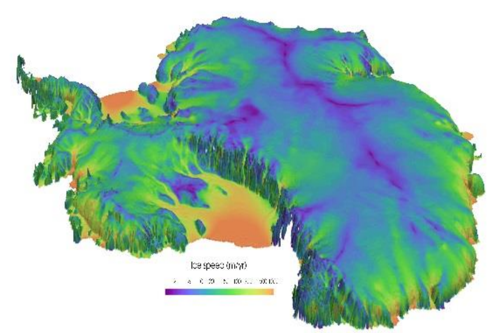
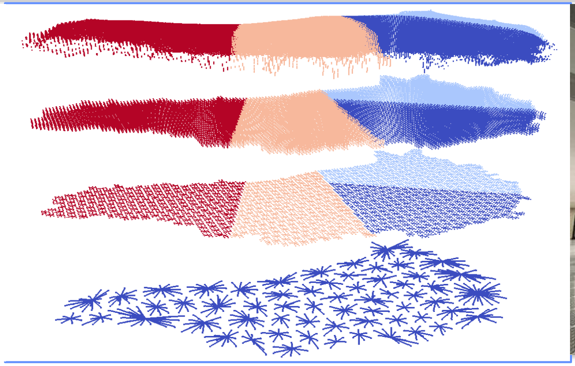
Exceptional service in the national interest

Sandia National Laboratories





Mathematical Modeling of the Polar Ice Sheets

Irina K. Tezaur, *et al.*

Quantitative Modeling and Analysis Department
Sandia National Laboratories, Livermore, CA

Guest Lecture

Federal University of Paraná, Brazil

June 22, 2021

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.

Acknowledgements

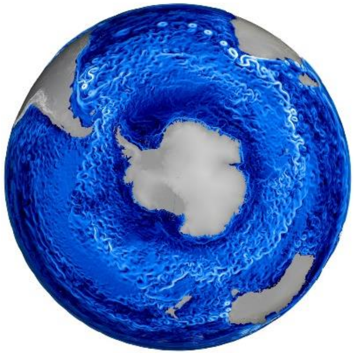
In collaboration with: ***L. Bertagna*, *M. Carlson***, *J. Jakeman, K. Liegeois*, ***M. Perego*, *A. Salinger***, *C. Sockwell*, ***R. Tuminaro, J. Watkins*** *[SNL];* ***I. Demeshko***, *T. Hillebrand*, ***M. Hoffman, S. Price, T. Zhang*** *[LANL]; K. Evans [ORNL]; J. Kennedy [UAF]; G. Stadler [NYU]; J. Bassis [U Michigan];* ***K. Shan*** *[Micron Technology]*

Computing resources: *NERSC, OLCF.*

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.**

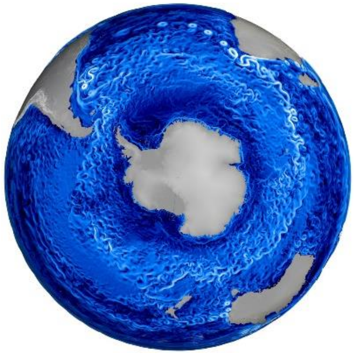
Outline

1. Background
   - Motivation for climate & land-ice modeling
   - ISMs, ESMs & projects
   - Land-ice equations
   - Our codes: ALI, MALI
2. Algorithms and software
   - Discretization & meshes
   - Nonlinear solvers
   - Linear solvers
   - Performance-portability
   - Ice sheet initialization
   - Towards UQ
3. Simulations
4. Summary



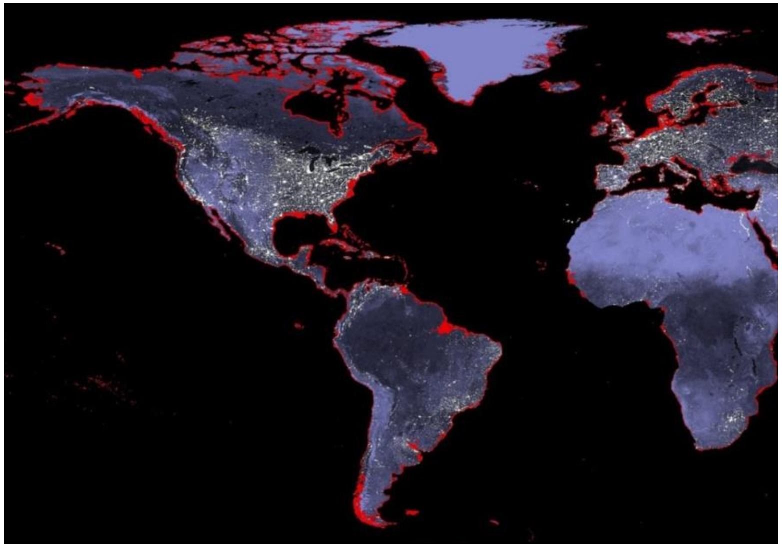
Outline

1. **Background**
   - **Motivation for climate & land-ice modeling**
   - ISMs, ESMs & projects
   - Land-ice equations
   - Our codes: ALI, MALI
2. Algorithms and software
   - Discretization & meshes
   - Nonlinear solvers
   - Linear solvers
   - Performance-portability
   - Ice sheet initialization
   - Towards UQ
3. Simulations
4. Summary

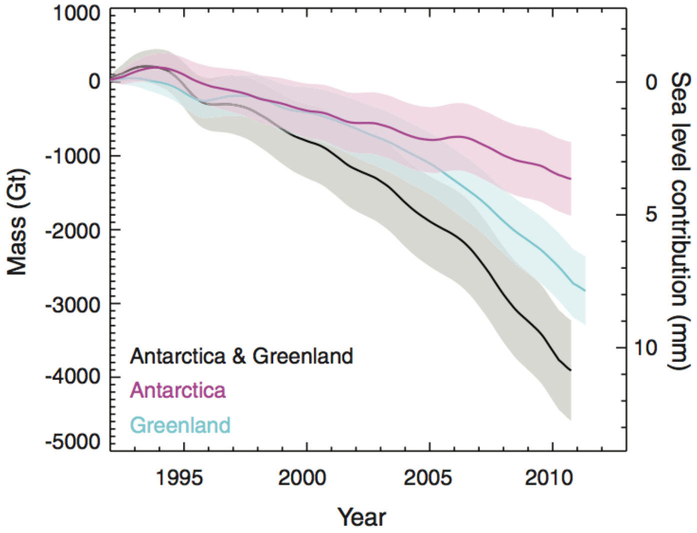


Motivation

- Climate change is a **global threat** to health, global security, infrastructure, ...
- Global mean sea-level is rising at the rate of **3.2 mm/year** and this rate is **increasing**, with the latest studies suggesting a possible increase in sea-level of **0.3-2.5 m** by 2100.
  - Due to **melting of the polar ice sheets** (Greenland, Antarctica).
- **Full deglaciation***: sea level could rise up to ~65 m (Antarctica: 58 m, Greenland: 7 m)



Map showing 6 m sea-level rise (NASA)

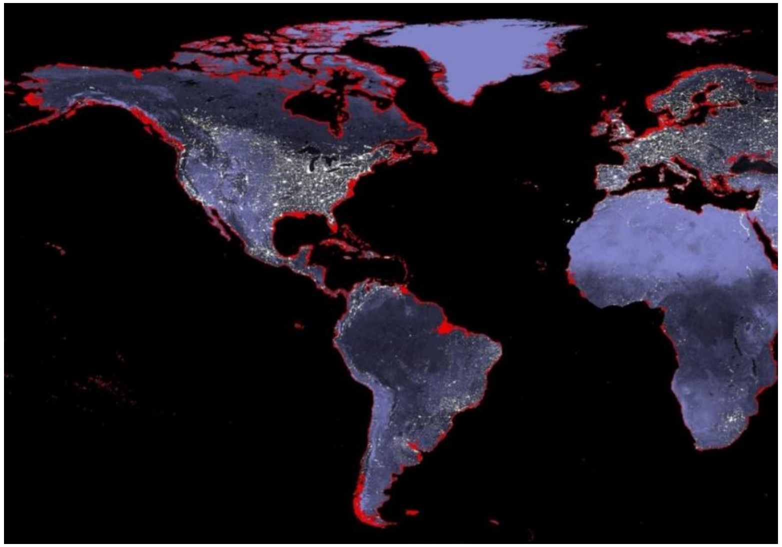


Total mass loss of ice sheets b/w 1992-2011

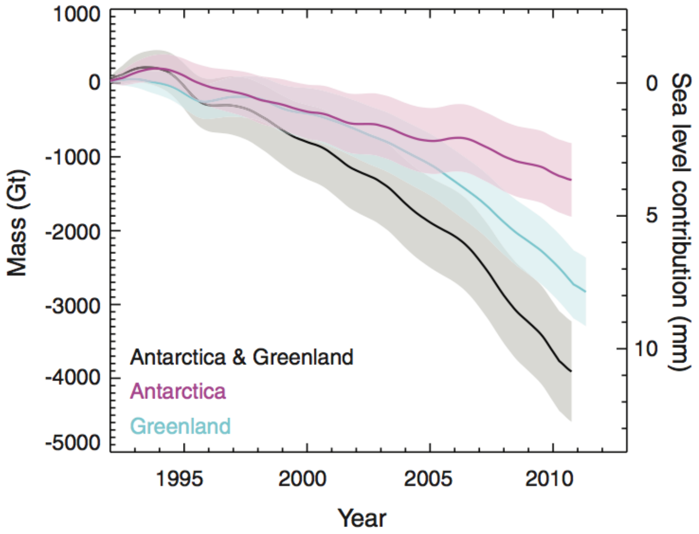
*Estimates given by Prof. Richard Alley of Penn State.

Motivation

- Climate change is a **global threat** to health, global security, infrastructure, ...
- Global mean sea-level is rising at the rate of **3.2 mm/year** and this rate is **increasing**, with the latest studies suggesting a possible increase in sea-level of **0.3-2.5 m** by 2100.
  - Due to **melting of the polar ice sheets** (Greenland, Antarctica).
- **Full deglaciation***: sea level could rise up to ~65 m (Antarctica: 58 m, Greenland: 7 m)



Map showing 6 m sea-level rise (NASA)



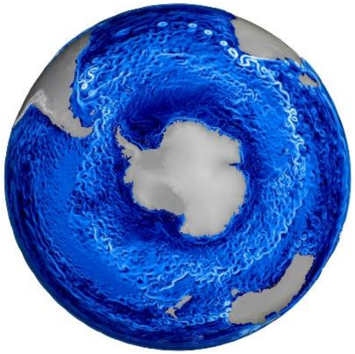
Total mass loss of ice sheets b/w 1992-2011

Modeling of ice sheet dynamics is **essential** for providing estimates of **sea-level rise**, towards understanding the local/global effects of **climate change**.

*Estimates given by Prof. Richard Alley of Penn State.

Outline

1. **Background**
   - Motivation for climate & land-ice modeling
   - **ISMs, ESMs & projects**
   - Land-ice equations
   - Our codes: ALI, MALI
2. Algorithms and software
   - Discretization & meshes
   - Nonlinear solvers
   - Linear solvers
   - Performance-portability
   - Ice sheet initialization
   - Towards UQ
3. Simulations
4. Summary



What is an Ice Sheet Model (ISM)?

Dynamical core ("dycore")

Conservation of:

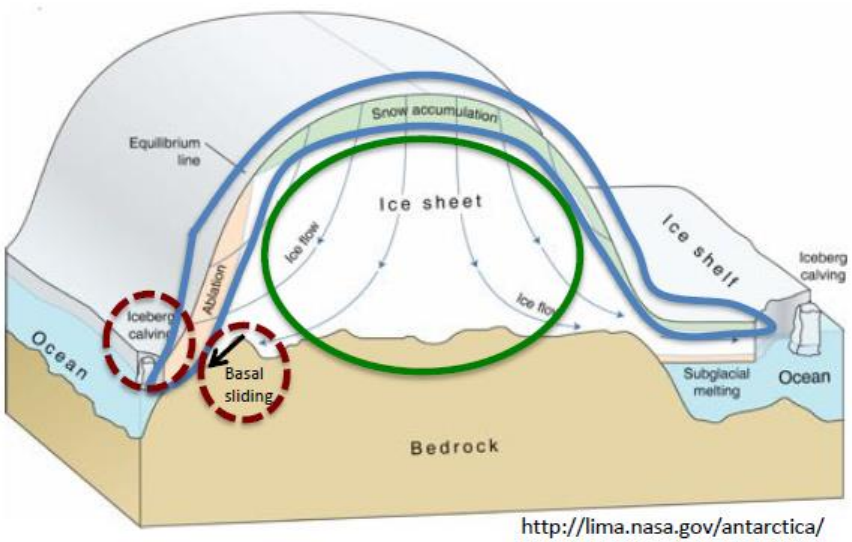
- Mass (ice thickness)
- Momentum (ice velocity)
- Energy (ice temperature)

Physical processes ("physics")

- Iceberg calving
- Basal sliding
- Etc…

Climate forcing

- Snowfall/melt
- Ocean melting/freezing
- Etc…



http://lima.nasa.gov/antarctica/

What is an Ice Sheet Model (ISM)?

Dynamical core ("dycore")

Conservation of:

- Mass (ice thickness)
- Momentum (ice velocity)
- Energy (ice temperature)

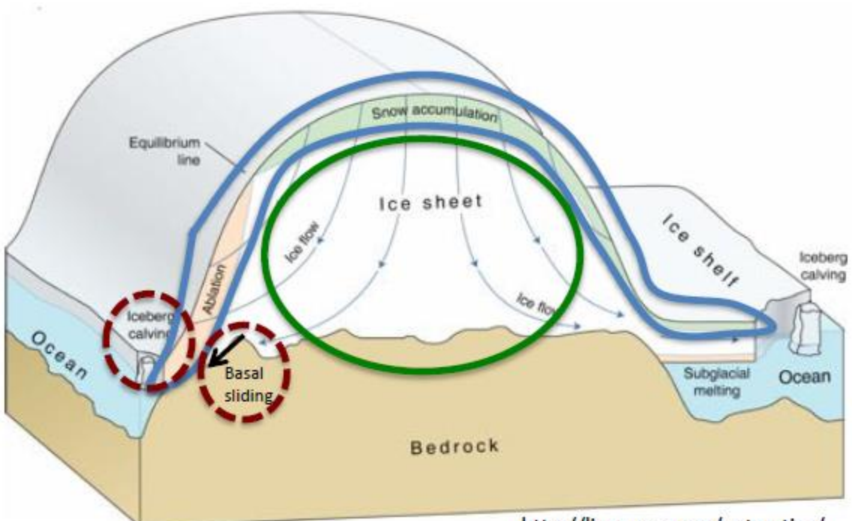
Physical processes ("physics")

- Iceberg calving
- Basal sliding
- Etc…

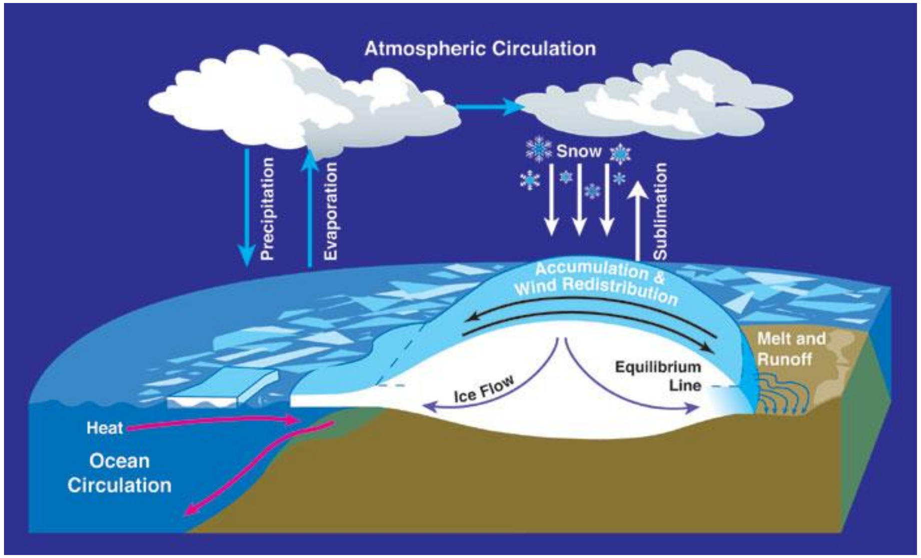
Climate forcing

- Snowfall/melt
- Ocean melting/freezing
- Etc…

Comes from ***Earth System Model (ESM)***



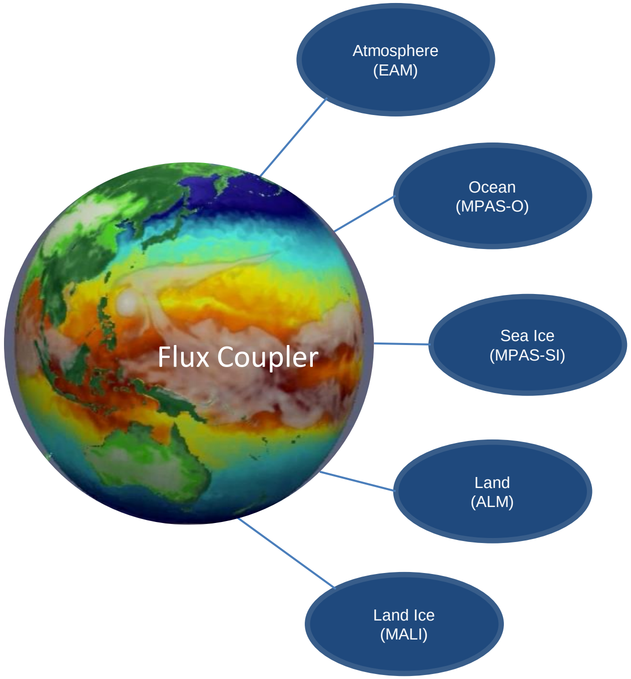
http://lima.nasa.gov/antarctica/



Earth System Models (ESMs)

Sandia National Laboratories

An Earth System Model (ESM) has ***six modular components***:



CESM
COMMUNITY EARTH SYSTEM MODEL

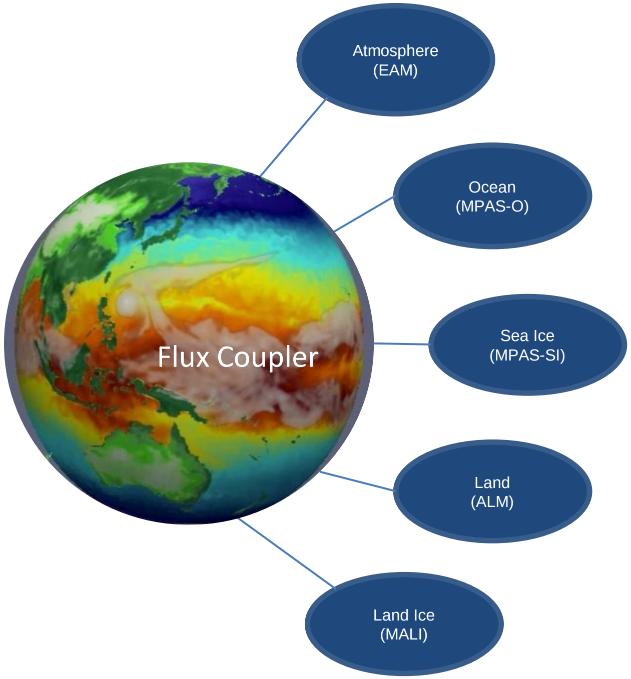
E3SM
Energy Exascale
Earth System Model

NorESM

EC EARTH

Earth System Models (ESMs)

An Earth System Model (ESM) has ***six modular components***:



Role of an ISM within an ESM: to provide actionable scientific predictions of 21st century sea-level change (including uncertainty bounds).

CESM

COMMUNITY EARTH SYSTEM MODEL

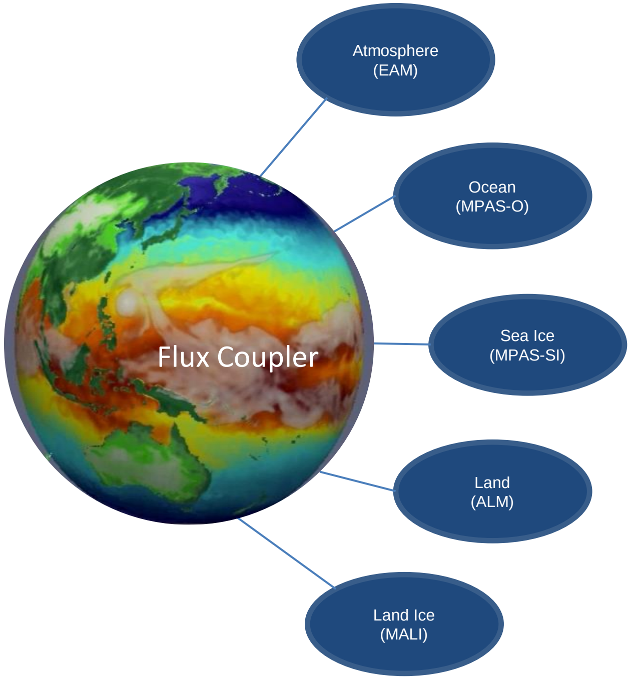
E3SM Energy Exascale Earth System Model

NorESM

EC EARTH

Earth System Models (ESMs)

An Earth System Model (ESM) has ***six modular components***:



Role of an ISM within an ESM: to provide actionable scientific predictions of 21st century sea-level change (including uncertainty bounds).

About a decade ago, existing land-ice models were **not robust enough** for ESM integration! ☹

CESM
COMMUNITY EARTH SYSTEM MODEL

E3SM
Energy Exascale Earth System Model

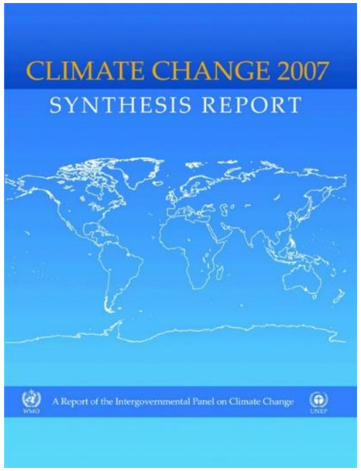
NorESM

EC EARTH

U.S. DOE Ice Sheet/Climate Model Efforts

Motivation:

- 2007 IPCC (Intergovernmental Panel on Climate Change) Fourth Assessment Report ***declined*** to include estimates of future sea-level rise from ice sheet dynamics due to the ***inability*** of ice sheet models to mimic/explain observed dynamic behaviors.
  - ***"Much work is needed*** to make [present-day ISMs] robust and efficient on continental scales and to quantify uncertainties in their projected outputs". – IPCC AR4 (2007)



CLIMATE CHANGE 2007

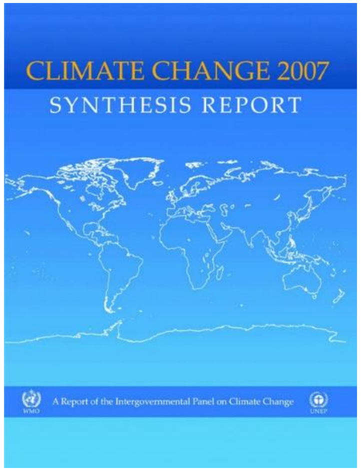
SYNTHESIS REPORT

WMO A Report of the Intergovernmental Panel on Climate Change UNEP

U.S. DOE Ice Sheet/Climate Model Efforts

Motivation:

- 2007 IPCC (Intergovernmental Panel on Climate Change) Fourth Assessment Report ***declined*** to include estimates of future sea-level rise from ice sheet dynamics due to the ***inability*** of ice sheet models to mimic/explain observed dynamic behaviors.
  - ***"Much work is needed*** to make [present-day ISMs] robust and efficient on continental scales and to quantify uncertainties in their projected outputs". – IPCC AR4 (2007)



CLIMATE CHANGE 2007

SYNTHESIS REPORT

A Report of the Intergovernmental Panel on Climate Change

U.S. DOE-funded Land-Ice Modeling Projects:

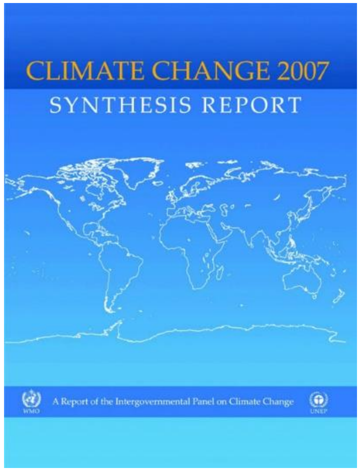
- *Predicting Ice Sheet & Climate Evolution at Extreme Scales* (**PISCEES**): 2012-2017.
- *Probabilistic Sea-Level Projections from Ice Sheet Models and ESMs* (**ProSPect**): 2017-2022.

Aim is to ***develop & apply robust, accurate, scalable*** dynamical cores for ice sheet modeling on ***unstructured*** meshes, enable ***uncertainty quantification*** (UQ), and ***integrate*** models/tools into DOE E3SM

U.S. DOE Ice Sheet/Climate Model Efforts

Motivation:

- 2007 IPCC (Intergovernmental Panel on Climate Change) Fourth Assessment Report ***declined*** to include estimates of future sea-level rise from ice sheet dynamics due to the ***inability*** of ice sheet models to mimic/explain observed dynamic behaviors.
    - ➢ ***"Much work is needed*** to make [present-day ISMs] robust and efficient on continental scales and to quantify uncertainties in their projected outputs". – IPCC AR4 (2007)



U.S. DOE-funded Land-Ice Modeling Projects:

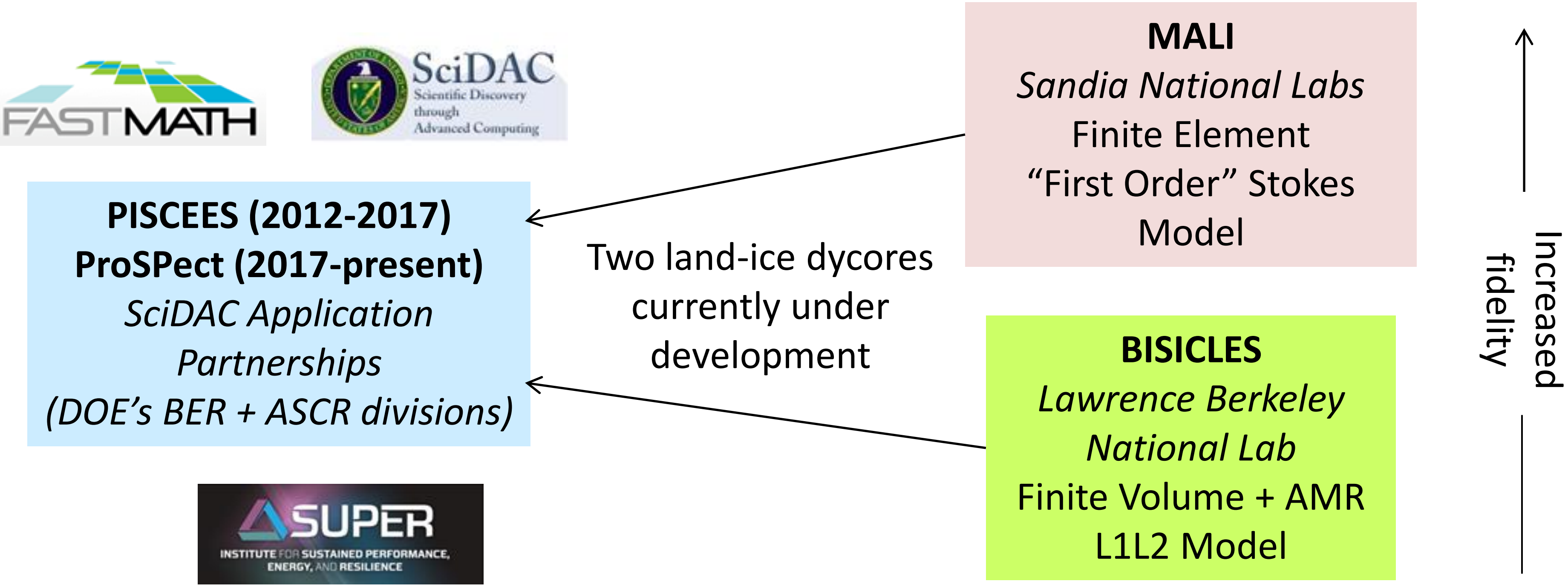
- *Predicting Ice Sheet & Climate Evolution at Extreme Scales* (**PISCEES**): 2012-2017.
- *Probabilistic Sea-Level Projections from Ice Sheet Models and ESMs* (**ProSPect**): 2017-2022.

Aim is to ***develop & apply robust, accurate, scalable*** dynamical cores for ice sheet modeling on ***unstructured*** meshes, enable ***uncertainty quantification*** (UQ), and ***integrate*** models/tools into DOE E3SM

U.S. DOE Energy Exascale Earth System Model (E3SM):

- "Next-generation" climate model with focus of ***decadal-century timescale projections, high-spatial resolution,*** next ***generation HPC,*** impacts to ***U.S. infrastructure.***

The PISCEES & ProSPect Projects



PISCEES (2012-2017)
ProSPect (2017-present)
SciDAC Application Partnerships
(DOE's BER + ASCR divisions)

Two land-ice dycores currently under development

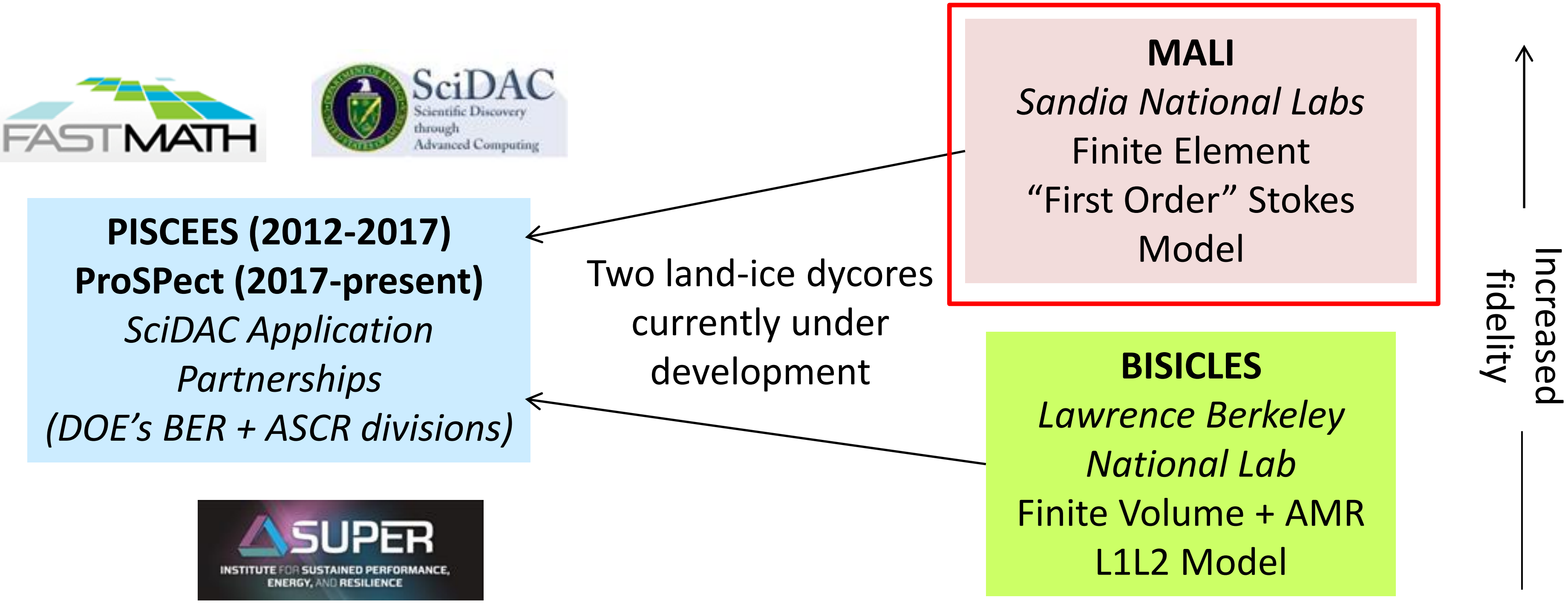
MALI
Sandia National Labs
Finite Element
"First Order" Stokes Model

BISICLES
Lawrence Berkeley National Lab
Finite Volume + AMR
L1L2 Model

Increased fidelity

MALI: MPAS-Albany Land Ice
BISICLES: Berkeley Ice Sheet Initiative for Climate at Extreme Scales

The PISCEES & ProSPect Projects



MALI
Sandia National Labs
Finite Element
"First Order" Stokes
Model

PISCEES (2012-2017)
ProSPect (2017-present)
SciDAC Application Partnerships
(DOE's BER + ASCR divisions)

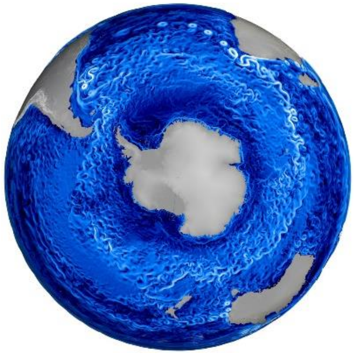
Two land-ice dycores currently under development

BISICLES
Lawrence Berkeley National Lab
Finite Volume + AMR
L1L2 Model

MALI: MPAS-Albany Land Ice
BISICLES: Berkeley Ice Sheet Initiative for Climate at Extreme Scales

Outline

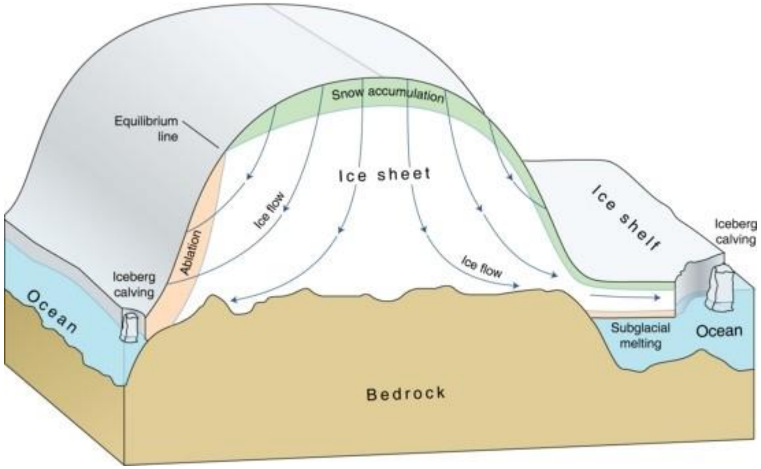
1. **Background**
   - Motivation for climate & land-ice modeling
   - ISMs, ESMs & projects
   - **Land-ice equations**
   - Our codes: ALI, MALI
2. Algorithms and software
   - Discretization & meshes
   - Nonlinear solvers
   - Linear solvers
   - Performance-portability
   - Ice sheet initialization
   - Towards UQ
3. Simulations
4. Summary



Stokes Ice Flow Equations

Ice behaves like a **very viscous non-Newtonian shear-thinning fluid** (like lava flow) and is modeled **quasi-statically** using **nonlinear incompressible Stokes equations**.

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



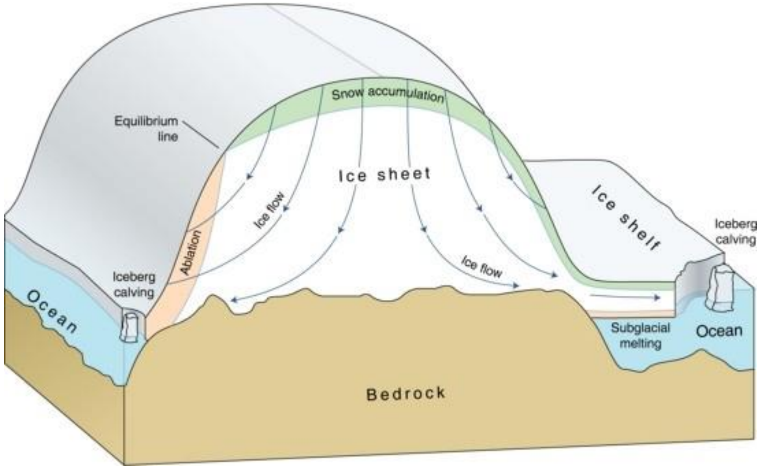
- Fluid velocity vector: $\boldsymbol{u} = (u_1, u_2, u_3)$
- Isotropic ice pressure: p
- Deviatoric stress tensor: $\boldsymbol{\tau} \; = \; 2\mu\boldsymbol{\epsilon}$
- Strain rate tensor: $\epsilon_{\mathrm{ij}} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$
- Glen's Law Viscosity*: $\mu = \frac{1}{2}A(T)^{-\frac{1}{n}}\left(\frac{1}{2}\sum_{ij}\boldsymbol{\epsilon}_{ij}{}^{2}\right)^{\left(\frac{1}{2n}-\frac{1}{2}\right)}$
- Flow factor: $A(T) = A_0 e^{-\frac{Q}{RT}}$

*Nye 1957; Cuffey *et* al., 2010. Typically we use $n = 3$.

Stokes Ice Flow Equations

Ice behaves like a **very viscous non-Newtonian shear-thinning fluid** (like lava flow) and is modeled **quasi-statically** using **nonlinear incompressible Stokes equations**.

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



- Fluid velocity vector: $\boldsymbol{u} = (u_1, u_2, u_3)$
- Isotropic ice pressure: p
- Deviatoric stress tensor: $\boldsymbol{\tau} = 2\mu\boldsymbol{\epsilon}$
- Strain rate tensor: $\epsilon_{\mathrm{ij}} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$
- Glen's Law Viscosity*: $\mu = \frac{1}{2} A(T)^{-\frac{1}{n}} \left(\frac{1}{2}\sum_{ij} \boldsymbol{\epsilon}_{ij}{}^{2}\right)^{\left(\frac{1}{2n}-\frac{1}{2}\right)}$
- Flow factor: $A(T) = A_0 e^{-\frac{Q}{RT}}$

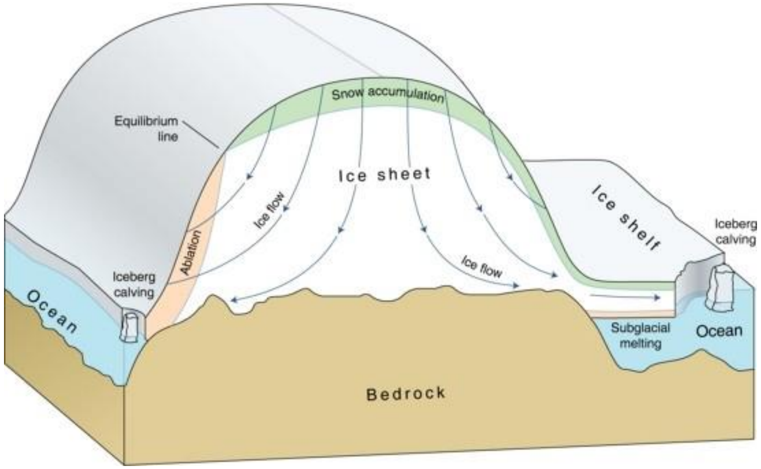
Highly nonlinear rheology!

*Nye 1957; Cuffey *et* al., 2010. Typically we use $n = 3$.

Stokes Ice Flow Equations

Ice behaves like a **very viscous non-Newtonian shear-thinning fluid** (like lava flow) and is modeled **quasi-statically** using **nonlinear incompressible Stokes equations**.

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



- Fluid velocity vector: $\boldsymbol{u} = (u_1, u_2, u_3)$
- Isotropic ice pressure: p
- Deviatoric stress tensor: $\boldsymbol{\tau} = 2\mu\boldsymbol{\epsilon}$
- Strain rate tensor: $\epsilon_{\text{ij}} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$
- Glen's Law Viscosity*: $\mu = \frac{1}{2} A(T)^{-\frac{1}{n}} \left(\frac{1}{2}\sum_{ij} \boldsymbol{\epsilon}_{ij}{}^2\right)^{\left(\frac{1}{2n} - \frac{1}{2}\right)}$
- Flow factor: $A(T) = A_0 e^{-\frac{Q}{RT}}$

☺ "Gold standard" model

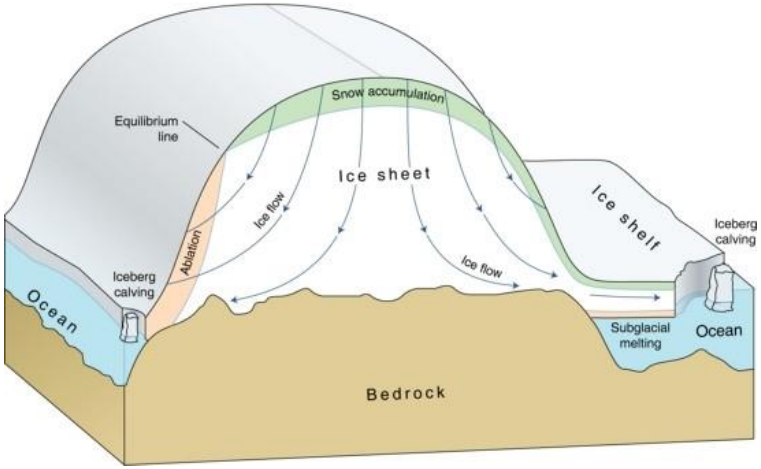
Highly nonlinear rheology!

*Nye 1957; Cuffey *et* al., 2010. Typically we use $n = 3$.

Stokes Ice Flow Equations

Ice behaves like a **very viscous non-Newtonian shear-thinning fluid** (like lava flow) and is modeled **quasi-statically** using **nonlinear incompressible Stokes equations**.

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



- Fluid velocity vector: $\boldsymbol{u} = (u_1, u_2, u_3)$
- Isotropic ice pressure: p
- Deviatoric stress tensor: $\boldsymbol{\tau} \ = \ 2\mu\boldsymbol{\epsilon}$
- Strain rate tensor: $\epsilon_{\mathrm{ij}} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$
- Glen's Law Viscosity*: $\mu = \frac{1}{2} A(T)^{-\frac{1}{n}} \left(\frac{1}{2}\sum_{ij} \boldsymbol{\epsilon}_{ij}{}^2\right)^{\left(\frac{1}{2n} - \frac{1}{2}\right)}$
- Flow factor: $A(T) = A_0 e^{-\frac{Q}{RT}}$

Highly nonlinear rheology!

☺ "Gold standard" model

☹ …but very expensive!

*Nye 1957; Cuffey *et* al., 2010. Typically we use $n = 3$.

First Order (FO) Stokes/Blatter-Pattyn Model*

Stokes$(\boldsymbol{u}, p)$ in $\Omega \in \mathbb{R}^3$

$$\boldsymbol{u} \equiv (u, v, w)$$

$$\boldsymbol{\epsilon}(\boldsymbol{u}) = \begin{pmatrix} u_x & \frac{1}{2}(u_y + v_x) & \frac{1}{2}(u_z + w_x) \\ \frac{1}{2}(u_y + v_x) & v_y & \frac{1}{2}(v_z + w_y) \\ \frac{1}{2}(u_z + w_x) & \frac{1}{2}(v_z + w_y) & w_z \end{pmatrix}$$

$$p = \rho g(s - z) - 2\mu(u_x + v_y)$$

*Pattyn, 2003; Blatter, 1995.

First Order (FO) Stokes/Blatter-Pattyn Model*

Stokes($\boldsymbol{u}, p$) in $\Omega \in \mathbb{R}^3$

***Hydrostatic approximation + scaling argument** based on the fact that ice sheets are thin and normals are almost vertical*

$$\boldsymbol{u} \equiv (u, v, w)$$

$$\epsilon(\boldsymbol{u}) = \begin{pmatrix} u_x & \frac{1}{2}(u_y + v_x) & \frac{1}{2}(u_z + \cancel{w_x}) \\ \frac{1}{2}(u_y + v_x) & v_y & \frac{1}{2}(v_z + \cancel{w_y}) \\ \frac{1}{2}(u_z + \cancel{w_x}) & \frac{1}{2}(v_z + \cancel{w_y}) & w_z \end{pmatrix}$$

$$p = \rho g(s - z) - 2\mu(u_x + v_y)$$

First Order Stokes (a.k.a. Blatter-Pattyn) Model

<u>FO Stokes($\boldsymbol{u}, \boldsymbol{v}$) in $\boldsymbol{\Omega} \in \mathbb{R}^3$</u>

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

$$\dot{\epsilon}(u, v) = \begin{pmatrix} 2u_x + v_y & \frac{1}{2}(u_y + v_x) & \frac{1}{2}u_z \\ \frac{1}{2}(u_y + v_x) & u_x + 2v_y & \frac{1}{2}v_z \end{pmatrix}$$

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

$$(n = 3)$$

Discussion:

- Nice "**elliptic**" approximation to full Stokes.
- 3D model for two unknowns (u, v) with nonlinear μ.
- Valid for both **Greenland** and **Antarctica** and used in **continental scale** simulations.

*Pattyn, 2003; Blatter, 1995.

Boundary Conditions

Boundary conditions have **tremendous effect** on ice sheet behavior!

Ice-Atmosphere Boundary:

- ***Stress-free BC:*** $2\mu\dot{\boldsymbol{\epsilon}}_i\cdot\boldsymbol{n}=0$ on Γ_s

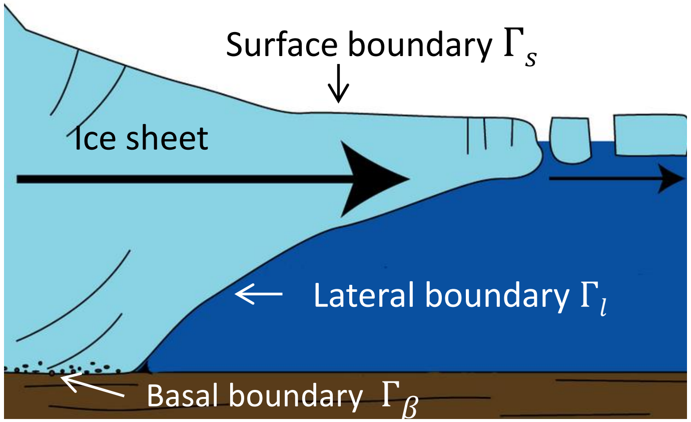
Ice-Bedrock Boundary:

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

β = basal sliding coefficient

$\beta=\beta(x,y)$ or $\beta=\beta(x,y,\boldsymbol{u},t)$

Can't be measured – must be estimated from data!

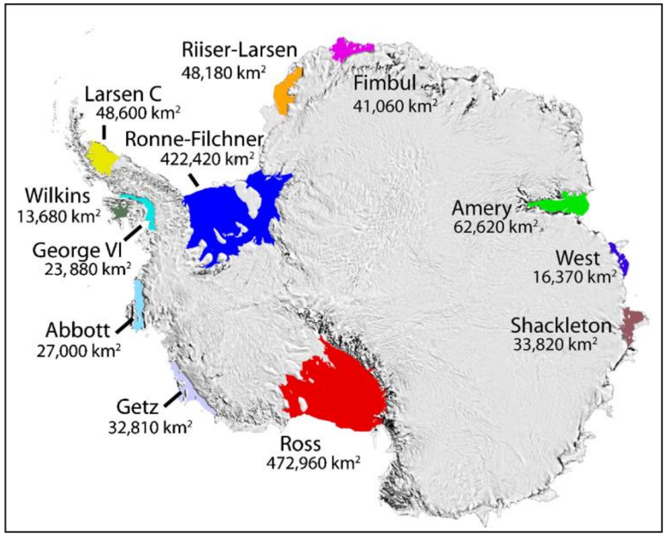

Surface boundary Γ_s
Ice sheet
Lateral boundary Γ_l
Basal boundary Γ_β

Ice-Ocean Boundary:

- ***Floating ice (a.k.a. open ocean) BC:***

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

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]."*



Antarctica's ice shelves shown in color

Ice Sheet Evolution

Ice velocity equations are **coupled** with equations for ice sheet evolution (thickness) and ice temperature.

- **Energy equation** for the temperature T:

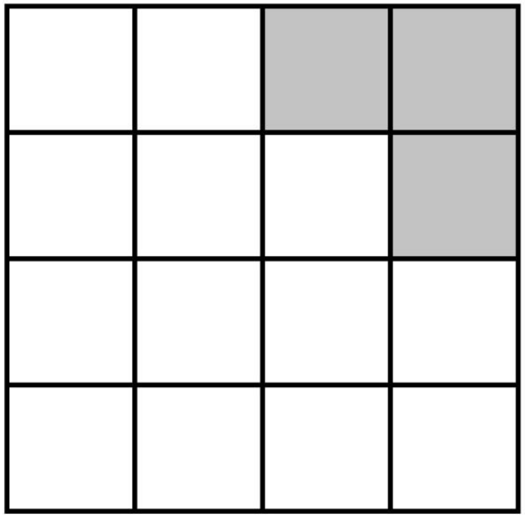
$$\rho c \frac{\partial T}{\partial t} + \rho c \boldsymbol{u} \cdot \nabla T = \nabla \cdot (k \nabla T) + 2\dot{\boldsymbol{\epsilon}}\boldsymbol{\sigma}, \quad \text{in} \quad \Omega_H$$

➢ Flow factor A in Glen's law viscosity μ is function of T.

- **Mass equation** for the ice thickness H:

$$\frac{\partial H}{\partial t} + \nabla \cdot (\overline{\boldsymbol{u}} H) + \dot{b}, \quad \text{on} \quad \Gamma$$

$\overline{\boldsymbol{u}}$ = vertically averaged $\boldsymbol{u}$
$\dot{b}$ = surface mass balance
(given accumulation-ablation function that accounts for e.g. accumulation due to snowfall)
Γ = horizontal extent of the ice



time t_0

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

➢ Thickness H determines the **geometry** for velocity equations.

Ice Sheet Evolution

Ice velocity equations are **coupled** with equations for ice sheet evolution (thickness) and ice temperature.

- **Energy equation** for the temperature T:

$$\rho c \frac{\partial T}{\partial t} + \rho c \boldsymbol{u} \cdot \nabla T = \nabla \cdot (k \nabla T) + 2 \dot{\boldsymbol{\epsilon}} \boldsymbol{\sigma}, \quad \text{in} \quad \Omega_H$$

➢ Flow factor A in Glen's law viscosity μ is function of T.

- **Mass equation** for the ice thickness H:

$$\frac{\partial H}{\partial t} + \nabla \cdot (\overline{\boldsymbol{u}} H) + \dot{b}, \quad \text{on} \quad \Gamma$$

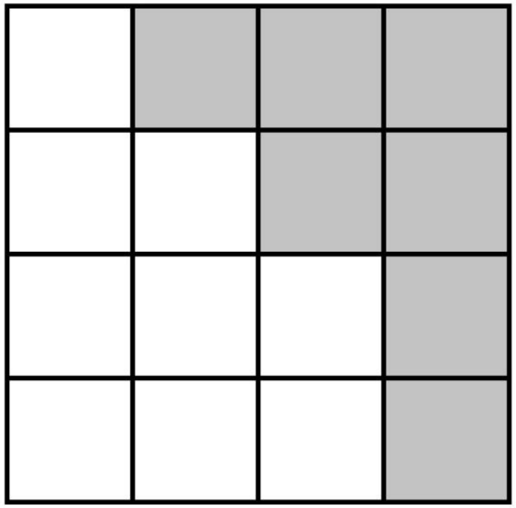
$\overline{\boldsymbol{u}}$ = vertically averaged $\boldsymbol{u}$

$\dot{b}$ = surface mass balance

(given accumulation-ablation function that accounts for e.g. accumulation due to snowfall)

Γ = horizontal extent of the ice

➢ Thickness H determines the **geometry** for velocity equations.



time t_1

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

Ice Sheet Evolution

Ice velocity equations are **coupled** with equations for ice sheet evolution (thickness) and ice temperature.

- **Energy equation** for the temperature T:

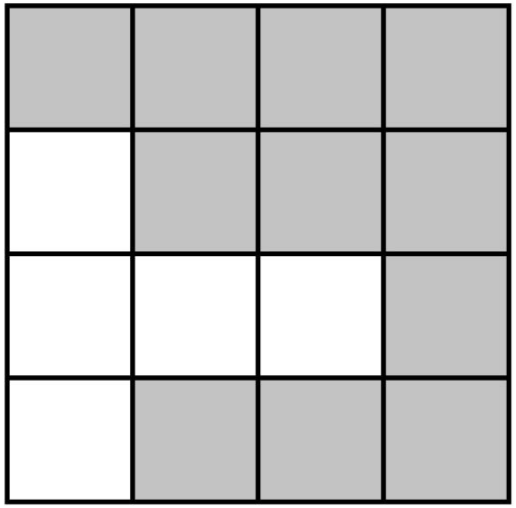
$$\rho c \frac{\partial T}{\partial t} + \rho c \boldsymbol{u} \cdot \nabla T = \nabla \cdot (k \nabla T) + 2 \dot{\boldsymbol{\epsilon}} \boldsymbol{\sigma}, \quad \text{in} \quad \Omega_H$$

➢ Flow factor A in Glen's law viscosity μ is function of T.

- **Mass equation** for the ice thickness H:

$$\frac{\partial H}{\partial t} + \nabla \cdot (\overline{\boldsymbol{u}} H) + \dot{b}, \quad \text{on} \quad \Gamma$$

$\overline{\boldsymbol{u}}$ = vertically averaged $\boldsymbol{u}$
$\dot{b}$ = surface mass balance
(given accumulation-ablation function that accounts for e.g. accumulation due to snowfall)
Γ = horizontal extent of the ice



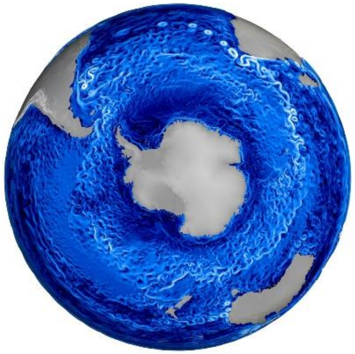
time t_2

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

➢ Thickness H determines the **geometry** for velocity equations.

Outline

1. **Background**
   - Motivation for climate & land-ice modeling
   - ISMs, ESMs & projects
   - Land-ice equations
   - **Our codes: ALI, MALI**
2. Algorithms and software
   - Discretization & meshes
   - Nonlinear solvers
   - Linear solvers
   - Performance-portability
   - Ice sheet initialization
   - Towards UQ
3. Simulations
4. Summary



Our Codes

MALI = MPAS + ALI

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

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

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

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

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

Codes:

=multi-physics PDE code*

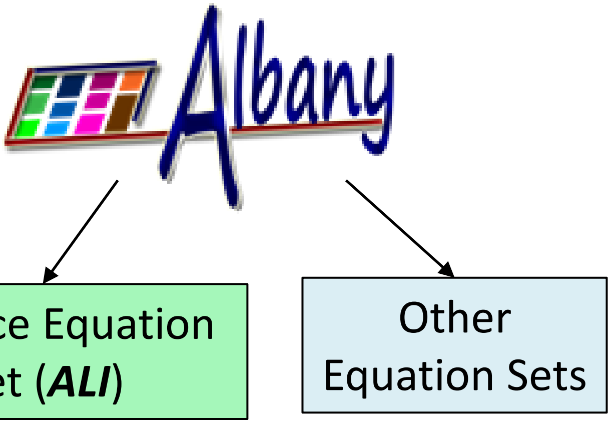
Albany Land-Ice (ALI)

Model for Prediction Across Scales

*https://github.com/SNLComputation/Albany.

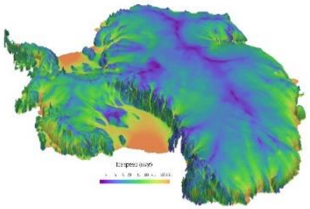
Albany Land-Ice (ALI) FO Stokes Solver

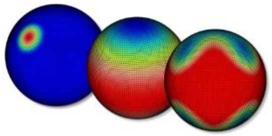
The ***Albany Land-Ice*** First Order Stokes solver is implemented in a Sandia open-source parallel C++ multi-physics finite element code known as…

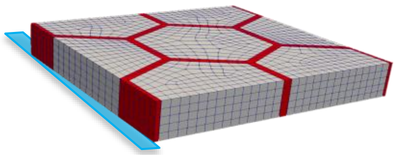


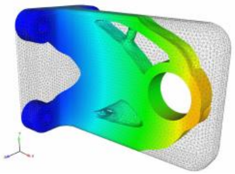
Land Ice Equation Set (***ALI***)

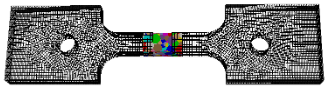
Other Equation Sets











Albany: https://github.com/SNLComputation/Albany

"Agile Components"

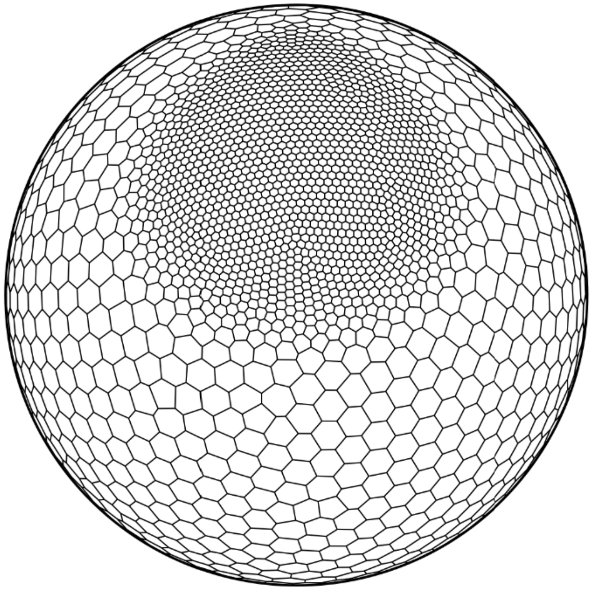
- Discretizations/meshes
- Solver libraries
- Preconditioners
- Automatic differentiation
- Performance portable kernels
- Many others!

- Parameter estimation
- Uncertainty quantification
- Optimization
- Bayesian inference

Trilinos: https://github.com/trilinos/Trilinos

Dakota: https://dakota.sandia.gov/

Model for Prediction Across Scales (MPAS)

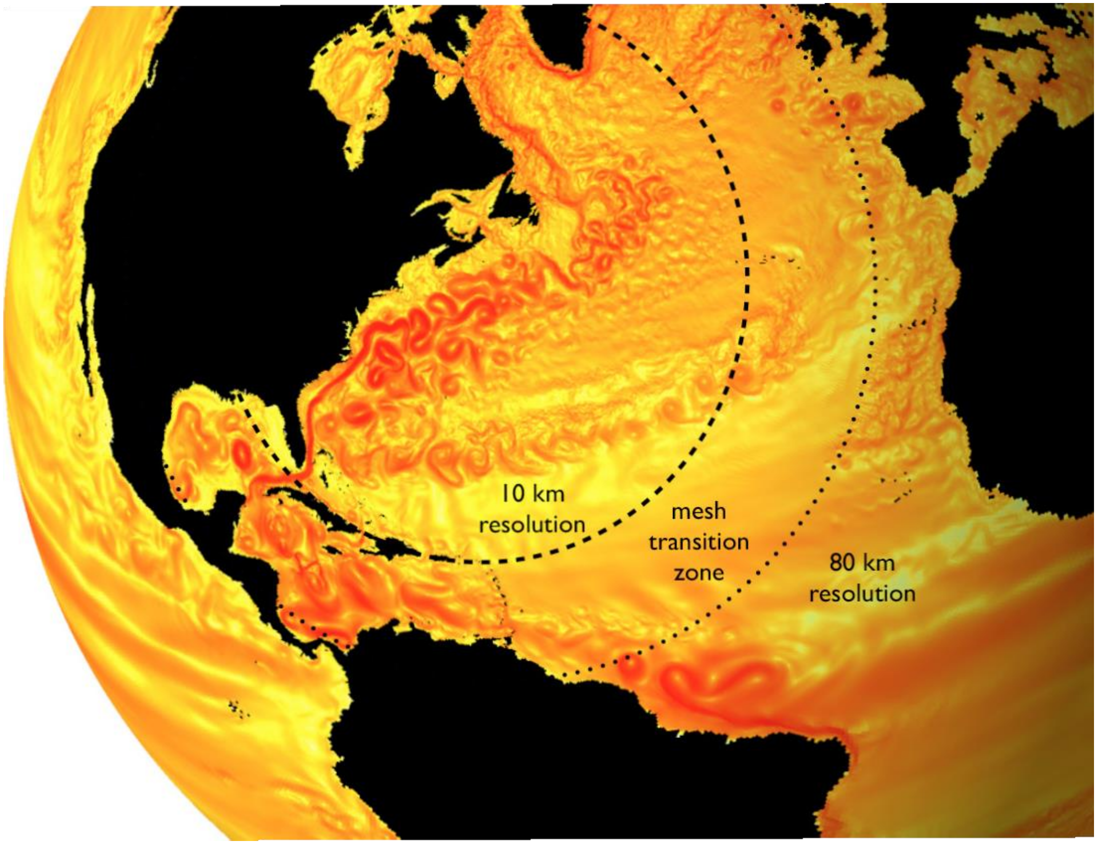


Model for Prediction Across Scales (MPAS): climate modeling framework built around SCVT* meshes (LANL + NCAR collaboration)

***SCVT** = Spherical Centroidal Voronoi Tesselations

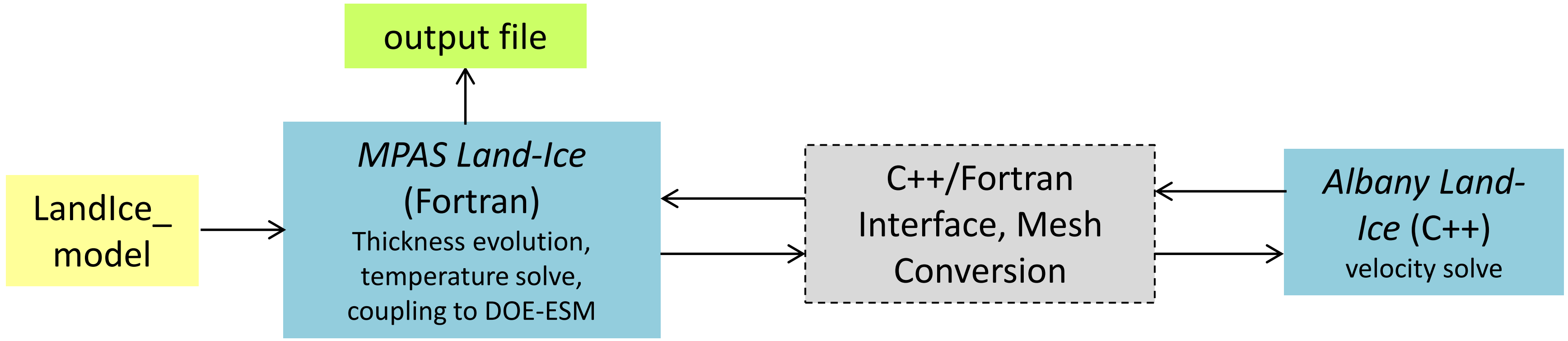
MPAS
Model for Prediction Across Scales

- Ocean[1], sea ice[2], and land ice[3] dynamical cores
- Built using shared software framework
- New capabilities added to one core benefit all others



[1] Ringler et al., 2013; [2] Turner et al. (in prep); [3] Hoffman et al. (in prep)

MPAS + ALI Coupling (MALI)



output file

LandIce_ model

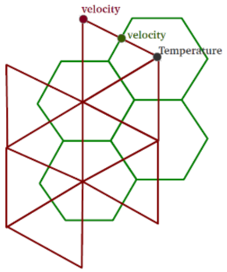
MPAS Land-Ice (Fortran)
Thickness evolution, temperature solve, coupling to DOE-ESM

C++/Fortran Interface, Mesh Conversion

Albany Land-Ice (C++)
velocity solve

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

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



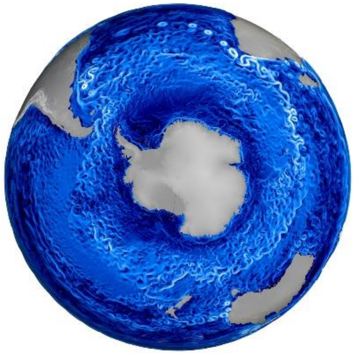
$$\begin{cases} -\nabla \cdot (2\mu \dot{\boldsymbol{\epsilon}}_1) = -\rho g \dfrac{\partial s}{\partial x} \\ -\nabla \cdot (2\mu \dot{\boldsymbol{\epsilon}}_2) = -\rho g \dfrac{\partial s}{\partial y} \end{cases}$$

"Loose" **sequential/staggered** coupling between $\boldsymbol{u}$ and (T, H).

- Making this coupling **tighter** by moving thickness and temperature evolution to Albany is WIP.

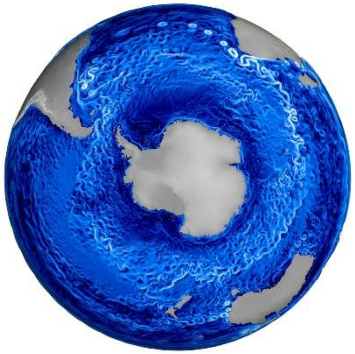
Outline

1. Background
    - Motivation for climate & land-ice modeling
    - ISMs, ESMs & projects
    - Land-ice equations
    - Our codes: ALI, MALI
2. **Algorithms and software**
    - Discretization & meshes
    - Nonlinear solvers
    - Linear solvers
    - Performance-portability
    - Ice sheet initialization
    - Towards UQ
3. Simulations
4. Summary



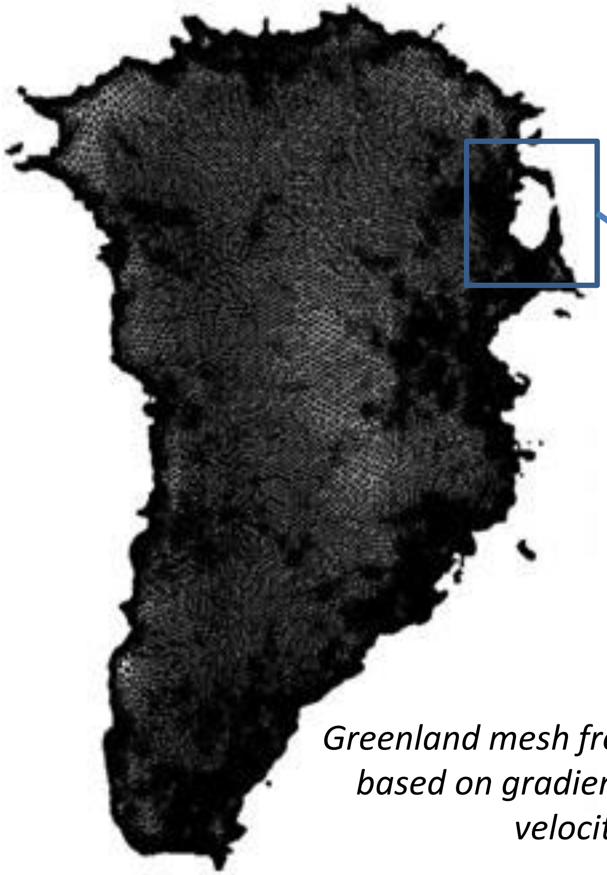
Outline

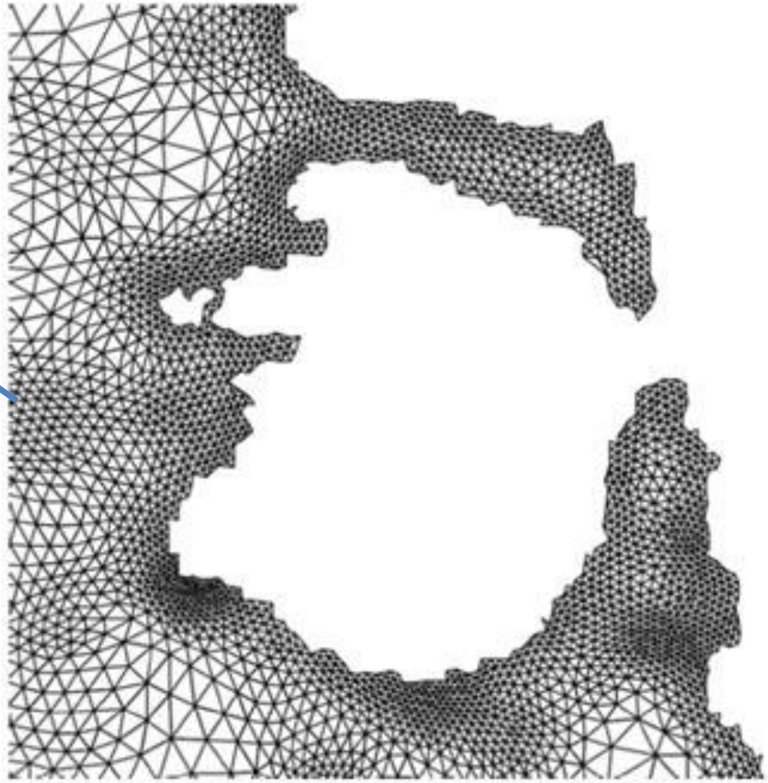
1. Background
   - Motivation for climate & land-ice modeling
   - ISMs, ESMs & projects
   - Land-ice equations
   - Our codes: ALI, MALI
2. **Algorithms and software**
   - **Discretization & meshes**
   - Nonlinear solvers
   - Linear solvers
   - Performance-portability
   - Ice sheet initialization
   - Towards UQ
3. Simulations
4. Summary



Finite Element Discretization

- Can handle well the ***boundary conditions*** arising in land ice modeling.
- Allow the use of ***unstructured meshes*** to concentrate the computational power where it is needed.





Greenland mesh from ALI refined based on gradient of surface velocity

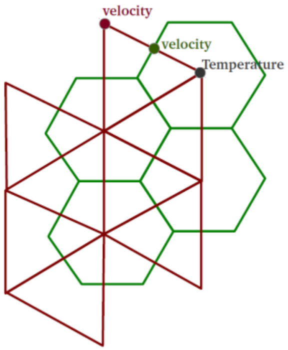
Meshes

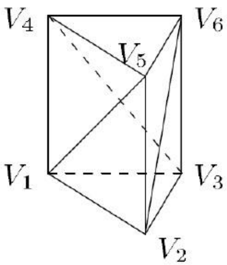
- ALI runs employ **dual of hexagonal mesh** from MPAS extruded to **tetrahedra** for the velocity solve in Albany.

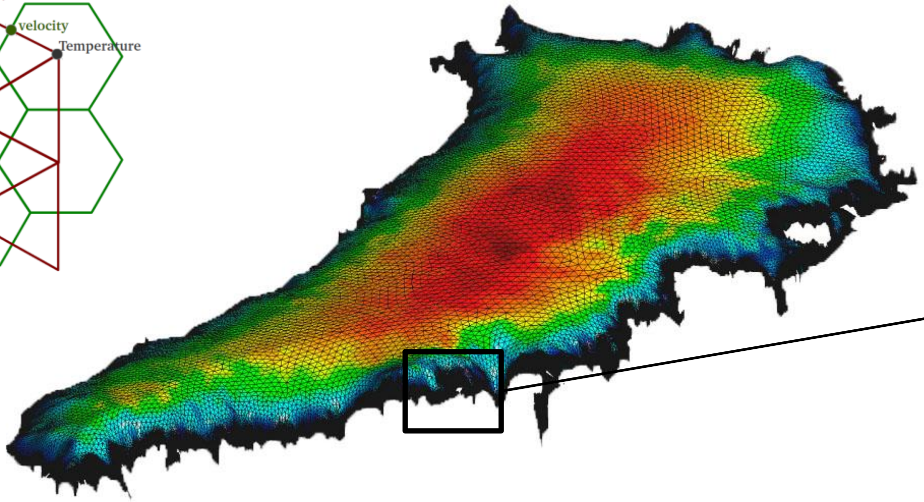
- Meshes are **structured** (**extruded**) in the vertical dimension.

- Ice sheets are **thin** (thickness up to 4 km, horizontal extension of thousands km), meaning we typically have elements with bad aspect ratios.

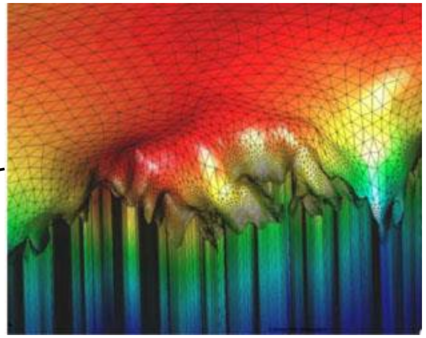
MALI uses dual of hexagonal mesh extruded to tetrahedra.





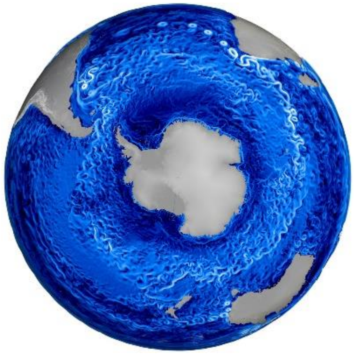


Variable resolution triangular mesh extruded to a (thin) tetrahedral mesh.



Outline

1. Background
   - Motivation for climate & land-ice modeling
   - ISMs, ESMs & projects
   - Land-ice equations
   - Our codes: ALI, MALI
2. **Algorithms and software**
   - Discretization & meshes
   - **Nonlinear solvers**
   - Linear solvers
   - Velocity-temperature coupling
   - Performance-portability
   - Ice sheet initialization
   - Towards UQ
3. Simulations
4. Summary



Nonlinear Solver for Discretized Problem

- **Picard iterations** have been method of choice in ice sheet modeling

Nonlinear Solver for Discretized Problem

- **Picard iterations** have been method of choice in ice sheet modeling
- ALI employs **Newton's method** with several advancements:

Nonlinear Solver for Discretized Problem

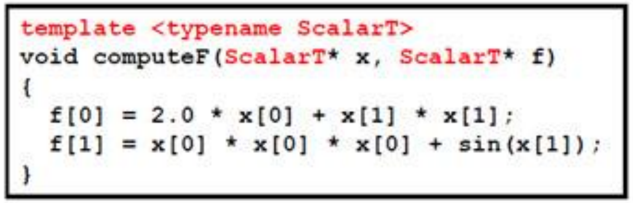
- **Picard iterations** have been method of choice in ice sheet modeling
- ALI employs **Newton's method** with several advancements:
  - ➢ **Automatic differentiation (AD)** Jacobian – gives you exact derivatives/Jacobians without deriving/hand-coding them!

Libraries (Sacado) provides new scalar types that ***overload the math operators*** to propagate embedded quantities via chain rule

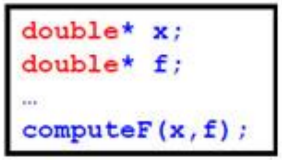
- ❖ *Derivatives:* DFad<double>
- ❖ *Hessians*: DFad<SFad<double,N>>
- ❖ *Stochastic Galerkin resid*: PCE<double>
- ❖ *Stochastic Galerkin Jac*: DFad<PCE<double>
- ❖ *Sensitivities*: DFad<double>

No finite difference truncation error!

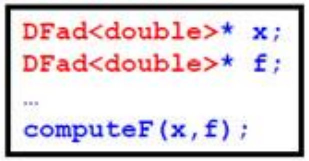
double	DFad<double>
Operation	**Overloaded AD impl**
$c = a \pm b$	$\dot{c} = \dot{a} \pm \dot{b}$
$c = ab$	$\dot{c} = a\dot{b} + \dot{a}b$
$c = a/b$	$\dot{c} = (\dot{a} - c\dot{b})/b$
$c = a^r$	$\dot{c} = ra^{r-1}\dot{a}$



```
template <typename ScalarT>
void computeF(ScalarT* x, ScalarT* f)
{
  f[0] = 2.0 * x[0] + x[1] * x[1];
  f[1] = x[0] * x[0] * x[0] + sin(x[1]);
}
```



```
double* x;
double* f;
...
computeF(x,f);
```



```
DFad<double>* x;
DFad<double>* f;
...
computeF(x,f);
```

*Tezaur *et al.* 2015.

Nonlinear Solver for Discretized Problem

- **Picard iterations** have been method of choice in ice sheet modeling
- ALI employs **Newton's method** with several advancements:
  - **Automatic differentiation (AD)** Jacobian – gives you exact derivatives/Jacobians without deriving/hand-coding them!
  - **Homotopy continuation*** to deal with "singular" viscosity.

*Tezaur *et al.* 2015.

Nonlinear Solver for Discretized Problem

- **Picard iterations** have been method of choice in ice sheet modeling
- ALI employs **Newton's method** with several advancements:
  - ➢ **Automatic differentiation (AD)** Jacobian – gives you exact derivatives/Jacobians without deriving/hand-coding them!
  - ➢ **Homotopy continuation*** to deal with "singular" viscosity.

Glen's Law Viscosity:

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

Undefined for $\boldsymbol{u}$=const!

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

*Tezaur *et al.* 2015.

Nonlinear Solver for Discretized Problem

- **Picard iterations** have been method of choice in ice sheet modeling
- ALI employs **Newton's method** with several advancements:
  - **Automatic differentiation (AD)** Jacobian – gives you exact derivatives/Jacobians without deriving/hand-coding them!
  - **Homotopy continuation*** to deal with "singular" viscosity.

Glen's Law Viscosity:

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

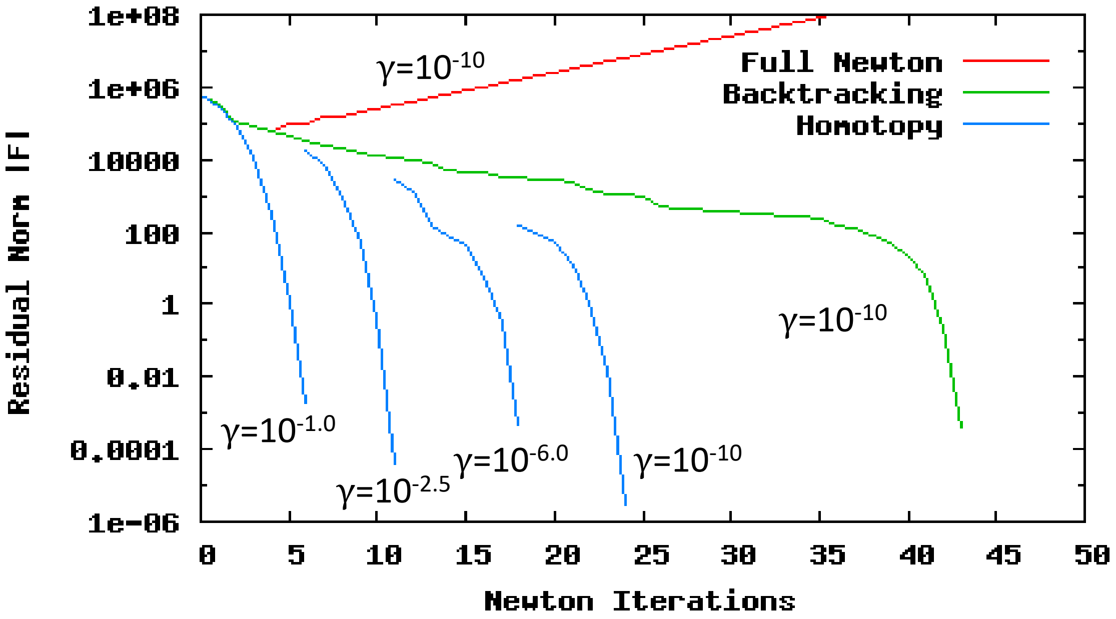
$\gamma =$ regularization parameter (O(1e-10))

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

*Tezaur *et al.* 2015.

Nonlinear Solver for Discretized Problem

- **Picard iterations** have been method of choice in ice sheet modeling
- ALI employs **Newton's method** with several advancements:
  - **Automatic differentiation (AD)** Jacobian – gives you exact derivatives/Jacobians without deriving/hand-coding them!
  - **Homotopy continuation*** to deal with "singular" viscosity.



Improved **robustness** and **faster** nonlinear convergence by doing a **homotopy continuation** w.r.t. γ

Glen's Law Viscosity:

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

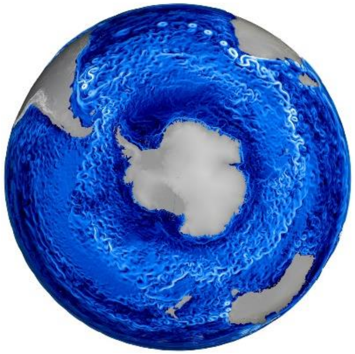
γ = regularization parameter

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

*Tezaur *et al.* 2015.

Outline

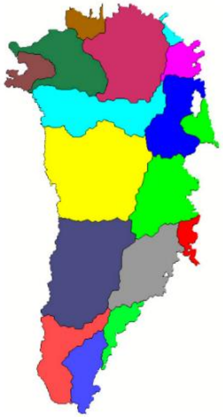
1. Background
   - Motivation for climate & land-ice modeling
   - ISMs, ESMs & projects
   - Land-ice equations
   - Our codes: ALI, MALI
2. **Algorithms and software**
   - Discretization & meshes
   - Nonlinear solvers
   - **Linear solvers**
   - Performance-portability
   - Ice sheet initialization
   - Towards UQ
3. Simulations
4. Summary



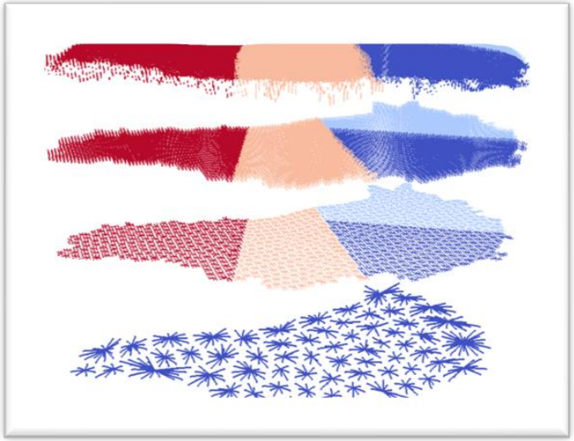
From Nonlinear Solvers to Linear Solvers

- **Krylov iterative linear solvers** are employed – CG or GMRES.
    - FO Stokes equations are **symmetric**.
- Grid partitioning is done on **2D base grid** for best linear solver performance (recall that mesh is layered).

- **Bad aspect ratios**, **floating ice**, and **island/ice hinges** can **wreak havoc** on linear solver!
    - Specialized **solvers/preconditioners** have been developed in Trilinos to deal w/ these issues.
        - AMG[1] preconditioner w/ semi-coarsening[2].
        - Fast and Robust Overlapping Schwarz (FROSch) preconditioner[3] w/ GDSW[4] coarse spaces
    - **Graph-based algorithms** for **removing islands/ice hinges** are being developed[2].



Example parallel decomposition of Greenland geometry.



Visualization of AMG preconditioner[2], which takes advantage of layered nature of 3D mesh.

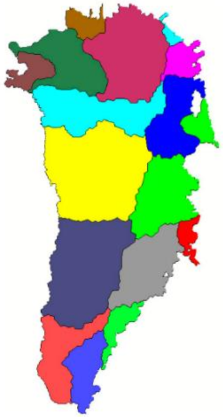
[1] Algebraic Multi-Grid. [2] Tuminaro *et al.* 2016. [3] Heinlein *et al.* 2020. [4] Generalized-Dryja-Smith-Widlund.

From Nonlinear Solvers to Linear Solvers

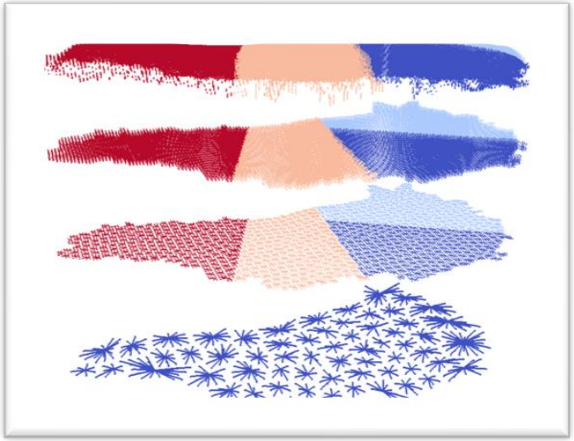
- **Krylov iterative linear solvers** are employed – CG or GMRES.
  - ➢ FO Stokes equations are **symmetric**.
- Grid partitioning is done on **2D base grid** for best linear solver performance (recall that mesh is layered).
- **Bad aspect ratios**, **floating ice**, and **island/ice hinges** can **wreak havoc** on linear solver!

Deep dive

- ➢ Specialized **solvers/preconditioners** have been developed in Trilinos to deal w/ these issues.
  - ❖ AMG[1] preconditioner w/ semi-coarsening[2].
  - ❖ Fast and Robust Overlapping Schwarz (FROSch) preconditioner[3] w/ GDSW[4] coarse spaces
- ➢ **Graph-based algorithms** for **removing islands/ice hinges** are being developed[2].



Example parallel decomposition of Greenland geometry.



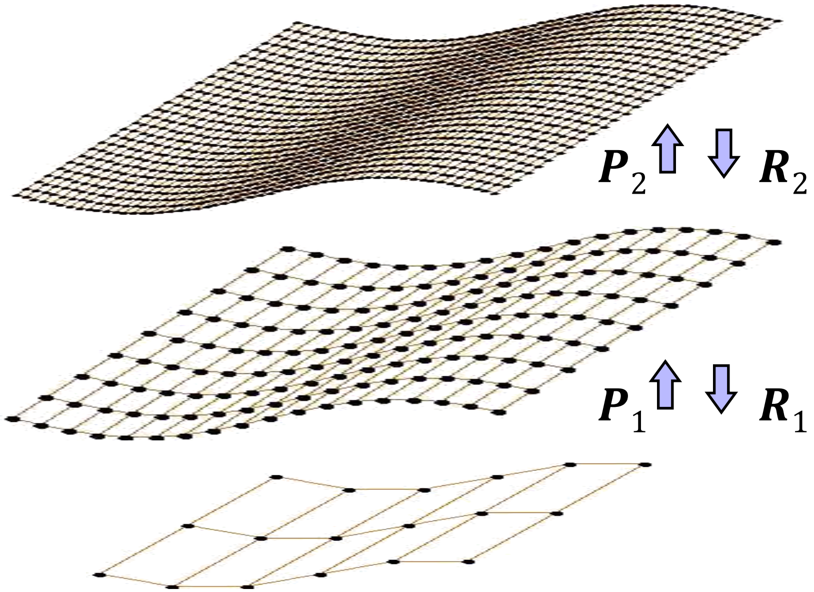
Visualization of AMG preconditioner[2], which takes advantage of layered nature of 3D mesh.

[1] Algebraic Multi-Grid. [2] Tuminaro *et al.* 2016. [3] Heinlein *et al.* 2020. [4] Generalized-Dryja-Smith-Widlund.

How Does Multi-Grid Work?

Basic idea: accelerate convergence of an iterative method on a given grid by solving a series of (cheaper) problems on coarser grids.

- Create set of ***coarse approximations.***
- Apply ***restriction operator*** $\boldsymbol{R}_i$ to interpolate from fine to coarse grid.
- ***Solve*** problem on coarse grid.
- Apply ***prolongation operator*** $\boldsymbol{P}_i$ to get back to original (fine) grid.
- ***Smoothers*** are applied throughout procedure to reduce short wavelength errors.

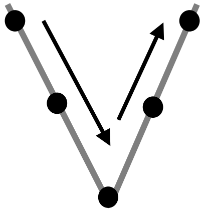


Solve $\boldsymbol{A}_3\boldsymbol{u}_3 = \boldsymbol{f}_3$

Smooth $\boldsymbol{A}_3\boldsymbol{u}_3 = \boldsymbol{f}_3$. Set $\boldsymbol{f}_2 = \boldsymbol{R}_2\boldsymbol{r}_3$.

Smooth $\boldsymbol{A}_2\boldsymbol{u}_2 = \boldsymbol{f}_2$. Set $\boldsymbol{f}_1 = \boldsymbol{R}_1\boldsymbol{r}_2$.

Solve $\boldsymbol{A}_1\boldsymbol{u}_1 = \boldsymbol{f}_1$ directly.



Set $\boldsymbol{u}_3 = \boldsymbol{u}_3 + \boldsymbol{P}_2\boldsymbol{u}_2$. Smooth $\boldsymbol{A}_3\boldsymbol{u}_3 = \boldsymbol{f}_3$.

Set $\boldsymbol{u}_2 = \boldsymbol{u}_2 + \boldsymbol{P}_1\boldsymbol{u}_1$. Smooth $\boldsymbol{A}_2\boldsymbol{u}_2 = \boldsymbol{f}_2$.

Scalable Algebraic Multi-Grid (AMG) Preconditioners

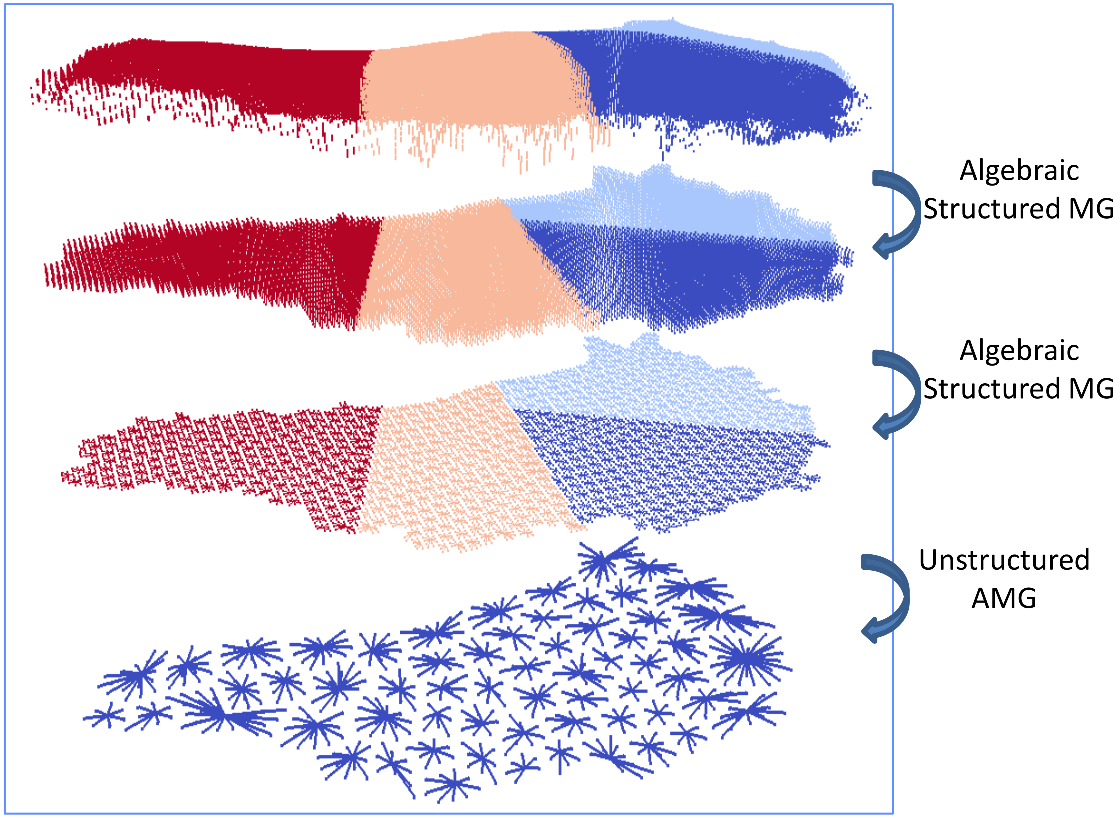
Bad aspect ratios $(dx \gg dz)$ ruin classical AMG convergence rates!

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

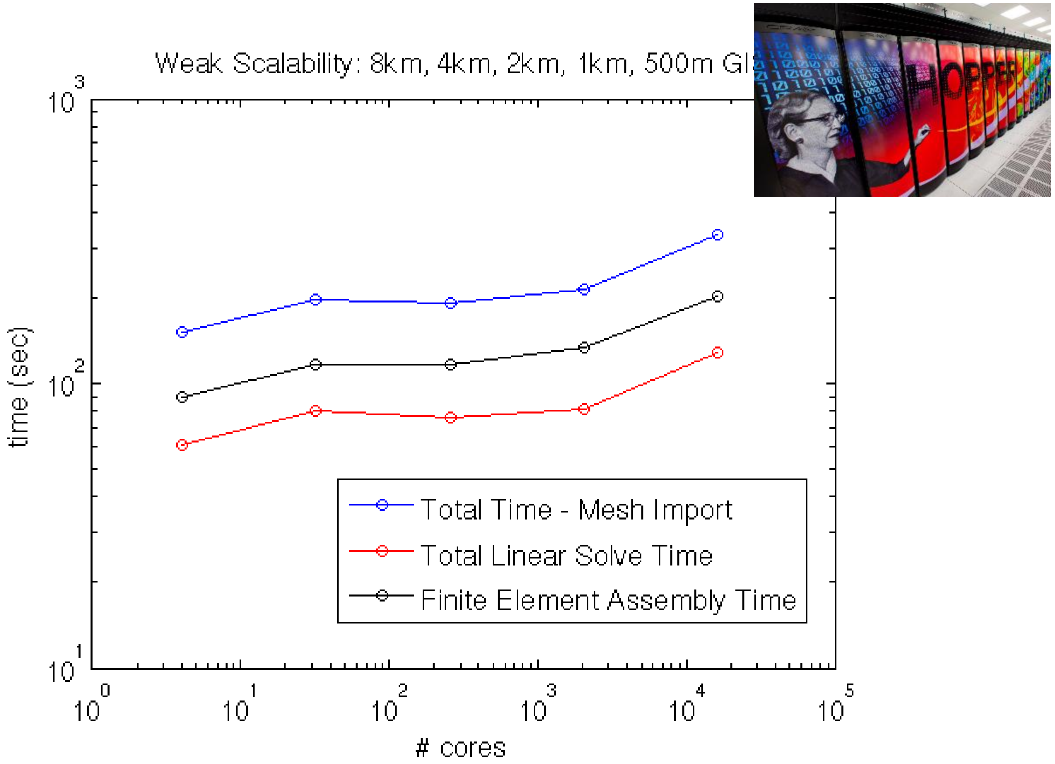
$\Rightarrow$ Solvers (AMG and ILU) must take **aspect ratios** into account!

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

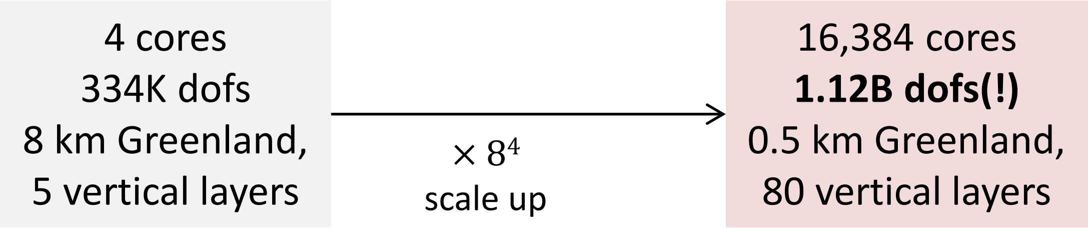
See (Tezaur *et al.*, 2015), (Tuminaro *et al.*, 2016).



Greenland Controlled Weak Scalability Study



- 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!

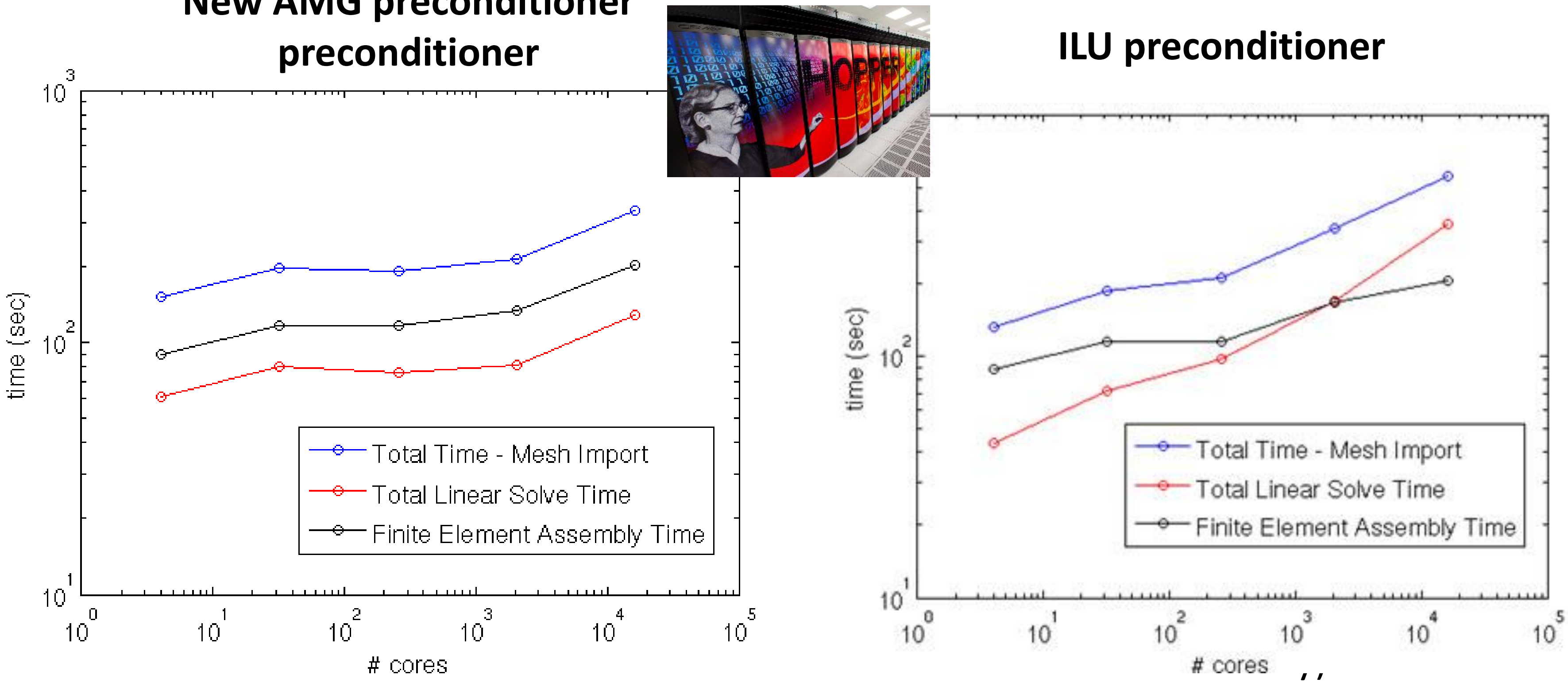


4 cores
334K dofs
8 km Greenland,
5 vertical layers

$\times 8^4$
scale up

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

Greenland Controlled Weak Scalability Study



New AMG preconditioner preconditioner

ILU preconditioner

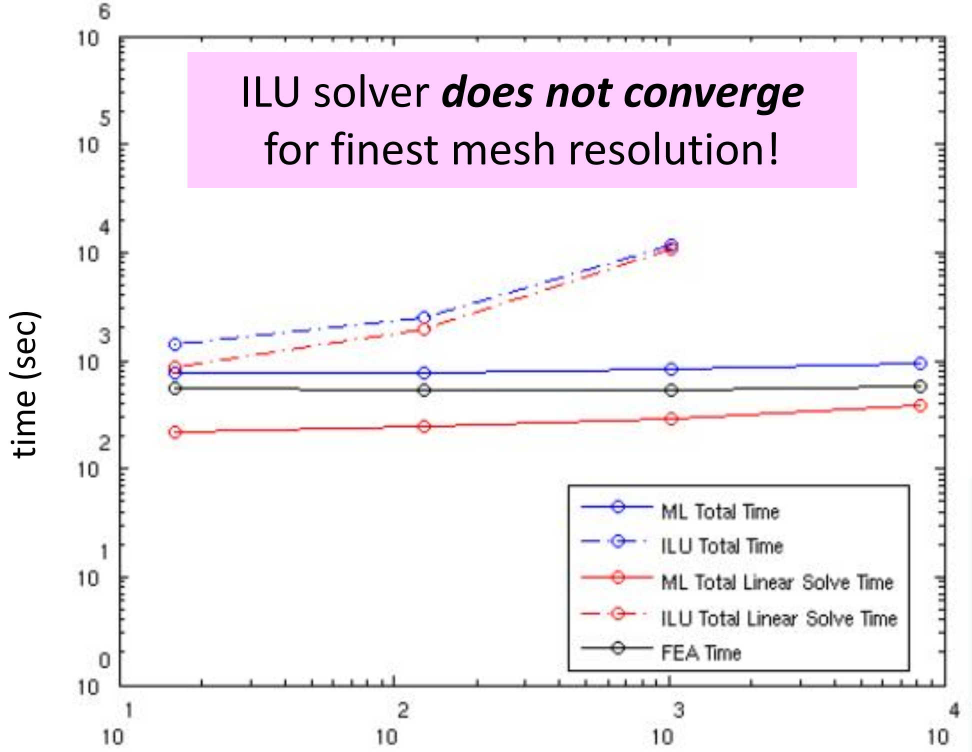
4 cores
334K dofs
8 km Greenland,
5 vertical layers

× 8^4
scale up

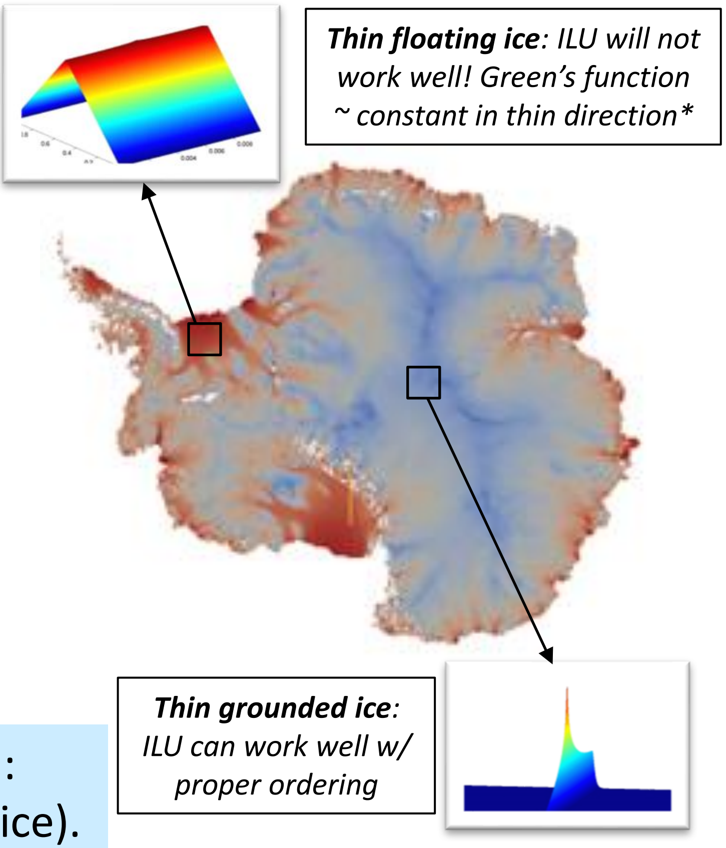
16,384 cores
1.12B dofs(!)
0.5 km Greenland,
80 vertical layers

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

Weak scalability: Antarctica



ILU solver ***does not converge*** for finest mesh resolution!



***Thin floating ice**: ILU will not work well! Green’s function ~ constant in thin direction**

***Thin grounded ice**: ILU can work well w/ proper ordering*

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

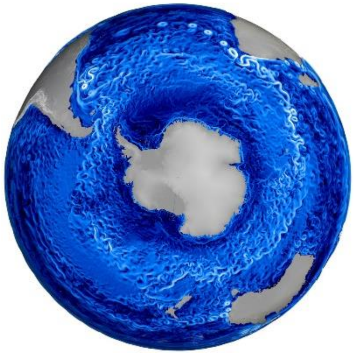
- **Weak scaling study**: 2.5M → 1.1B dofs, 16 → 8192 cores
- Initialized with realistic basal friction and temperature field from BEDMAP2.
- **Iterative linear solver**: GMRES.
- **Preconditioner**: ILU vs. new AMG based on aggressive semi-coarsening.

See (Tuminaro *et al.*, *SISC*, 2016).

* A^{-1} will have large number of non-zeroes, so approximate inverse ILU preconditioner is ineffective.

Outline

1. Background
   - Motivation for climate & land-ice modeling
   - ISMs, ESMs & projects
   - Land-ice equations
   - Our codes: ALI, MALI
2. **Algorithms and software**
   - Discretization & meshes
   - Nonlinear solvers
   - Linear solvers
   - **Performance-portability**
   - Ice sheet initialization
   - Towards UQ
3. Simulations
4. Summary



Performance-Portability via *Kokkos*

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

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

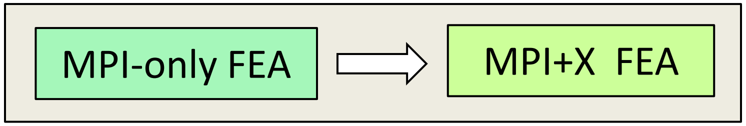
- ***Kokkos*****: open-source C++ library that provides performance portability across diverse devises with different memory models.
  - ➢ A *programming model* as much as a software library.
  - ➢ Provides automatic access to OpenMP, CUDA, Pthreads, ...
  - ➢ Templated meta-programming: parallel_for, 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 Land-Ice* has been rewritten using *Kokkos* functors.
- Performance portability for **linear solvers** is an ongoing research topic within Trilinos.

*X = OpenMP, CUDA, etc. **https://github.com/kokkos/kokkos

Kokkos-ification of Finite Element Assembly (FEA)



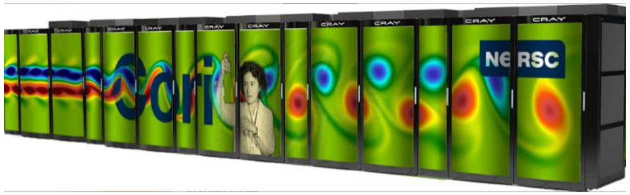
MPI-only FEA → MPI+X FEA

```
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/Results

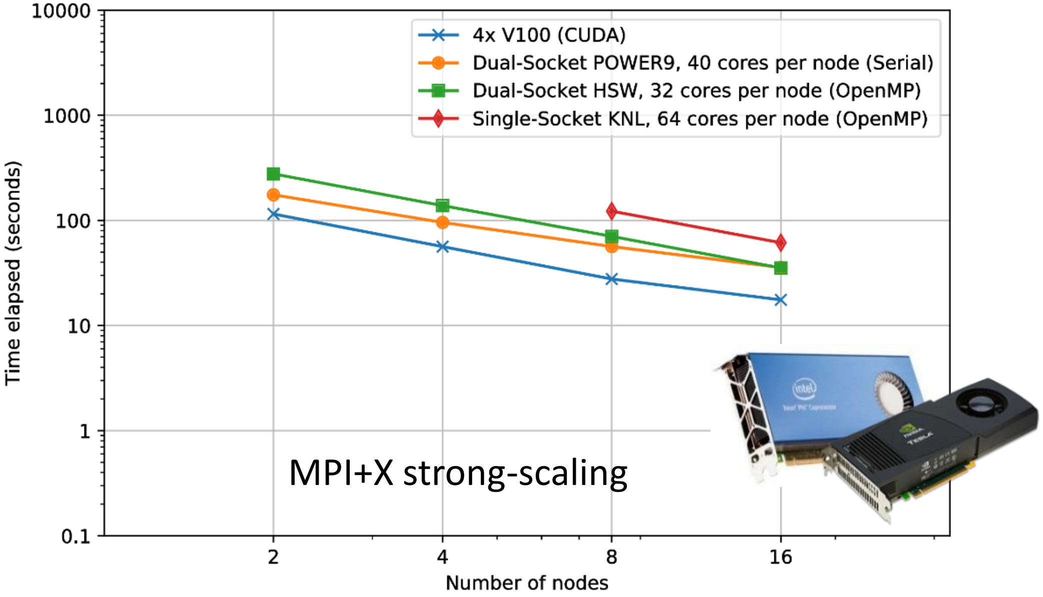
Performance-portability of FEA in ALI has been tested across ***multiple architectures***: Intel Sandy Bridge, Intel Skylake, IBM POWER8, IBM POWER9, Keplar/Pascal/Volta/Ampere GPUs, KNL Xeon Phi



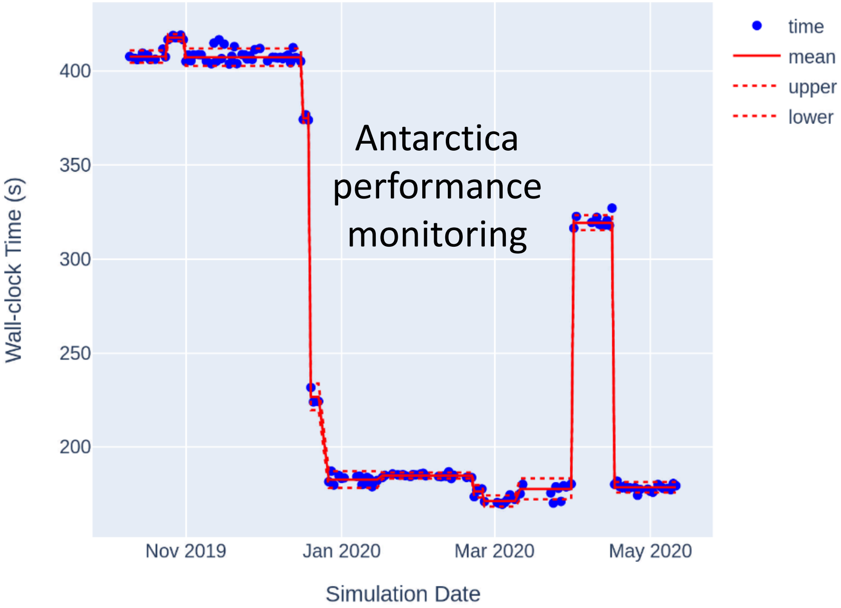
Cori (NERSC): 2,388 Haswell nodes [2 Haswell (32 cores)]
9,688 KNL nodes [1 Xeon Phi KNL (68 cores)]

Summit (OLCF): 4600 nodes [2 P9 (22 cores) + V100 (6 GPUs)]

Future targets: Aurora Intel GPU (ALCF), Frontier AMD GPU (OLCF)



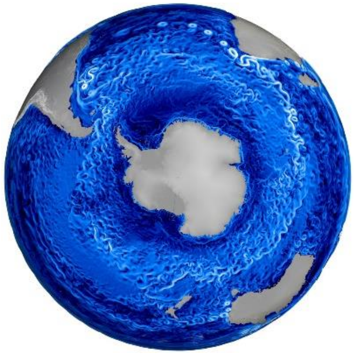
MPI+X strong-scaling



Antarctica performance monitoring

Outline

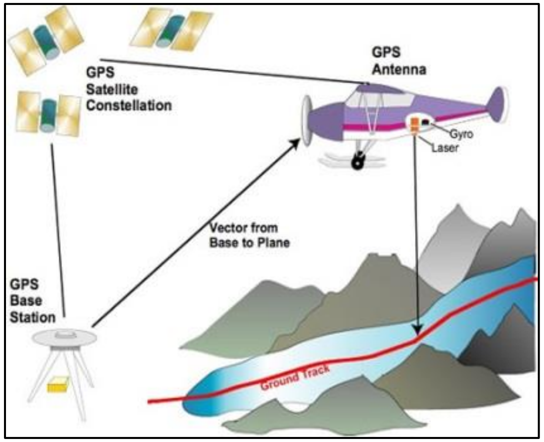
1. Background
   - Motivation for climate & land-ice modeling
   - ISMs, ESMs & projects
   - Land-ice equations
   - Our codes: ALI, MALI
2. **Algorithms and software**
   - Discretization & meshes
   - Nonlinear solvers
   - Linear solvers
   - Performance-portability
   - **Ice sheet initialization**
   - Towards UQ
3. Simulations
4. Summary



Inversion for Ice Sheet Initialization

Goal: 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).



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

Available data/measurements:

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

Fields to be estimated:

- Basal friction β, ice thickness H

"Spin-up" approach: initialize model with (imperfect/unknown) present state and integrate forward until states consistent with observations are reached.

- Can require **a lot of CPU time** ("spin-up time"): long timescale adjustments to past BC forcing requires a model "spin-up" of order 10^4-10^5 years*.
- "Spun-up" initial conditions can result in "**shocks**", which initiate large transients that can **derail** dynamic ice simulations*.

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

Deterministic Inversion

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

minimize $_{\beta,H}\ m(\boldsymbol{u}, H)$

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)

$\mathcal{R}(\boldsymbol{u}, H)$: regularization term

σ: standard deviation (weight of uncertanties)

Modeling Assumptions: ice described by FO Stokes equations; ice close to mechanical equilibrium.

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

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

$$+ \mathcal{R}(\boldsymbol{u}, H)$$

regularization terms

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

Deterministic Inversion

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

$$\text{minimize}_{\,\beta,H} \;\; m(\boldsymbol{u}, H)$$

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)

$\mathcal{R}(\boldsymbol{u}, H)$: regularization term

σ: standard deviation (weight of uncertanties)

Modeling Assumptions: ice described by FO Stokes equations; ice close to mechanical equilibrium.

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

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

$$+ \mathcal{R}(\boldsymbol{u}, H)$$

regularization terms

common

novel

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

Deterministic Inversion

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

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

$\boldsymbol{U}$: computed depth averaged velocity

H: ice thickness

β: basal sliding friction coefficient

τ_s: surface mass balance (SMB)

$\mathcal{R}(\boldsymbol{u}, H)$: regularization term

σ: standard deviation (weight of uncertanties)

Modeling Assumptions: ice described by FO Stokes equations; ice close to mechanical equilibrium.

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

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

$$+ \mathcal{R}(\boldsymbol{u}, H)$$

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 $_{\beta,H}$ $m(\boldsymbol{u}, H)$

s.t. FO Stokes PDEs

Solved via embedded ***adjoint-based PDE-constrained optimization*** algorithm in Albany Land-Ice.

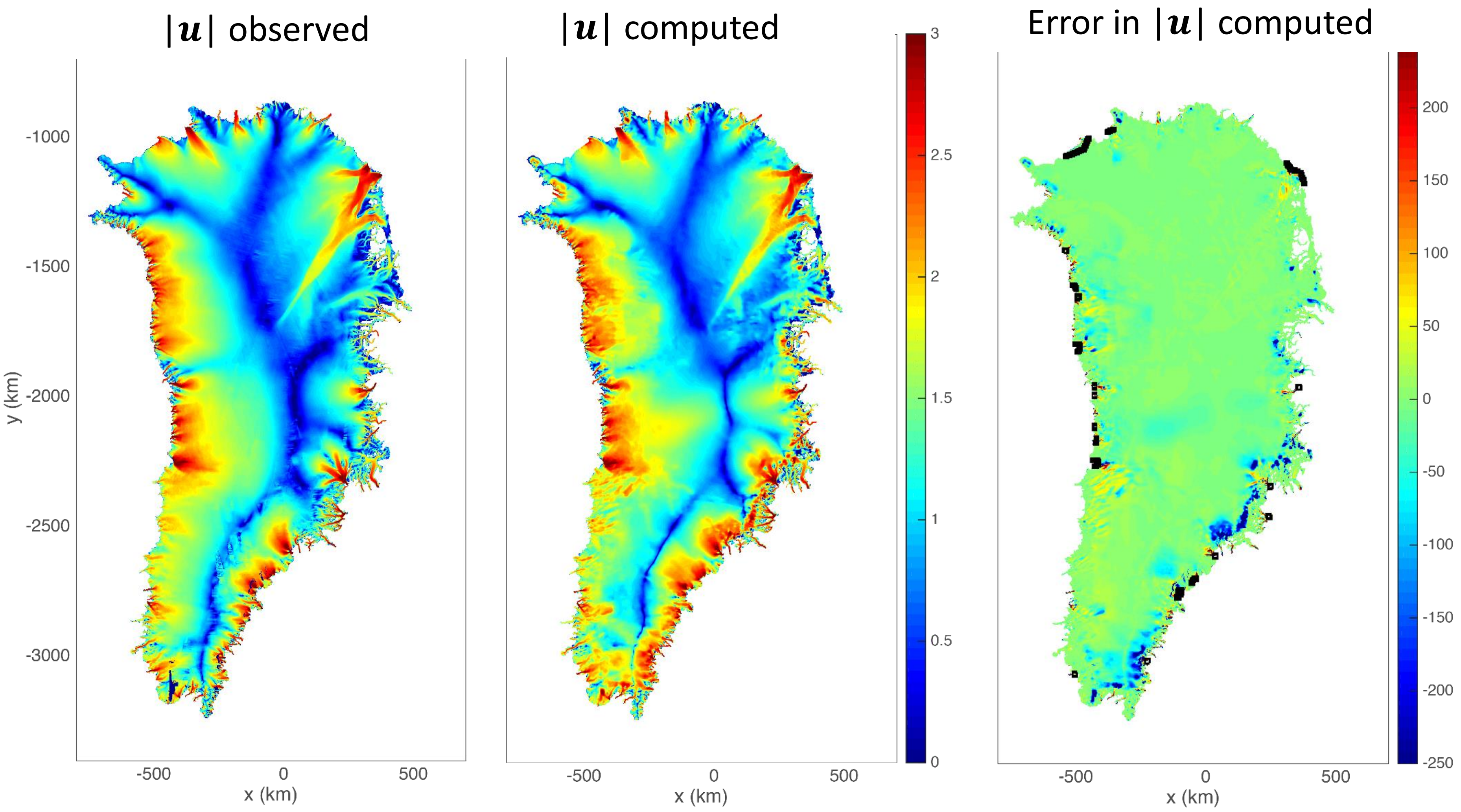
*Approach efficiently computes **gradients** of $m(\boldsymbol{u}, H)$ by solving **linear adjoint PDEs**.*

Algorithm	Software
Finite Element Method discretization	Albany
Quasi-Newton optimization (L-BFGS)	ROL
Nonlinear solver (Newton)	NOX
Krylov linear solvers	Belos+Ifpack2/Muelu

- **Some details:**
  - ***Regularization:*** Tikhonov.
  - Total derivatives of objective functional $m(\boldsymbol{u}, H)$ 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.

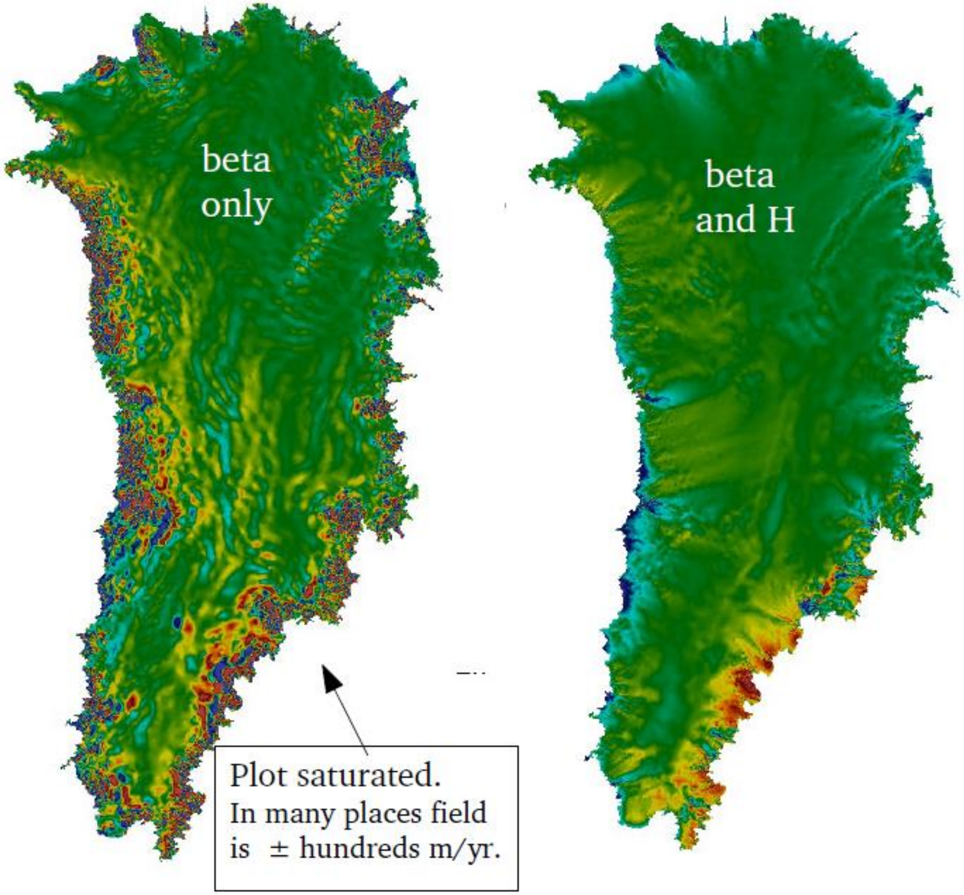
Deterministic Inversion: 1km Greenland Initial Condition*



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

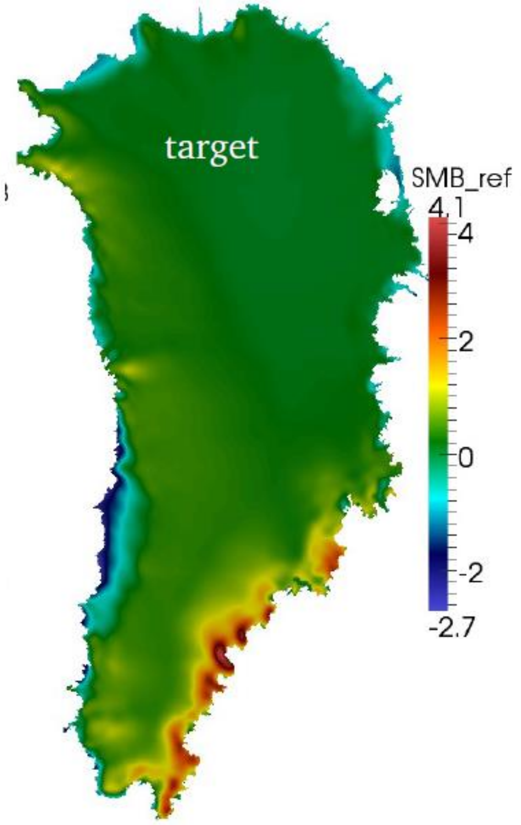
Deterministic Inversion: Common vs. Novel Approach*

SMB (m/yr) needed for equilibrium



Plot saturated.
In many places field is ± hundreds m/yr.

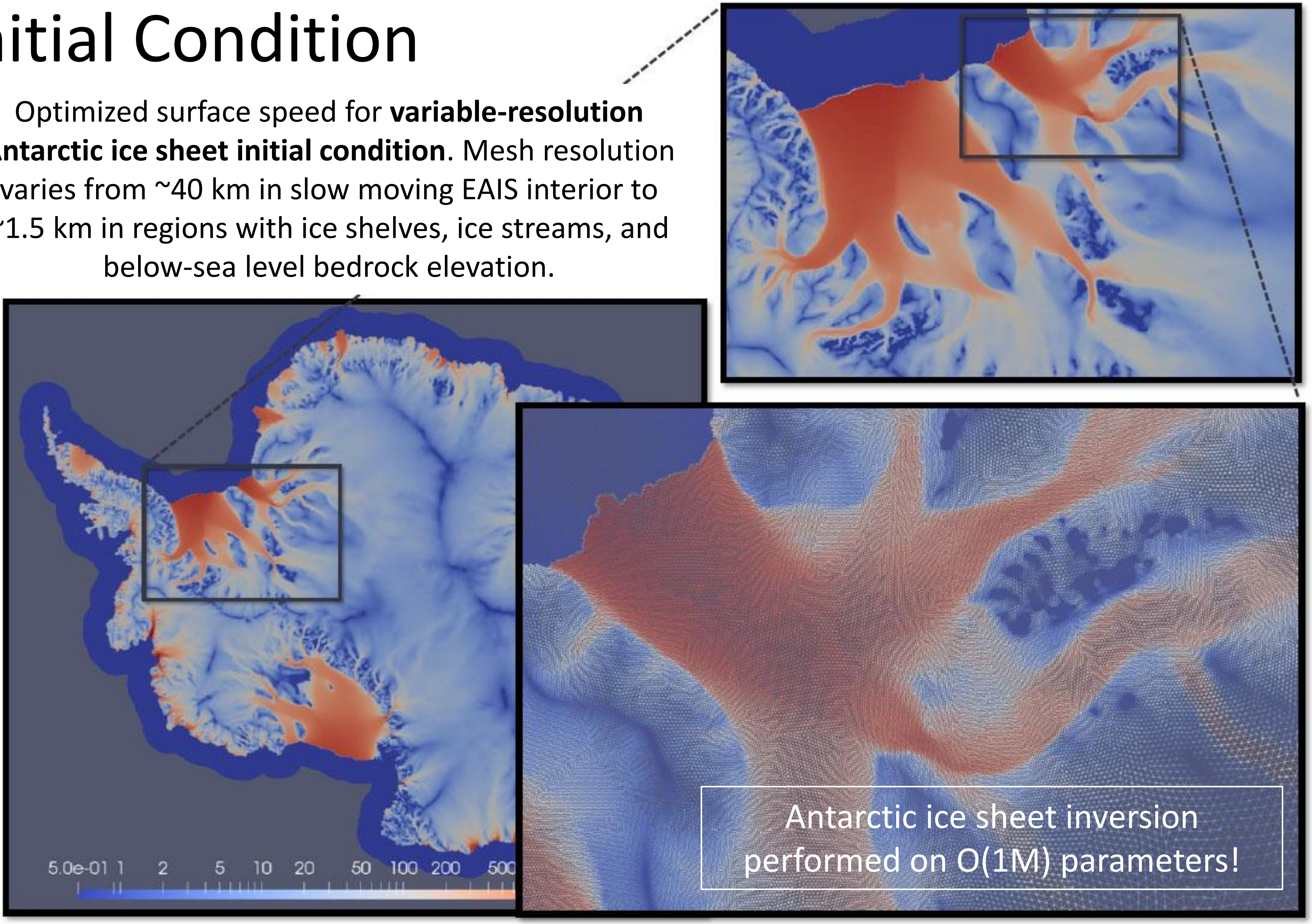
SMB (m/yr) from climate model (Ettema et al. 2009, RACMO2/GR)



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

High-Resolution Antarctica Optimal Initial Condition

Optimized surface speed for **variable-resolution Antarctic ice sheet initial condition**. Mesh resolution varies from ~40 km in slow moving EAIS interior to ~1.5 km in regions with ice shelves, ice streams, and below-sea level bedrock elevation.



Antarctic ice sheet inversion performed on O(1M) parameters!

Velocity-Temperature Coupling

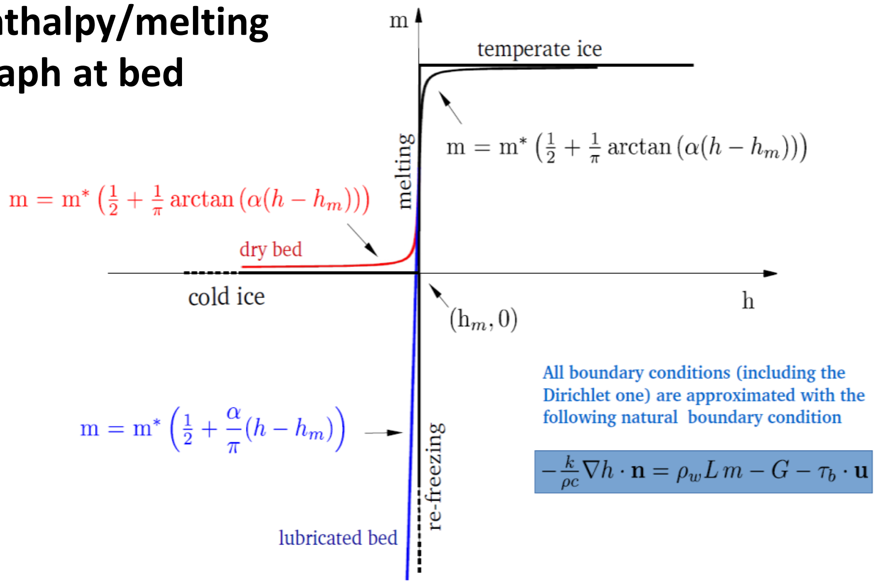
- MALI default coupling between FO Stokes and temperature is **sequential**
- We are working towards **fully-coupled flow** + **temperature** model
  - ➢ Enables computation of **self-consistent** ice sheet initial state (with ice temperature).
- Current implementation in Albany Land-Ice: steady-state **enthalpy equation** coupled monolithically with **FO Stokes equations**

Enthalpy equation: $\boldsymbol{u} \cdot \nabla h + \nabla \cdot \boldsymbol{q} = \tau : \dot{\boldsymbol{\epsilon}}$

h = enthalpy
τ = dissipation heat
$\boldsymbol{q}$ = total heat flux

- Challenges include **strong nonlinearity** of basal BC due to **phase changes** and **robust solvers.**

Enthalpy/melting graph at bed



All boundary conditions (including the Dirichlet one) are approximated with the following natural boundary condition

$$-\frac{k}{\rho c}\nabla h \cdot \mathbf{n} = \rho_w L\, m - G - \tau_b \cdot \mathbf{u}$$

Strategy: approximate enthalpy/melting graph at bed by smooth function, perform parameter **continuation** to smoothly transition from cold to temperate ice (left).

*Developing **robust linear solvers** for coupled velocity-temperature equations is WIP.*

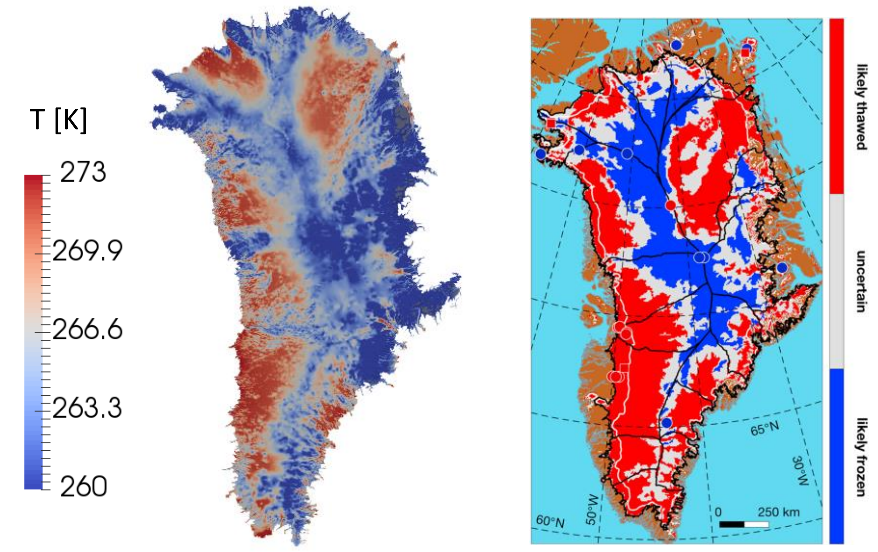
Simultaneous Velocity-Temperature Initialization (Inversion)

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

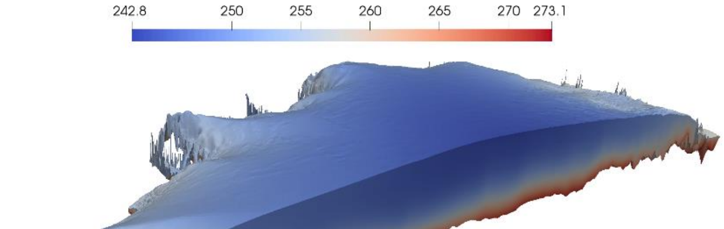
$$\text{minimize}_{\beta,H} \; m(\boldsymbol{u}, H)$$

s.t. FO Stokes PDEs + **Enthalpy PDE**

Left: Computed basal temperature
Right: Thawed/frozen map from MacGregor *et al., JGR*, 2016



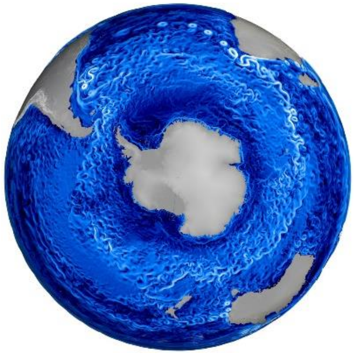
- With an implicit steady-state coupled **temperature-velocity model**, one can obtain **self-consistent** state in **one shot**.
- Initialization capability is **unmatched** by other land-ice codes:
  - Typically **~10K years** are needed to equilibrate ice temperature
  - Our solver **robustly** computes the steady-state temperature coupled w/ velocity at every iteration of the optimization



modeled
ice temperature

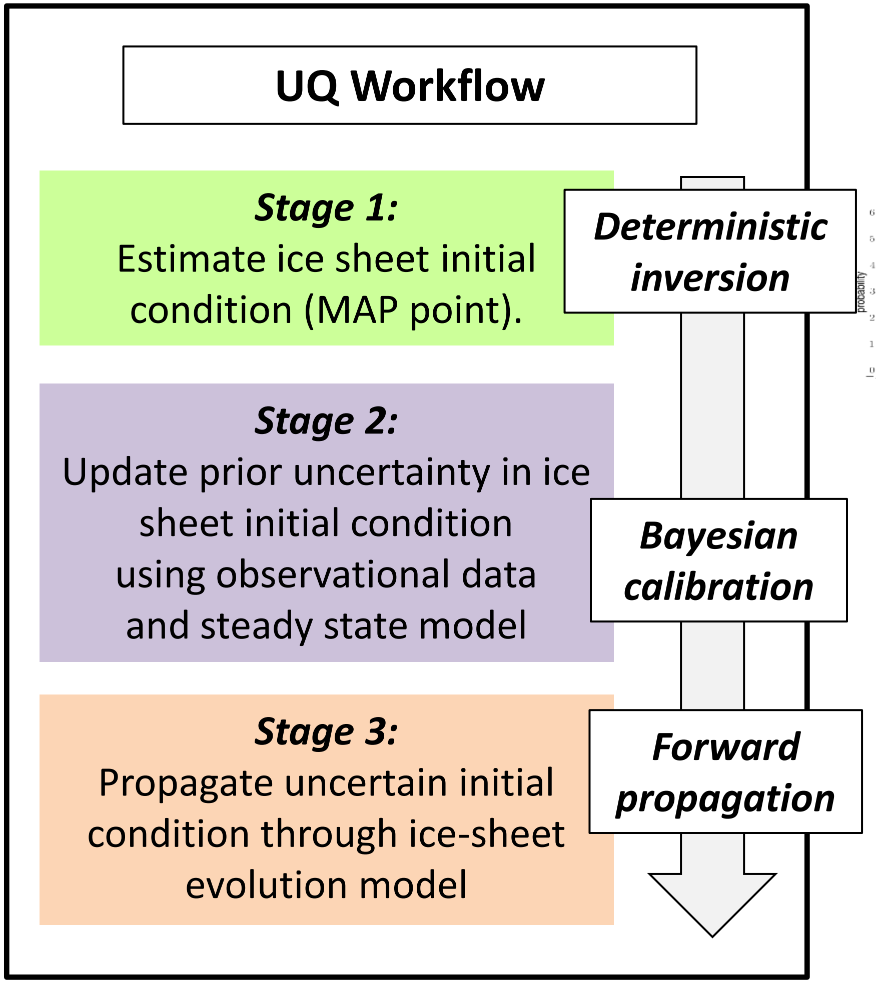
Outline

1. Background
   - Motivation for climate & land-ice modeling
   - ISMs, ESMs & projects
   - Land-ice equations
   - Our codes: ALI, MALI
2. **Algorithms and software**
   - Discretization & meshes
   - Nonlinear solvers
   - Linear solvers
   - Performance-portability
   - Ice sheet initialization
   - **Towards UQ**
3. Simulations
4. Summary



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).

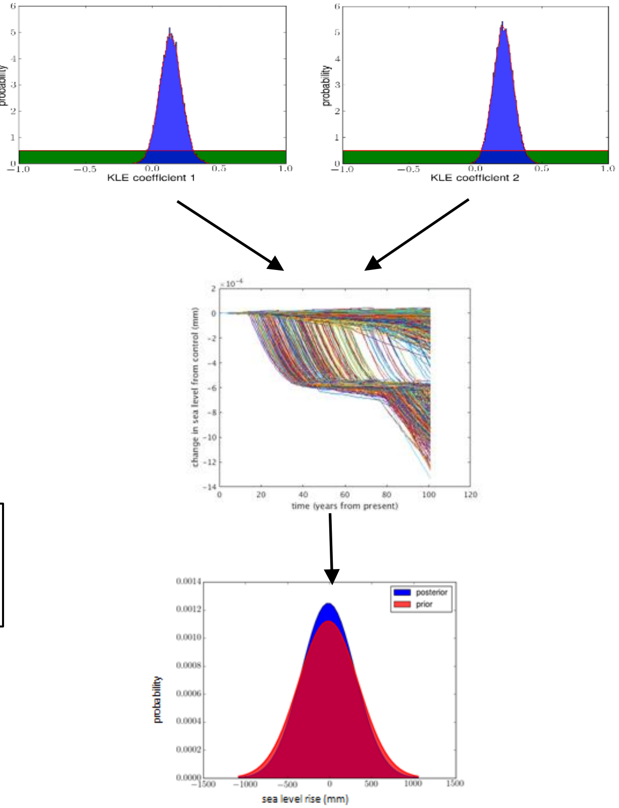
Deterministic inversion

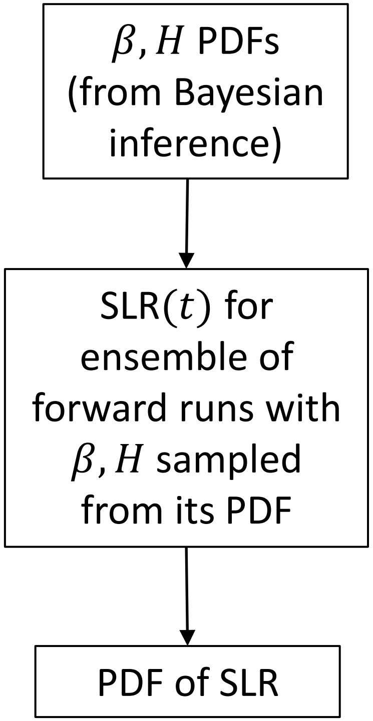
Stage 2:
Update prior uncertainty in ice sheet initial condition using observational data and steady state model

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), 2021.

Uncertainty Quantification*

UQ Workflow

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

Deterministic inversion

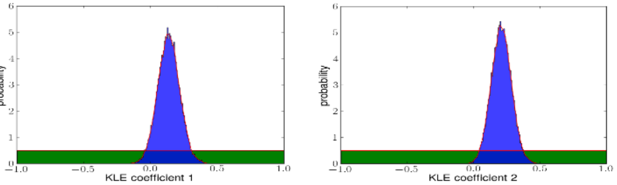
Stage 2:
Update prior uncertainty in ice sheet initial condition using observational data and steady state model

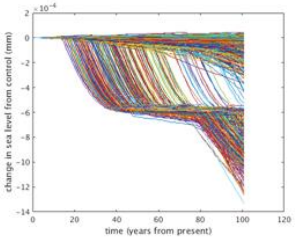
Bayesian calibration

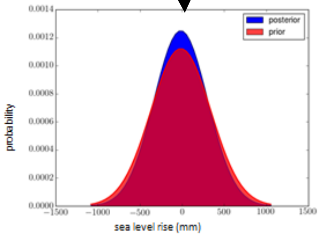
Stage 3:
Propagate uncertain initial condition through ice-sheet evolution model

Forward propagation

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***







β, H PDFs (from Bayesian inference)

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

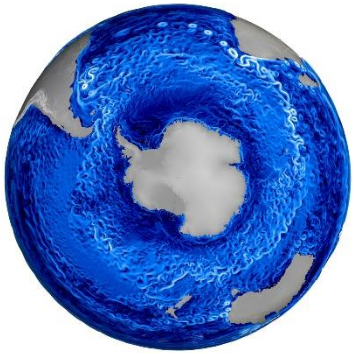
PDF of SLR

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

* Jakeman *et al.* (in prep), 2021.

Outline

1. Background
   - Motivation for climate & land-ice modeling
   - ISMs, ESMs & projects
   - Land-ice equations
   - Our codes: ALI, MALI
2. Algorithms and software
   - Discretization & meshes
   - Nonlinear solvers
   - Linear solvers
   - Performance-portability
   - Ice sheet initialization
   - Towards UQ
3. **Simulations**
4. Summary

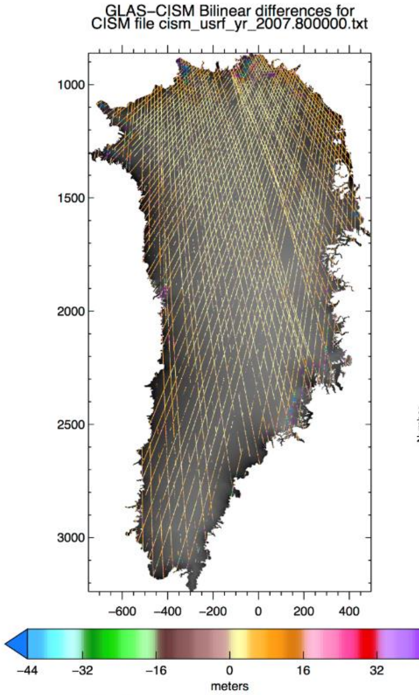


Model Validation

Our model has been **validated*** using data from two satellites: ICESat, GRACE.

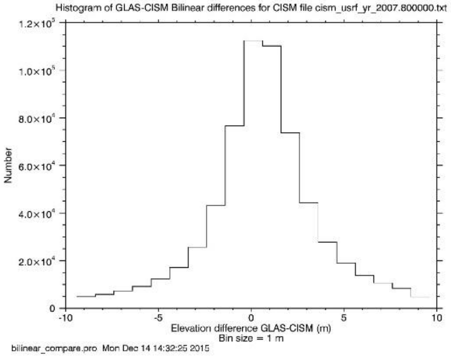
ICESat1 [states]	2003 – 2009
GRACE [trends]	2002 – 201? (ongoing)

GLAS–CISM Bilinear differences for CISM file cism_usrf_yr_2007.800000.txt

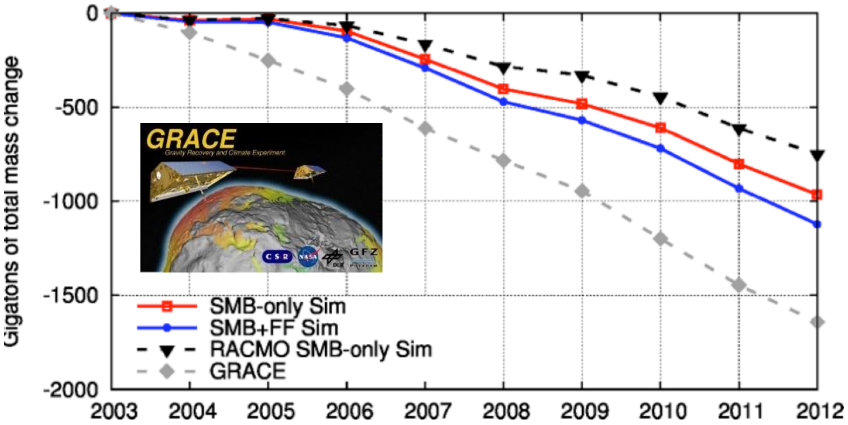


bilinear_compare.pro Mon Dec 14 14:32:25 2015

Histogram of GLAS-CISM Bilinear differences for CISM file cism_usrf_yr_2007.800000.txt



bilinear_compare.pro Mon Dec 14 14:32:25 2015



Surface elevation predictions (states) agree pretty well with ***GLAS (Geoscience Laser Altimeter System) aboard ICESat***: ***mean differences*** are ***<1 m***

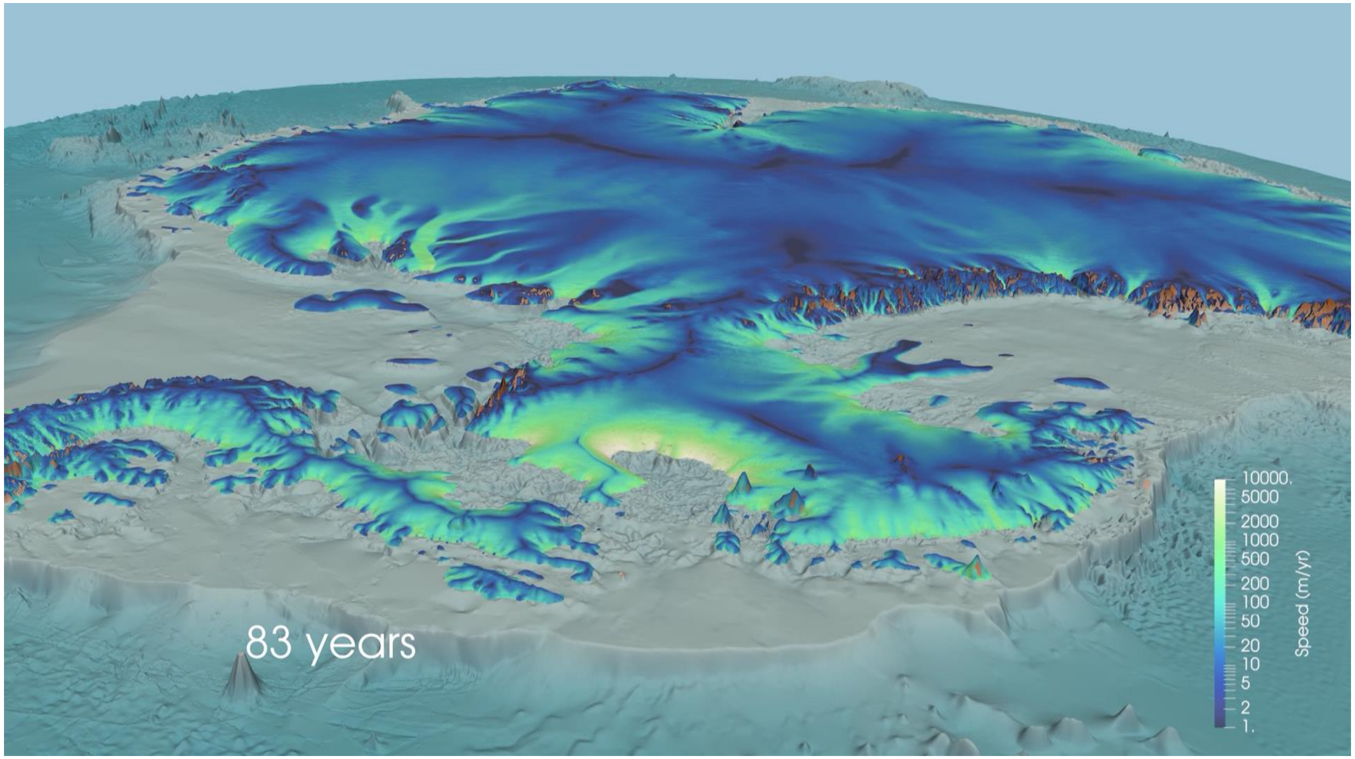
Forcings**:**

- *SMB-only*: Mass change computed by solving an ISM forced w/ RACMO SMB (2003-2012)
- *SMB+FF*: Mass change computed as in SMB-only with additional flux term on significant ice streams
- *RACMO*: mass change computed directly from SMB without using an ice sheet model

* S. Price *et al. GMD* (2017). **van Angelen et al. (*Surv. Geophys.*, 2013), Enderlin et al. *GRL* (2014)

ABUMIP*-Antarctica Experiment

Basic idea: instantaneously remove all ice shelves and see what happens in the next 200 years, preventing any floating ice from ever forming again
→ Provides an ***extreme upper bound*** on SLR contributions from Antarctica



~32M unknowns solved for on **6400 procs**, with average model throughput of **~120 simulated yrs/wall clock day.**

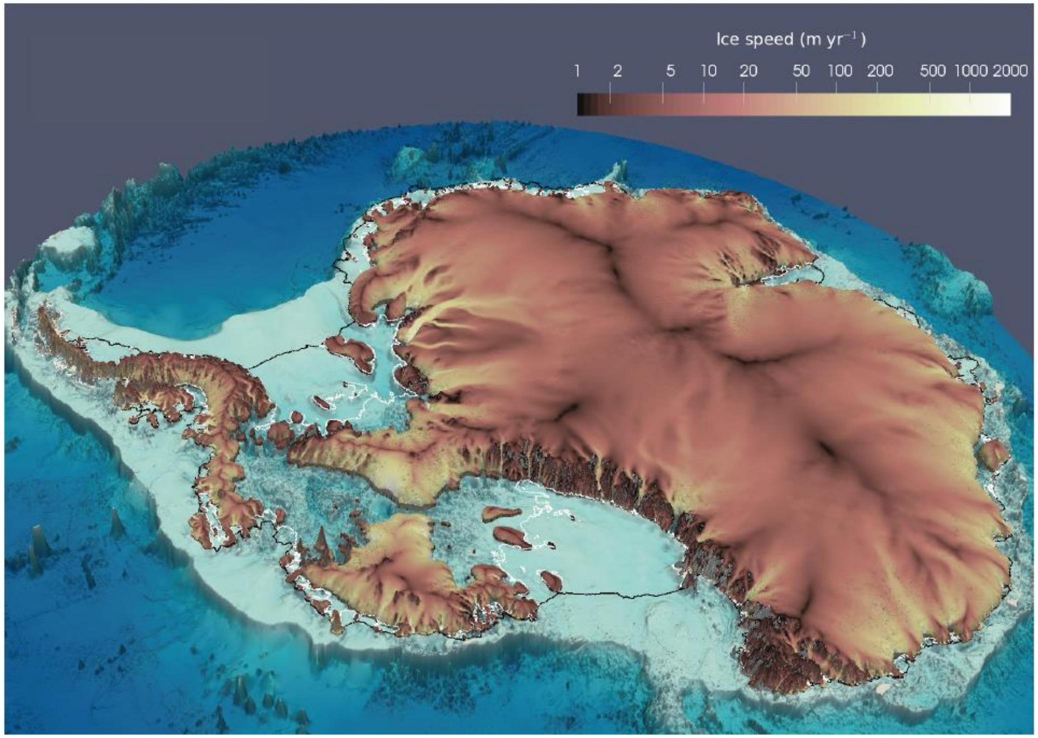
Movie Above: 200 year MALI Antarctic ice sheet simulation after instantaneous removal of all floating ice shelves

Courtesy of M. Hoffman, S. Price, T. Zhang. N. Woods, J. Patchet (LANL)

* Antarctic BUttressing Model Intercomparison Project

ABUMIP*-Antarctica Experiment

<u>**Basic idea**</u>: instantaneously ***remove all ice shelves*** and see what happens in the next 200 years, preventing any floating ice from ever forming again
→ Provides an ***extreme upper bound*** on SLR contributions from Antarctica



~32M unknowns solved for on **6400 procs**, with average model throughput of **~120 simulated yrs/wall clock day.**

Figure Above: Antarctic ice sheet simulation after instantaneous removal of all floating ice shelves at year 200

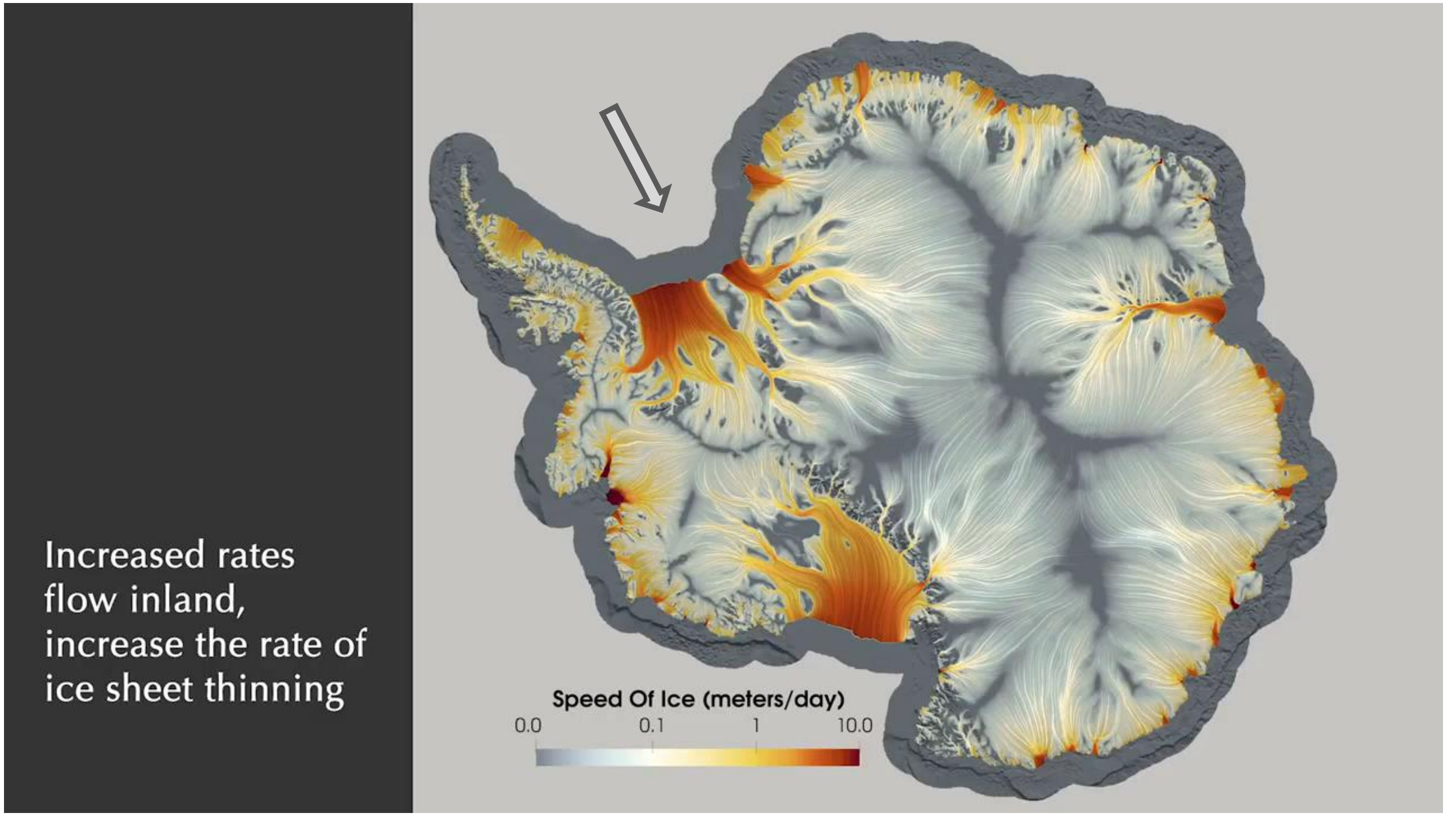
Courtesy of M. Hoffman, S. Price, T. Zhang. N. Woods, J. Patchet (LANL)

* Antarctic BUttressing Model Intercomparison Project

LARMIP*-Antarctica Experiment

Similar to control run (forced with historical observations) in most parts of Antarctica, but includes ***warmer ocean water*** flowing into the cavity beneath the ***Filchner-Ronne Ice Shelf*** → provides example of ice sheet's response to ***aggressive melting and thinning***

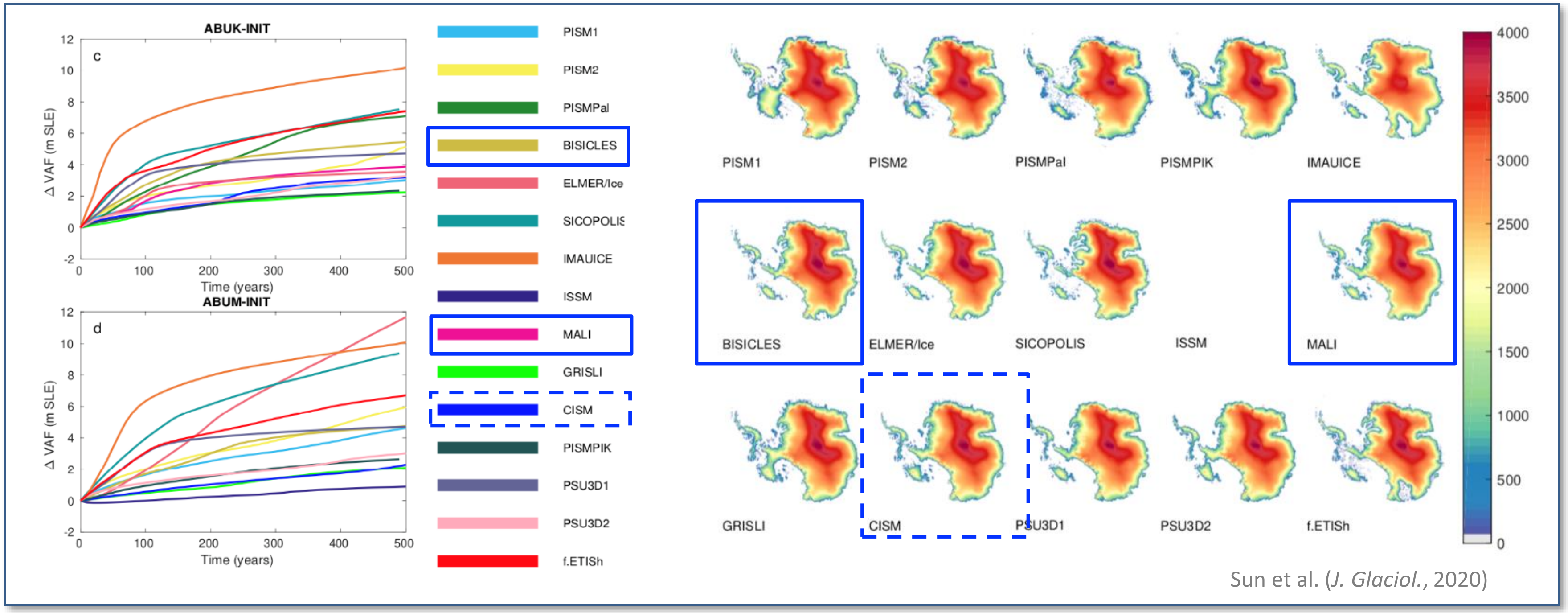
https://www.youtube.com/watch?v=Wt0TvNjYsOs&feature=youtu.be



Increased rates flow inland, increase the rate of ice sheet thinning

* Linear Antarctic Response Model Intercomparison Project

Simulations: Ice Sheets & SLR under ISMIP6*



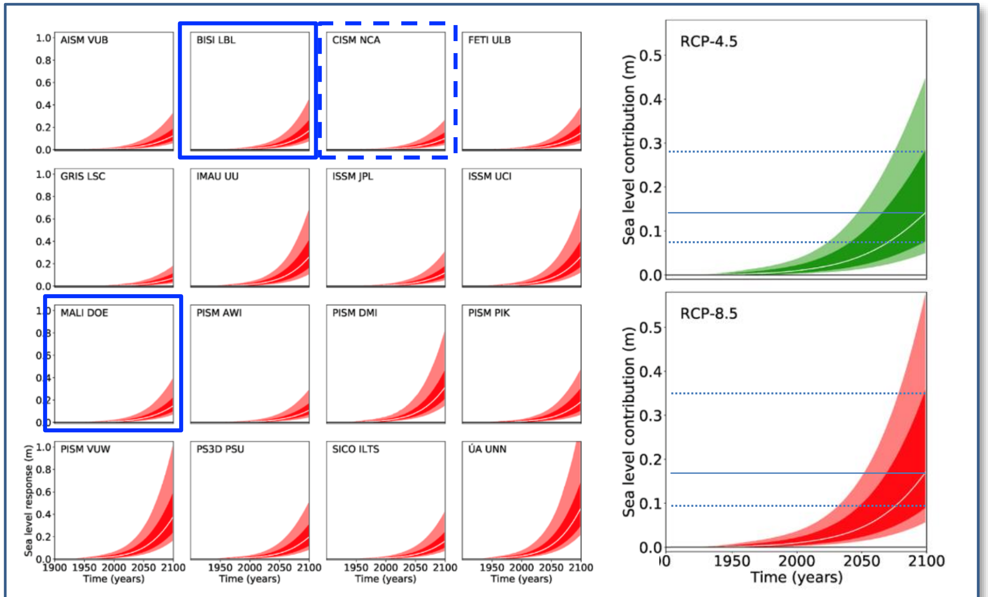
Sun et al. (*J. Glaciol.*, 2020)

Above: Future Antarctica sea level contribution under rapid ice shelf melting (LARMIP) and ice shelf collapse (ABUMIP) for ensemble of models.

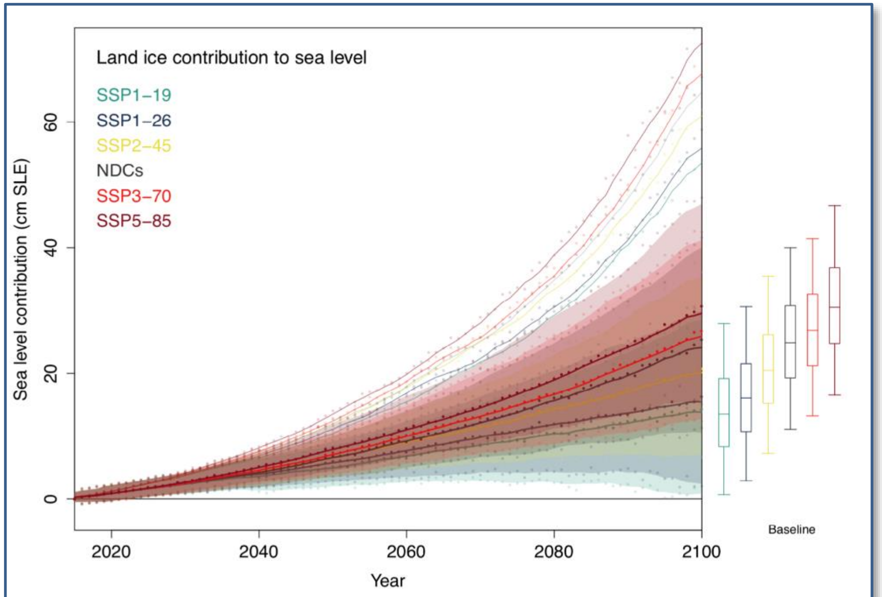
- ~20% of ice sheet contributions from **U.S. DOE-developed models**
- Most non-DOE models are **2D *ad hoc* hybrids** or are run at **relatively coarse resolution**

* Ice Sheet Model Intercomparison Project

Simulations: Ice Sheets & SLR under ISMIP6*

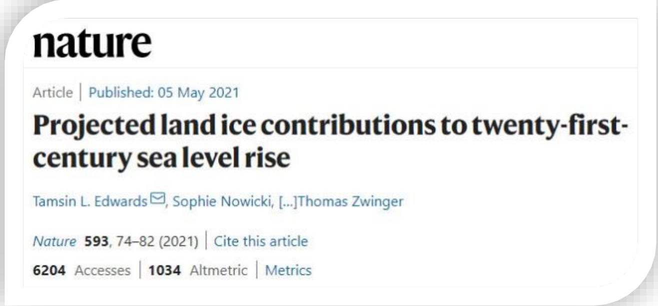


Levermann et al. (*Earth Sys. Dyn.*, 2020)



Edwards et al. (*Nature*, 2021)

Above: Sea-level contributions from Antarctica under different emissions (RCP) scenarios



nature

Article | Published: 05 May 2021

Projected land ice contributions to twenty-first-century sea level rise

Tamsin L. Edwards, Sophie Nowicki, [...]Thomas Zwinger

Nature **593**, 74–82 (2021) | Cite this article

6204 Accesses | **1034** Altmetric | Metrics

5 May 2021

Limit global warming to 1.5°C and halve the land ice contribution to sea level this century

KING'S College LONDON

New research from a large international community of scientists predicts that sea level rise from the melting of ice could be halved this century if we meet the Paris Agreement target of limiting warming to 1.5°C.

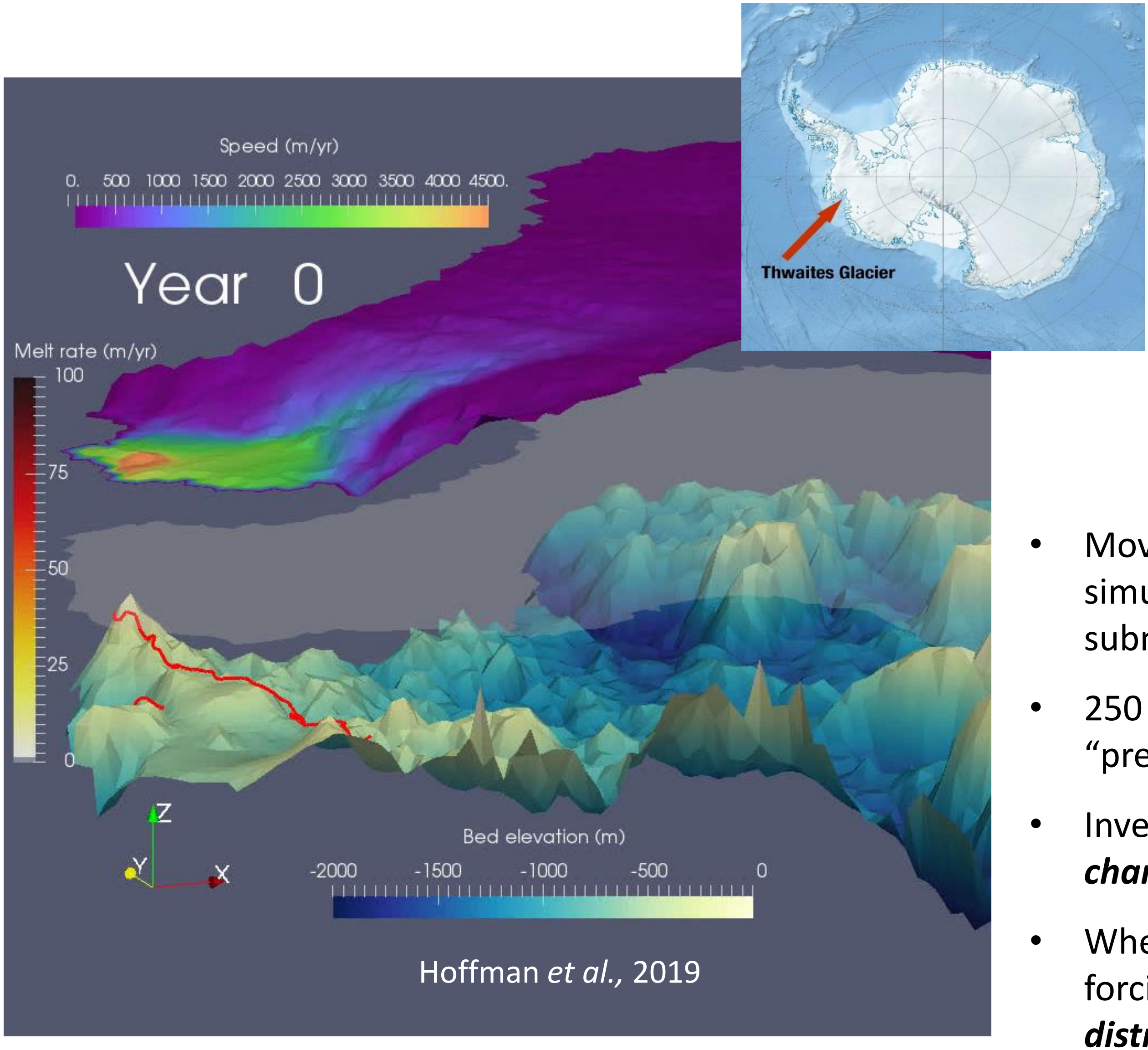
The New York Times

Emissions Cuts Could Drop the Impact of Melting Ice on Oceans by Half

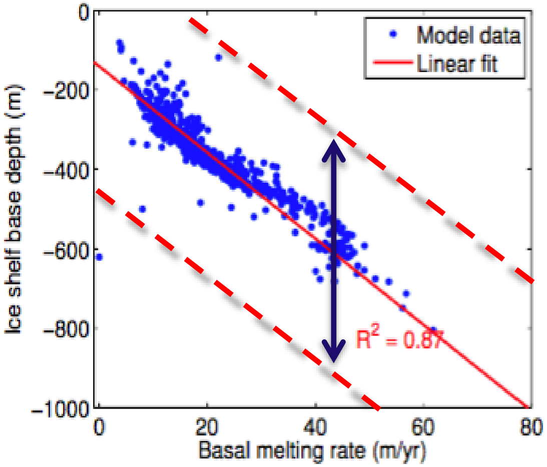
A new study said that limiting warming to 1.5 degrees Celsius could reduce sea level rise from melting ice sheets from about 10 inches to about five by 2100.

* Ice Sheet Model Intercomparison Project

MALI Thwaites Glacier Simulation



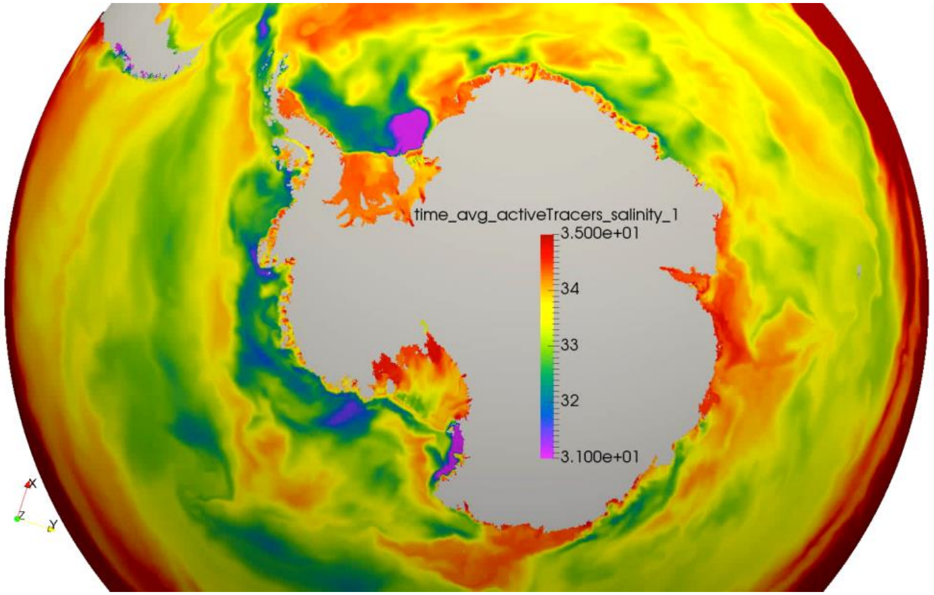
Hoffman *et al.*, 2019



- 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.

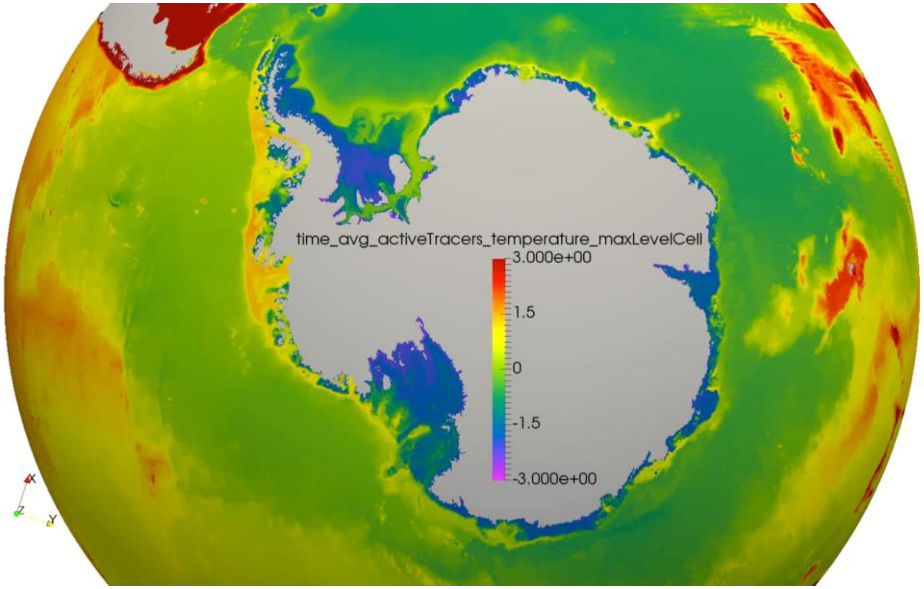
* CDW = Circumpolar Deep Water.

MALI & E3SM Coupling



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).



Ocean Bottom Temperature

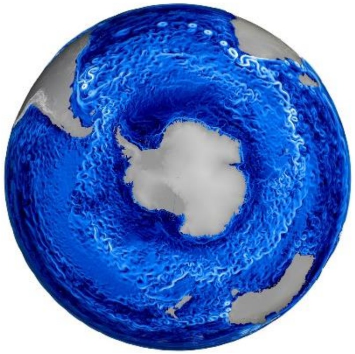
MALI is (partially) coupled to E3SM and currently supports **static ice shelves** and **fixed grounding lines** (enabling dynamic ice shelves is WIP).

Top: *sea-surface salinity*

Right: *ocean bottom temperature*

Outline

1. Background
   - Motivation for climate & land-ice modeling
   - ISMs, ESMs & projects
   - Land-ice equations
   - Our codes: ALI, MALI
2. Algorithms and software
   - Discretization & meshes
   - Nonlinear solvers
   - Linear solvers
   - Performance-portability
   - Ice sheet initialization
   - Towards UQ
3. Simulations
4. **Summary**



Summary

- ***Actionable projections of climate change*** and ***sea-level rise impacts*** are important worldwide!
- A ***mature ice-sheet modeling capability*** (high-fidelity, high-performance) was developed as a part of the PISCEES & ProSPect SciDAC projects. 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 Land-Ice*** written using the libraries of the *Trilinos* libraries.
  - ***Coupling*** of *Albany Land-Ice* to *MPAS LI* codes for transient simulations of ice sheet evolution.
  - Some ***advanced concepts*** in ice sheet modeling: ice sheet initialization/ inversion.
- Related capabilities on the E3SM side are rapidly ***maturing***.
- Ongoing projects are focusing on the remaining work (physics, coupling, uncertainty quantification frameworks) necessary to provide ***sea-level rise projections and uncertainties***.

References

https://www.sandia.gov/~ikalash/journal.html

[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)

https://www.sandia.gov/~ikalash/journal.html

[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", *J. HPC Appl.* (2018), 1-23.

References (cont'd)

https://www.sandia.gov/~ikalash/journal.html

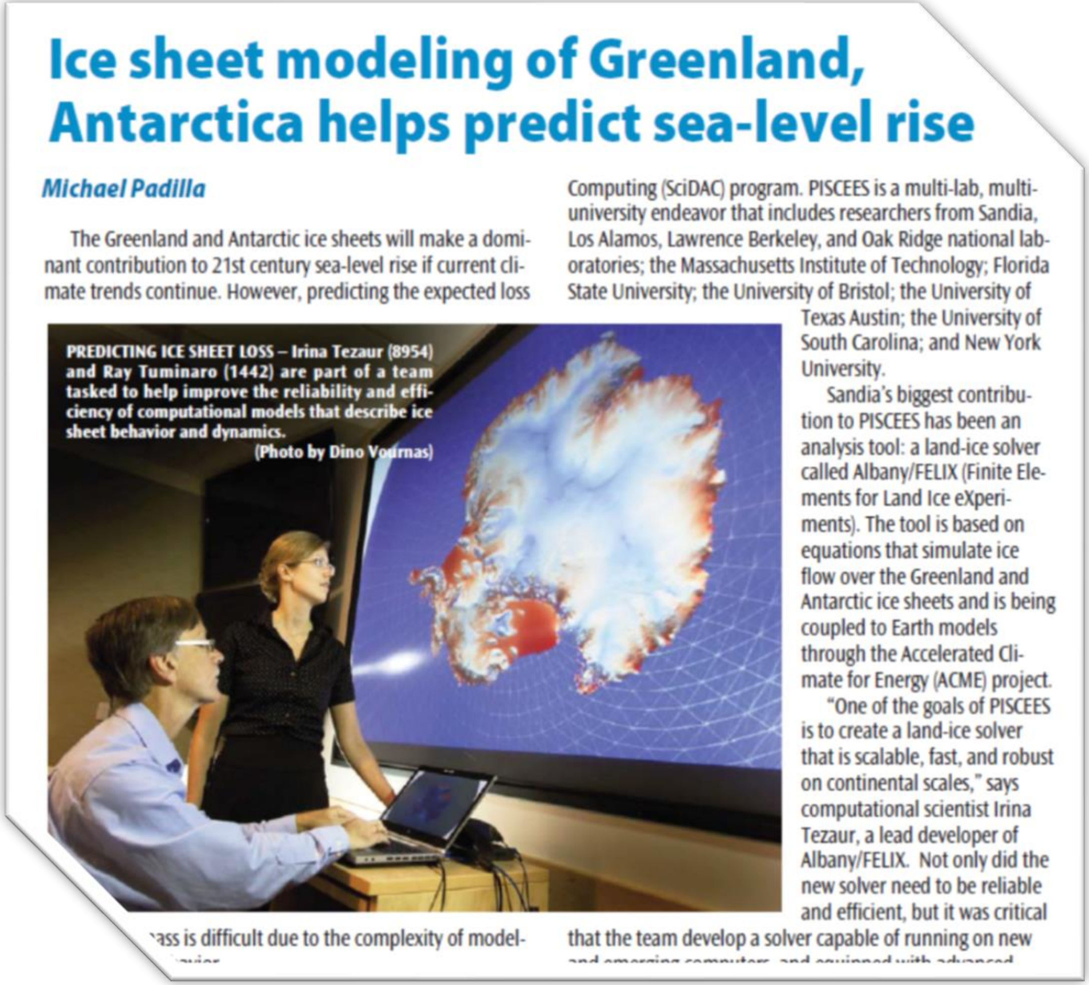
[15] J. Watkins, **I. Tezaur**, I. Demeshko. "A study on the performance portability of the finite element assembly process within the Albany land ice solver", E. van Brummelen, A. Corsini, S. Perotto, G. Rozza, eds. *Numerical Methods for Flows: FEF 2017 Selected Contributions*, Elsevier, 2019.

[16] K. Evans, J. Kennedy, D. Lu, M. Forrester, S. Price, J. Fyke, A. Bennett, M. Hoffman, **I. Tezaur**, C. Zender, M. Vizcaino. "LIVVkit 2.1: Automated and extensible ice sheet model validation", *Geosci. Model Develop* 12 (2019) 1067-1086.

[17] M. Hoffman, M. Perego, S. Price, W. Lipscomb, D. Jacobsen, **I. Tezaur**, A. Salinger, R. Tuminaro, T. Zhang, "MPAS-Albany Land Ice (MALI): A variable resolution ice sheet model for Earth system modeling using Voronoi grid", *Geosci. Model Develop* 11 (2018) 3747-3780.

[18] M. Hoffman, X. Asay-Davis, S. Price, J. Fyke, M. Perego. "Effect of subshelf melt variability on sea level rise contribution from Thwaites Glacier, Antarctica". *JGR Earth Surface* 124 (2019) 2798-2822.

Sandia Land-Ice Work 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

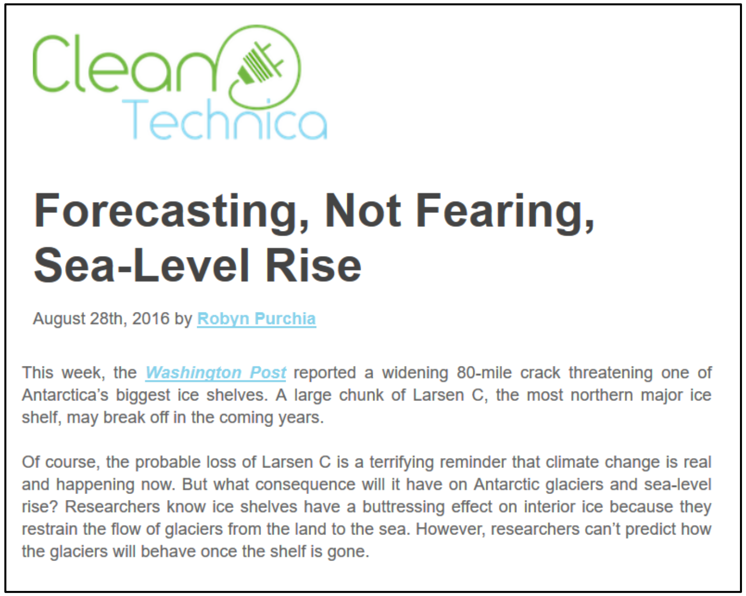
PREDICTING ICE SHEET LOSS – Irina Tezaur (8954) and Ray Tuminaro (1442) are part of a team tasked to help improve the reliability and efficiency of computational models that describe ice sheet behavior and dynamics.
(Photo by Dino Vournas)

ass is difficult due to the complexity of model-
avior

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



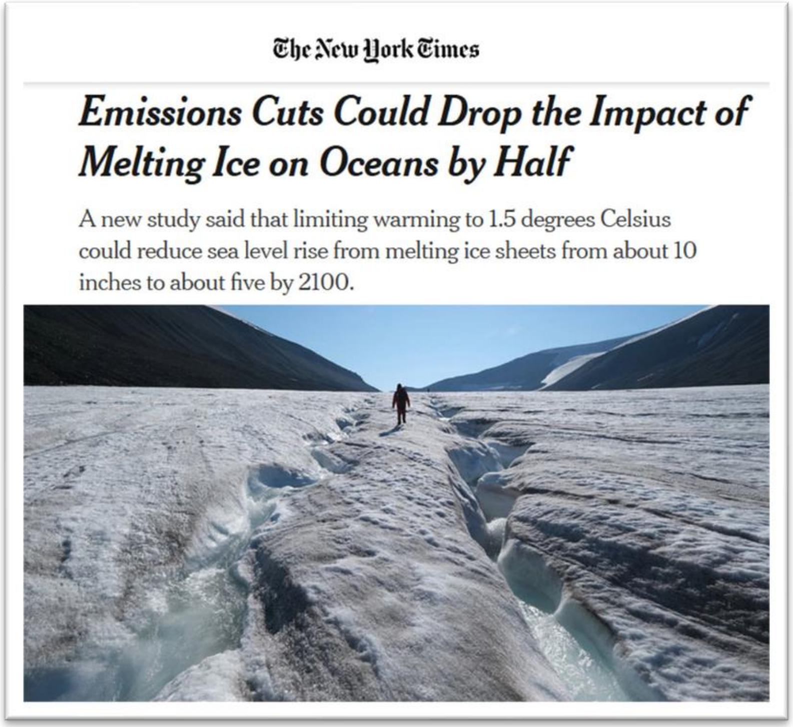
CleanTechnica

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.



The New York Times

Emissions Cuts Could Drop the Impact of Melting Ice on Oceans by Half

A new study said that limiting warming to 1.5 degrees Celsius could reduce sea level rise from melting ice sheets from about 10 inches to about five by 2100.

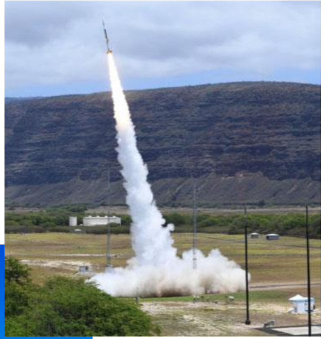
https://www.sandia.gov/~ikalash

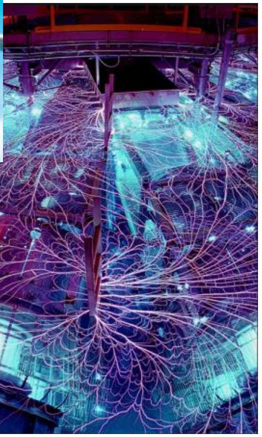
Backup Slides

Careers at Sandia

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

Sandia
National
Laboratories





- Sandia is a **multidisciplinary** national lab and Federally Funded Research & Development Center (FFRDC).
- Contractor for U.S. DOE's National Nuclear Security Administration (**NNSA**).
- **Two main sites**: Albuquerque, NM and Livermore, CA

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 applied math/computational science and impact ***mission-critical applications***.
  - Great ***work/life balance***.

Sandia National Laboratories
MANAGED FOR DOE BY SANDIA CORPORATION
A LOCKHEED MARTIN COMPANY

- ***Opportunities*** at/with Sandia:
  - Interns (including PSAAP)
  - Post docs
  - Several prestigious post doctoral fellowships (von Neumann, Truman, Hruby)
  - Staff

Please see: www.sandia.gov/careers for info about current opportunities.

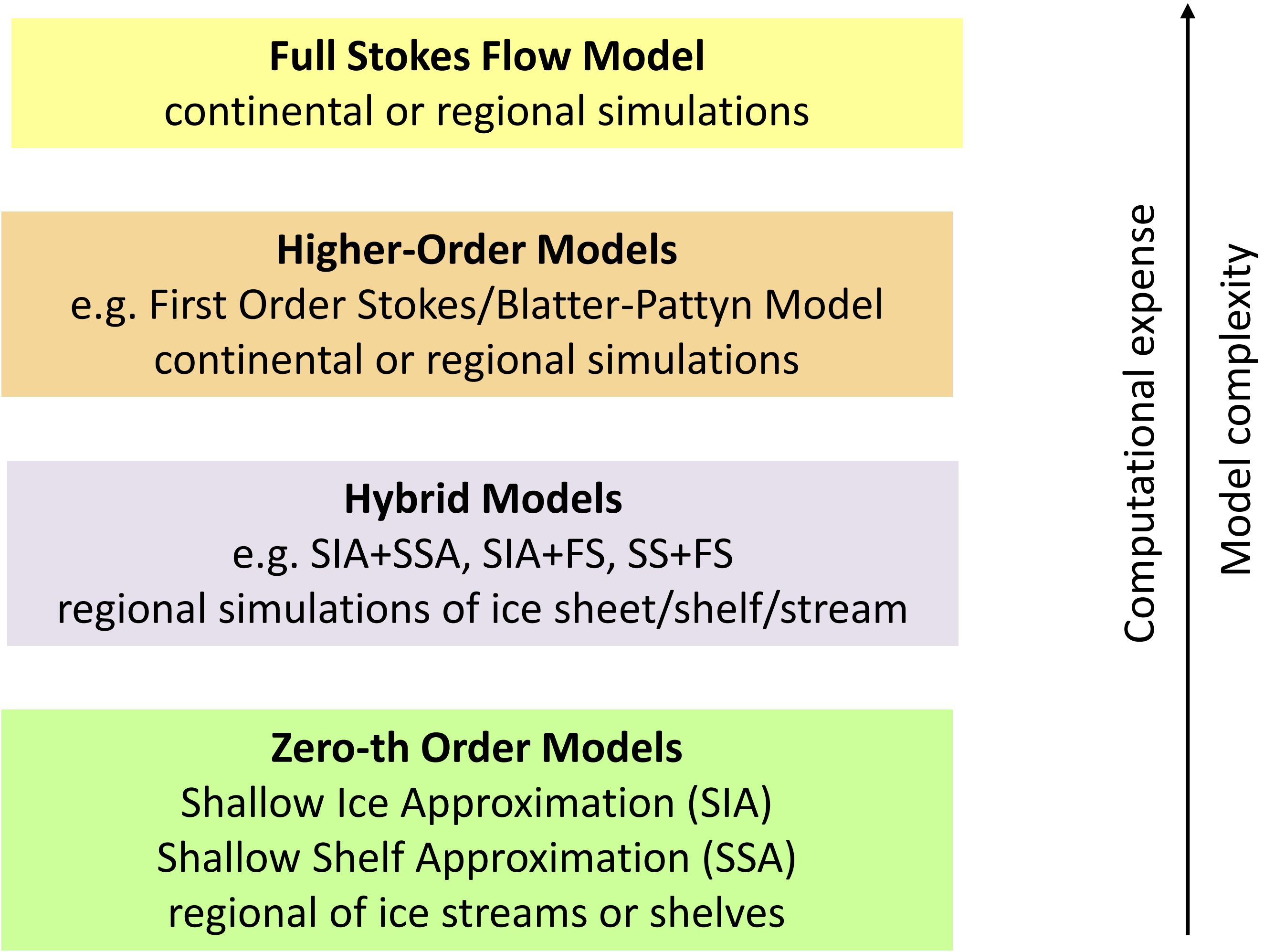
Motivation

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

A Hierarchy of Ice Sheet Models



Full Stokes Flow Model
continental or regional simulations

Higher-Order Models
e.g. First Order Stokes/Blatter-Pattyn Model
continental or regional simulations

Hybrid Models
e.g. SIA+SSA, SIA+FS, SS+FS
regional simulations of ice sheet/shelf/stream

Zero-th Order Models
Shallow Ice Approximation (SIA)
Shallow Shelf Approximation (SSA)
regional of ice streams or shelves

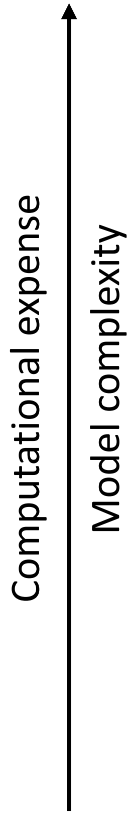
Computational expense

Model complexity

http://www.antarcticglaciers.org/glaciers-and-climate/numerical-ice-sheet-models/hierarchy-ice-sheet-models-introduction/

A Hierarchy of Ice Sheet Models (ISMs)

Model Name	**Terms Kept**	**Comments**	**Validity**
Stokes	All	3D model for $(\boldsymbol{u}, p)$	continental scale
First-Order Stokes/Blatter-Pattyn[1]	$O(\delta)$	3D model for (u_1, u_2)	continental scale
L1L1, L1L2[2]	$O(\delta)$	Depth integrated, 2D models for (u_1, u_2)	Antarctica
Shallow Ice (SIA)[3]	$O(1)$	Depth integrated, 2D model for (u_1, u_2)	grounded ice with frozen bed
Shallow Shelf (SSA)[4]	$O(1)$	Closed form for u_1	shelves or fast sliding grounded ice



[1] Blatter, 1995; Pattyn, 2003. [2] Schoof and Hindmarsh, 2010. [3] Hutter, 1983. [4] Morland, 1987.

A Hierarchy of Ice Sheet Models (ISMs)

Model Name	Terms Kept	Comments	Validity
Stokes	All	3D model for $(\boldsymbol{u}, p)$	continental scale
First-Order Stokes/Blatter-Pattyn[1]	$O(\delta)$	3D model for (u_1, u_2)	continental scale
L1L1, L1L2[2]	$O(\delta)$	Depth integrated, 2D models for (u_1, u_2)	Antarctica
Shallow Ice (SIA)[3]	$O(1)$	Depth integrated, 2D model for (u_1, u_2)	grounded ice with frozen bed
Shallow Shelf (SSA)[4]	$O(1)$	Closed form for u_1	shelves or fast sliding grounded ice

Computational expense

Model complexity

- Stokes flow model is "**gold standard**" but expensive.

[1] Blatter, 1995; Pattyn, 2003. [2] Schoof and Hindmarsh, 2010. [3] Hutter, 1983. [4] Morland, 1987.

A Hierarchy of Ice Sheet Models (ISMs)

Model Name	Terms Kept	Comments	Validity
Stokes	All	3D model for $(\boldsymbol{u}, p)$	continental scale
First-Order Stokes/Blatter-Pattyn[1]	$O(\delta)$	3D model for (u_1, u_2)	continental scale
L1L1, L1L2[2]	$O(\delta)$	Depth integrated, 2D models for (u_1, u_2)	Antarctica
Shallow Ice (SIA)[3]	$O(1)$	Depth integrated, 2D model for (u_1, u_2)	grounded ice with frozen bed
Shallow Shelf (SSA)[4]	$O(1)$	Closed form for u_1	shelves or fast sliding grounded ice

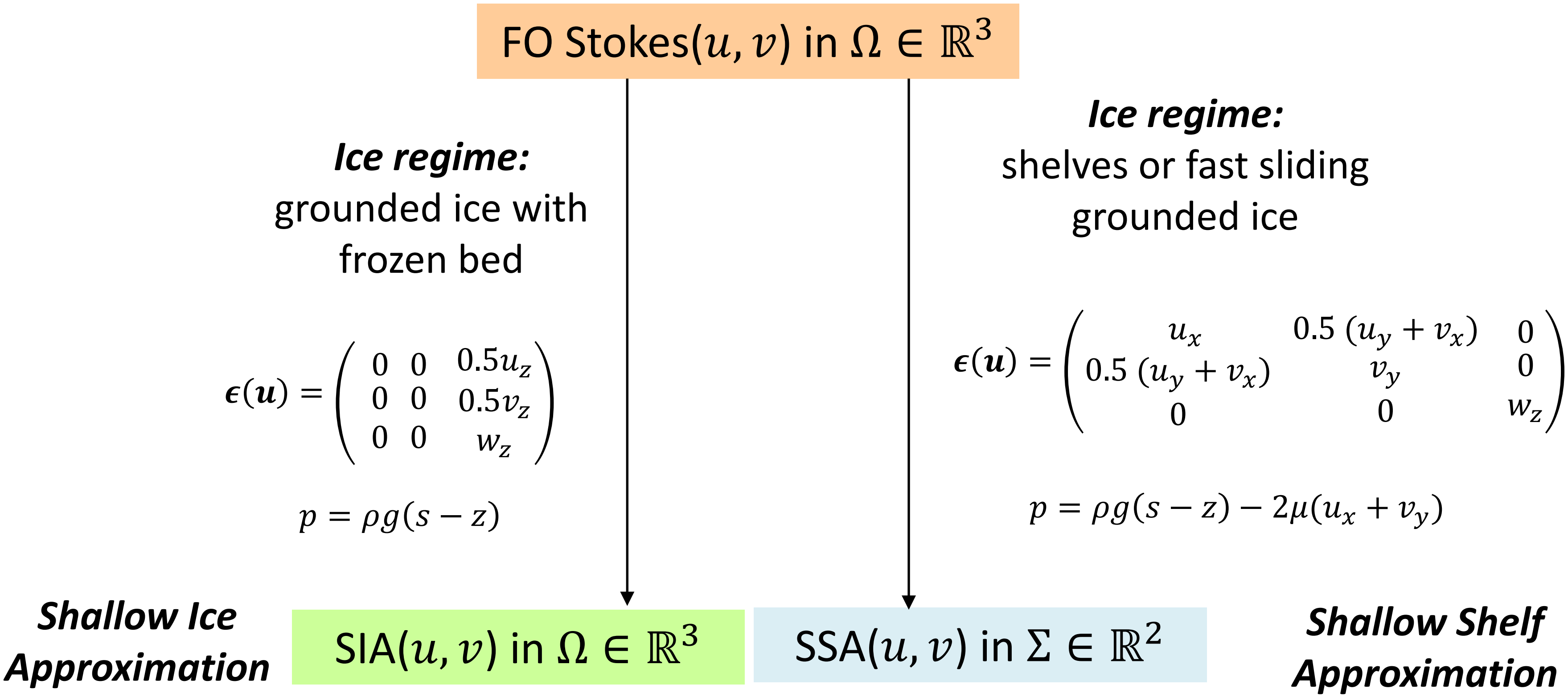
Computational expense

Model complexity

- Stokes flow model is "**gold standard**" but expensive.
- **Simplified models** are derived from full Stokes model and take advantage of the fact that ice sheets are **thin**: $\delta \ll 1$.

[1] Blatter, 1995; Pattyn, 2003. [2] Schoof and Hindmarsh, 2010. [3] Hutter, 1983. [4] Morland, 1987.

Shallow Shelf and Shallow Ice Approximation



FO Stokes(u, v) in $\Omega \in \mathbb{R}^3$

Ice regime:
grounded ice with
frozen bed

$$\epsilon(\boldsymbol{u}) = \begin{pmatrix} 0 & 0 & 0.5u_z \\ 0 & 0 & 0.5v_z \\ 0 & 0 & w_z \end{pmatrix}$$

$$p = \rho g(s - z)$$

Ice regime:
shelves or fast sliding
grounded ice

$$\epsilon(\boldsymbol{u}) = \begin{pmatrix} u_x & 0.5\,(u_y + v_x) & 0 \\ 0.5\,(u_y + v_x) & v_y & 0 \\ 0 & 0 & w_z \end{pmatrix}$$

$$p = \rho g(s - z) - 2\mu(u_x + v_y)$$

Shallow Ice Approximation

SIA(u, v) in $\Omega \in \mathbb{R}^3$

SSA(u, v) in $\Sigma \in \mathbb{R}^2$

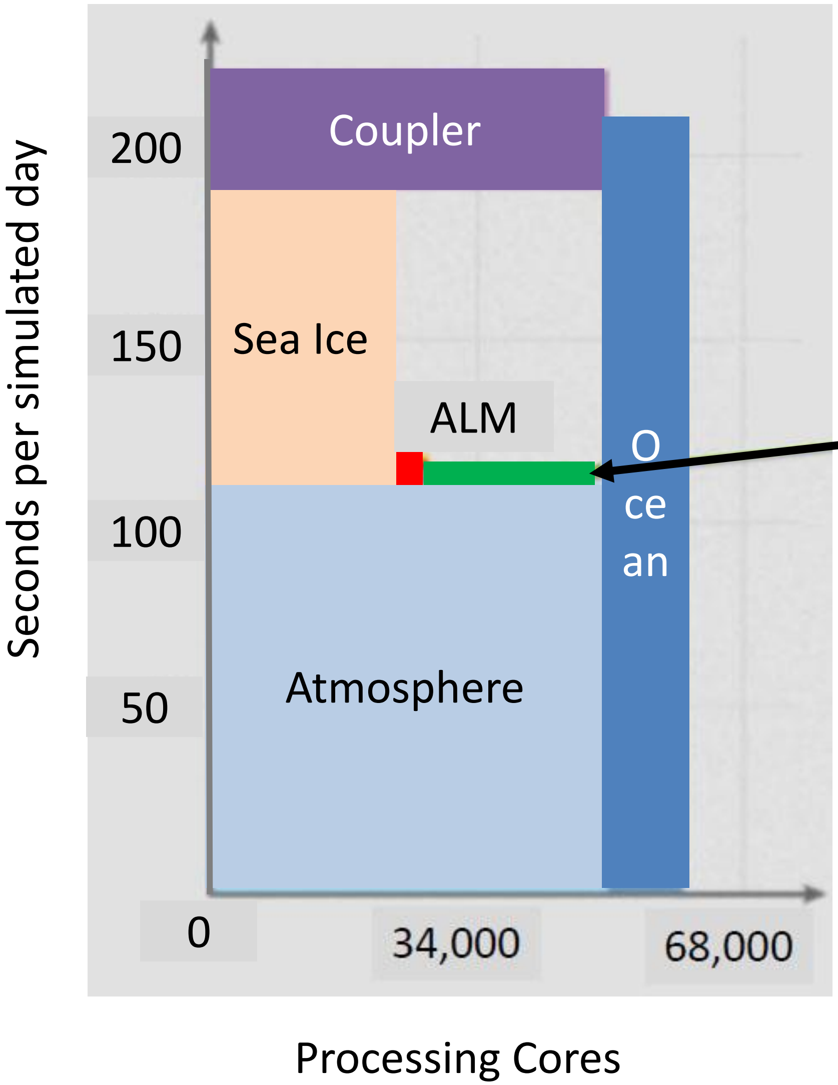
Shallow Shelf Approximation

Discussion:

- **Neither** SIA nor SSA applies at **continental scale**.
- SIA and SSA are referred to as "**zero-th order**" models
- Both models have **two unknowns** (u, v).
- SSA is 2D model obtained by **vertically integrating** the equations.

ISM Computation Cost in ESM

High-res climate model processor layout



DOE Energy Exascale Earth System Model (E3SM)

grid size	component	horizontal	vertical
25km	ATM/LND	0.8M	72
18-6km	OCN/ICE	3.7M	80
2-20km	AIS ISM	1.6M	10

- **ISM throughput:** 1 SYPD (simulated year per wallclock day)
- **ISM cost:** 4M core-hours per simulated year

Numerical & Computational Challenges

- **Mesh adaptivity** close to the grounding line.
- FO Stokes equations are **highly nonlinear**.
- Large, **thin geometries** (thickness up to 4km, horizontal extension 1000s of kms).
  - Gives rise to meshes with **bad aspect ratios** and **poorly conditioned** linear systems.
- **Boundary conditions** pose challenges to solvers.
- **Porting** of software to **new architectures** (hybrid systems, GPUs, etc.).
- **Initialization**/estimation of unknown parameters (basal friction, thickness, etc.).
- **Uncertainty quantification**.
  - Curse of dimensionality!
- **Thickness evolution** (ice advancement/retreat)
  - Sequential coupling with FO Stokes equations gives rise to very small time-steps by CFL condition!
- **Phase changes** (temperature equation).
- **Coupling** to **climate components**.

Mesh Adaptivity

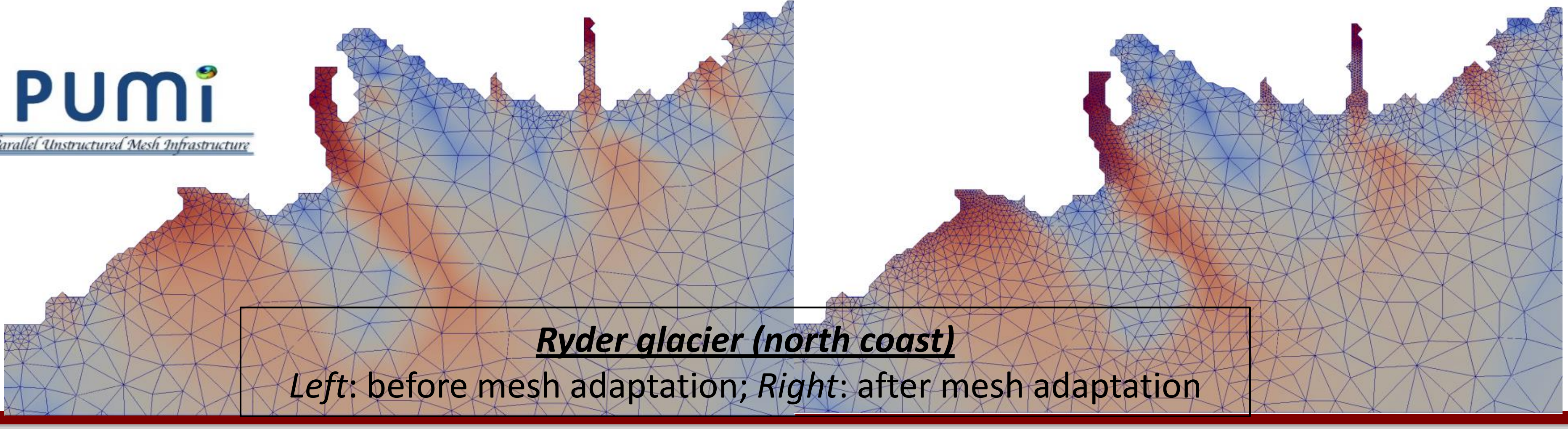
PAALS = Parallel Albany Adaptive Loop with SCOREC*

- 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***.



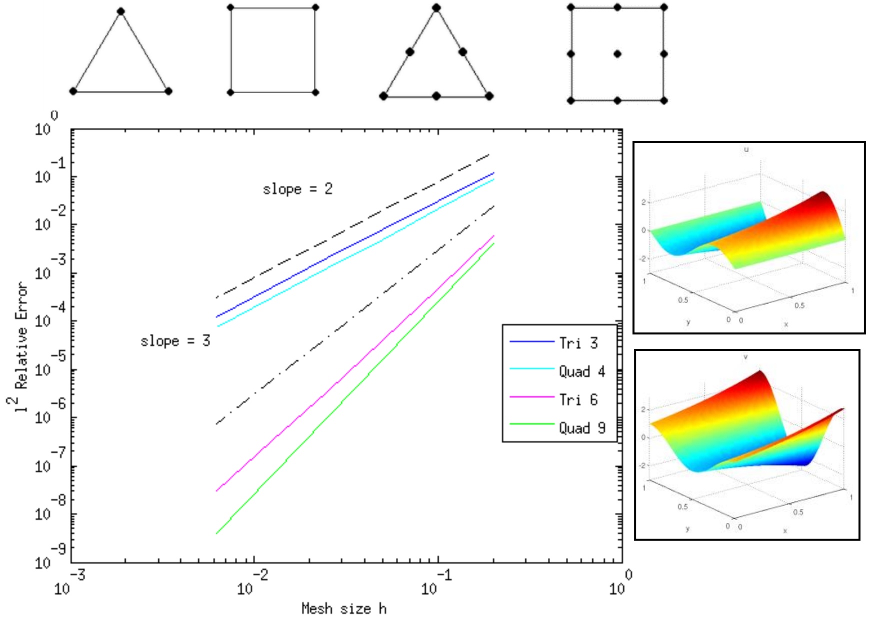
Ryder glacier (north coast)

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

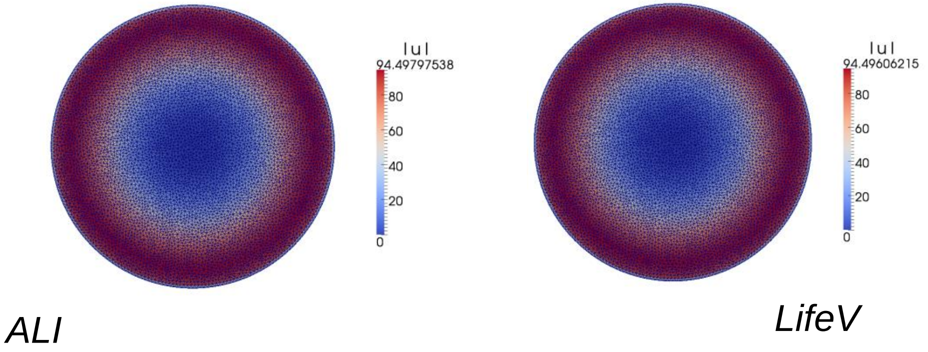
**Super-convergent Patch Recovery: technique for estimating ∇u using quadratic approximation within a patch of elements.

Mesh Convergence Studies

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

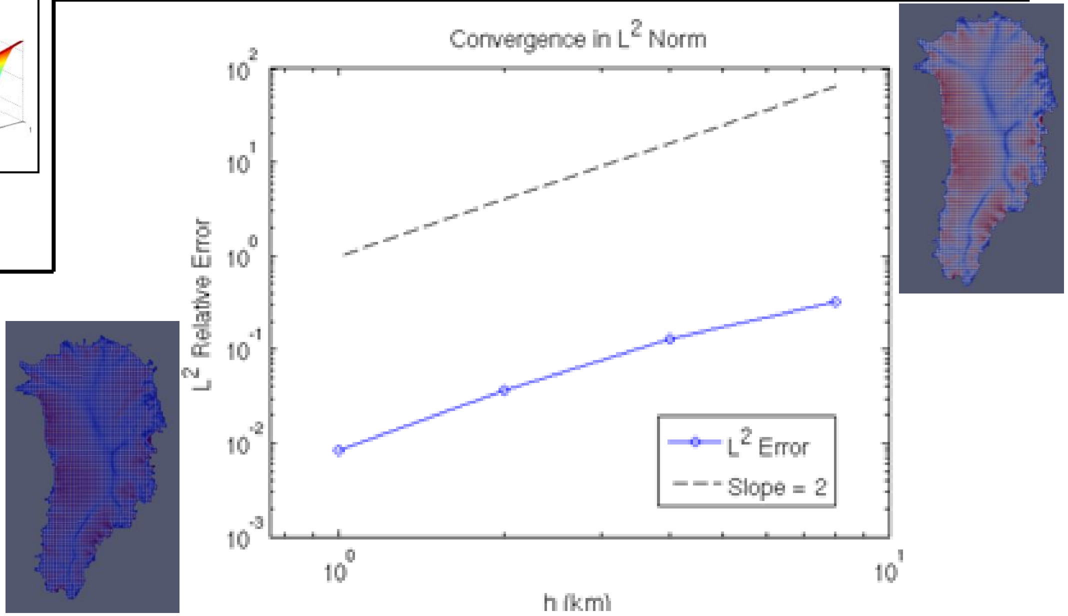


Stage 2: code-to-code comparisons on canonical ice sheet problems.



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

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

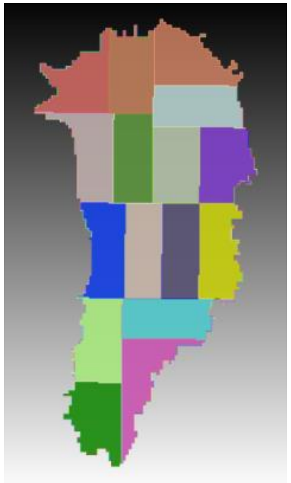


Mesh Partitioning & Vertical Refinement

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).*

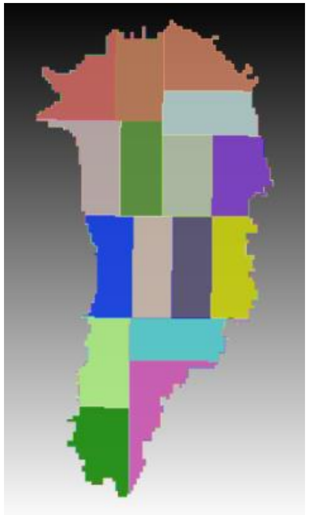


Horiz. res.\vert. layers	**5**	**10**	**20**	**40**	**80**
8km	2.0e-1				
4km	9.0e-2	7.8e-2			
2km	4.6e-2	2.4e-2	2.3e-2		
1km	3.8e-2	8.9e-3	5.5e-3	5.1e-3	
500m	3.7e-2	6.7e-3	1.7e-3	3.9e-4	8.1e-5

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

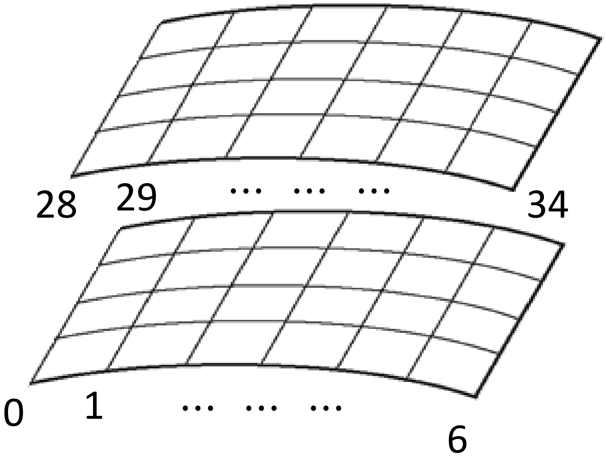
Importance of Node Ordering & Mesh Partitioning

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



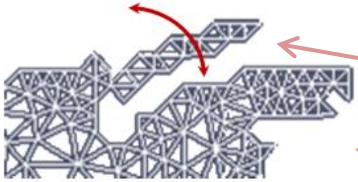
- It is essential that incomplete factorization accurately captures vertical coupling, which is dominant due to anisotropic mesh.

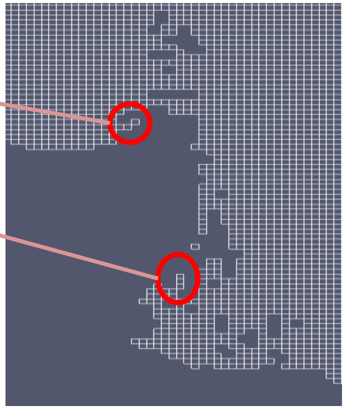
- This is accomplished by:
  - Ensuring all points along a vertically extruded grid line reside within a single processor (“**2D mesh partitioning**”; top right).
  - Ordering the equations such that grid layer k’s nodes are ordered before all dofs associated with grid layer $k+1$ (“**row-wise ordering**”; bottom right).



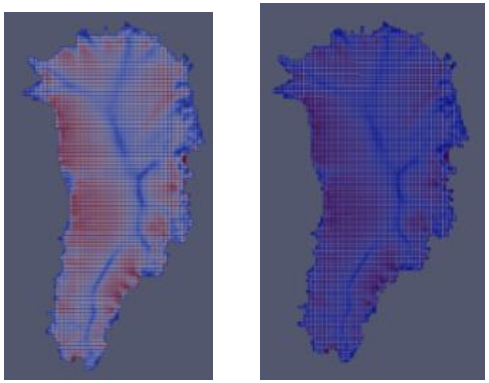
Improved Linear Solver Performance through Hinge Removal

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.

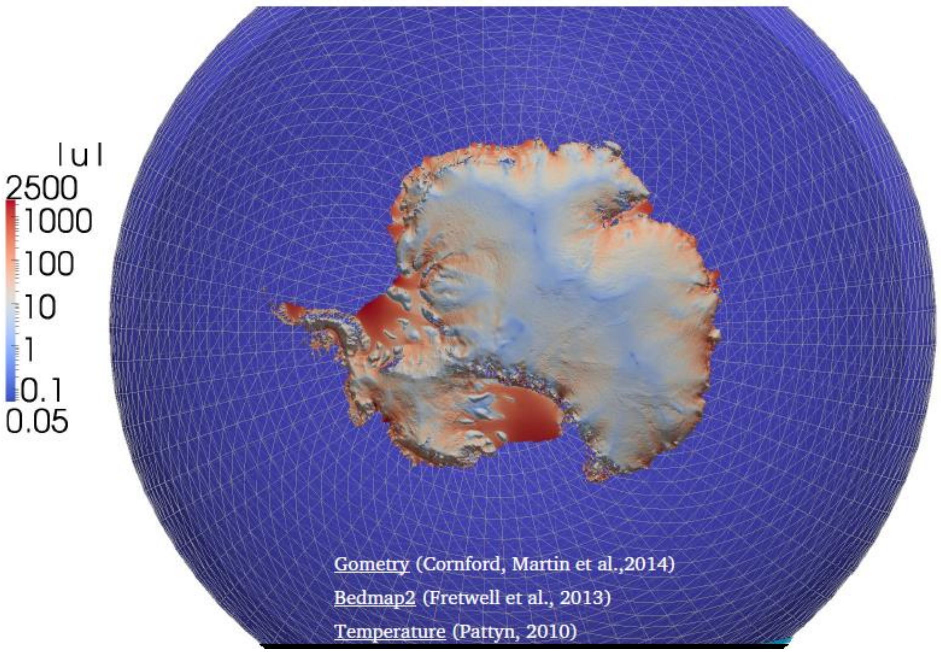


Greenland Problem

Resolution	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

Spherical Grids

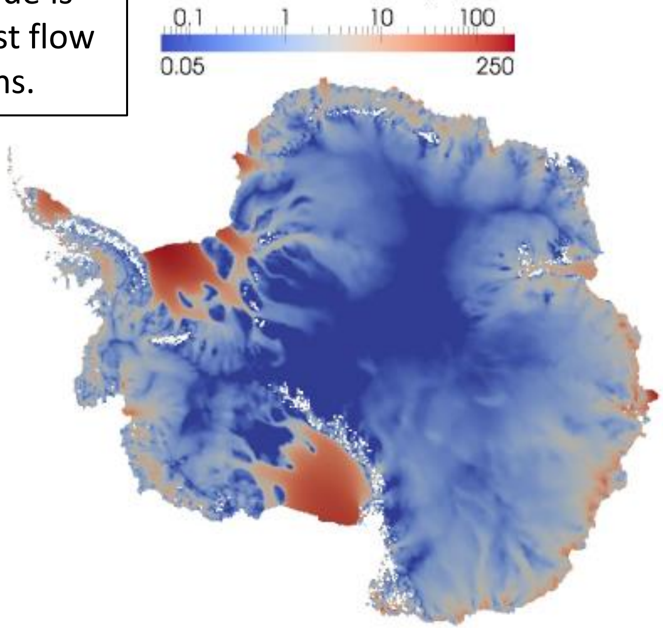
Surface velocity magnitude [m/yr], ice sheet thickness not at scale (100 X)



Gometry (Cornford, Martin et al.,2014)
Bedmap2 (Fretwell et al., 2013)
Temperature (Pattyn, 2010)

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

magnitude of surface velocity difference [m/yr]



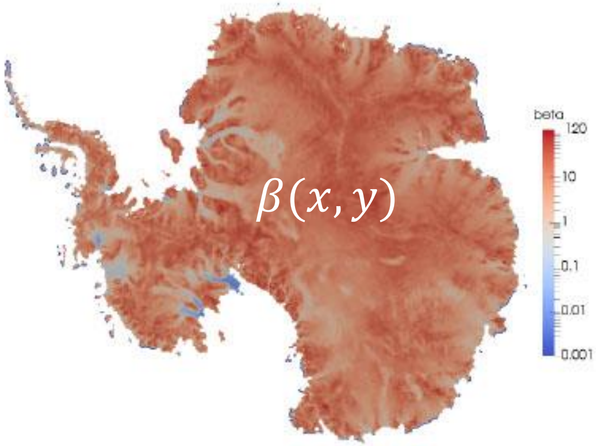
- Current ice sheet models are derived using planar geometries – reasonable, especially for Greenland.
- The effect of Earth's curvature is largely unknown – may be nontrivial for Antarctica.
- We have derived a FO Stokes model on sphere using stereographic projection.

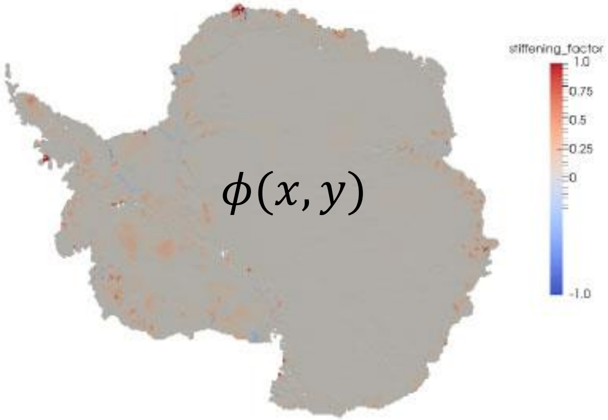
Deterministic Inversion: Stiffening Factor

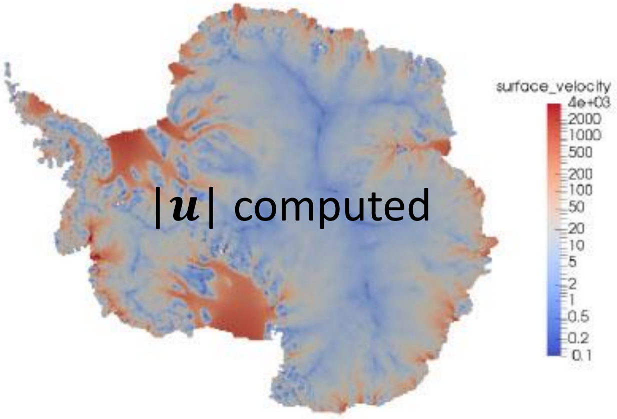
Glen's viscosity with ***stiffening/damage***: $\boxed{\mu^*(x,y,z) = \phi(x,y)\mu(x,y,z)}$

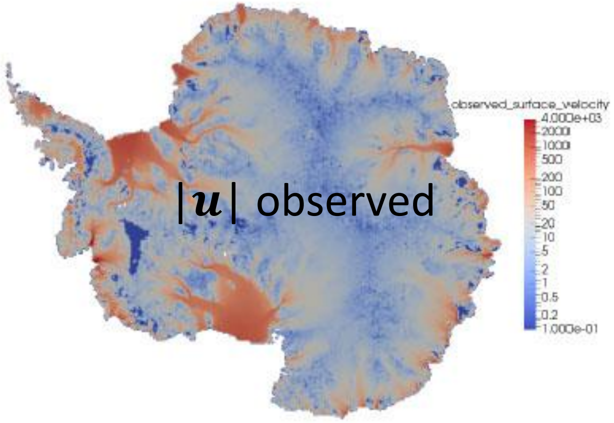
where $\phi(x,y) =$ stiffening/damage factor that accounts for modeling errors in rheology.

AIS inversion for $\beta(x,y)$ and $\phi(x,y)$ **simultaneously**.









UQ Problem Definition

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

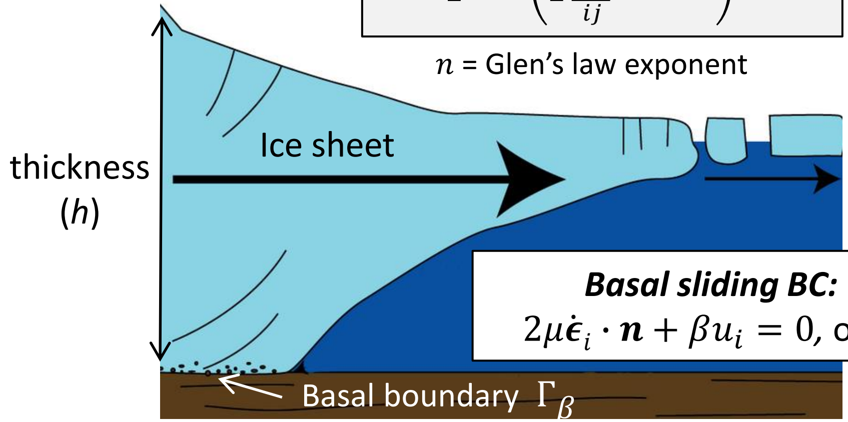
As a first step, we focus on effect of uncertainty in β only.

Sources of uncertainty affecting this QoI include:

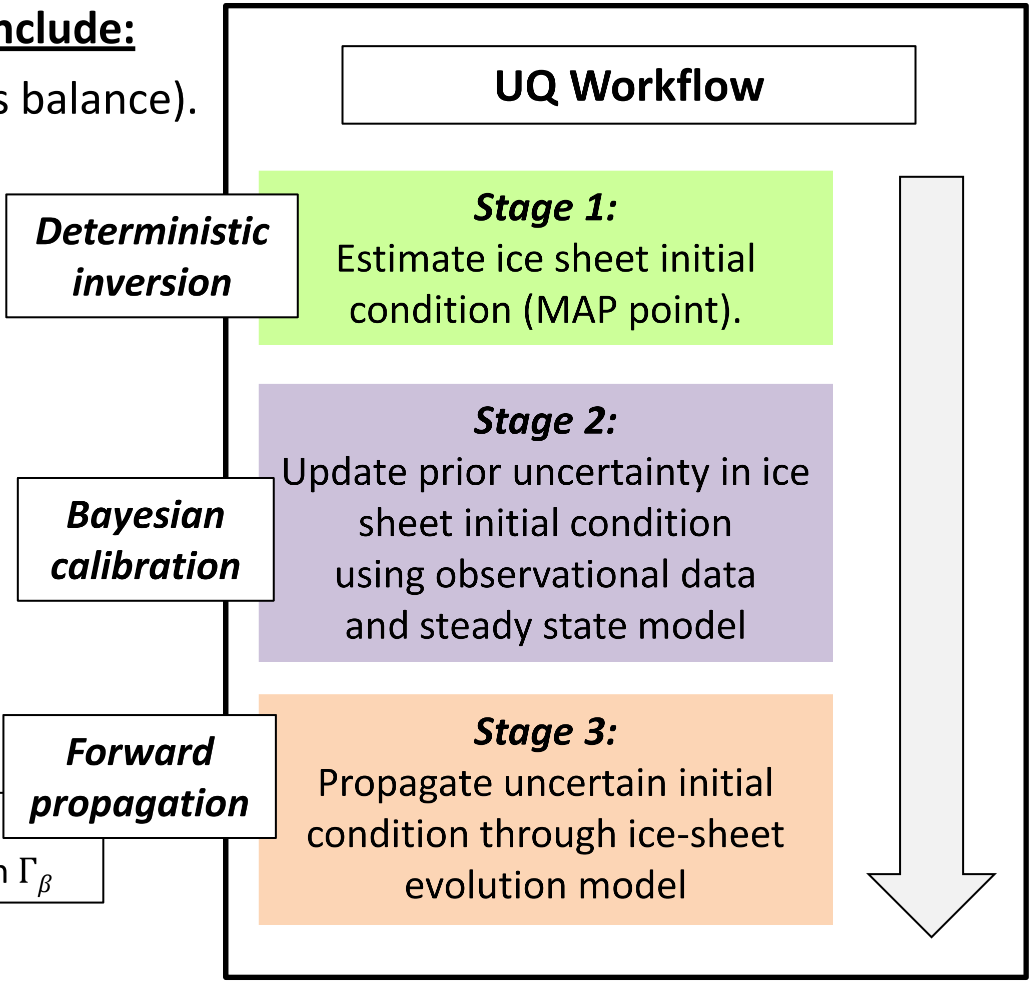
- Climate forcings (e.g., surface mass balance).
- **Basal friction ($\boldsymbol{\beta}$).**
- Ice sheet thickness (h).
- Geothermal heat flux.
- Model parameters (e.g., Glen's flow law exponent).

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

n = Glen's law exponent



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



UQ Workflow

Deterministic inversion

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

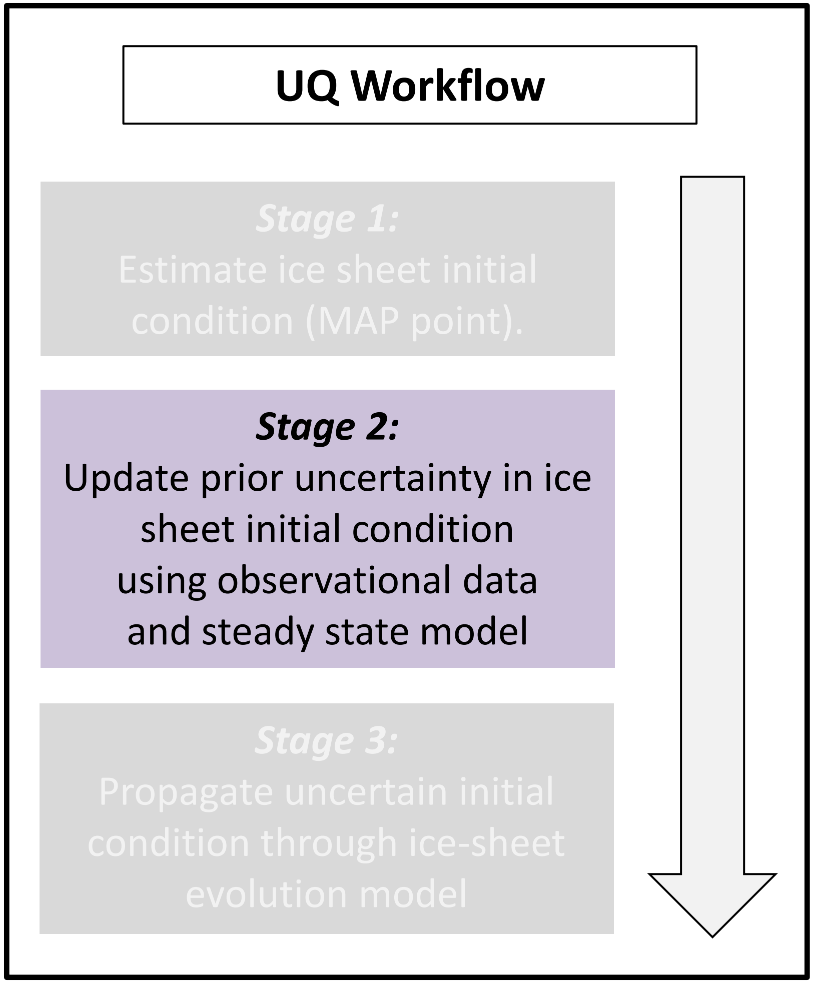
Bayesian calibration

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

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 $\bar{\beta}(\boldsymbol{x})$:

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

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

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

Bayesian Inference Assumptions

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

⇒ 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.

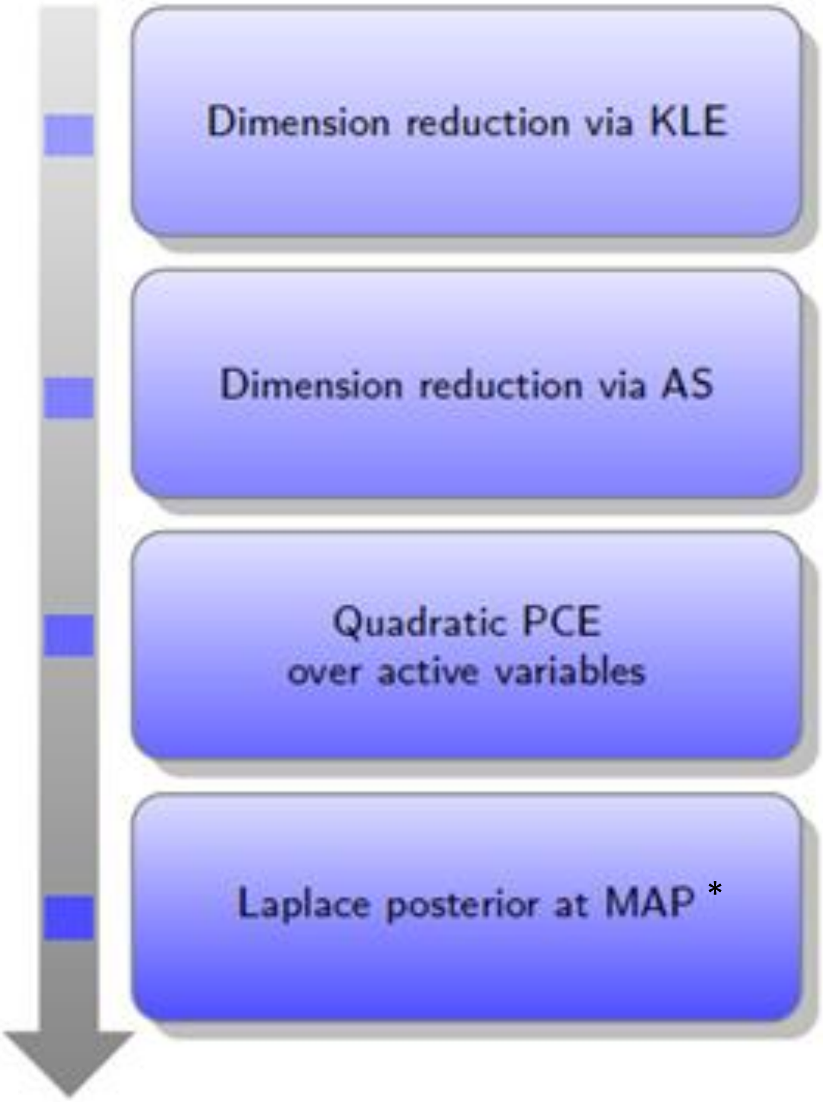
- ***Likelihood*** is: $\hat{\pi}_{\text{lhood}}(\boldsymbol{z}) = e^{-m_{\text{lin}}(\boldsymbol{z})}$

- ***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.

Bayesian Inference Workflow



Dimension reduction via KLE

Dimension reduction via AS

Quadratic PCE
over active variables

Laplace posterior at MAP *

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

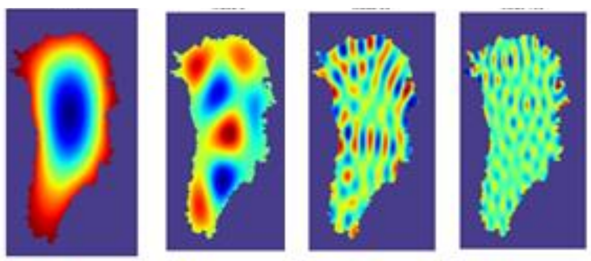
Procedure for computing ***covariance of normal Laplace posterior***, $\boldsymbol{\Gamma}_{\text{post}}$

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

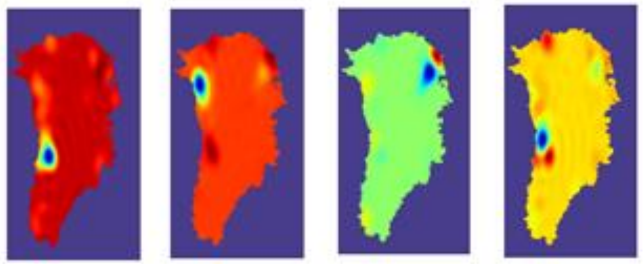
* Bui-Thanh, Ghattas, Martin, Stadler, *SISC*, 2013.

GIS Bayesian Inference via KLE + AS

KLE modes

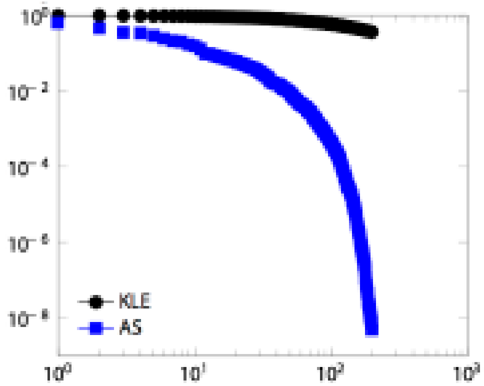


Data-informed (AS) directions (d=73*)



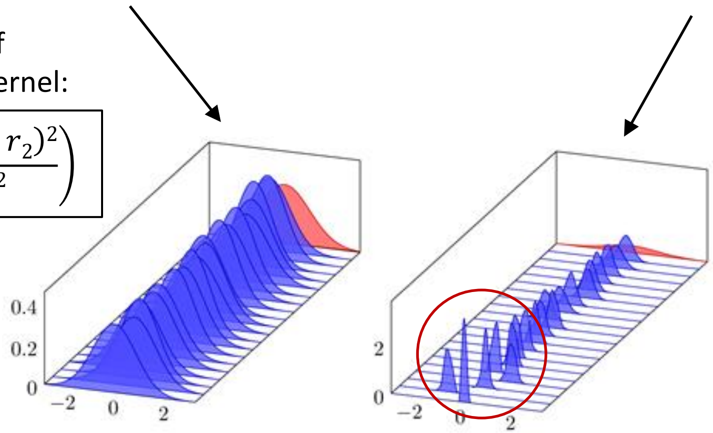
Gradients of mismatch function obtained via ***adjoint solve*** in ALI.

KLE and AS amplitudes

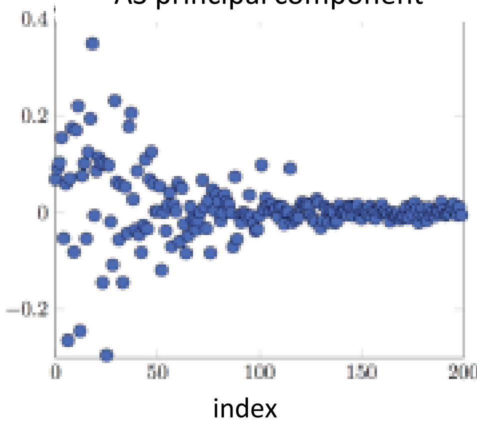


KLE modes = eigenvecs of exponential covariance kernel:

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



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.

Bayesian Inference

- There are many sources of uncertainty, e.g.
  - Climate forcing (e.g., surface mass balance)
  - Basal friction
  - Bedrock topography (noisy and sparse data)
  - Geothermal heat flux
  - Modeling errors
  - Model parameters (e.g., Glen's Flow Law exponent)

Bayesian Inference

- There are many sources of uncertainty, e.g.
  - Climate forcing (e.g., surface mass balance)
  - **Basal friction**
  - Bedrock topography (noisy and sparse data)
  - Geothermal heat flux
  - Modeling errors
  - Model parameters (e.g., Glen's Flow Law exponent)

We focus initially only in uncertainty in **basal friction** β.

Bayesian Inference

- There are many sources of uncertainty, e.g.
  - Climate forcing (e.g., surface mass balance)
  - **Basal friction**
  - Bedrock topography (noisy and sparse data)
  - Geothermal heat flux
  - Modeling errors
  - Model parameters (e.g., Glen's Flow Law exponent)

We focus initially only in uncertainty in **basal friction** β.

- ***Bayes' Theorem:*** assume prior distribution, update using data:

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

Bayesian Inference

- There are many sources of uncertainty, e.g.
  - Climate forcing (e.g., surface mass balance)
  - **Basal friction**
  - Bedrock topography (noisy and sparse data)
  - Geothermal heat flux
  - Modeling errors
  - Model parameters (e.g., Glen's Flow Law exponent)

We focus initially only in uncertainty in **basal friction** β.

- ***Bayes' Theorem:*** assume prior distribution; update using data:

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

- ***Naïve parameterization***: represent each degree of freedom on mesh be an uncertain variable $\beta(\boldsymbol{x}) = (z_1, z_2, \ldots, z_{n_{\mathrm{dof}}})$

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

Bayesian Inference

- There are many sources of uncertainty, e.g.
  - Climate forcing (e.g., surface mass balance)
  - **Basal friction**
  - Bedrock topography (noisy and sparse data)
  - Geothermal heat flux
  - Modeling errors
  - Model parameters (e.g., Glen's Flow Law exponent)

We focus initially only in uncertainty in **basal friction** β.

- ***Bayes' Theorem:*** assume prior distribution; update using data:

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

- ***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**: $n_{\text{dof}} = O(100K)$!

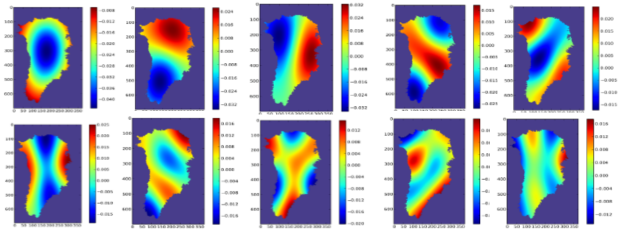
Bayesian Inference

Approach 1: KLE + PCE + MCMC

- ***KLE = Karhunen Loeve Expansion:*** assume $\beta(\boldsymbol{x})$ can be represented in ***reduced basis*** of KLE modes centered around mean $\bar{\beta}(\boldsymbol{x})$:

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

First 10 KLE modes



- ***PCE = Polynomial Chaos Expansion***: create PCE emulator for mismatch (over surface velocity, SMB, thickness) discrepancy.
- ***MCMC = Markov Chain Monte Carlo:*** do MCMC calibration using PCE emulator to infer Maximum A Posteriori (MAP) point.

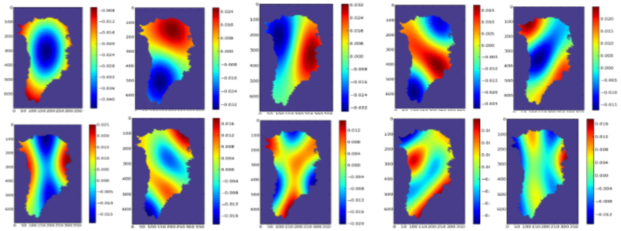
Bayesian Inference

Approach 1: KLE + PCE + MCMC

- ***KLE = Karhunen Loeve Expansion:*** assume $\beta(\boldsymbol{x})$ can be represented in ***reduced basis*** of KLE modes centered around mean $\bar{\beta}(\boldsymbol{x})$:

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

First 10 KLE modes



- ***PCE = Polynomial Chaos Expansion***: create PCE emulator for mismatch (over surface velocity, SMB, thickness) discrepancy.

- ***MCMC = Markov Chain Monte Carlo:*** do MCMC calibration using PCE emulator to infer Maximum A Posteriori (MAP) point.

Upshots:

☺ Can obtain ***arbitrary*** posterior distribution.

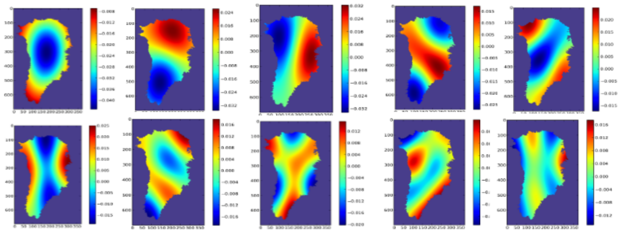
Bayesian Inference

Approach 1: KLE + PCE + MCMC

- ***KLE = Karhunen Loeve Expansion:*** assume $\beta(\boldsymbol{x})$ can be represented in ***reduced basis*** of KLE modes centered around mean $\bar{\beta}(\boldsymbol{x})$:

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

First 10 KLE modes



- ***PCE = Polynomial Chaos Expansion***: create PCE emulator for mismatch (over surface velocity, SMB, thickness) discrepancy.
- ***MCMC = Markov Chain Monte Carlo:*** do MCMC calibration using PCE emulator to infer Maximum A Posteriori (MAP) point.

Upshots:

☺ Can obtain ***arbitrary*** posterior distribution.

Issues:

☹ KLE requires ***correlation length parameter***, which is unknown.

☹ MCMC only lets you use $\boldsymbol{O}(\mathbf{10})$ KLE modes – many more are needed to represent basal friction field $O(1000)$; more modes needed for finer resolution problems.

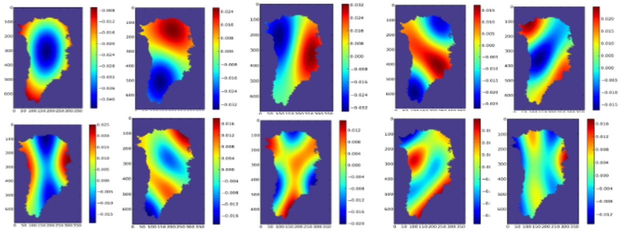
Bayesian Inference

Approach 1: KLE + PCE + MCMC

- ***KLE = Karhunen Loeve Expansion:*** assume $\beta(\boldsymbol{x})$ can be represented in ***reduced basis*** of KLE modes centered around mean $\bar{\beta}(\boldsymbol{x})$:

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

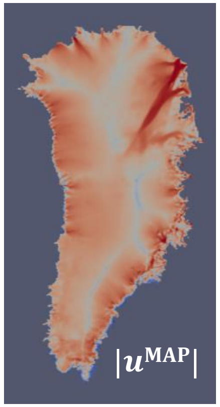
First 10 KLE modes

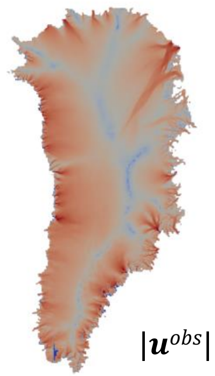


- ***PCE = Polynomial Chaos Expansion***: create PCE emulator for mismatch (over surface velocity, SMB, thickness) discrepancy.

- ***MCMC = Markov Chain Monte Carlo:*** do MCMC calibration using PCE emulator to infer Maximum A Posteriori (MAP) point.

10 KLE modes, 4km GIS: ice too fast (mismatch at MAP point: 1.87 × mismatch at $\bar{\beta}$)





Upshots:

☺ Can obtain ***arbitrary*** posterior distribution.

Issues:

☹ KLE requires ***correlation length parameter***, which is unknown.

☹ MCMC only lets you use $\boldsymbol{O}(\mathbf{10})$ KLE modes – many more are needed to represent basal friction field $O(1000)$; more modes needed for finer resolution problems.

Bayesian Inference

Approach 2: Normal Approximation + Low Rank Laplace Approximation*

- Gaussian prior, likelihood ⇒ ***Gaussian posterior:*** $\pi_{\text{pos}}\left(\boldsymbol{z} \mid \boldsymbol{y}^{\text{obs}}\right) = N(\boldsymbol{z}_{\text{MAP}}, \boldsymbol{\Gamma}_{\text{post}})$

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

Bayesian Inference

Approach 2: Normal Approximation + Low Rank Laplace Approximation*

- Gaussian prior, likelihood ⇒ ***Gaussian posterior:*** $\pi_{\text{pos}}\left(\boldsymbol{z} \mid \boldsymbol{y}^{\text{obs}}\right) = N(\boldsymbol{z}_{\text{MAP}}, \boldsymbol{\Gamma}_{\text{post}})$

- ***Linearize*** parameter-to-observable map around MAP point:

$$\boldsymbol{y}^{\text{obs}} = \boldsymbol{f}(\boldsymbol{z}) + \epsilon \approx \boldsymbol{f}(\boldsymbol{z}_{\text{MAP}}) + \boldsymbol{F}(\boldsymbol{z} - \boldsymbol{z}_{\text{MAP}}) + \epsilon$$

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

Bayesian Inference

Approach 2: Normal Approximation + Low Rank Laplace Approximation*

- Gaussian prior, likelihood ⇒ ***Gaussian posterior:*** $\pi_{\text{pos}}\left(\boldsymbol{z} \,|\, \boldsymbol{y}^{\text{obs}}\right) = N(\boldsymbol{z}_{\text{MAP}}, \boldsymbol{\Gamma}_{\text{post}})$

- ***Linearize*** parameter-to-observable map around MAP point:

$$\boldsymbol{y}^{\text{obs}} = \boldsymbol{f}(\boldsymbol{z}) + \epsilon \approx \boldsymbol{f}(\boldsymbol{z}_{\text{MAP}}) + \boldsymbol{F}(\boldsymbol{z} - \boldsymbol{z}_{\text{MAP}}) + \epsilon$$

- ***Covariance*** of Gaussian ***posterior*** given by:

$$\boxed{\boldsymbol{\Gamma}_{\text{post}} = \left(\boldsymbol{H}_{\text{misfit}}^{\text{PCE}} + \boldsymbol{\Gamma}_{\text{prior}}^{-1}\right)^{-1}}$$

Symbols*:

$\boldsymbol{V}_r, \boldsymbol{D}_r$: eigenvecs, eigenvals of $\widetilde{\boldsymbol{H}}_{\text{misfit}}$

$\widetilde{\boldsymbol{H}}_{\text{misfit}}$ = prior-preconditioned Hessian of data misfit = $\boldsymbol{\Gamma}_{\text{prior}}^{1/2}\boldsymbol{H}_{\text{misfit}}\boldsymbol{\Gamma}_{\text{prior}}^{1/2}$

$\boldsymbol{H}_{\text{misfit}}$ = Gauss-Newton portion of Hessian misfit = $\boldsymbol{F}^{\natural}\boldsymbol{\Gamma}_{\text{obs}}^{-1}\boldsymbol{F}$

$\widetilde{\boldsymbol{V}}_r$ = $\boldsymbol{\Gamma}_{\text{prior}}^{1/2}\boldsymbol{V}_r$, $\widetilde{\boldsymbol{V}}_r^{\Diamond}$ = adjoint of $\widetilde{\boldsymbol{V}}_r$

$\boldsymbol{\Gamma}_{\text{prior}}^{-1}$ = $\boldsymbol{M}^{-1}\boldsymbol{K}$, $\boldsymbol{K}$= Laplace stiffness.

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

Bayesian Inference

Approach 2: Normal Approximation + Low Rank Laplace Approximation*

- Gaussian prior, likelihood ⇒ ***Gaussian posterior:*** $\pi_{\mathrm{pos}}\left(\boldsymbol{z} \mid \boldsymbol{y}^{\mathrm{obs}}\right) = N(\boldsymbol{z}_{\mathrm{MAP}}, \boldsymbol{\Gamma}_{\mathrm{post}})$

- ***Linearize*** parameter-to-observable map around MAP point:

$$\boldsymbol{y}^{\mathrm{obs}} = \boldsymbol{f}(\boldsymbol{z}) + \epsilon \approx \boldsymbol{f}(\boldsymbol{z}_{\mathrm{MAP}}) + \boldsymbol{F}(\boldsymbol{z} - \boldsymbol{z}_{\mathrm{MAP}}) + \epsilon$$

- ***Covariance*** of Gaussian ***posterior*** given by:

$$\boxed{\boldsymbol{\Gamma}_{\mathrm{post}} = \left(\boldsymbol{H}_{\mathrm{misfit}}^{\mathrm{PCE}} + \boldsymbol{\Gamma}_{\mathrm{prior}}^{-1}\right)^{-1}}$$

Symbols*:

$\boldsymbol{V}_r, \boldsymbol{D}_r$: eigenvecs, eigenvals of $\widetilde{\boldsymbol{H}}_{\mathrm{misfit}}$

$\widetilde{\boldsymbol{H}}_{\mathrm{misfit}}$ = prior-preconditioned Hessian of data misfit = $\boldsymbol{\Gamma}_{\mathrm{prior}}^{1/2}\boldsymbol{H}_{\mathrm{misfit}}\boldsymbol{\Gamma}_{\mathrm{prior}}^{1/2}$

$\boldsymbol{H}_{\mathrm{misfit}}$ = Gauss-Newton portion of Hessian misfit = $\boldsymbol{F}^{\natural}\boldsymbol{\Gamma}_{\mathrm{obs}}^{-1}\boldsymbol{F}$

$\widetilde{\boldsymbol{V}}_r$ = $\boldsymbol{\Gamma}_{\mathrm{prior}}^{1/2}\boldsymbol{V}_r$, $\widetilde{\boldsymbol{V}}_r^{\lozenge}$ = adjoint of $\widetilde{\boldsymbol{V}}_r$

$\boldsymbol{\Gamma}_{\mathrm{prior}}^{-1}$ = $\boldsymbol{M}^{-1}\boldsymbol{K}$, $\boldsymbol{K}$= Laplace stiffness.

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

Bayesian Inference

Approach 2: Normal Approximation + Low Rank Laplace Approximation*

- Gaussian prior, likelihood ⇒ ***Gaussian posterior:*** $\pi_{\text{pos}}\left(\boldsymbol{z} \mid \boldsymbol{y}^{\text{obs}}\right) = N(\boldsymbol{z}_{\text{MAP}}, \boldsymbol{\Gamma}_{\text{post}})$

- ***Linearize*** parameter-to-observable map around MAP point:

$$\boldsymbol{y}^{\text{obs}} = \boldsymbol{f}(\boldsymbol{z}) + \epsilon \approx \boldsymbol{f}(\boldsymbol{z}_{\text{MAP}}) + \boldsymbol{F}(\boldsymbol{z} - \boldsymbol{z}_{\text{MAP}}) + \epsilon$$

- ***Covariance*** of Gaussian ***posterior*** given by:

$$\boxed{\boldsymbol{\Gamma}_{\text{post}} = \left(\boldsymbol{H}_{\text{misfit}}^{\text{PCE}} + \boldsymbol{\Gamma}_{\text{prior}}^{-1}\right)^{-1}}$$

Dense!

Symbols*:

$\boldsymbol{V}_r, \boldsymbol{D}_r$: eigenvecs, eigenvals of $\widetilde{\boldsymbol{H}}_{\text{misfit}}$

$\widetilde{\boldsymbol{H}}_{\text{misfit}}$ = prior-preconditioned Hessian of data misfit = $\boldsymbol{\Gamma}_{\text{prior}}^{1/2}\boldsymbol{H}_{\text{misfit}}\boldsymbol{\Gamma}_{\text{prior}}^{1/2}$

$\boldsymbol{H}_{\text{misfit}}$ = Gauss-Newton portion of Hessian misfit = $\boldsymbol{F}^{\natural}\boldsymbol{\Gamma}_{\text{obs}}^{-1}\boldsymbol{F}$

$\widetilde{\boldsymbol{V}}_r$ = $\boldsymbol{\Gamma}_{\text{prior}}^{1/2}\boldsymbol{V}_r$, $\widetilde{\boldsymbol{V}}_r^{\Diamond}$ = adjoint of $\widetilde{\boldsymbol{V}}_r$

$\boldsymbol{\Gamma}_{\text{prior}}^{-1}$ = $\boldsymbol{M}^{-1}\boldsymbol{K}$, $\boldsymbol{K}$= Laplace stiffness.

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

Bayesian Inference

Approach 2: Normal Approximation + Low Rank Laplace Approximation*

- Gaussian prior, likelihood ⇒ ***Gaussian posterior:*** $\pi_{\text{pos}}\left(\boldsymbol{z} \mid \boldsymbol{y}^{\text{obs}}\right) = N(\boldsymbol{z}_{\text{MAP}}, \boldsymbol{\Gamma}_{\text{post}})$

- ***Linearize*** parameter-to-observable map around MAP point:

$$\boldsymbol{y}^{\text{obs}} = \boldsymbol{f}(\boldsymbol{z}) + \epsilon \approx \boldsymbol{f}(\boldsymbol{z}_{\text{MAP}}) + \boldsymbol{F}(\boldsymbol{z} - \boldsymbol{z}_{\text{MAP}}) + \epsilon$$

- ***Covariance*** of Gaussian ***posterior*** given by:

$$\boxed{\boldsymbol{\Gamma}_{\text{post}} = \left(\boldsymbol{H}_{\text{misfit}}^{\text{PCE}} + \boldsymbol{\Gamma}_{\text{prior}}^{-1}\right)^{-1}}$$

Dense!

- ***Low-rank approximation*** of $\boldsymbol{\Gamma}_{\text{post}}$ obtained using Sherman-Morrison-Woodbury formula:

$$\boxed{\boldsymbol{\Gamma}_{\text{post}} \approx \boldsymbol{\Gamma}_{\text{prior}} - \widetilde{\boldsymbol{V}}_r \boldsymbol{D}_r \widetilde{\boldsymbol{V}}_r^{\Diamond}}$$

Symbols*:

$\boldsymbol{V}_r, \boldsymbol{D}_r$: eigenvecs, eigenvals of $\widetilde{\boldsymbol{H}}_{\text{misfit}}$

$\widetilde{\boldsymbol{H}}_{\text{misfit}}$ = prior-preconditioned Hessian of data misfit = $\boldsymbol{\Gamma}_{\text{prior}}^{1/2} \boldsymbol{H}_{\text{misfit}} \boldsymbol{\Gamma}_{\text{prior}}^{1/2}$

$\boldsymbol{H}_{\text{misfit}}$ = Gauss-Newton portion of Hessian misfit = $\boldsymbol{F}^{\natural} \boldsymbol{\Gamma}_{\text{obs}}^{-1} \boldsymbol{F}$

$\widetilde{\boldsymbol{V}}_r$ = $\boldsymbol{\Gamma}_{\text{prior}}^{1/2} \boldsymbol{V}_r$, $\widetilde{\boldsymbol{V}}_r^{\Diamond}$ = adjoint of $\widetilde{\boldsymbol{V}}_r$

$\boldsymbol{\Gamma}_{\text{prior}}^{-1}$ = $\boldsymbol{M}^{-1}\boldsymbol{K}, \boldsymbol{K}$= Laplace stiffness.

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

Bayesian Inference

Approach 2: Normal Approximation + Low Rank Laplace Approximation*

- Gaussian prior, likelihood ⇒ ***Gaussian posterior:*** $\pi_{\text{pos}}\left(\boldsymbol{z} \mid \boldsymbol{y}^{\text{obs}}\right) = N(\boldsymbol{z}_{\text{MAP}}, \boldsymbol{\Gamma}_{\text{post}})$

- ***Linearize*** parameter-to-observable map around MAP point:

$$\boldsymbol{y}^{\text{obs}} = \boldsymbol{f}(\boldsymbol{z}) + \epsilon \approx \boldsymbol{f}(\boldsymbol{z}_{\text{MAP}}) + \boldsymbol{F}(\boldsymbol{z} - \boldsymbol{z}_{\text{MAP}}) + \epsilon$$

- ***Covariance*** of Gaussian ***posterior*** given by:

$$\boxed{\boldsymbol{\Gamma}_{\text{post}} = \left(\boldsymbol{H}_{\text{misfit}}^{\text{PCE}} + \boldsymbol{\Gamma}_{\text{prior}}^{-1}\right)^{-1}}$$

Dense!

- ***Low-rank approximation*** of $\boldsymbol{\Gamma}_{\text{post}}$ obtained using Sherman-Morrison-Woodbury formula:

$$\boxed{\boldsymbol{\Gamma}_{\text{post}} \approx \boldsymbol{\Gamma}_{\text{prior}} - \widetilde{\boldsymbol{V}}_r \boldsymbol{D}_r \widetilde{\boldsymbol{V}}_r^{\diamond}}$$

Symbols*:

$\boldsymbol{V}_r, \boldsymbol{D}_r$: eigenvecs, eigenvals of $\widetilde{\boldsymbol{H}}_{\text{misfit}}$

$\widetilde{\boldsymbol{H}}_{\text{misfit}}$ = prior-preconditioned Hessian of data misfit = $\boldsymbol{\Gamma}_{\text{prior}}^{1/2} \boldsymbol{H}_{\text{misfit}} \boldsymbol{\Gamma}_{\text{prior}}^{1/2}$

$\boldsymbol{H}_{\text{misfit}}$ = Gauss-Newton portion of Hessian misfit = $\boldsymbol{F}^{\natural} \boldsymbol{\Gamma}_{\text{obs}}^{-1} \boldsymbol{F}$

$\widetilde{\boldsymbol{V}}_r$ = $\boldsymbol{\Gamma}_{\text{prior}}^{1/2} \boldsymbol{V}_r$, $\widetilde{\boldsymbol{V}}_r^{\diamond}$ = adjoint of $\widetilde{\boldsymbol{V}}_r$

$\boldsymbol{\Gamma}_{\text{prior}}^{-1}$ = $\boldsymbol{M}^{-1}\boldsymbol{K}, \boldsymbol{K}$= Laplace stiffness.

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

Bayesian Inference

Approach 2: Normal Approximation + Low Rank Laplace Approximation*

- Gaussian prior, likelihood ⇒ ***Gaussian posterior:*** $\pi_{\text{pos}}\left(\boldsymbol{z} \mid \boldsymbol{y}^{\text{obs}}\right) = N(\boldsymbol{z}_{\text{MAP}}, \boldsymbol{\Gamma}_{\text{post}})$

- ***Linearize*** parameter-to-observable map around MAP point:

$$\boldsymbol{y}^{\text{obs}} = \boldsymbol{f}(\boldsymbol{z}) + \epsilon \approx \boldsymbol{f}(\boldsymbol{z}_{\text{MAP}}) + \boldsymbol{F}(\boldsymbol{z} - \boldsymbol{z}_{\text{MAP}}) + \epsilon$$

- ***Covariance*** of Gaussian ***posterior*** given by:

$$\boxed{\boldsymbol{\Gamma}_{\text{post}} = \left(\boldsymbol{H}_{\text{misfit}}^{\text{PCE}} + \boldsymbol{\Gamma}_{\text{prior}}^{-1}\right)^{-1}}$$

Dense!

- ***Low-rank approximation*** of $\boldsymbol{\Gamma}_{\text{post}}$ obtained using Sherman-Morrison-Woodbury formula:

$$\boxed{\boldsymbol{\Gamma}_{\text{post}} \approx \boldsymbol{\Gamma}_{\text{prior}} - \widetilde{\boldsymbol{V}}_r \boldsymbol{D}_r \widetilde{\boldsymbol{V}}_r^{\Diamond}}$$

Symbols*:

$\boldsymbol{V}_r, \boldsymbol{D}_r$: eigenvecs, eigenvals of $\widetilde{\boldsymbol{H}}_{\text{misfit}}$

$\widetilde{\boldsymbol{H}}_{\text{misfit}}$ = prior-preconditioned Hessian of data misfit = $\boldsymbol{\Gamma}_{\text{prior}}^{1/2} \boldsymbol{H}_{\text{misfit}} \boldsymbol{\Gamma}_{\text{prior}}^{1/2}$

$\boldsymbol{H}_{\text{misfit}}$ = Gauss-Newton portion of Hessian misfit = $\boldsymbol{F}^{\natural} \boldsymbol{\Gamma}_{\text{obs}}^{-1} \boldsymbol{F}$

$\widetilde{\boldsymbol{V}}_r$ = $\boldsymbol{\Gamma}_{\text{prior}}^{1/2} \boldsymbol{V}_r$, $\widetilde{\boldsymbol{V}}_r^{\Diamond}$ = adjoint of $\widetilde{\boldsymbol{V}}_r$

$\boldsymbol{\Gamma}_{\text{prior}}^{-1}$ = $\boldsymbol{M}^{-1}\boldsymbol{K}$, $\boldsymbol{K}$= Laplace stiffness.

- $\widetilde{\boldsymbol{H}}_{\text{misfit}}$ and its eigenvalue decomposition can be computed efficiently using a parallel ***matrix-free Lanczos method***.

- ***Rank*** $(\boldsymbol{\Gamma}_{\text{post}})$ = # modes informing directions of posterior (active subspace vectors**).

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

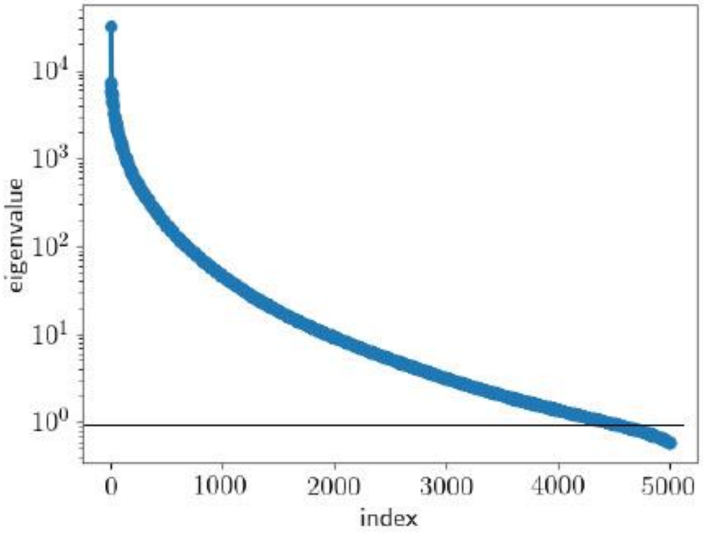
Bayesian Inference

Approach 2: Normal Approximation + Low Rank Laplace Approximation*

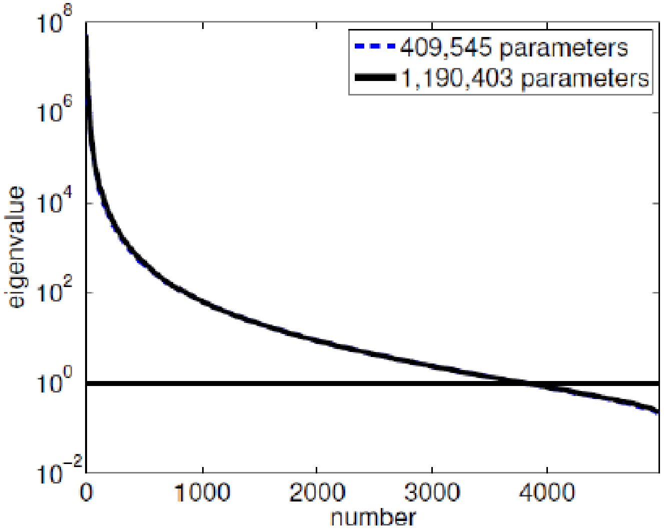
Upshots:

☺ Eigenvalues of prior-preconditioned misfit Hessian $\widetilde{\boldsymbol{H}}_{\text{misfit}}$ decay rapidly and decay is independent of # parameters.

Greenland



Antarctica*



Figures above: eigenvalue decay of prior preconditioned misfit Hessian

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

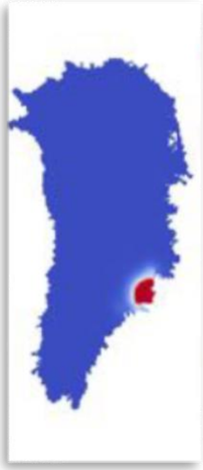
Bayesian Inference

<u>Approach 2:</u> Normal Approximation + Low Rank Laplace Approximation*

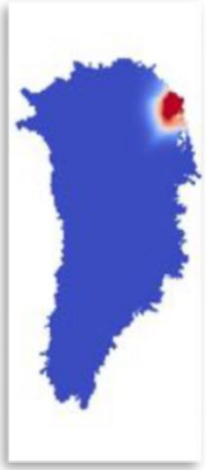
Upshots:

☺ Prior preconditioned misfit ***eigenvectors*** have ***physical interpretation***:

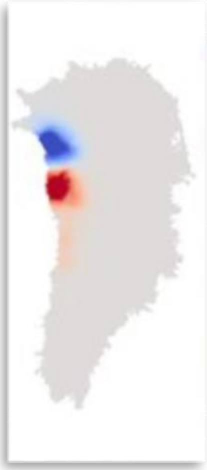
- ➢ First modes correspond to regions which are ***highly informed by data***
- ➢ Modes become more ***global*** as eigenvalues decay



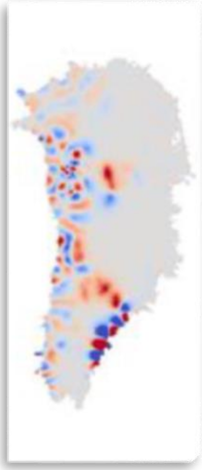
Mode 1



Mode 2



Mode 3



Mode 200

Bayesian Inference

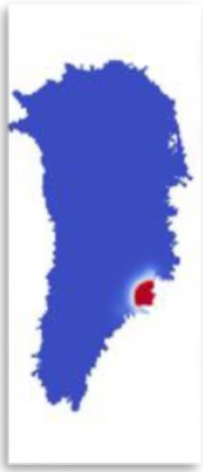
Approach 2: Normal Approximation + Low Rank Laplace Approximation*

Upshots:

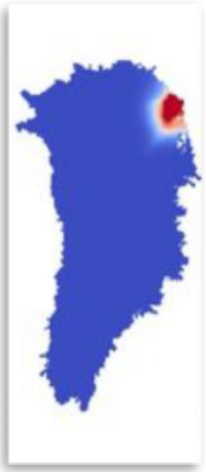
☺ Prior preconditioned misfit ***eigenvectors*** have ***physical interpretation***:

- ➢ First modes correspond to regions which are ***highly informed by data***
- ➢ Modes become more ***global*** as eigenvalues decay

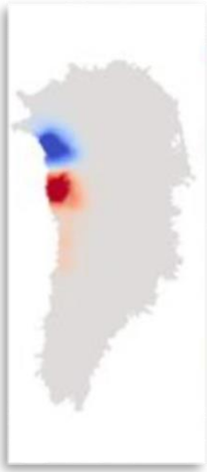
☺ The use of data has ***drastically reduces*** the ***posterior variance***



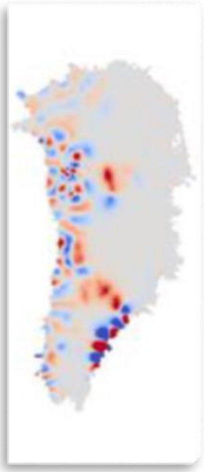
Mode 1



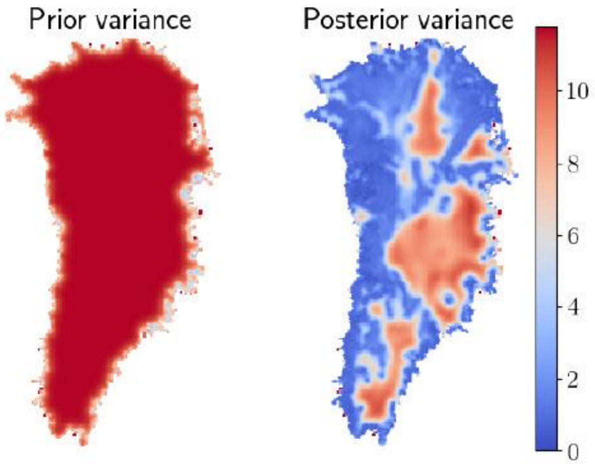
Mode 2



Mode 3



Mode 200



Prior variance

Posterior variance

Bayesian Inference

Approach 2: Normal Approximation + Low Rank Laplace Approximation*

Issues:

☹ PDF will be **Gaussian** – general PDFs cannot be obtained.

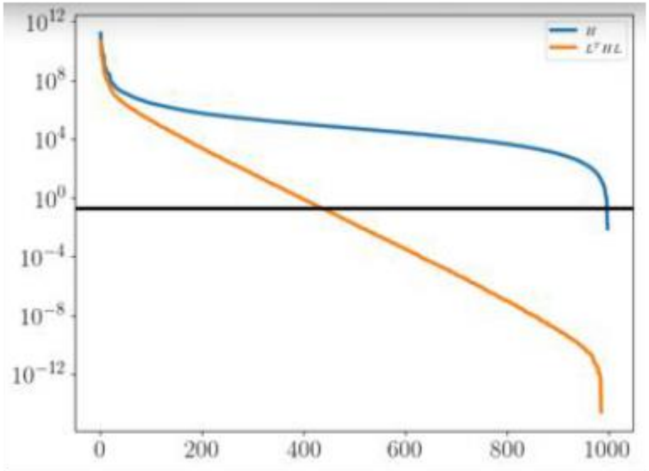
Bayesian Inference

Approach 2: Normal Approximation + Low Rank Laplace Approximation*

Issues:

☹ PDF will be **Gaussian** – general PDFs cannot be obtained.

☹ Laplace equation (regularization) **involves correlation length parameter** that changes decay of eigenvalues of prior preconditioned Hessian.



Bayesian Inference

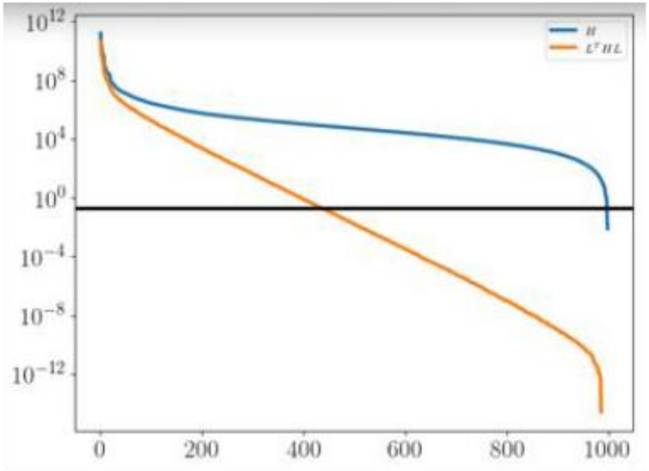
Approach 2: Normal Approximation + Low Rank Laplace Approximation*

Issues:

☹ PDF will be **Gaussian** – general PDFs cannot be obtained.

☹ Laplace equation (regularization) **involves correlation length parameter** that changes decay of eigenvalues of prior preconditioned Hessian.

☹ **Dimension** of parameter space is **too high O(1000)** for forward propagation.



Bayesian Inference

Approach 2: Normal Approximation + Low Rank Laplace Approximation*

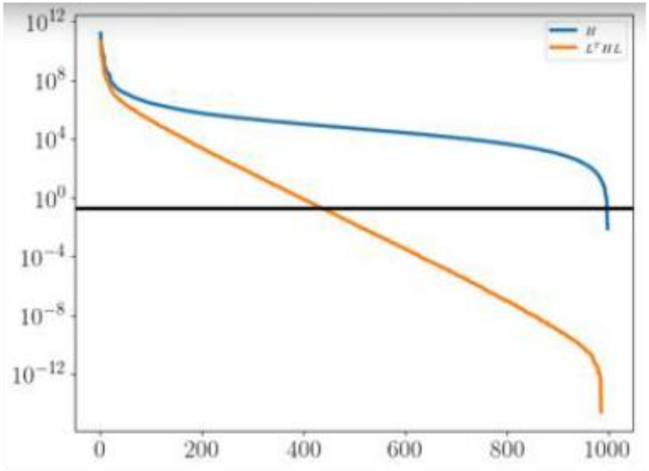
Issues:

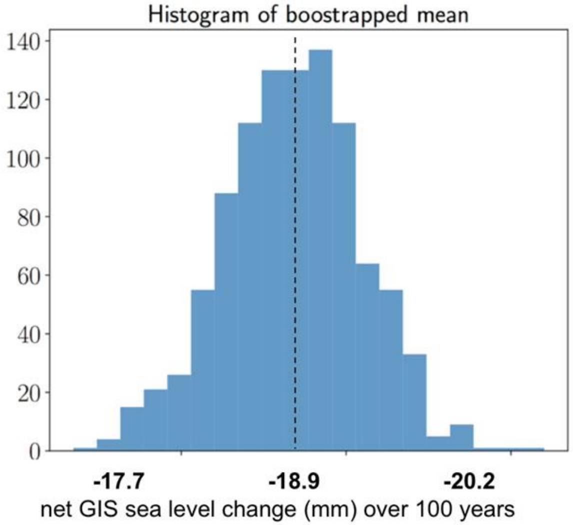
☹ PDF will be **Gaussian** – general PDFs cannot be obtained.

☹ Laplace equation (regularization) **involves correlation length parameter** that changes decay of eigenvalues of prior preconditioned Hessian.

☹ **Dimension** of parameter space is **too high O(1000)** for forward propagation.

☹ Log-normal prior may be cause of (nonphysical) **bias** towards mass increase when performing **forward propagation.**





Bayesian Inference

Ongoing work:

- Use **low fidelity** models (e.g. SIA) to study problems (such as bias in SLR on previous slide) with the large-scale, high-resolution, expensive end-to-end framework.
- Use dimension reduction, leveraging **transient adjoints** obtained from new model suite, to reduce cost of **propagating uncertainties** through **transient model**.

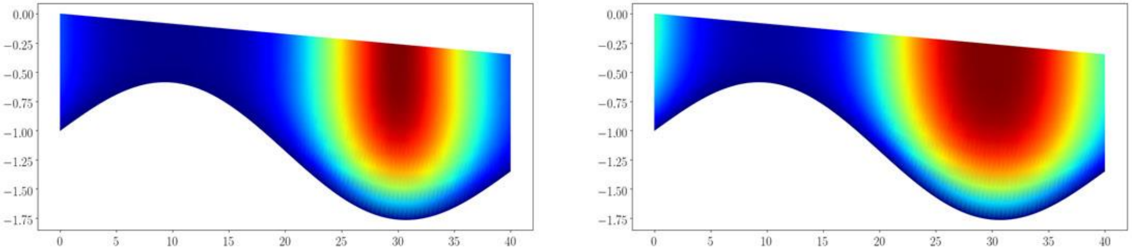


Figure 1: ISMIP-HOM B test + SIA and BP models is >1000× less than GIS.

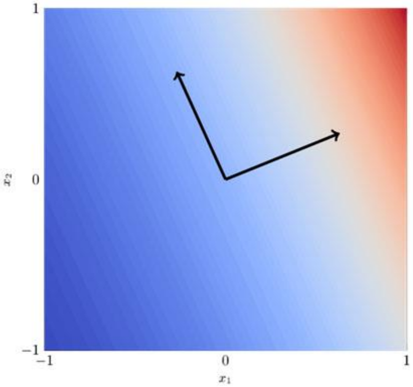


Figure 2: gradients can determine directions that significantly impact SLR.

- **Dimension reduction** by **adding physics**: subglacial hydrology models rely on only a handful of parameters that, to first approximation, can be considered uniform

$$\beta(\boldsymbol{u}) = \mu_f N \left(\frac{|\boldsymbol{u}|}{|\boldsymbol{u}| + \lambda A N^n} \right)^q \frac{1}{|\boldsymbol{u}|}$$

+

Thickness equation
(subglacial hydrology)

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;
```

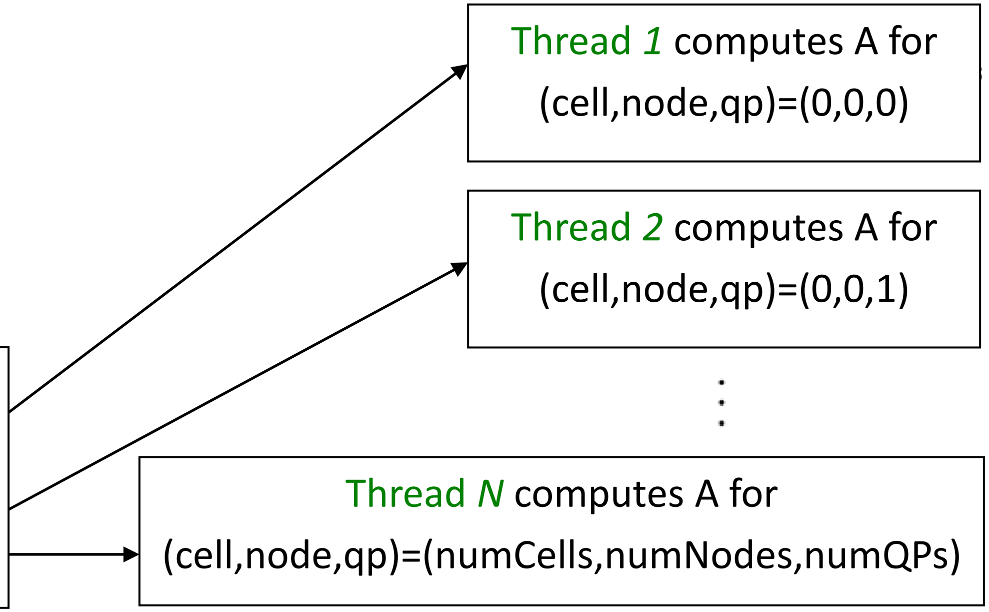
MPI process *n*

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)

Single Threading
Task 1
Task 2
Task 3
Task 4
Thread 1
Core 1

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);
```

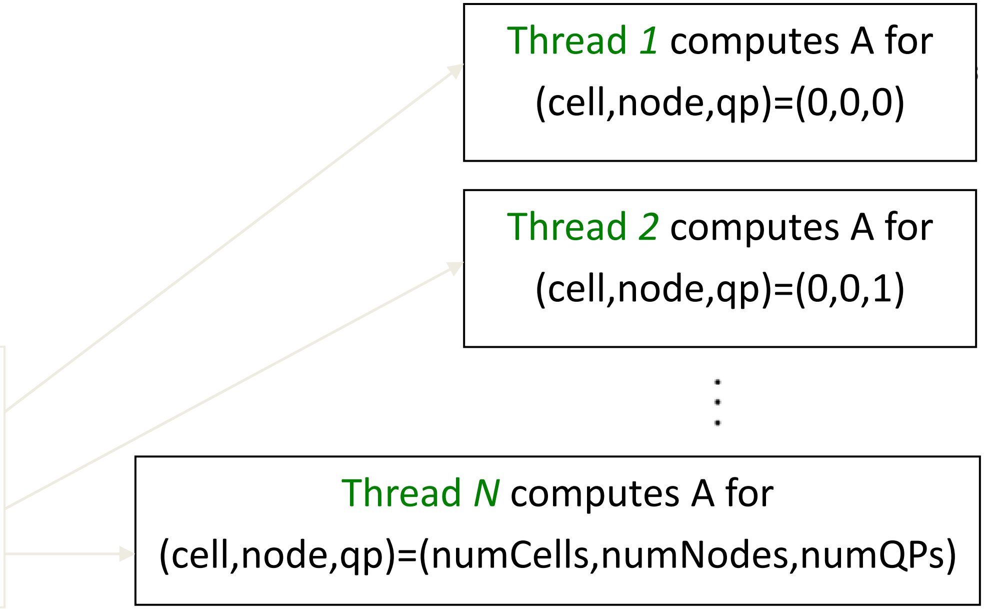
Single Threading
Task 1
Task 2
Task 3
Task 4
Thread 1
Core 1

* Unified Virtual Memory.

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
```

Single Threading
Task 1
Task 2
Task 3
Task 4
Thread 1
Core 1

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

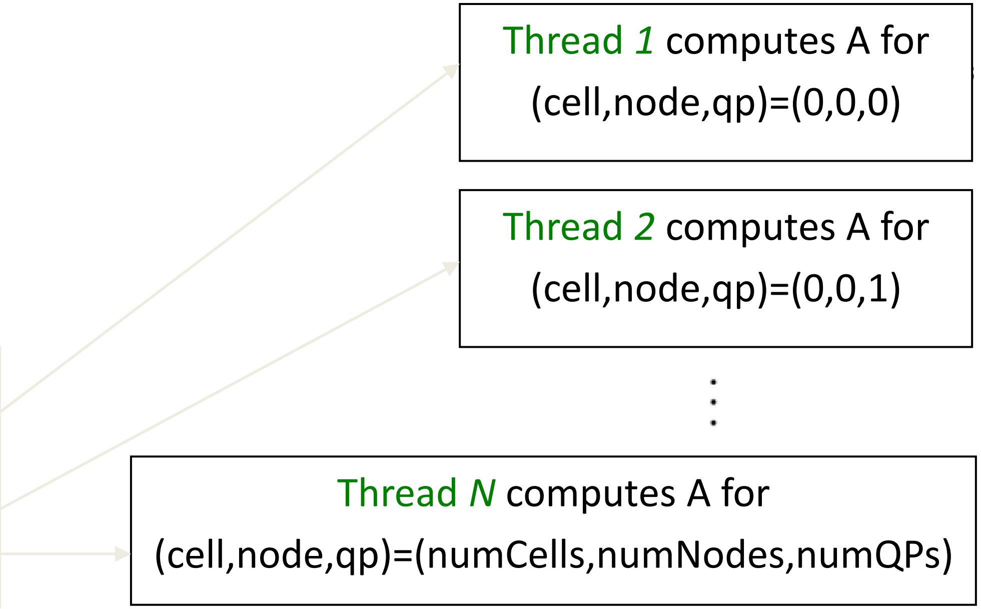
  Kokkos::atomic_fetch_add

Single Threading
Task 1
Task 2
Task 3
Task 4
Thread 1
Core 1

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
```

Single Threading
Task 1
Task 2
Task 3
Task 4
Thread 1
Core 1

- ***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.

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 ALI 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
```

Single Threading
Task 1
Task 2
Task 3
Task 4
Thread 1
Core 1

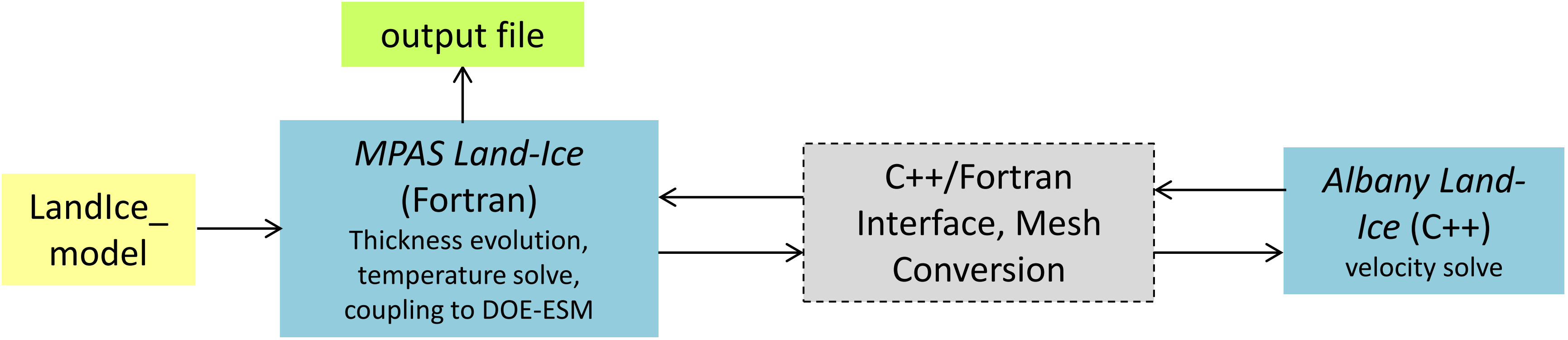
- ***Atomics*** used to scatter local data to global data structures

```
Kokkos::atomic_fetch_add
```

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

* Unified Virtual Memory.

MPAS + ALI Coupling

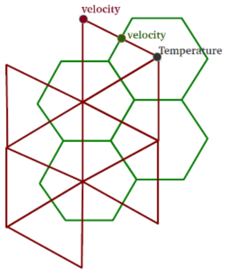

output file
LandIce_ model
MPAS Land-Ice (Fortran)
Thickness evolution, temperature solve, coupling to DOE-ESM
C++/Fortran Interface, Mesh Conversion
Albany Land-Ice (C++)
velocity solve

E3SM
Energy Exascale Earth System Model

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

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

MPAS
Model for Prediction Across Scales

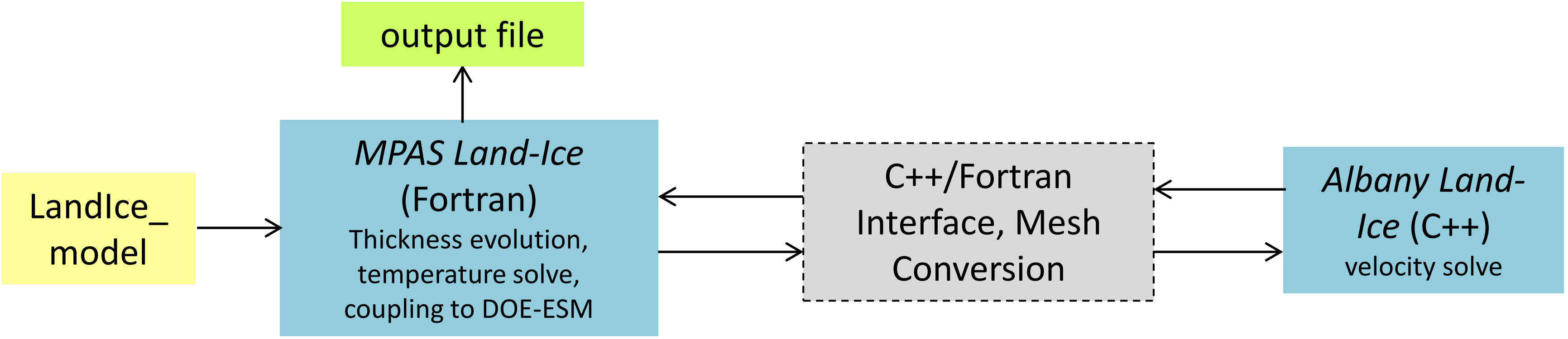


$$\begin{cases} -\nabla\cdot(2\mu\dot{\boldsymbol{\epsilon}}_1) = -\rho g\dfrac{\partial s}{\partial x} \\ -\nabla\cdot(2\mu\dot{\boldsymbol{\epsilon}}_2) = -\rho g\dfrac{\partial s}{\partial y} \end{cases}$$

Albany

"Loose" sequential/staggered coupling between $\boldsymbol{u}$ and (T, H).

FO Stokes-Thickness Coupling



output file

LandIce_ model

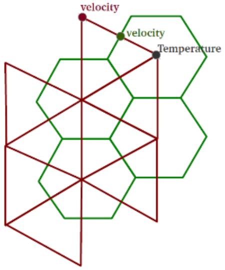
MPAS Land-Ice (Fortran)
Thickness evolution, temperature solve, coupling to DOE-ESM

C++/Fortran Interface, Mesh Conversion

Albany Land-Ice (C++)
velocity solve

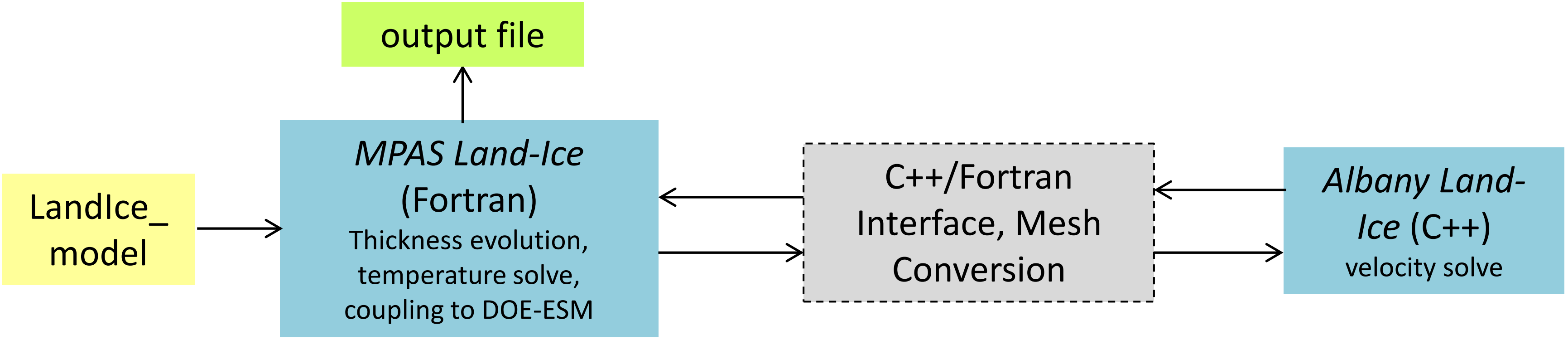
$$\frac{\partial H}{\partial t} = -\nabla \cdot (\bar{\boldsymbol{u}} H) + \dot{b}$$

H equation is solved with upwind scheme + incremental remap.



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

FO Stokes-Thickness Coupling



output file

LandIce_ model

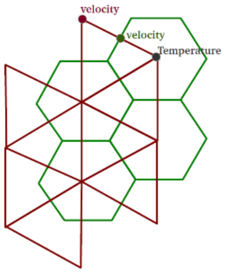
MPAS Land-Ice (Fortran)
Thickness evolution, temperature solve, coupling to DOE-ESM

C++/Fortran Interface, Mesh Conversion

Albany Land-Ice (C++)
velocity solve

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

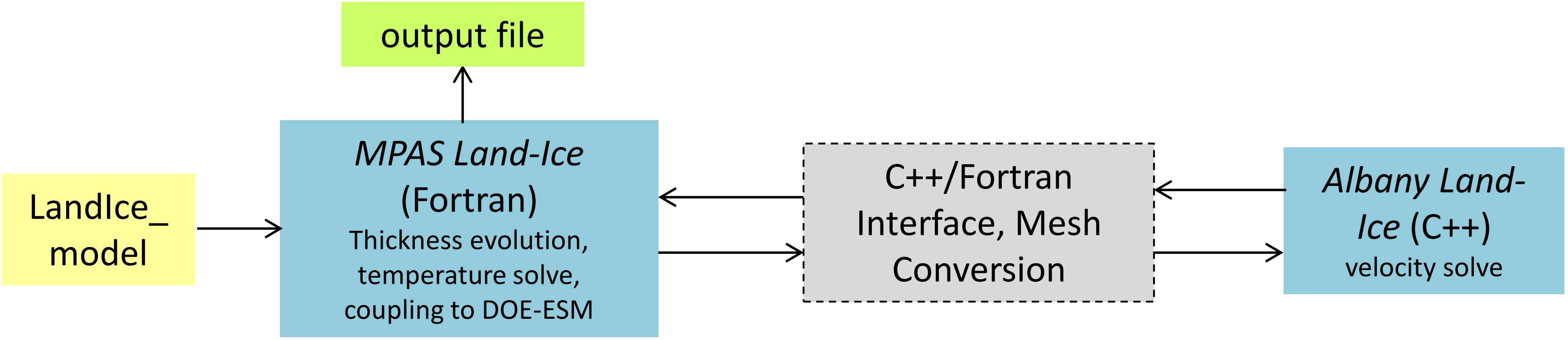
H equation is solved with upwind scheme + incremental remap.



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

☺ **Upside:** scheme fits nicely into existing codes

FO Stokes-Thickness Coupling



output file

LandIce_ model

MPAS Land-Ice (Fortran)

Thickness evolution, temperature solve, coupling to DOE-ESM

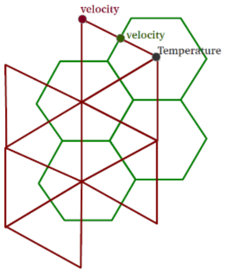
C++/Fortran Interface, Mesh Conversion

Albany Land-Ice (C++)

velocity solve

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

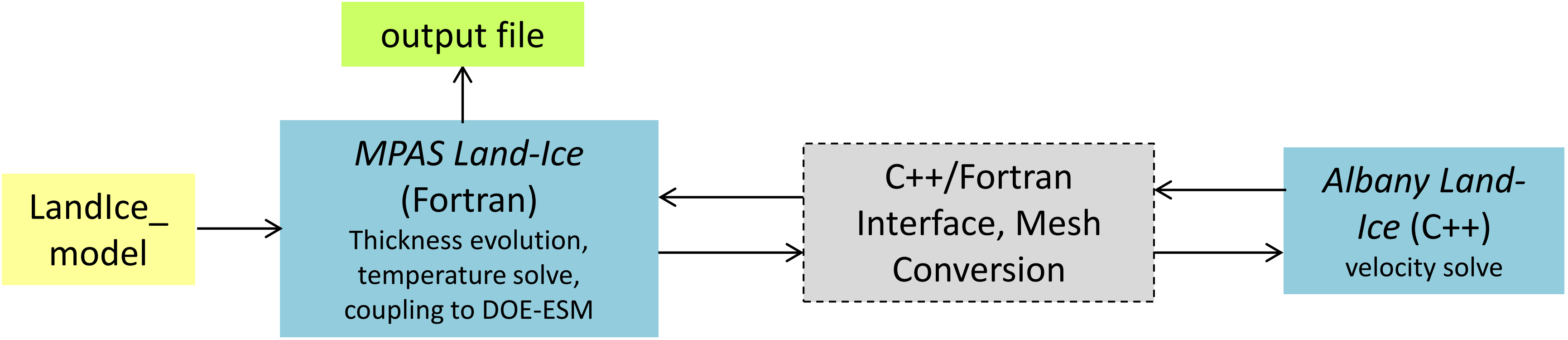
H equation is solved with upwind scheme + incremental remap.



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

- ☺ **Upside:** scheme fits nicely into existing codes
- ☹ **Downside:** for problems with shallow ice on frozen bedrock, need to satisfy very restrictive **diffusive CFL** condition*: $\Delta t \leq CFL_{\text{diff}}(\Delta \boldsymbol{x})^2$

FO Stokes-Thickness Coupling



output file

LandIce_ model

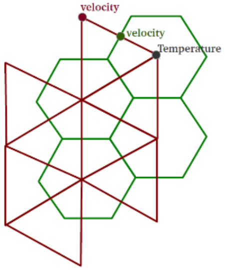
MPAS Land-Ice (Fortran)
Thickness evolution, temperature solve, coupling to DOE-ESM

C++/Fortran Interface, Mesh Conversion

Albany Land-Ice (C++)
velocity solve

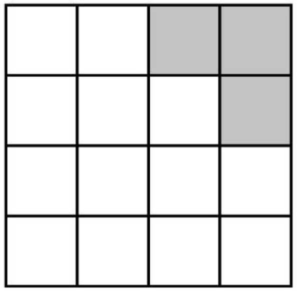
$$\frac{\partial H}{\partial t} = -\nabla \cdot (\bar{\boldsymbol{u}} H) + \dot{b}$$

H equation is solved with upwind scheme + incremental remap.

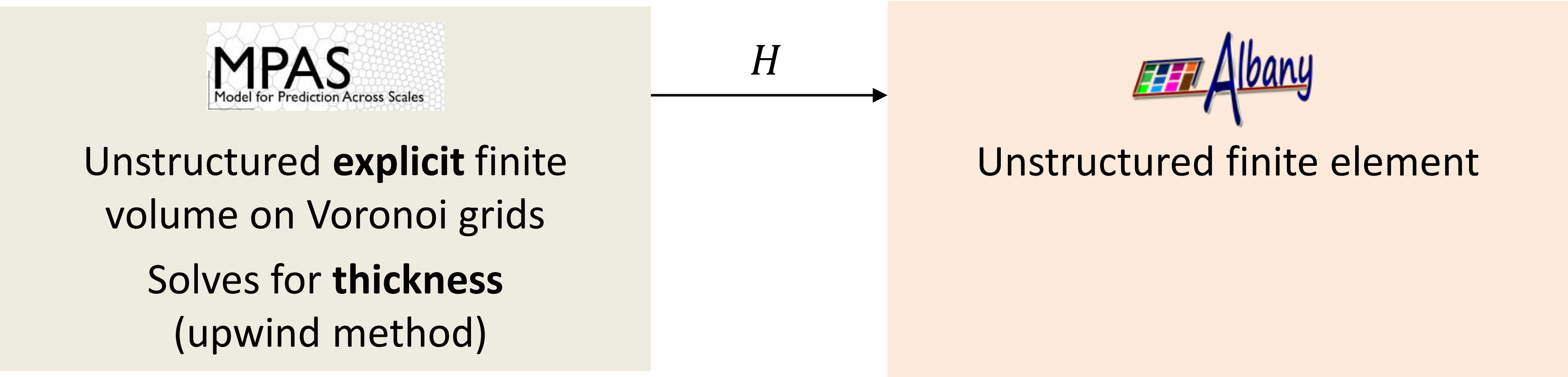


$$\begin{cases} -\nabla \cdot (2\mu\dot{\boldsymbol{\epsilon}}_1) = -\rho g \dfrac{\partial s}{\partial x} \\ -\nabla \cdot (2\mu\dot{\boldsymbol{\epsilon}}_2) = -\rho g \dfrac{\partial s}{\partial y} \end{cases}$$

- ☺ **Upside:** scheme fits nicely into existing codes
- ☹ **Downside:** for problems with shallow ice on frozen bedrock, need to satisfy very restrictive **diffusive CFL** condition*: $\Delta t \leq CFL_{\text{diff}}(\Delta \boldsymbol{x})^2$
- ☹ **Downside:** Very crude representation of **ice advancement/retreat**



Semi-Implicit Coupling



MPAS

Model for Prediction Across Scales

Unstructured **explicit** finite volume on Voronoi grids

Solves for **thickness** (upwind method)

H

Albany

Unstructured finite element

- MPAS computes thickness H, uses it to define geometry, which is passed to ALI.

Semi-Implicit Coupling

MPAS
Model for Prediction Across Scales

Unstructured **explicit** finite volume on Voronoi grids

Solves for **thickness** (upwind method)

$\xrightarrow{H}$

Albany

Unstructured finite element

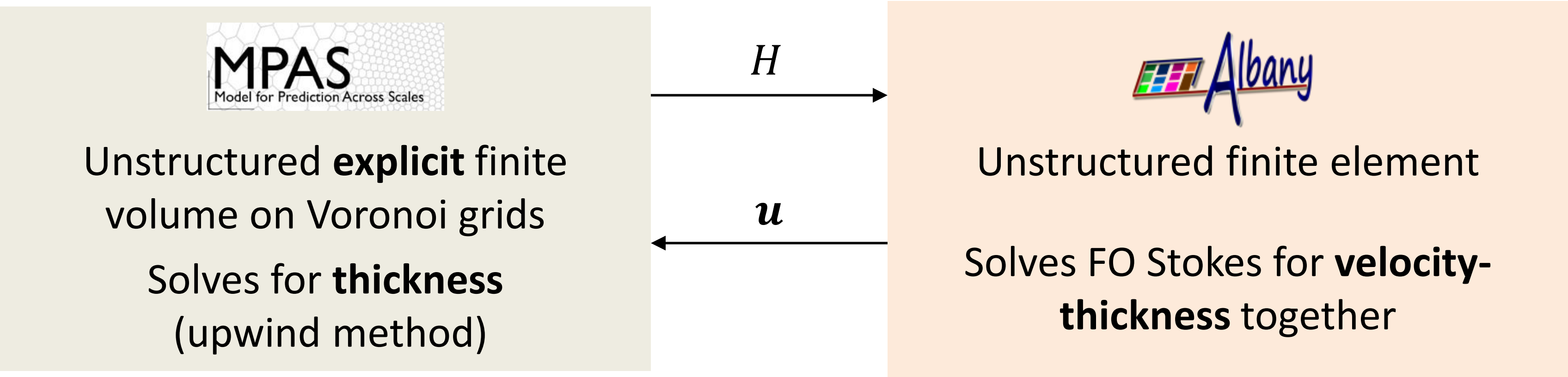
Solves FO Stokes for **velocity-thickness** together

- MPAS computes thickness H, uses it to define geometry, which is passed to ALI.
- ALI computes coupled velocity-thickness $(\boldsymbol{u}, H)$ pair:

$$\boxed{\begin{gathered} -2\mu\left(\boldsymbol{u}^{(n+1)}\right)\nabla \cdot \dot{\epsilon}\left(\boldsymbol{u}^{(n+1)}\right) = -\rho g \nabla\left(b + H^{(n+1)}\right), \quad \text{in } \Omega_{H^{(n+1)}} \\ \frac{H^{(n+1)} - H^{(n)}}{\Delta t} = -\nabla \cdot \left(\overline{\boldsymbol{u}}^{(n+1)} H^{(n+1)}\right) + \dot{b} \end{gathered}}$$

<u>Idea:</u> the velocity computed by the coupled system FO-thickness equation will be **more stable** than the one computed by FO Stokes only and will allow use of larger Δt

Semi-Implicit Coupling



MPAS
Model for Prediction Across Scales

Unstructured **explicit** finite volume on Voronoi grids

Solves for **thickness** (upwind method)

Albany

Unstructured finite element

Solves FO Stokes for **velocity-thickness** together

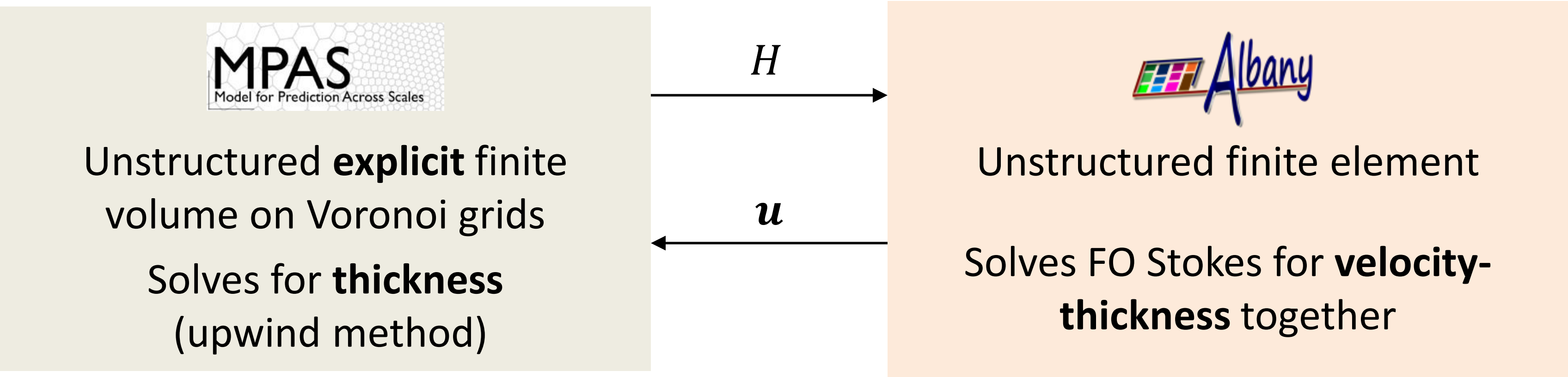
- MPAS computes thickness H, uses it to define geometry, which is passed to ALI.
- ALI computes coupled velocity-thickness $(\boldsymbol{u}, H)$ pair:

$$\boxed{\begin{aligned} -2\mu\left(\boldsymbol{u}^{(n+1)}\right)\nabla\cdot\dot{\boldsymbol{\epsilon}}\left(\boldsymbol{u}^{(n+1)}\right) &= -\rho g\nabla\left(b + H^{(n+1)}\right), \quad \text{in } \Omega_{H^{(n+1)}} \\ \frac{H^{(n+1)} - H^{(n)}}{\Delta t} &= -\nabla\cdot\left(\overline{\boldsymbol{u}}^{(n+1)} H^{(n+1)}\right) + \dot{b} \end{aligned}}$$

Idea: the velocity computed by the coupled system FO-thickness equation will be **more stable** than the one computed by FO Stokes only and will allow use of larger Δt

- Only velocity $\boldsymbol{u}$ is passed back to MPAS.

Semi-Implicit Coupling



MPAS
Model for Prediction Across Scales

Unstructured **explicit** finite volume on Voronoi grids

Solves for **thickness** (upwind method)

H

$\boldsymbol{u}$

Albany

Unstructured finite element

Solves FO Stokes for **velocity-thickness** together

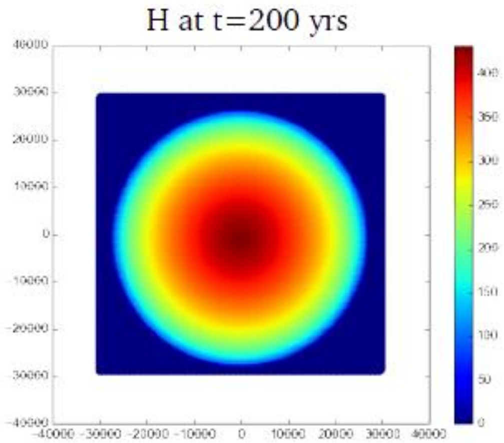
- MPAS computes thickness H, uses it to define geometry, which is passed to ALI.
- ALI computes coupled velocity-thickness $(\boldsymbol{u}, H)$ pair:

$$\boxed{\begin{aligned} -2\mu\left(\boldsymbol{u}^{(n+1)}\right)\nabla\cdot\dot{\epsilon}\left(\boldsymbol{u}^{(n+1)}\right) &= -\rho g \nabla\left(b + H^{(n+1)}\right), \quad \text{in } \Omega_{H^{(n+1)}} \\ \frac{H^{(n+1)} - H^{(n)}}{\Delta t} &= -\nabla\cdot\left(\overline{\boldsymbol{u}}^{(n+1)} H^{(n+1)}\right) + \dot{b} \end{aligned}}$$

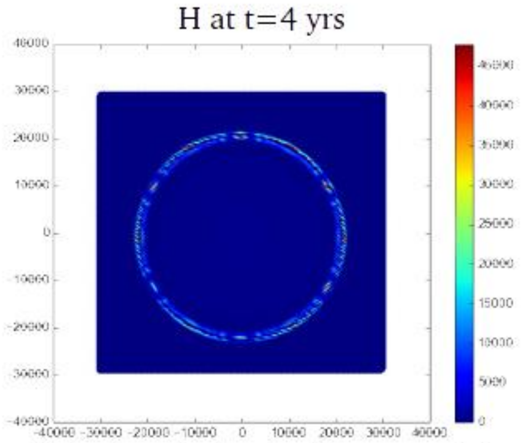
<u>Idea:</u> the velocity computed by the coupled system FO-thickness equation will be **more stable** than the one computed by FO Stokes only and will allow use of larger Δt

- Only velocity $\boldsymbol{u}$ is passed back to MPAS.
- **Downside:** more intrusive implementation; larger system; expense associated to geometry changing between iterations (use Newton to compute shape derivatives).

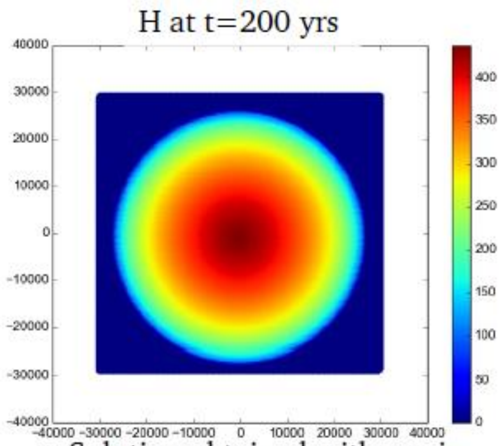
Semi-Implicit Approach: Dome Test Case



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



Solution obtained with sequential coupling, dt = 1 yr

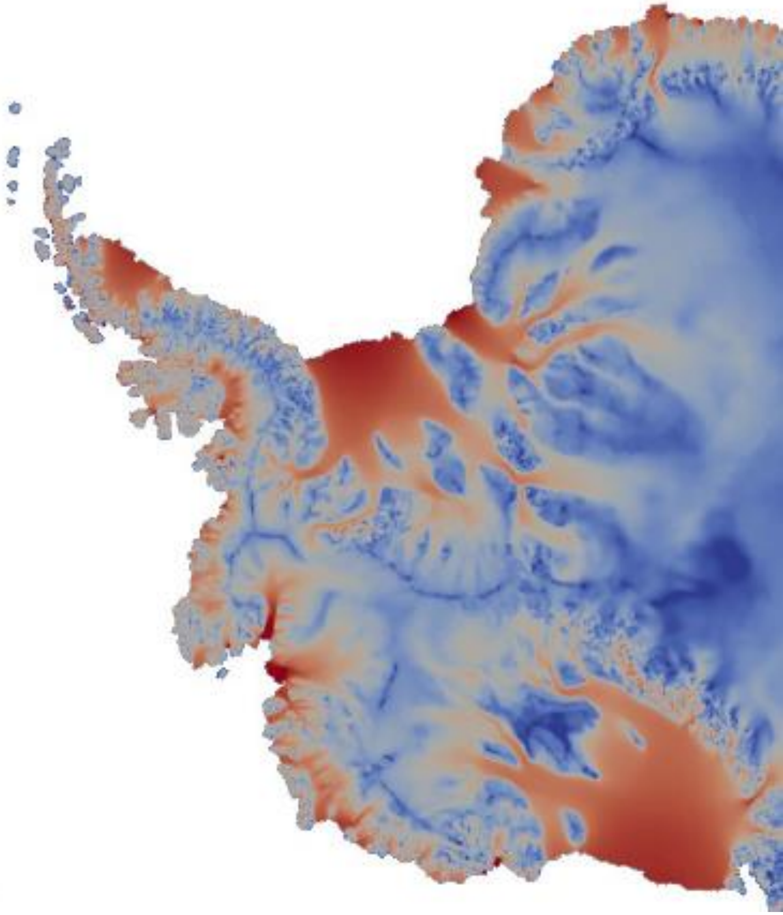


Solution obtained with semi-implicit coupling, dt = 5 yrs

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

Semi-Implicit Approach: Antarctica

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

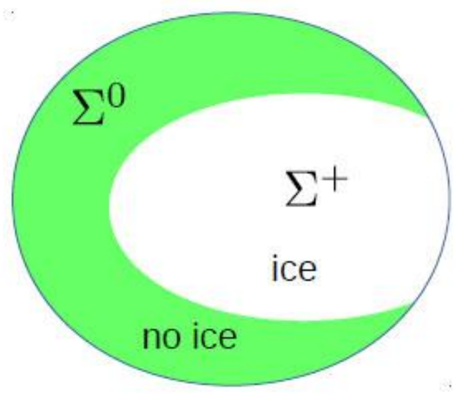


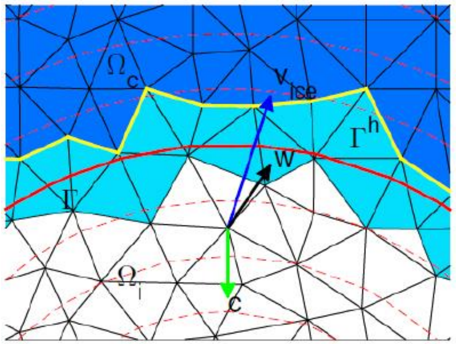
Basal friction: obtained with inversion.
Geometry: Bedmap2 (Fretwell *et al.,* Cryosphere, 2013), managed by D. Martin and X. Asay-Davis.
Temperature: Cornford, Martin *et al,* 2014; Pattyn *et al., 2010.*
Mesh: unstructured Delaynay mesh refined based on surface velocity (MPAS planar Voronoi grid generator by M. Duda, NCAR).

Towards Fully Implicit FO Stokes-Thickness Coupling

- We are looking at the following **fully implicit** formulations:
  - **Level set** formulation coupled with the thickness evolution equation is used to track the front position*: no need to modify mesh, can handle changes in topography.
  - Thickness equation as an **obstacle problem/variational inequality****: no need to track boundary, amenable to implicit integration

$$\boxed{\frac{\partial H}{\partial t} = -\nabla \cdot (\bar{\boldsymbol{u}} H) + \dot{b}, \quad \text{in } \Sigma^{+}}$$





$$\Big\Downarrow$$

$$\boxed{\int \frac{\partial H}{\partial t}(v - H) \geq \int (\bar{\boldsymbol{u}} H) \cdot \nabla (v - H) + \int \theta (v - H), \quad H \geq 0, \forall v \geq 0, \text{in } \Sigma}$$

*Bondzio *et al.* 2016. **Bueler, 2016.

PISCEES & E3SM Coupling Validation

Sub-shelf melt rates (RRS30to10km resolution)

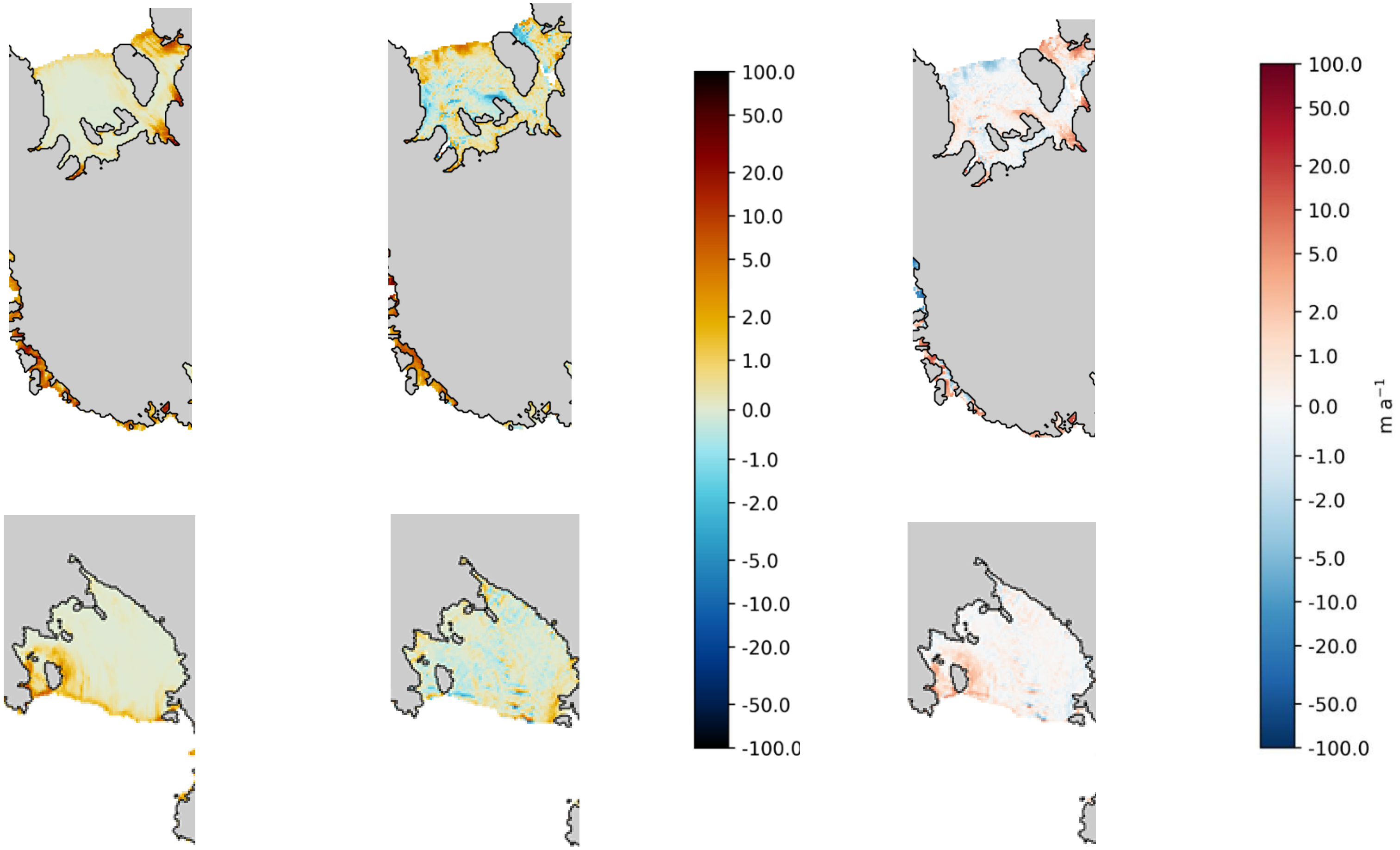
model

observations*

model – obs.

Filchner-Ronne Ice Shelf

Ross Ice Shelf



* Rignot et al., *Science*, 2013

Summary

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