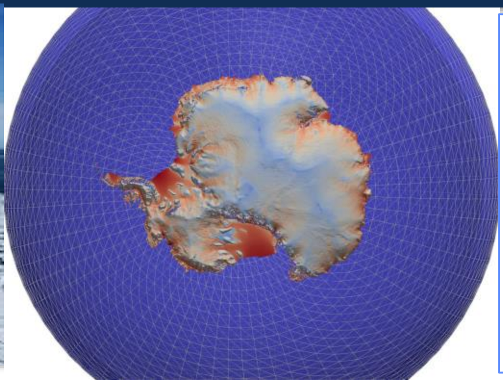
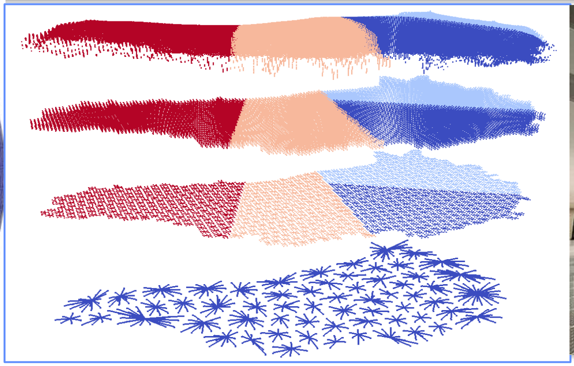
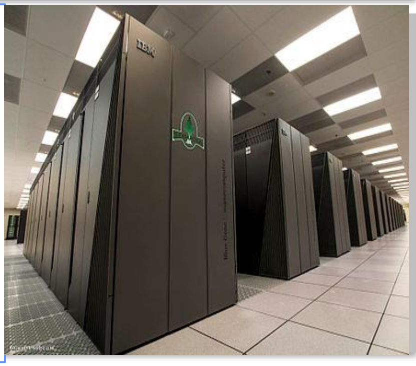
Exceptional service in the national interest

Sandia National Laboratories







The *Albany/FELIX* First-Order Stokes Finite Element Ice Sheet Dynamical Core Built Using Trilinos Software Components: Performance, Next-Generation Capabilities and Validation

I. Tezaur[1], A. Salinger[1], M. Perego[1], R. Tuminaro[1], S. Price[2], J. Watkins[1], G. Hansen[1]

[1] Sandia National Laboratories, Albuquerque, NM and Livermore, CA, USA.

[2] Los Alamos National Laboratory, Los Alamos, NM, USA.

SIAM MPE 2016   Philadelphia, PA   Sept. 30 – Oct. 2, 2016

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

SAND2016-8943 C

PISCEES* Project for Land-Ice Modeling

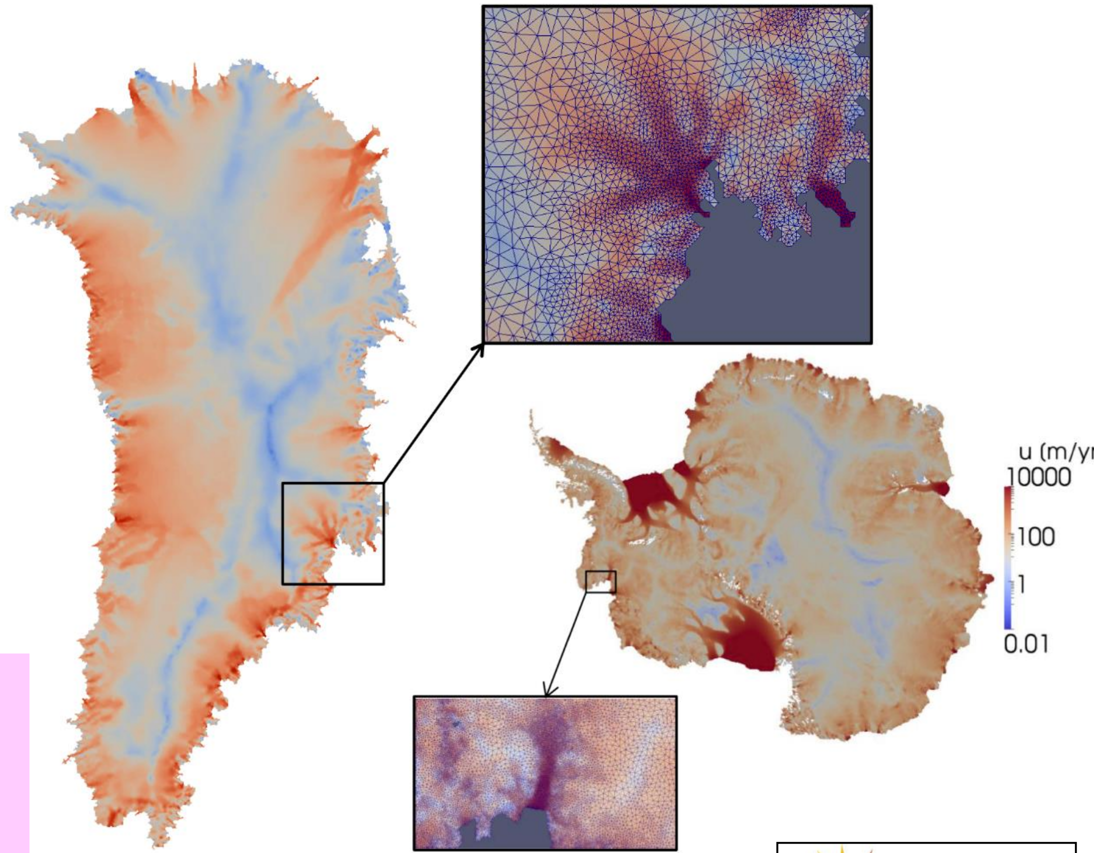
<u>Sandia's Role in the PISCEES Project</u>: to **develop** and **support** a robust and scalable land ice solver based on the "First-Order" (FO) Stokes equations → *Albany/FELIX***

<u>Requirements for *Albany/FELIX*:</u>

- ***Unstructured grid*** finite elements.
- ***Scalable, fast*** and ***robust***
- ***Verified*** and ***validated.***
- ***Portable*** to new/emerging architecture machines (multi-core, many-core, GPU)
- ***Advanced analysis*** capabilities: deterministic inversion, calibration, uncertainty quantification.

As part of **ACME *DOE earth system model***, solver will provide actionable predictions of 21st century sea-level rise (including uncertainty).

**Finite Elements for Land Ice eXperiments



*PISCEES = Predicting Ice Sheet Climate Evolution at Extreme Scales.

PISCEES* Project for Land-Ice Modeling

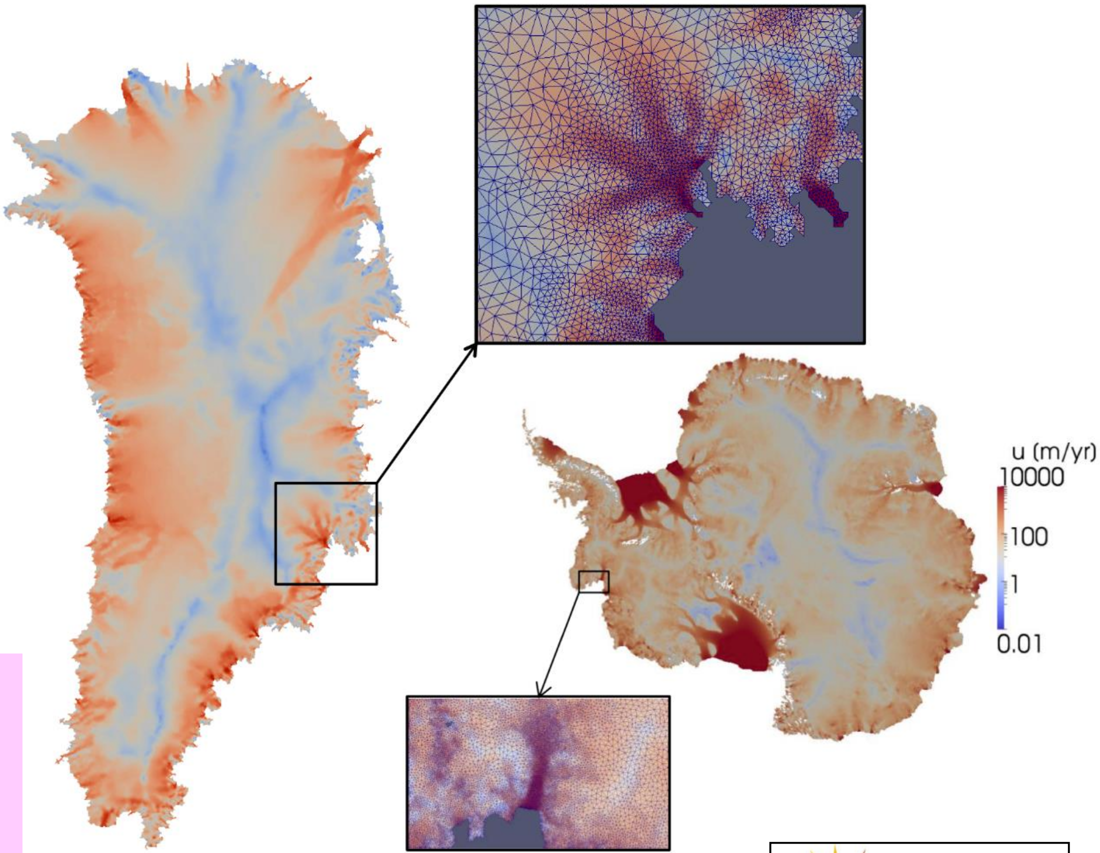
<u>Sandia's Role in the PISCEES Project</u>: to **develop** and **support** a robust and scalable land ice solver based on the "First-Order" (FO) Stokes equations → *Albany/FELIX***

<u>Requirements for *Albany/FELIX*</u>:

- ***<u>Unstructured grid</u>*** finite elements.
- ***<u>Scalable</u>, fast*** and ***robust***
- ***Verified*** and ***<u>validated</u>*.**
- ***<u>Portable</u>*** to new/emerging architecture machines (multi-core, many-core, GPU)
- ***Advanced analysis*** capabilities: deterministic inversion, calibration, uncertainty quantification.

As part of **ACME *DOE earth system model***, solver will provide actionable predictions of 21st century sea-level rise (including uncertainty).

**Finite Elements for Land Ice eXperiments



*PISCEES = Predicting Ice Sheet Climate Evolution at Extreme Scales.

Outline

1. Ice Sheet Equations and Codes.

2. Mesh Adaptivity.

3. Scalability.

4. Performance Portability.

5. Validation.

6. Summary and Future Work.

Outline

1. Ice Sheet Equations and Codes.

2. Mesh Adaptivity.

3. Scalability.

4. Performance Portability.

5. Validation.

6. Summary and Future Work.

Steps towards creating a production-ready code

Outline

1. Ice Sheet Equations and Codes.

2. Mesh Adaptivity.

3. Scalability.

4. Performance Portability.

5. Validation.

6. Summary and Future Work.

Steps towards creating a production-ready code

Ice Sheet Equations and Codes

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

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

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

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

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

Ice Sheet Equations and Codes

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

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

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

Code:

Albany =multi-physics PDE code

Albany/FELIX***

Trilinos

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

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

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

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

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

Ice Sheet Equations and Codes

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

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

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

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

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

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

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

Code:

Albany =multi-physics PDE code

Albany/FELIX***

Trilinos

CESM COMMUNITY EARTH SYSTEM MODEL

MPAS Model for Prediction Across Scales

CISM/Albany
MPAS/Albany

ACME Accelerated Climate Model for Energy

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

Outline

1. Ice Sheet Equations and Codes.

2. Mesh Adaptivity.

3. Scalability.

4. Performance Portability.

5. Validation.

6. Summary and Future Work.

Steps towards creating a production-ready code

Mesh Adaptivity

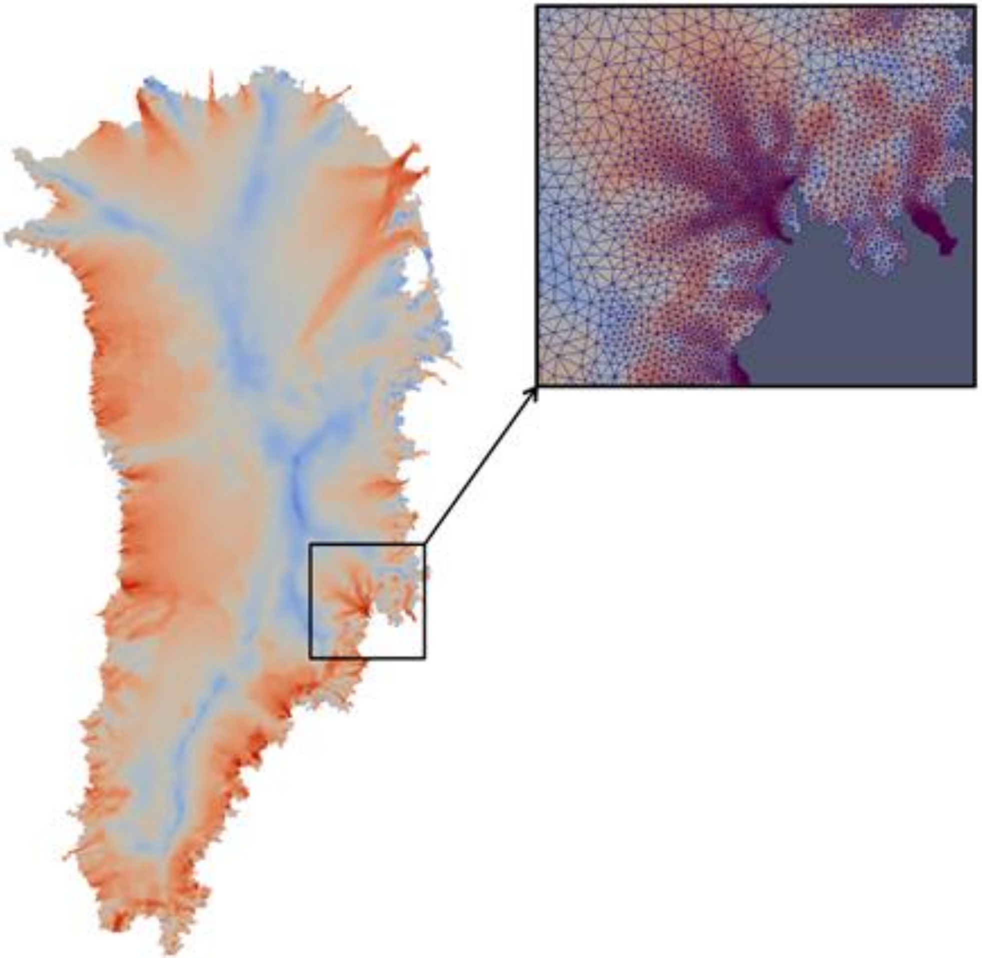
Why?

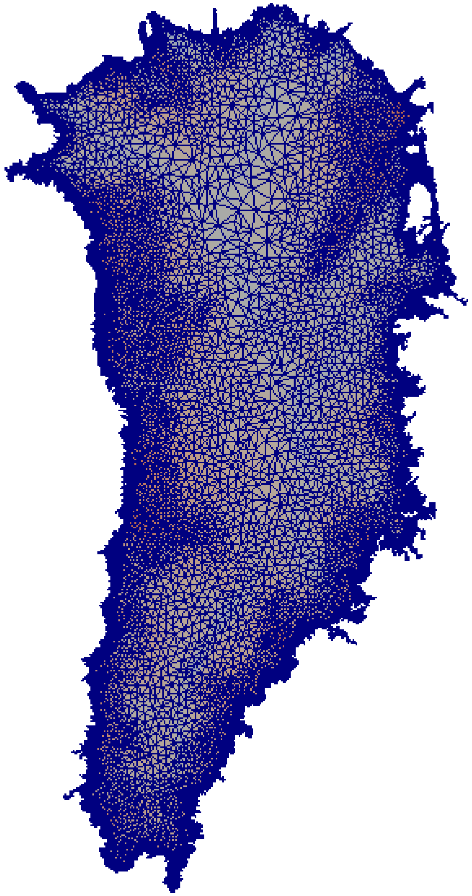
New capability for land-ice solver!

Mesh Adaptivity

Why? To ***concentrate*** computational power where it is needed.

New capability for land-ice solver!

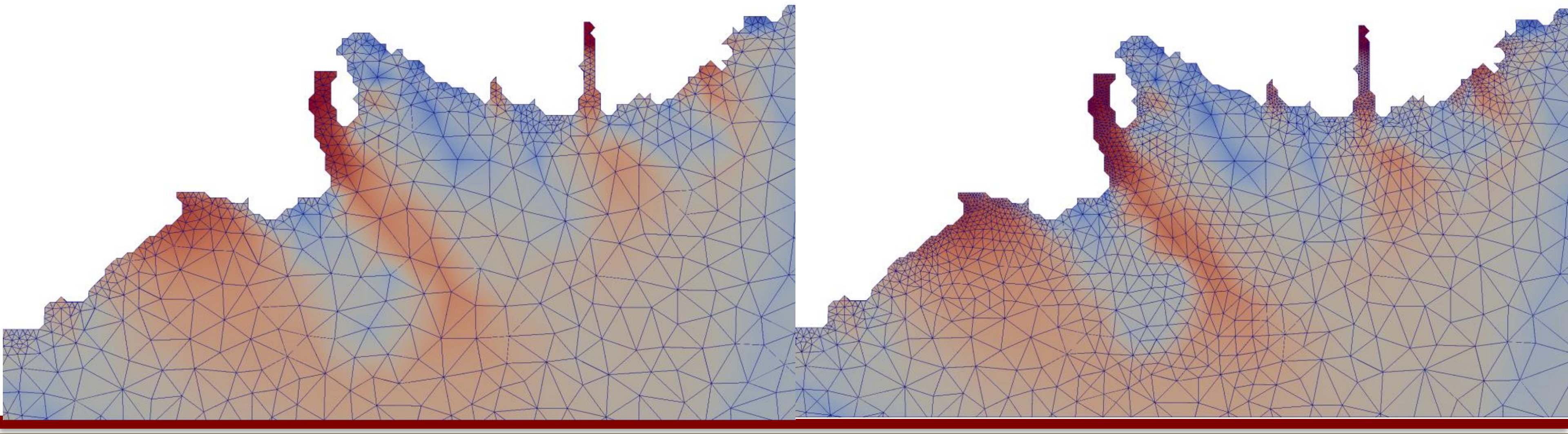




Mesh Adaptivity

New capability for land-ice solver!

How?

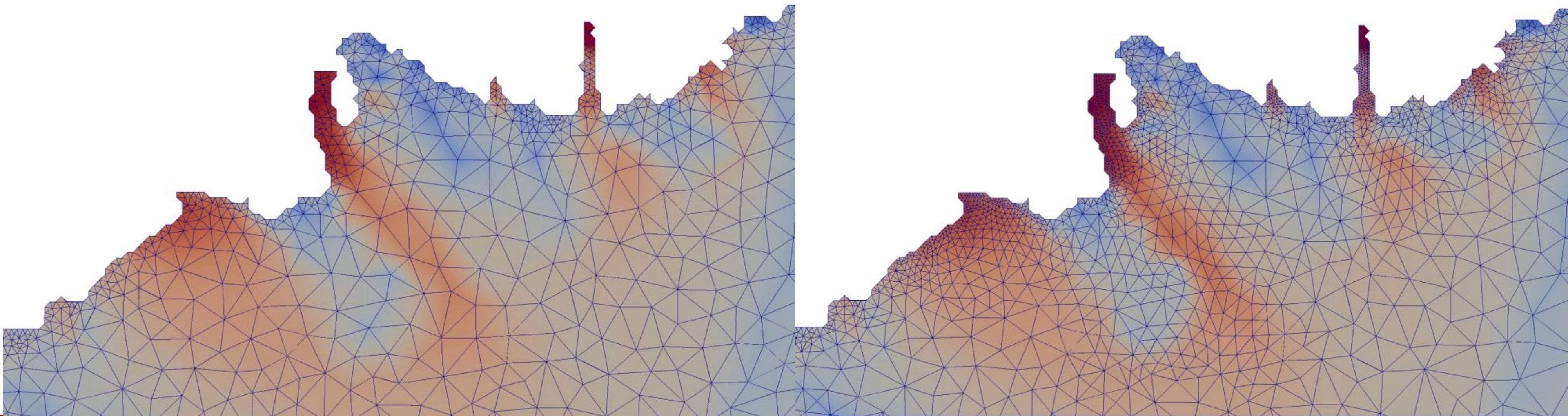


Mesh Adaptivity

New capability for land-ice solver!

How?

PAALS = Parallel Albany Adaptive Loop with SCOREC



Mesh Adaptivity

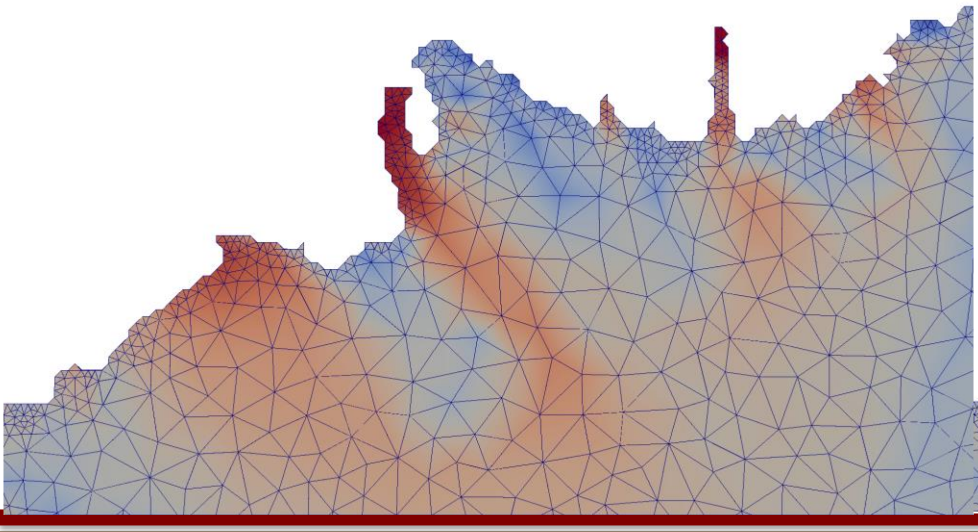
How?

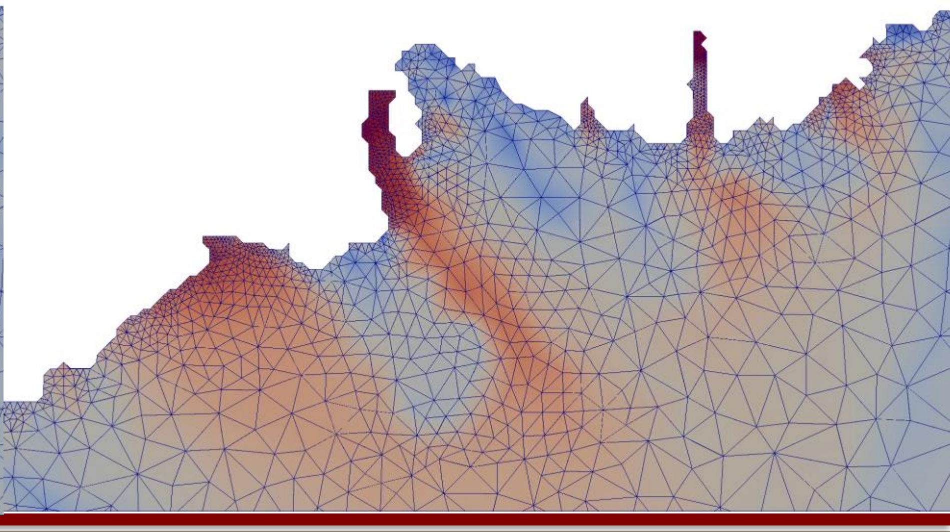
PAALS = Parallel Albany Adaptive Loop with SCOREC*

New capability for land-ice solver!

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

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





Mesh Adaptivity

How?

PAALS = Parallel Albany Adaptive Loop with SCOREC*

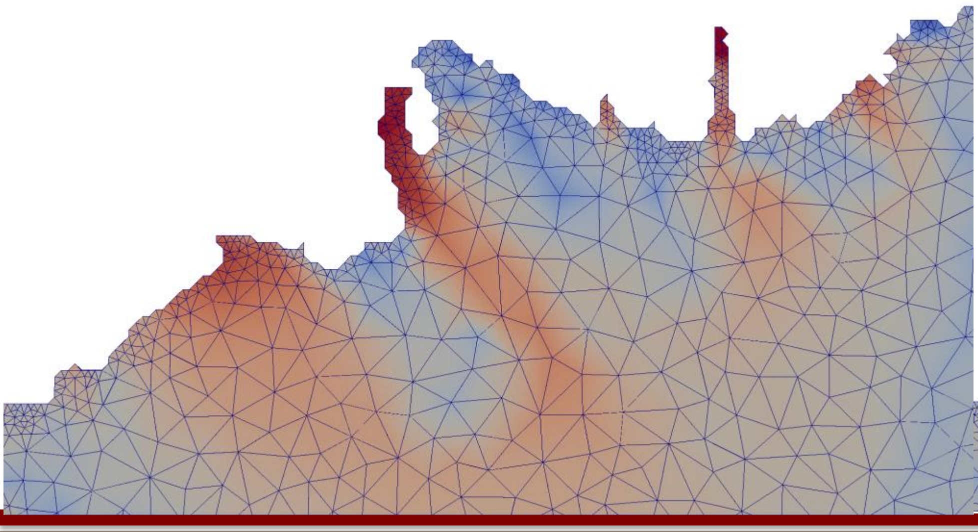
New capability for land-ice solver!

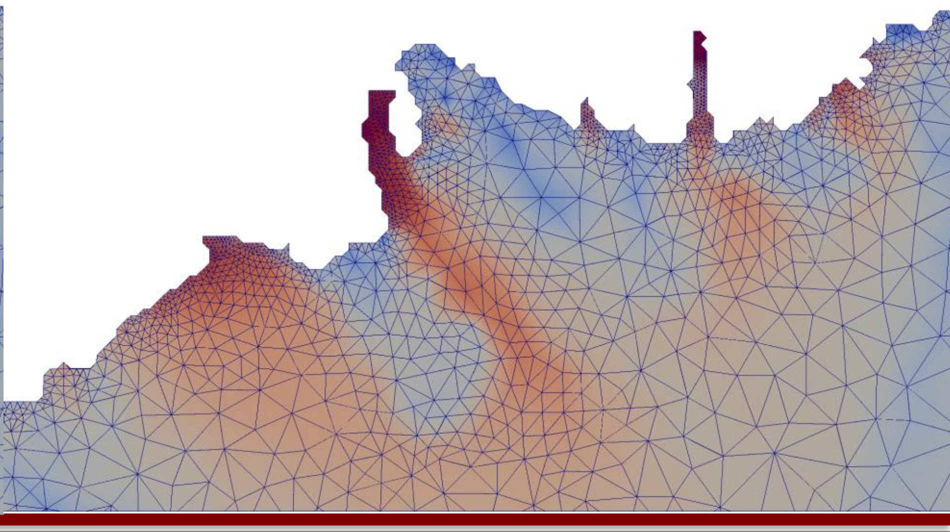
- In collaboration with ***Rensselaer Polytechnical Institute*** (M. Shephard, C. Smith, D. Ibanez): added mesh adaptation capabilities (PAALS) to *Albany.*

PAALS provides:

- Fully-coupled, ***in-memory adaptation*** and solution transfer services.

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





Mesh Adaptivity

How?

PAALS = Parallel Albany Adaptive Loop with SCOREC*

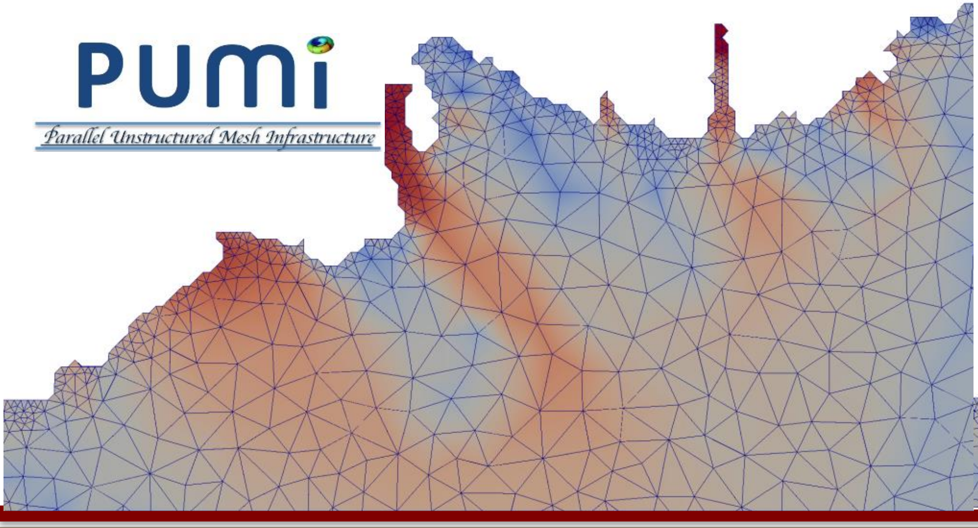
New capability for land-ice solver!

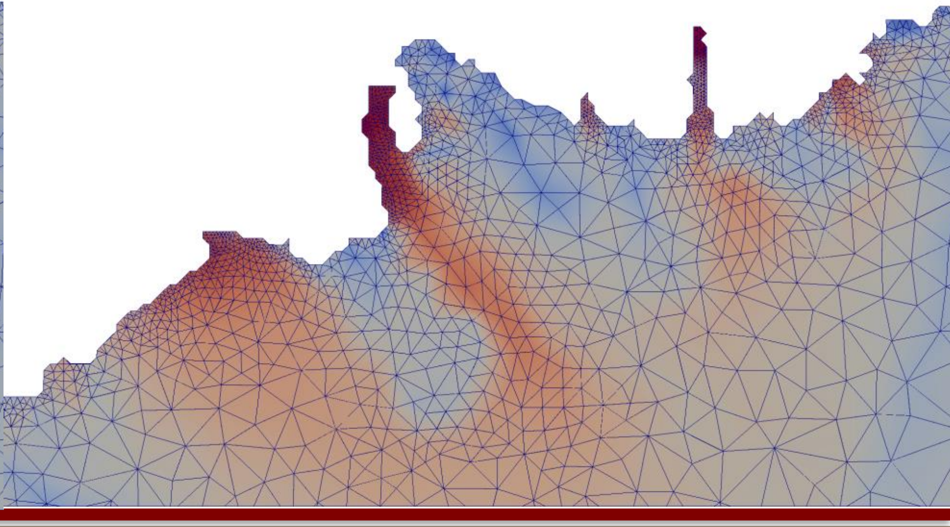
- In collaboration with ***Rensselaer Polytechnical Institute*** (M. Shephard, C. Smith, D. Ibanez): added mesh adaptation capabilities (PAALS) to *Albany.*

*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.





Mesh Adaptivity

How?

PAALS = Parallel Albany Adaptive Loop with SCOREC*

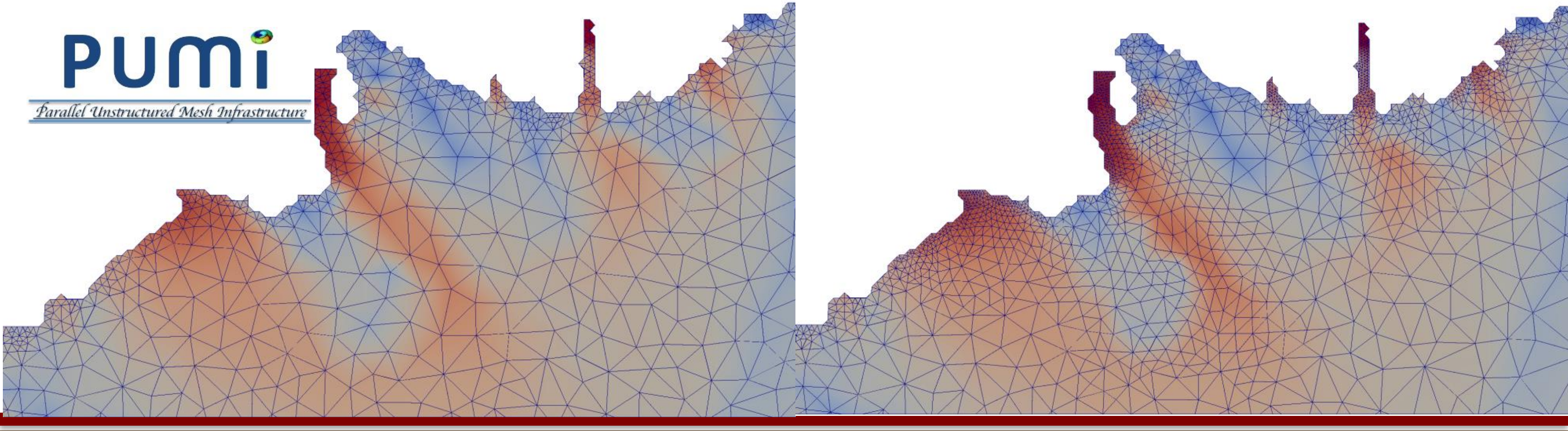
New capability for land-ice solver!

- In collaboration with ***Rensselaer Polytechnical Institute*** (M. Shephard, C. Smith, D. Ibanez): 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.***



Mesh Adaptivity

How?

PAALS = Parallel Albany Adaptive Loop with SCOREC*

New capability for land-ice solver!

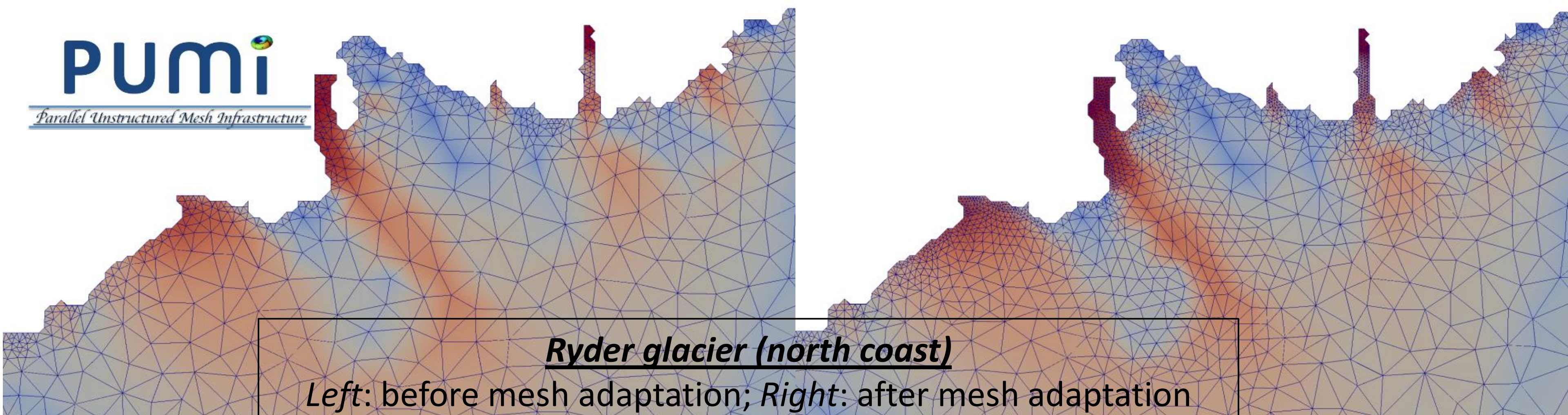
- In collaboration with ***Rensselaer Polytechnical Institute*** (M. Shephard, C. Smith, D. Ibanez): added mesh adaptation capabilities (PAALS) to *Albany.*

*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.

PUMI
Parallel Unstructured Mesh Infrastructure

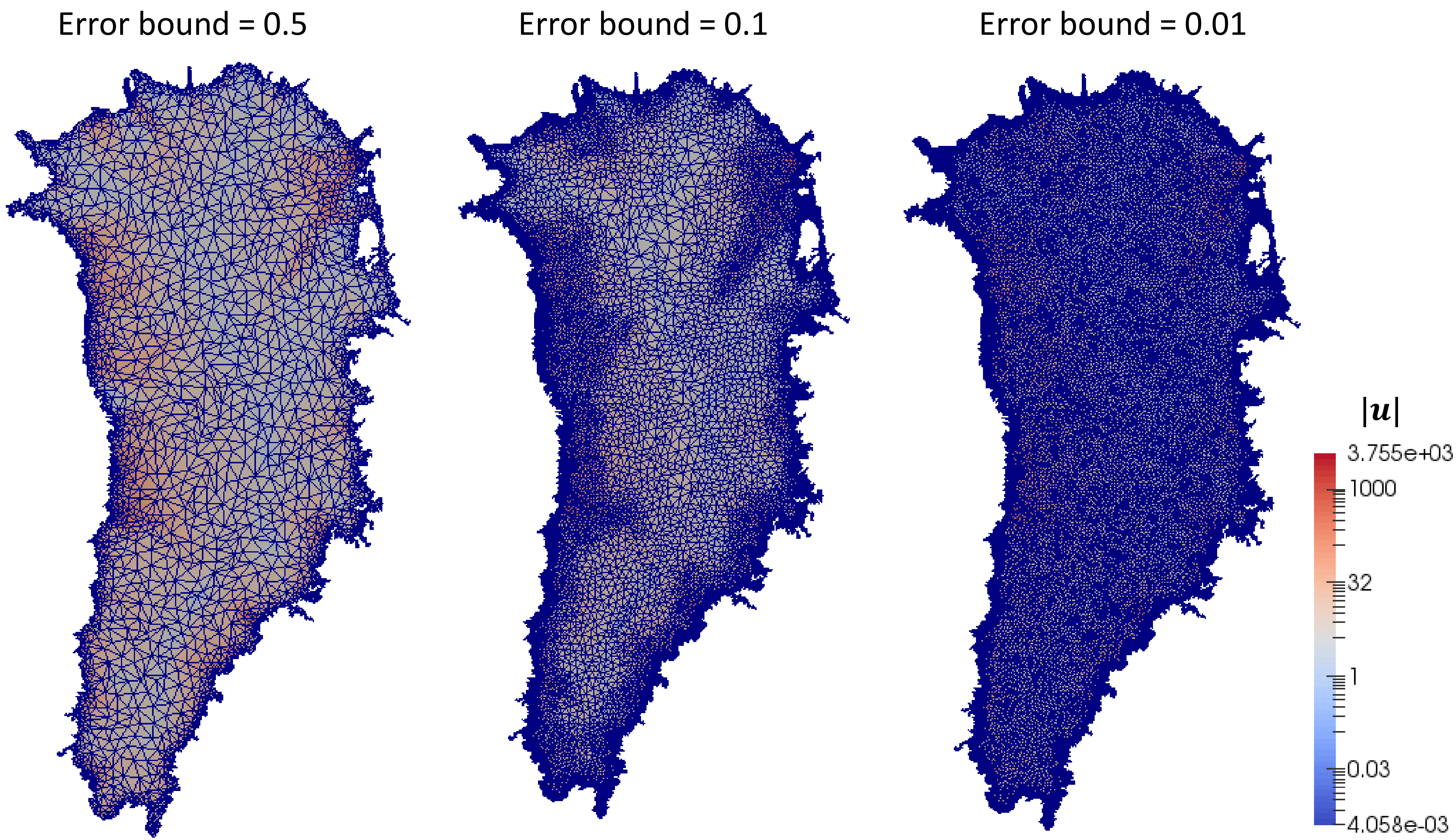


<u>Ryder glacier (north coast)</u>

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

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

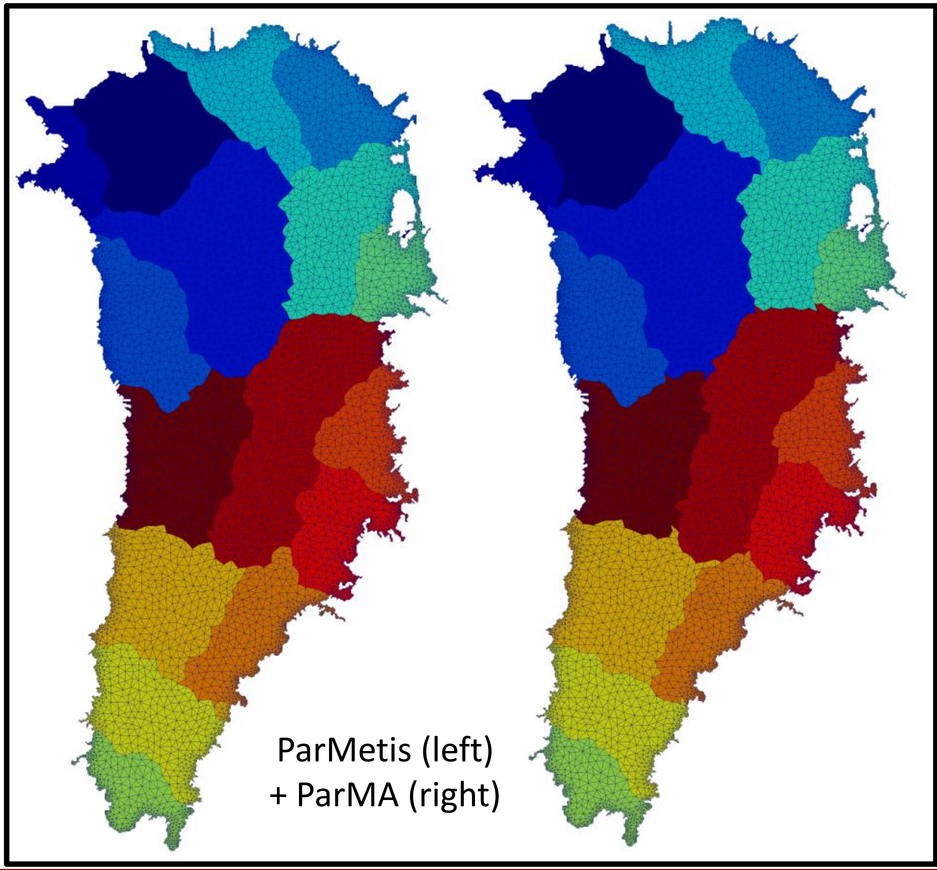
Parallel Adaptive Loop with SCOREC (PAALS)

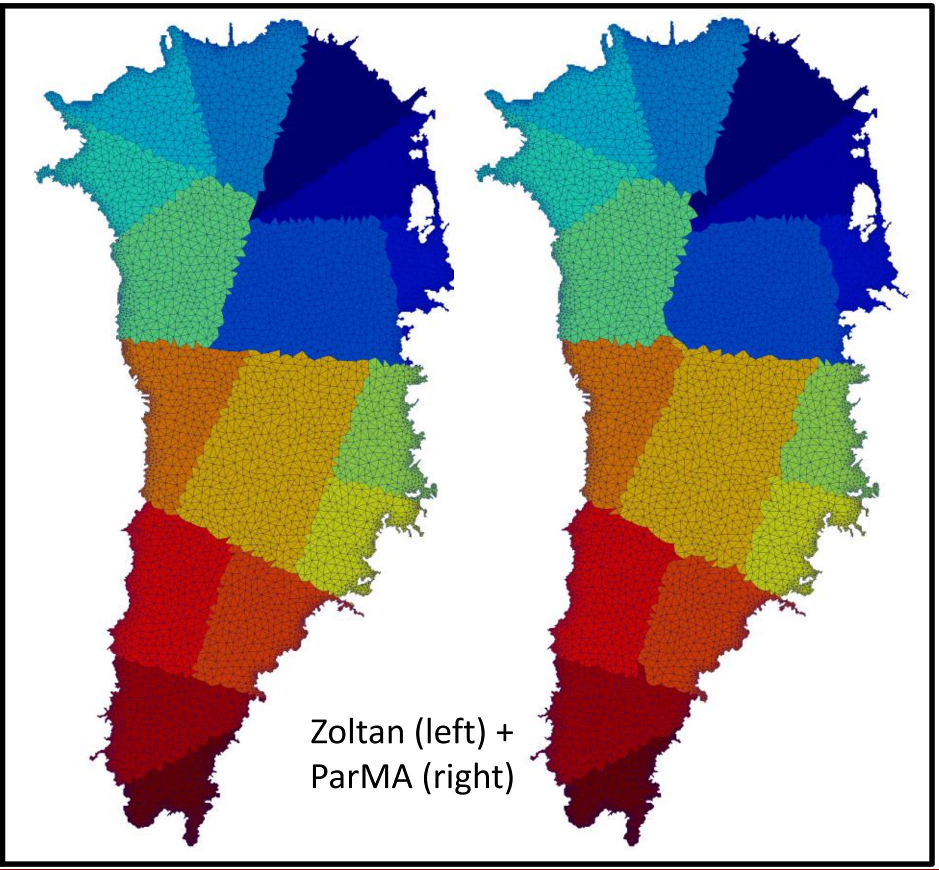


Dynamic Load Balancing

- Partition via ***Zoltan*** or ***ParMetis*** + ***ParMA***.
- **Libraries/Algorithms:**
  - ***ParMetis***: multi-level graph partitioning → minimizes communication
  - ***Zoltan***: Recursive Inertial Bisection (RIB) → faster than graph partitioning
  - ***ParMA***: Unstructured mesh-based diffusive improvement → rebalances refined mesh

In each box below
Left: Initial ParMetis/Zoltan partition
Right: ParMA partition





Mesh Generation/Adaptation Algorithm

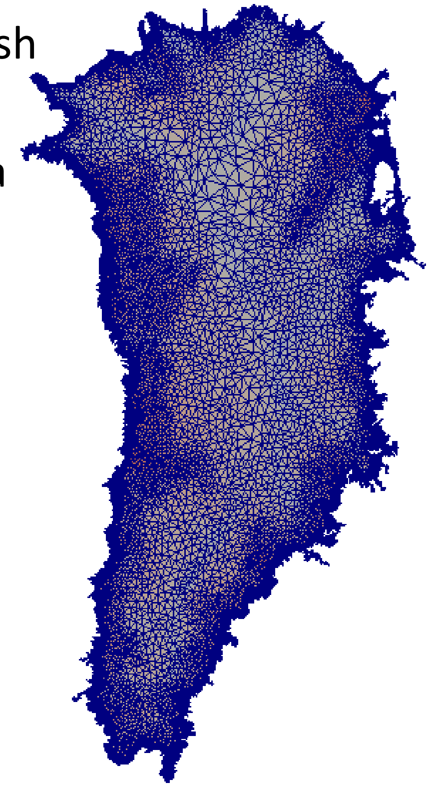
PAALS = Parallel Albany Adaptive Loop with SCOREC

offline

- **Step 1**: determine geometry and generate initial tetrahedral mesh (e.g., using ***Triangle*** or ***Simmetrix***).

online

- **Step 2**: 2D slice of initial mesh ***adaptively refined in-memory*** via ***PAALS*** based on gradient of velocity in ***Albany.***
- **Step 3**: 3D mesh obtained by extruding 2D mesh vertically as ***prisms*** or ***tetrahedra*** in ***Albany.***



Mesh Generation/Adaptation Algorithm

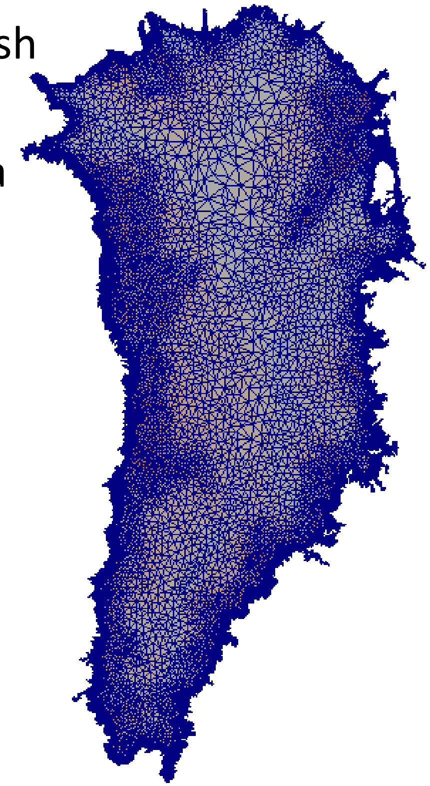
PAALS = Parallel Albany Adaptive Loop with SCOREC

offline

- **Step 1**: determine geometry and generate initial tetrahedral mesh (e.g., using ***Triangle*** or ***Simmetrix***).

online

- **Step 2**: 2D slice of initial mesh ***adaptively refined in-memory*** via ***PAALS*** based on gradient of velocity in ***Albany.***
- **Step 3**: 3D mesh obtained by extruding 2D mesh vertically as ***prisms*** or ***tetrahedra*** in ***Albany.***



Discussion:

PUMI Parallel Unstructured Mesh Infrastructure   Albany   Rensselaer   Sandia National Laboratories

Mesh Generation/Adaptation Algorithm

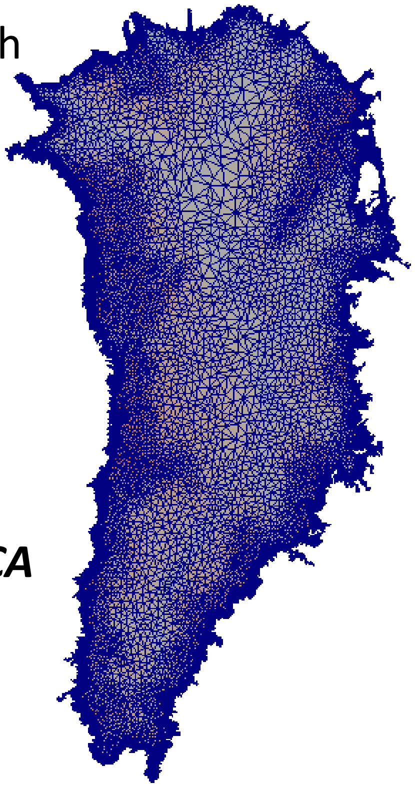
PAALS = Parallel Albany Adaptive Loop with SCOREC

offline

- **<u>Step 1</u>**: determine geometry and generate initial tetrahedral mesh (e.g., using ***Triangle*** or ***Simmetrix***).

online

- **<u>Step 2</u>**: 2D slice of initial mesh ***adaptively refined in-memory*** via ***PAALS*** based on gradient of velocity in ***Albany.***
- **<u>Step 3</u>**: 3D mesh obtained by extruding 2D mesh vertically as ***prisms*** or ***tetrahedra*** in ***Albany.***



Discussion:

- PAALS adaptive loop driven by ***homotopy continuation*** using ***LOCA*** package in ***Trilinos***.

Mesh Generation/Adaptation Algorithm

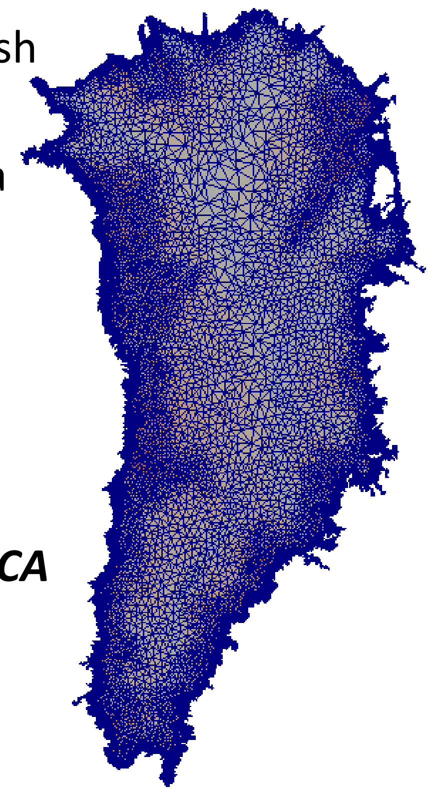
PAALS = Parallel Albany Adaptive Loop with SCOREC

offline

- **<u>Step 1</u>**: determine geometry and generate initial tetrahedral mesh (e.g., using ***Triangle*** or ***Simmetrix***).

online

- **<u>Step 2</u>**: 2D slice of initial mesh ***adaptively refined in-memory*** via ***PAALS*** based on gradient of velocity in ***Albany.***
- **<u>Step 3</u>**: 3D mesh obtained by extruding 2D mesh vertically as ***prisms*** or ***tetrahedra*** in ***Albany.***



Discussion:

- PAALS adaptive loop driven by ***homotopy continuation*** using ***LOCA*** package in ***Trilinos***.
- Currently adaptation is done in ***stand-alone*** *Albany*.

PUMI Parallel Unstructured Mesh Infrastructure

Albany

Rensselaer

Sandia National Laboratories

Mesh Generation/Adaptation Algorithm

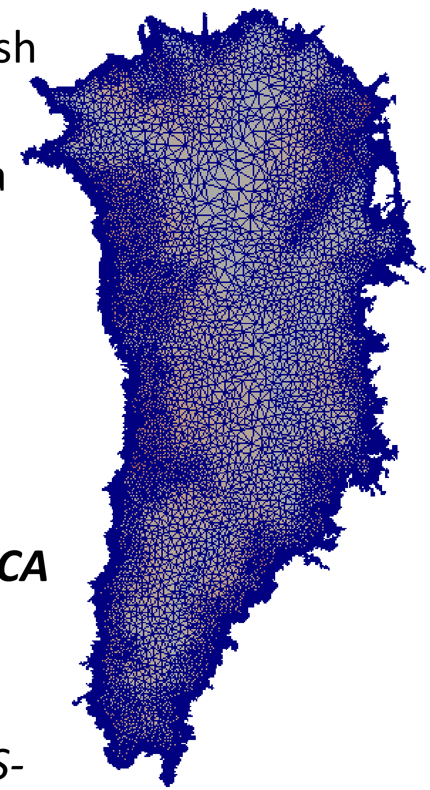
PAALS = Parallel Albany Adaptive Loop with SCOREC

offline

- **Step 1**: determine geometry and generate initial tetrahedral mesh (e.g., using ***Triangle*** or ***Simmetrix***).

online

- **Step 2**: 2D slice of initial mesh ***adaptively refined in-memory*** via ***PAALS*** based on gradient of velocity in ***Albany.***
- **Step 3**: 3D mesh obtained by extruding 2D mesh vertically as ***prisms*** or ***tetrahedra*** in ***Albany.***



Discussion:

- PAALS adaptive loop driven by ***homotopy continuation*** using ***LOCA*** package in ***Trilinos***.
- Currently adaptation is done in ***stand-alone*** *Albany*.
- **Future work**: integrating adaptation into ***dynamical cores*** (*MPAS-Albany*).

PUMI Parallel Unstructured Mesh Infrastructure Albany Rensselaer Sandia National Laboratories

Outline

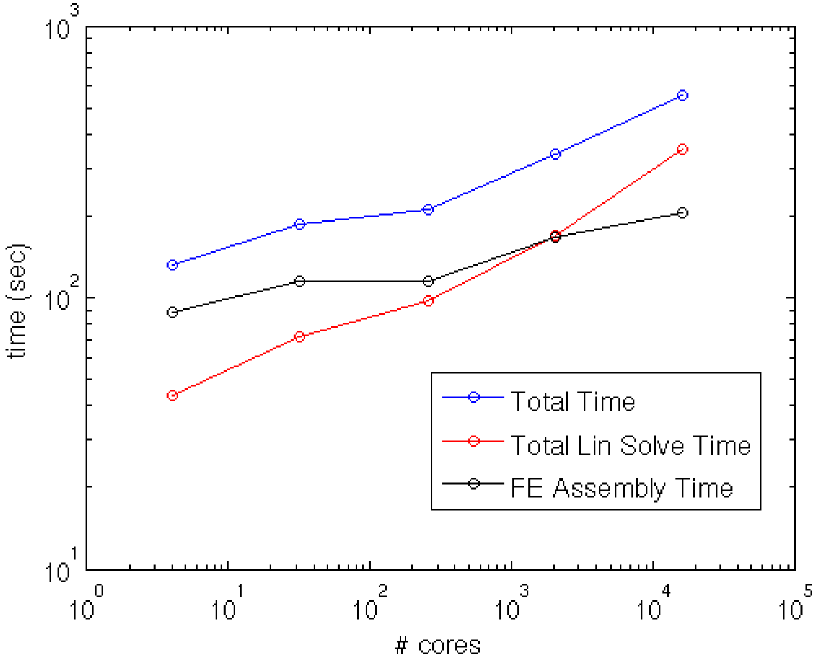
Sandia National Laboratories

1. Ice Sheet Equations and Codes.
2. Mesh Adaptivity.
3. **Scalability.**
4. Performance Portability.
5. Validation.
6. Summary and Future Work.

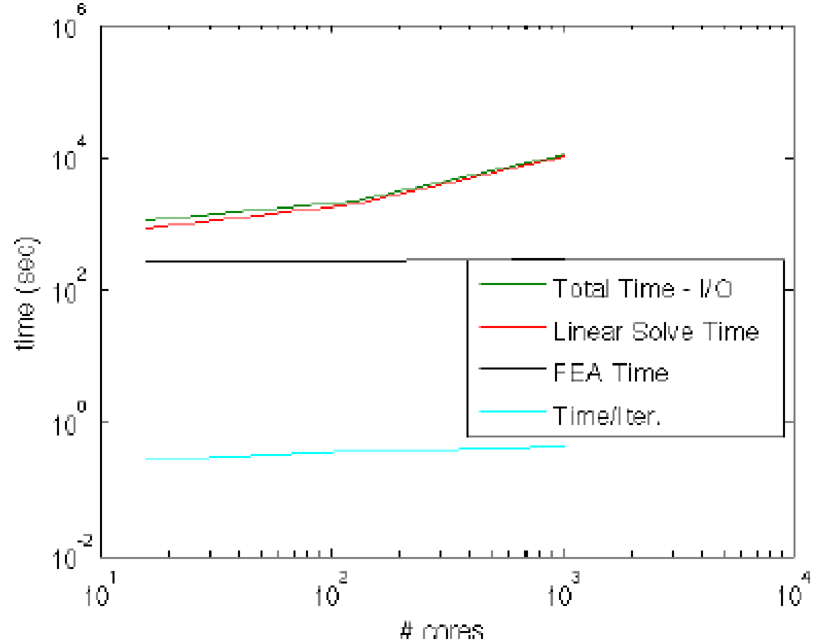
Steps towards creating a production-ready code

Initial Weak Scalability Study Using ILU

Greenland Ice Sheet

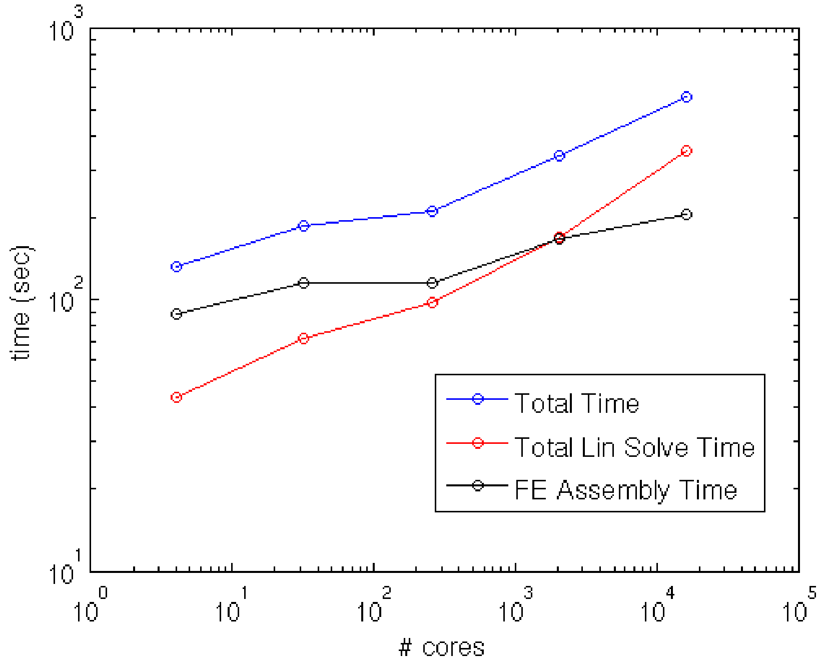


Antarctic Ice Sheet

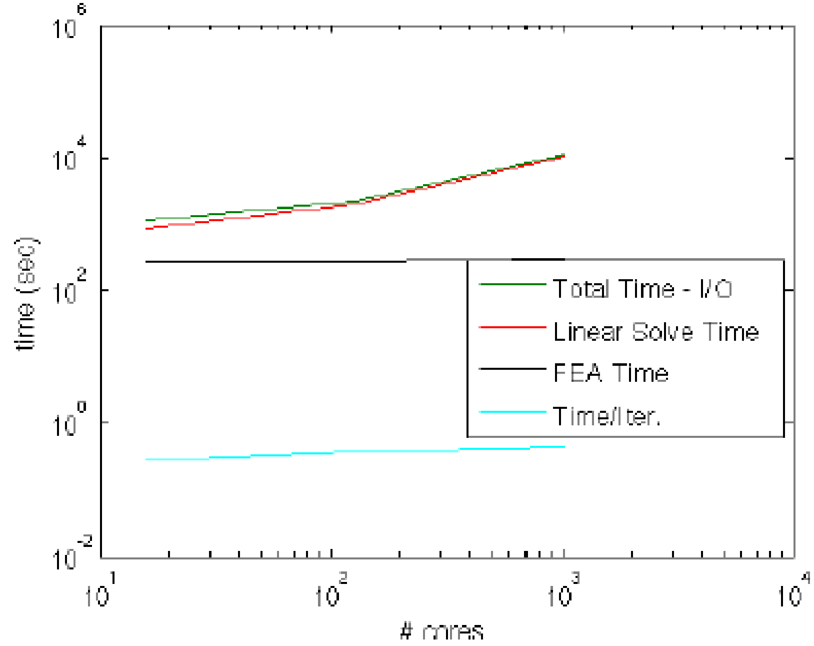


Initial Weak Scalability Study Using ILU

Greenland Ice Sheet

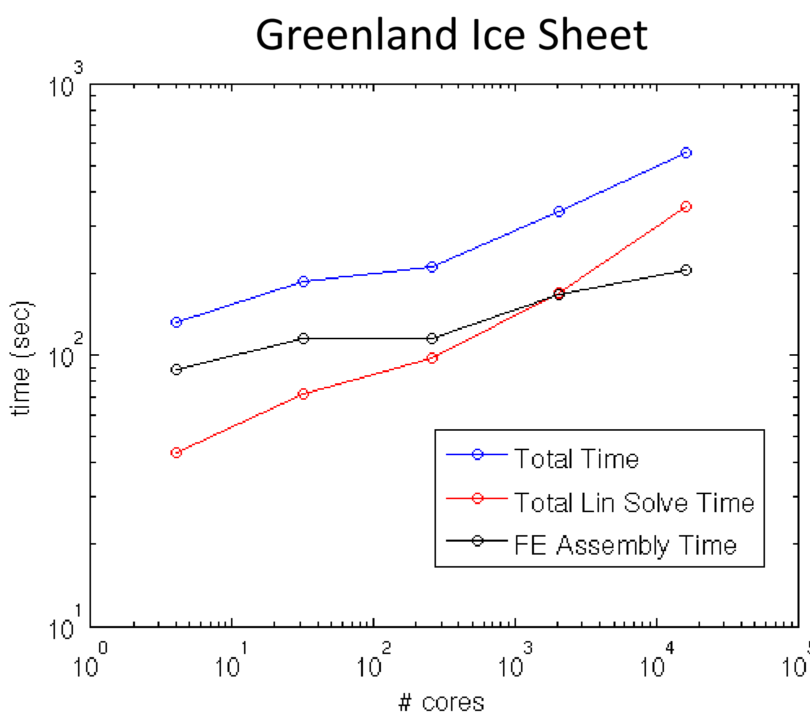


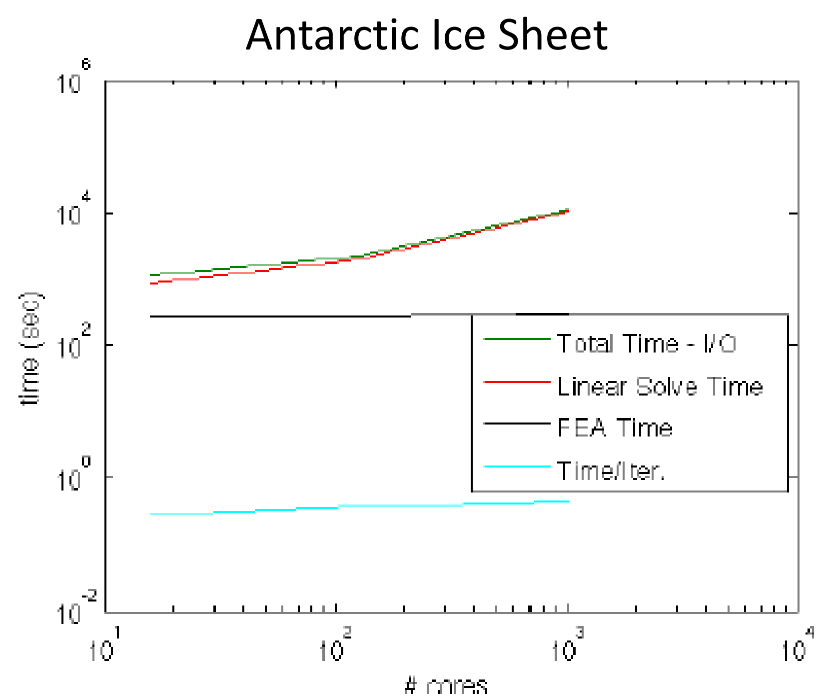
Antarctic Ice Sheet



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

Initial Weak Scalability Study Using ILU





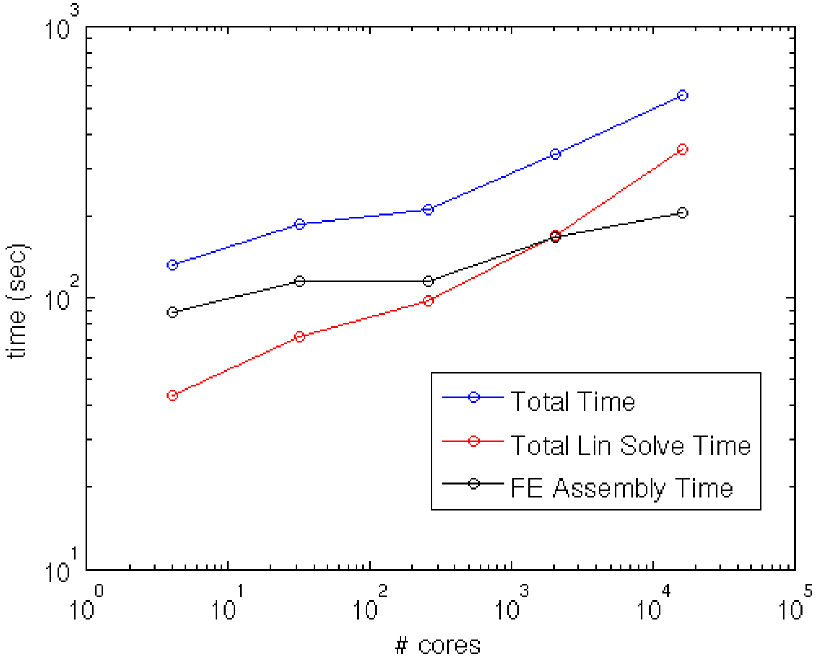
Scalability results are ***not*** acceptable!

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

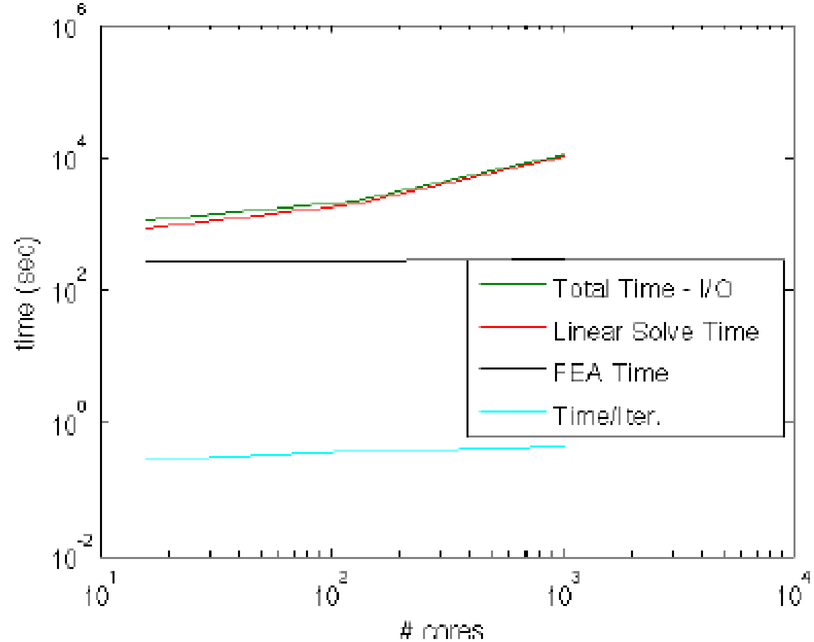
1. Ice sheet geometries have ***bad aspect ratios*** $(dx \gg dz)$.
2. ***Ice shelves*** give rise to ***severely ill-conditioned*** matrices.
3. ***Islands*** and ***hinged peninsulas*** lead to ***solver failures***.

Initial Weak Scalability Study Using ILU

Greenland Ice Sheet



Antarctic Ice Sheet



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

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

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

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

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

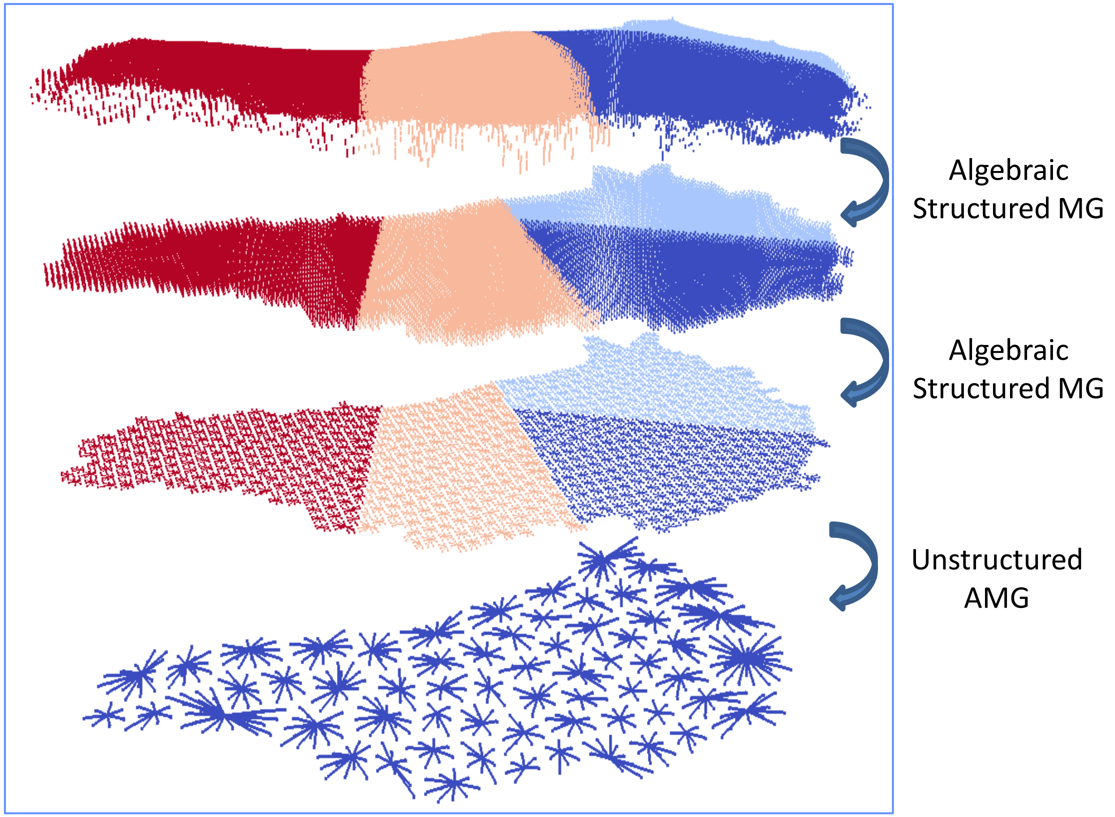
Scalability via Algebraic Multi-Grid Preconditioning with Semi-Coarsening

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

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

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

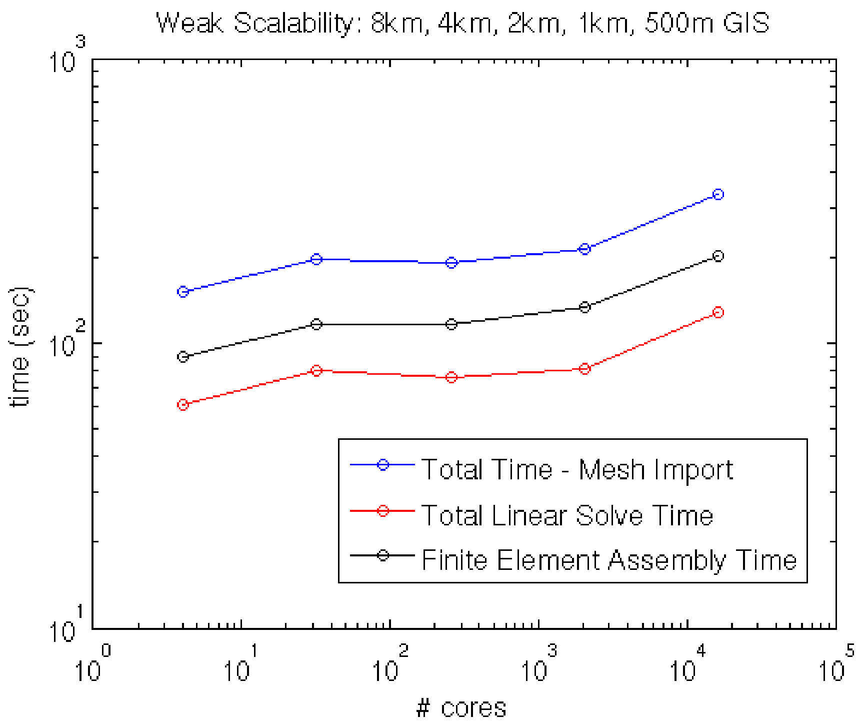
We developed a **new AMG solver** based on aggressive **semi-coarsening** (available in *ML/MueLu* packages of *Trilinos*)



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

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

Greenland Controlled Weak Scalability Study



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

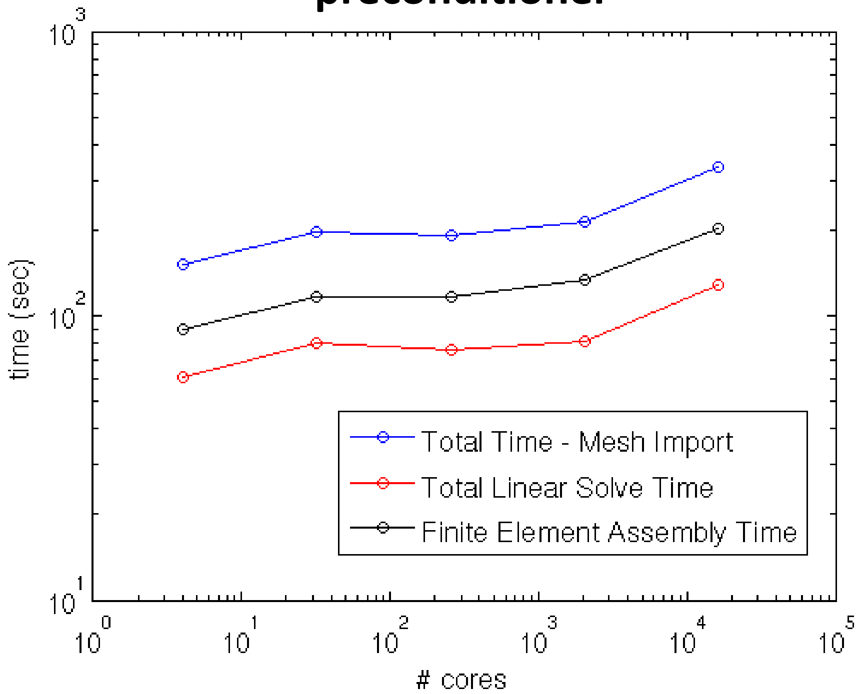
$\times 8^4$
scale up

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

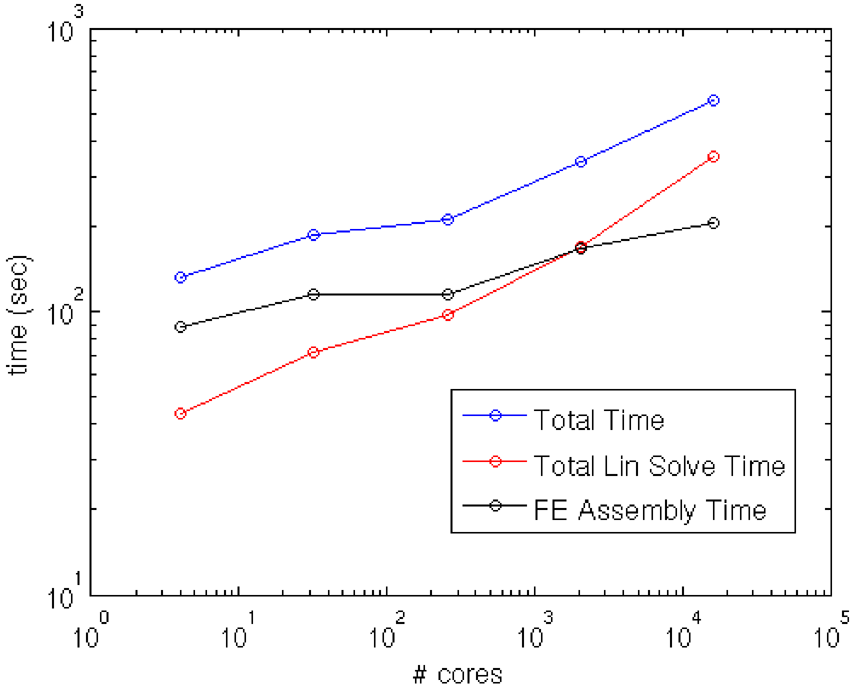
- Weak scaling study with fixed dataset, 4 mesh bisections.
- ~70-80K dofs/core.
- ***Conjugate Gradient (CG) iterative method*** for linear solves (faster convergence than GMRES).
- ***New AMG preconditioner*** developed by R. Tuminaro based on ***semi-coarsening*** (coarsening in z-direction only).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

Greenland Controlled Weak Scalability Study

New AMG preconditioner preconditioner



ILU preconditioner



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

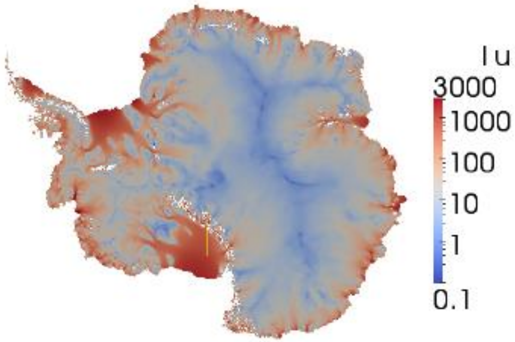
$\times 8^4$
scale up

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

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

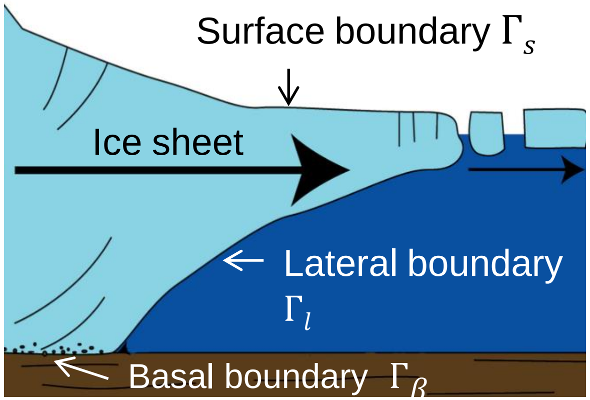
Moderate Resolution Antarctica Weak Scaling Study

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



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

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

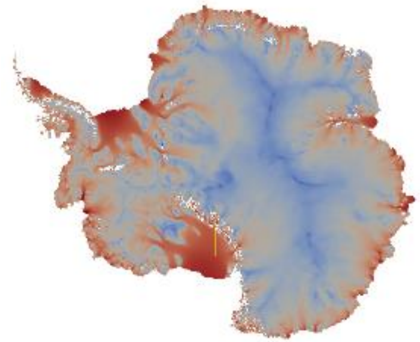


(vertical > horizontal coupling)
+
Neumann BCs
=
nearly singular submatrix associated with vertical lines

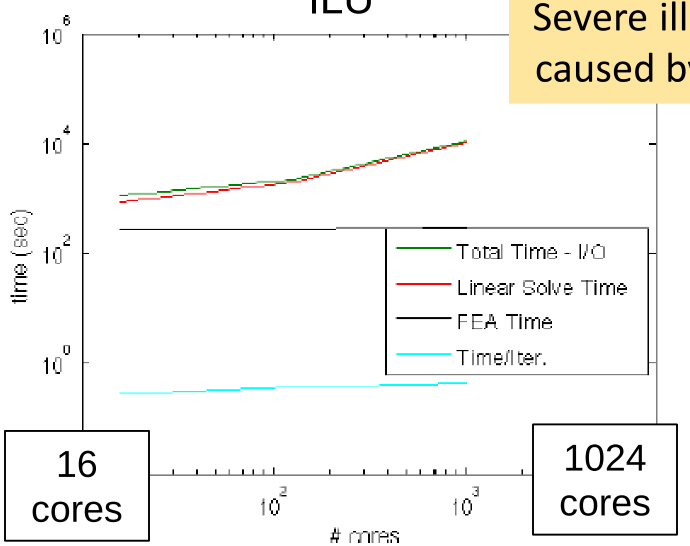
⇒ Ice shelves give rise to severe ill-conditioning of linear systems!

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

Moderate Resolution Antarctica Weak Scaling Study

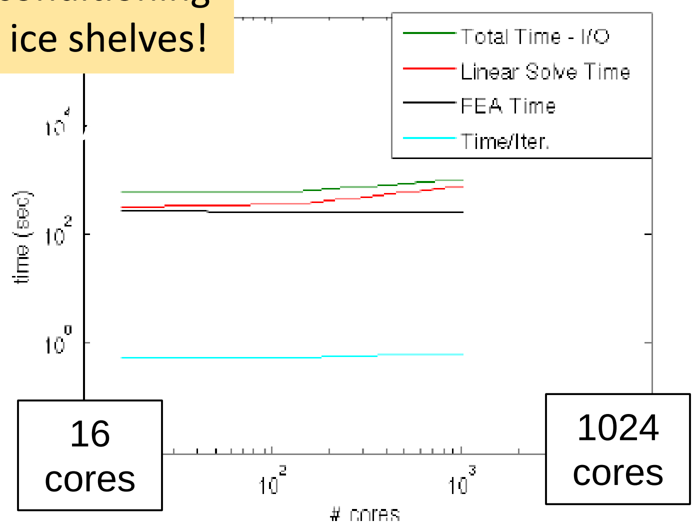


- Weak scaling study on Antarctic problem (8km w/ 5 layers → 2km w/ 20 layers).
- Initialized with realistic basal friction (from deterministic inversion) and temperature field from BEDMAP2.
- **Iterative linear solver**: GMRES.
- **Preconditioner**: ILU vs. new AMG based on aggressive semi-coarsening.

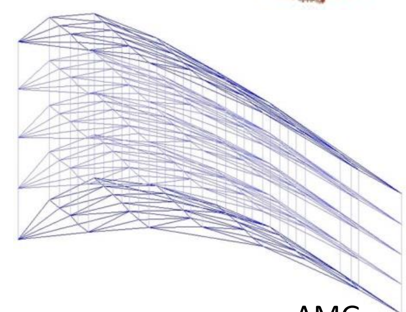


cores

Severe ill-conditioning caused by ice shelves!



cores



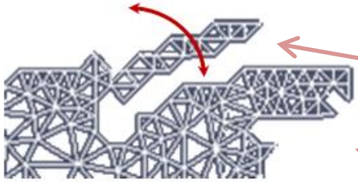
AMG preconditioner

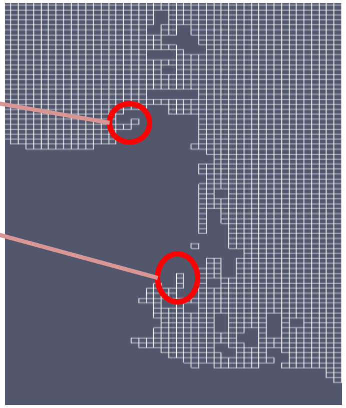
(vertical > horizontal coupling)
+
Neumann BCs
=
nearly singular submatrix associated with vertical lines

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

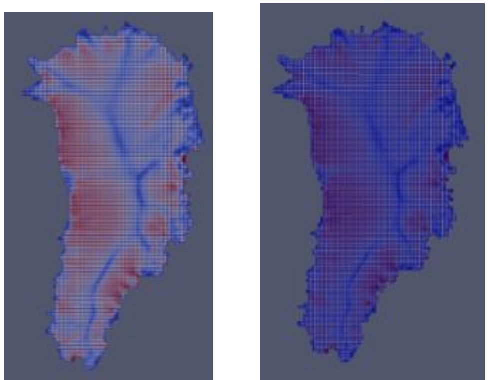
Improved Linear Solver Performance through Removal of Hinged Peninsulas

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





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



Greenland Problem

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

Outline

1. Ice Sheet Equations and Codes.

2. Mesh Adaptivity.

3. Scalability.

4. Performance Portability.

5. Validation.

6. Summary and Future Work.

Steps towards creating a production-ready code

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.) .

Performance-Portability via *Kokkos*

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

MPI (task parallelism) **+ X*** (thread + data-level parallelism)

*X = OpenMP, CUDA, etc.

Performance-Portability via *Kokkos*

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

MPI (task parallelism) **+ X*** (thread + data-level parallelism)

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

With *Kokkos*, you write an algorithm **once**, and just change a template parameter to get the optimal data layout for your hardware (e.g., (i,j,k) vs. (k,i,j).

*X = OpenMP, CUDA, etc.

Performance-Portability via *Kokkos*

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

MPI (task parallelism) **+ X*** (thread + data-level parallelism)

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

With *Kokkos*, you write an algorithm **once**, and just change a template parameter to get the optimal data layout for your hardware (e.g., (i,j,k) vs. (k,i,j).

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

*X = OpenMP, CUDA, etc.

Performance-Portability via *Kokkos*

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

MPI (task parallelism) **+ X*** (thread + data-level parallelism)

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

With *Kokkos*, you write an algorithm **once**, and just change a template parameter to get the optimal data layout for your hardware (e.g., (i,j,k) vs. (k,i,j).

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

*X = OpenMP, CUDA, etc.

Kokkos-ification of Finite Element Assembly

Sandia National Laboratories

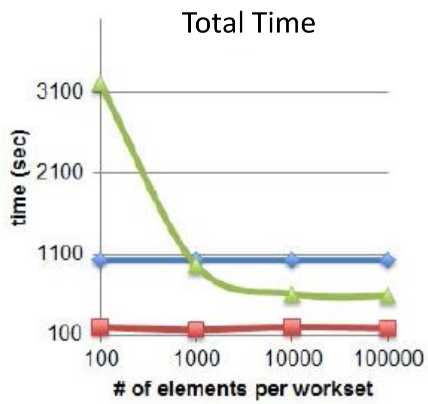
```
typedef Kokkos::OpenMP ExecutionSpace;
//typedef Kokkos::CUDA ExecutionSpace;
//typedef Kokkos::Serial ExecutionSpace;
template<typename ScalarT>
vectorGrad<ScalarT>::vectorGrad()
{
Kokkos::View<ScalarT****, ExecutionSpace> vecGrad("vecGrad", numCells, numQP, numVec, numDim);
}
**********************************************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

Albany

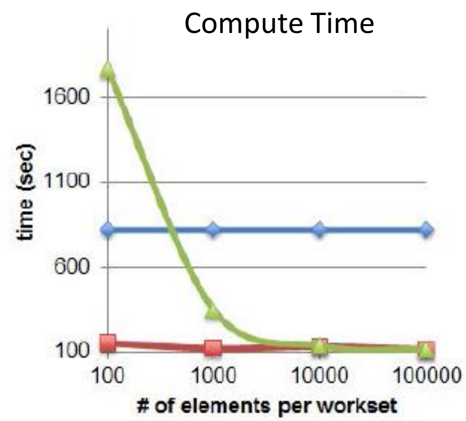
ExecutionSpace parameter tailors code for device (e.g., OpenMP, CUDA, etc.)

Results on *Shannon*: 4km Greenland & 8km Antarctica Problems

4km Greenland

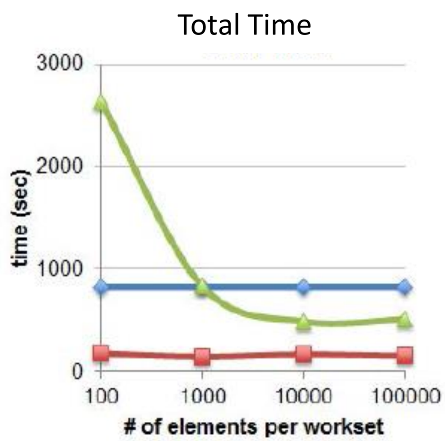


(a)

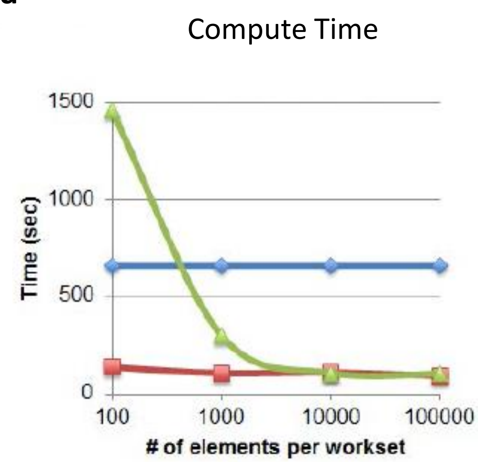


(b)

8km Antarctica



(c)



(d)

Shannon: *32 nodes*

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

— Serial: 1 MPI thread/node
— OpenMP: 16 OpenMP threads/node
— CUDA: 1 GPU/node

Max speedup over Serial for workset size > 1000

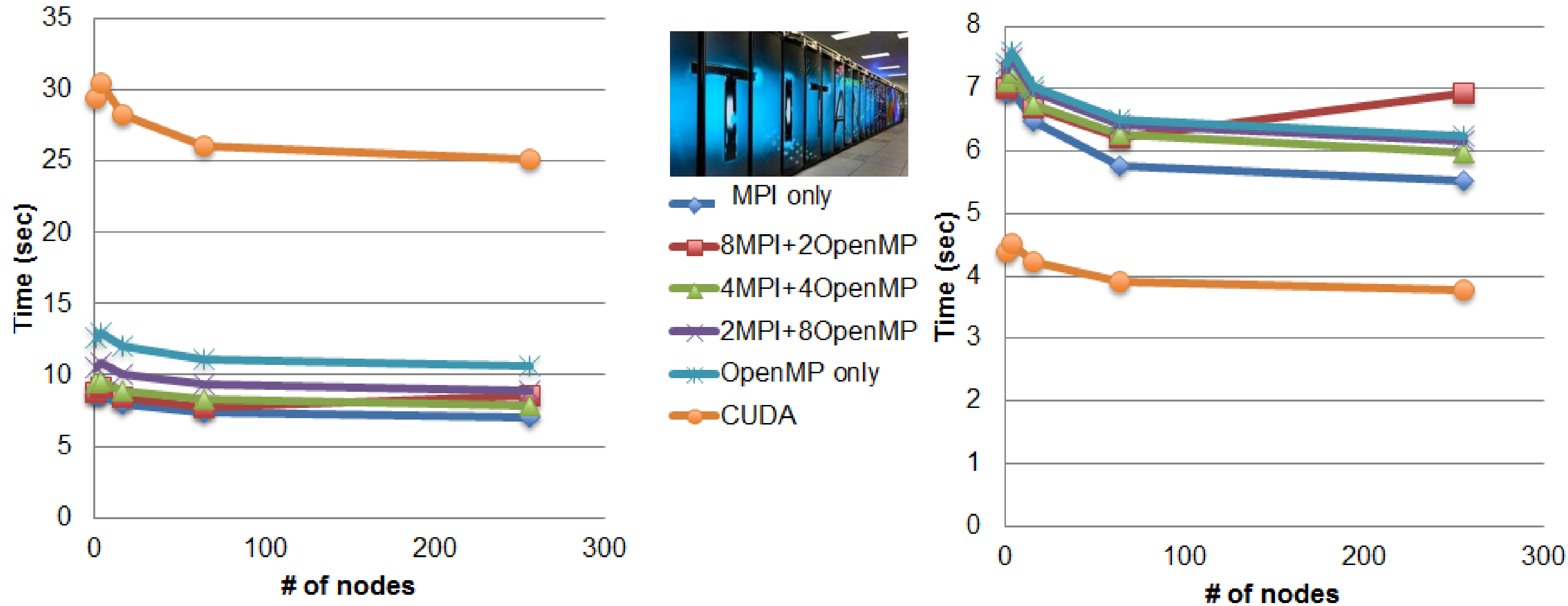
	OpenMP	CUDA
Total Time	5.6x	1.7x
Compute Time	7.2x	6.7x

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

Results on *Titan*: Weak Scalability for Greenland (8km, 4km, 2km, 1km)

Total Time

Compute Time (Total Time – Gather/Scatter)



- ***CUDA*** implementation is ***slower*** than MPI-only for total time but ***faster*** for compute time (due to communication costs).
- Filling the matrix requires use of ***atomics***, which are difficult to optimize (work in progress by *Kokkos* team).
- Work is ***distributed more thinly*** on *Titan* than on *Shannon* → worse for GPU (less work, more communications).

Titan: 18,688 AMD Opteron nodes

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

Outline

1. Ice Sheet Equations and Codes.

2. Mesh Adaptivity.

3. Scalability.

4. Performance Portability.

5. Validation.

6. Summary and Future Work.

Steps towards creating a production-ready code

Validation: Definition & Workflow

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

*vs. verification: is our code bug free? (some of my past SIAM talks on PISCEES)

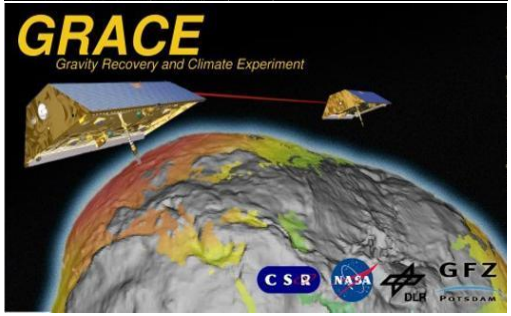
Validation: Definition & Workflow

Sandia National Laboratories

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

- There are currently (up to) 2 decades of large-scale ***satellite*** observations of Greenland ice sheet geometry change:

ICESat1	2003 – 2009
GRACE	2002 – 201? (ongoing)



*vs. verification: is our code bug free? (some of my past SIAM talks on PISCEES)

Validation: Definition & Workflow

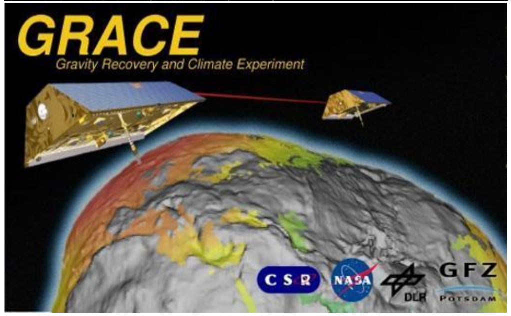
Validation*: how well does our model represent the real ice sheet?

- There are currently (up to) 2 decades of large-scale ***satellite*** observations of Greenland ice sheet geometry change:

ICESat1	2003 – 2009
GRACE	2002 – 201? (ongoing)

- Future missions will ***extend*** these observational time series:

ICESat1	2017-20??
GRACE "follow-on"	2017-20??
GRACE2	2020s-?



*vs. verification: is our code bug free? (some of my past SIAM talks on PISCEES)

Validation: Definition & Workflow

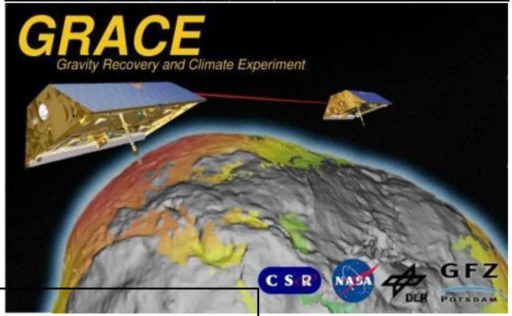
Validation*: how well does our model represent the real ice sheet?

- There are currently (up to) 2 decades of large-scale ***satellite*** observations of Greenland ice sheet geometry change:

ICESat1	2003 – 2009
GRACE	2002 – 201? (ongoing)

- Future missions will ***extend*** these observational time series:

ICESat1	2017-20??
GRACE "follow-on"	2017-20??
GRACE2	2020s-?



Validation Workflow:

- Run ice sheet model over period where ICESat/GRACE observations exist.
- Process model output for comparison to these observations.
- Process observations for comparison to model output.
- Evaluate model performance relative to observations.
  - ***ICESat***: ice sheet surface elevation [state comparison]
  - ***GRACE***: rate of mass change [trend comparison]

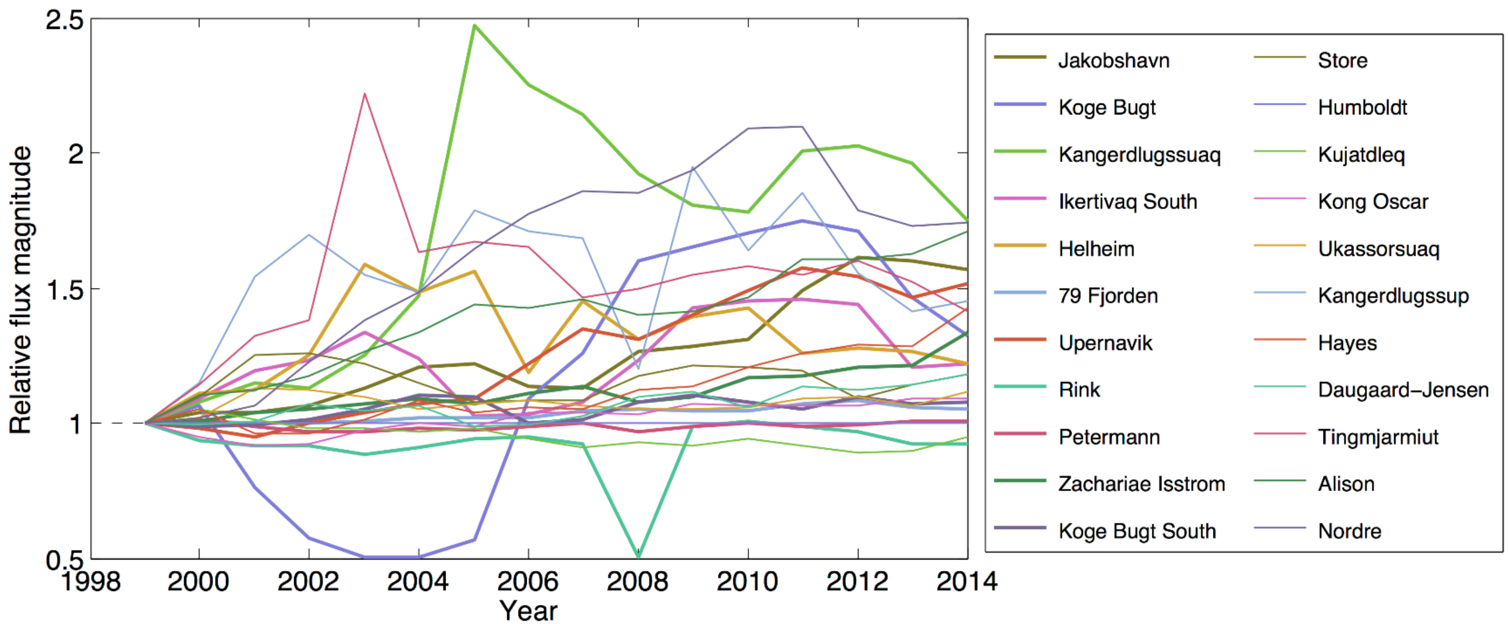
*vs. verification: is our code bug free? (some of my past SIAM talks on PISCEES)

Validation: Observations & Forcing

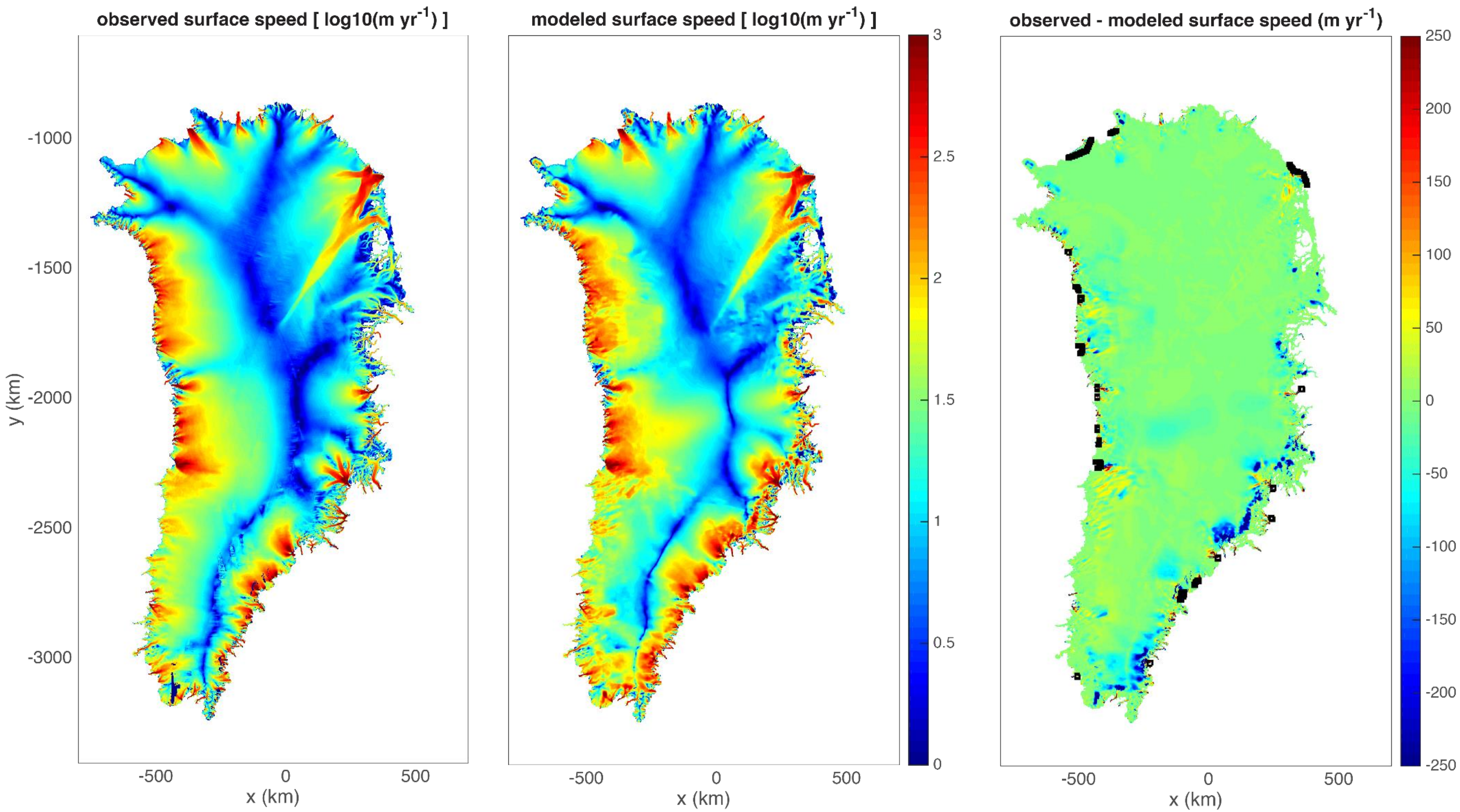
- **Validation time period**:
  - ICESat: 2003-2009
  - GRACE: 2003-2011 (CSR Release-05)

Code: *CISM-Albany* (*CISM2.0* + *Albany/FELIX*)

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

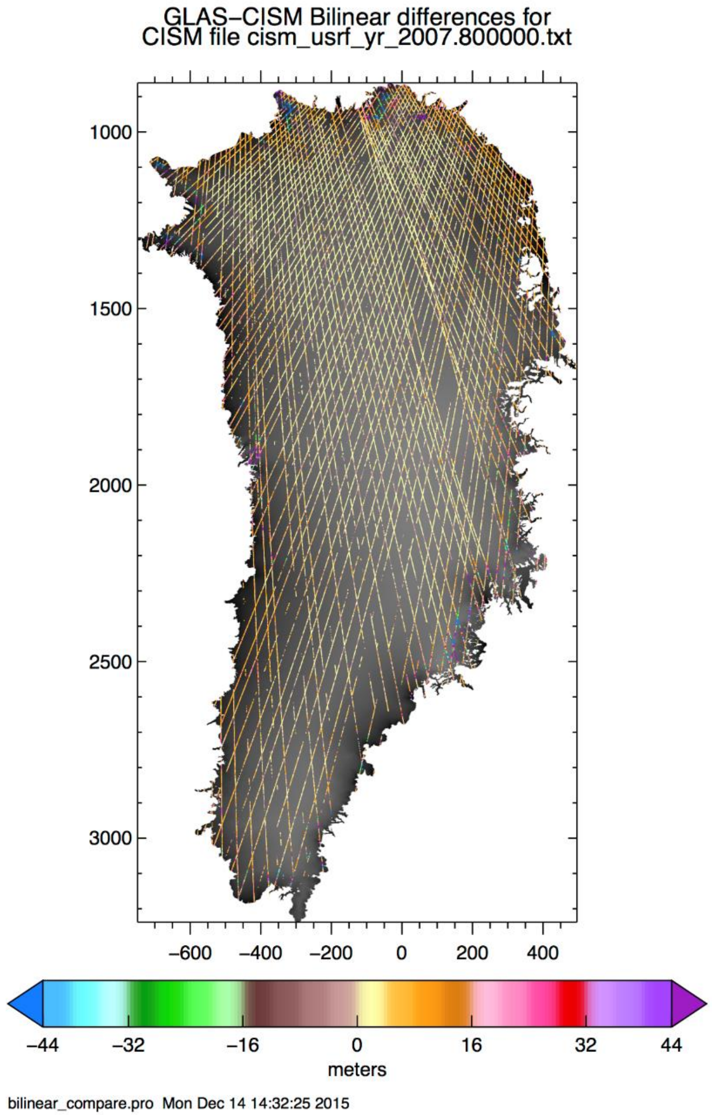


Validation: 1km GIS initial condition*

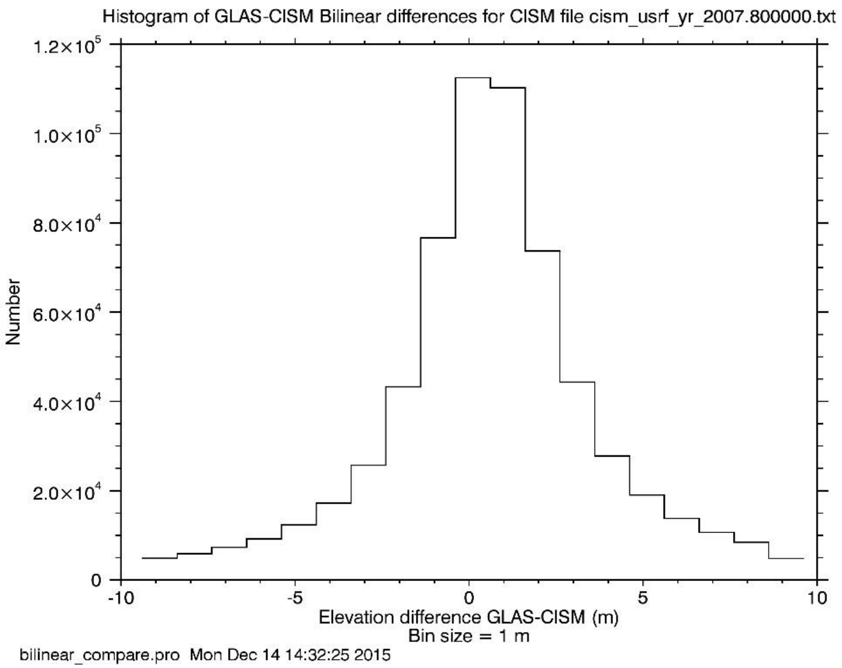


*Initial condition obtained through ***deterministic inversion***; see talk by M. Perego.

Validation: Surface Elevation Observations

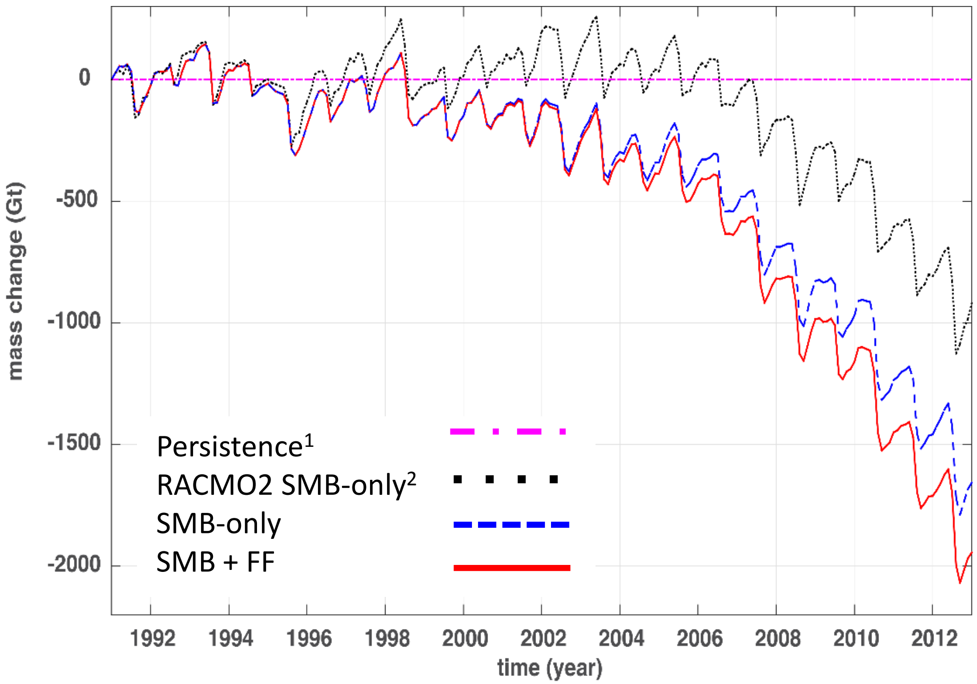


- **<u>Model evaluated</u>:** ***CISM-Albany*** with ***SMB + FF*** in October 2007.
- Surface elevation predictions (states) agree pretty well with ***GLAS*** (***Geoscience Laser Altimeter System aboard ICESat***): ***mean differences*** are ***<1 m***

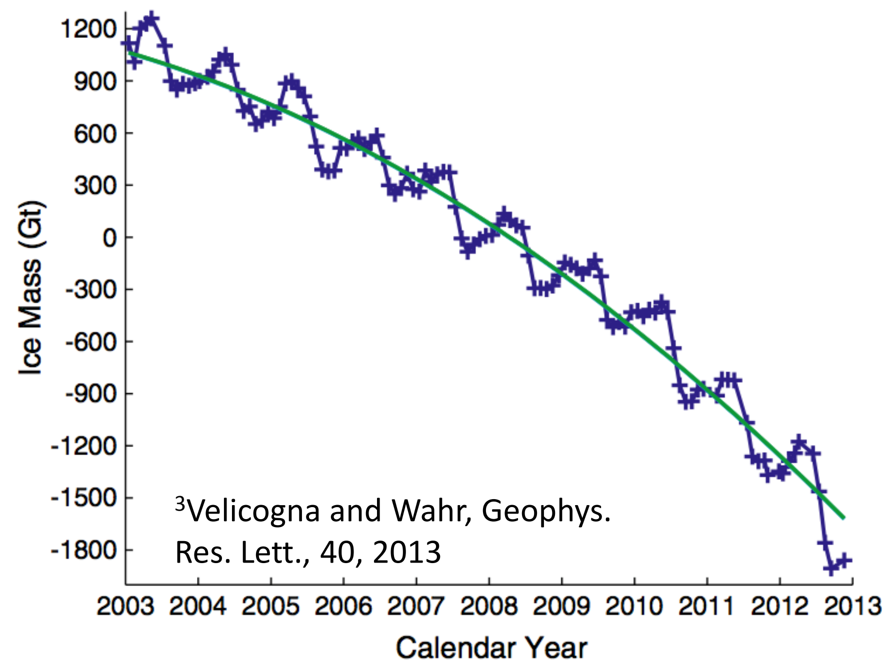


Validation: Whole Ice Sheet Mass Trends

Ice Sheet Model Predictions



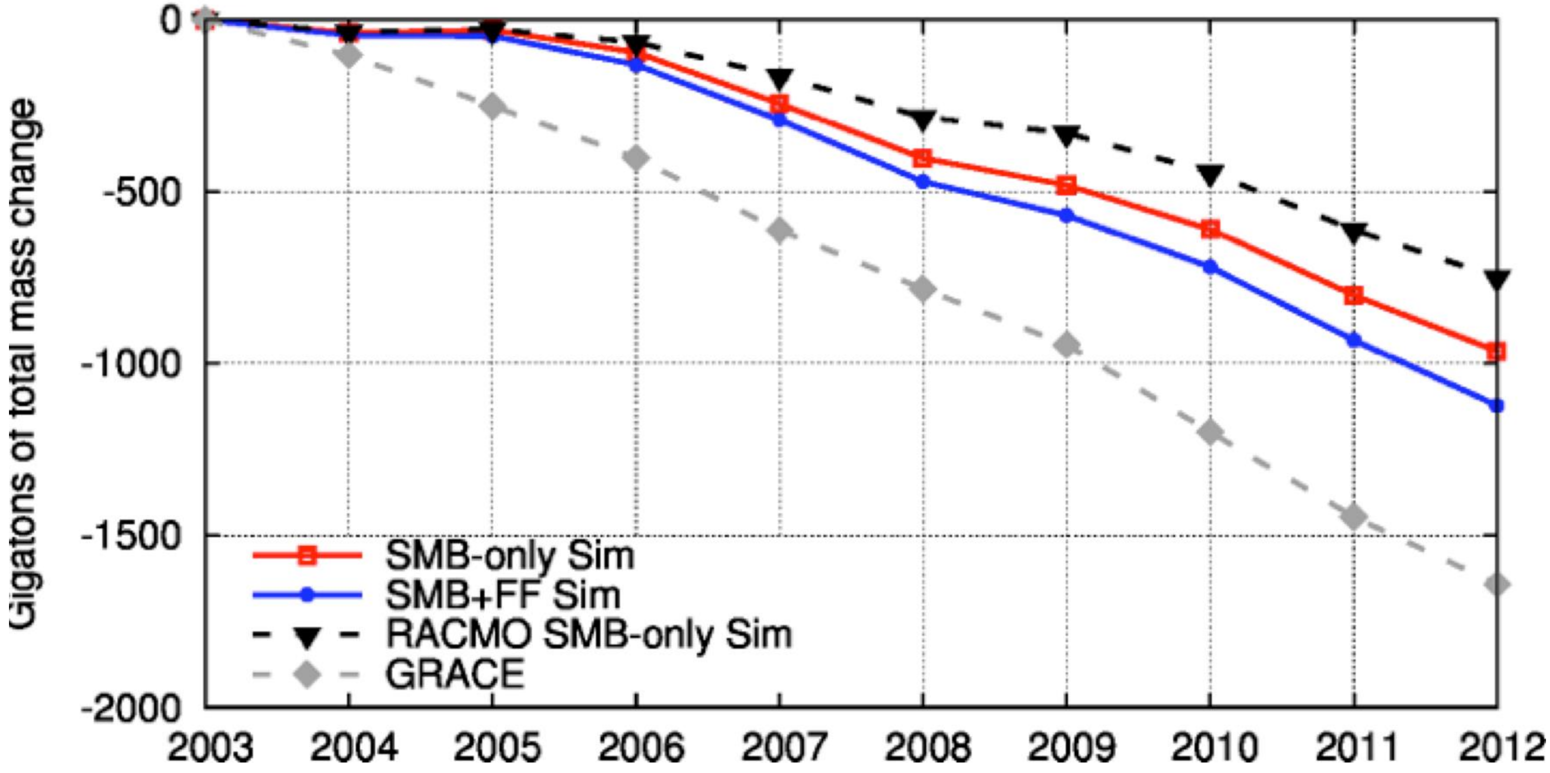
GRACE Observations at Sub-Annual Scale[3]



- Overall mass trends from the ***CISM-Albany*** simulations look ***fairly realistic*** (left figure, red & blue)
- ***CISM-Albany*** trends look ***more realistic*** (closer to observations; right figure) than the "idealized" simulations (left figure, pink & black).
- There is ***more mass loss*** from the simulation when we account for changes in outlet glacier flux (i.e., the evolution of the ice sheet is not only forced by surface mass balance changes, but also be changes in outlet glacier dynamics).

[1]Geometry held constant in time. [2]RACMO2 SMB time series applied to dh/dt; includes SS "discharge" using 1960-1990 mean SMB.

Validation: Whole Ice Sheet Mass Trends



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

Validation Takeaways

Current generation ice sheet models, ***when appropriately forced***, show ***skill*** at mimicking ice sheet observations

Validation Takeaways

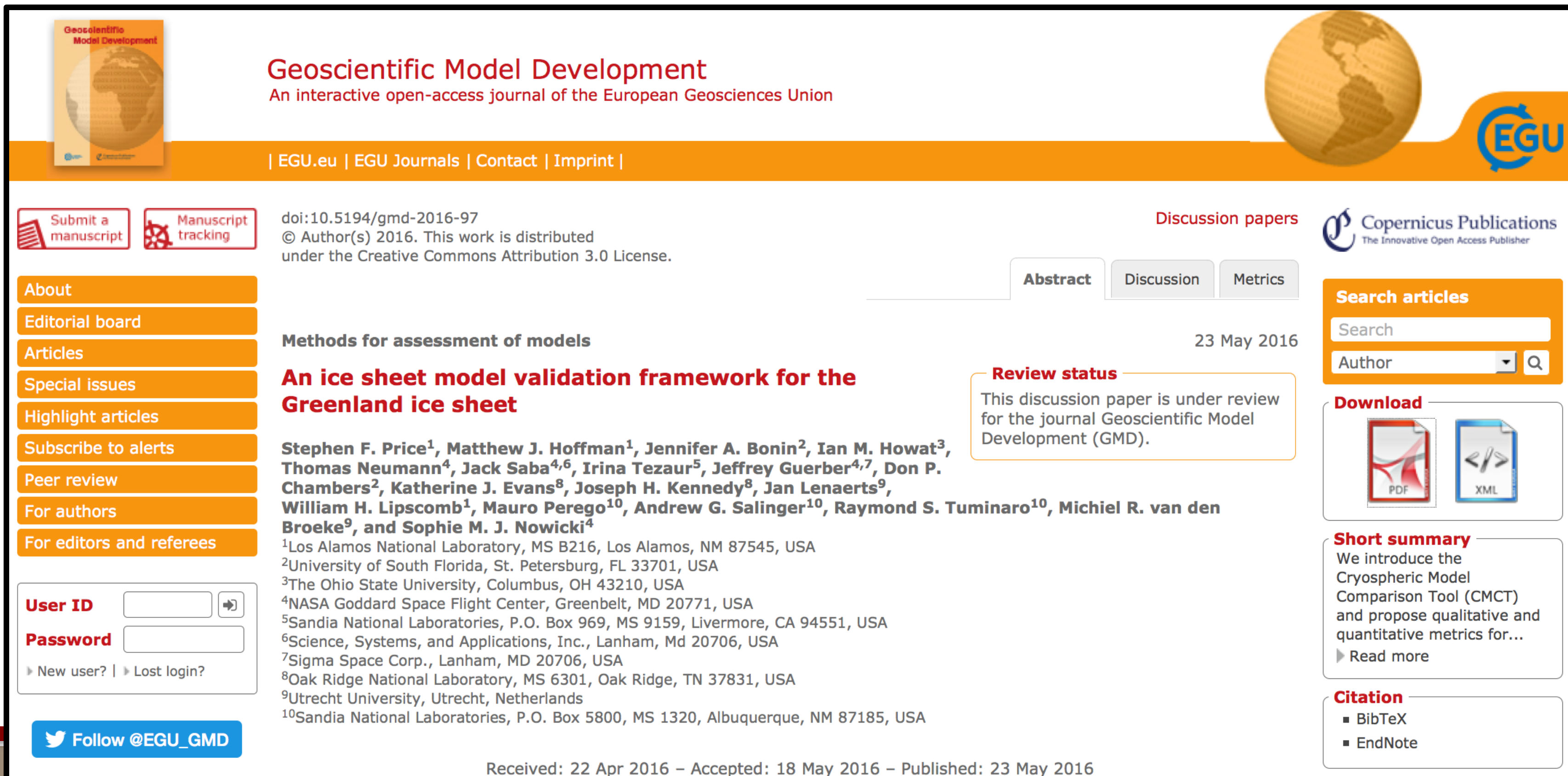
Current generation ice sheet models, ***<u>when appropriately forced</u>***, show ***skill*** at mimicking ice sheet observations

- ***Clear improvement over a decade ago:*** SLR projections from ice sheet models were not included in the IPCC's AR4 b/c models could not explain observed ice dynamical behaviors.

Validation Takeaways

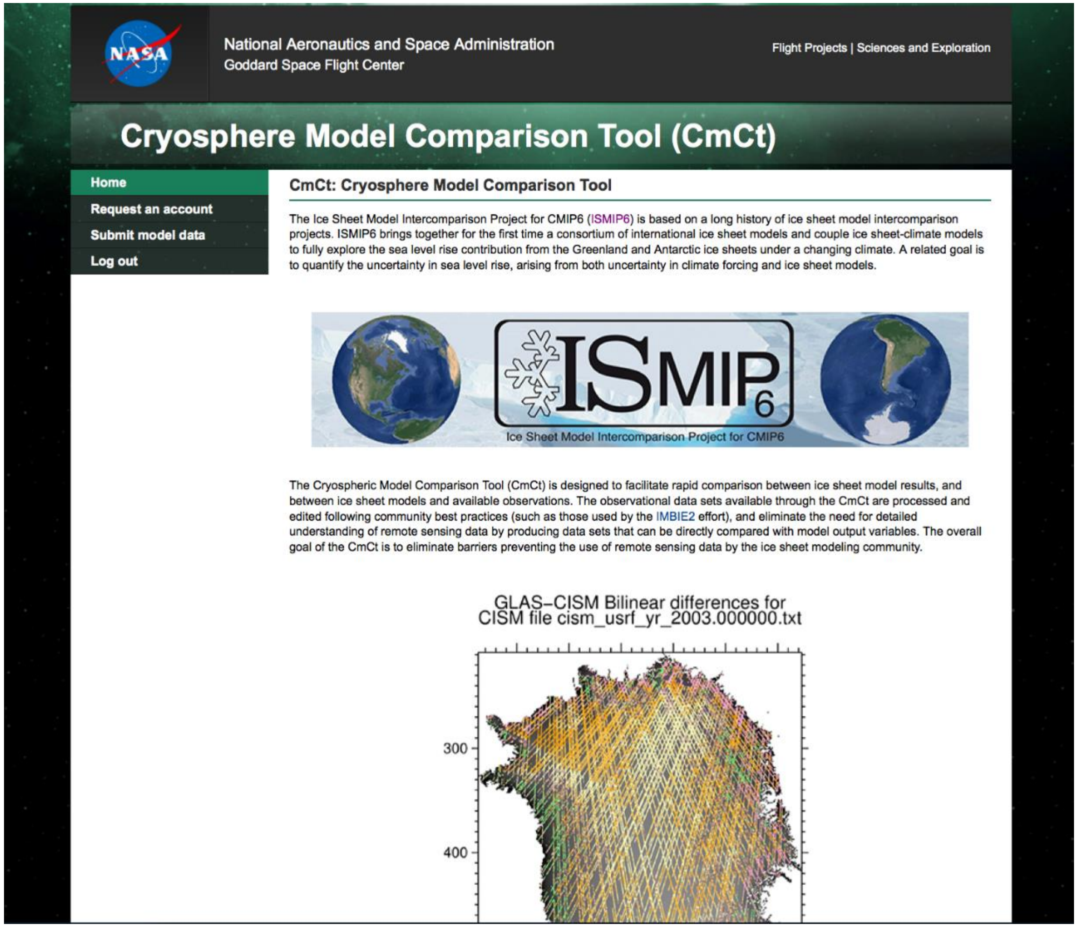
Current generation ice sheet models, ***when appropriately forced***, show ***skill*** at mimicking ice sheet observations

- ***Clear improvement over a decade ago:*** SLR projections from ice sheet models were not included in the IPCC's AR4 b/c models could not explain observed ice dynamical behaviors.
- For more details on validation study, see our ***GMD paper***, currently under review.



Cryospheric Model Comparison Tool (CmCt)

- Validation work involved development of new ice sheet model validation framework – ***the Cryospheric Model Comparison Tool (CmCt)*** – which includes proposed qualitative and quantitative metrics for use in comparing models to observations.



CmCt is ***online*** and ready for testing by community

https://ggsghpcc.sgt-inc.com/cmct/

Outline

1. Ice Sheet Equations and Codes.

2. Mesh Adaptivity.

3. Scalability.

4. Performance Portability.

5. Validation.

6. Summary and Future Work.

Steps towards creating a production-ready code

Summary and Future Work

Sandia National Laboratories

Summary:

- We have developed a land-ice solver known as ***Albany/FELIX*** using ***Trilinos*** libraries.
- This solver is:
  - Equipped with in-memory parallel ***unstructured mesh adaption***.
  - ***Scalable***, ***fast***, ***robust***.
  - Coupled to *CISM* and *MPAS* codes for ***dynamic*** runs and integration into ESMs.
  - ***Verified*** and ***validated***.
  - ***Portable*** to new and emerging architecture machines.
  - Equipped with ***advanced analysis capabilities*** (deterministic inversion, UQ).

Ongoing/future work:

- ***Mesh adaptivity*** for ***transient*** runs.
- Integration of ***hinge removal algorithm*** into ***transient*** runs.
- ***Science runs*** using *CISM-Albany* and *MPAS-Albany*.
- Code optimizations for ***new architecture machines*** (GPUs, multi-core, many-core).
- ***Uncertainty quantification*** (not covered in this talk).
- Delivering code to climate community and ***coupling to ESMs***.

Thank you! Questions?

Video acknowledgement: B. Carvey (SNL)

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.*

References

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

[2] A. Salinger, *et al.* "Albany: Using Agile Components to Develop a Flexible, Generic Multiphysics Analysis Code", accepted for publication in *Int. J. Multiscale Comput. Engng*.

[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 press).

[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", submitted to *Geoscientific Model Development (GMD).*

References (cont'd)

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

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

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

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

[11] J.H. van Angelen, M.R. Van Den Broeke, B. Wouters, J.T.M. Lenaerts, "Contemporary (1960–2012) Evolution of the Climate and Surface Mass Balance of the Greenland Ice Sheet", *Surveys in Geophysics*, 35, 1155–1174, 2013.

[12] E.M. Enderlin, I.M. Howat, S. Jeong, M.J. Noh, J.H. Angelen, J. H., M.R. Broeke, "An improved mass budget for the Greenland ice sheet", *Geophysical Research Letters*, 41, 866–872, 2014.

[13] I. Velicogna, J. Wahr, "Time-variable gravity observations of ice sheet mass balance: Precision and limitations of the GRACE satelliteData", Geophysical Research Letters, 2013.

PISCEES Project for Land-Ice Modeling

PISCEES*
SciDaC Application Partnership
(DOE's BER + ASCR divisions)
2012-2017

3 land-ice dycores developed under PISCEES

FSU FELIX
FSU
Finite Element
Full Stokes Model

Albany/FELIX
SNL
Finite Element
"Higher-Order" Stokes Model

BISICLES
LBNL
Finite Volume
L1L2 Model

Increased fidelity

Goal: support DOE climate missions (sea-level rise predictions)

*Predicting Ice Sheet Climate & Evolution at Extreme Scales.

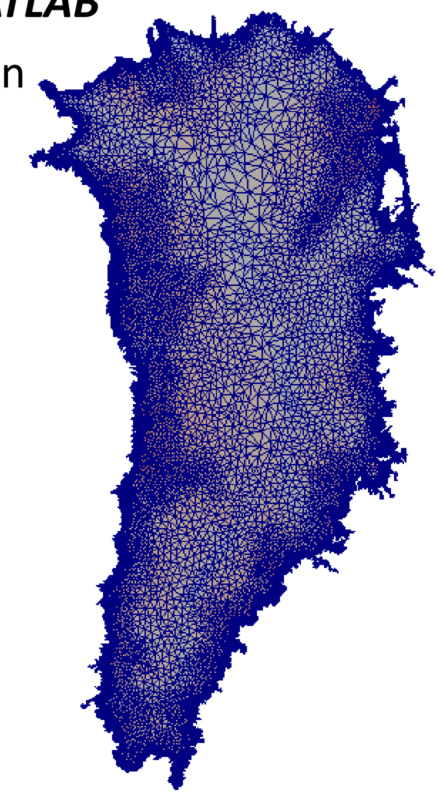
- ***Multi-lab/multi-university*** project involving mathematicians, climate scientists, and computer scientists.
- Leverages software/expertise from ***SciDAC Institutes*** (FASTMath, QUEST, SUPER) and hardware from ***DOE Leadership Class Facilities***.
- ***Sandia staff:*** Irina Tezaur, Andy Salinger, Mauro Perego, Ray Tuminaro, John Jakeman, Mike Eldred; ***PI:*** Steve Price (LANL)

Unstructured Mesh Generation & Adaptivity

Original approach:

offline
- **<u>Step 1</u>**: geometry boundary and possible holes determined in ***MATLAB***
- **<u>Step 2</u>**: uniform triangular mesh is generated and refined based on ***surface velocity gradient*** in ***Triangle*** (2D meshing software)

online
- **<u>Step 3</u>**: 3D mesh obtained by extruding 2D mesh vertically as ***prisms***, then splitting each prism into 3 ***tetrahedra (Albany).***



New approach:

Parallel Albany Adaptive Loop with SCOREC* (PAALS)

offline
- **<u>Step 1</u>**: geometry determined from *.nc file using ***scripts***.
- **<u>Step 2</u>**: uniform tetrahedral mesh generated (using ***Triangle*** or ***Simmetrix***).

online
- **<u>Step 3</u>**: 2D slice of initial mesh adaptively refined via ***in-memory Parallel Albany Adaptive Loop with SCOREC*** (***PAALS***).
- **<u>Step 4</u>**: 3D mesh obtained by extruding 2D mesh vertically as ***prisms*** in ***Albany.***

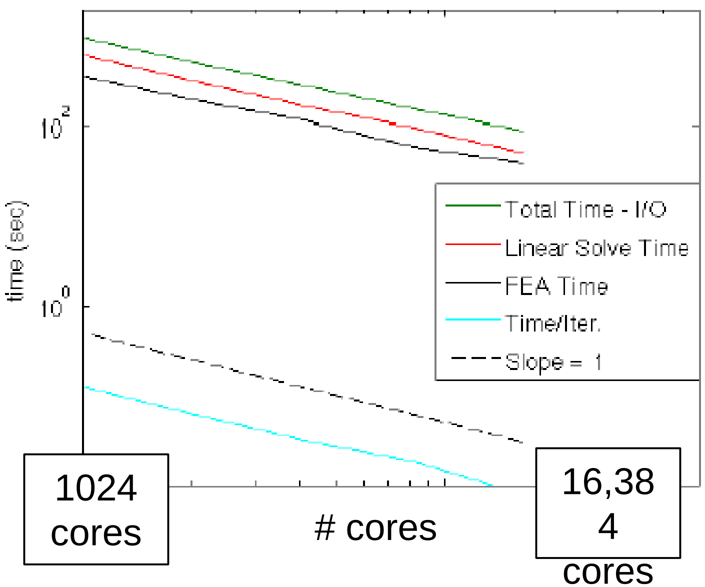
PUMi Parallel Unstructured Mesh Infrastructure · Albany · Rensselaer · Sandia National Laboratories

*Scientific Computation Research Center at Rensselaer Polytechnic Institute (RPI); https://github.com/SCOREC

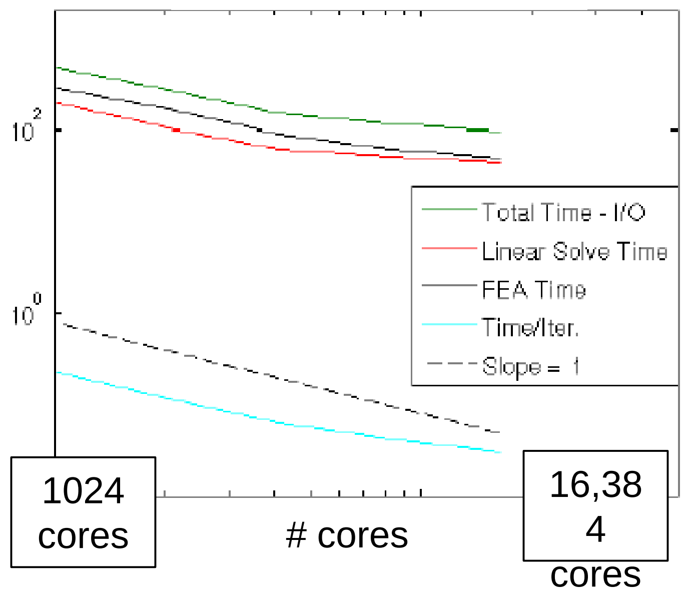
Fine-Resolution Greenland Strong Scaling Study

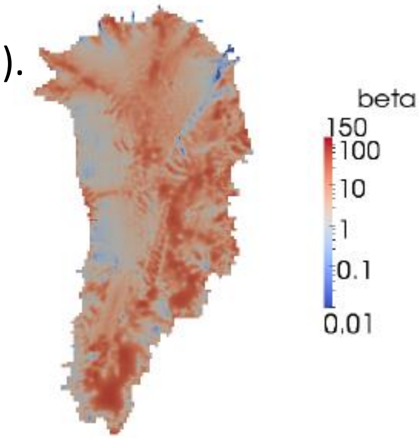
- Strong scaling on 1km Greenland with 40 vertical layers (143M dofs, hex elements).
- Initialized with realistic basal friction (from deterministic inversion) and temperature fields → interpolated from coarser to fine mesh.
- **Iterative linear solver**: CG.
- **Preconditioner**: ILU vs. new AMG (based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

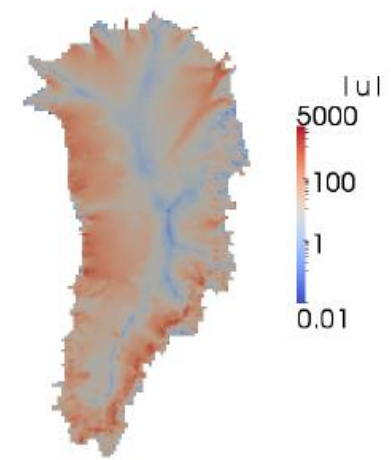
ILU



AMG







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

Validation - Total Mass Change 2003-2012