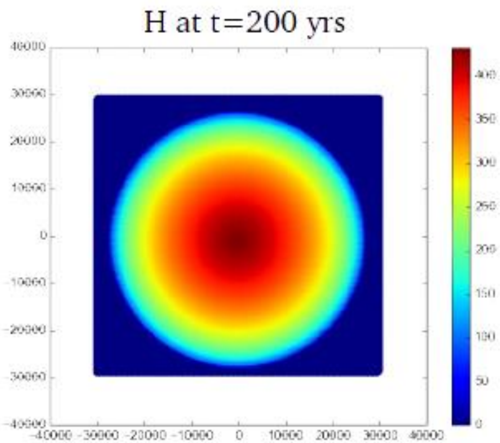
Thickness [m] change after 100 yr

FO equations solved using Albany/FELIX (finite elements) [left]

Evolution equations solves using MPAS (finite volume) [right]

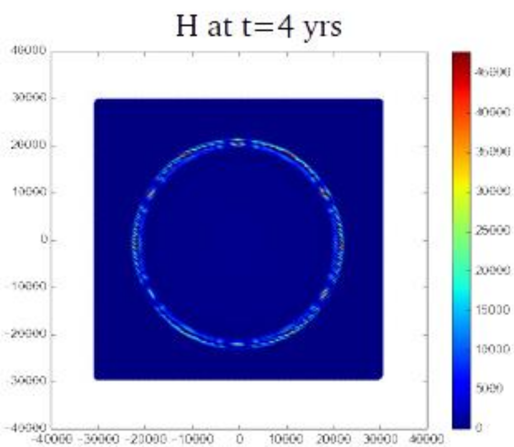
- Sequential approach works fine for relatively coarse meshes (e.g., 5km resolution).
- Data from ice2sea experiment A.J. Payne *et al.*, PNAS 2013.

Preliminary Results Using Semi-Implicit Approach: Dome

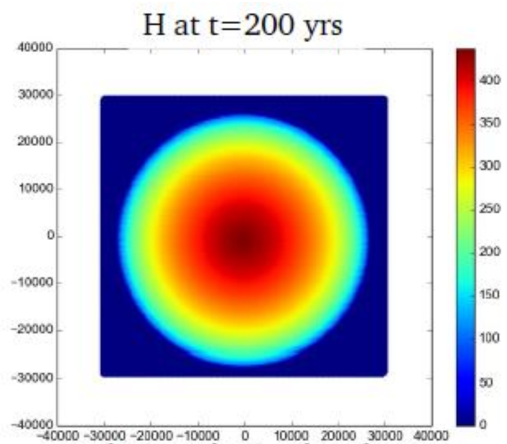


Top left: reference solution computed using sequential approach and time step of 5 months

Top left: reference solution computed using sequential approach and time step of 5 months



Solution obtained with sequential coupling, **dt** =1 yr

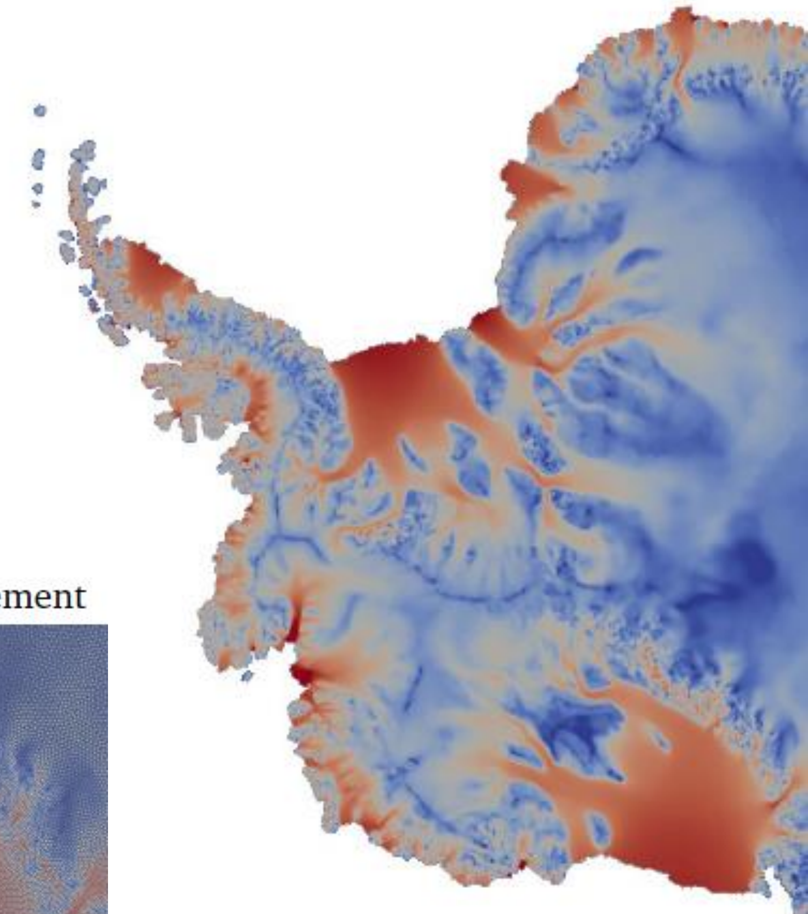


Solution obtained with semi-implicit coupling, **dt** =5 yrs

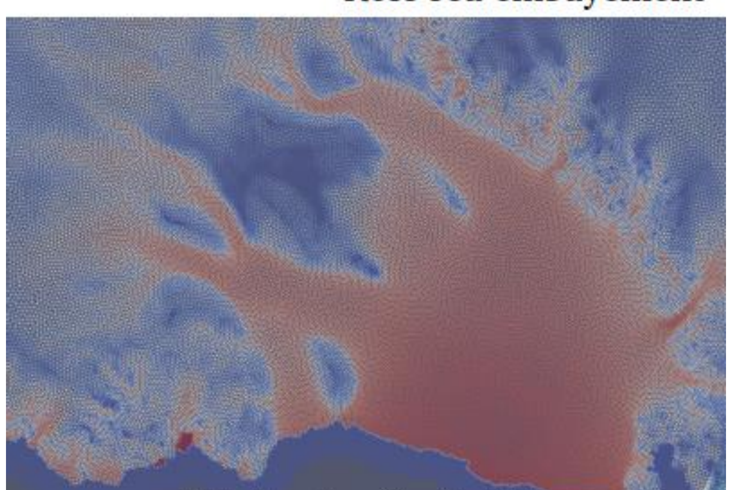
Semi-implicit approach allows the use of much larger time-steps than sequential approach!

Preliminary Results Using Semi-Implicit Approach: Antarctica

- Variable-resolution Antarctica grid with maximum resolution of 3km.
- ***Sequential approach:*** $\Delta t = O(\text{days})$
- ***Semi-Implicit approach:*** $\Delta t = O(\text{months})$
- Cost of iteration is larger for semi-implicit scheme because of increased dimension of nonlinear system (more expensive assembly and solve).
- Nonetheless, with semi-implicit scheme, we obtained **speedup of 4.5**$\times$ (~2 year run).



Ross sea embayement



Outline

- Motivation.
- The PISCEES project.
- The Ice Sheet Equations.
- The Albany/FELIX Steady Stress-Velocity Solver.
- Deterministic Inversion for Ice Sheet Initialization.
- Dynamic Simulations of Ice Sheet Evolution.
- **Uncertainty Quantification.**
- Summary & future work.

Uncertainty Quantification (UQ) Problem Definition

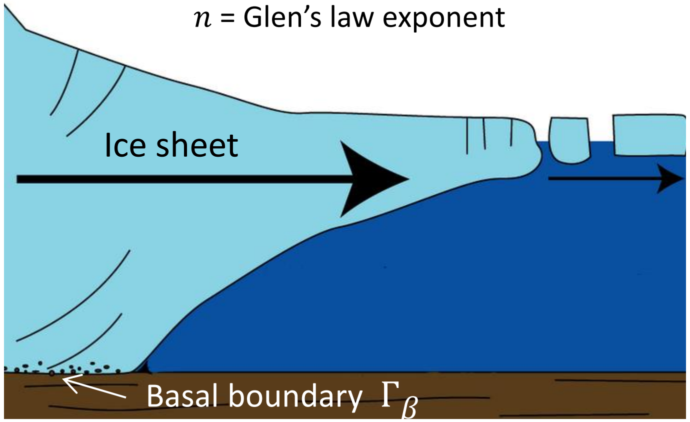
Quantity of Interest (QoI) in Ice Sheet Modeling:
total ice mass loss/gain during 21st century
→ ***sea level rise prediction.***

$$\mu = \frac{1}{2} A^{-\frac{1}{n}} \left(\frac{1}{2} \sum_{ij} \dot{\epsilon}_{ij}{}^2 + \gamma \right)^{\left(\frac{1}{2n} - \frac{1}{2}\right)}$$

n = Glen's law exponent

There are several sources of uncertainty, most notably:

- Climate forcings (e.g., surface mass balance).
- Basal friction (β)
- Bedrock topography
- Geothermal heat flux
- Model parameters (e.g., Glen's flow law exponent)



Basal sliding BC:
$2\mu\dot{\boldsymbol{\epsilon}}_i \cdot \boldsymbol{n} + \beta u_i = 0$, on Γ_β

Uncertainty Quantification (UQ) Problem Definition

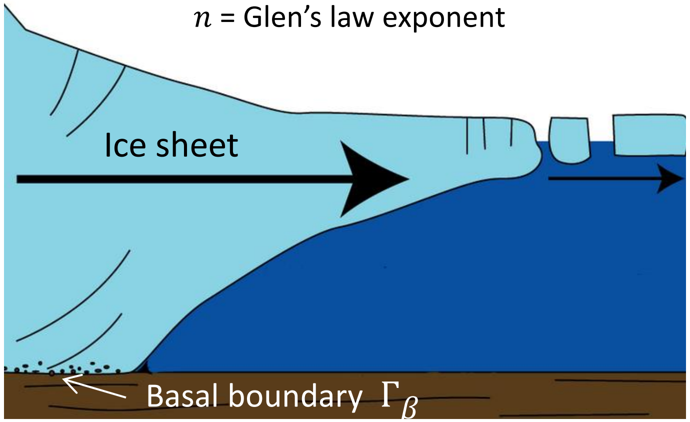
Quantity of Interest (QoI) in Ice Sheet Modeling:
total ice mass loss/gain during 21st century
→ ***sea level rise prediction.***

$$\mu = \frac{1}{2} A^{-\frac{1}{n}} \left(\frac{1}{2} \sum_{ij} \dot{\epsilon}_{ij}^{\,2} + \gamma \right)^{\left(\frac{1}{2n} - \frac{1}{2}\right)}$$

n = Glen's law exponent

There are several sources of uncertainty, most notably:

- Climate forcings (e.g., surface mass balance).
- **Basal friction ($\boldsymbol{\beta}$).**
- Bedrock topography.
- Geothermal heat flux.
- Model parameters (e.g., Glen's flow law exponent).



As a first step, we focus on effect of uncertainty in **basal friction ($\boldsymbol{\beta}$)** only.

Basal sliding BC:
$2\mu\dot{\boldsymbol{\epsilon}}_i \cdot \boldsymbol{n} + \beta u_i = 0$, on Γ_β

Uncertainty Quantification Workflow

Goal: Uncertainty Quantification in 21st century sea level (QoI)

- **Deterministic inversion:** perform adjoint-based deterministic inversion to estimate initial ice sheet state (i.e., characterize the present state of the ice sheet to be used for performing prediction runs).

What are the parameters that render a given set of observations?

- **Bayesian calibration:** construct the posterior distribution using Markov Chain Monte Carlo (MCMC) run on an emulator of the forward model → *Bayes' Theorem:* assume prior distribution; update using data:

$$\underbrace{\pi(\theta|d)}_{\text{posterior}} = \frac{\overbrace{\pi(d|\theta)}^{\text{likelihood}}\ \overbrace{\pi(\theta)}^{\text{prior}}}{\pi(d)} = \frac{\pi(d|\theta)\ \pi(\theta)}{\int \pi(d|\theta)\ \pi(\theta)\ d\theta}$$

- **Forward propagation:** sample the obtained distribution and perform ensemble of forward propagation runs to compute the uncertainty in the QoI.

What is the impact of uncertain parameters in the model on quantities of interest (QoI)?

Uncertainty Quantification Workflow

Goal: Uncertainty Quantification in 21st century sea level (QoI)

- **Deterministic inversion:** perform adjoint-based deterministic inversion to estimate initial ice sheet state (i.e., characterize the present state of the ice sheet to be used for performing prediction runs).

What are the parameters that render a given set of observations?

- **Bayesian calibration:** construct the posterior distribution using Markov Chain Monte Carlo (MCMC) run on an emulator of the forward model → *Bayes' Theorem:* assume prior distribution; update using data:

$$\underbrace{\pi(\theta|d)}_{\text{posterior}} = \frac{\overbrace{\pi(d|\theta)}^{\text{likelihood}} \; \overbrace{\pi(\theta)}^{\text{prior}}}{\pi(d)} = \frac{\pi(d|\theta)\,\pi(\theta)}{\int \pi(d|\theta)\,\pi(\theta)\,d\theta}$$

- **Forward propagation:** sample the obtained distribution and perform ensemble of forward propagation runs to compute the uncertainty in the QoI.

What is the impact of uncertain parameters in the model on quantities of interest (QoI)?

Bayesian Calibration: Demonstration of Workflow using KLE

Albany/FELIX has been hooked up to ***DAKOTA*** (in "black-box" mode) for ***UQ/ Bayesian calibration.***

Difficulty in UQ: "Curse of Dimensionality"
The β-field inversion problem has $O(100K)$ dimensions!

Bayesian Calibration: Demonstration of Workflow using KLE

Albany/FELIX has been hooked up to ***DAKOTA*** (in "black-box" mode) for ***UQ/ Bayesian calibration.***

Difficulty in UQ: "Curse of Dimensionality"
The β-field inversion problem has $O(100K)$ dimensions!

<u>**Approach:**</u> Reduce $O(100K)$ dimensional problem to $O(10)$ dimensional problem.

Bayesian Calibration: Demonstration of Workflow using KLE

Albany/FELIX has been hooked up to ***DAKOTA*** (in “black-box” mode) for ***UQ/ Bayesian calibration.***

Difficulty in UQ: “Curse of Dimensionality”
The β-field inversion problem has $O(100K)$ dimensions!

Approach: Reduce $O(100K)$ dimensional problem to $O(10)$ dimensional problem.

- For initial demonstration of workflow, we use the ***Karhunen-Loeve Expansion (KLE)***:

Sandia National Laboratories

Bayesian Calibration: Demonstration of Workflow using KLE

Albany/FELIX has been hooked up to ***DAKOTA*** (in "black-box" mode) for ***UQ/ Bayesian calibration.***

Difficulty in UQ: "Curse of Dimensionality"
The β-field inversion problem has $O(100K)$ dimensions!

Approach: Reduce $O(100K)$ dimensional problem to $O(10)$ dimensional problem.

- For initial demonstration of workflow, we use the ***Karhunen-Loeve Expansion (KLE)***:

  1. Assume analytic covariance kernel $C(r_1, r_2) = exp\left(-\frac{(r_1 - r_2)^2}{L^2}\right)$.

Sandia National Laboratories

Bayesian Calibration: Demonstration of Workflow using KLE

Albany/FELIX has been hooked up to ***DAKOTA*** (in "black-box" mode) for ***UQ/ Bayesian calibration.***

Difficulty in UQ: "Curse of Dimensionality"
The β-field inversion problem has $O(100K)$ dimensions!

Approach: Reduce $O(100K)$ dimensional problem to $O(10)$ dimensional problem.

- For initial demonstration of workflow, we use the ***Karhunen-Loeve Expansion (KLE)***:
  1. Assume analytic covariance kernel $C(r_1, r_2) = exp\left(-\frac{(r_1 - r_2)^2}{L^2}\right)$.
  2. Perform eigenvalue decomposition of C.

Bayesian Calibration: Demonstration of Workflow using KLE

Albany/FELIX has been hooked up to ***DAKOTA*** (in "black-box" mode) for ***UQ/ Bayesian calibration.***

Difficulty in UQ: "Curse of Dimensionality"
The β-field inversion problem has $O(100K)$ dimensions!

<u>Approach:</u> Reduce $O(100K)$ dimensional problem to $O(10)$ dimensional problem.

- For initial demonstration of workflow, we use the ***Karhunen-Loeve Expansion (KLE)***:
  1. Assume analytic covariance kernel $C(r_1, r_2) = exp\left(-\frac{(r_1-r_2)^2}{L^2}\right)$.
  2. Perform eigenvalue decomposition of C.
  3. Expand* $\beta - \bar{\beta}$ in basis of eigenvectors $\{\boldsymbol{\phi}_k\}$ of C, with random variables $\{\xi_k\}$:

$$\boxed{\beta(\omega) = \bar{\beta} + \sum_{k=1}^{K} \sqrt{\lambda_k} \boldsymbol{\phi}_k \xi_k(\omega)}$$

$\bar{\beta}$ = initial condition for β (result of deterministic inversion or spin-up)

*In practice, expansion is done on $\log(\beta)$ to avoid negative values of β.

Bayesian Calibration: Demonstration of Workflow using KLE

Albany/FELIX has been hooked up to ***DAKOTA*** (in "black-box" mode) for ***UQ/ Bayesian calibration.***

Difficulty in UQ: "Curse of Dimensionality"
The β-field inversion problem has $O(100K)$ dimensions!

Approach: Reduce $O(100K)$ dimensional problem to $O(10)$ dimensional problem.

- For initial demonstration of workflow, we use the ***Karhunen-Loeve Expansion (KLE)***:

Offline

1. Assume analytic covariance kernel $C(r_1, r_2) = exp\left(-\frac{(r_1 - r_2)^2}{L^2}\right)$.
2. Perform eigenvalue decomposition of C.
3. Expand* $\beta - \bar{\beta}$ in basis of eigenvectors $\{\boldsymbol{\phi}_k\}$ of C, with random variables $\{\xi_k\}$:

$$\boxed{\beta(\omega) = \bar{\beta} + \sum_{k=1}^{K} \sqrt{\lambda_k} \boldsymbol{\phi}_k \xi_k(\omega)}$$

$\bar{\beta}$ = initial condition for β (result of deterministic inversion or spin-up)

*In practice, expansion is done on $\log(\beta)$ to avoid negative values of β.

Sandia National Laboratories

Bayesian Calibration: Demonstration of Workflow using KLE

Albany/FELIX has been hooked up to ***DAKOTA*** (in "black-box" mode) for ***UQ/ Bayesian calibration.***

Difficulty in UQ: "Curse of Dimensionality"
The β-field inversion problem has $O(100K)$ dimensions!

Approach: Reduce $O(100K)$ dimensional problem to $O(10)$ dimensional problem.

- For initial demonstration of workflow, we use the ***Karhunen-Loeve Expansion (KLE)***:

Offline

1. Assume analytic covariance kernel $C(r_1, r_2) = exp\left(-\frac{(r_1 - r_2)^2}{L^2}\right)$.
2. Perform eigenvalue decomposition of C.

Online

3. Expand* $\beta - \bar{\beta}$ in basis of eigenvectors $\{\boldsymbol{\phi}_k\}$ of C, with random variables $\{\xi_k\}$:

$$\boxed{\beta(\omega) = \bar{\beta} + \sum_{k=1}^{K} \sqrt{\lambda_k}\boldsymbol{\phi}_k \xi_k(\omega)}$$

$\bar{\beta}$ = initial condition for β (result of deterministic inversion or spin-up)

*In practice, expansion is done on $\log(\beta)$ to avoid negative values of β.

Inference/calibration is for coefficients of KLE
$\Rightarrow$ ***significant dimension reduction***.

Sandia National Laboratories

Bayesian Calibration: Demonstration of Workflow using KLE (cont'd)

- **<u>Step 1 (*Trilinos*):</u>** Reduce $O(100K)$ dimensional problem to $O(10)$ dimensional problem using ***Karhunen-Loeve Expansion (KLE)***:

Offline

1. Assume analytic covariance kernel $C(r_1, r_2) = exp\left(-\frac{(r_1-r_2)^2}{L^2}\right)$.
2. Perform eigenvalue decomposition of C.

Online

3. Expand* $\beta - \bar{\beta}$ in basis of eigenvectors $\{\boldsymbol{\phi}_k\}$ of C, with random variables $\{\xi_k\}$:

$$\boxed{\beta(\omega) = \bar{\beta} + \sum_{k=1}^{K} \sqrt{\lambda_k} \boldsymbol{\phi}_k \xi_k(\omega)}$$

$\bar{\beta}$ = initial condition for β (result of deterministic inversion or spin-up)

*In practice, expansion is done on $\log(\beta)$ to avoid negative values of β.

Trilinos

Sandia National Laboratories

Bayesian Calibration: Demonstration of Workflow using KLE (cont'd)

- **Step 1 (*Trilinos*):** Reduce $O(100K)$ dimensional problem to $O(10)$ dimensional problem using ***Karhunen-Loeve Expansion (KLE)***:

Offline

1. Assume analytic covariance kernel $C(r_1, r_2) = exp\left(-\frac{(r_1 - r_2)^2}{L^2}\right)$.
2. Perform eigenvalue decomposition of C.

Online

3. Expand* $\beta - \bar{\beta}$ in basis of eigenvectors $\{\boldsymbol{\phi}_k\}$ of C, with random variables $\{\xi_k\}$:

$$\boxed{\beta(\omega) = \bar{\beta} + \sum_{k=1}^{K} \sqrt{\lambda_k}\boldsymbol{\phi}_k \xi_k(\omega)}$$

$\bar{\beta}$ = initial condition for β (result of deterministic inversion or spin-up)

- **Step 2 *(DAKOTA)*:** ***Polynomial Chaos Expansion (PCE)*** emulator for mismatch over surface velocity discrepancy.

*In practice, expansion is done on $\log(\beta)$ to avoid negative values of β.

Bayesian Calibration: Demonstration of Workflow using KLE (cont'd)

- **Step 1 (*Trilinos*):** Reduce $O(100K)$ dimensional problem to $O(10)$ dimensional problem using ***Karhunen-Loeve Expansion (KLE)***:

Offline

1. Assume analytic covariance kernel $C(r_1, r_2) = exp\left(-\frac{(r_1-r_2)^2}{L^2}\right)$.
2. Perform eigenvalue decomposition of C.

Online

3. Expand* $\beta - \bar{\beta}$ in basis of eigenvectors $\{\boldsymbol{\phi}_k\}$ of C, with random variables $\{\xi_k\}$:

$$\boxed{\beta(\omega) = \bar{\beta} + \sum_{k=1}^{K} \sqrt{\lambda_k}\boldsymbol{\phi}_k \xi_k(\omega)}$$

$\bar{\beta}$ = initial condition for β (result of deterministic inversion or spin-up)

- **Step 2 *(DAKOTA)*:** ***Polynomial Chaos Expansion (PCE)*** emulator for mismatch over surface velocity discrepancy.

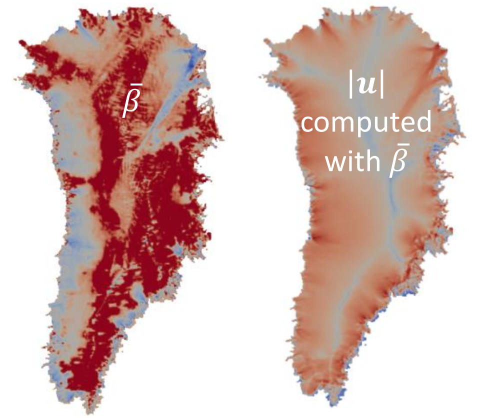
- **Step 3 *(QUESO)*:** ***Markov Chain Monte Carlo (MCMC)*** calibration using PCE emulator.
  →can obtain MAP point and posterior distributions on KLE coefficients.

*In practice, expansion is done on $\log(\beta)$ to avoid negative values of β.

Bayesian Calibration: Illustration on 4km GIS Problem

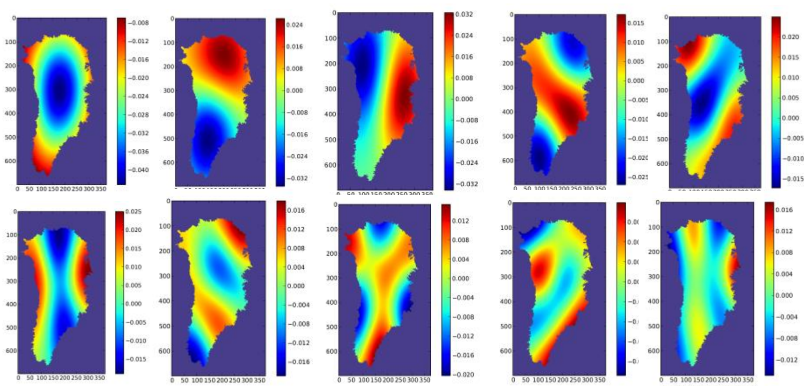
Disclaimer: results presented demonstrate that we have UQ workflow in place; quantifying uncertainty in β and SLR will require re-running with better data.

- Mean $\bar{\beta}$ field obtained through spin-up over 100 years (cheaper than inversion, gives reasonable agreement with present-day velocity field).
- Correlation length L selected s.t. slow decay of KLE eigenvalues to enable refinement *(left):* 10 KLE modes capture 27.3% of covariance energy.

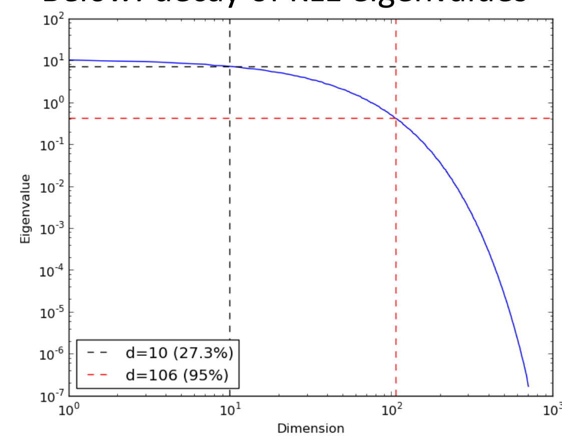


Modes 1-5:

Modes 6-10:



Below: decay of KLE eigenvalues



- Mismatch function (calculated in *Albany/FELIX*):

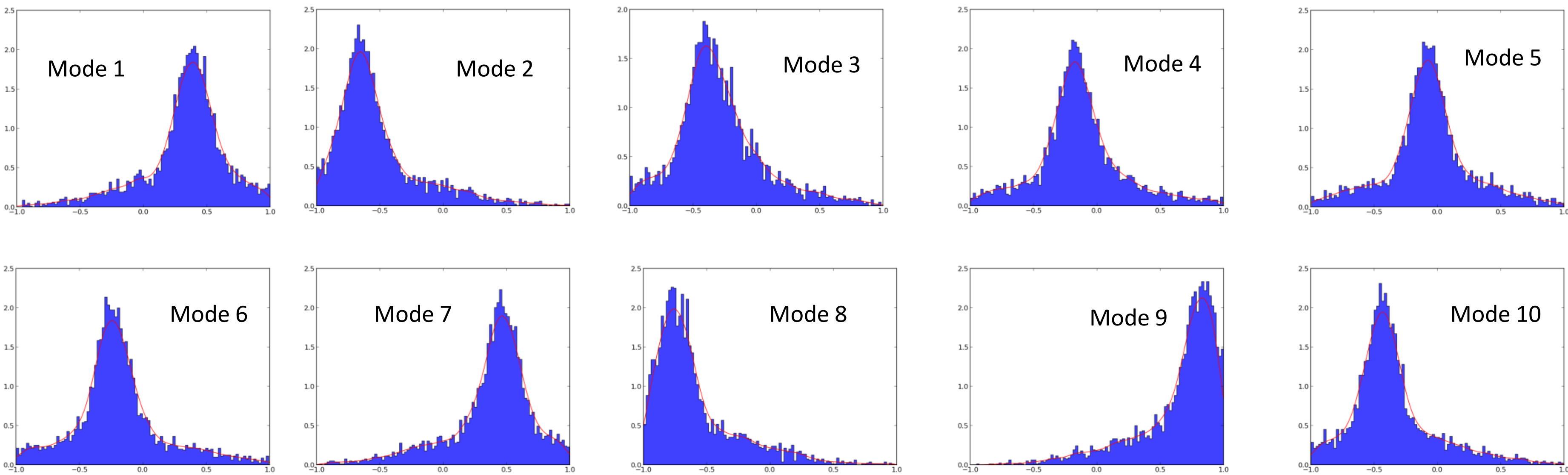
$$J(\beta) = \int_{\Gamma_{top}} \frac{1}{{\sigma_u}^2} |\boldsymbol{u} - \boldsymbol{u}^{obs}|^2 ds$$

- PCE emulator was formed for the mismatch $J(\beta)$ using uniform $[-1,1]$ prior distributions and 286 high-fidelity runs on Hopper (286 points = 3rd degree polynomial in 10D).
- For calibration, MCMC was performed on the PCE with 2K samples.

Sandia National Laboratories

Bayesian Calibration: Illustration on 4km GIS Problem (cont'd)

- ***Posterior distributions*** for 10 KLE coefficients:

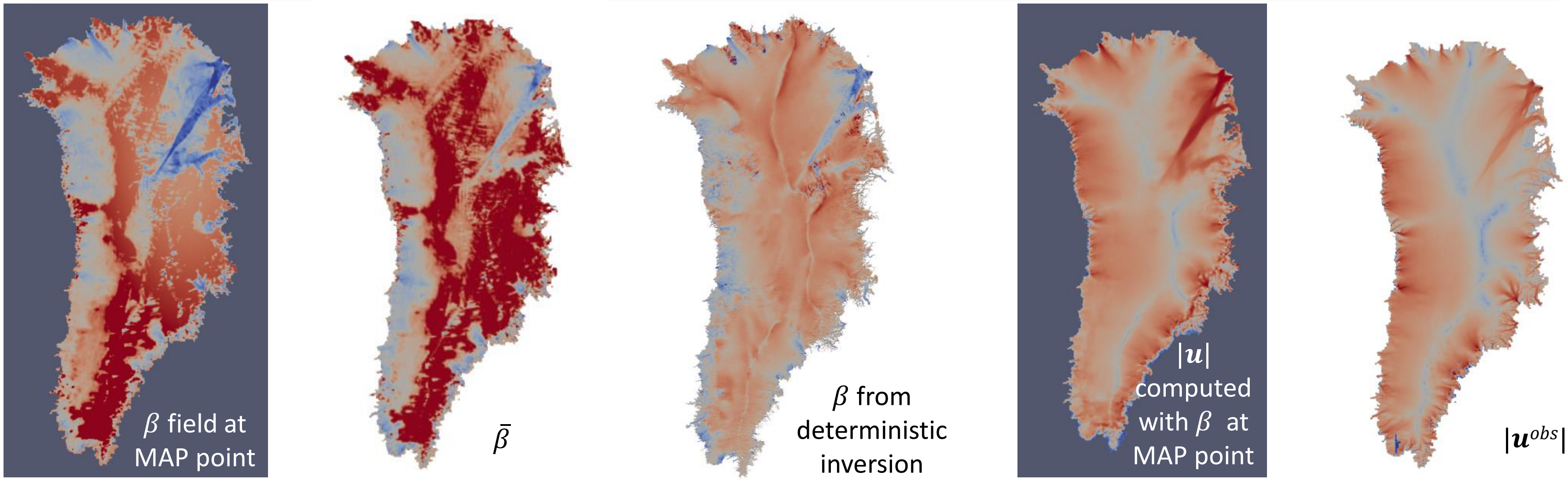


- Distributions are peaked rather than uniform ⇒ data informed the posteriors.

- ***MAP point***: $\boldsymbol{\xi} = (0.372, -0.679, -0.420, -0.189, -7.38\text{e}{-2}, -0.255, 0.449, -0.757, 0.847, -0.447)$

Sandia National Laboratories

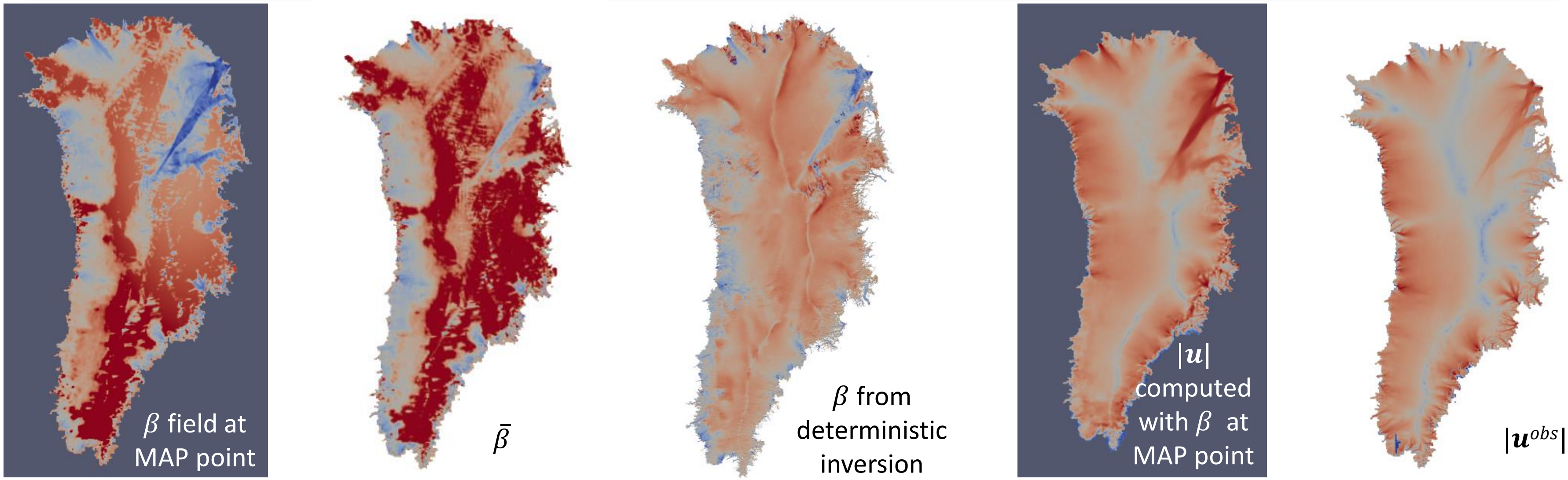
Bayesian Calibration: Illustration on 4km GIS Problem (cont'd)



- Ice is too fast at MAP point. Possible explanations:
  - Surrogate error (based on cross-validation).
  - Mean field error.
  - Bad modes (modes lack fine scale features).

Mismatch $J(\beta)$ at MAP point: $1.87 \times$ mismatch at $\bar{\beta}$

Bayesian Calibration: Illustration on 4km GIS Problem (cont'd)



- Ice is too fast at MAP point. Possible explanations:
  - Surrogate error (based on cross-validation).
  - Mean field error.
  - Bad modes (modes lack fine scale features).

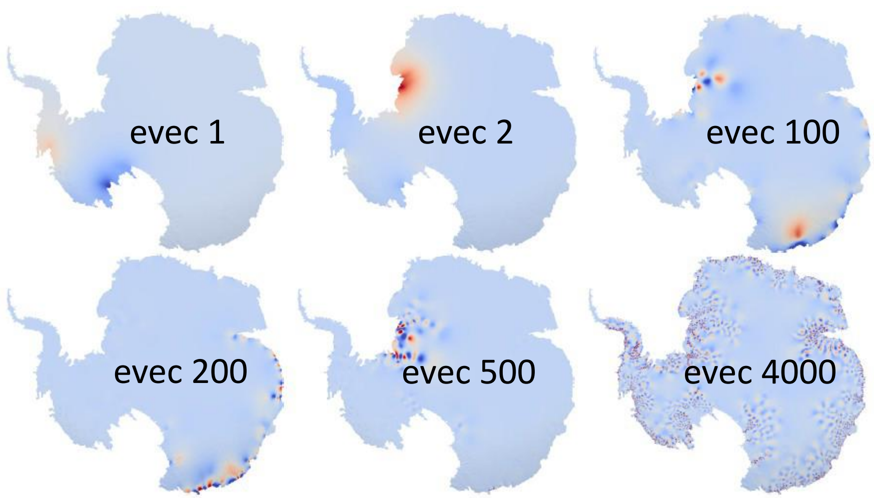
Mismatch $J(\beta)$ at MAP point: $1.87 \times$ mismatch at $\bar{\beta}$

Bayesian Calibration: Better Reduced Bases

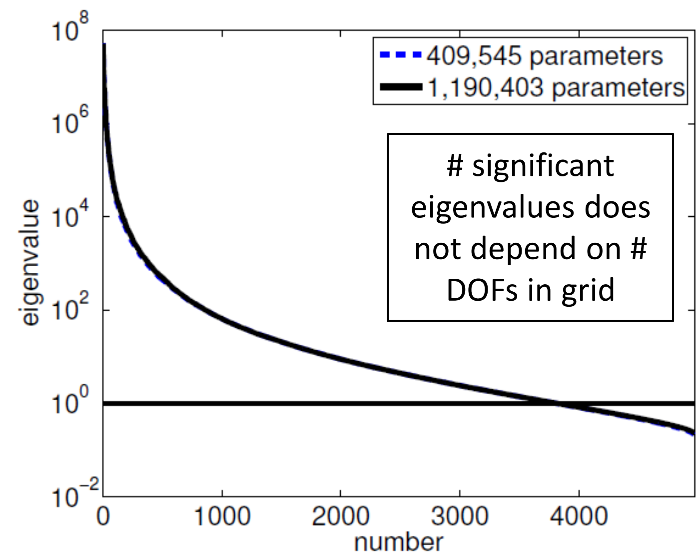
- Hessian of the merit functional (velocity mismatch) can provide a way to compute the covariance of a Gaussian posterior:

$$\boldsymbol{C}_{post} = (\boldsymbol{C}_{prior}\boldsymbol{H}_{misfit} + \boldsymbol{I})^{-1}\boldsymbol{C}_{prior}$$

- We want to limit only the most important directions (eigenvectors) of $\boldsymbol{C}_{post}$.



Figures courtesy of O. Ghattas' group (Isaac et al., 2004)



Right: log-linear plot of the spectra of a prior-preconditioned data misfit Hessian at the MAP point for two successively finer parameter/state meshes of the inverse ice sheet problem.

Sandia National Laboratories

Outline

- Motivation.
- The PISCEES project.
- The Ice Sheet Equations.
- The Albany/FELIX Steady Stress-Velocity Solver.
- Deterministic Inversion for Ice Sheet Initialization.
- Dynamic Simulations of Ice Sheet Evolution.
- Uncertainty Quantification.
- **Summary & future work.**

Summary and Ongoing Work

Summary: this talk described…

- Equations, algorithms, software used in ice sheet modeling.
- The development of a finite element land ice solver known as *Albany/FELIX* written using the libraries of the *Trilinos* libraries.
- Coupling of *Albany/FELIX* to the *CISM* and *MPAS LI* codes for transient simulations of ice sheet evolution.
- Some advanced concepts in ice sheet modeling: deterministic inversion, UQ.

Ongoing/future work:

- Science runs using *CISM-Albany* and *MPAS-Albany*.
- Deploy UQ workflow with better basis than KLE (e.g., Hessian eigenvectors).
- Porting of code to new architecture machines (GPUs, multi-core, many-core).
- Delivering code to climate community and coupling to earth system models.

Funding/Acknowledgements

Support for this work was provided through Scientific Discovery through Advanced Computing (**SciDAC**) projects funded by the U.S. Department of Energy, Office of Science (**OSCR**), Advanced Scientific Computing Research and Biological and Environmental Research (**BER**) → **PISCEES SciDAC Application Partnership.**

PISCEES team members: *K. Evans, M. Gunzburger, M. Hoffman, C. Jackson, P. Jones, W. Lipscomb, M. Perego, S. Price, A. Salinger, I. Tezaur, R. Tuminaro, P. Worley.*

Trilinos/DAKOTA collaborators*: M. Eldred, J. Jakeman, E. Phipps, L. Swiler.*

Computing resources: *NERSC, OLCF.*

Thank you! Questions?

References

[1] M.A. Heroux *et al.* "An overview of the Trilinos project." *ACM Trans. Math. Softw.* **31**(3) (2005).

[2] A.G. Salinger *et al*. "Albany: Using Agile Components to Develop a Flexible, Generic Multiphysics Analysis Code", *Comput. Sci. Disc.* (submitted, 2015).

[3] **I. Tezaur**, M. Perego, A. Salinger, R. Tuminaro, S. Price. "*Albany/FELIX*: A Parallel, Scalable and Robust Finite Element Higher-Order Stokes Ice Sheet Solver Built for Advanced Analysis", *Geosci. Model Develop.* 8 (2015) 1-24.

[4] **I. Tezaur**, R. Tuminaro, M. Perego, A. Salinger, S. Price. "On the scalability of the *Albany/FELIX* first-order Stokes approximation ice sheet solver for large-scale simulations of the Greenland and Antarctic ice sheets", *Procedia Computer Science* 51 (2015) 2026-2035.

[5] R.S. Tuminaro, **I. Tezaur**, M. Perego, A.G. Salinger. "A Hybrid Operator Dependent Multi-Grid/Algebraic Multi-Grid Approach: Application to Ice Sheet Modeling", *SIAM J. Sci. Comput.* (in prep).

[6] R. Tuminaro. "ML's SemiCoarsening Feature, Addition to ML 5.0 Smoothed Aggregation User's Guide" , Sandia National Laboratories Report, SAND2006-2649, Sandia National Laboratories, Albuquerque, NM, 2014.

References (cont'd)

[7] S. Shannon, *et al.* "Enhanced basal lubrication and the contribution of the Greenland ice sheet to future sea-level rise", *P. Natl. Acad. Sci.,* 110 (2013) 14156-14161.

[8] P. Fretwell, *et al.* "BEDMAP2: Improved ice bed, surface, and thickness datasets for Antarctica", *The Cryosphere* 7(1) (2013) 375-393.

[9] F. Pattyn. "Antarctic subglacial conditions inferred from a hybrid ice sheet/ice stream model", *Earth and Planetary Science Letters 295 (2010).*

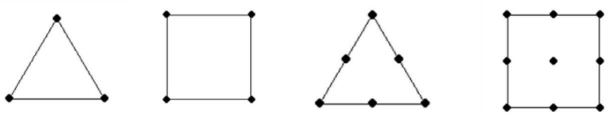
[10] M. Perego, S. Price, G. Stadler. "Optimal Initial Conditions for Coupling Ice Sheet Models to Earth System Models", *J. Geophys. Res.* 119 (2014) 1894-1917.

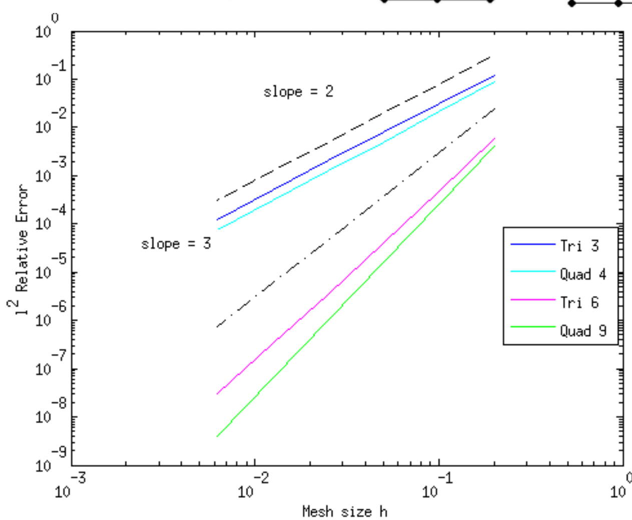
[11] R. Tuminaro, M. Perego, **I. Tezaur**, A. Salinger, S. Price. "A matrix dependent/algebraic multigrid approach for extruded meshes with applications to ice sheet modeling", submitted to *SIAM J. Sci. Comput.*

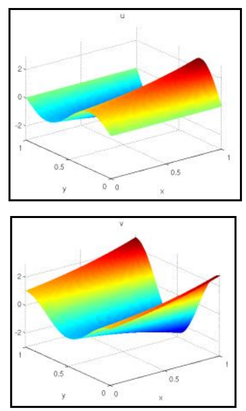
[12] Solomon, S., Qin, D., Manning, M., Chen, Z., Marquis, M., Averyt, K., Tignor, M., and Miller, H. "Climate change 2007: The physical science basis, Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change", Cambridge Univ. Press, Cambridge, UK, 2007.

Appendix: Verification/Mesh Convergence Studies

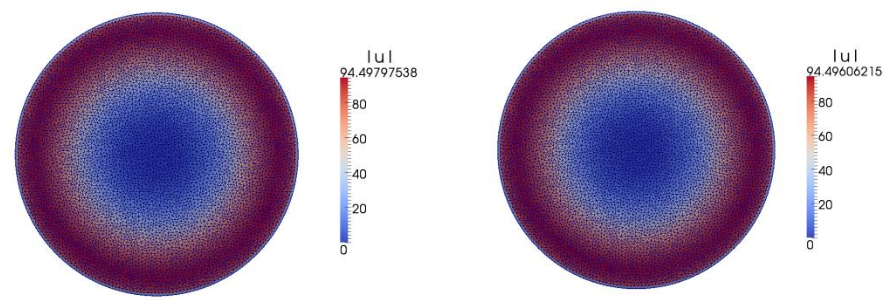
Stage 1: solution verification on 2D MMS problems we derived.







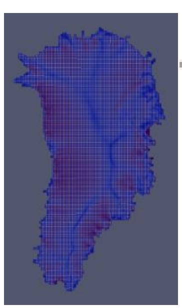
Stage 2: code-to-code comparisons on canonical ice sheet problems.

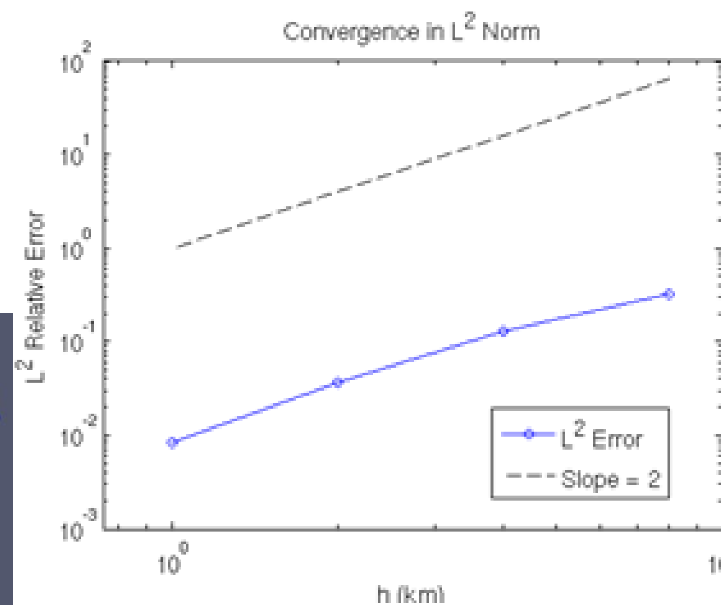


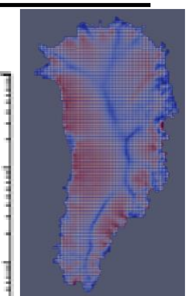
Albany/FELIX *LifeV*

Stage 3: full 3D mesh convergence study on Greenland w.r.t. reference solution.

> *Are the Greenland problems resolved?*
> *Is theoretical convergence rate achieved?*



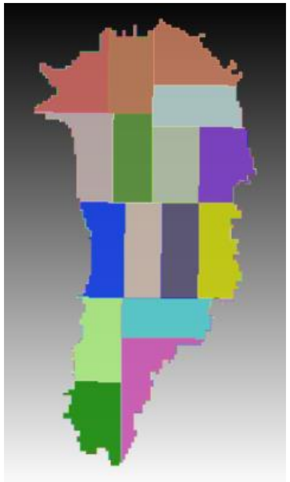




Appendix: Mesh Partitioning & Vertical Refinement

Mesh convergence studies led to some useful practical recommendations
(for ice sheet modelers *and* geo-scientists)!

- ***Partitioning matters***: good solver performance obtained with 2D partition of mesh (all elements with same x, y coordinates on same processor - *right*).



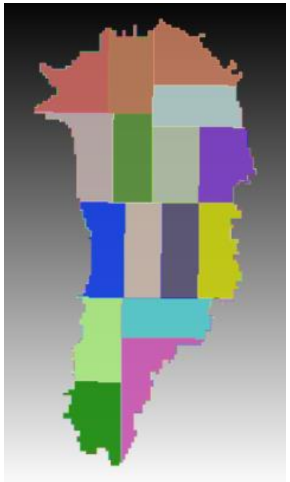
- ***Number of vertical layers matters:*** more gained in refining # vertical layers than horizontal resolution *(below – relative errors for Greenland).*

Horiz. res.\vert. layers	5	10	20	40	80
8km	2.0e-1				
4km	9.0e-2	7.8e-2			
2km	4.6e-2	2.4e-2	2.3e-2		
1km	3.8e-2	8.9e-3	5.5e-3	5.1e-3	
500m	3.7e-2	6.7e-3	1.7e-3	3.9e-4	8.1e-5

Appendix: Mesh Partitioning & Vertical Refinement

Mesh convergence studies led to some useful practical recommendations (for ice sheet modelers *and* geo-scientists)!

- ***Partitioning matters***: good solver performance obtained with 2D partition of mesh (all elements with same x, y coordinates on same processor - *right*).
- ***Number of vertical layers matters:*** more gained in refining # vertical layers than horizontal resolution *(below – relative errors for Greenland).*



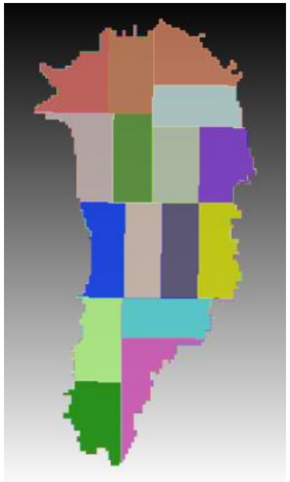
Horiz. res.\vert. layers	**5**	**10**	**20**	**40**	**80**
8km	2.0e-1				
4km	9.0e-2	7.8e-2			
2km	4.6e-2	2.4e-2	2.3e-2		
1km	3.8e-2	8.9e-3	5.5e-3	5.1e-3	
500m	3.7e-2	6.7e-3	1.7e-3	3.9e-4	8.1e-5

Sandia National Laboratories

Appendix: Mesh Partitioning & Vertical Refinement

Mesh convergence studies led to some useful practical recommendations (for ice sheet modelers *and* geo-scientists)!

- ***Partitioning matters***: good solver performance obtained with 2D partition of mesh (all elements with same x, y coordinates on same processor - *right*).



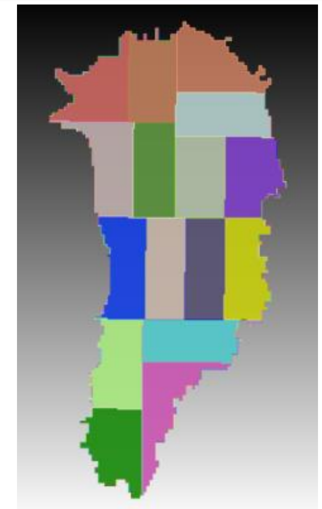
- ***Number of vertical layers matters:*** more gained in refining # vertical layers than horizontal resolution *(below – relative errors for Greenland).*

Horiz. res.\vert. layers	5	10	20	40	80
8km	2.0e-1				
4km	9.0e-2	7.8e-2			
2km	4.6e-2	2.4e-2	2.3e-2		
1km	3.8e-2	8.9e-3	5.5e-3	5.1e-3	
500m	3.7e-2	6.7e-3	1.7e-3	3.9e-4	8.1e-5

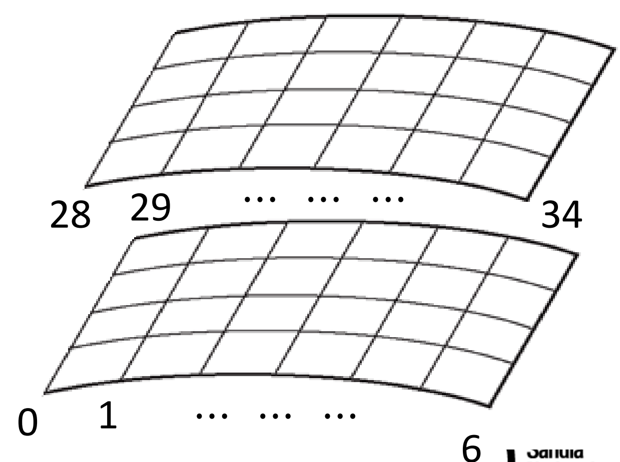
Vertical refinement to 20 layers recommended for 1km resolution over horizontal refinement.

Appendix: Importance of Node Ordering & Mesh Partitioning

Our studies revealed that **node ordering** and **mesh partitioning** matters for linear solver performance, especially for the ILU preconditioner!



- It is essential that incomplete factorization accurately captures vertical coupling, which is dominant due to anisotropic mesh.
- This is accomplished by:
  - Ensuring all points along a vertically extruded grid line reside within a single processor ("**2D mesh partitioning**"; top right).
  - Ordering the equations such that grid layer k's nodes are ordered before all dofs associated with grid layer $k+1$ ("**row-wise ordering**"; bottom right).



Appendix: Performance-Portability via *Kokkos*

We need to be able to run *Albany/FELIX* on ***new architecture machines*** (hybrid systems) and ***manycore devices*** (multi-core CPU, NVIDIA GPU, Intel Xeon Phi, etc.) .

- ***Kokkos***: *Trilinos* library that provides performance portability across diverse devises with different memory models.
  - A *programming model* as much as a software library.
  - Provides automatic access to OpenMP, CUDA, Pthreads, etc.
  - Templated meta-programming: parallel_for, parallel_reduce (templated on an *execution space*).
  - Memory layout abstraction ("array of structs" vs. "struct of arrays", locality).

With *Kokkos*, you write an algorithm **once**, and just change a template parameter to get the optimal data layout for your hardware.

Appendix: Performance-Portability via *Kokkos*

We need to be able to run *Albany/FELIX* on ***new architecture machines*** (hybrid systems) and ***manycore devices*** (multi-core CPU, NVIDIA GPU, Intel Xeon Phi, etc.) .

- ***Kokkos***: *Trilinos* library that provides performance portability across diverse devises with different memory models.

  - A *programming model* as much as a software library.
  - Provides automatic access to OpenMP, CUDA, Pthreads, etc.
  - Templated meta-programming: parallel_for, parallel_reduce (templated on an *execution space*).
  - Memory layout abstraction ("array of structs" vs. "struct of arrays", locality).

With *Kokkos*, you write an algorithm **once**, and just change a template parameter to get the optimal data layout for your hardware.

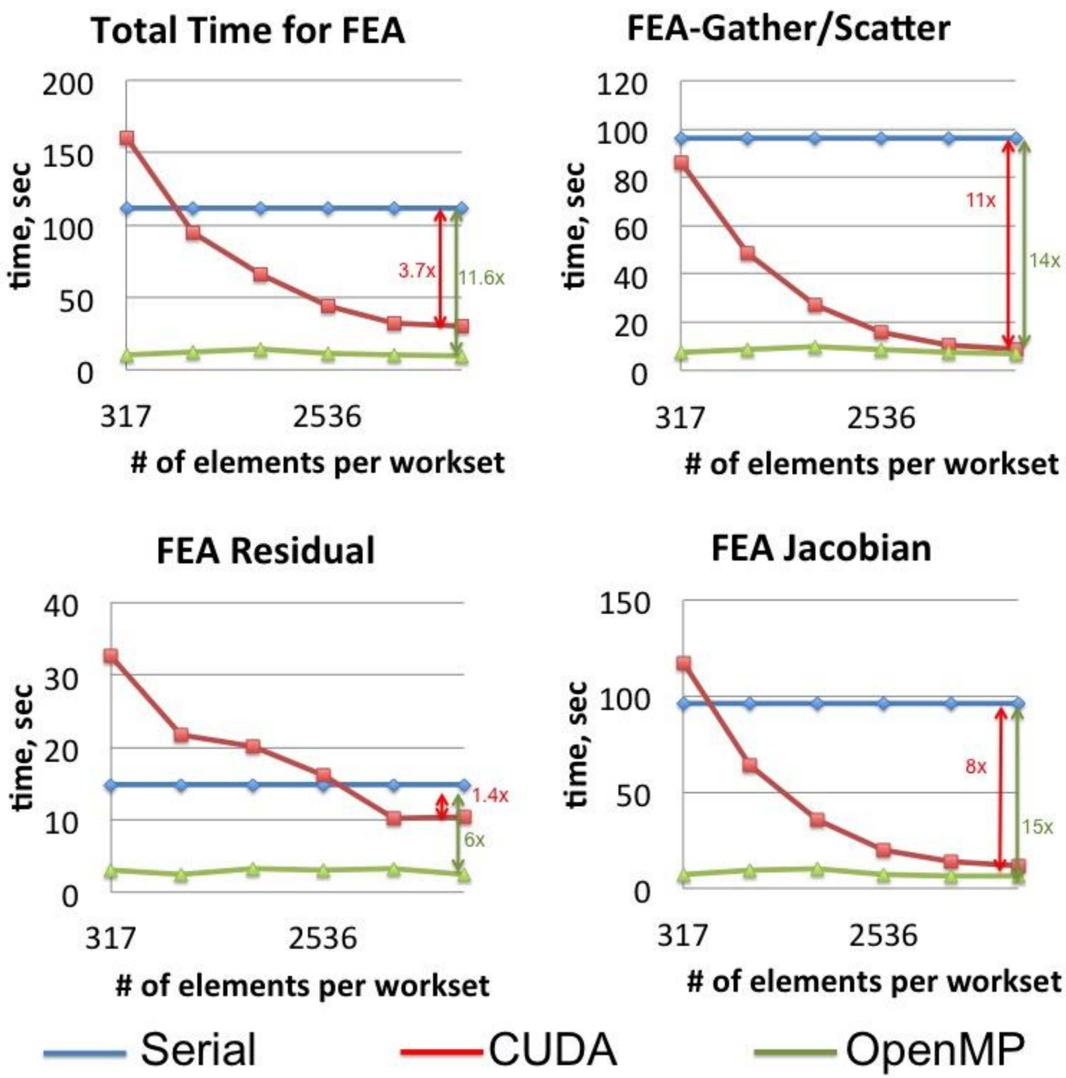
- **Finite element assembly** in *Albany* has recently been rewritten using *Kokkos* functors.

Appendix: *Kokkos*-ification of Finite Element Assembly

```
typedef Kokkos::OpenMP ExecutionSpace;
//typedef Kokkos::CUDA ExecutionSpace;
//typedef Kokkos::Serial ExecutionSpace;
template<typename ScalarT>
vectorGrad<ScalarT>::vectorGrad()
{
Kokkos::View<ScalarT****, ExecutionSpace> vecGrad("vecGrad", numCells, numQP, numVec, numDim);
}
**********************************************
template<typename ScalarT>
void vectorGrad<ScalarT>::evaluateFields()
{
  Kokkos::parallel_for<ExecutionSpace> (numCells, *this);
}
**********************************************
template<typename ScalarT>
KOKKOS_INLINE_FUNCTION
void vectorGrad<ScalarT>:: operator() (const int cell) const
{
  for (int cell = 0; cell < numCells; cell++)
  for (int qp = 0; qp < numQP; qp++) {
    for (int dim = 0; dim < numVec; dim++) {
      for (int i = 0; i < numDim; i++) {
        for (int nd = 0; nd < numNode; nd++) {
          vecGrad(cell, qp, dim, i) += val(cell, nd, dim) * basisGrad(nd, qp, i);
} } } } }
```

ExecutionSpace parameter tailors code for device (e.g., OpenMP, CUDA, etc.)

Appendix: Performance-Portability via *Kokkos*: 20km GIS Problem



Shannon: *32 nodes*

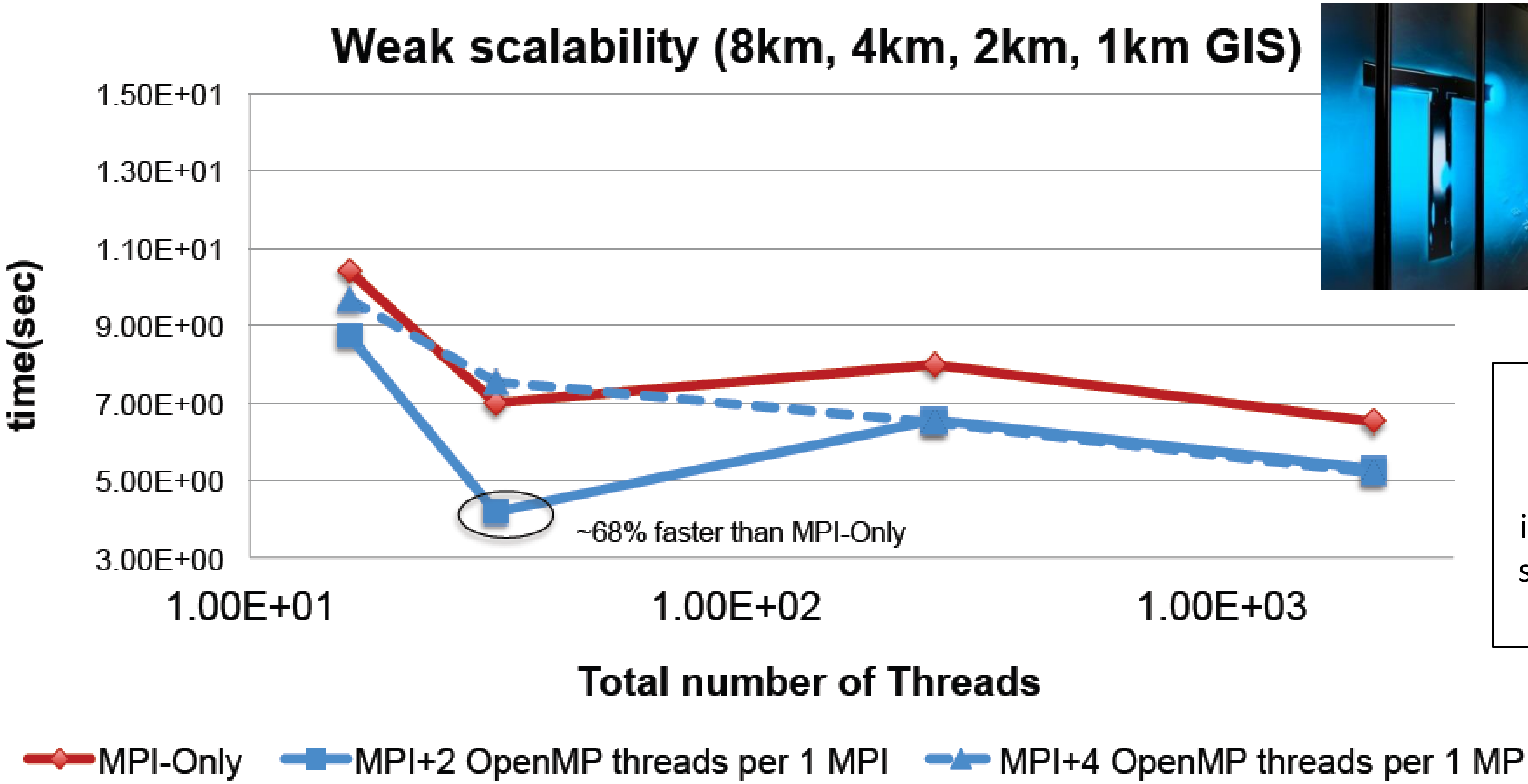
- *Two 8-core Sandy Bridge Xeon E5-2670 @ 2.6GHz (HT deactivated) per node.*
- *128GB DDR3 memory per node*
- *2x NVIDIA K20x per node.*

FEA Residual: less work done by GPU, so kernel launch overhead becomes significant.

"# of elements/workset" = threading index (allows for on-node parallelism)

Sandia National Laboratories

Appendix: Performance-Portability via *Kokkos*: Weak Scalability for GIS on *Titan*



Increasing # OpenMP threads can increase thread synchronization overheads.

Titan: 18,688 AMD Opteron nodes

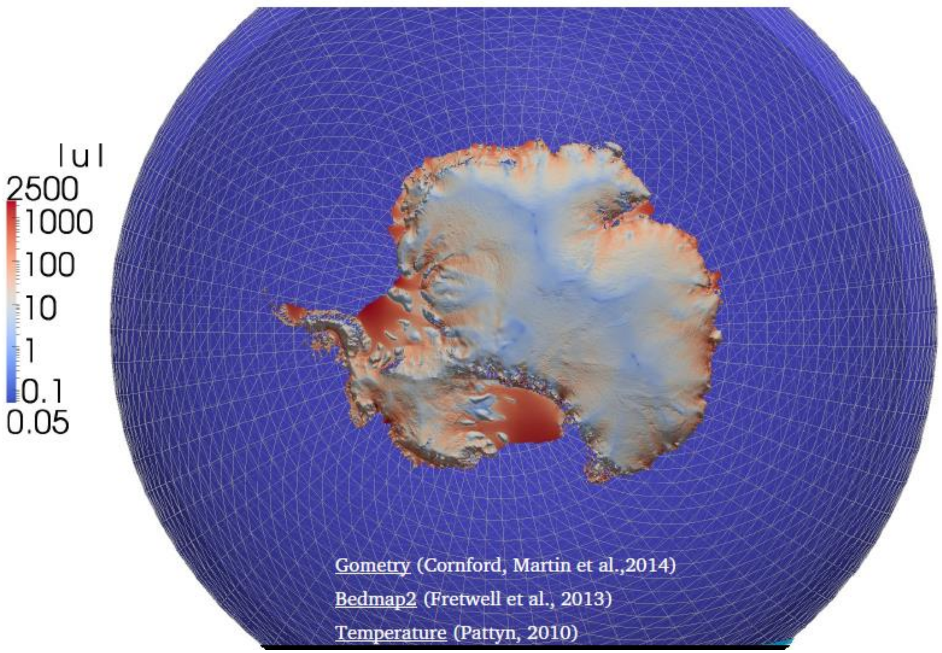
- 16 cores per node
- 1 K20X Kepler GPUs per node
- 32GB + 6GB memory per node

MPI+CUDA results on *Titan* coming soon! (waiting for gcc 4.7.2 compiler support for Cray)

Sandia National Laboratories

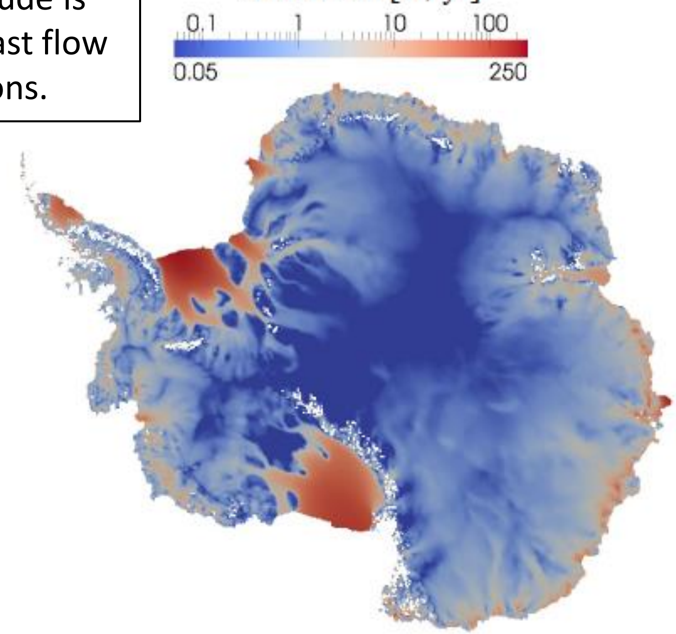
Appendix: Spherical Grids

Surface velocity magnitude [m/yr], ice sheet thickness not at scale (100 X)



Relative difference in surface velocity magnitude is 10% in fast flow regions.

magnitude of surface velocity difference [m/yr]

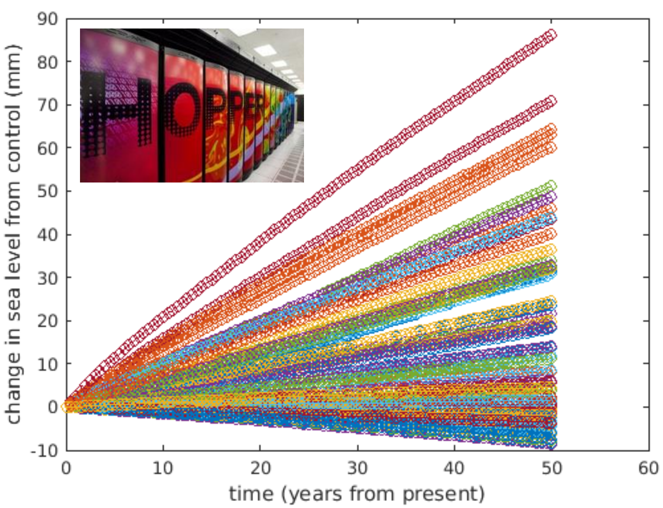


- Current ice sheet models are derived using planar geometries – reasonable, especially for Greenland.
- The effect of Earth's curvature is largely unknown – may be nontrivial for Antarctica.
- We have derived a FO Stokes model on sphere using stereographic projection.

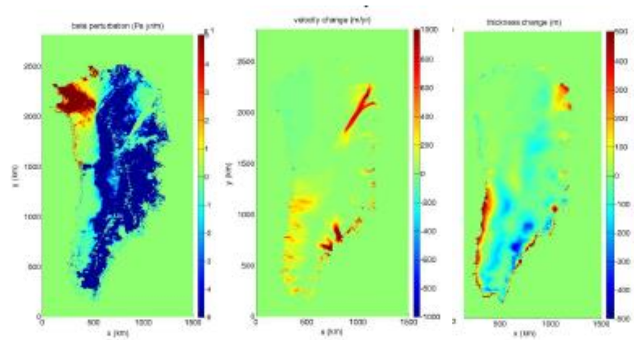
Appendix: Forward Propagation: Illustration on 4km GIS Problem

Procedure:

- We first ran 66* *CISM-Albany* high-fidelity simulations on Hopper with β sampled from a uniform $[-1,1]$ distribution and no forcing for 50 years.



Left: SLR distribution from ensemble of 66 high-fidelity simulations (differenced against control run using the $\bar{\beta}$ distribution). ***All 66 runs ran to completion out-of-the-box on Hopper!***



Above: β, velocity and thickness perturbations. Ice thickness changed > 500m in some places.

- We then used the results of these runs to create a PCE emulator for the SLR.
- Using emulator, propagated posterior distributions computed in Bayesian calibration (using KLE) through the model to get posteriors on SLR (MCMC on PCE emulator with 2K samples).

*66 points = 2D polynomial in 10D.

Sandia National Laboratories

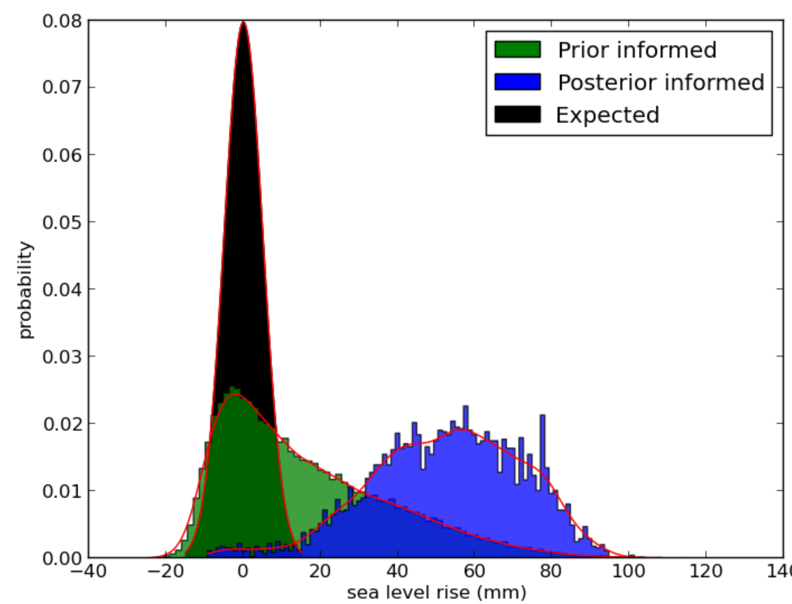
Appendix: Forward Propagation: Illustration on 4km GIS Problem (cont'd)

Expected (black): normal distribution centered around 0 SLR since no forcing.

Prior informed (green): uniform distribution translates to distribution skewed w.r.t. model outputs.

- Larger fraction of the ice sheet currently has a β value that forces no (or slow) basal sliding.
- Areas with little sliding: not affected by increase in β, but greatly affected by decrease in β (velocity in these regions will change significantly from initial condition).
- Since we sample from a uniform distribution when perturbing β, we expect to see a disproportionately large signal when reducing β vs. increasing it.

PDF of SLR



Posterior informed (blue): centered on positive tail of prior – not consistent with observations.

- Could be due to "ad hoc" β used as mean field (spin-up over 100 years).
- May be that emulator was been built with a (non-physical) positive mass balance while calibration was done on present-day observations (consistent with ice losing mass).

Sandia National Laboratories

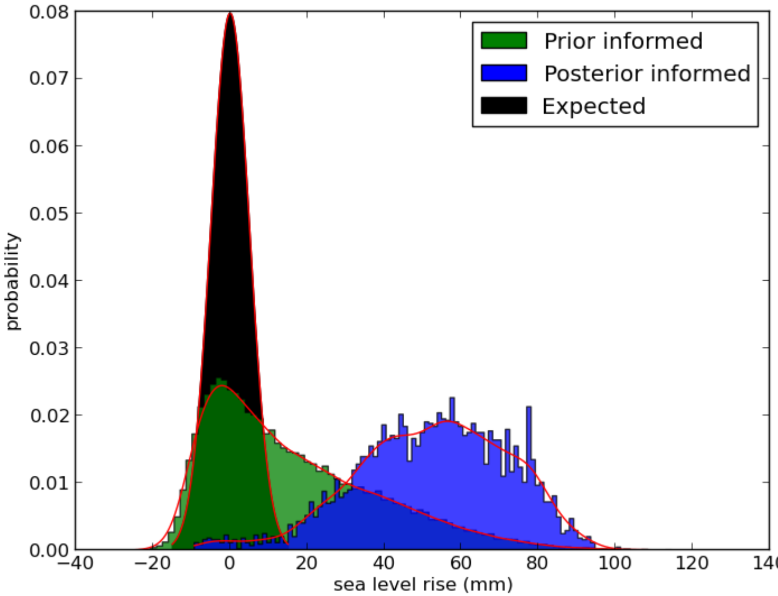
Appendix: Forward Propagation: Illustration on 4km GIS Problem (cont'd)

Expected (black): normal distribution centered around 0 SLR since no forcing.

Prior informed (green): uniform distribution translates to distribution skewed w.r.t. model outputs.

- Larger fraction of the ice sheet currently has a β value that forces no (or slow) basal sliding.
- Areas with little sliding: not affected by increase in β, but greatly affected by decrease in β (velocity in these regions will change significantly from initial condition).
- Since we sample from a uniform distribution when perturbing β, we expect to see a disproportionately large signal when reducing β vs. increasing it.

PDF of SLR



Posterior informed (blue): centered on positive tail of prior – not consistent with observations.

- Could be due to "ad hoc" β used as mean field (spin-up over 100 years).
- May be that emulator was been built with a (non-physical) positive mass balance while calibration was done on present-day observations (consistent with ice losing mass).

Results illustrate that we have in place all steps of our UQ workflow;
they are NOT yet actual uncertainty bounds for sea-level rise.
Next step: repeat UQ procedure with better modes, surrogates and $\bar{\beta}$.

Sandia National Laboratories