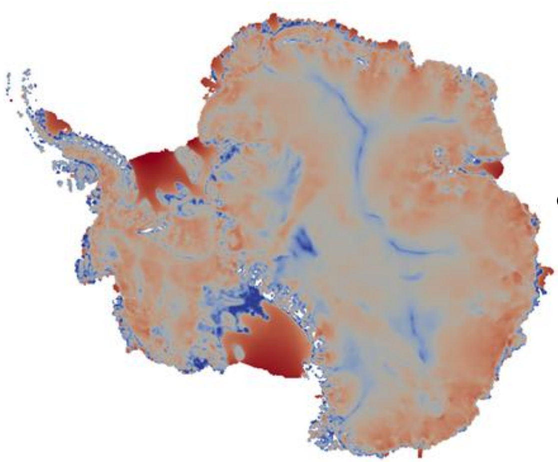
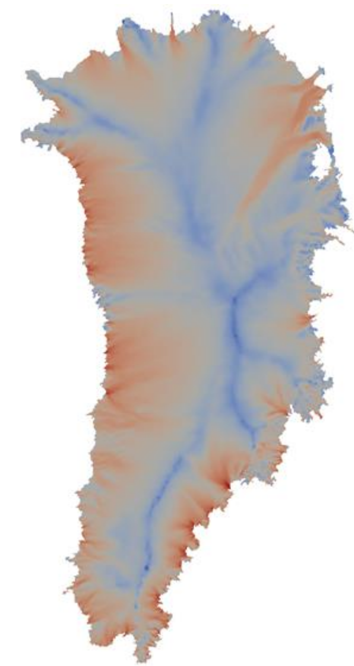
Albany/FELIX: A Robust & Scalable *Trilinos*-Based Finite-Element Ice Flow Dycore Built for Advanced Architectures & Analysis

I. Tezaur[1], A. Salinger[1], M. Perego[1], J. Jakeman[1], M. Eldred[1], I. Demeshko[1], R. Tuminaro[1], S. Price[2]

[1] Sandia National Laboratories
Livermore, CA and Albuquerque, NM, USA
[2] Los Alamos National Laboratory
Los Alamos, NM, USA



8th International Congress on Industrial and Applied Mathematics (ICIAM 2015)
August 10-14, 2015
Beijing, China



SAND2015-6419C

Sandia National Laboratories

Outline

Albany/FELIX = new land-ice solver with ***next-generation*** capabilities.

- **Overview:**
  - ***First Order (FO) Stokes*** model.
  - ***Albany/FELIX*** First-Order (FO) Stokes diagnostic solver.
  - ***CISM-Albany*** and ***MPAS-Albany*** codes for prognostic simulations of the ice sheet evolution.
- **Uncertainty Quantification (UQ):**
  - Deterministic inversion.
  - Bayesian calibration.
  - Forward propagation of uncertainty.
- **Performance portability.**
- **Summary** and ongoing work.
- **Questions?**

Outline

Albany/FELIX = new land-ice solver with ***next-generation*** capabilities.

- **Overview:**
  - ***First Order (FO) Stokes*** model.
  - ***Albany/FELIX*** First-Order (FO) Stokes diagnostic solver.
  - ***CISM-Albany*** and ***MPAS-Albany*** codes for prognostic simulations of the ice sheet evolution.
- **Uncertainty Quantification (UQ):**
  - Deterministic inversion.
  - Bayesian calibration.
  - Forward propagation of uncertainty.
- **Performance portability.**
- **Summary** and ongoing work.
- **Questions?**

The PISCEES Project and the *Albany/FELIX* Solver

SciDAC Scientific Discovery through Advanced Computing

"PISCEES" = Predicting Ice Sheet Climate & Evolution at Extreme Scales
5 year project funded by SciDAC, which began in June 2012

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

- Steady-state stress-velocity solver based on FO Stokes physics is known as ***Albany/FELIX****.
- **Requirements for *Albany/FELIX*:**
  - Scalable, fast, robust.
  - Dynamical core (dycore) when coupled to codes that solve thickness and temperature evolution equations (*CISM/MPAS LI* codes).
  - Advanced analysis capabilities (adjoint-based deterministic inversion, Bayesian calibration, UQ, sensitivity analysis).
  - Performance-portability.

This talk

Dycore will provide actionable predictions of 21st century sea-level rise (including uncertainty).

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

***Albany/FELIX* Solver (steady):**
Ice Sheet PDEs (First Order Stokes)
(stress-velocity solve)

⇄

***CISM/MPAS* Land Ice Codes (dynamic):**
Ice Sheet Evolution PDEs
(thickness, temperature evolution)

Sandia National Laboratories

The First-Order Stokes Model for Ice Sheets & Glaciers

- Ice sheet dynamics are given by the ***"First-Order" Stokes PDEs***: approximation* to viscous incompressible ***quasi-static*** Stokes flow with power-law viscosity.

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

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

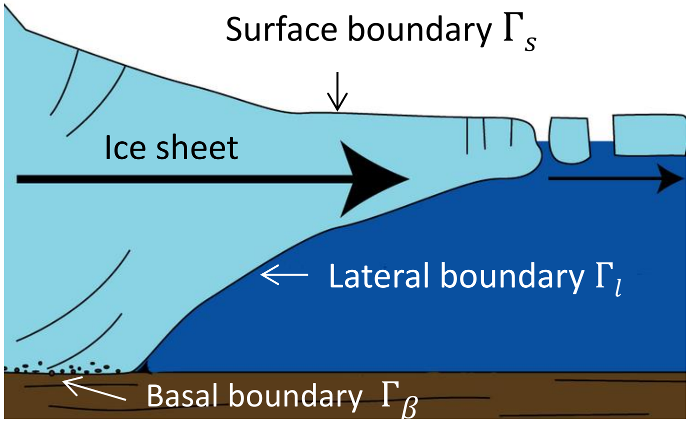
- Viscosity μ is nonlinear function given by "***Glen's law***":

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

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

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

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

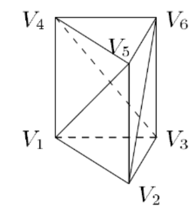


$$\beta = \text{sliding coefficient} \ge 0$$

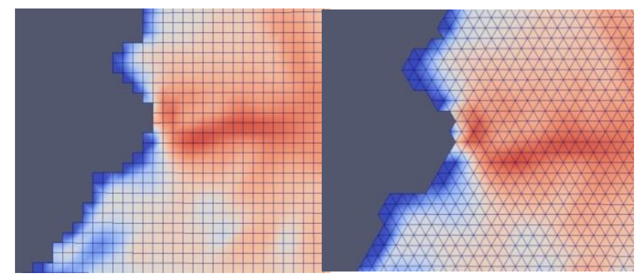
***Assumption:** aspect ratio δ is small and normals to upper/lower surfaces are almost vertical.

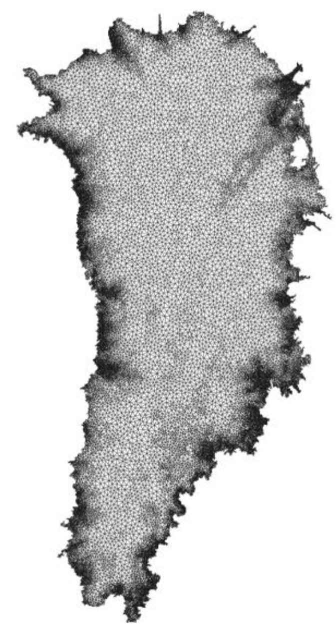
Sandia National Laboratories

Algorithmic Choices for *Albany/FELIX*: Discretization & Meshes



- **<u>Discretization:</u>** unstructured grid finite element method (FEM)
  - Can handle readily complex geometries.
  - Natural treatment of stress boundary conditions.
  - Enables regional refinement/unstructured meshes.
  - Wealth of software and algorithms.



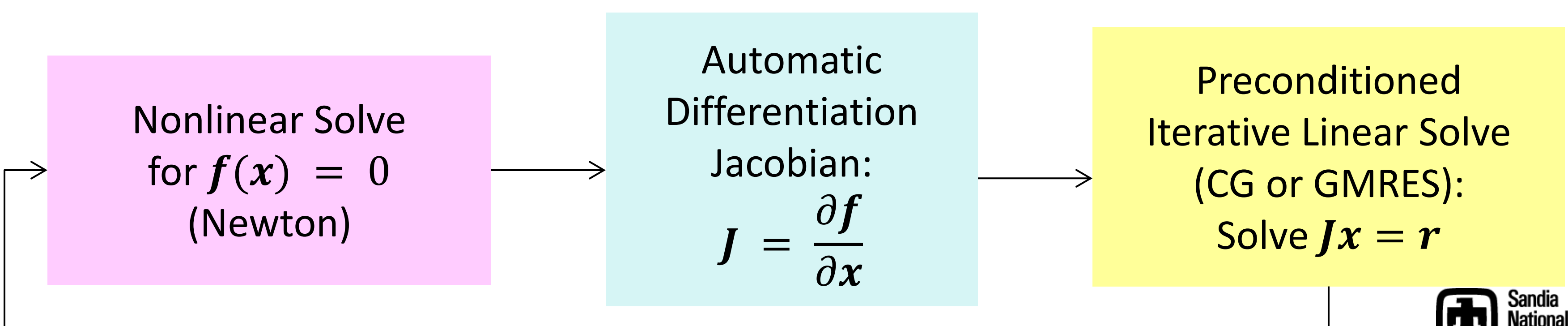


- **<u>Meshes:</u>** can use any mesh but interested specifically in
  - ***Structured hexahedral*** meshes (compatible with *CISM*).
  - ***Tetrahedral*** meshes (compatible with *MPAS LI*)
  - ***Unstructured Delaunay triangle*** meshes with regional refinement based on gradient of surface velocity.
  - All meshes are extruded (structured) in vertical direction as tetrahedra or hexahedra.

Algorithmic Choices for *Albany/FELIX*: Nonlinear & Linear Solver

- **Nonlinear solver:** full Newton with analytic (automatic differentiation) derivatives and homotopy continuation
    - Most robust and efficient for steady-state solves.
    - Jacobian available for preconditioners and matrix-vector products.
    - Analytic sensitivity analysis.
    - Analytic gradients for inversion.

- **Linear solver:** preconditioned iterative method
    - **Solvers:** Conjugate Gradient (CG) or GMRES
    - **Preconditioners:** ILU or algebraic multi-grid (AMG)



The *Albany/FELIX* Solver: Implementation in *Albany* using *Trilinos*

The ***Albany/FELIX*** First Order Stokes solver is implemented in a Sandia parallel C++ finite element code called…

Started by A. Salinger

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

Other Albany Physics Sets

"Agile Components"

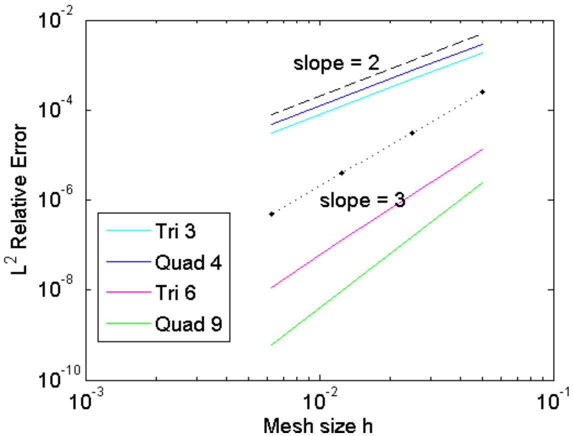
- Discretizations/meshes
- Solver libraries
- Preconditioners
- Automatic differentiation
- Many others!

- Parameter estimation
- Uncertainty quantification
- Optimization
- Bayesian inference

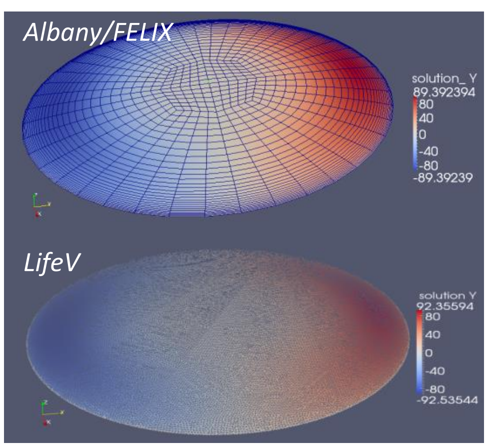
- Configure/build/test/documentation

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

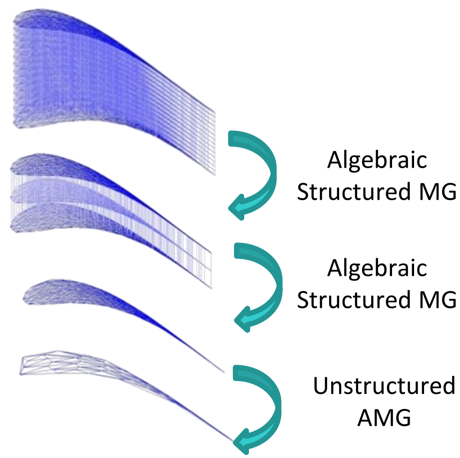
The *Albany/FELIX* Solver is Verified, Scalable, Fast and Robust!

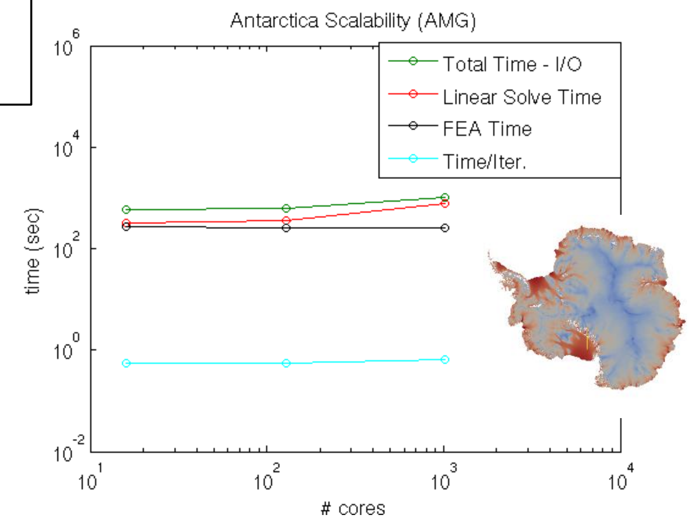


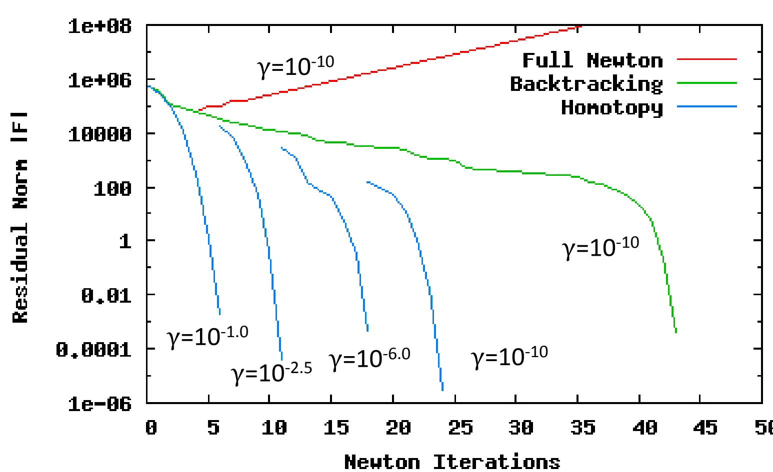
Verified via MMS and code-to-code comparisons.



Scalable via algebraic multi-grid (AMG) preconditioning







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

Robust via homotopy continuation w.r.t. γ.

Ice Sheet Evolution Models

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

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

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

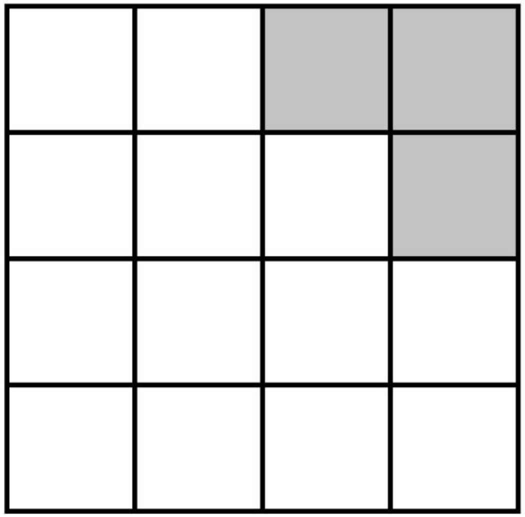
- ***Temperature equation*** (advection-diffusion):

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

(energy balance).

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

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



time t_0

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

Ice Sheet Evolution Models

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

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

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

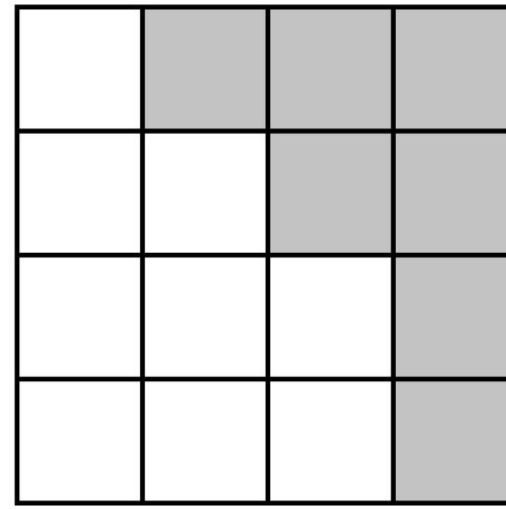
- ***Temperature equation*** (advection-diffusion):

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

(energy balance).

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

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



time t_1

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

Ice Sheet Evolution Models

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

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

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

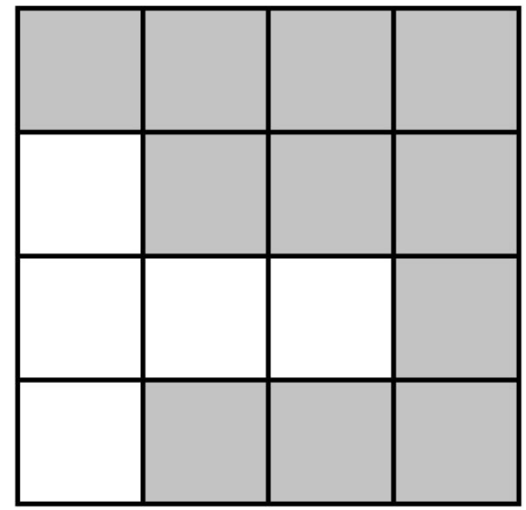
- ***Temperature equation*** (advection-diffusion):

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

(energy balance).

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

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



time t_2

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

Interfaces to *CISM* and *MPAS LI* for Transient Simulations

CISM-Albany

cism_driver

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

C++/Fortran Interface, Mesh Conversion

output file

Albany/FELIX (C++)
velocity solve

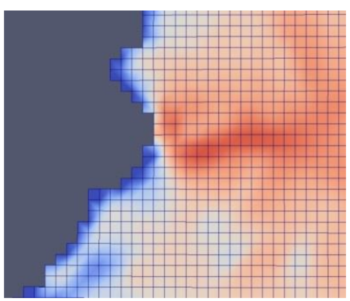
MPAS LI-Albany

LandIce_model

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

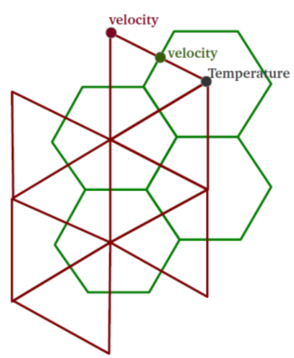
C++/Fortran Interface, Mesh Conversion

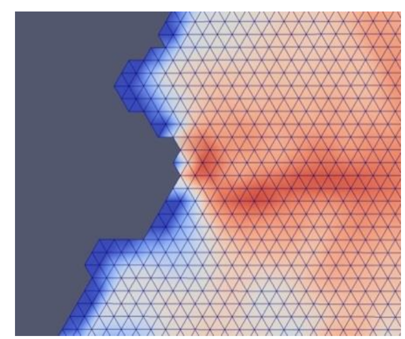
output file



- Structured hexahedral meshes (rectangles extruded to hexes).

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





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

Outline

Albany/FELIX = new land-ice solver with ***next-generation*** capabilities.

- **Overview:**
  - ***First Order (FO) Stokes*** model.
  - ***Albany/FELIX*** First-Order (FO) Stokes diagnostic solver.
  - ***CISM-Albany*** and ***MPAS-Albany*** codes for prognostic simulations of the ice sheet evolution.
- **Uncertainty Quantification (UQ):**
  - Deterministic inversion.
  - Bayesian calibration.
  - Forward propagation of uncertainty.
- **Performance portability.**
- **Summary** and ongoing work.
- **Questions?**

Uncertainty Quantification (UQ) Problem Definition

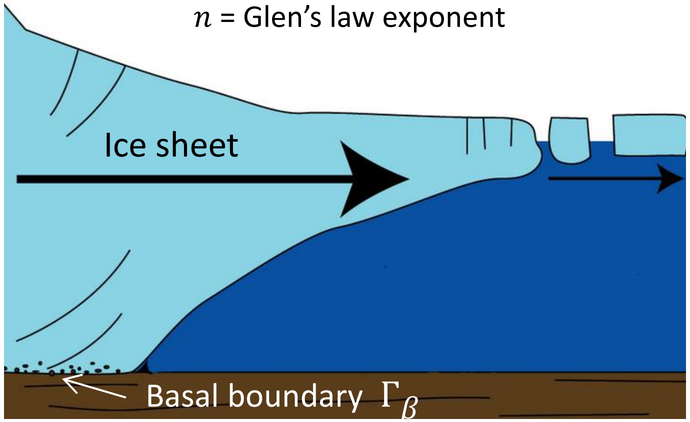
Quantity of Interest (QoI) in Ice Sheet Modeling:
total ice mass loss/gain during 21st century
→ ***sea level rise prediction.***

There are several sources of uncertainty, most notably:

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

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

n = Glen's law exponent



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

Uncertainty Quantification (UQ) Problem Definition

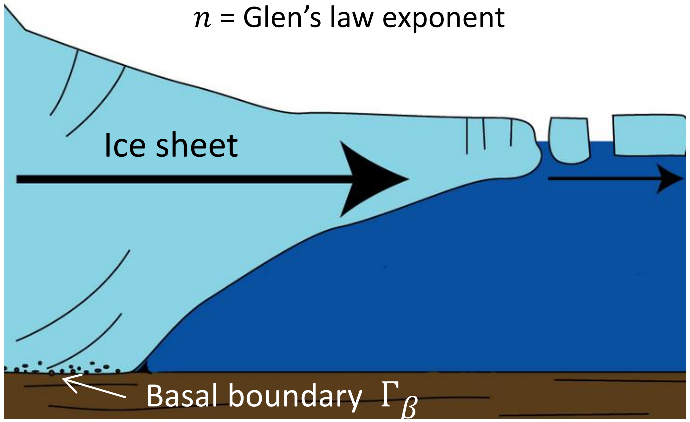
Quantity of Interest (QoI) in Ice Sheet Modeling:
total ice mass loss/gain during 21st century
→ ***sea level rise prediction.***

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

n = Glen's law exponent

There are several sources of uncertainty, most notably:

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



As a first step, we focus on effect of uncertainty in **basal friction (β)** only.

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

Uncertainty Quantification Workflow

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

- **Deterministic inversion:** perform adjoint-based deterministic inversion to estimate initial ice sheet state (i.e., characterize the present state of the ice sheet to be used for performing prediction runs).
- **Bayesian calibration:** construct the posterior distribution using Markov Chain Monte Carlo (MCMC) run on an emulator of the forward model.
- **Forward propagation:** sample the obtained distribution and perform ensemble of forward propagation runs to compute the uncertainty in the QoI.

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

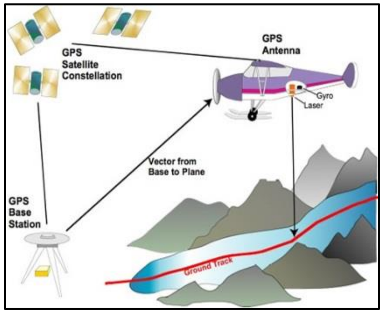
What is the impact of uncertain parameters in the model on quantities of interest (QoI)?

Deterministic Inversion: Estimation of Ice Sheet Initial State

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

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

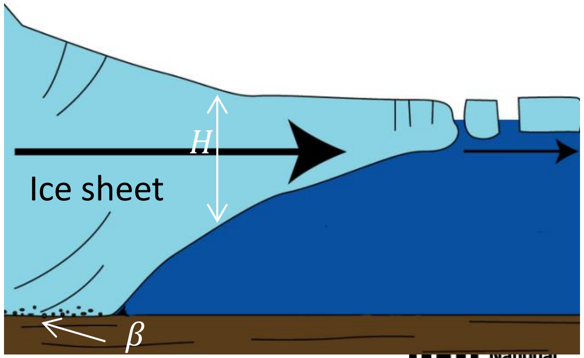
Available data/measurements:
- Ice extent and surface topography.
- Surface velocity.
- Surface mass balance (SMB).
- Ice thickness H (sparse measurements).



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

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

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



Deterministic Inversion: Greenland

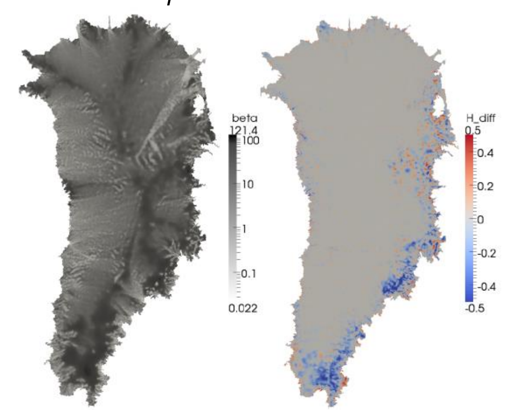
First Order Stokes PDE Constrained Optimization Problem:

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

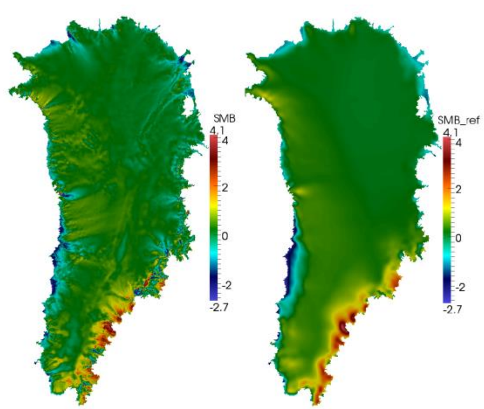
- Minimize difference between:
  - Computed and measured ***surface velocity ($\boldsymbol{u}^{obs}$)*** → *common*
  - Computed divergence flux and measured ***surface mass balance (SMB)*** → *novel*
  - Computed and ***reference thickness (H^{obs})*** → *novel*

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

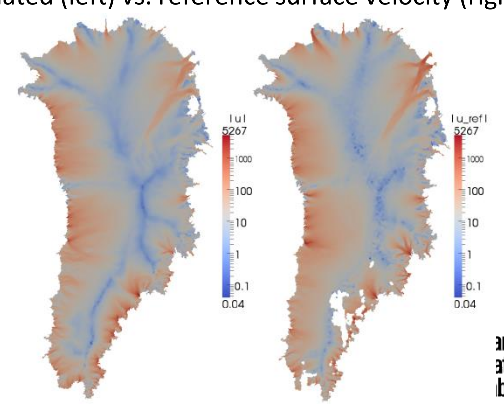
Estimated β $H - H^{obs}$



Estimated divergence (left) vs. reference SMB (right)



Estimated (left) vs. reference surface velocity (right)



<u>*Software for adjoint-based inversion:*</u>

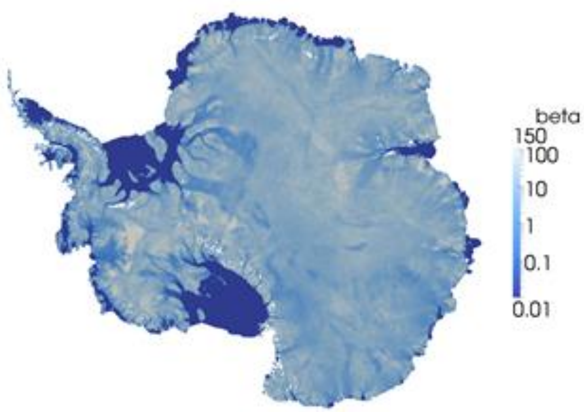
- ***Albany/FELIX*** (assembly)
- ***Trilinos*** (linear/nonlinear solvers)
- ***ROL*** (gradient-based optimization)
  - Limited memory BFGS.
  - Backtrack line-search.

Deterministic Inversion: Antarctica (basal friction only)

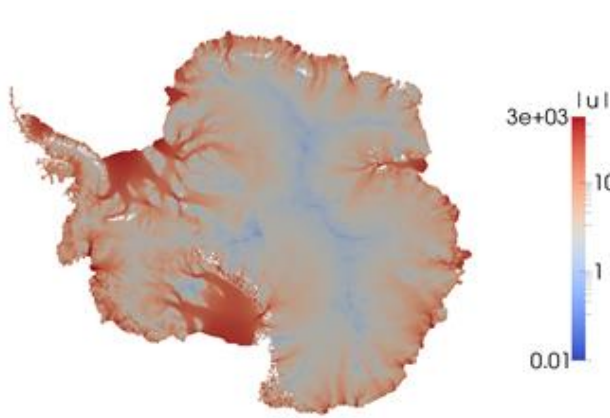
FO Stokes PDE Constrained Optimization Problem:

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

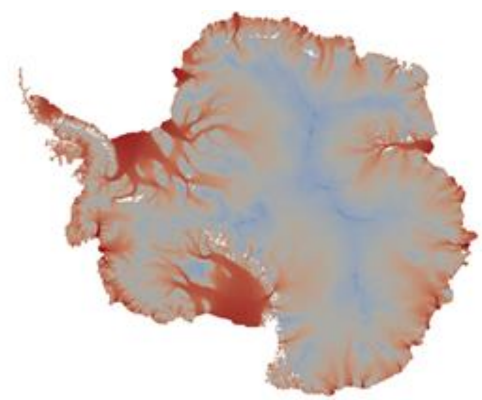
Geometry: Cornford, Martin *et al.* (in prep.)
Bedmap2: Fretwell et al., 2013
Temperature: Pattyn, 2010.



β (kPa y/m) obtained through inversion



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



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

Antarctic ice sheet inversion performed on **700K** parameters

Bayesian Calibration: Demonstration of Workflow using KLE

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

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

Bayesian Calibration: Demonstration of Workflow using KLE

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

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

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

Bayesian Calibration: Demonstration of Workflow using KLE

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

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

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

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

Bayesian Calibration: Demonstration of Workflow using KLE

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

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

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

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

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

Bayesian Calibration: Demonstration of Workflow using KLE

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

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

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

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

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

Bayesian Calibration: Demonstration of Workflow using KLE

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

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

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

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

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

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

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

Bayesian Calibration: Demonstration of Workflow using KLE

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

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

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

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

Offline

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

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

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

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

Sandia National Laboratories

Bayesian Calibration: Demonstration of Workflow using KLE

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

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

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

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

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

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

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

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

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

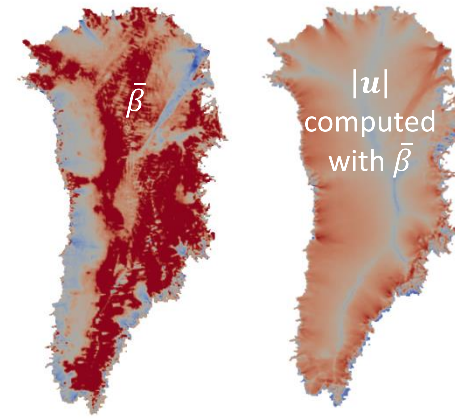
*In practice, expansion is done on $\log(\beta)$ to avoid negative values of β.

Sandia National Laboratories

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

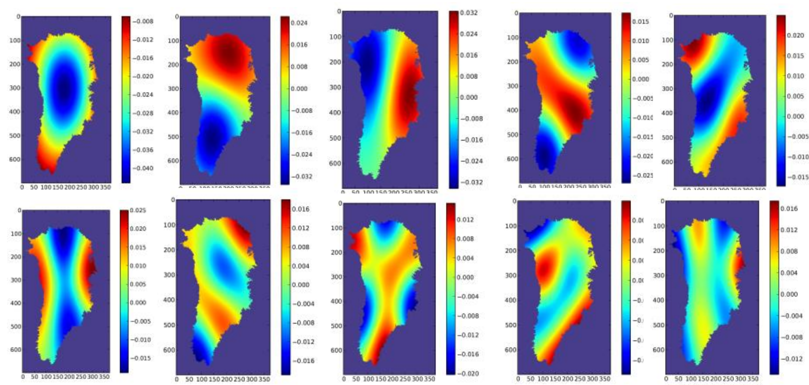
Bayesian Calibration: Illustration on 4km GIS Problem

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

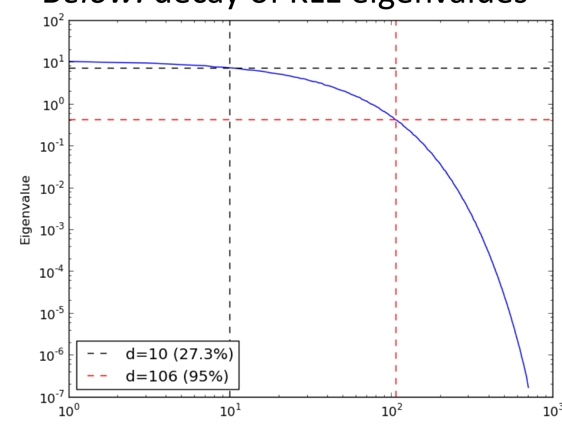


Modes 1-5:

Modes 6-10:



Below: decay of KLE eigenvalues



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

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

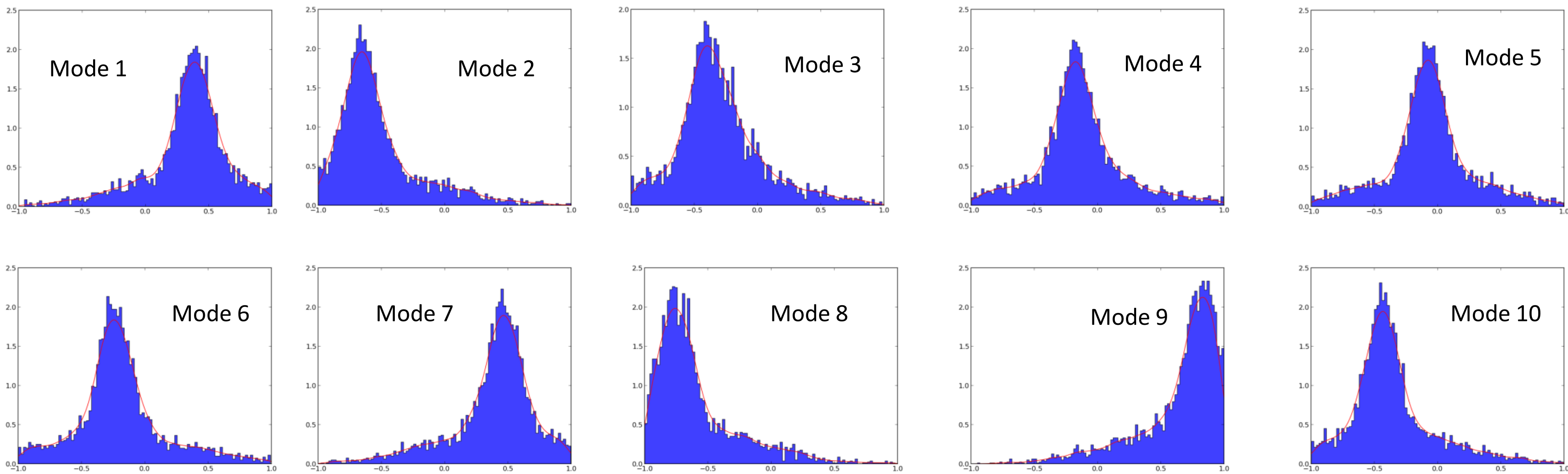
- PCE emulator was formed for the mismatch $J(\beta)$ using uniform $[-1,1]$ prior distributions and 286* high-fidelity runs on Hopper (*DAKOTA*).
- For calibration, MCMC was performed on the PCE with 2K samples *(QUESO).*

*286 points = 3rd degree polynomial in 10D

Sandia National Laboratories

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

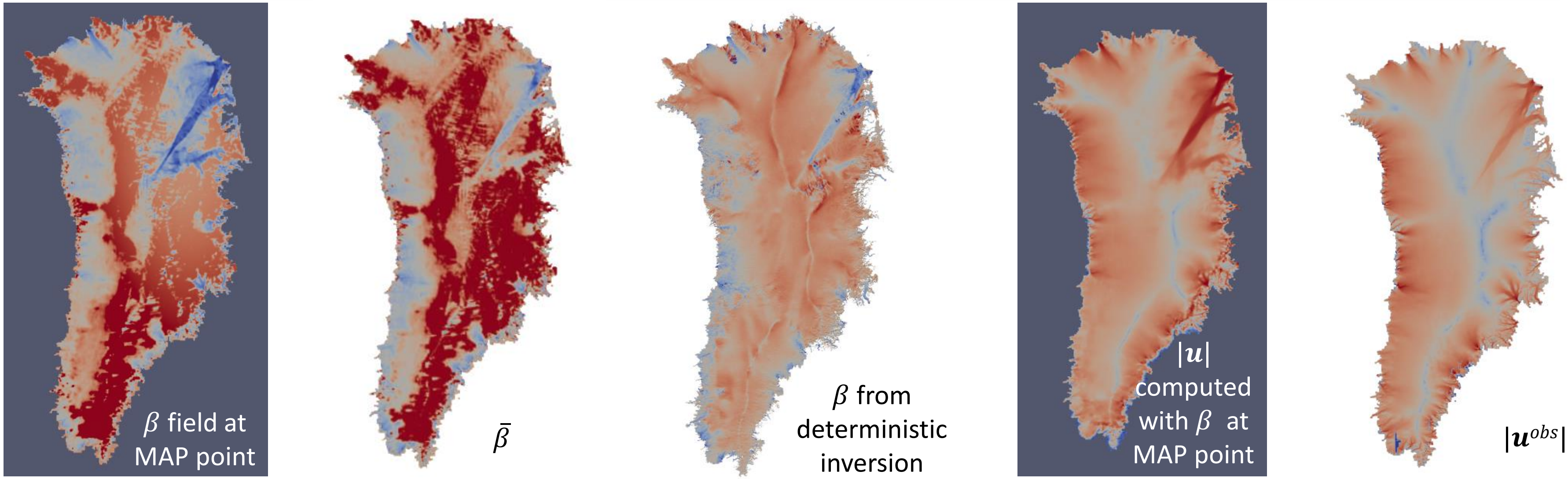
- ***Posterior distributions*** for 10 KLE coefficients:



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

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

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



- Ice is too fast at MAP point. Possible explanations:
    - Surrogate error (based on cross-validation).
    - Mean field error.
    - Bad modes and/or not enough modes.

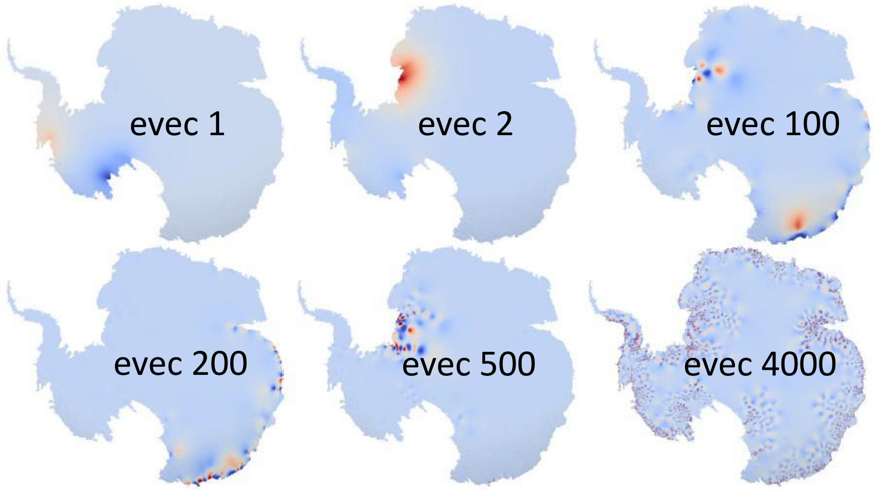
Mismatch $J(\beta)$ at MAP point: $1.87 \times$ mismatch at $\bar{\beta}$

Bayesian Calibration: Building Gaussian Distribution using Hessian

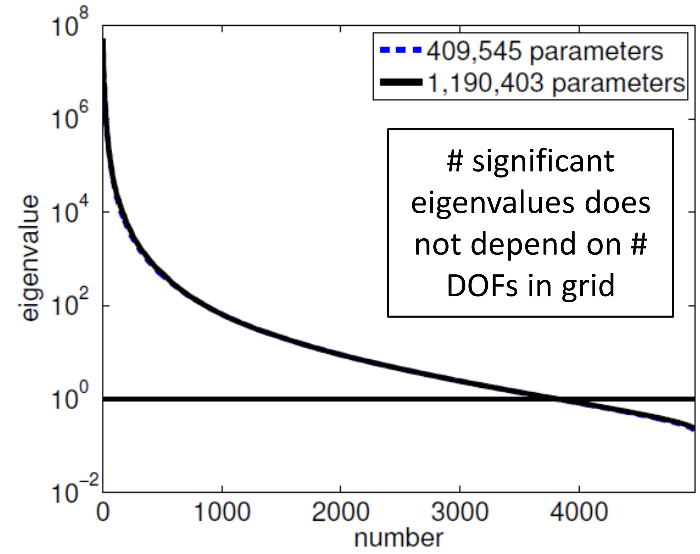
- Hessian of the merit functional (velocity mismatch) can provide a way to compute the covariance of a Gaussian posterior:

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

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



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



significant eigenvalues does not depend on # DOFs in grid

Issue: *there are too many (~1000) significant eigenvalues*

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

Sandia National Laboratories

Forward Propagation

PCE Emulator

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

Model realizations
Forward propagation
(e.g., 2000-2050)

DAKOTA, Albany/FELIX
QoI(β)
(total ice mass loss)

- Parameter (β) distribution can either be assumed to be Gaussian (based on Hessian information) or can be the result of Bayesian calibration.
- Emulator is built using *DAKOTA* coupled with *CISM-Albany* for forward runs.
  - Use ***compressed sensing*** to adaptively select significant modes that affect QoI. The hope is that only a few modes affect the QoI.
  - Could use cheaper physical models to reduce computational time of forward model.
- MCMC (*QUESO)* used to perform uncertainty propagation.

DAKOTA

QUEST

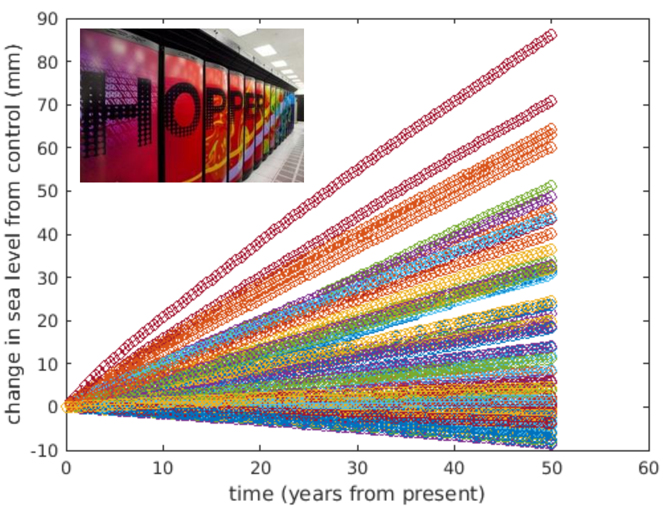
Sandia National Laboratories

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

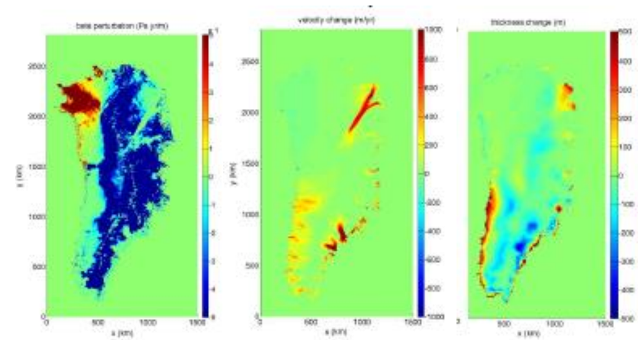
Forward Propagation: Illustration on 4km GIS Problem

Procedure:

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



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



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

- We then used the results of these runs to create a PCE emulator for the SLR.

- Using emulator, propagated posterior distributions computed in Bayesian calibration (using KLE) through the model to get posteriors on SLR (MCMC on PCE emulator with 2K samples).

*66 points = 2D polynomial in 10D.

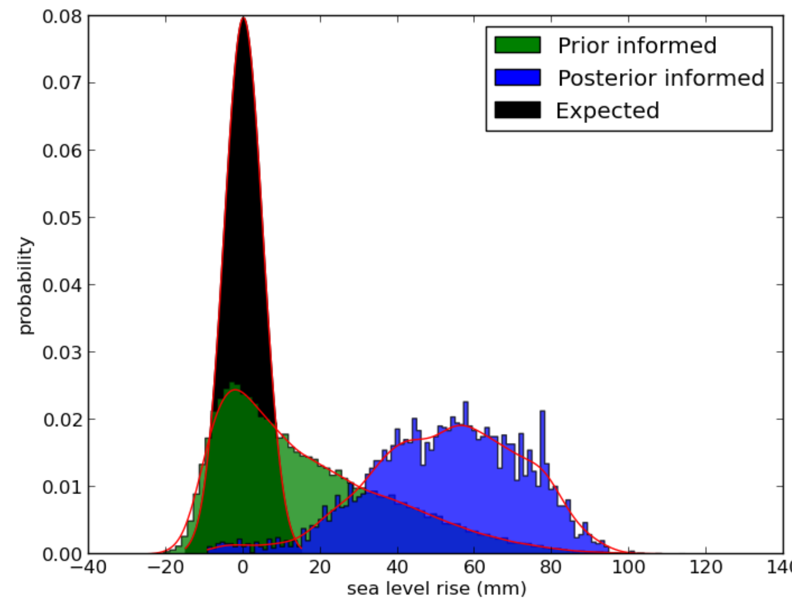
Forward Propagation: Illustration on 4km GIS Problem (cont'd)

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

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

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

PDF of SLR



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

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

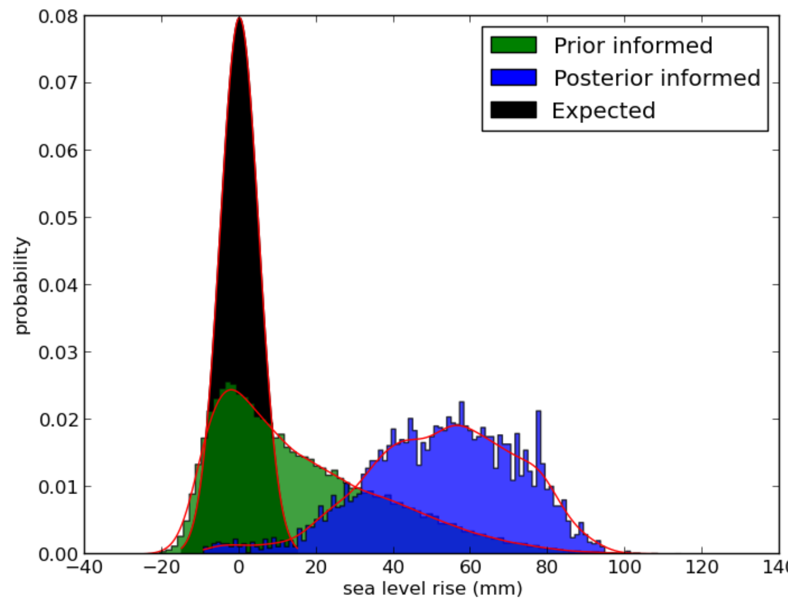
Forward Propagation: Illustration on 4km GIS Problem (cont'd)

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

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

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

PDF of SLR



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

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

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

Sandia National Laboratories

Outline

Albany/FELIX = new land-ice solver with ***next-generation*** capabilities.

- **Overview:**
  - ***First Order (FO) Stokes*** model.
  - ***Albany/FELIX*** First-Order (FO) Stokes diagnostic solver.
  - ***CISM-Albany*** and ***MPAS-Albany*** codes for prognostic simulations of the ice sheet evolution.
- **Uncertainty Quantification (UQ):**
  - Deterministic inversion.
  - Bayesian calibration.
  - Forward propagation of uncertainty.
- **Performance portability.**
- **Summary** and ongoing work.
- **Questions?**

Performance-Portability via *Kokkos*

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

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

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

Performance-Portability via *Kokkos*

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

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

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

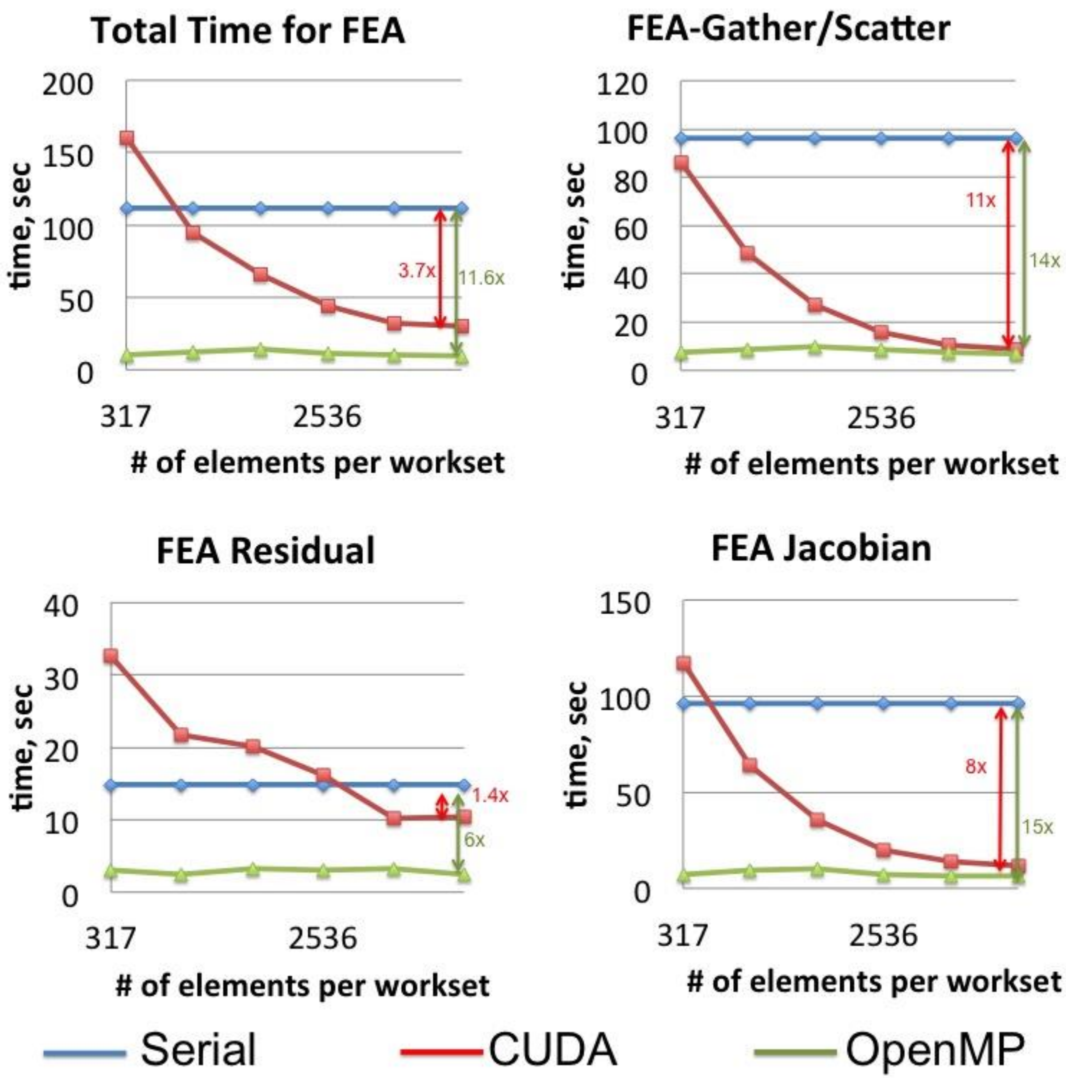
- **Finite element assembly** in *Albany* has recently been rewritten using *Kokkos* functors.

Kokkos-ification of Finite Element Assembly

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

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

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



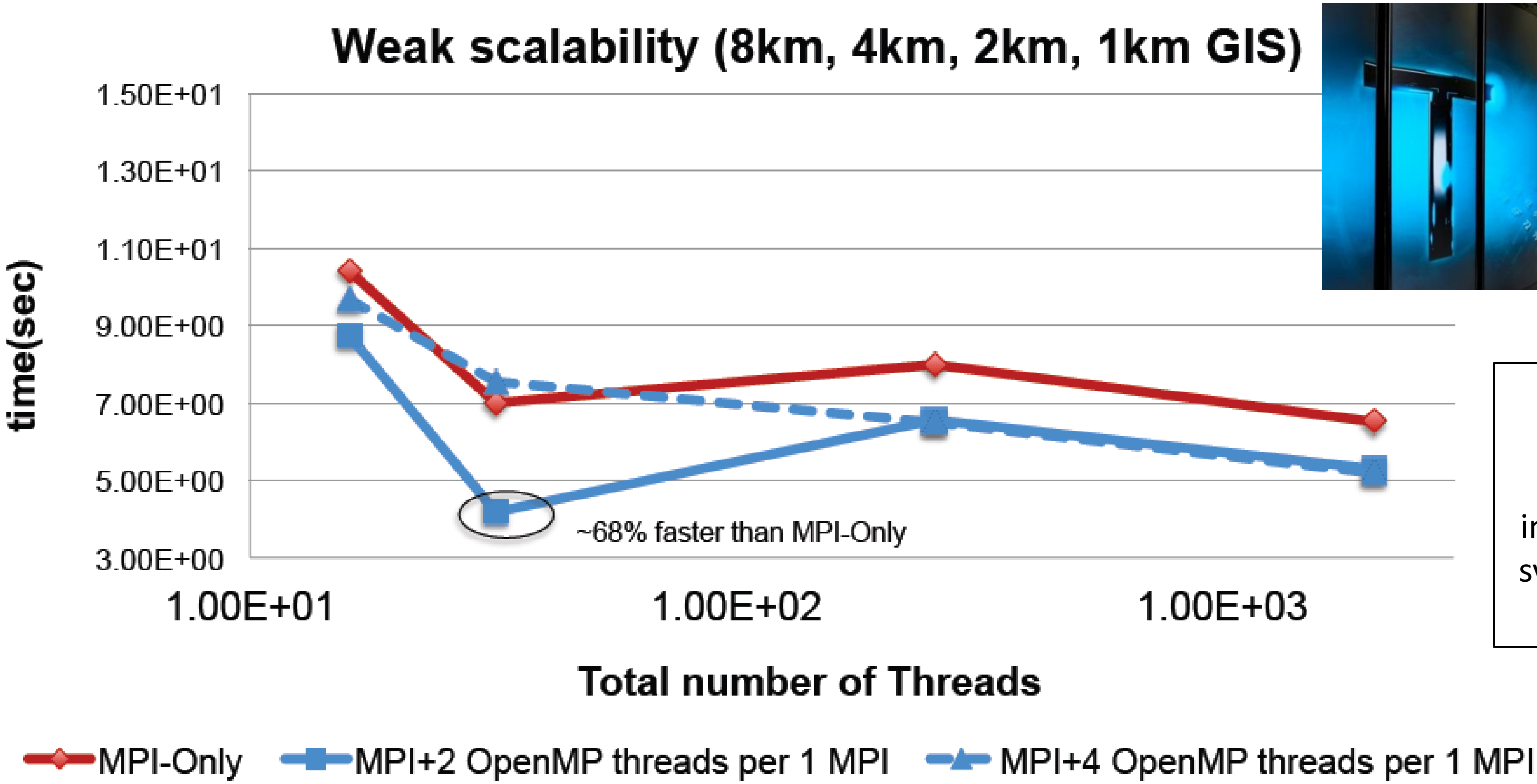
Shannon: *32 nodes*

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

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

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

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



Increasing # OpenMP threads can increase thread synchronization overheads.

Titan: 18,688 AMD Opteron nodes

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

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

Outline

Albany/FELIX = new land-ice solver with ***next-generation*** capabilities.

- **Overview:**
  - ***First Order (FO) Stokes*** model.
  - ***Albany/FELIX*** First-Order (FO) Stokes diagnostic solver.
  - ***CISM-Albany*** and ***MPAS-Albany*** codes for prognostic simulations of the ice sheet evolution.
- **Uncertainty Quantification (UQ):**
  - Deterministic inversion.
  - Bayesian calibration.
  - Forward propagation of uncertainty.
- **Performance portability.**
- **Summary** and ongoing work.
- **Questions?**

Summary and Ongoing Work

Summary: this talk described…

- The development of a finite element land ice solver known as *Albany/FELIX* written using the *Trilinos* libraries.
- Coupling of *Albany/FELIX* to the *CISM* and *MPAS LI* codes for transient simulations of ice sheet evolution.
- Advanced, next generation capabilities (UQ, performance portability) of *Albany//FELIX*.

Ongoing/future work:

- Science runs using *CISM-Albany* and *MPAS-Albany*.
- Deploy UQ workflow with better basis than KLE (e.g., Hessian eigenvectors).
- Continued porting of code to new architecture supercomputers (GPUs on Titan, Summit, Cori Phase I).
- Delivering code to climate community and coupling to earth system models.

Funding/Acknowledgements

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

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

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

Computing resources: *NERSC, OLCF.*

Thank you! Questions?

References

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

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

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

[4] **I. Tezaur**, R. Tuminaro, M. Perego, A. Salinger, S. Price. "On the scalability of the *Albany/FELIX* first-order Stokes approximation ice sheet solver for large-scale simulations of the Greenland and Antarctic ice sheets", *MSESM/ICCS15*, Reykjavik, Iceland (June 2014).

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

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

References (cont'd)

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

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

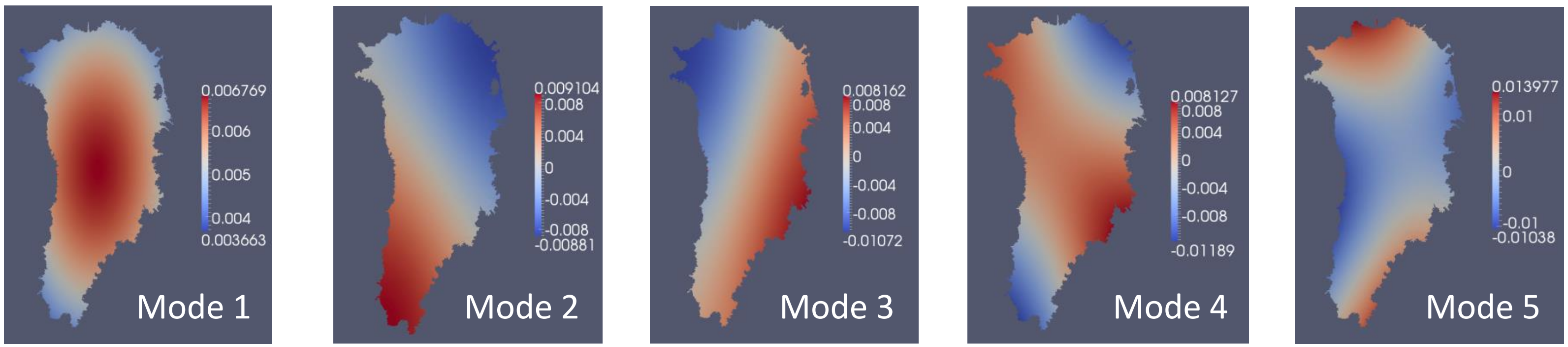
[9] F. Pattyn. "Antarctic subglacial conditions inferred from a hybrid ice sheet/ice stream model", *Earth and Planetary Science Letters 295 (2010).*

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

Sandia National Laboratories

Appendix: Preliminary Results for GIS Bayesian Inference

- 5 KLE modes capture 95% of covariance energy → parallel C++/Trilinos code (**Anasazi**).



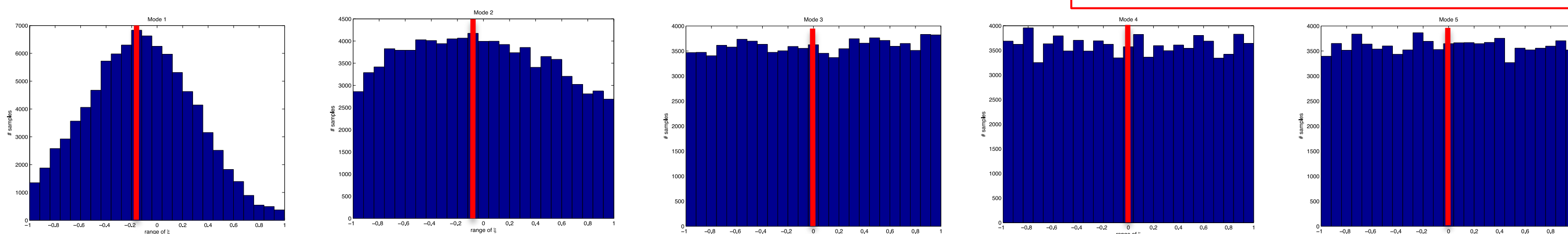
- Mismatch = sum of squares of surface velocity discrepancy → **Albany.**
- Polynomial chaos expansion (PCE) was formed for the mismatch over ξ_k using uniform prior distributions and isotropic sparse grid level = 3 → **DAKOTA**.
- Markov Chain Monte Carlo (MCMC) was performed on the PCE with 100K samples → **QUESO**.

***Collaborators**: M. Perego, J. Jakeman, M. Eldred, L. Swiler (SNL)*

Appendix: Preliminary Results for GIS Bayesian Inference (cont'd)

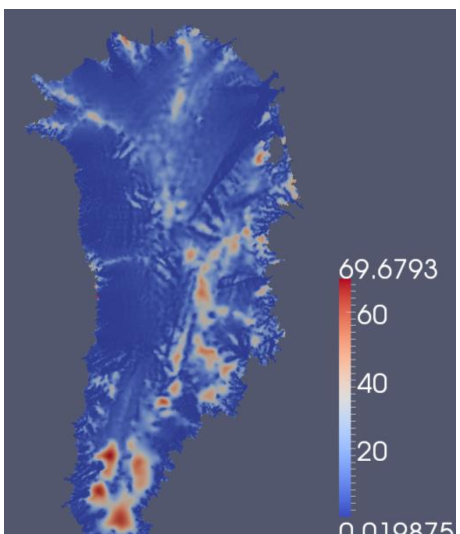
MAP solution:
$x = (-0.16, -0.08, 0, 0, 0)$

- Posterior distributions for the 5 KLE coefficients:



Coefficient 1 | Coefficient 2 | Coefficient 3 | Coefficient 4 | Coefficient 5

- Inference of KLE random field:

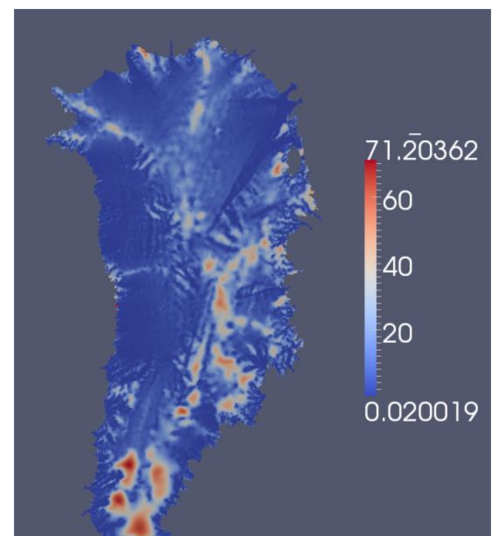


True β field

Left: true β field

Right: *reconstructed β field*

Good agreement between true and reconstructed β field!



Reconstructed β field