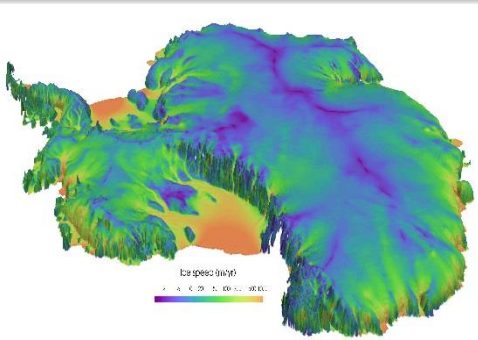
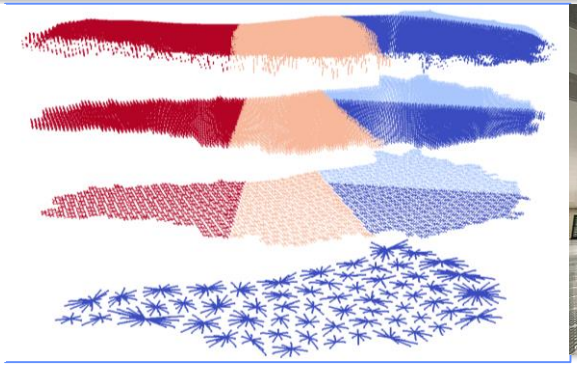
Exceptional service in the national interest

Sandia National Laboratories





Next-generation modeling & simulation of large-scale ice sheets towards probabilistic sea-level change projections

Irina K. Tezaur

Extreme Scale Data Science & Analytics Department
Sandia National Laboratories, Livermore, CA.

MICDE Seminar Ann Arbor, MI October 25, 2017

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

SAND2017-10641PE

Funding/Acknowledgements

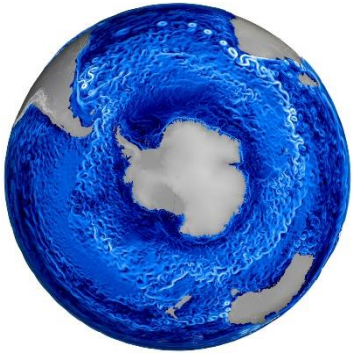
Support for this work was provided through Scientific Discovery through Advanced Computing (**SciDAC**) projects funded by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research (**ASCR**) and Biological and Environmental Research (**BER**) → **SciDAC Application Partnership.**

In collaboration with: *L. Bertagna, **M. Eldred**, **J. Jakeman**, **M. Perego**, **A. Salinger**, **R. Tuminaro**, **J. Watkins** [SNL]; **I. Demeshko**, **M. Hoffman**, **S. Price** [LANL]; **K. Evans**, **J. Kennedy**, P. Worley [ORNL]; D. Martin [LBNL]; C. Jackson [UT Austin]; G. Stadler [NYU]; M. Gunzburger [FSU]; L. Zhu [USC]; J. Bassis [U Michigan].*

Computing resources: *NERSC, OLCF.*

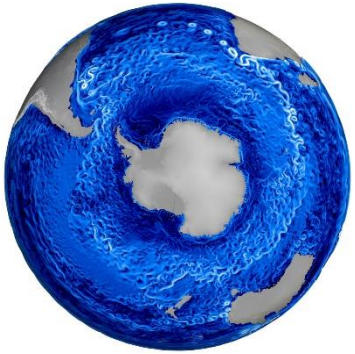
Outline

1. Background
   - Motivation for climate & land-ice modeling
   - Earth system models & projects
   - Land-ice equations
2. Algorithms and software
   - Albany/FELIX steady stress-velocity solver
     - Meshes/data
     - Solvers & preconditioners
     - Performance-portability
     - Ice sheet initialization/UQ
   - CISM/MPAS-Albany for dynamic simulations
     - Velocity-thickness coupling
     - Validation
     - Towards science runs
3. Ongoing & future work
4. Summary



Outline

1. **Background**
   - **Motivation for climate & land-ice modeling**
   - Earth system models & projects
   - Land-ice equations
2. Algorithms and software
   - Albany/FELIX steady stress-velocity solver
     - Meshes/data
     - Solvers & preconditioners
     - Performance-portability
     - Ice sheet initialization/UQ
   - CISM/MPAS-Albany for dynamic simulations
     - Velocity-thickness coupling
     - Validation
     - Towards science runs
3. Ongoing & future work
4. Summary



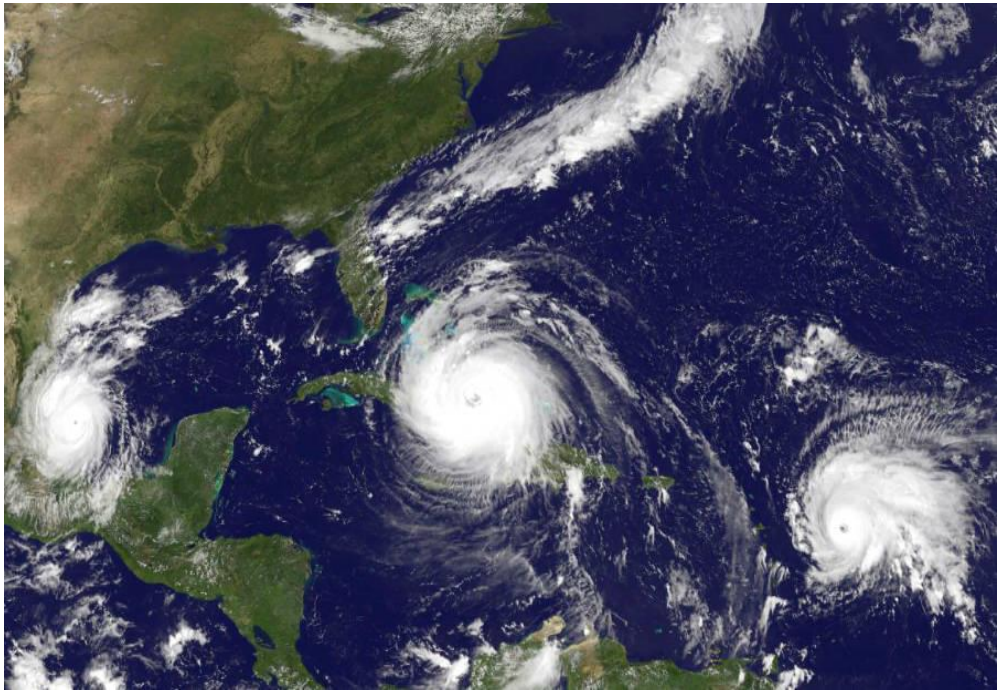
Motivation

Recent events prompt concern about global **sea-level rise** (SLR) and **climate change**.

Motivation

Recent events prompt concern about global **sea-level rise** (SLR) and **climate change**.

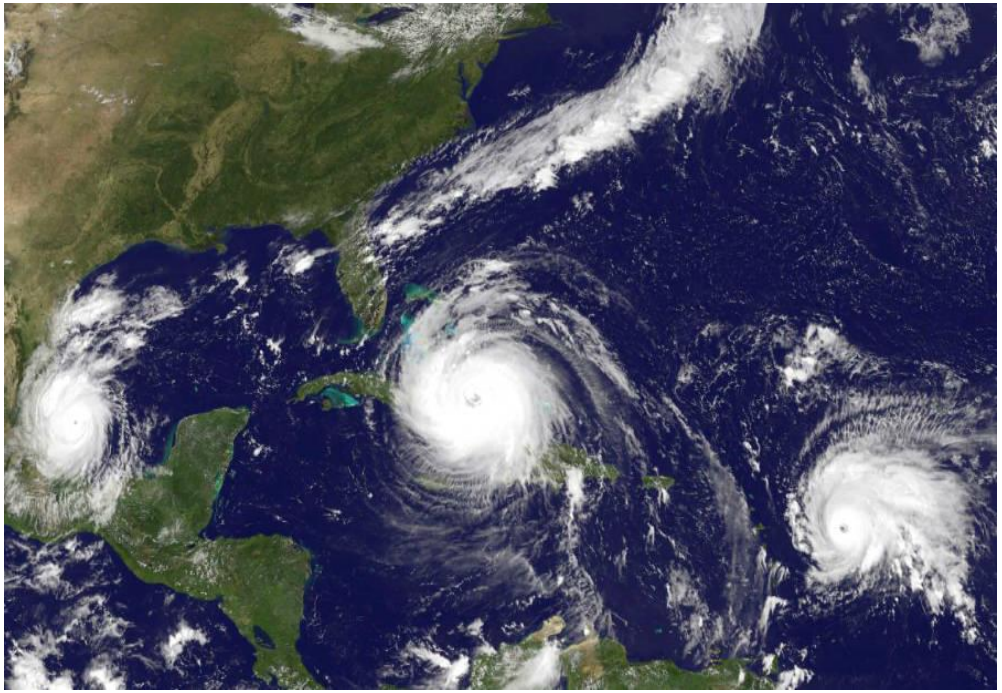
- *September 2017*: several massive **hurricanes** (Harvey, Irma, Maria) devastate Gulf Coast



Motivation

Recent events prompt concern about global **sea-level rise** (SLR) and **climate change**.

- *September 2017*: several massive **hurricanes** (Harvey, Irma, Maria) devastate Gulf Coast
- 8 of the 10 most destructive hurricanes of all time occurred during *past 15 years**.

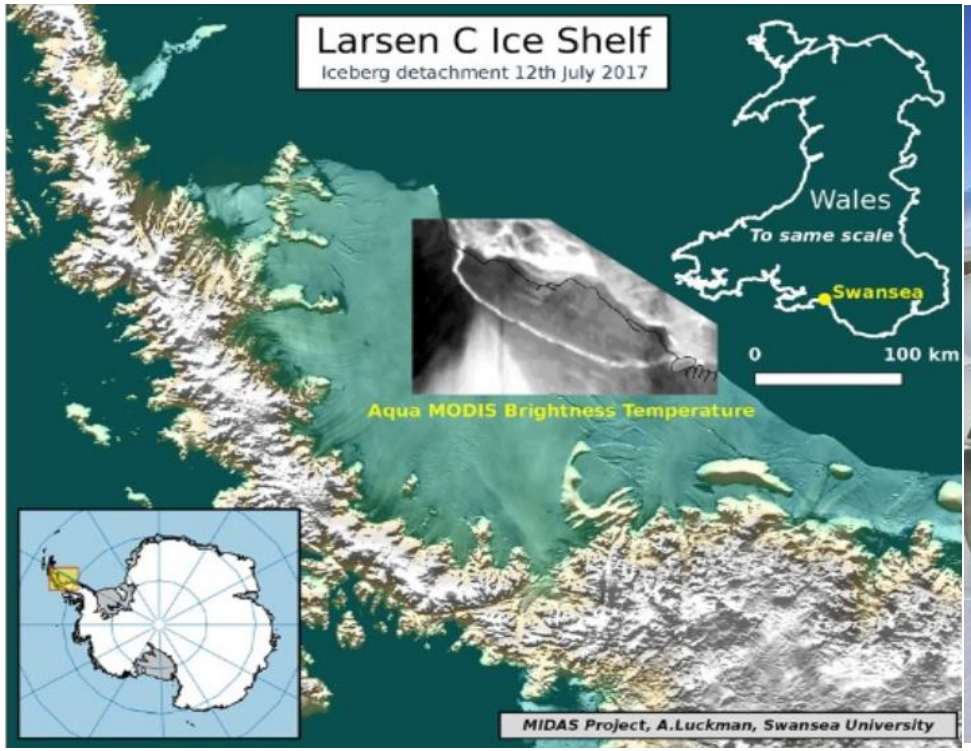


* https://energy.gov/science-innovation/climate-change.

Motivation

Recent events prompt concern about global **sea-level rise** (SLR) and **climate change**.

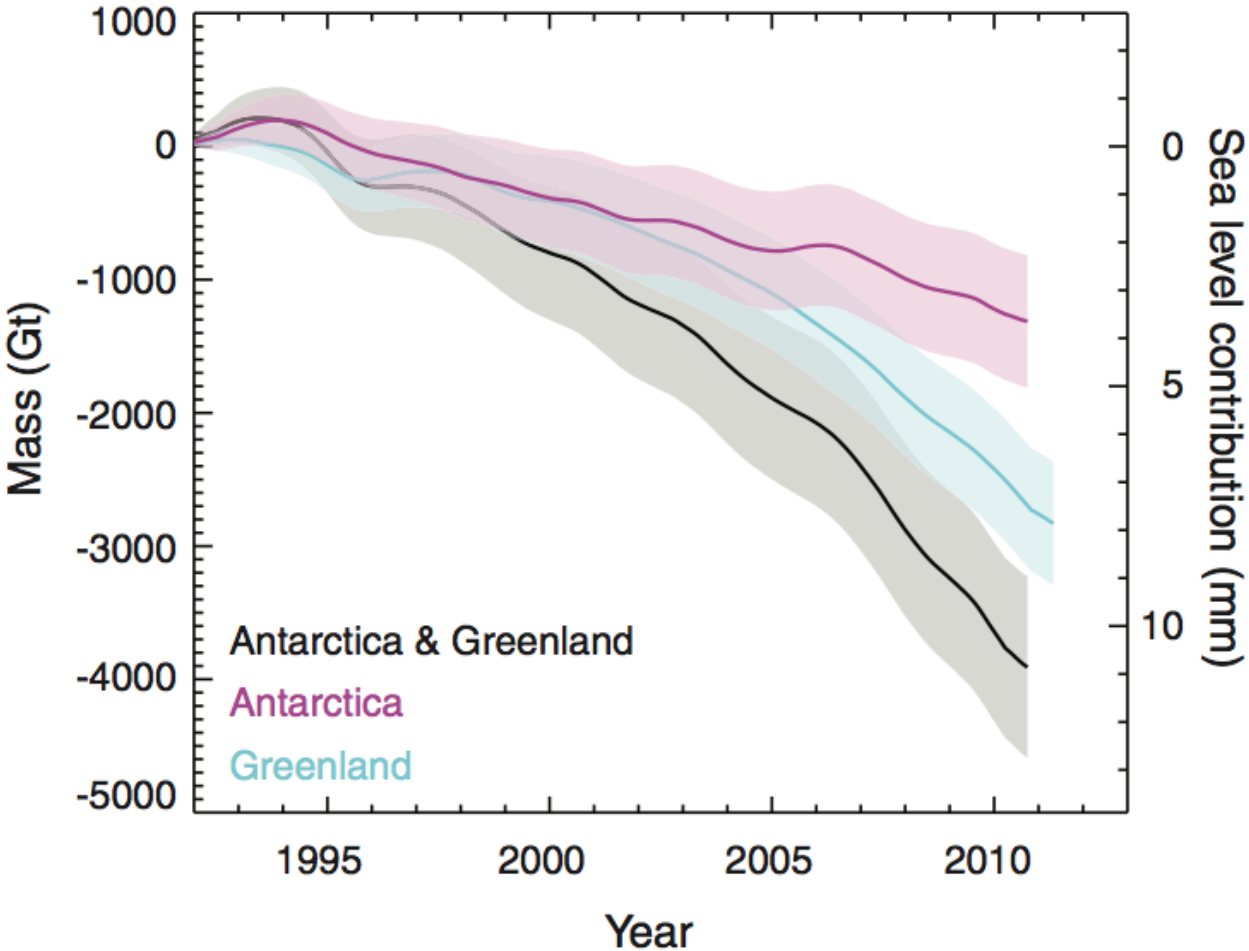
- *July 2017*: **massive iceberg** weighing > 1 trillion tons breaks away from **Larsen C** ice shelf in western Antarctica.



Right: Nov. 2016 aerial NASA photo showing rift in Larsen C ice shelf.

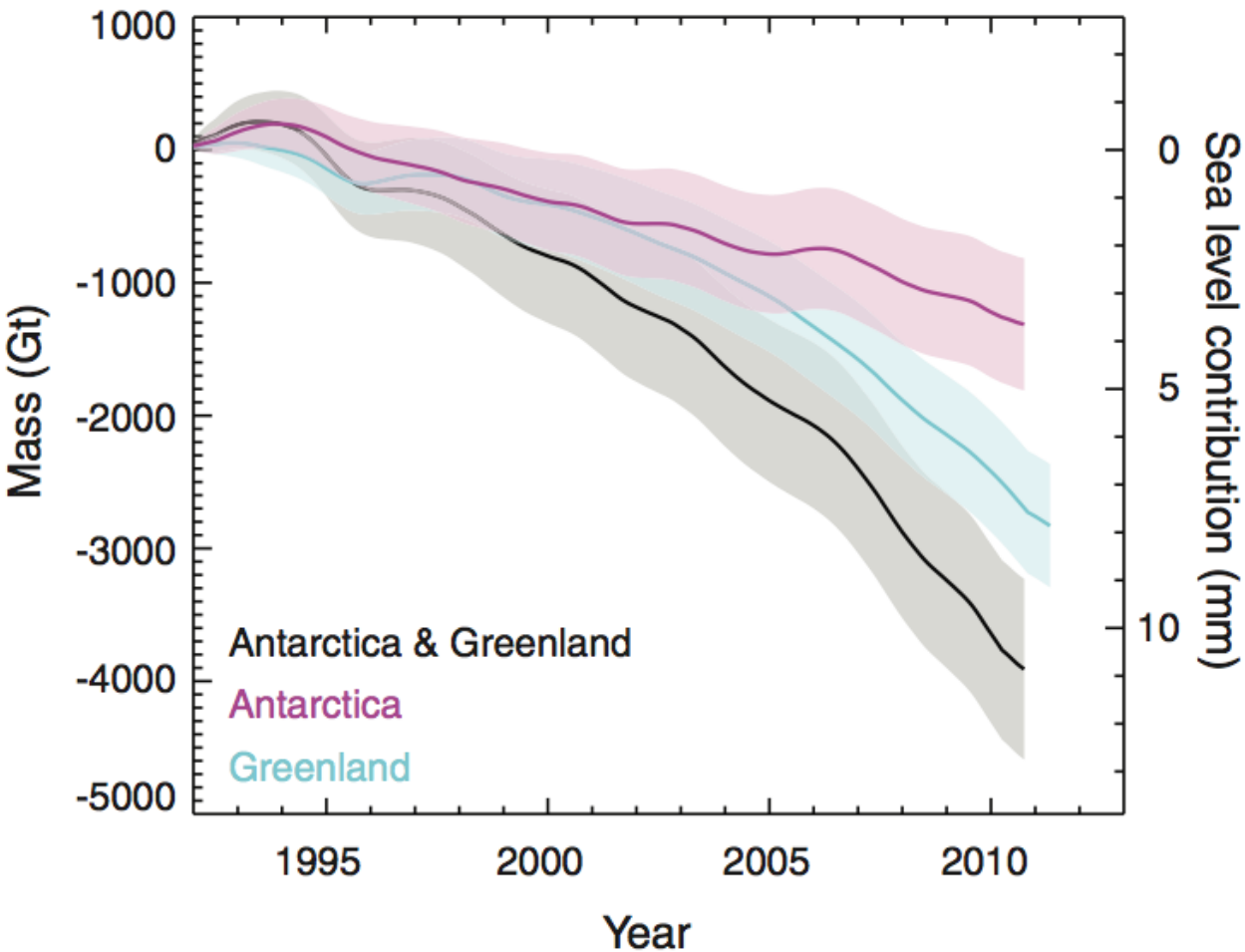
Motivation

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



Motivation

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

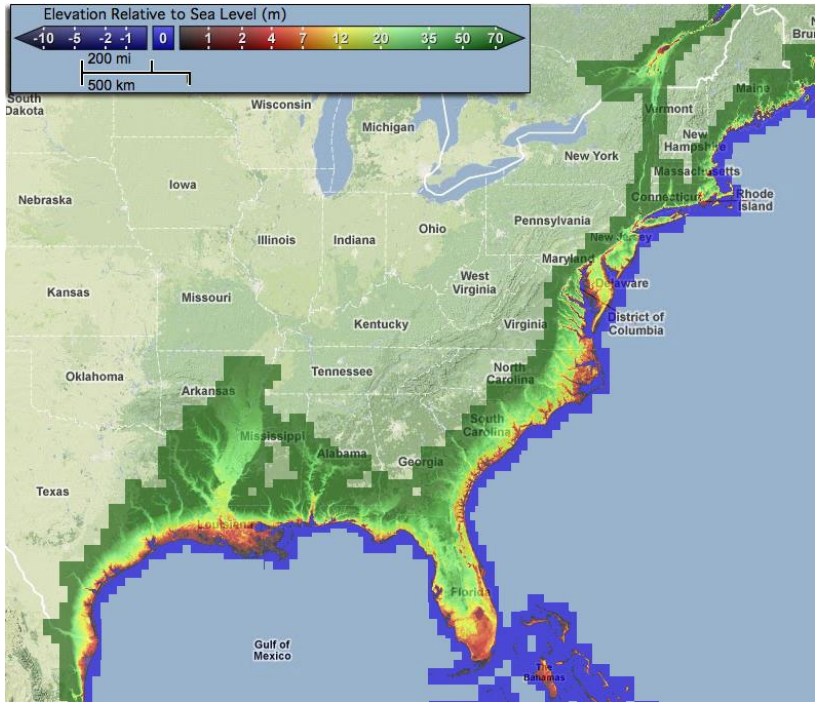


Motivation

Some estimates* of potential sea-level rise:

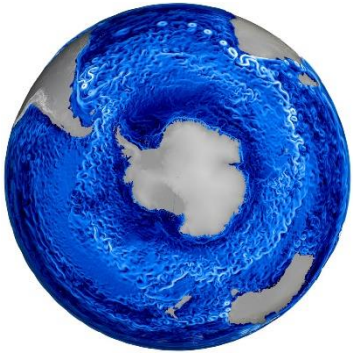
- *Full deglaciation:* sea level could rise up to ~65 meters.
- *Potential contributions to sea level rise by ice sheet*:
  - Greenland ice sheet: ~7 meters
  - East Antarctic ice sheet: ~53 meters
  - West Antarctic ice sheet: ~5 meters

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



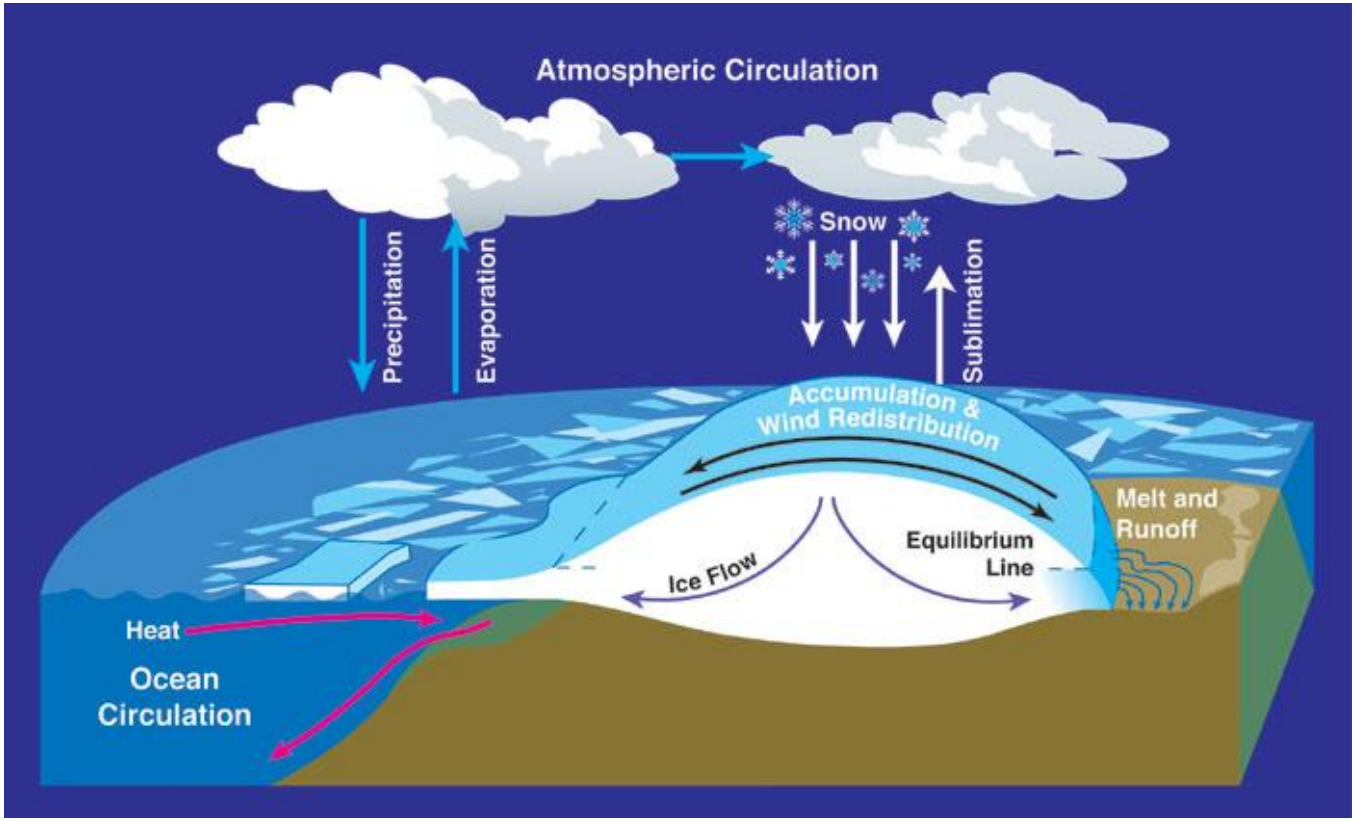
Outline

1. **Background**
   - Motivation for climate & land-ice modeling
   - **Earth system models & projects**
   - Land-ice equations
2. Algorithms and software
   - Albany/FELIX steady stress-velocity solver
     - Meshes/data
     - Solvers & preconditioners
     - Performance-portability
     - Ice sheet initialization/UQ
   - CISM/MPAS-Albany for dynamic simulations
     - Velocity-thickness coupling
     - Validation
     - Towards science runs
3. Ongoing & future work
4. Summary



Earth System Models (ESMs)

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

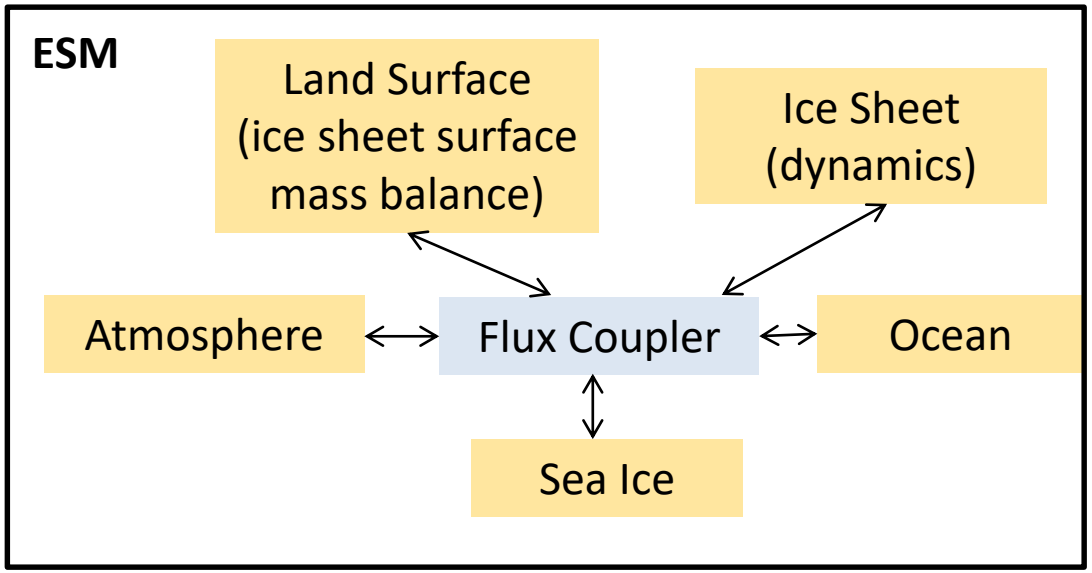


Mass balance: change in ice sheet mass = mass in – mass out

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

Earth System Models (ESMs)

- 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



Climate Model passes:

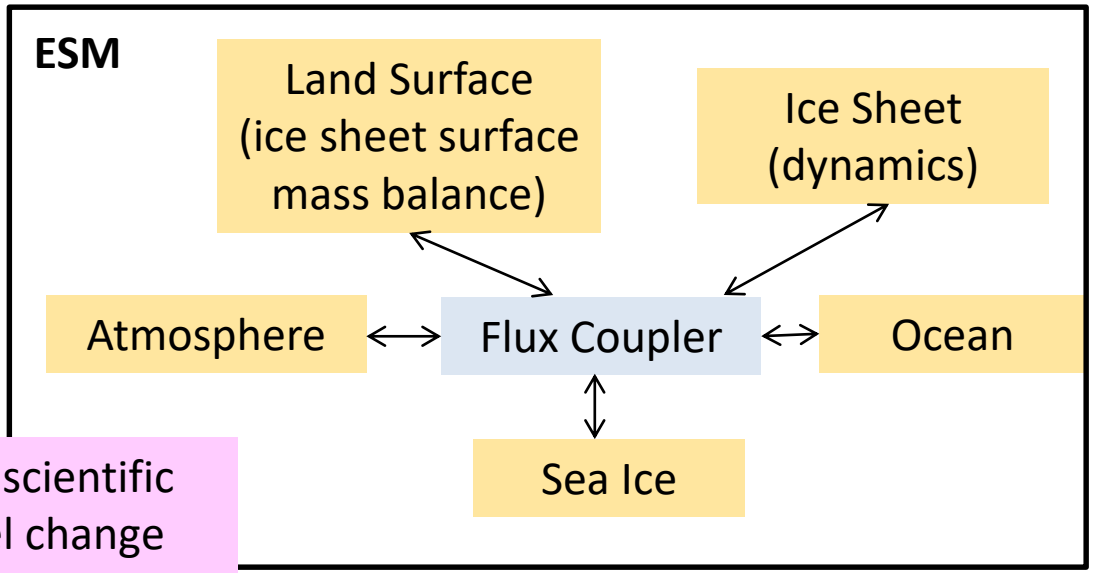
- Surface mass balance (SMB)
- Boundary temperatures
- Sub-shelf melting

Land Ice Model passes:

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

Earth System Models (ESMs)

- 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 change (including uncertainty bounds).

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

DOE ESM: Energy Exascale Earth System Model (E3SM)*

- ***E3SM model*** was created 2014 to address perceived need to distinguish DOE efforts from other U.S. climate modeling efforts.

- Multi-lab effort to develop/apply ESM for ***DOE needs.***
  - Focus on ***decadal to century timescale projections***: 1970-2010 hindcast, 2010-2050 projection
  - Focus on impacts to ***DOE, DOD and other U.S. infrastructure***
  - Focus on ***high-spatial resolution*** and next ***generation HPC***

- ***Science focus areas***:
  1. Water Cycle: how do hydrological cycle and water resources interact with the climate on local and global scales?
  2. Biogeochemistry: how do biogeochemical cycles interact with global climate change?
  3. Cryosphere-Ocean System: how do rapid changes in cryosphere-ocean systems interact with climate system?

* Formerly *Accelerated Climate Model for Energy* (ACME).

DOE ESM: Energy Exascale Earth System Model (E3SM)*

- ***E3SM model*** was created 2014 to address perceived need to distinguish DOE efforts from other U.S. climate modeling efforts.

- Multi-lab effort to develop/apply ESM for ***DOE needs.***
  - Focus on ***decadal to century timescale projections***: 1970-2010 hindcast, 2010-2050 projection
  - Focus on impacts to ***DOE, DOD and other U.S. infrastructure***
  - Focus on ***high-spatial resolution*** and next ***generation HPC***

- ***Science focus areas***:
  1. Water Cycle: how do hydrological cycle and water resources interact with the climate on local and global scales?
  2. Biogeochemistry: how do biogeochemical cycles interact with global climate change?
  3. **Cryosphere-Ocean System: how do rapid changes in cryosphere-ocean systems interact with climate system?**

* Formerly *Accelerated Climate Model for Energy* (ACME).

DOE Ice Sheet Modeling Programs/Projects

DOE programs funding ice sheet, sea level, & climate modeling:

- *Biological and Environmental Research* (BER).
- *Advanced Scientific Computing Research* (ASCR).
- *Scientific Discovery through Advanced Computing* (SciDAC; BER + ASCR)

Projects:

- *Energy Exascale Earth System Model* (E3SM): current
- *Predicting Ice Sheet & Climate Evolution at Extreme Scales* (PISCEES): 2012-2017
- *Probabilistic Sea-Level Projections from Ice Sheet and Earth System Models* (ProSPect): 2017-2022

DOE Ice Sheet Modeling Programs/Projects

DOE programs funding ice sheet, sea level, & climate modeling:

- *Biological and Environmental Research* (BER).
- *Advanced Scientific Computing Research* (ASCR).
- *Scientific Discovery through Advanced Computing* (SciDAC; BER + ASCR)

Bulk of this talk is on our work for the **PISCEES project**.

Projects:

- *Energy Exascale Earth System Model* (E3SM): current
- ***Predicting Ice Sheet & Climate Evolution at Extreme Scales*** (PISCEES): 2012-2017
- *Probabilistic Sea-Level Projections from Ice Sheet and Earth System Models* (ProSPect): 2017-2022

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.
- **Situation in 2012 motivating PISCEES**: "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.
- **Situation in 2012 motivating PISCEES**: "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 (Predicting Ice Sheet Climate & Evolution at Extreme Scales) 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***.

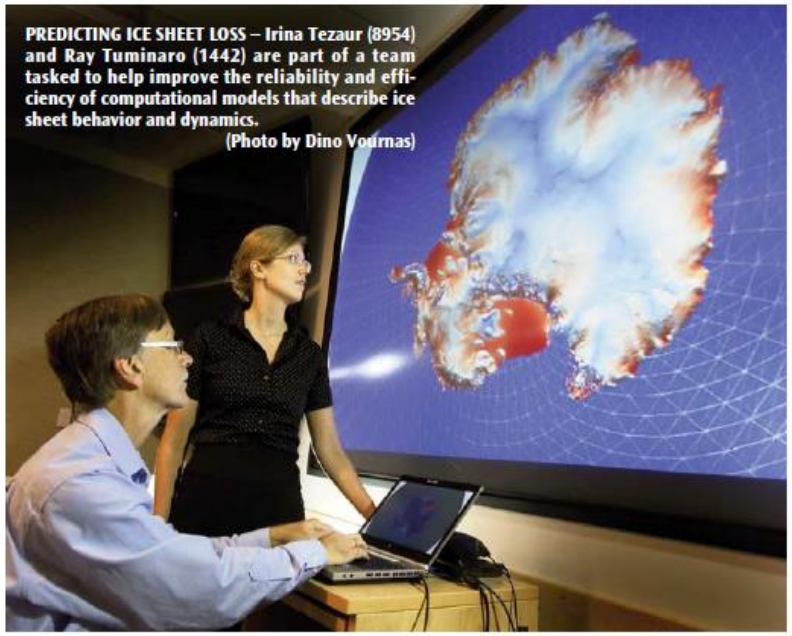
* Solomon, S., Qin, D., Manning, M., Chen, Z., Marquis, M., Averyt, K., Tignor, M., and Miller, H. "Climate change 2007: The physical science basis, Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change", Cambridge Univ. Press, Cambridge, UK,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

PISCEES In-The-News

Ice sheet modeling of Greenland, Antarctica helps predict sea-level rise

Michael Padilla

The Greenland and Antarctic ice sheets will make a dominant contribution to 21st century sea-level rise if current climate trends continue. However, predicting the expected loss of ice sheet mass is difficult due to the complexity of modeling ice sheet behavior.



PREDICTING ICE SHEET LOSS – Irina Tezaur (8954) and Ray Tuminaro (1442) are part of a team tasked to help improve the reliability and efficiency of computational models that describe ice sheet behavior and dynamics.
(Photo by Dino Vournas)

Computing (SciDAC) program. PISCEES is a multi-lab, multi-university endeavor that includes researchers from Sandia, Los Alamos, Lawrence Berkeley, and Oak Ridge national laboratories; the Massachusetts Institute of Technology; Florida State University; the University of Bristol; the University of Texas Austin; the University of South Carolina; and New York University.

Sandia's biggest contribution to PISCEES has been an analysis tool: a land-ice solver called Albany/FELIX (Finite Elements for Land Ice eXperiments). The tool is based on equations that simulate ice flow over the Greenland and Antarctic ice sheets and is being coupled to Earth models through the Accelerated Climate for Energy (ACME) project.

"One of the goals of PISCEES is to create a land-ice solver that is scalable, fast, and robust on continental scales," says computational scientist Irina Tezaur, a lead developer of Albany/FELIX. Not only did the new solver need to be reliable and efficient, but it was critical that the team develop a solver capable of running on new and emerging computers, and equipped with advanced

Clean Technica

Forecasting, Not Fearing, Sea-Level Rise

August 28th, 2016 by Robyn Purchia

This week, the *Washington Post* reported a widening 80-mile crack threatening one of Antarctica's biggest ice shelves. A large chunk of Larsen C, the most northern major ice shelf, may break off in the coming years.

Of course, the probable loss of Larsen C is a terrifying reminder that climate change is real and happening now. But what consequence will it have on Antarctic glaciers and sea-level rise? Researchers know ice shelves have a buttressing effect on interior ice because they restrain the flow of glaciers from the land to the sea. However, researchers can't predict how the glaciers will behave once the shelf is gone.

RAPID DEVELOPMENT OF AN ICE SHEET CLIMATE APPLICATION USING THE COMPONENTS-BASED APPROACH

ANDREW SALINGER, PI
IRINA TEZAUR
MAURO PEREGO
RAYMOND TUMINARO
Sandia National Labs

STEPHEN PRICE
Los Alamos National Labs

PROCESSING HOURS
1,000,000

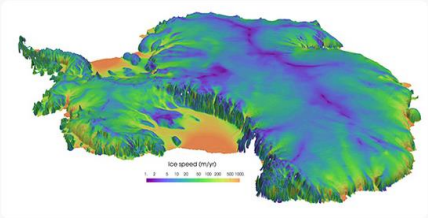
agsalin@sandia.gov

"As computational scientists with expertise in math and algorithms, it is challenging to get deep enough into a new science application area to make an impact. This team has made a sustained effort in learning about ice sheets and building relationships with climate scientists, and has been rewarded seeing our code on the critical path of DOE's climate science program."

– Andy Salinger

Climate

A land-ice simulation code

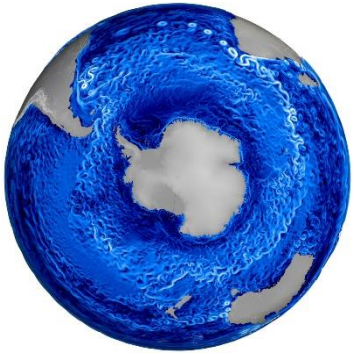


An Albany/FELIX simulation of Antarctica shows surface velocities draped over a surface topography computed on a variable resolution mesh.

As part of the five-year multi-institution DOE/SciDAC [Scientific Discovery Through Advanced Computing] project PISCEES, Sandia has developed a land-ice simulation code that has been integrated into DOE's Accelerated Climate Model for Energy earth system model for use in climate projections. The Albany/FELIX code enables the calculation of initial conditions for land-ice simulations, critical for stable and accurate dynamic simulations of ice sheet evolution and the quantification of uncertainties in 21st century sea level rise. With NASA, the team has successfully validated simulations in comparison to actual Greenland ice sheet measurements. (8900,1400)

Outline

1. **Background**
   - Motivation for climate & land-ice modeling
   - Earth system models & projects
   - **Land-ice equations**
2. Algorithms and software
   - Albany/FELIX steady stress-velocity solver
     - Meshes/data
     - Solvers & preconditioners
     - Performance-portability
     - Ice sheet initialization/UQ
   - CISM/MPAS-Albany for dynamic simulations
     - Velocity-thickness coupling
     - Validation
     - Towards science runs
3. Ongoing & future work
4. Summary



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>

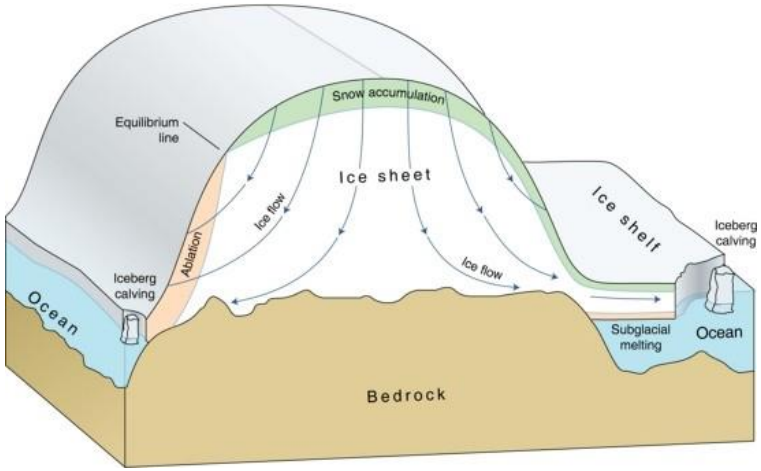
$$\begin{cases} -\nabla \cdot \boldsymbol{\sigma} = \rho \boldsymbol{g} \\ -\nabla \cdot u = 0 \end{cases}, \quad \text{in } \Omega$$

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.

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

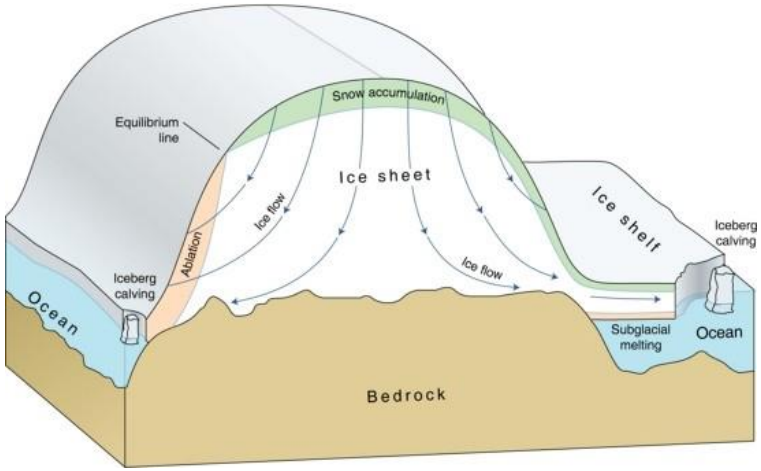
$$\begin{cases} -\nabla \cdot \boldsymbol{\sigma} = \rho \boldsymbol{g} \\ -\nabla \cdot u = 0 \end{cases}, \quad \text{in } \Omega$$

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

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



"Nasty" saddle point problem!

The First-Order (FO) Stokes Model

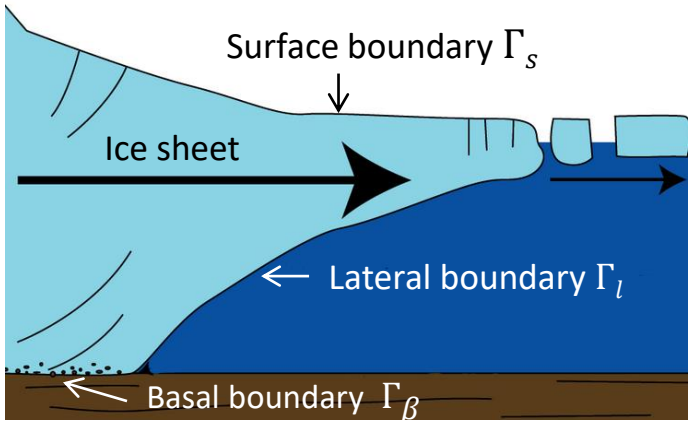
- In our model, ice sheet dynamics are given by ***"First-Order" Stokes PDEs***: "nice" elliptic approximation* to Stokes' flow equations.

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

$$\dot{\boldsymbol{\epsilon}}_1{}^T = (2\dot{\epsilon}_{11} + \dot{\epsilon}_{22}, \dot{\epsilon}_{12}, \dot{\epsilon}_{13})$$
$$\dot{\boldsymbol{\epsilon}}_2{}^T = (2\dot{\epsilon}_{12}, \dot{\epsilon}_{11} + 2\dot{\epsilon}_{22}, \dot{\epsilon}_{23})$$
$$\dot{\epsilon}_{\mathrm{ij}} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \; i,j = 1,2$$
$$\dot{\epsilon}_{\mathrm{i3}} = \frac{1}{2}\frac{\partial u_i}{\partial x_3}, \qquad i,j = 1,2$$

- 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.

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
  - $\beta = \beta(x, y) =$ basal friction coefficient (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).

Importance of Boundary Conditions!

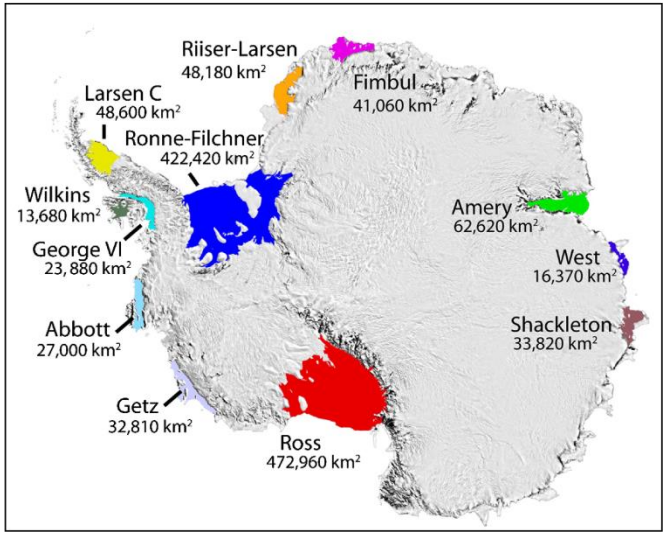
Boundary conditions have tremendous effect on ice sheet dynamics!

- ***Basal sliding BC:*** $2\mu\dot{\boldsymbol{\epsilon}}_i \cdot \boldsymbol{n} + \beta u_i = 0$, on Γ_β
  - $\beta = \beta(x, y) =$ basal friction coefficient (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:***

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

  - Floating ice = "ice shelves" $\Rightarrow$ Antarctica
  - Ice shelves buttress interior ice $\Rightarrow$ 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].*"

Thickness & Temperature Equations

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

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

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

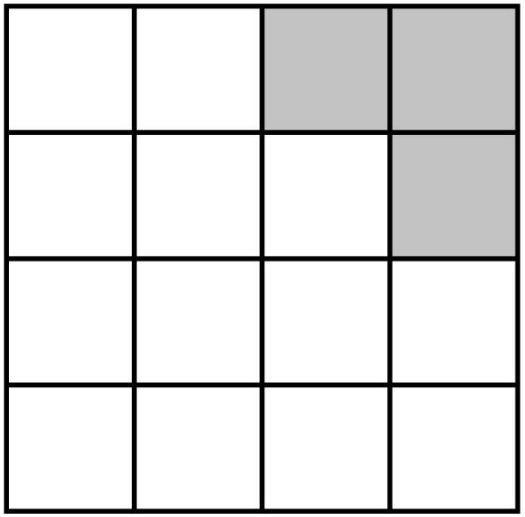
- ***Temperature equation*** (advection-diffusion):

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

(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 ($H > H_{min}$)

Thickness & Temperature Equations

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

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

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

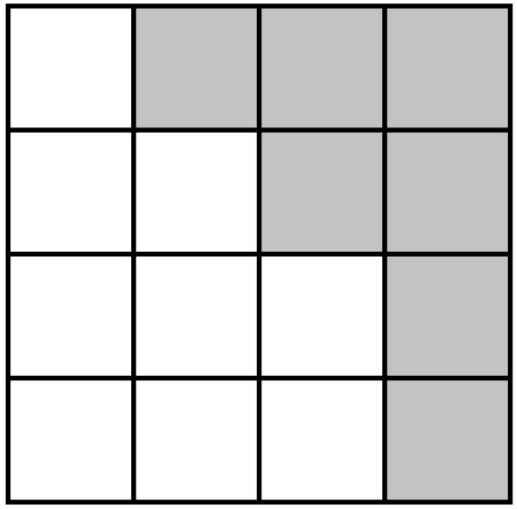
- ***Temperature equation*** (advection-diffusion):

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

(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):

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

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

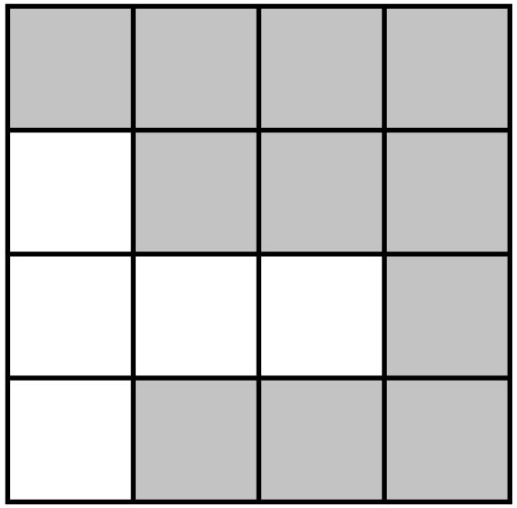
- ***Temperature equation*** (advection-diffusion):

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

(energy balance).

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

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

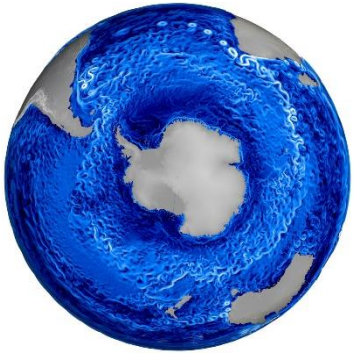


time t_2

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

Outline

1. Background
   - Motivation for climate & land-ice modeling
   - Earth system models & projects
   - Land-ice equations
2. **Algorithms and software**
   - Albany/FELIX steady stress-velocity solver
     - Meshes/data
     - Solvers & preconditioners
     - Performance-portability
     - Ice sheet initialization/UQ
   - CISM/MPAS-Albany for dynamic simulations
     - Velocity-thickness coupling
     - Validation
     - Towards science runs
3. Ongoing & future work
4. Summary



Algorithmic Choices

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

Algorithmic Choices

<u>Sandia's Role in the PISCEES (& ProSPect) 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 (& ProSPect) 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 (& ProSPect) 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).

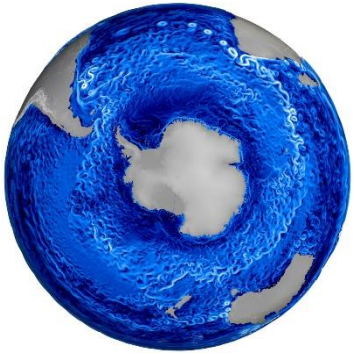
Our algorithmic choices aligned with our objectives!

<u>Algorithms:</u>

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

Outline

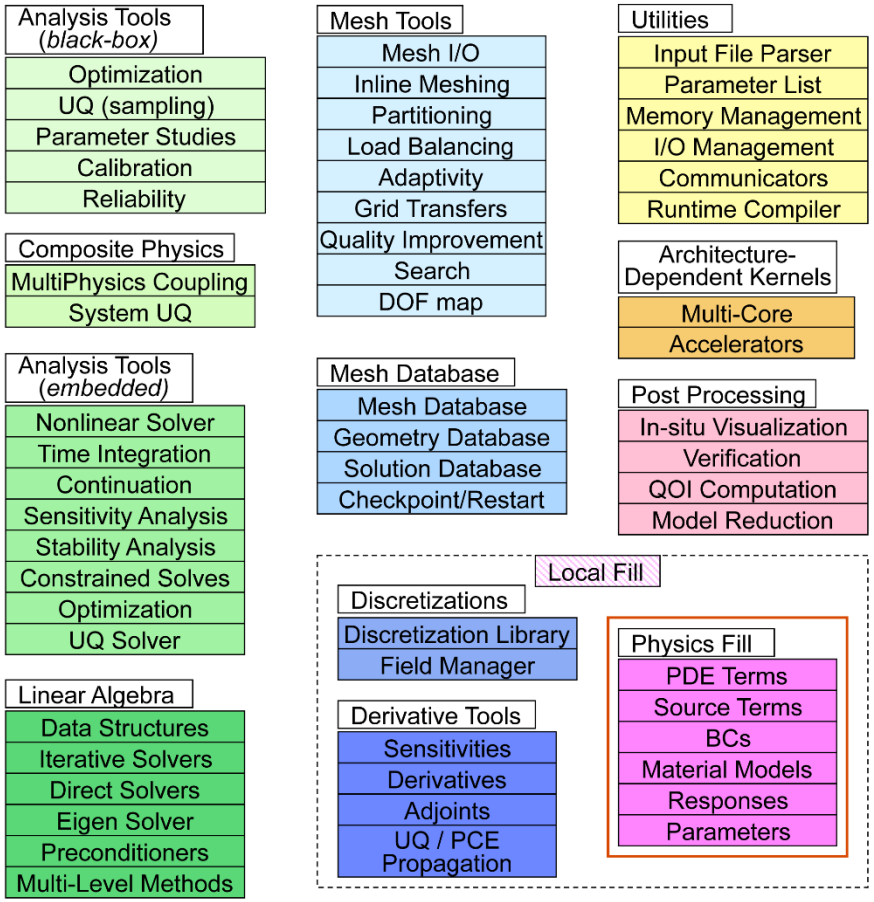
1. Background
   - Motivation for climate & land-ice modeling
   - Earth system models & projects
   - Land-ice equations
2. **Algorithms and software**
   - **Albany/FELIX steady stress-velocity solver**
     - Meshes/data
     - Solvers & preconditioners
     - Performance-portability
     - Ice sheet initialization/UQ
   - CISM/MPAS-Albany for dynamic simulations
     - Velocity-thickness coupling
     - Validation
     - Towards science runs
3. Ongoing & future work
4. Summary



Albany*/FELIX** FEM FO Stokes Solver

Albany/FELIX land-ice solver is implemented in ***Albany***: Sandia open-source parallel, C++, multi-physics finite element code.

- ***Component-based*** design for rapid development of new physics & capabilities.
- Extensive use of libraries from the open-source ***Trilinos*** project:
  - Automatic differentiation.
  - Discretizations/meshes, mesh adaptivity.
  - Solvers, time-integration schemes.
  - Performance-portable kernels.
- ***Advanced analysis*** capabilities:
  - Parameter estimation.
  - Uncertainty quantification (DAKOTA).
  - Optimization.
  - Sensitivity analysis.

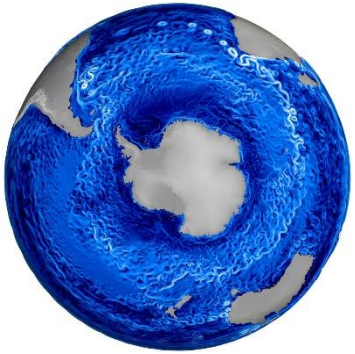


40+ packages; 120+ libraries

* https://github.com/gahansen/Albany. ** FELIX = Finite Elements for Land Ice eXperiments.

Outline

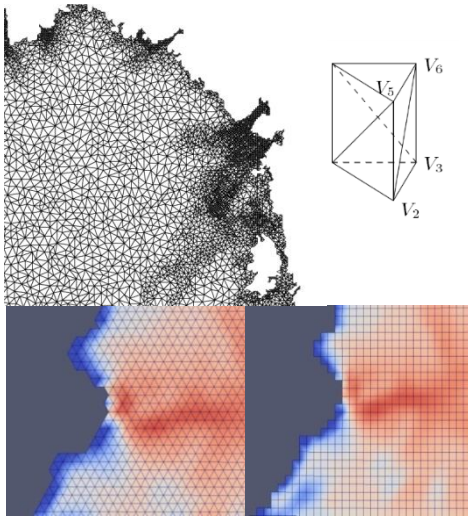
1. Background
   - Motivation for climate & land-ice modeling
   - Earth system models & projects
   - Land-ice equations
2. **Algorithms and software**
   - **Albany/FELIX steady stress-velocity solver**
     - **Meshes/data**
     - Solvers & preconditioners
     - Performance-portability
     - Ice sheet initialization/UQ
   - CISM/MPAS-Albany for dynamic simulations
     - Velocity-thickness coupling
     - Validation
     - Towards science runs
3. Ongoing & future work
4. Summary



Meshes and Data

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

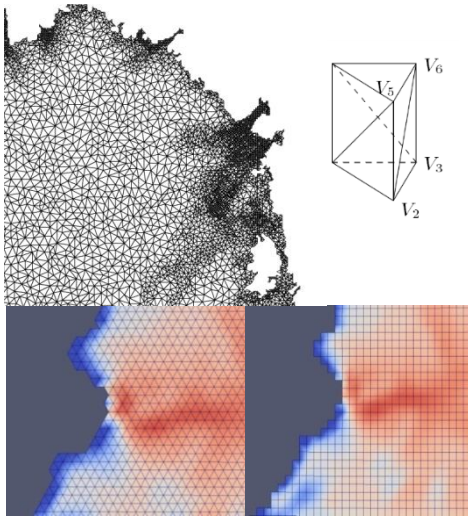
- ***Structured hexahedral*** meshes (compatible with *CISM*).
- ***Tetrahedral*** meshes (compatible with *MPAS LI*)
    - ***Unstructured Delaunay triangle*** meshes with regional refinement based on gradient of surface velocity.
- All meshes are extruded (structured) in vertical direction as tetrahedra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



Meshes and Data

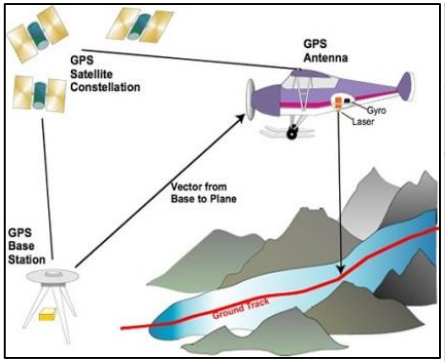
<u>Meshes:</u> can use any mesh but interested specifically in

- ***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.



<u>Data:</u> 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.



Mesh Adaptivity

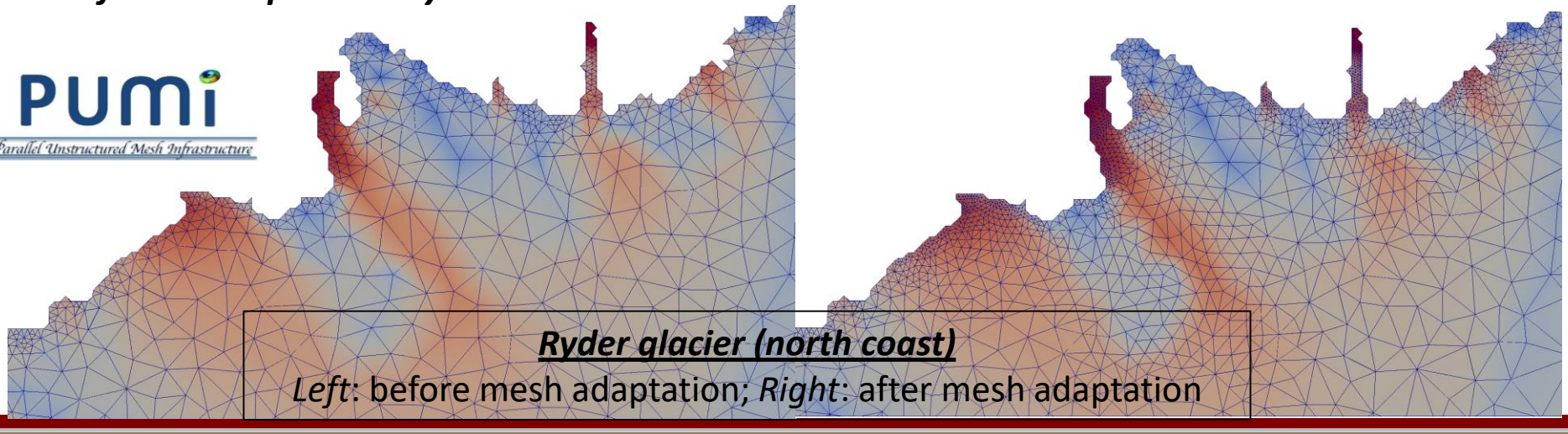
PAALS = Parallel Albany Adaptive Loop with SCOREC*

- In collaboration with ***Rensselaer Polytechnical Institute*** (M. Shephard, C. Smith, B. Granzow): added mesh adaptation capabilities (PAALS) to *Albany.*

*****SCOREC*** = Scientific Computation Research Center at RPI: https://github.com/SCOREC

PAALS provides:

- Fully-coupled, ***in-memory adaptation*** and solution transfer services.
- ***Parallel mesh infrastructure*** and services via ***PUMI*** (Parallel Unstructured Mesh Infrastructure): an efficient, distributed mesh data structure that supports adaptivity.
- Predictive ***dynamic load balancing*** via ***ParMetis/Zoltan*** + ***ParMA.***
- SPR**-based generalized ***error estimation*** of ***velocity gradient*** drives adaptation.
- ***Performance portability*** to GPUs via ***Kokkos***.



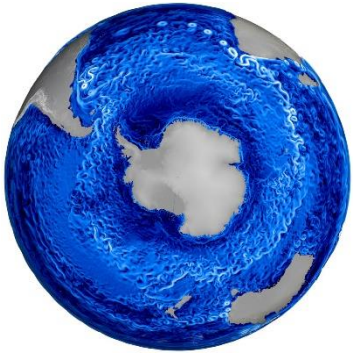
Ryder glacier (north coast)

Left: before mesh adaptation; *Right*: after mesh adaptation

**Super-convergent Patch Recovery: technique for estimating $\nabla \boldsymbol{u}$ using quadratic approximation within a patch of elements.

Outline

1. Background
   - Motivation for climate & land-ice modeling
   - Earth system models & projects
   - Land-ice equations
2. **Algorithms and software**
   - **Albany/FELIX steady stress-velocity solver**
     - Meshes/data
     - **Solvers & preconditioners**
     - Performance-portability
     - Ice sheet initialization/UQ
   - CISM/MPAS-Albany for dynamic simulations
     - Velocity-thickness coupling
     - Validation
     - Towards science runs
3. Ongoing & future work
4. Summary



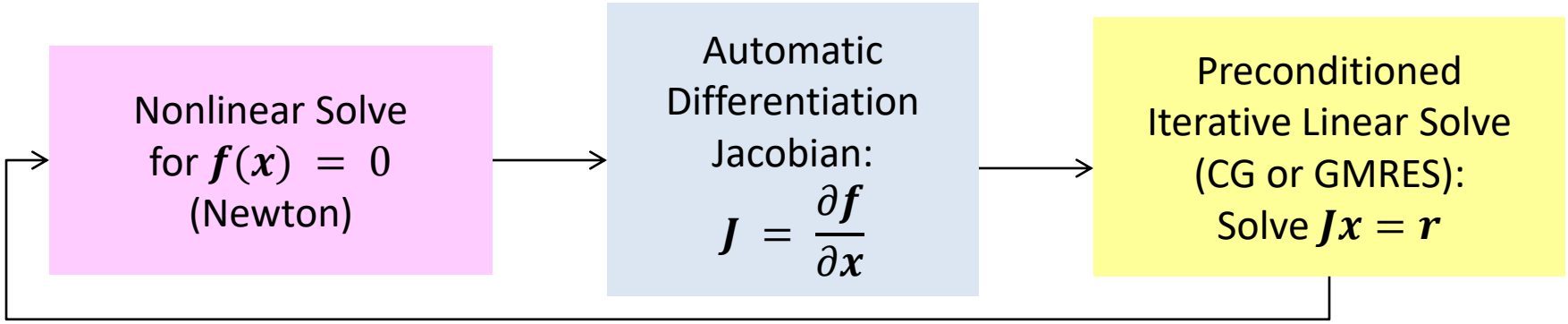
Nonlinear & Linear Solvers

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

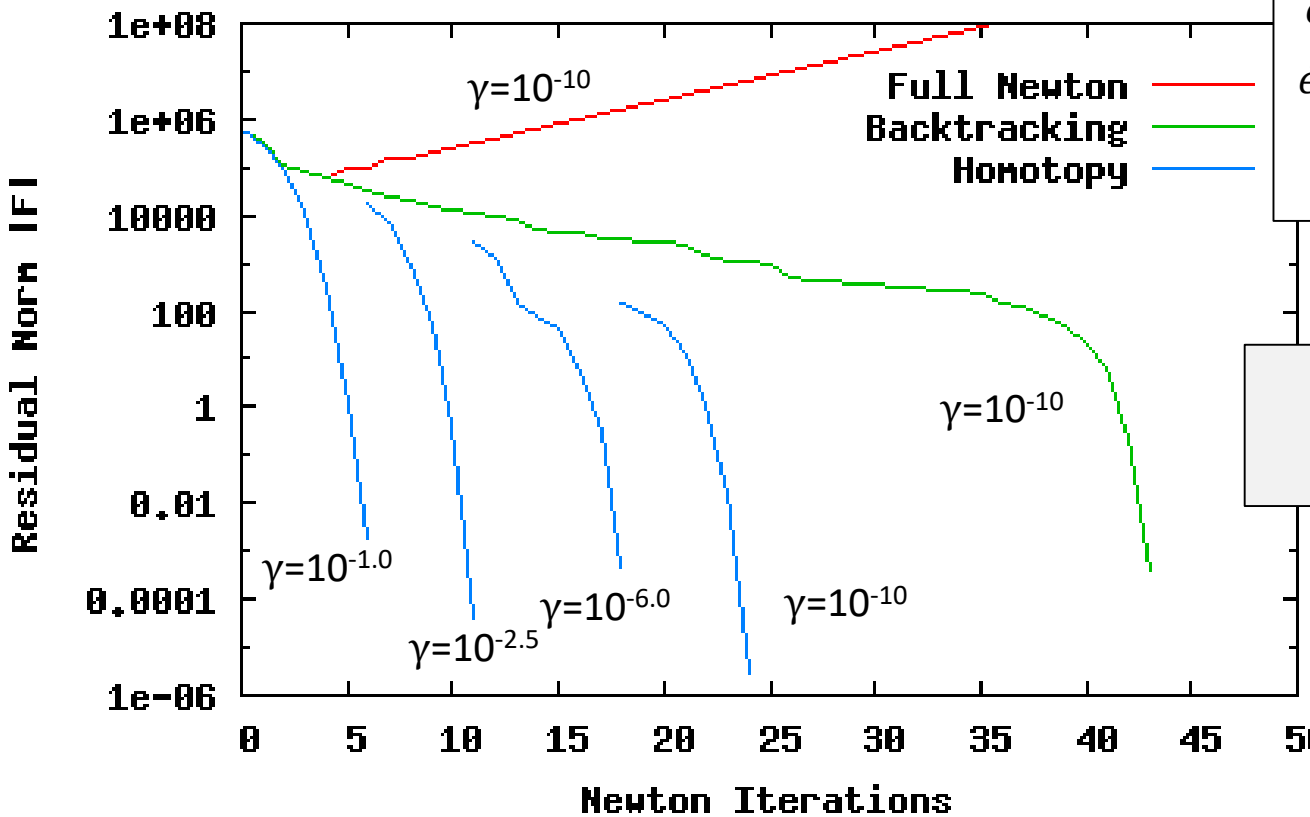
- Most robust and efficient for steady-state solves.
- Jacobian available for preconditioners and matrix-vector products.
- Analytic sensitivity analysis.
- Analytic gradients for inversion.

Linear solver: preconditioned iterative method

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



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



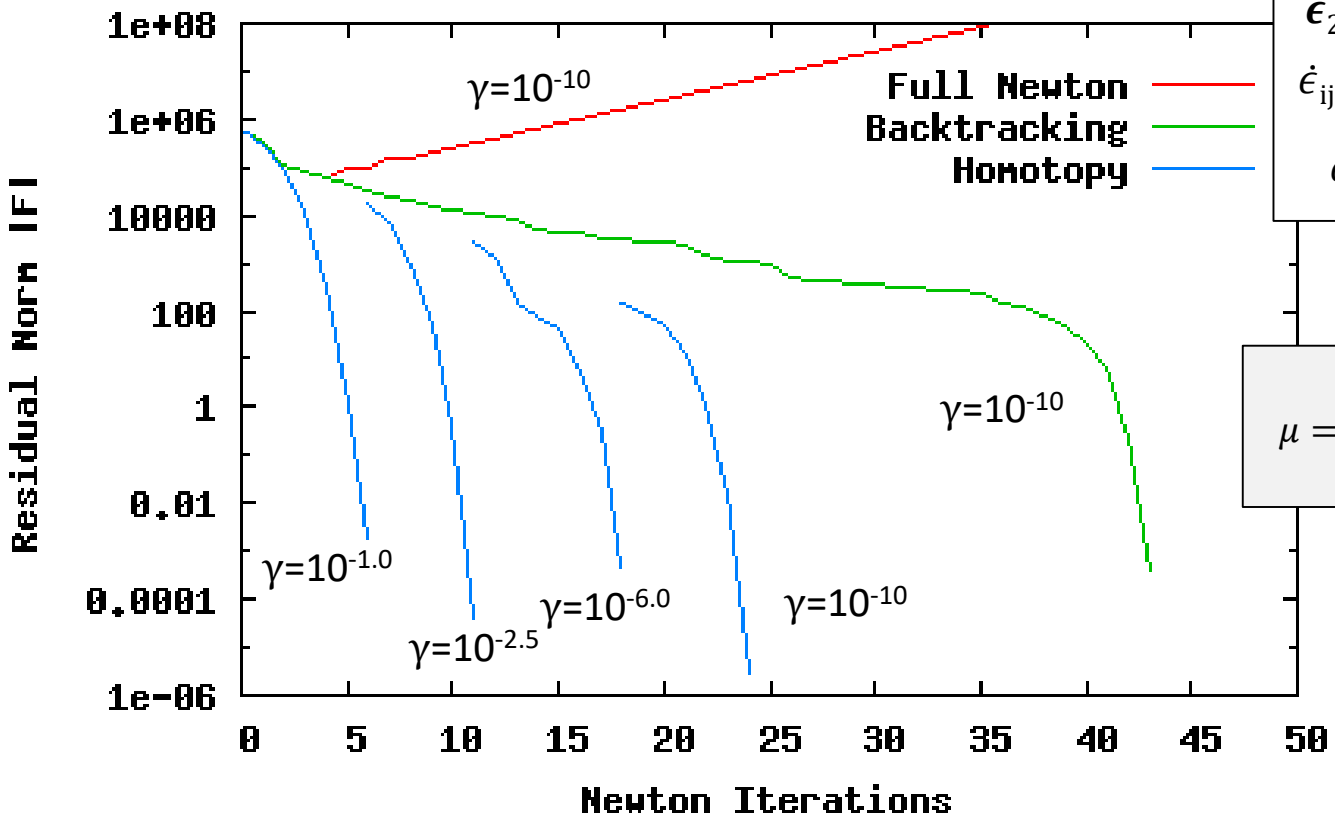
$$\dot{\boldsymbol{\epsilon}}_1{}^T = (2\dot{\epsilon}_{11} + \dot{\epsilon}_{22}, \dot{\epsilon}_{12}, \dot{\epsilon}_{13})$$
$$\dot{\boldsymbol{\epsilon}}_2{}^T = (2\dot{\epsilon}_{12}, \dot{\epsilon}_{11} + 2\dot{\epsilon}_{22}, \dot{\epsilon}_{23})$$
$$\dot{\epsilon}_{\mathrm{ij}} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \; i,j = 1,2$$
$$\dot{\epsilon}_{\mathrm{i3}} = \frac{1}{2}\frac{\partial u_i}{\partial x_3}, \qquad i,j = 1,2$$

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)



$$\dot{\boldsymbol{\epsilon}}_1{}^T = (2\dot{\epsilon}_{11} + \dot{\epsilon}_{22}, \dot{\epsilon}_{12}, \dot{\epsilon}_{13})$$
$$\dot{\boldsymbol{\epsilon}}_2{}^T = (2\dot{\epsilon}_{12}, \dot{\epsilon}_{11} + 2\dot{\epsilon}_{22}, \dot{\epsilon}_{23})$$
$$\dot{\epsilon}_{\mathrm{ij}} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \; i,j = 1,2$$
$$\dot{\epsilon}_{\mathrm{i3}} = \frac{1}{2}\frac{\partial u_i}{\partial x_3}, \qquad i,j = 1,2$$

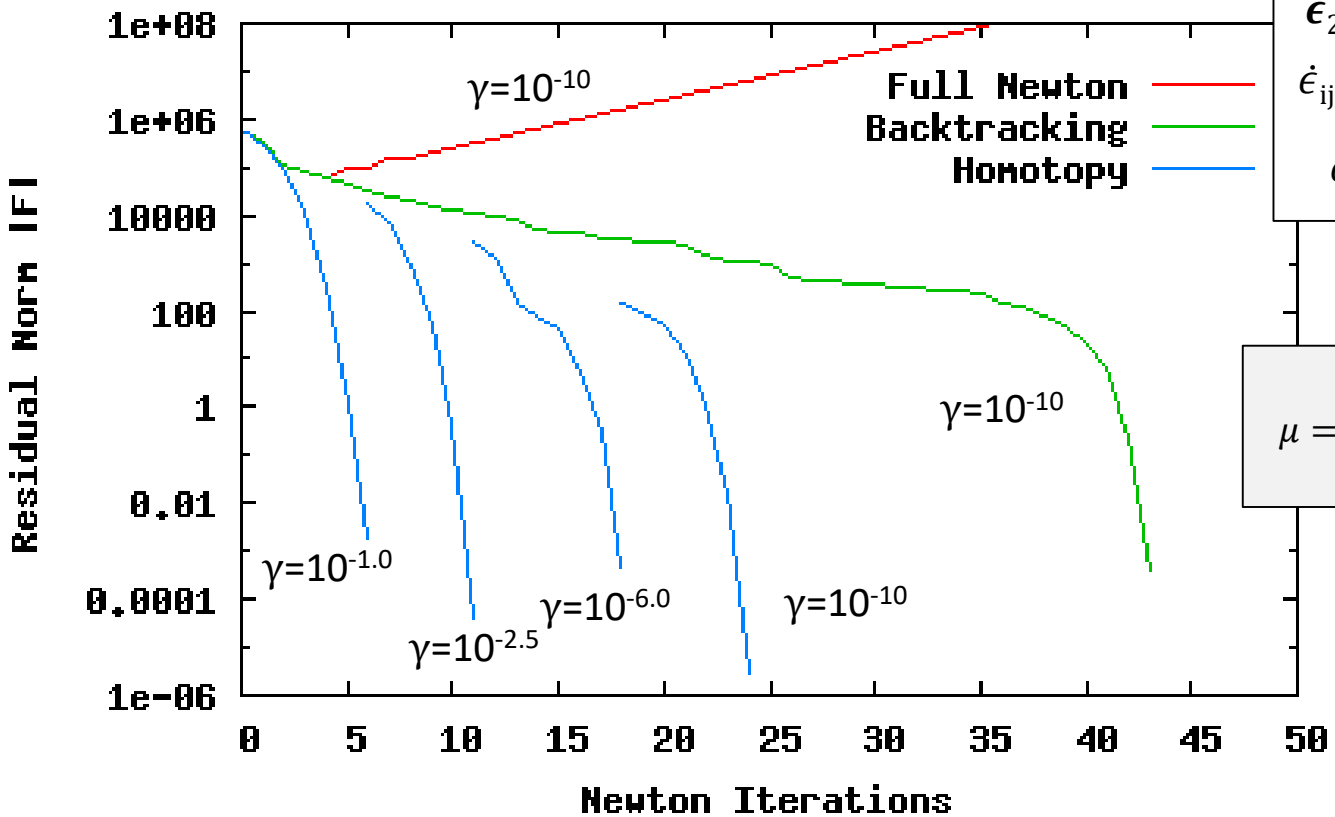
Glen's Law Viscosity:

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

γ = regularization parameter

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

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



$$\dot{\boldsymbol{\epsilon}}_1{}^T = (2\dot{\epsilon}_{11} + \dot{\epsilon}_{22}, \dot{\epsilon}_{12}, \dot{\epsilon}_{13})$$
$$\dot{\boldsymbol{\epsilon}}_2{}^T = (2\dot{\epsilon}_{12}, \dot{\epsilon}_{11} + 2\dot{\epsilon}_{22}, \dot{\epsilon}_{23})$$
$$\dot{\epsilon}_{\mathrm{ij}} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \; i,j = 1,2$$
$$\dot{\epsilon}_{\mathrm{i3}} = \frac{1}{2}\frac{\partial u_i}{\partial x_3}, \qquad i,j = 1,2$$

Glen's Law Viscosity:

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

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: $\alpha_k = 1$

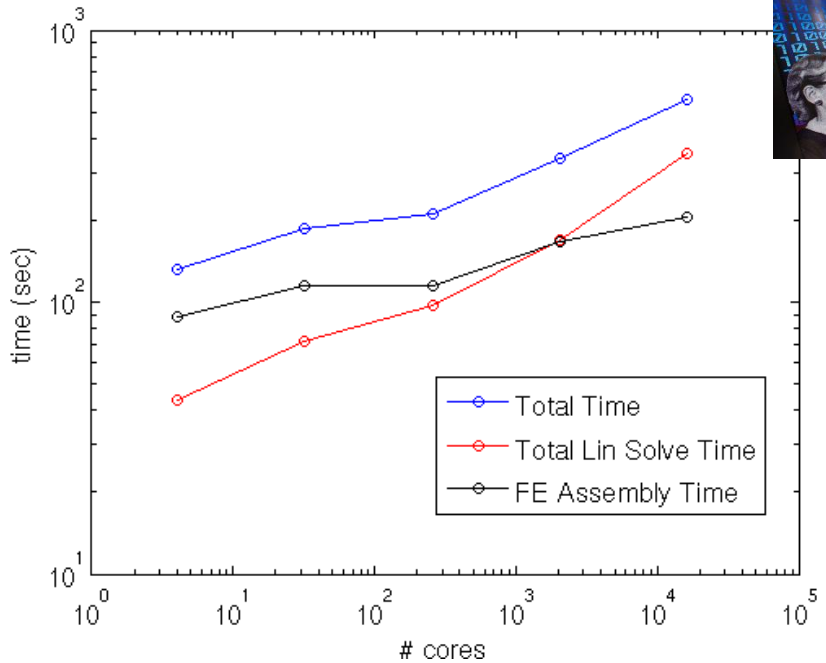
→ Backtracking: line-search for α_k

Iterative Linear Solvers & Preconditioning

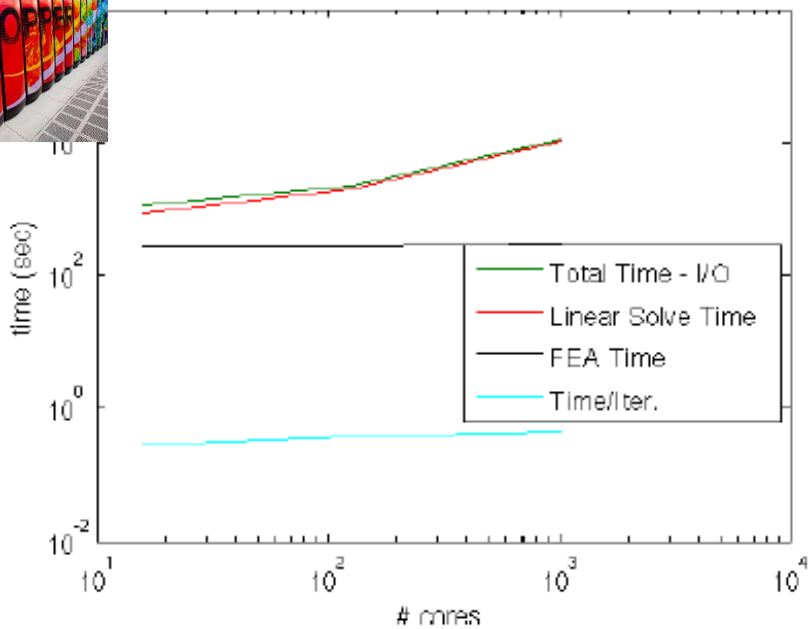
- In practice, large sparse linear systems $\boldsymbol{Ax} = \boldsymbol{b}$ are solved using **iterative methods**.
  - GMRES (Generalized Minimal RESidual).
  - CG (Conjugate Gradient) – for symmetric positive definite $\boldsymbol{A}$.
- Convergence of iterative methods for solving $\boldsymbol{Ax} = \boldsymbol{b}$ 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: $\boldsymbol{P} = \boldsymbol{A}^{-1}$.
- **<u>Common preconditioners:</u>**
  - ***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)***.

Initial Weak Scalability Study Using ILU

Greenland Ice Sheet

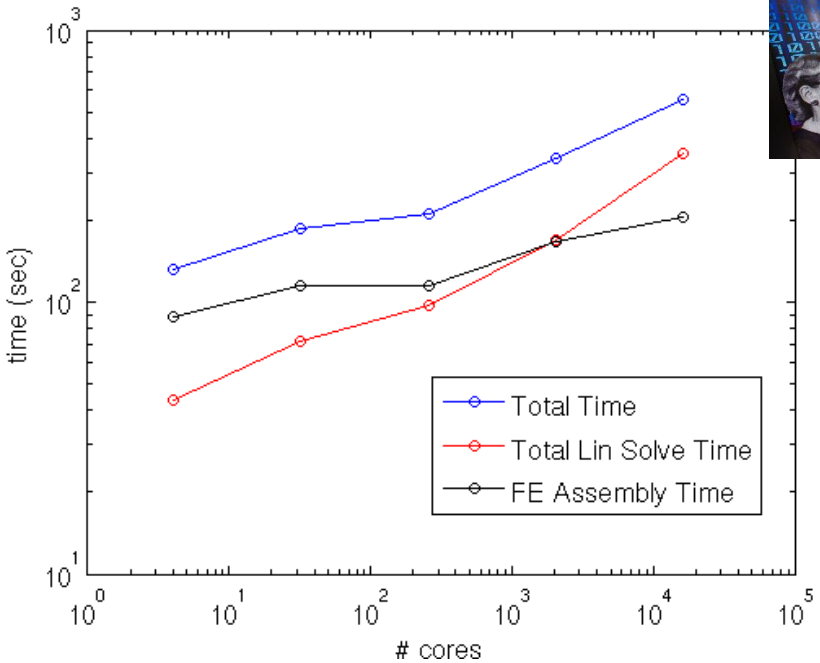


Antarctic Ice Sheet

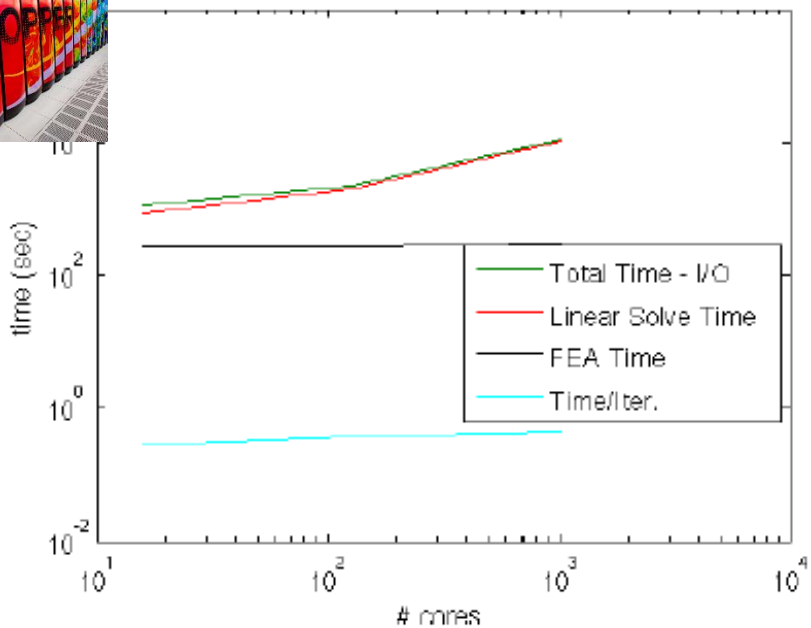


Initial Weak Scalability Study Using ILU

Greenland Ice Sheet



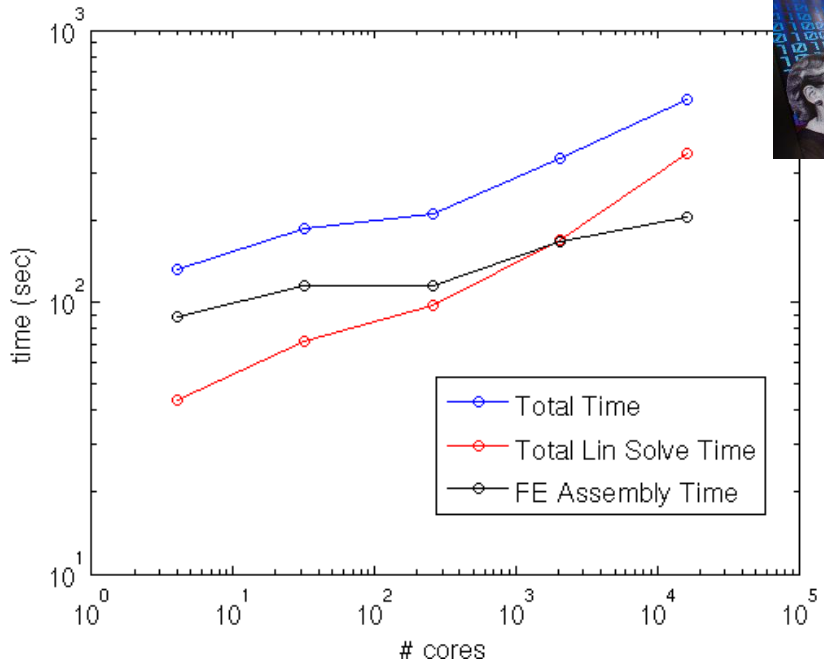
Antarctic Ice Sheet



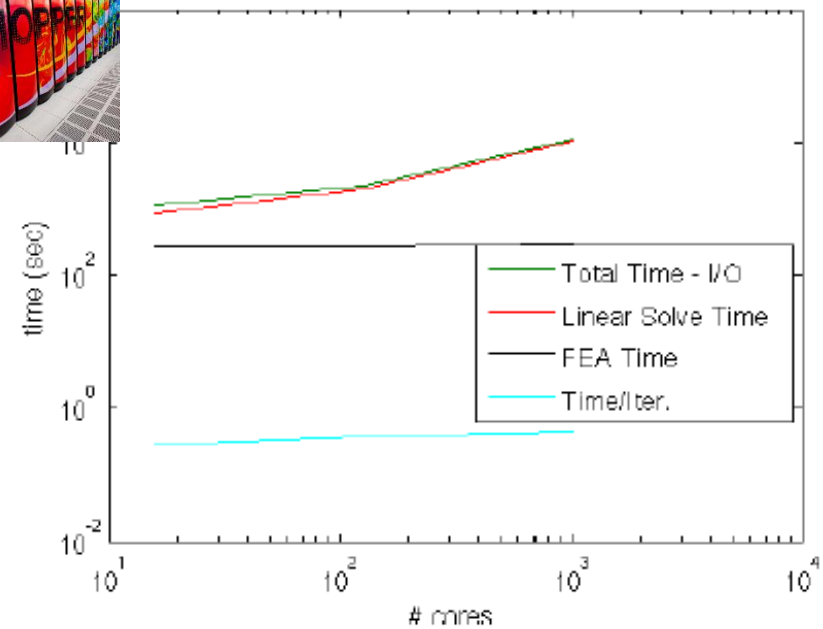
Scalability results are ***not*** acceptable!

Initial Weak Scalability Study Using ILU

Greenland Ice Sheet



Antarctic Ice Sheet



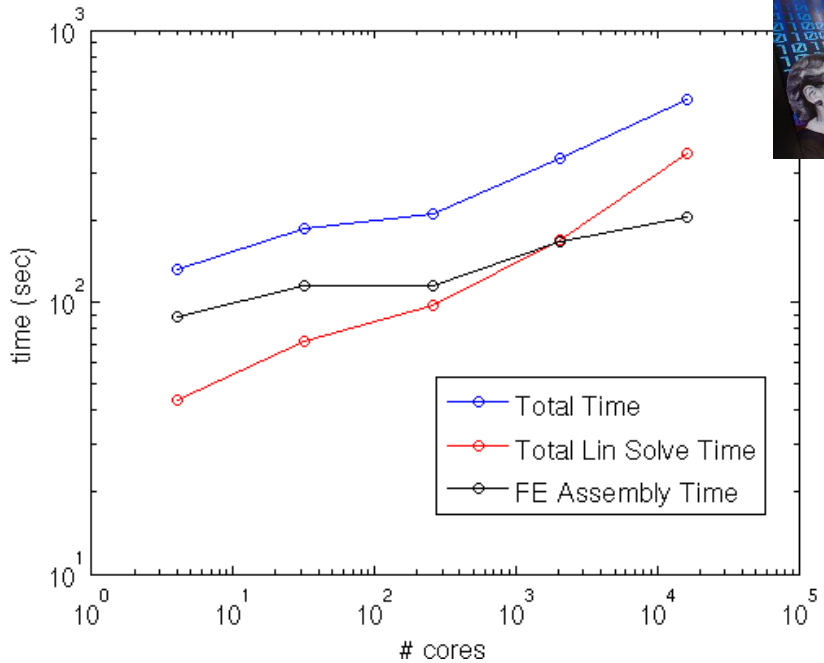
Scalability results are ***not*** acceptable!

*Why is scalability so **bad** for out-of-the-box preconditioners?*

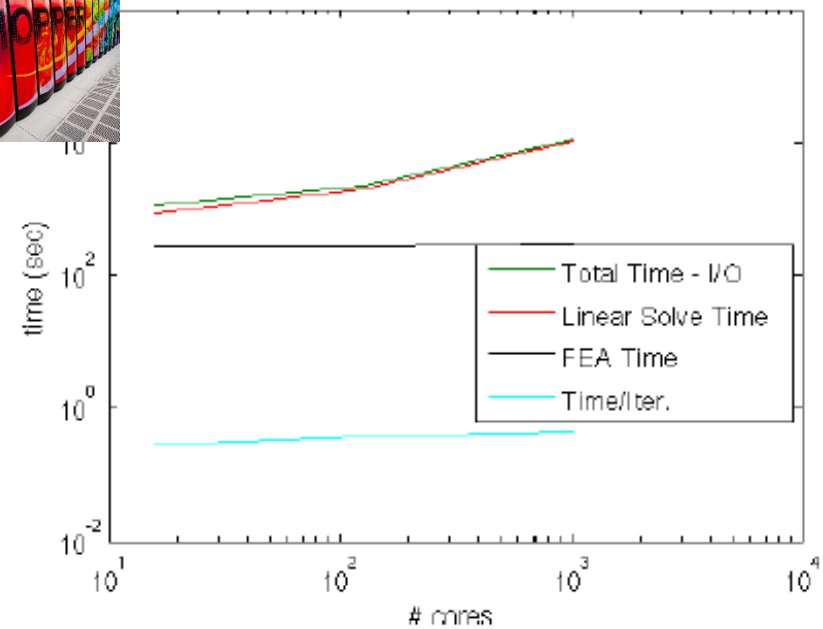
1. Ice sheet geometries have ***bad aspect ratios*** $(dx \gg dz)$.
2. ***Ice shelves*** give rise to ***severely ill-conditioned*** matrices.
3. ***Islands*** and ***hinged peninsulas*** lead to ***solver failures***.

Initial Weak Scalability Study Using ILU

Greenland Ice Sheet



Antarctic Ice Sheet



Scalability results are ***not*** acceptable!

*Why is scalability so **bad** for out-of-the-box preconditioners?*

1. Ice sheet geometries have ***bad aspect ratios*** $(dx \gg dz)$.
2. ***Ice shelves*** give rise to ***severely ill-conditioned*** matrices.
3. ***Islands*** and ***hinged peninsulas*** lead to ***solver failures***.

We **mitigate** these difficulties through the development of:

- New ***AMG* preconditioner*** based on ***semi-coarsening***.
- *Island/hinge removal algorithm*.

* Algebraic Multi-Grid.

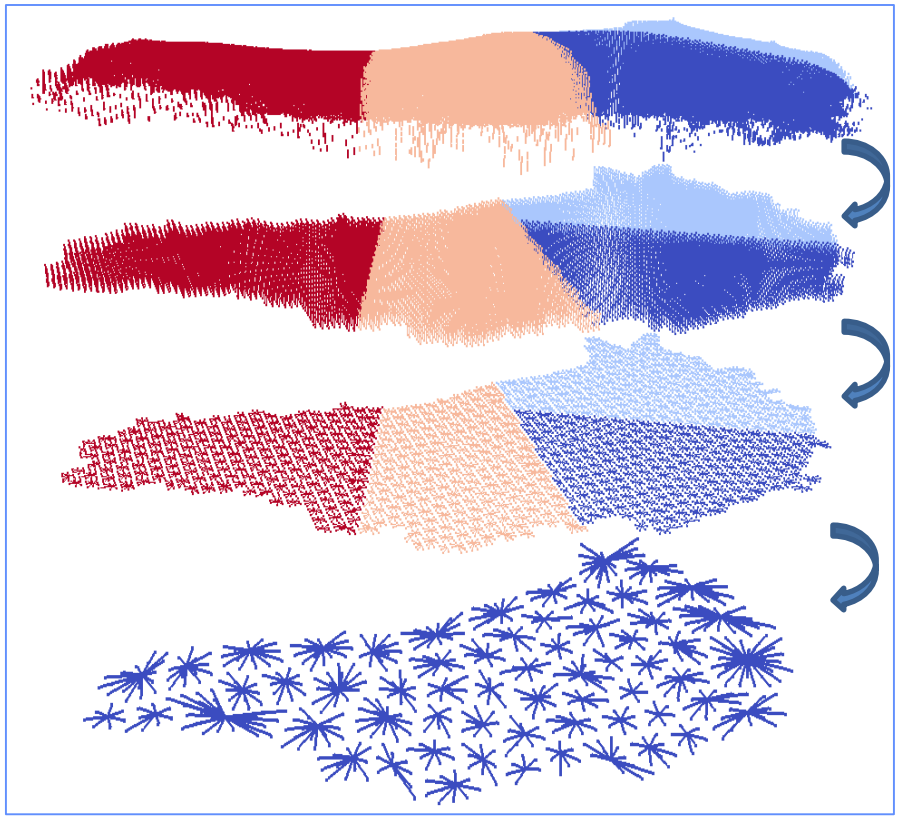
Scalability via Algebraic Multi-Grid Preconditioning with Semi-Coarsening

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

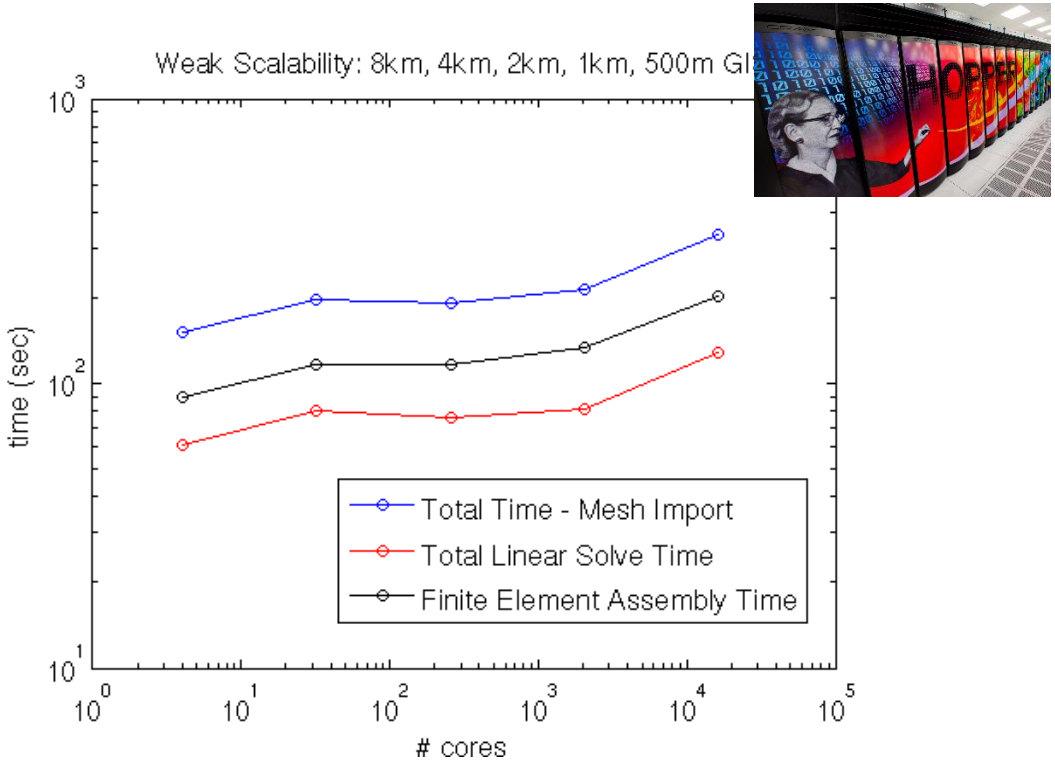
We developed a **new AMG solver** based on aggressive **semi-coarsening** (available in *ML/MueLu* packages of *Trilinos*)



See (Tezaur *et al.*, 2015), (Tuminaro *et al.*, 2016).

Scaling studies (next slides):
New AMG preconditioner vs. ILU

Greenland Controlled Weak Scalability Study



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

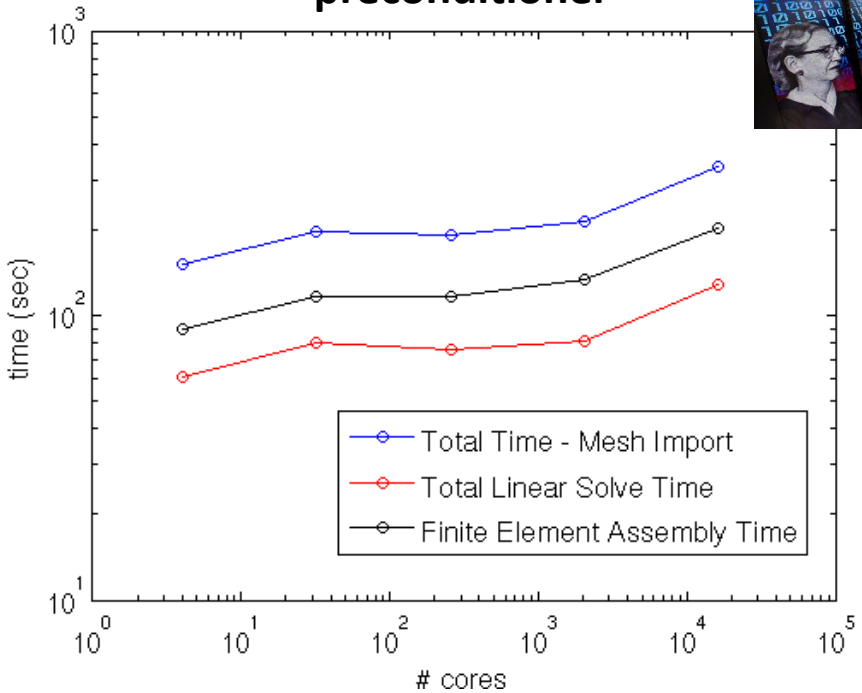
$\times 8^4$
scale up

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

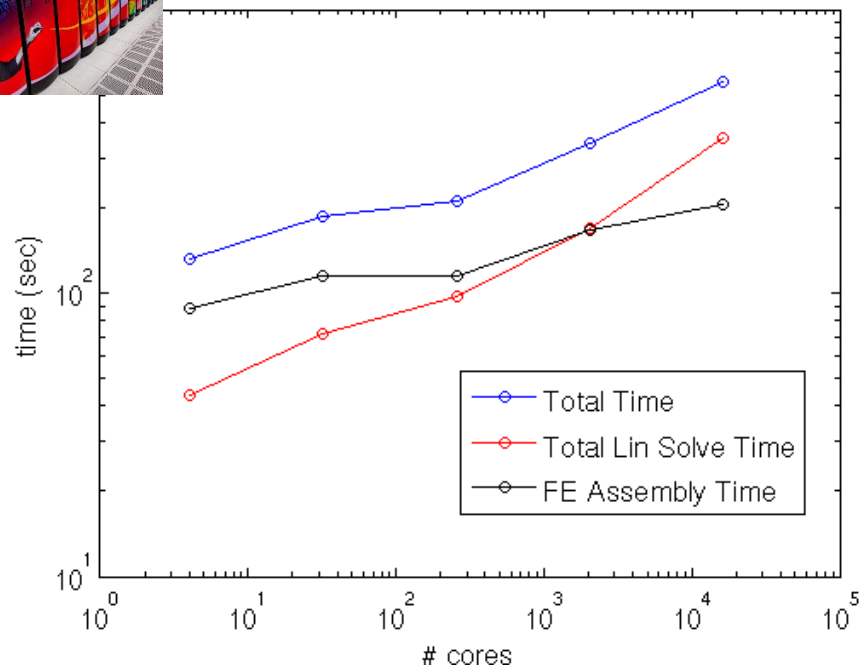
- Weak scaling study with fixed dataset, 4 mesh bisections.
- ~70-80K dofs/core.
- ***Conjugate Gradient (CG) iterative method*** for linear solves (faster convergence than GMRES).
- ***New AMG preconditioner*** developed by R. Tuminaro based on ***semi-coarsening*** (coarsening in z-direction only).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

Greenland Controlled Weak Scalability Study

New AMG preconditioner preconditioner



ILU preconditioner



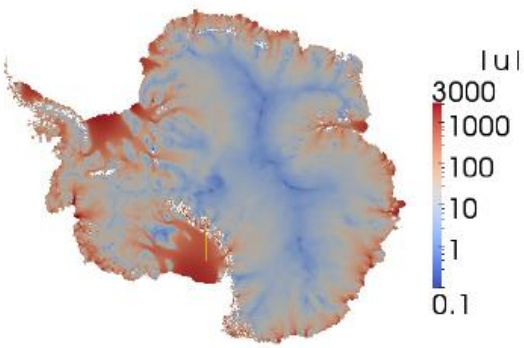
4 cores
334K dofs
8 km Greenland,
5 vertical layers

$\times 8^4$
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!

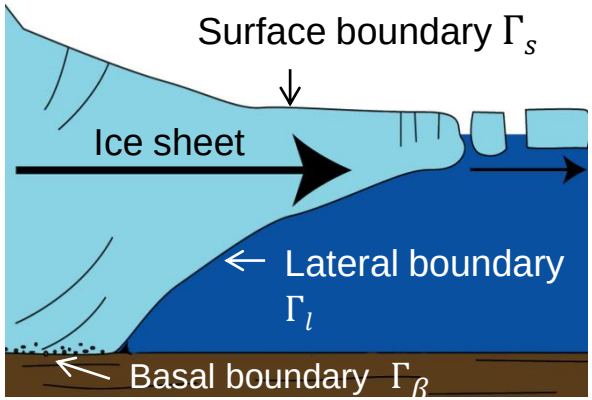
Moderate Resolution Antarctica Weak Scaling Study



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.



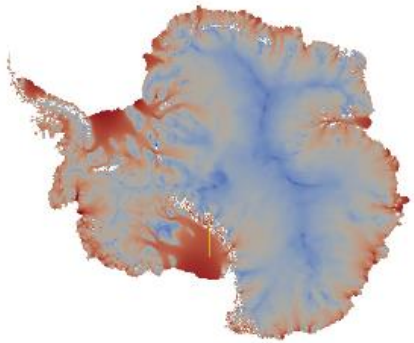
(vertical > horizontal coupling)
+
Neumann BCs
=
nearly singular submatrix associated with vertical lines

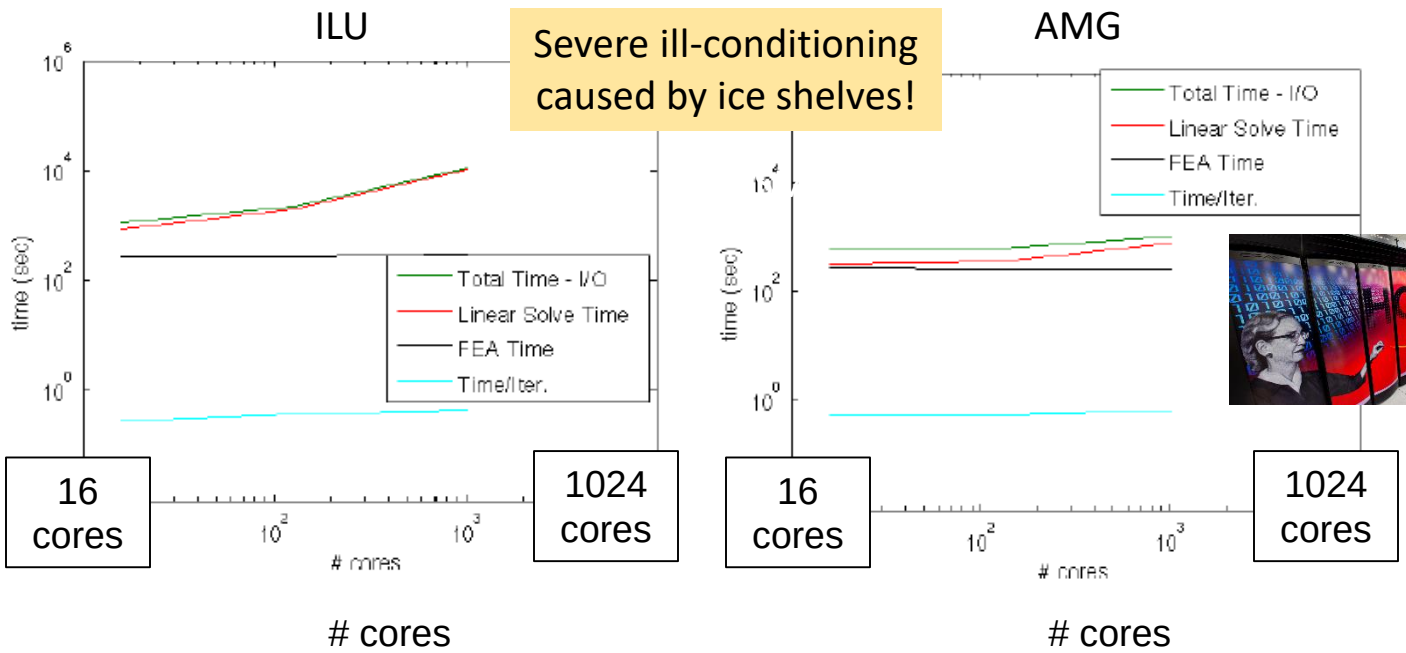
⇒ Ice shelves give rise to severe ill-conditioning of linear systems!

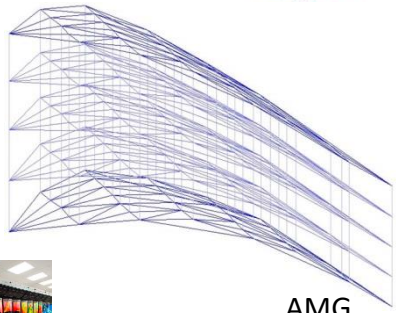
*Completely singular in the presence of islands and some ice tongues.

Moderate Resolution Antarctica Weak Scaling Study

- Weak scaling study on Antarctic problem (8km w/ 5 layers → 2km w/ 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.







AMG preconditioner

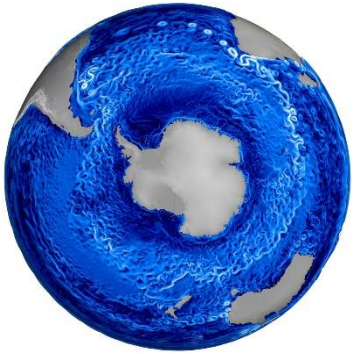
(vertical > horizontal coupling)
+
Neumann BCs
=
nearly singular submatrix associated with vertical lines

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

* Tuminaro *et al., SISC,* 2016.

Outline

1. Background
   - Motivation for climate & land-ice modeling
   - Earth system models & projects
   - Land-ice equations
2. **Algorithms and software**
   - **Albany/FELIX steady stress-velocity solver**
     - Meshes/data
     - Solvers & preconditioners
     - **Performance-portability**
     - Ice sheet initialization/UQ
   - CISM/MPAS-Albany for dynamic simulations
     - Velocity-thickness coupling
     - Validation
     - Towards science runs
3. Ongoing & future work
4. Summary



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.) .

MPI (inter-node parallelism) **+ X*** (intra-node parallelism)

- ***Kokkos***: open-source 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, ...
  - 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 (e.g., (i,j,k) vs. (k,i,j).

- **Finite element assembly** in *Albany* has recently been rewritten using *Kokkos* functors.
- Linear solvers in ***Belos*** package of *Trilinos* can run on next-generation platforms with simple preconditioners (Jacobi, Gauss-Seidel, Chebyshev, ILU).

*X = OpenMP, CUDA, etc.

Kokkos-ification of Finite Element Assembly

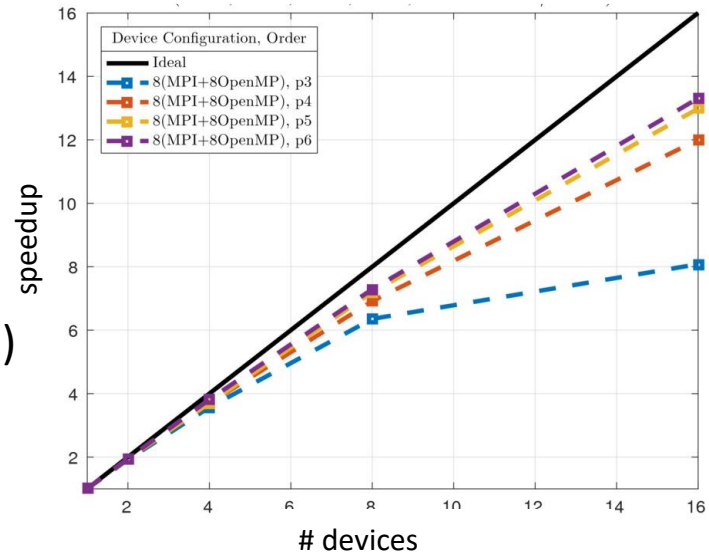
```
typedef Kokkos::OpenMP ExecutionSpace;
//typedef Kokkos::CUDA ExecutionSpace;
//typedef Kokkos::Serial ExecutionSpace;
template<typename ScalarT>
vectorGrad<ScalarT>::vectorGrad()
{
Kokkos::View<ScalarT****, ExecutionSpace> vecGrad("vecGrad", numCells, numQP, numVec, numDim);
}
**********************************************
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.)

Targeted Computer Architectures

Performance-portability of FEA in Albany has been tested across ***multiple architectures***: Intel Sandy Bridge, IBM Power8, Keplar/Pascal GPUs, KNL Xeon Phi

KNL MPI+OpenMP strong-scalability*

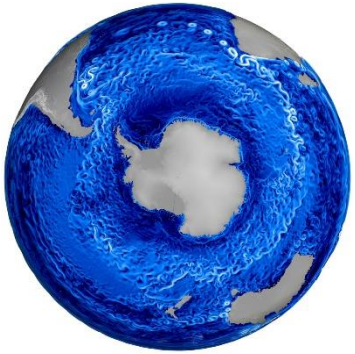


- **Ride** (SNL) used for verification, performance tests
  12 nodes (dual-Power8 (16 cores) + P100 quad-GPU)
- **Bowman** (SNL) used for verification
  10 nodes (Intel Xeon Phi KNL (68 cores))
- **Cori** (NERSC) used for verification, performance tests
  9688 nodes (Intel Xeon Phi KNL (68 cores))
- **Summit** (ORLCF) is near-future GPU target
  4600 nodes (dual-Power9 + 6 NVIDIA Volta)

* Demeshko *et al., JHPC,* 2017 (under review).

Outline

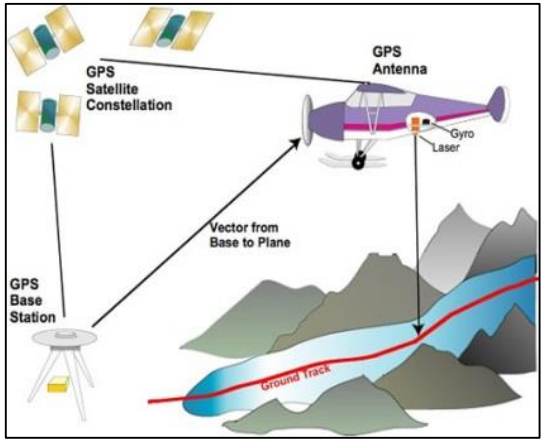
1. Background
   - Motivation for climate & land-ice modeling
   - Earth system models & projects
   - Land-ice equations
2. **Algorithms and software**
   - **Albany/FELIX steady stress-velocity solver**
     - Meshes/data
     - Solvers & preconditioners
     - Performance-portability
     - **Ice sheet initialization/UQ**
   - CISM/MPAS-Albany for dynamic simulations
     - Velocity-thickness coupling
     - Validation
     - Towards science runs
3. Ongoing & future work
4. Summary



Inversion for Ice Sheet Initialization

<u>Goal:</u> find ice sheet initial state that:

- matches observations (e.g. surface velocity, temperature).
- matches present-day geometry (elevation, thickness).
- is in "equilibrium" with climate forcings (SMB).



Available data/measurements:

- ice extent and surface topography.
- surface velocity.
- surface mass balance (SMB).
- ice thickness H (sparse measurements).

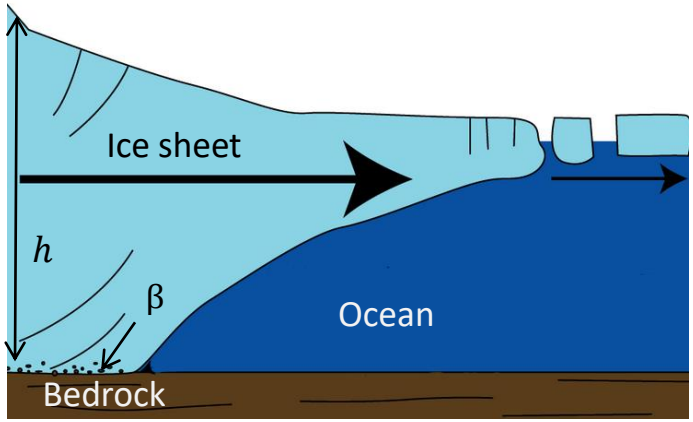
Fields to be estimated:

- ice thickness H (allowed to vary but weighted by observational uncertainties).
- basal friction β (spatially variable proxy for all basal processes).

Modeling Assumptions:

- ice flow described by nonlinear first-order Stokes equations.
- ice close to mechanical equilibrium.

Sources of data: satellite infrarometry, radar, altimetry, etc.



Deterministic Inversion

Sandia National Laboratories

First-Order Stokes PDE-Constrained optimization problem for initial condition*:

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

$\boldsymbol{U}$: computed depth averaged velocity
H: ice thickness
β: basal sliding friction coefficient
τ_s: surface mass balance (SMB)
$\mathcal{R}(\beta, H)$: regularization term

$$m(\beta, H) = \int_\Gamma \frac{1}{\sigma_u^2} |\boldsymbol{u} - \boldsymbol{u}^{obs}|^2 ds$$ surface velocity mismatch

$$+ \int_\Gamma \frac{1}{\sigma_\tau^2} |div(\boldsymbol{U}H) - \tau_s|^2 ds$$ SMB mismatch

$$+ \int_\Gamma \frac{1}{\sigma_H^2} |H - H^{obs}|^2 ds$$ thickness mismatch

$$+ \mathcal{R}(\beta, H)$$ regularization terms

* Perego, Stadler, Price, *JGR*, 2014.

Deterministic Inversion

Sandia National Laboratories

First-Order Stokes PDE-Constrained optimization problem for initial condition*:

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

$\boldsymbol{U}$: computed depth averaged velocity
H: ice thickness
β: basal sliding friction coefficient
τ_s: surface mass balance (SMB)
$\mathcal{R}(\beta, H)$: regularization term

$$m(\beta, H) = \int_\Gamma \frac{1}{\sigma_u^2} |\boldsymbol{u} - \boldsymbol{u}^{obs}|^2 ds$$ surface velocity mismatch

$$+ \int_\Gamma \frac{1}{\sigma_\tau^2} |div(\boldsymbol{U}H) - \tau_s|^2 ds$$ SMB mismatch

$$+ \int_\Gamma \frac{1}{\sigma_H^2} |H - H^{obs}|^2 ds$$ thickness mismatch

$$+ \mathcal{R}(\beta, H)$$ regularization terms

Solving FO Stokes PDE-constrained optimization problem for initial condition significantly **reduces non-physical model transients**!

* Perego, Stadler, Price, *JGR*, 2014.

Deterministic Inversion

Sandia National Laboratories

First-Order Stokes PDE-Constrained optimization problem for initial condition*:

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

$\boldsymbol{U}$: computed depth averaged velocity
H: ice thickness
β: basal sliding friction coefficient
τ_s: surface mass balance (SMB)
$\mathcal{R}(\beta, H)$: regularization term

$$m(\beta, H) = \int_\Gamma \frac{1}{\sigma_u^2} |\boldsymbol{u} - \boldsymbol{u}^{obs}|^2 ds$$ surface velocity mismatch

$$+ \int_\Gamma \frac{1}{\sigma_\tau^2} |div(\boldsymbol{U}H) - \tau_s|^2 ds$$ SMB mismatch

$$+ \int_\Gamma \frac{1}{\sigma_H^2} |H - H^{obs}|^2 ds$$ thickness mismatch

$$+ \mathcal{R}(\beta, H)$$ regularization terms

common

novel

Solving FO Stokes PDE-constrained optimization problem for initial condition significantly **reduces non-physical model transients**!

* Perego, Stadler, Price, *JGR*, 2014.

Deterministic Inversion Algorithm & Software

First-Order Stokes PDE-Constrained optimization problem for initial condition*:

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

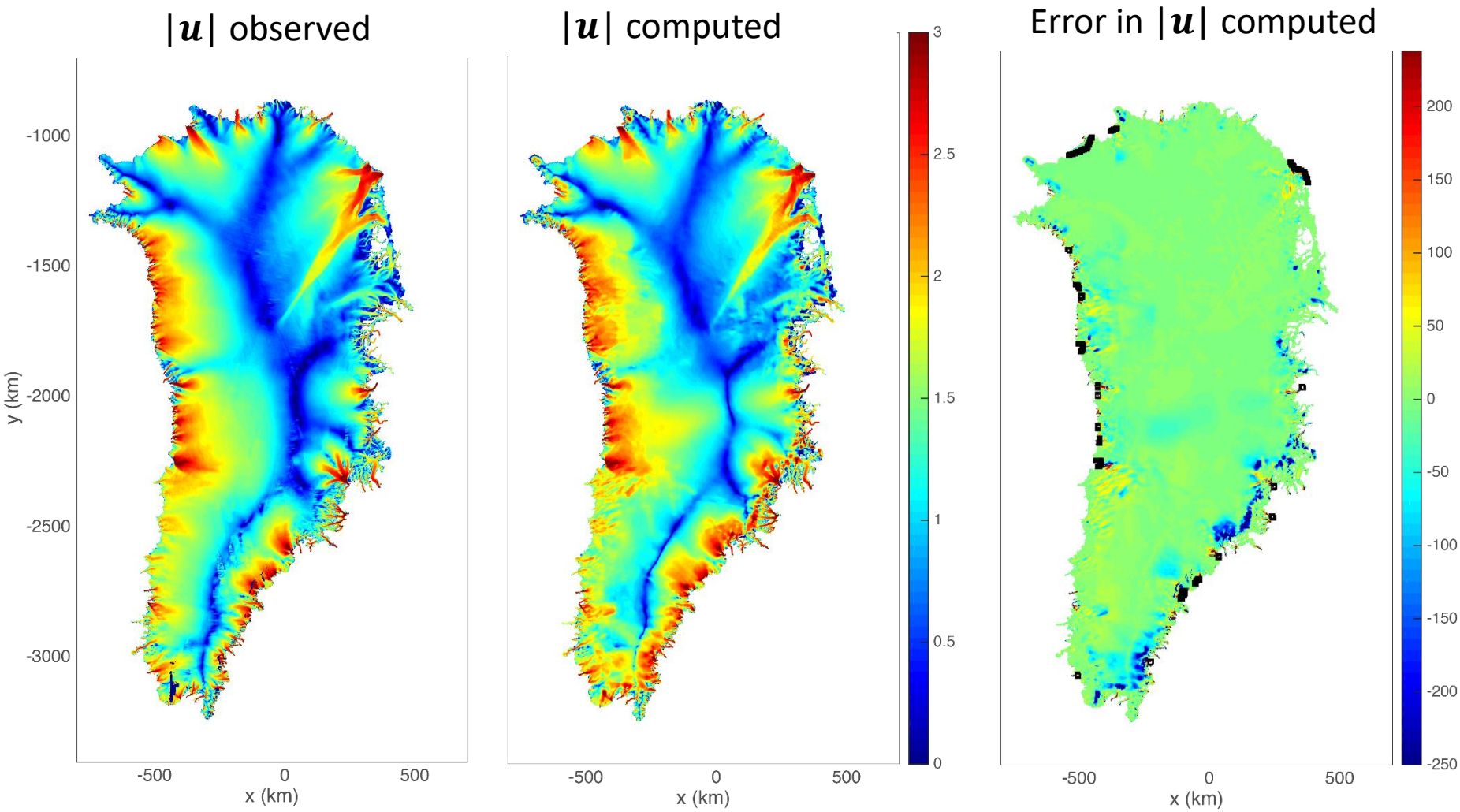
→ Solved via embedded ***adjoint-based PDE-constrained optimization*** algorithm in Albany/FELIX.

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

- **<u>Some details:</u>**
  - ***Regularization:*** Tikhonov.
  - Total derivatives of objective functional $m(\beta, H)$ computed using ***adjoints*** and ***automatic differentiation*** (Sacado package of Trilinos).
  - ***Gradient-based optimization***: limited memory BFGS initialized with Hessian of regularization terms (ROL) with backtrack linesearch.

* Perego, Stadler, Price, *JGR*, 2014.

Deterministic Inversion: 1km Greenland Initial Condition*

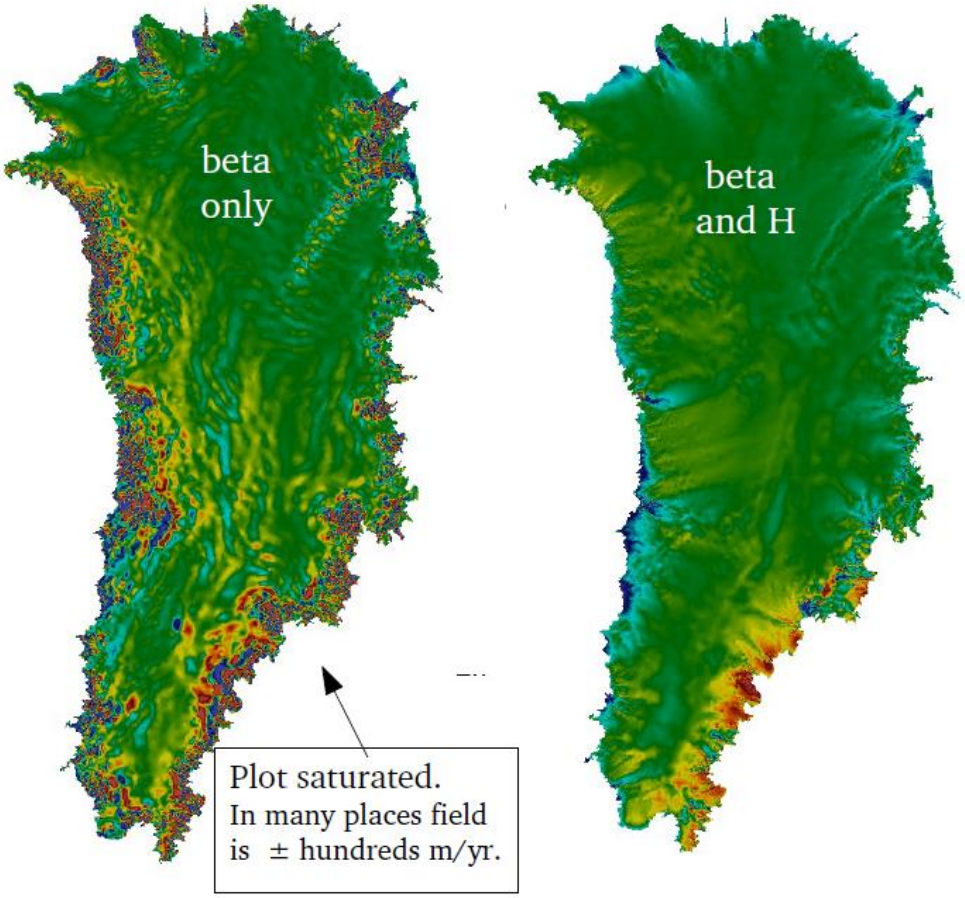


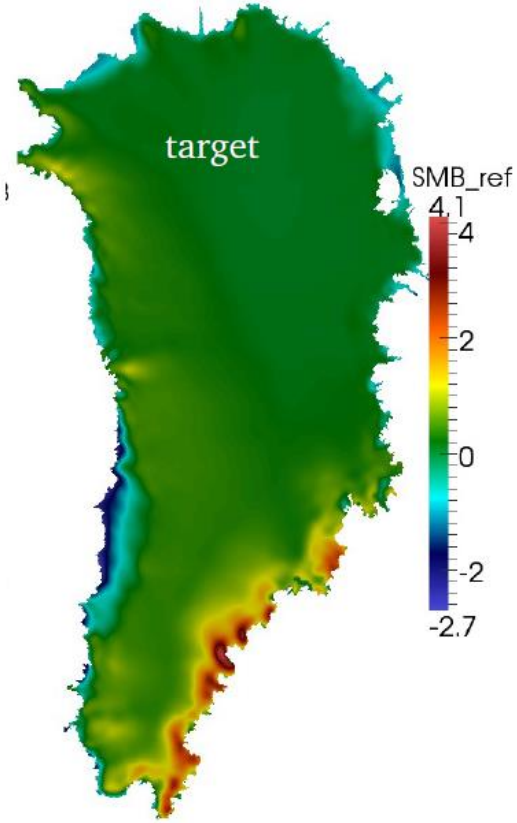
* Perego, Stadler, Price, *JGR*, 2014.

Deterministic Inversion: Common vs. Novel Approach*

SMB (m/yr) needed for equilibrium

SMB (m/yr) from climate model (Ettema et al. 2009, RACMO2/GR)





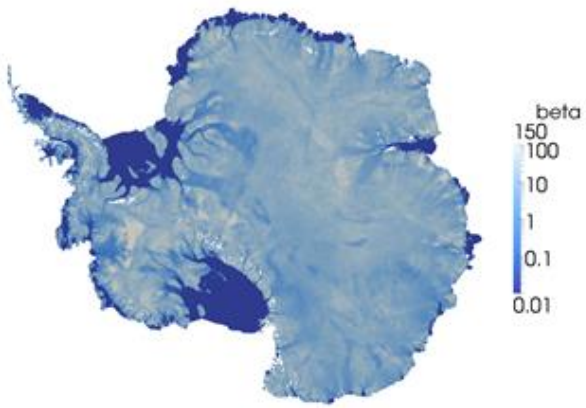
* Perego, Stadler, Price, *JGR*, 2014.

Deterministic Inversion: Antarctica (basal friction only)

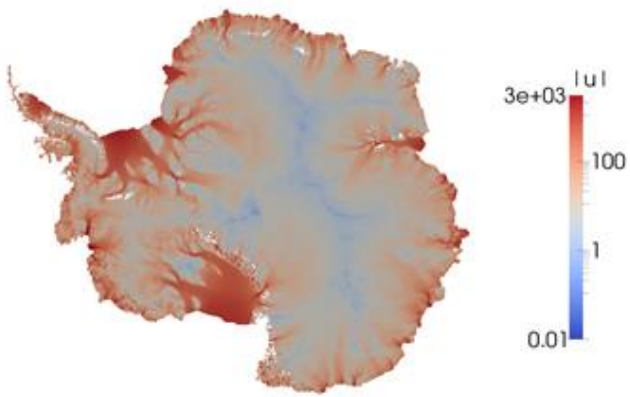
FO Stokes PDE Constrained Optimization Problem:

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

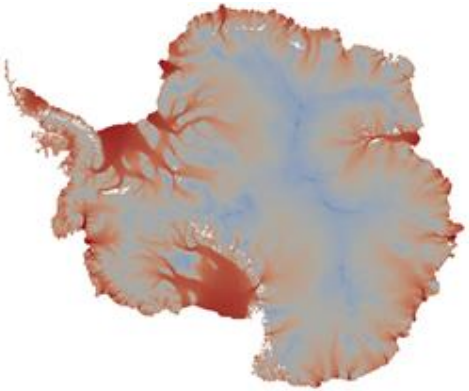
Geometry: Cornford, Martin *et al.* (in prep.)
Bedmap2: Fretwell et al., 2013
Temperature: Pattyn, 2010.



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

Uncertainty Quantification*

Goal: obtain PDF of initial condition using Bayesian inference and propagate this PDF through model to get PDF of ***total ice mass loss/gain during 21st century***

UQ Workflow

Stage 1:
Estimate ice sheet initial condition (MAP point).

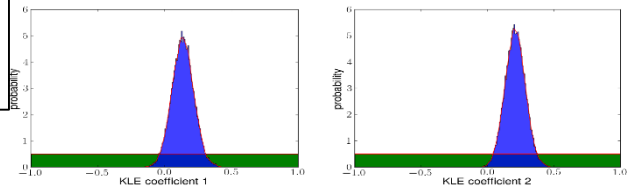
Deterministic inversion

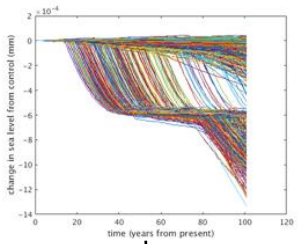
Stage 2:
Update prior uncertainty in ice sheet initial condition using observational data and steady state model

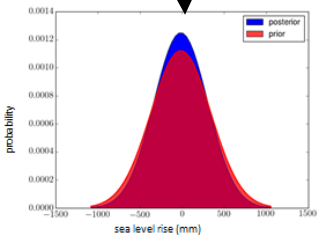
Bayesian calibration

Stage 3:
Propagate uncertain initial condition through ice-sheet evolution model

Forward propagation







β, H PDFs (from Bayesian inference)

SLR(t) for ensemble of forward runs with β, H sampled from its PDF

PDF of SLR

* Jakeman *et al.* (in prep), 2017.

Uncertainty Quantification*

Goal: obtain PDF of initial condition using Bayesian inference and propagate this PDF through model to get PDF of ***total ice mass loss/gain during 21st century***

UQ Workflow

Stage 1:
Estimate ice sheet initial condition (MAP point).

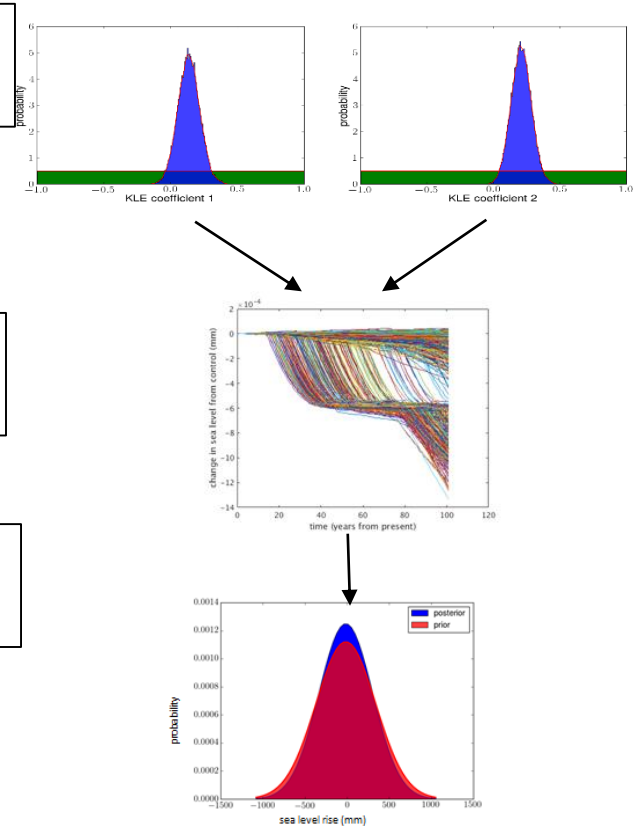
Deterministic inversion

Stage 2:
Update prior uncertainty in ice sheet initial condition using observational data and steady state model

Bayesian calibration

Stage 3:
Propagate uncertain initial condition through ice-sheet evolution model

Forward propagation



β, H PDFs (from Bayesian inference)

SLR(t) for ensemble of forward runs with β, H sampled from its PDF

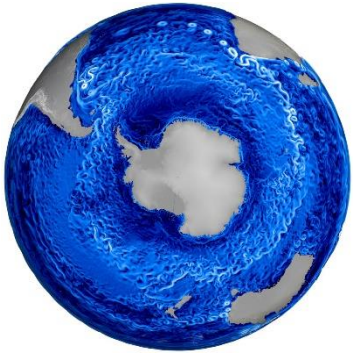
PDF of SLR

Very challenging! Lots of obstacles, e.g., curse of dimensionality.

* Jakeman *et al.* (in prep), 2017.

Outline

1. Background
   - Motivation for climate & land-ice modeling
   - Earth system models & projects
   - Land-ice equations
2. **Algorithms and software**
   - Albany/FELIX steady stress-velocity solver
     - Meshes/data
     - Solvers & preconditioners
     - Performance-portability
     - Ice sheet initialization/UQ
   - **CISM/MPAS-Albany for dynamic simulations**
     - Velocity-thickness coupling
     - Validation
     - Towards science runs
3. Ongoing & future work
4. Summary



Dynamic Land-Ice Simulations

Momentum Balance: ***First-Order Stokes*** PDEs

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

with ***Glen's law*** viscosity $\mu = \frac{1}{2} A(T)^{-\frac{1}{3}} \left(\frac{1}{2} \sum_{ij} \dot{\boldsymbol{\epsilon}}_{ij}{}^2 \right)^{\left(-\frac{2}{3}\right)}$.

Conservation of Mass: ***thickness*** evolution PDE

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

Energy Balance: ***temperature*** advection-diffusion PDE

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

Code:

Albany =multi-physics PDE code

Albany/FELIX***

Trilinos

CISM

MPAS Model for Prediction Across Scales

CISM/Albany
MPAS/Albany

CESM COMMUNITY EARTH SYSTEM MODEL

E3SM Energy Exascale Earth System Model

*https://github.com/gahansen/Albany. **FELIX = Finite Elements for Land Ice eXperiments.

Interfaces to *CISM/MPAS LI* for Transient Simulations

Sandia National Laboratories

E3SM Energy Exascale Earth System Model

CISM-Albany

cism_driver

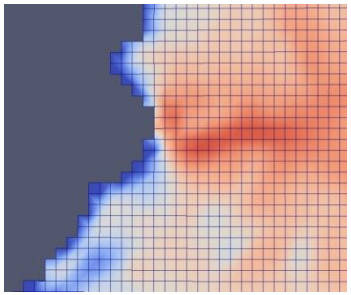
CISM (Fortran)
Thickness evolution, temperature solve, coupling to CESM

C++/Fortran Interface, Mesh Conversion

Albany/FELIX (C++)
velocity solve

output file

CESM COMMUNITY EARTH SYSTEM MODEL



- Structured hexahedral meshes (rectangles extruded to hexes).

MPAS LI-Albany

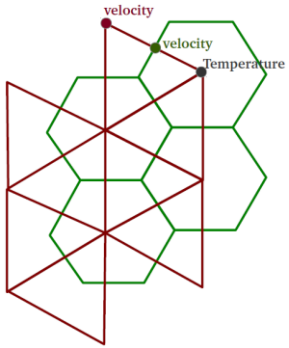
LandIce_model

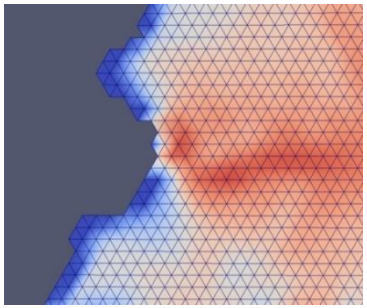
C++/Fortran Interface, Mesh Conversion

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

output file

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

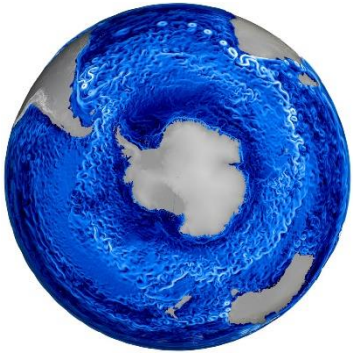




Albany/FELIX has been coupled to two land ice dycores: **C**ommunity **I**ce **S**heet **M**odel (***CISM***) and **M**odel for **P**rediction **A**cross **S**cales for **L**and **I**ce (***MPAS LI***)

Outline

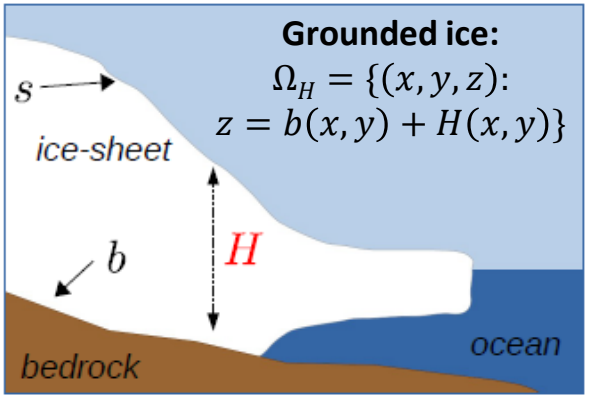
1. Background
   - Motivation for climate & land-ice modeling
   - Earth system models & projects
   - Land-ice equations
2. **Algorithms and software**
   - Albany/FELIX steady stress-velocity solver
     - Meshes/data
     - Solvers & preconditioners
     - Performance-portability
     - Ice sheet initialization/UQ
   - **CISM/MPAS-Albany for dynamic simulations**
     - **Velocity-thickness coupling**
     - Validation
     - Towards science runs
3. Ongoing & future work
4. Summary



First Order Stokes-Thickness Coupling Methods

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

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



- ***Sequential coupling*** (common approach):
  - Given H^n, solve FO Stokes system for $\boldsymbol{u}^n$.
  - Compute $\overline{\boldsymbol{u}}$ 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.

- ***Semi-Implicit coupling*** (new approach):

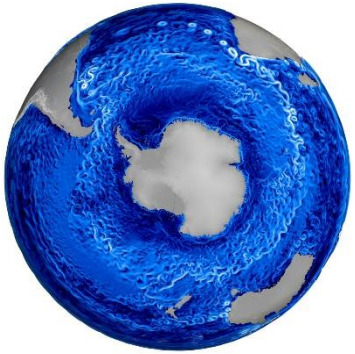
$$-2\mu\nabla\cdot\dot{\boldsymbol{\epsilon}}(\boldsymbol{u}) = -\rho g\nabla(b+H)\,, \quad \text{in } \Omega_H$$

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

  - $\boldsymbol{u}$ 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).

Outline

1. Background
   - Motivation for climate & land-ice modeling
   - Earth system models & projects
   - Land-ice equations
2. **Algorithms and software**
   - Albany/FELIX steady stress-velocity solver
     - Meshes/data
     - Solvers & preconditioners
     - Performance-portability
     - Ice sheet initialization/UQ
   - **CISM/MPAS-Albany for dynamic simulations**
     - Velocity-thickness coupling
     - **Validation**
     - Towards science runs
3. Ongoing & future work
4. Summary

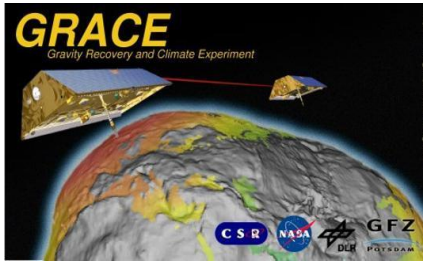


Dynamic Simulations: Validation

Validation*: how well does our model represent the real ice sheet?

- CISM-Albany 2.0 on Greenland has been validated using data from 2 ***satellites:***

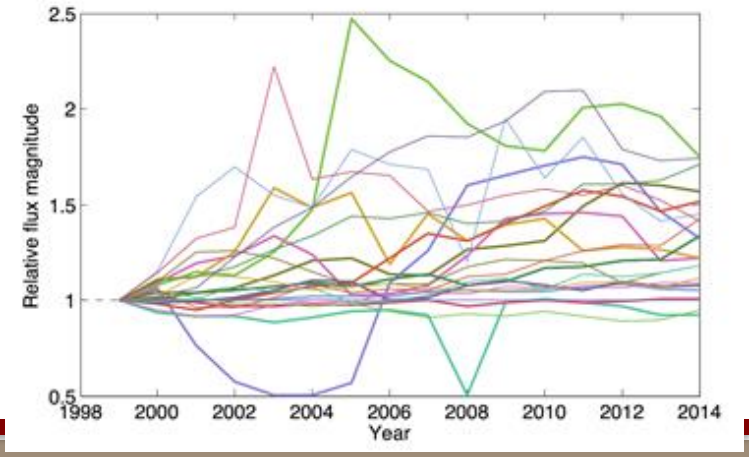
ICESat1 [states]	2003 – 2009
GRACE [trends]	2002 – 201? (ongoing)



[1] van Angelen et al. (*Surv. Geophys.*, 2013)
[2] Enderlin et al. (*GRL*, 2014)

- **<u>Validation time period</u>**:
  - ICESat: 2003-2009
  - GRACE: 2003-2011 (CSR Release-05)

- **<u>Model forcing</u>**: the following datasets taken as "truth"
  - Monthly ***surface mass balance*** (SMB) from RACMO2[1] anomalies applied (1960-present)
    - Well-validated over Greenland
  - Mean-annual ***outlet glacier flux***[2] applied at grounding line (1990-present)
    - Right figure: outlet glacier ***flux forcing (FF)*** series



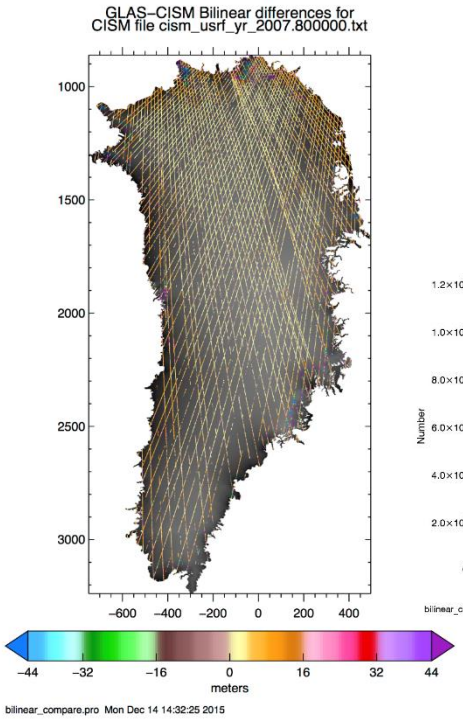
*vs. verification: is our code bug free

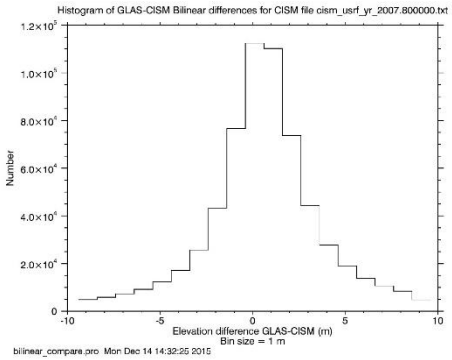
Validation Results*

Cryosphere Model Comparison Tool (CmCt) **

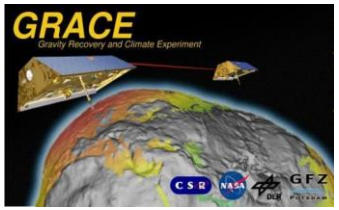
ICESat: Surface Elevation Comparison

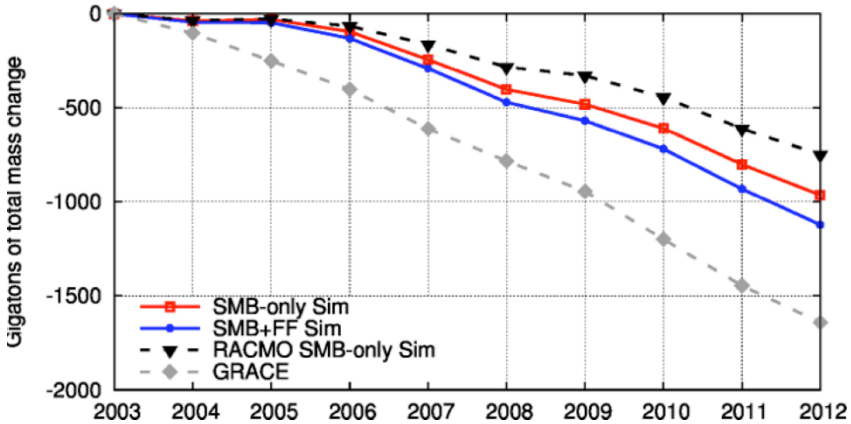
- ***CISM-Albany*** with ***SMB + FF*** in October 2007.
- Surface elevation predictions (states) agree pretty well with ***GLAS*** (***Geoscience Laser Altimeter System aboard ICESat***): ***mean differences*** are ***<1 m***





GRACE: Whole Ice Sheet Mass Trends





- ***Apples-to-apples*** comparison between models and GRACE:
  - Information is all on the ***same plot***.
  - Model output and observations were both ***processed in the same way.***

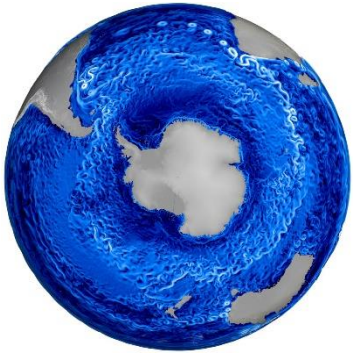
* Price et al, *GMD*, 2017. ** https://ggsghpcc.sgt-inc.com/cmct/

Visualization of Validation Results

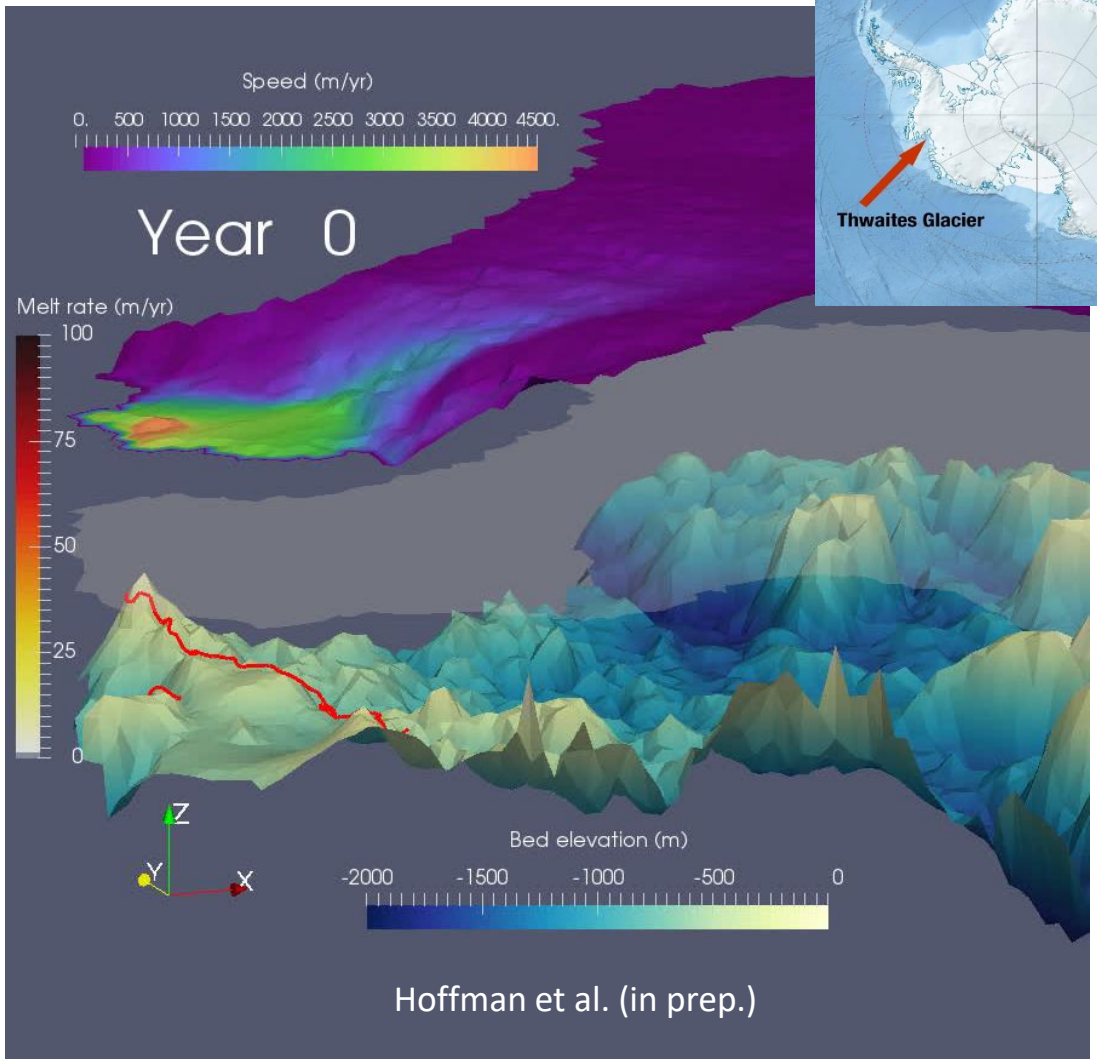
Video acknowledgement: B. Carvey (SNL)

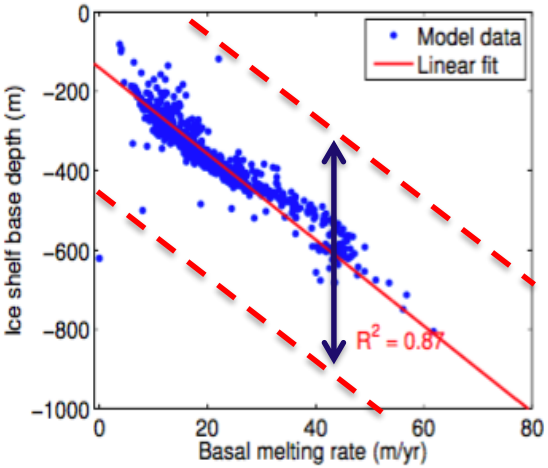
Outline

1. Background
   - Motivation for climate & land-ice modeling
   - Earth system models & projects
   - Land-ice equations
2. **Algorithms and software**
   - Albany/FELIX steady stress-velocity solver
     - Meshes/data
     - Solvers & preconditioners
     - Performance-portability
     - Ice sheet initialization/UQ
   - **CISM/MPAS-Albany for dynamic simulations**
     - Velocity-thickness coupling
     - Validation
     - **Towards science runs**
3. Ongoing & future work
4. Summary



MPAS-Albany Thwaites Glacier Simulation



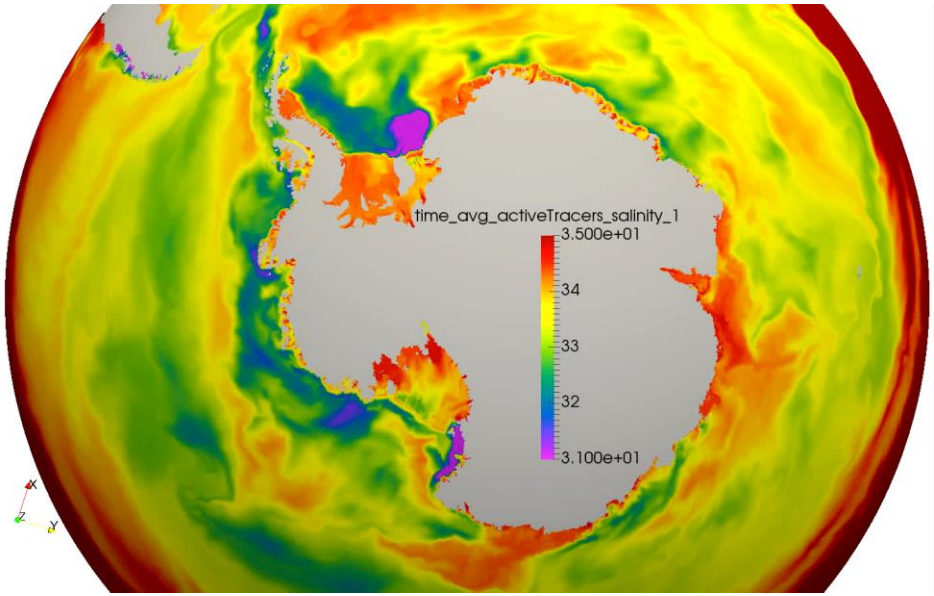


- Movie shows ***Thwaites Glacier*** retreat simulation under parameterized submarine melting.
- 250 year ***regional simulation*** with "present day" initial condition.
- Investigate importance of ***CDW* depth changes*** due to climate variability.
- When ***climate variability*** in sub-shelf forcing is accounted for, we get a ***distribution*** of possible SLR curves.

* CDW = Circumpolar Deep Water.

PISCEES & E3SM Coupling

MPAS-Albany is (partially) coupled to E3SM

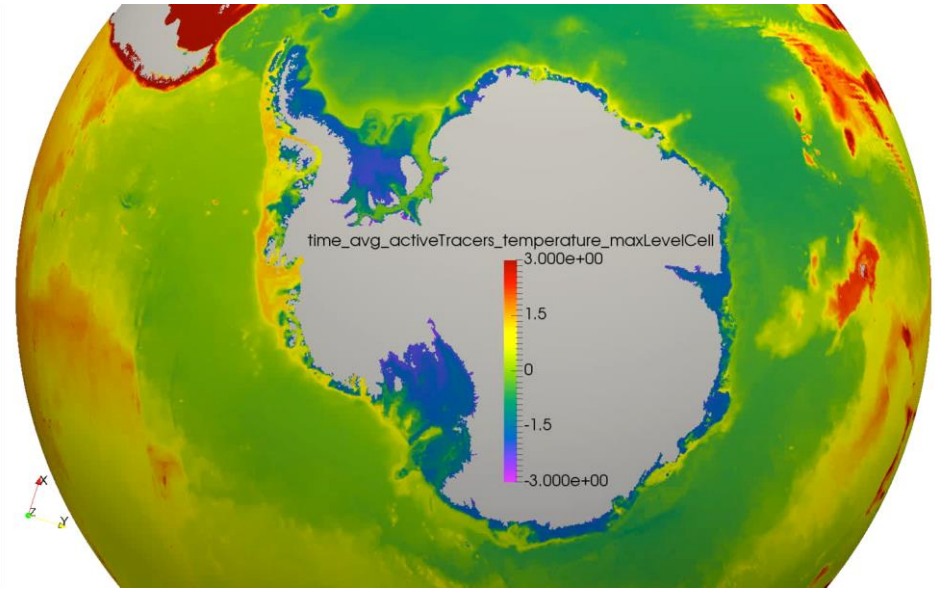


Sea Surface Salinity

- ***Global***, ***coupled*** E3SM simulation with sub-ice shelf circulation + pre-industrial forcing + static ice shelves (***illustration/spin-up over ~7 yrs***).
- RRS30to10km mesh (eddy permitting).

Fully ***coupled***, ***dynamic ice sheet*** simulations will be done through ***ProSPect***: awarded ~85M CPU hours at 3 DOE computing centers.

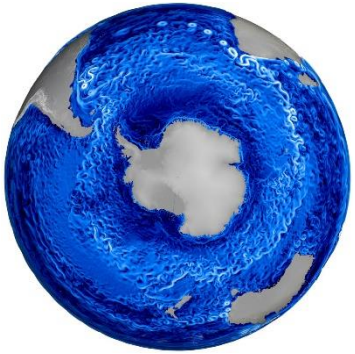
- Top: sea-surface salinity; right: ocean bottom temperature



Ocean Bottom Temperature

Outline

1. Background
   - Motivation for climate & land-ice modeling
   - Earth system models & projects
   - Land-ice equations
2. Algorithms and software
   - Albany/FELIX steady stress-velocity solver
     - Meshes/data
     - Solvers & preconditioners
     - Performance-portability
     - Ice sheet initialization/UQ
   - CISM/MPAS-Albany for dynamic simulations
     - Velocity-thickness coupling
     - Validation
     - Towards science runs
3. **Ongoing & future work**
4. Summary



Ongoing & future work

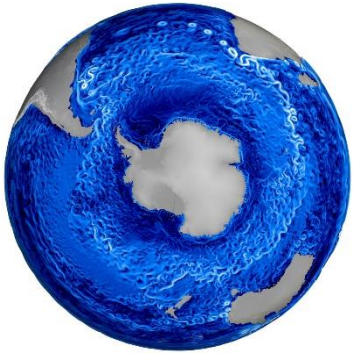
Probabilistic Sea-Level Projections from Ice Sheet and Earth System Models (ProSPect) is a new 5 year (2017-2022) SciDAC project on:

1) ice sheet and ocean model ***physics*** critical for accurate projections of sea-level change (e.g., subglacial hydrology, damage evolution + fracture + calving).
2) ice sheet, ocean, and ESM ***coupling*** critical for accurate projections of sea-level change
3) ice sheet model ***initialization*** and ***optimization*** methods needed for realistic coupling of ISMs and ESMs
4) frameworks for quantifying parametric and structural ice sheet model ***uncertainties***
5) ***performance portability*** on new, heterogeneous HPC architectures

New developments will be targeted at ***standalone*** and ***coupled*** simulations of sea-level rise from ice sheets

Outline

1. Background
   - Motivation for climate & land-ice modeling
   - Earth system models & projects
   - Land-ice equations
2. Algorithms and software
   - Albany/FELIX steady stress-velocity solver
     - Meshes/data
     - Solvers & preconditioners
     - Performance-portability
     - Ice sheet initialization/UQ
   - CISM/MPAS-Albany for dynamic simulations
     - Velocity-thickness coupling
     - Validation
     - Towards science runs
3. Ongoing & future work
4. **Summary**



Summary

- ***Actionable projections of climate change*** and SLR impacts are part of DOE's mission.
- DOE has and continues to invest in ***developing, testing and coupling*** of models and methods towards these problems.
- A ***mature ice-sheet modeling capability*** (high-fidelity, high-performance) was developed as a part of the PISCEES SciDAC project. This talk described the following aspects of creating this capability:
  - ***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.
- Related capabilities on the ESM side are rapidly ***maturing***.
- Ongoing and new projects are focusing on the remaining work (physics, coupling, uncertainty quantification frameworks) necessary to provide ***SLR projections and uncertainties***.

Careers at Sandia

<u>Students:</u> please consider Sandia and other national labs as a potential employer for summer internships and when you graduate!

- Sandia is a ***great place to work***!
  - Lots of interesting problems that require ***fundamental research*** in mathematics/computational science **and** impact ***mission-critical*** applications.
  - Great ***work/life balance***!

- Sandia has ***openings*** for:
  - Summer internships.
  - Post docs.
  - Named post doc fellowships (von Neumann, Truman, Hruby).
  - Staff.

- Sandia has ***two locations***:
  - Albuquerque, NM.
  - Livermore, CA.

Please see: www.sandia.gov/careers and/or email me: ikalash@sandia.gov about opportunities.

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", *Int. J. Multiscale Comput. Engng* 14(4) (2016) 415-438.

[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. Tuminaro, M. Perego, **I. Tezaur**, A. Salinger, S. Price. "A matrix dependent/algebraic multigrid approach for extruded meshes with applications to ice sheet modeling", *SIAM J. Sci. Comput.* 38(5) (2016) C504-C532.

[6] S. Price, M. Hoffman, J. Bonin, T. Neumann, I. Howat, J. Guerber, **I. Tezaur**, J. Saba, J. Lanaerts, D. Chambers, W. Lipscomb, M. Perego, A. Salinger, R. Tuminaro. "An ice sheet model validation framework for the Greenland ice sheet", *Geosci. Model Dev.* 10 (2017) 255-270.

[7] M. Salloum, N. Fabian, D. Hensinger, J. Lee, E. Allendorf, A. Bhagatwala, M. Blaylock, J. Chen, J. Templeton, **I. Tezaur**. "Optimal Compressed Sensing and Reconstruction of Unstructured Mesh Datasets", *Data Sci. & Engng.* (in press).

References (cont'd)

[8] 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.

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

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

[11] 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.

[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.

[13] I. Demeshko, J. Watkins, **I. Tezaur**, O. Guba, W. Spotz, A. Salinger, R. Pawlowski, M. Heroux. "Towards performance-portability of the Albany finite element analysis code using the Kokkos library", submitted to *J. HPC Appl*.

[14] J. Jakeman, **I. Tezaur**, M. Perego, S. Price. "Probabalistic Projections of Sea-Level Change from the Greenland Ice Sheet", in preparation.

Appendix: Motivation

Department of Energy (DOE) interests in climate change and sea-level rise:

- *"Addressing the effects of climate change is a **top priority** of the DOE."**
- DOE report on energy sector vulnerabilities: *"… **higher risks** to energy infrastructure located along the coasts thanks to sea level rise, the increasing intensity of storms, and higher storm surge and flooding."***

*http://energy.gov/science-innovation/climate-change
**http://energy.gov/articles/climate-change-effects-our-energy

Appendix: The PISCEES Project

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

Appendix: The PISCEES Project

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

Appendix :Albany/FELIX FEM FO 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

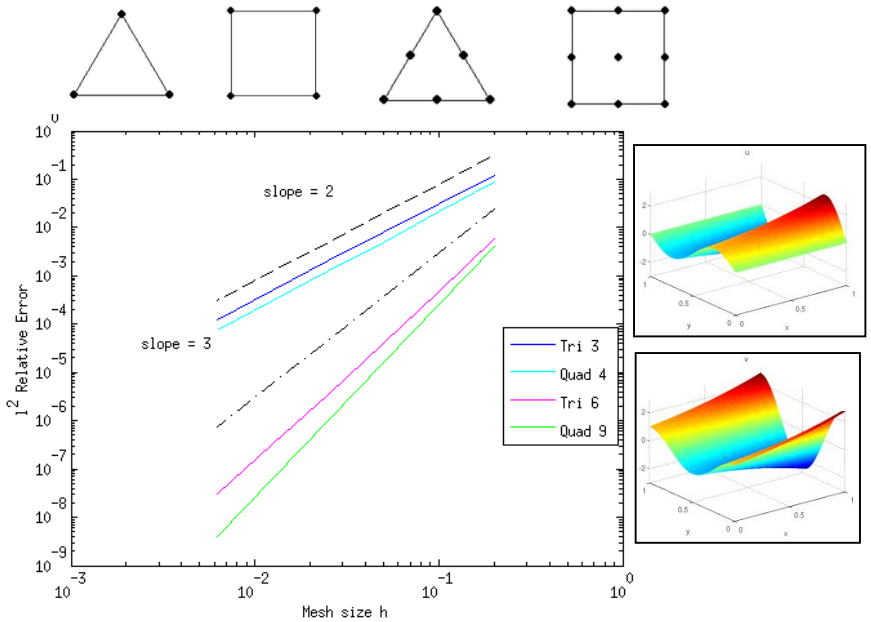
*Use of **Trilinos** components has enabled the **rapid** development of the **Albany/FELIX** First Order Stokes dycore!*

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

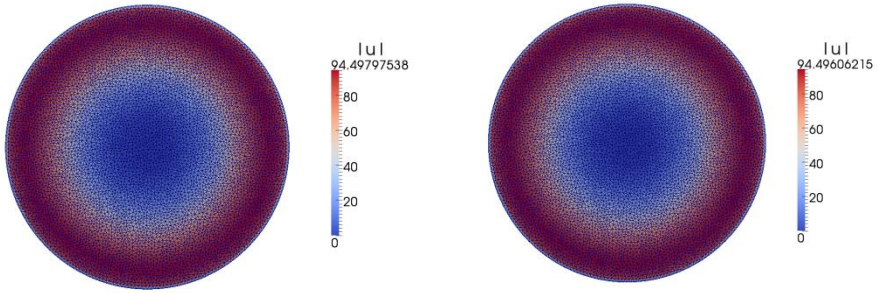
** "FELIX" = Finite Elements for Land-Ice eXperiments.

Appendix: Mesh Convergence Studies

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

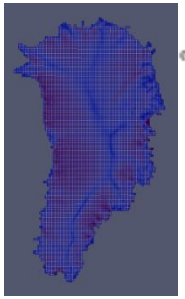


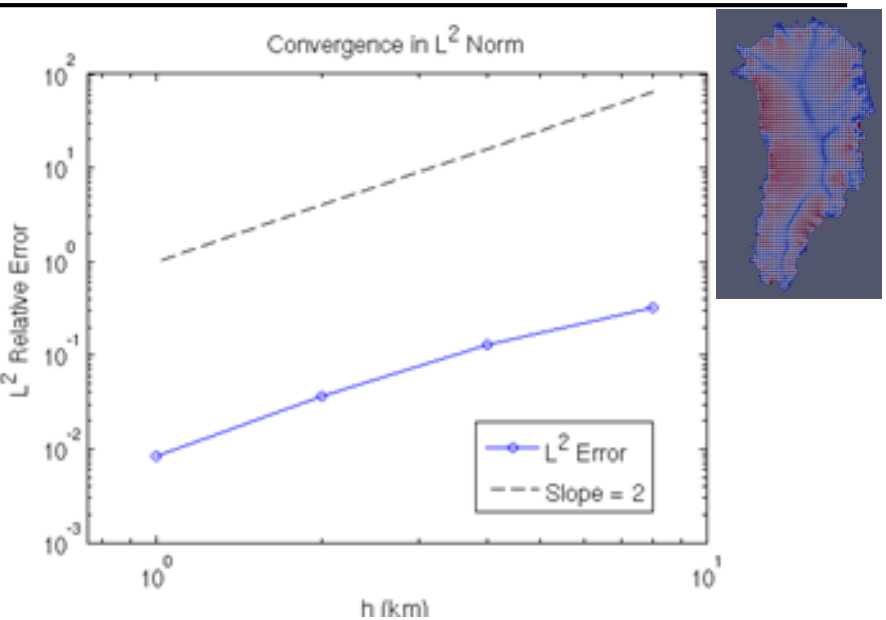
Stage 2: code-to-code comparisons on canonical ice sheet problems.



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: 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:

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

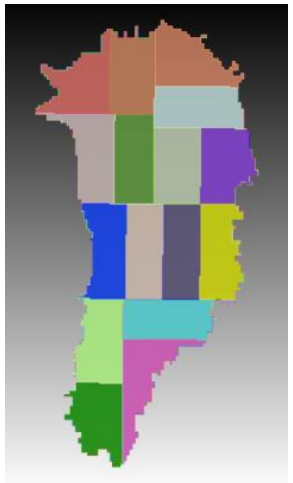
		$\frac{d}{dx}$
$x \leftarrow 2$	$\frac{dx}{dx} \leftarrow 1$	1.000
$t \leftarrow e^x$	$\frac{dt}{dx} \leftarrow t\frac{dx}{dx}$	7.389
$u \leftarrow \log x$	$\frac{du}{dx} \leftarrow \frac{1}{x}\frac{dx}{dx}$	0.500
$v \leftarrow xu$	$\frac{dv}{dx} \leftarrow u\frac{dx}{dx} + x\frac{du}{dx}$	1.301
$w \leftarrow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

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).*

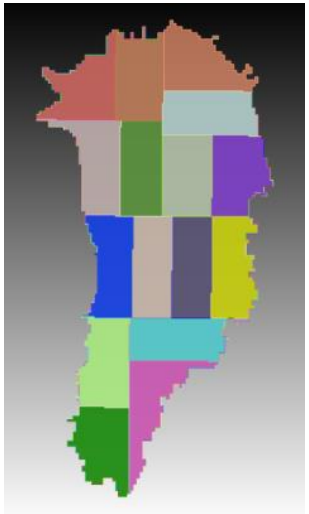


Horiz. res.\vert. layers	5	10	20	40	80
8km	2.0e-1				
4km	9.0e-2	7.8e-2			
2km	4.6e-2	2.4e-2	2.3e-2		
1km	3.8e-2	8.9e-3	5.5e-3	5.1e-3	
500m	3.7e-2	6.7e-3	1.7e-3	3.9e-4	8.1e-5

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

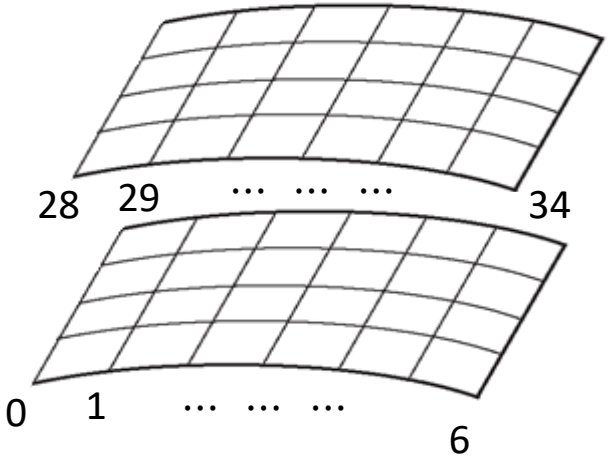
Appendix: Importance of Node Ordering & Mesh Partitioning

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).

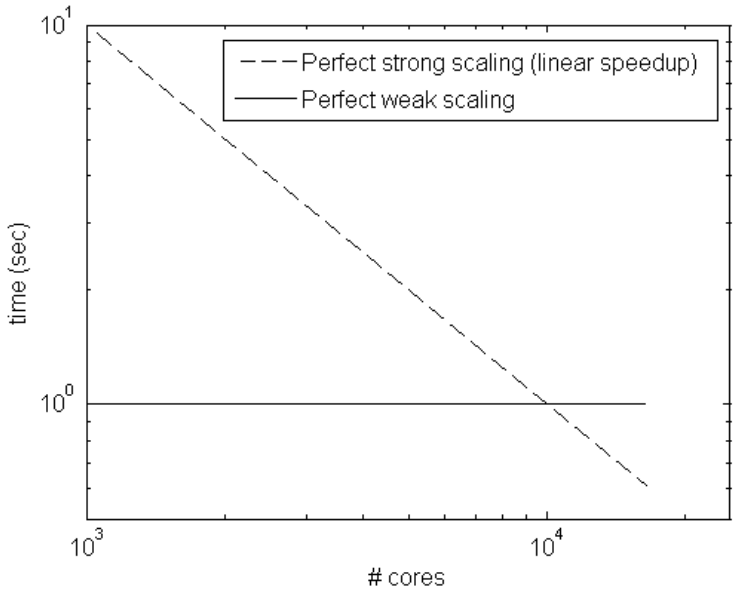


Appendix: Strong vs. Weak Scaling

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.
  - ⇒ Fix problem size, increase # cores.
  - **Ideal:** 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.
  - ⇒ Increase problem size and # cores s.t. # dofs/core is approximately constant.
  - **Ideal:** solution time remains constant as problem size and # cores increases.

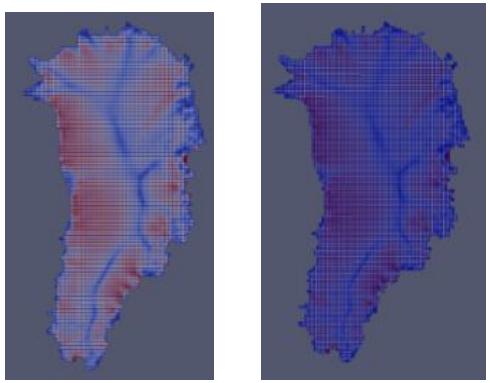


Note: scalability usually declines above some threshold number of cores (Amdahl's Law, Gustafson's Law) → some operations are serial, communication is not free, etc.

Appendix: Improved Linear Solver Performance through Hinge Removal

Islands and certain hinged peninsulas lead to **solver failures**

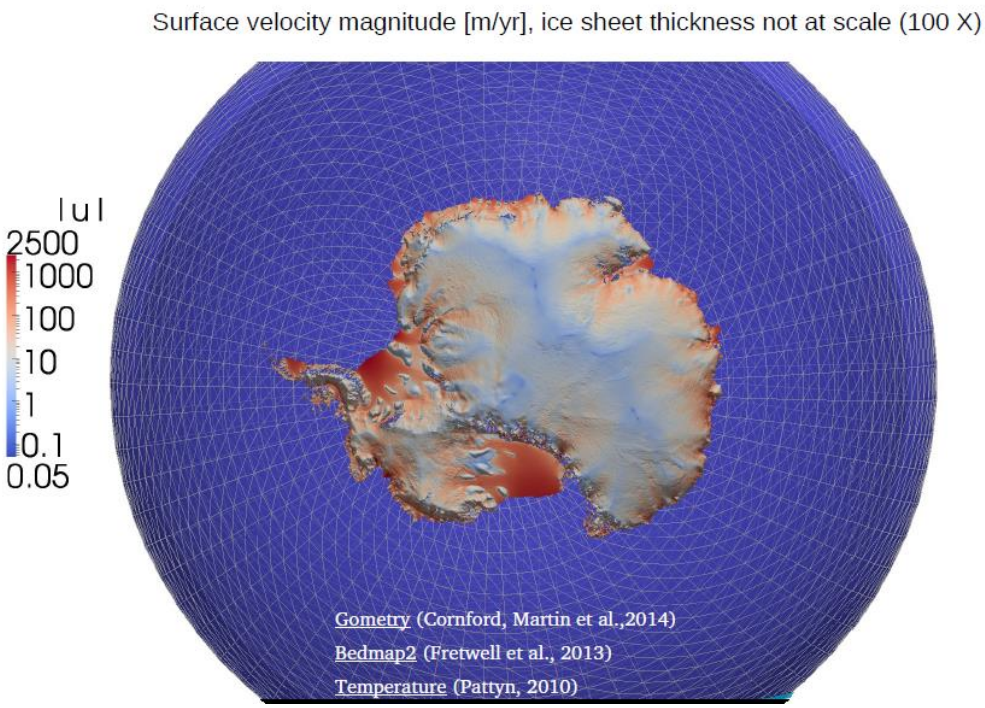
- We have developed an algorithm to detect/remove problematic ***hinged peninsulas*** & ***islands*** based on coloring and repeated use of connected component algorithms (Tuminaro *et al.*, 2016).
- Solves are ***~2x faster*** with hinges removed.
- Current implementation is MATLAB, but working on C++ implementation for integration into dycores.



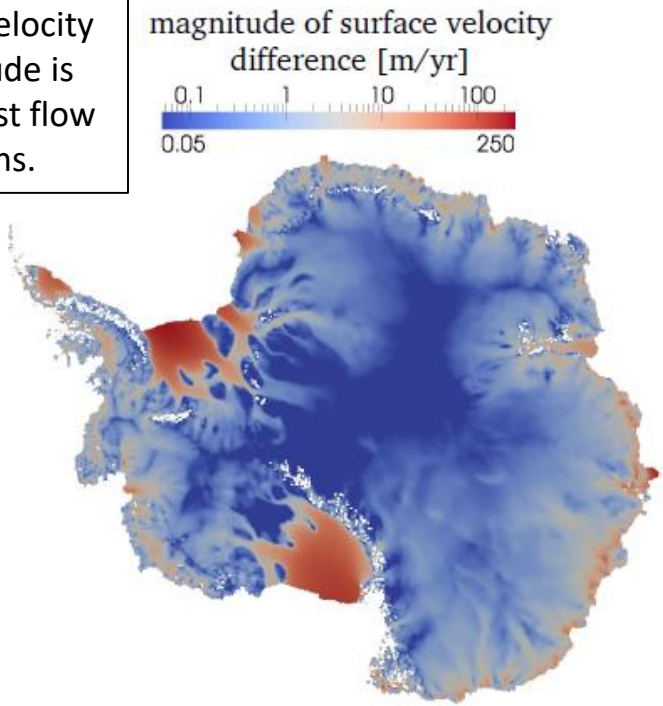
Greenland Problem

Resolu-tion	ILU – hinges	ILU – no hinges	ML – hinges	ML – no hinges
8km/5 layers	878 sec, 84 iter/solve	693 sec, 71 iter/solve	254 sec, 11 iter/solve	220 sec, 9 iter/solve
4km/10 layers	1953 sec, 160 iter/solve	1969 sec, 160 iter/solve	285 sec, 13 iter/solve	245 sec, 12 iter/solve
2km/20 layers	10942 sec, 710 iter/solve	5576 sec, 426 iter/solve	482 sec, 24 iter/solve	294 sec, 15 iter/solve
1km/40 layers	--	15716 sec, 881 iter/solve	668 sec, 34 iter/solve	378 sec, 20 iter/solve

Appendix: Spherical Grids



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



- 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: UQ Problem Definition

QoI in Ice Sheet Modeling: total ice mass loss/gain during 21st century → ***sea level change prediction.***

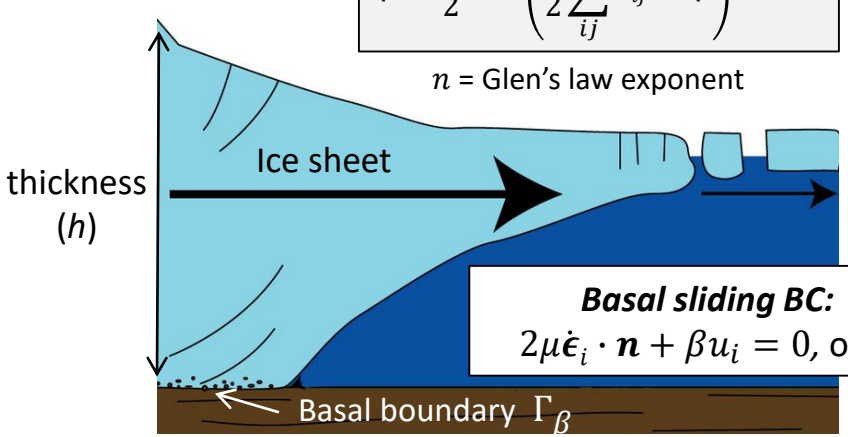
As a first step, we focus on effect of uncertainty in β only.

<u>Sources of uncertainty affecting this QoI include:</u>

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

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

n = Glen's law exponent



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

UQ Workflow

Deterministic inversion

Stage 1:
Estimate ice sheet initial condition (MAP point).

Bayesian calibration

Stage 2:
Update prior uncertainty in ice sheet initial condition using observational data and steady state model

Forward propagation

Stage 3:
Propagate uncertain initial condition through ice-sheet evolution model

Appendix: Bayesian Inference

UQ Workflow

Stage 1:
Estimate ice sheet initial condition (MAP point).

Stage 2:
Update prior uncertainty in ice sheet initial condition using observational data and steady state model

Stage 3:
Propagate uncertain initial condition through ice-sheet evolution model

Deterministic inversion is consistent with Bayesian analog: it is used to find the MAP point of posterior.

Goal: solve inverse problem for ice sheet initial state but in ***Bayesian framework***

- ***Naïve parameterization***: represent each degree of freedom on mesh be an uncertain variable

$$\beta(\boldsymbol{x}) = (z_1, z_2, \dots, z_{n_{\text{dof}}})$$

Intractable due to **curse of dimensionality**: $n_{\text{dof}} = O(100K)$!

- ***To circumvent this difficulty:*** assume $\beta(\boldsymbol{x})$ can be represented in ***reduced basis*** (e.g., KLE modes, Hessian eigenvectors*) centered around mean $\bar{\beta}(\boldsymbol{x})$:

$$\log(\beta(\boldsymbol{x})) = \log(\bar{\beta}) + \sum_{i=1}^{d} \sqrt{\lambda_i}\, \phi_i(\boldsymbol{x}) z_i$$

- Mean field $\bar{\beta}(\boldsymbol{x})$ = initial condition.

* Isaac, Petra, Stadler, Ghattas, *JCP*, 2015.

Appendix: Bayesian Inference Assumptions

- Additive ***Gaussian noise*** model: $\boldsymbol{y}^{\text{obs}} = \boldsymbol{f}(\boldsymbol{z}) + \epsilon, \quad \epsilon \sim N(\boldsymbol{0}, \boldsymbol{\Gamma}_{\text{obs}})$

⇒ Mismatch functional to be minimized:

$$m(\boldsymbol{z}) = \frac{1}{2}\left(\boldsymbol{y}^{\text{obs}} - \boldsymbol{f}(\boldsymbol{z})\right)^T \boldsymbol{\Gamma}_{\text{obs}}^{-1}\left(\boldsymbol{y}^{\text{obs}} - \boldsymbol{f}(\boldsymbol{z})\right)$$

Evaluation of misfit Hessian is **expensive!** ⇒ further approximation required.

- ***Gaussian prior*** with exponential covariance and mean $\boldsymbol{z}_{MAP} = \bar{\beta}$.

Bayes' rule

+ linearization of $\boldsymbol{f}(\boldsymbol{z})$ around $\boldsymbol{z}_{MAP}$

Covariance of Gaussian posterior related to ***inverse of misfit Hessian*** at MAP point**.

Notation*:

$\boldsymbol{y}^{\text{obs}}$= observations

$\boldsymbol{z}$ = random params

$\boldsymbol{f}(\boldsymbol{z})$ = deterministic map from params to observables.

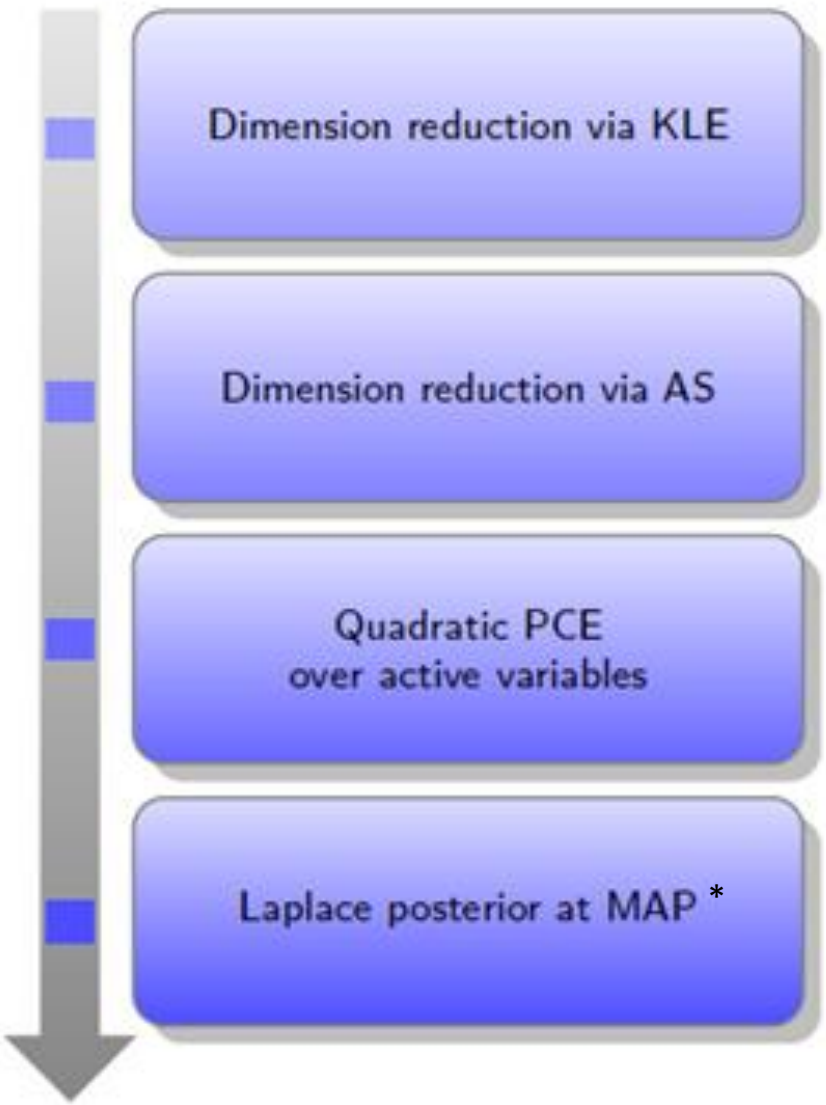
- ***Likelihood*** is: $\hat{\pi}_{\text{lhood}}(\boldsymbol{z}) = e^{-m_{\text{lin}}(\boldsymbol{z})}$

- ***Normal Laplace posterior*** given by: $\boxed{\pi_{\text{pos}}(\boldsymbol{z}) = C_{\text{evid}}^{-1}\hat{\pi}_{\text{lhood}}(\boldsymbol{z})\pi_{\text{pr}}(\boldsymbol{z})}$

  where $C_{evid} = \int \hat{\pi}_{\text{lhood}}(\boldsymbol{z})\pi_{\text{pr}}(\boldsymbol{z})d\boldsymbol{z}$.

* Constantine, Kent, Bui-Thanh, *SISC*, 2016. **Bui-Thanh, Ghattas, Martin, Stadler, *SISC*, 2013.

Appendix: Bayesian Inference Workflow



Two-part ***dimension reduction*** procedure to obtain modes $\phi_i(\boldsymbol{x})$

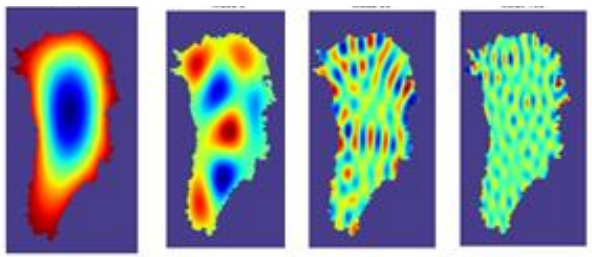
Procedure for computing ***covariance of normal Laplace posterior***, $\boldsymbol{\Gamma}_{\text{post}}$

KLE = Karhunen-Loeve Expansion
AS = Active Subspace
PCE = Polynomial Chaos Expansion
MAP = Maximum a Posteriori

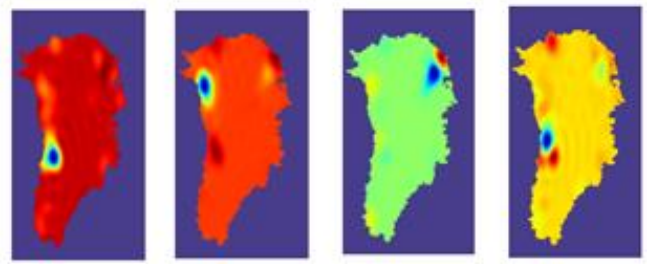
* Bui-Thanh, Ghattas, Martin, Stadler, *SISC*, 2013.

Appendix: GIS Bayesian Inference via KLE + AS

KLE modes



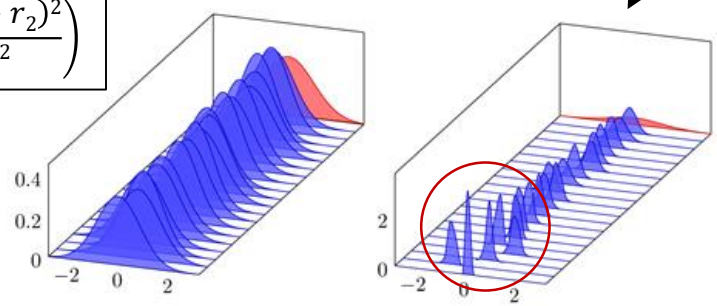
Data-informed (AS) directions (d=73*)



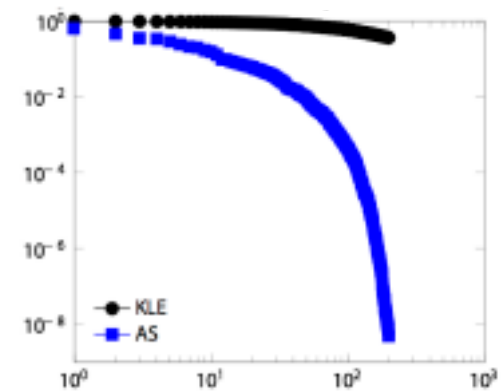
Gradients of mismatch function obtained via ***adjoint solve*** in Albany/FELIX.

KLE modes = eigenvecs of exponential covariance kernel:

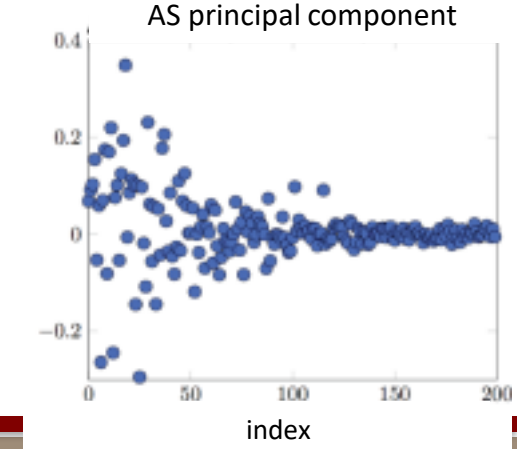
$$C(r_1, r_2) = \exp\left(-\frac{(r_1 - r_2)^2}{L^2}\right)$$



KLE and AS amplitudes



AS principal component



- ***Above:*** marginal distributions of Gaussian posterior computed using KLE vs. KLE+AS; ***any shift from mean of 0 is due to observations***.
  - KLE eigenvectors have variance and mean close to prior.
  - Data-informed eigenvectors have smaller variance and are most shifted w.r.t. prior distribution (as expected).

* Value of d was obtained via cross-validation.

Appendix: Forward Propagation of Uncertainty

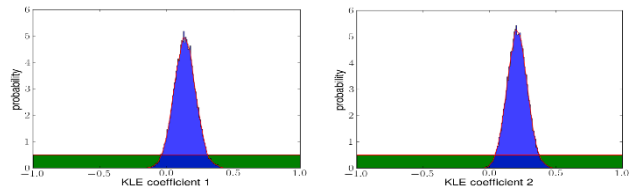
UQ Workflow

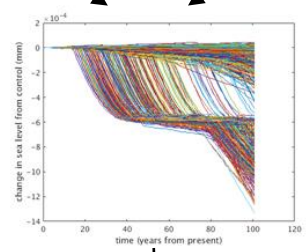
Stage 1:
Estimate ice sheet initial condition (MAP point).

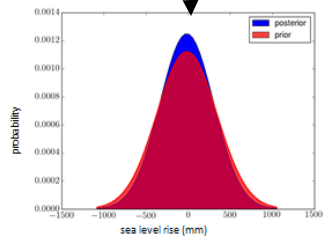
Stage 2:
Update prior uncertainty in ice sheet initial condition using observational data and steady state model

Stage 3:
Propagate uncertain initial condition through ice-sheet evolution model

Goal: propagate PDF obtained in Bayesian inference through model to get distributions on ***total ice mass loss/gain during 21st century***







β PDFs (from Bayesian inference)

SLR(t) for ensemble of forward runs with β sampled from its PDF

PDF of SLR

Appendix: MPI+X FEA via *Kokkos*

- ***MPI-only*** nested for loop:

```
for (int cell=0; cell<numCells; ++cell)
   for (int node=0; node<numNodes; ++node)
      for (int qp=0; qp<numQPs; ++qp)
         compute A;
```

MPI process *n*

Appendix: MPI+X FEA via *Kokkos*

- ***Multi-dimensional parallelism*** for nested for loops via *Kokkos:*

```
for (int cell=0; cell<numCells; ++cell)
   for (int node=0; node<numNodes; ++node)
      for (int qp=0; qp<numQPs; ++qp)
         compute A;
```

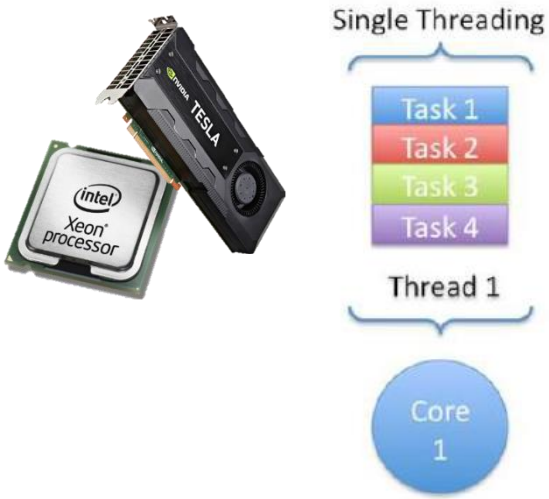
MPI process *n*

Thread *1* computes A for (cell,node,qp)=(0,0,0)

Thread *2* computes A for (cell,node,qp)=(0,0,1)

⋮

Thread *N* computes A for (cell,node,qp)=(numCells,numNodes,numQPs)



Appendix: MPI+X FEA via *Kokkos*

- ***Multi-dimensional parallelism*** for nested for loops via *Kokkos:*

```
for (int cell=0; cell<numCells; ++cell)
  for (int node=0; node<numNodes; ++node)
    for (int qp=0; qp<numQPs; ++qp)
      compute A;
```

MPI process *n*

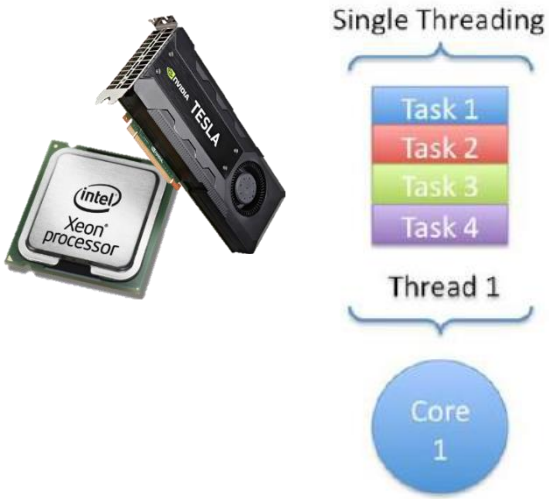
Thread *1* computes A for (cell,node,qp)=(0,0,0)

Thread *2* computes A for (cell,node,qp)=(0,0,1)

⋮

Thread *N* computes A for (cell,node,qp)=(numCells,numNodes,numQPs)

```
computeA_Policy range({0,0,0},{(int)numCells,(int)numNodes,(int)numQPs});
Kokkos::Experimental::md_parallel_for<ExecutionSpace>(range,*this);
```



* Unified Virtual Memory.

Appendix: MPI+X FEA via *Kokkos*

- ***Multi-dimensional parallelism*** for nested for loops via *Kokkos:*

```
for (int cell=0; cell<numCells; ++cell)
  for (int node=0; node<numNodes; ++node)
    for (int qp=0; qp<numQPs; ++qp)
      compute A;
```

MPI process *n*

Thread *1* computes A for (cell,node,qp)=(0,0,0)

Thread *2* computes A for (cell,node,qp)=(0,0,1)

⋮

Thread *N* computes A for (cell,node,qp)=(numCells,numNodes,numQPs)

```
computeA_Policy range({0,0,0},{(int)numCells,(int)numNodes,(int)numQPs});
Kokkos::Experimental::md_parallel_for<ExecutionSpace>(range,*this);
```

- ExecutionSpace defined at ***compile time***, e.g.

```
typedef Kokkos::OpenMP ExecutionSpace; //MPI+OpenMP
typedef Kokkos::CUDA ExecutionSpace; //MPI+CUDA
typedef Kokkos::Serial ExecutionSpace; //MPI-only
```

Appendix: MPI+X FEA via *Kokkos*

- ***Multi-dimensional parallelism*** for nested for loops via *Kokkos:*

```
for (int cell=0; cell<numCells; ++cell)
  for (int node=0; node<numNodes; ++node)
    for (int qp=0; qp<numQPs; ++qp)
      compute A;
```

MPI process *n*

Thread *1* computes A for (cell,node,qp)=(0,0,0)

Thread *2* computes A for (cell,node,qp)=(0,0,1)

⋮

Thread *N* computes A for (cell,node,qp)=(numCells,numNodes,numQPs)

```
computeA_Policy range({0,0,0},{(int)numCells,(int)numNodes,(int)numQPs});
Kokkos::Experimental::md_parallel_for<ExecutionSpace>(range,*this);
```

- ExecutionSpace defined at ***compile time***, e.g.

```
typedef Kokkos::OpenMP ExecutionSpace; //MPI+OpenMP
typedef Kokkos::CUDA ExecutionSpace; //MPI+CUDA
typedef Kokkos::Serial ExecutionSpace; //MPI-only
```

- ***Atomics*** used to scatter local data to global data structures

```
Kokkos::atomic_fetch_add
```

Appendix: MPI+X FEA via *Kokkos*

- ***Multi-dimensional parallelism*** for nested for loops via *Kokkos:*

```
for (int cell=0; cell<numCells; ++cell)
  for (int node=0; node<numNodes; ++node)
    for (int qp=0; qp<numQPs; ++qp)
      compute A;
```

MPI process *n*

Thread *1* computes A for (cell,node,qp)=(0,0,0)

Thread *2* computes A for (cell,node,qp)=(0,0,1)

⋮

Thread *N* computes A for (cell,node,qp)=(numCells,numNodes,numQPs)

```
computeA_Policy range({0,0,0},{(int)numCells,(int)numNodes,(int)numQPs});
Kokkos::Experimental::md_parallel_for<ExecutionSpace>(range,*this);
```

- ExecutionSpace defined at ***compile time***, e.g.

```
typedef Kokkos::OpenMP ExecutionSpace; //MPI+OpenMP
typedef Kokkos::CUDA ExecutionSpace; //MPI+CUDA
typedef Kokkos::Serial ExecutionSpace; //MPI-only
```

- ***Atomics*** used to scatter local data to global data structures

```
Kokkos::atomic_fetch_add
```

- For MPI+CUDA, data transfer from host to device handled by ***CUDA UVM****.

* Unified Virtual Memory.

Appendix: MPI+X FEA via *Kokkos*

- ***Multi-dimensional parallelism*** for nested for loops via *Kokkos:*

```
for (int cell=0; cell<numCells; ++cell)
  for (int node=0; node<numNodes; ++node)
    for (int qp=0; qp<numQPs; ++qp)
      compute A;
```

MPI process *n*

Kokkos parallelization in FELIX is only over **cells**.

Thread *1* computes A for (cell,node,qp)=(0,0,0)

Thread *2* computes A for (cell,node,qp)=(0,0,1)

⋮

Thread *N* computes A for (cell,node,qp)=(numCells,numNodes,numQPs)

```
computeA_Policy range({0,0,0},{(int)numCells,(int)numNodes,(int)numQPs});
Kokkos::Experimental::md_parallel_for<ExecutionSpace>(range,*this);
```

- ExecutionSpace defined at ***compile time***, e.g.

```
typedef Kokkos::OpenMP ExecutionSpace; //MPI+OpenMP
typedef Kokkos::CUDA ExecutionSpace; //MPI+CUDA
typedef Kokkos::Serial ExecutionSpace; //MPI-only
```

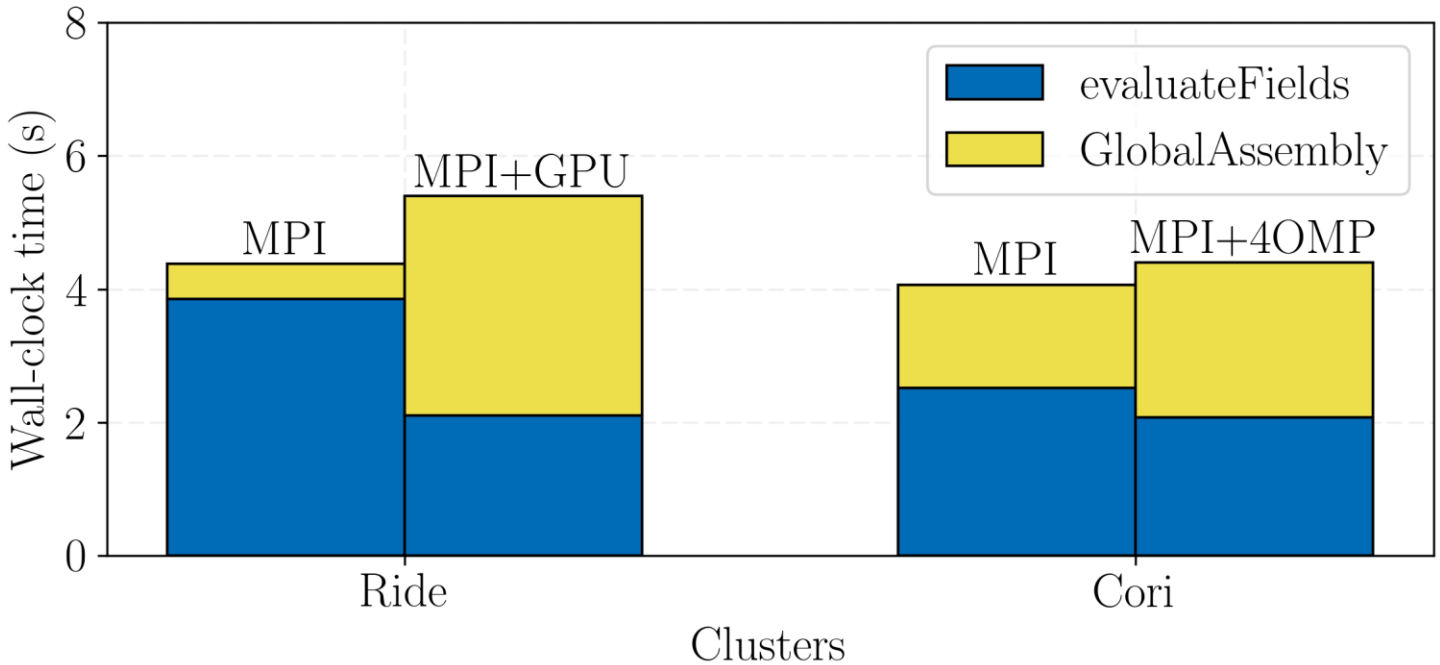
- ***Atomics*** used to scatter local data to global data structures

```
Kokkos::atomic_fetch_add
```

- For MPI+CUDA, data transfer from host to device handled by ***CUDA UVM****.

* Unified Virtual Memory.

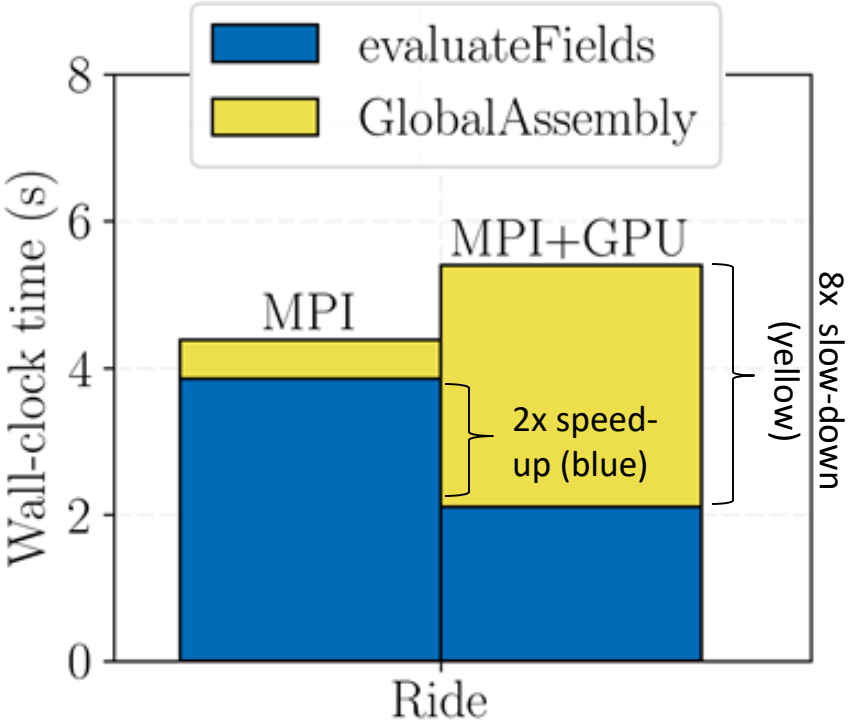
Appendix: GIS Kokkos Execution Space Comparison



<u>A single node comparison</u>: 16MPI vs. 4(MPI+GPU) [*Left*], 68MPI vs. 68(MPI+4OMP) [*Right*]
(GIS 4km-20km unstructured mesh with 1.51M tet elements)

Blue (evaluateFields): mostly Residual/Jacobian computation + Gather/Scatter (Albany/FELIX);
Yellow (GlobalAssembly): mostly communication + Trilinos operations.

Appendix: GIS Kokkos 16 MPI vs. 4(MPI+GPU)



GlobalAssembly (yellow): mostly communication + Trilinos operations

- 8x slow-down is not reasonable.
- Most slow-downs are in Trilinos (Tpetra) → Trilinos packages are currently being reworked using CUDA-aware MPI*.

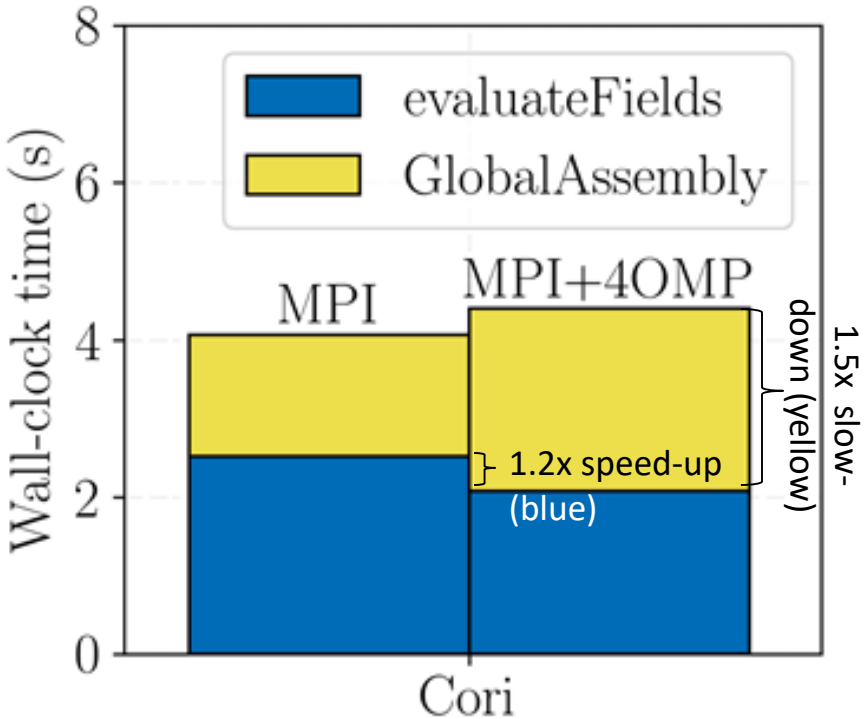
Summary: speed-ups on GPU are not yet as expected but may be improved by introducing padding, removing CUDA UVM and unnecessary data movement, switching to CUDA-aware MPI.

evaluateFields (blue): mostly residual/Jacobian computation + Gather/Scatter

- evaluateFields<Jacobian> much faster than evaluateFields<Residual> b/c there is more work in computing Jacobian.
- 2x speedup not much considering 4 GPUs (desirable speedup: 10x or more).
  - Data movement is lagging speedup – can be improved by removing CUDA UVM, data padding to prevent data misalignment.
  - A few kernels still for boundary conditions still need to be ported to Kokkos.

* With CUDA-aware MPI, GPU buffers can be passed directly to MPI w/o staging using cudaMemcpy.

Appendix: GIS Kokkos 68MPI vs. 68(MPI+4OMP)



<u>GlobalAssembly (yellow)</u>: mostly communication + Trilinos operations

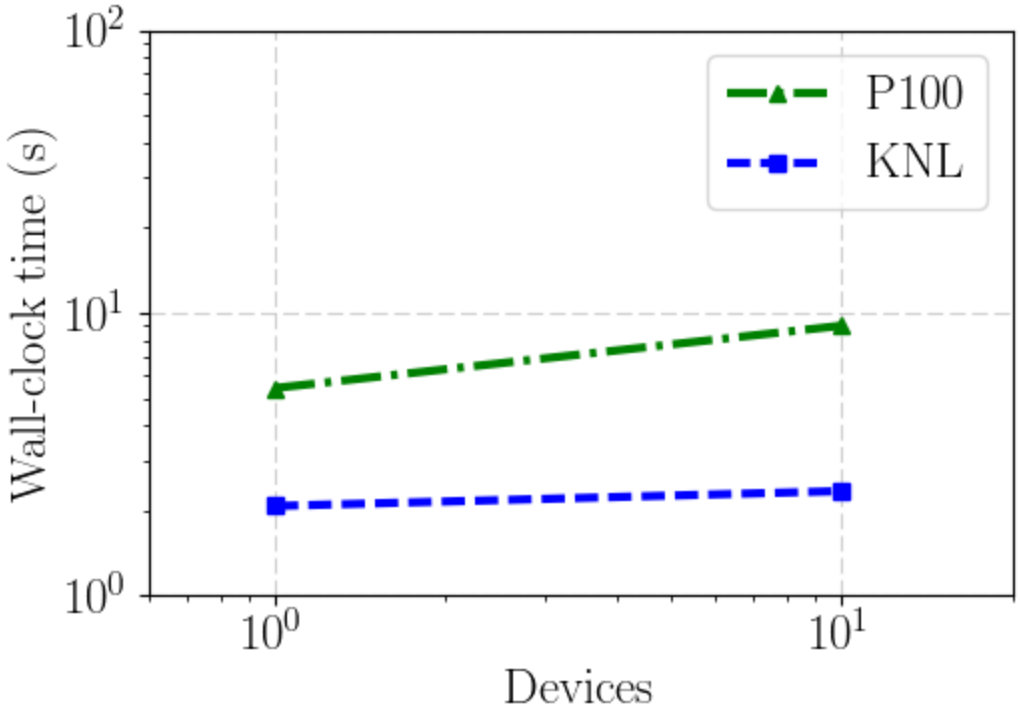
- 1.5x slowdown is in Jacobian assembly not Residual assembly → Trilinos team is investigating this now.

Summary: besides the global Jacobian assembly, results are promising. More studies are needed once we profile nonlinear/linear solvers.

<u>evaluateFields (blue)</u>: mostly residual/Jacobian computation + Gather/Scatter

- 1.2x speedup from hardware threads is reasonable (there are 2 VPUs/KNL core, so 2x speedup is ideal but will be limited by L1 cache size in core for bandwidth bound operation)
- Once the nonlinear/linear solver is included, more OpenMP threads will likely be used (e.g. 4(MPI+68OMP)) to improve speedup.
  - More OpenMP threads on cores in FEA reduces performance because it takes away from coarser grain MPI parallelism

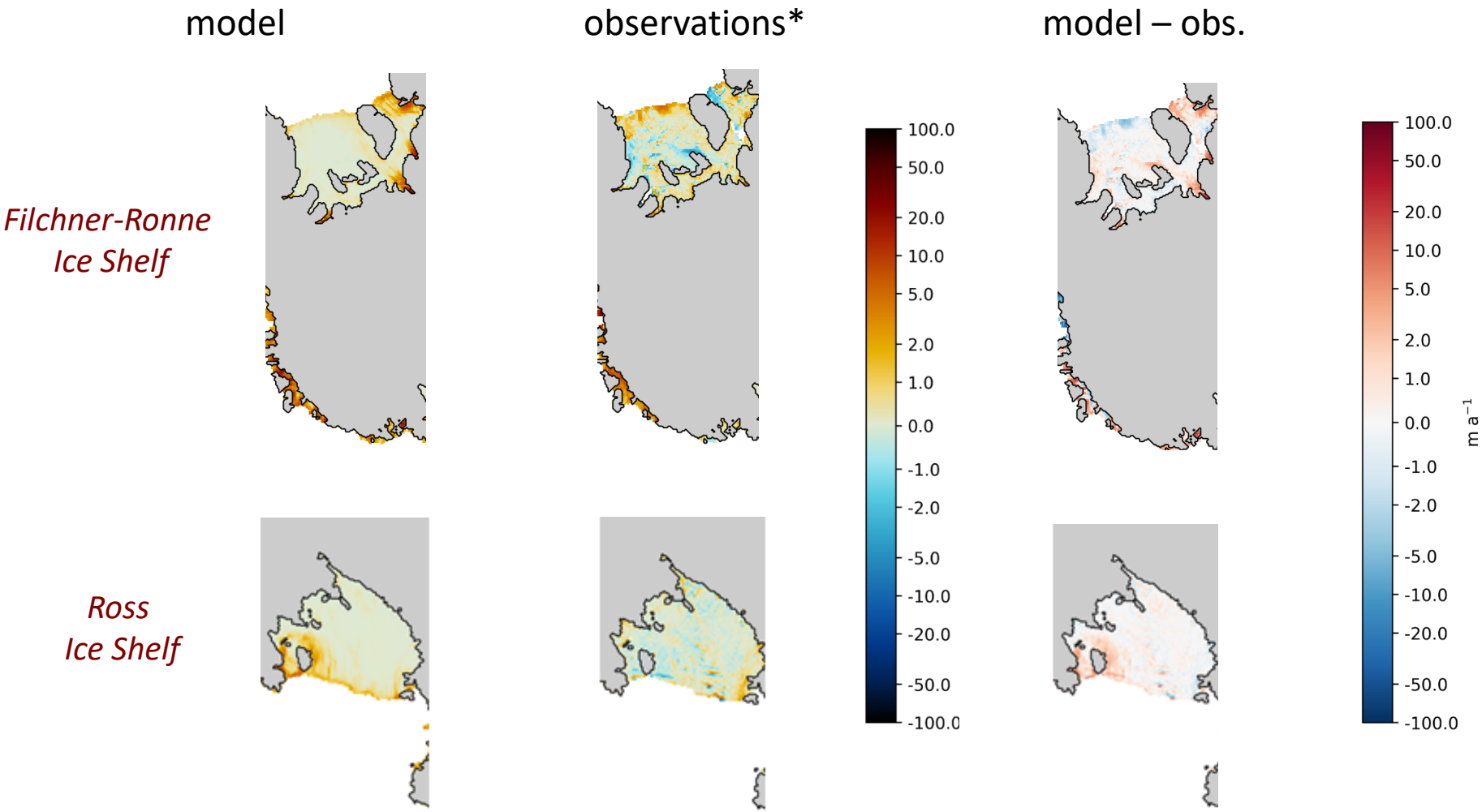
Appendix: GIS MPI+Device Weak Scalability



- Scalability of **GlobalAssembly + evaluateFields** is studied.
- **Weak Scalability** is for GIS 4km-20km and 1km-7km (1.51M and 14.4M elements) tet meshes.
- **Weak scalability** is comparable for P100 and KNL.
  - KNL performs **better** because of heavy use of MPI.
- Optimizations/profiling required for **strong scalability** of FELIX; we have demonstrated strong scalability for other applications in Albany (I. Demeshko *et al* 2017).

Appendix: PISCEES & E3SM Coupling Validation

Sub-shelf melt rates (RRS30to10km resolution)



* Rignot et al., *Science*, 2013