



Exceptional service in the national interest

OVERVIEW OF TRILINOS DISCRETIZATION AND ANALYSIS CAPABILITIES

Mauro Perego

Sandia National Laboratories

HPSF conference 2026,
March 16-20, Chicago, IL



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

SANDXXX

DISCRETIZATION AND ANALYSIS CAPABILITIES



- Utility tools for [Automatic differentiation](#) and or [graph-based field dependency manager](#)
- [Mesh tools](#) (generation, storage, adaptation, level-sets)
- [Discretizations](#) (mostly finite elements, but also finite volume and mesh-free methods)
- [Time integrators](#) (adaptive time-stepping schemes)
- [Analysis tools](#) (sensitivities, numerical optimization, uncertainty quantifications)

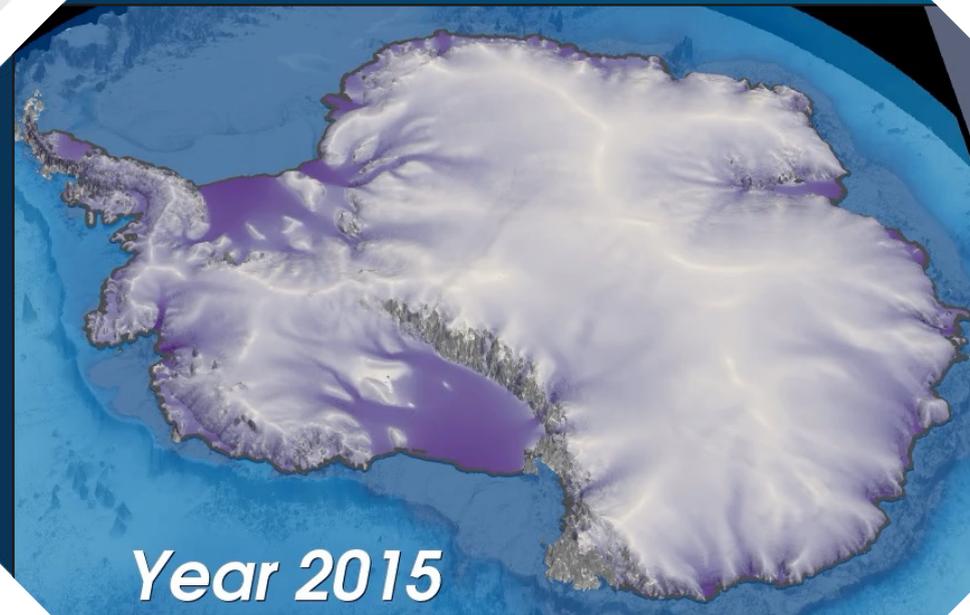


DISCRETIZATION AND ANALYSIS PACKAGES



Utilities		
Automatic differentiation	Sacado	Eric Phipps <etphipp@sandia.gov>
Graph-based field dependency manager	Phalanx	Roger Pawlowski <rppawlo@sandia.gov>
Mesh tools		
Mesh data structure	STK	Alan Williams <william@sandia.gov>
Level set tools	Krino	David Noble <drnoble@sandia.gov>
Spatial and temporal discretizations		
Particle-based discretization and data transfer	Compadre	Paul Kuberry <pakuber@sandia.gov>
Local finite element discretization	Intrepid2	Perego Mauro <mperego@sandia.gov> Nathan Roberts <nvrober@sandia.gov>
Time integration	Tempus	Curt Ober <ccober@sandia.gov>
Finite element library	Panzer	Roger Pawlowski <rppawlo@sandia.gov>
Analysis tools		
Uncertainty Quantification	Stokhos	Eric Phipps <etphipp@sandia.gov>
Constrained Optimization	ROL	Denis Ridzal <dridzal@sandia.gov>

TRILINOS SHOWCASE VIA ICE-SHEET MODELING



Ice-sheet modeling team:

L. Bertagna, M. Carlson, K. Liegeois, J. Hu,
M. Perego, J. Watkins, A. Salinger,
I. Tezaur, R. Tuminaro - [Sandia](#)
T. Hillebrand, M. Hoffman, S. Price - [LANL](#)

SOFTWARE: MPAS-ALBANY LAND ICE MODEL (MALI)



CAPABILITY	SOFTWARE TOOLS
Finite Elements ice-sheet solver	Albany
Constrained Optimization	ROL
Nonlinear solver	NOX
Time integration	Tempus
Krylov linear solvers/Prec	Belos/MueLu, FROSch
Automatic differentiation	Sacado
Field Dependency manager	Phalanx
Global DoF numbering	Panzer – DoF manager
Mesh data structure	STK
Nodal Finite element	Intrepid2



References:

1. Watkins et al., *IJHPCA*, 2023
2. Hoffman et al. *GMD*, 2018
3. Tuminaro, Perego, Tezaur, Salinger, Price, *SISC*, 2016.
4. Tezaur, Perego, Salinger, Tuminaro, Price, Hoffman, *GMD*, 2015
5. Perego, Price, Stadler, *JGR*, 2014

SOFTWARE: MPAS-ALBANY LAND ICE MODEL (MALI)

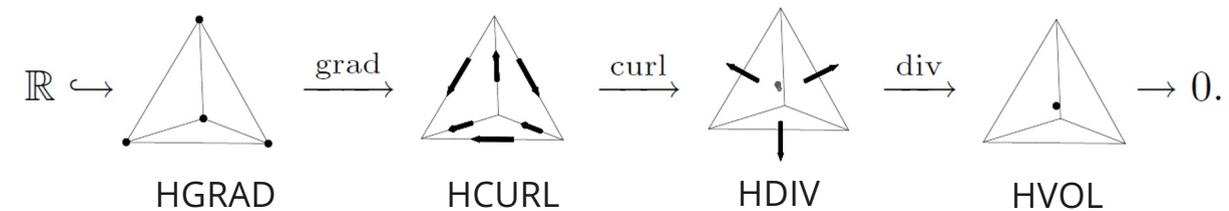


CAPABILITY	SOFTWARE TOOLS
Finite Elements ice-sheet solver	Albany
Constrained Optimization	ROL
Nonlinear solver	NOX
Time integration	Tempus
Krylov linear solvers/Prec	Belos/MueLu, FROSch
Automatic differentiation	Sacado
Field Dependency manager	Phalanx
Global DoF numbering	Panzer – DoF manager
Mesh data structure	STK
Nodal Finite element	Intrepid2



Intrepid2 (local discretizations)

- High-order FEM bases:
 - Lagrangian, hierarchical, serendipity
 - Full De-Rham complex (HGRAD-HCURL-HDIV-HVOL)
 - Orientation tools for matching DoFs
- Mappings between reference and physical elements
- Quadrature rules, Cell tools (e.g., inclusion tests)
- Projection tools
- kernels for integration exploiting structure: Tensor product, Extruded elements, Affine meshes



SOFTWARE: MPAS-ALBANY LAND ICE MODEL (MALI)



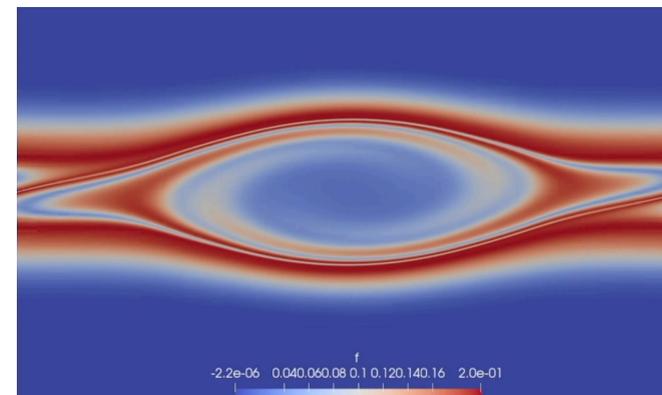
CAPABILITY	SOFTWARE TOOLS
Finite Elements ice-sheet solver	Albany
Constrained Optimization	ROL
Nonlinear solver	NOX
Time integration	Tempus
Krylov linear solvers/Prec	Belos/MueLu, FROSch
Automatic differentiation	Sacado
Field Dependency manager	Phalanx
Global DoF numbering	Panzer – DoF manager
Mesh data structure	STK
Nodal Finite element	Intrepid2



Intrepid2 (local discretizations)

- High-order FEM bases:
 - Lagrangian, hierarchical, serendipity
 - Full De-Rham complex (HGRAD-HCURL-HDIV-HVOL)
 - Orientation tools for matching DoFs
- Mappings between reference and physical elements
- Quadrature rules, Cell tools (e.g., inclusion tests)
- Projection tools
- kernels for integration exploiting structure:
 - Tensor product, Extruded elements, Affine meshes

Applications: MHD, Vlasov, solid mechanics, etc.



Phase-field distribution of Vlasov two-stream instability problem: Camellia simulation, courtesy of Nathan Roberts

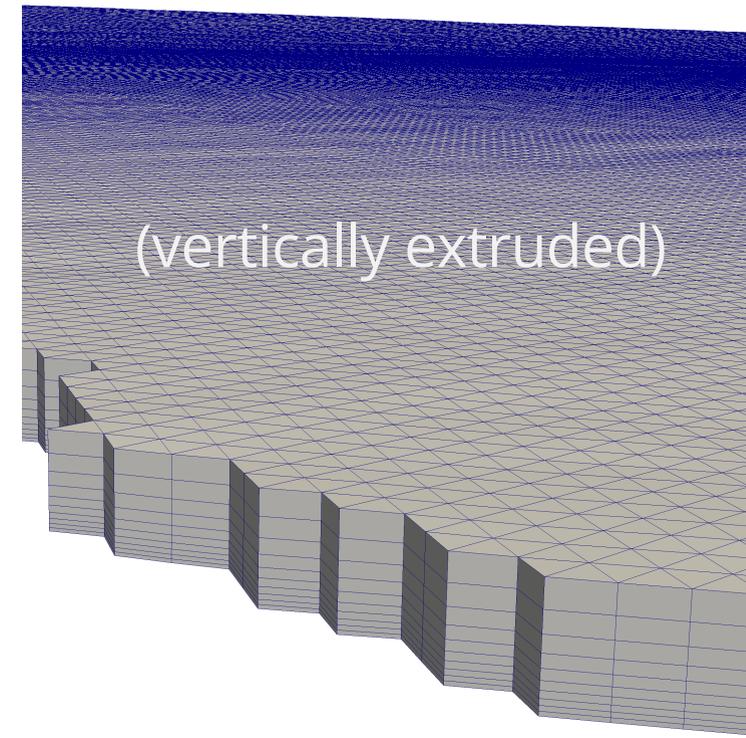
SOFTWARE: MPAS-ALBANY LAND ICE MODEL (MALI)



CAPABILITY	SOFTWARE TOOLS
Finite Elements ice-sheet solver	Albany
Constrained Optimization	ROL
Nonlinear solver	NOX
Time integration	Tempus
Krylov linear solvers/Prec	Belos/MueLu, FROSch
Automatic differentiation	Sacado
Field Dependency manager	Phalanx
▶ Global DoF numbering	Panzer – DoF manager
▶ Mesh data structure	STK
Nodal Finite element	Intrepid2



STK mesh:



Panzer DoF manager:

Global DoF numbered based on local FE DoFs, solution fields and mesh connectivity. Important for Multiphysics problems with nonnodal FE spaces

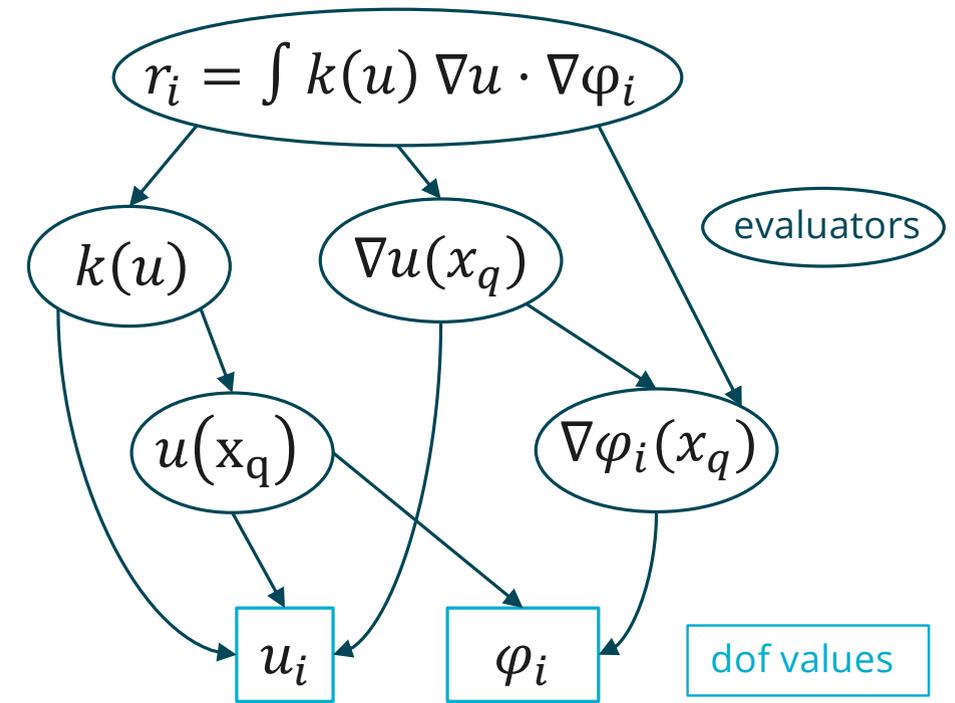
SOFTWARE: MPAS-ALBANY LAND ICE MODEL (MALI)



CAPABILITY	SOFTWARE TOOLS
Finite Elements ice-sheet solver	Albany
Constrained Optimization	ROL
Nonlinear solver	NOX
Time integration	Tempus
Krylov linear solvers/Prec	Belos/MueLu, FROSch
Automatic differentiation	Sacado
Field Dependency manager	Phalanx
Global DoF numbering	Panzer – DoF manager
Mesh data structure	STK
Nodal Finite element	Intrepid2



Phalanx* Field Dependency manager



Direct Acyclic Graph (DAG) of dependencies

*Notz, Pawlowski and Sutherland; ACM Trans. Math., 2012

SOFTWARE: MPAS-ALBANY LAND ICE MODEL (MALI)

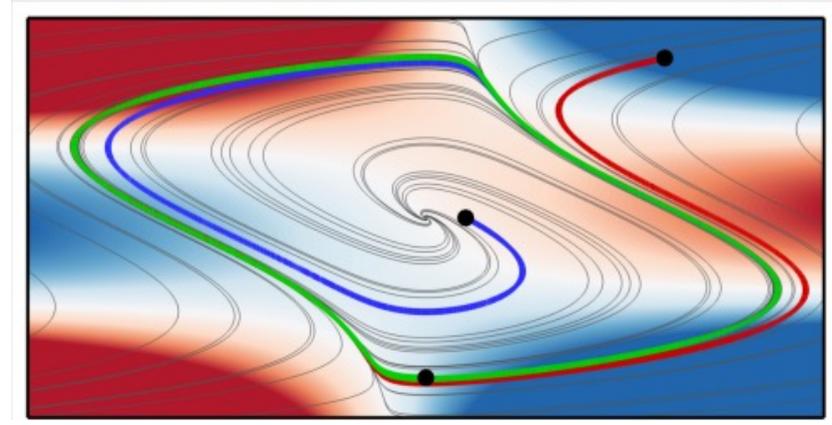


CAPABILITY	SOFTWARE TOOLS
Finite Elements ice-sheet solver	Albany
Constrained Optimization	ROL
Nonlinear solver	NOX
Time integration	Tempus
Krylov linear solvers/Prec	Belos/MueLu, FROSch
Automatic differentiation	Sacado
Field Dependency manager	Phalanx
Global DoF numbering	Panzer – DoF manager
Mesh data structure	STK
Nodal Finite element	Intrepid2



Tempus, Time integration

- Many explicit and implicit steppers: BDF, RK, IMEX, Leapfrog, ...
- Adaptive time-stepping
- (Adjoint) sensitivity analysis
- Interfaces with NOX/ROL



Courtesy of C. Ober

ICE-SHEET MODEL CALIBRATION

Goal: Find the initial/present-day thermo-mechanical state of the ice sheet and estimate the unknown/poorly known model parameters, by matching observations

Approach: ***PDE-constrained optimization***

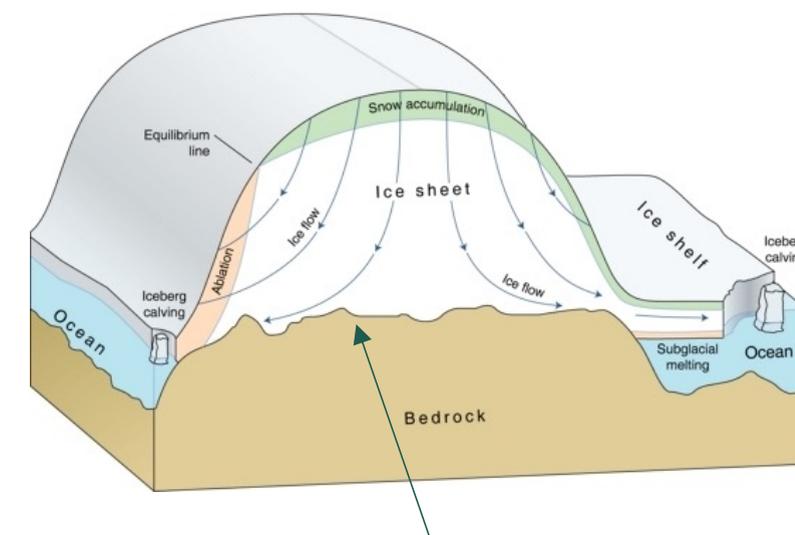
Find basal friction coefficient $\beta = \exp(\theta)$ that minimizes the mismatch with surface velocity:

$$\min_{\theta} \mathcal{J}(\theta, u) = \int_{\Omega} \frac{|u - u_{obs}|^2}{\sigma^2} + \mathcal{R}(\theta)$$

Subject to the coupled ice flow problem (constraint).

The constraint maps $\theta \rightarrow u(\theta)$.

Typical regularization term: $\mathcal{R}(\theta) = \alpha_0 |\theta|^2 + \alpha_1 |\nabla \theta|^2$



unknown sliding
parameter $\beta = \exp(\theta)$

THERMO-MECHANICAL CALIBRATION OF GREENLAND ICE SHEET

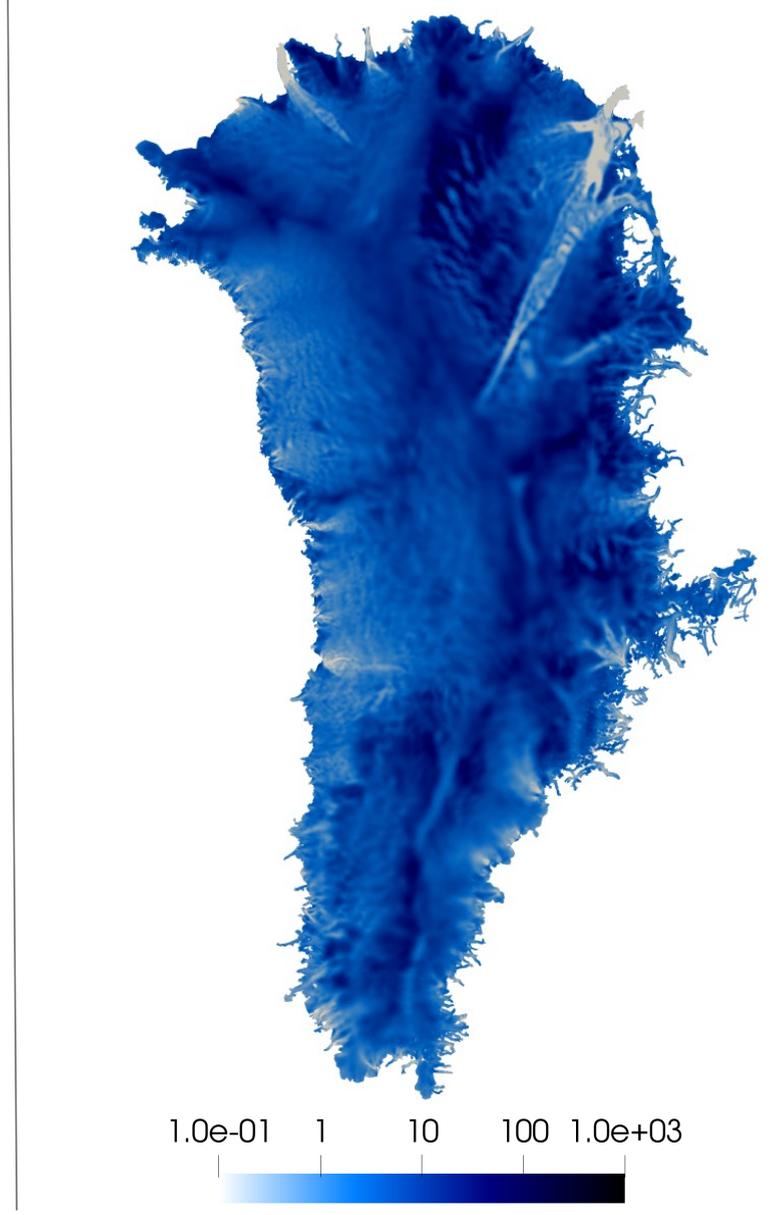
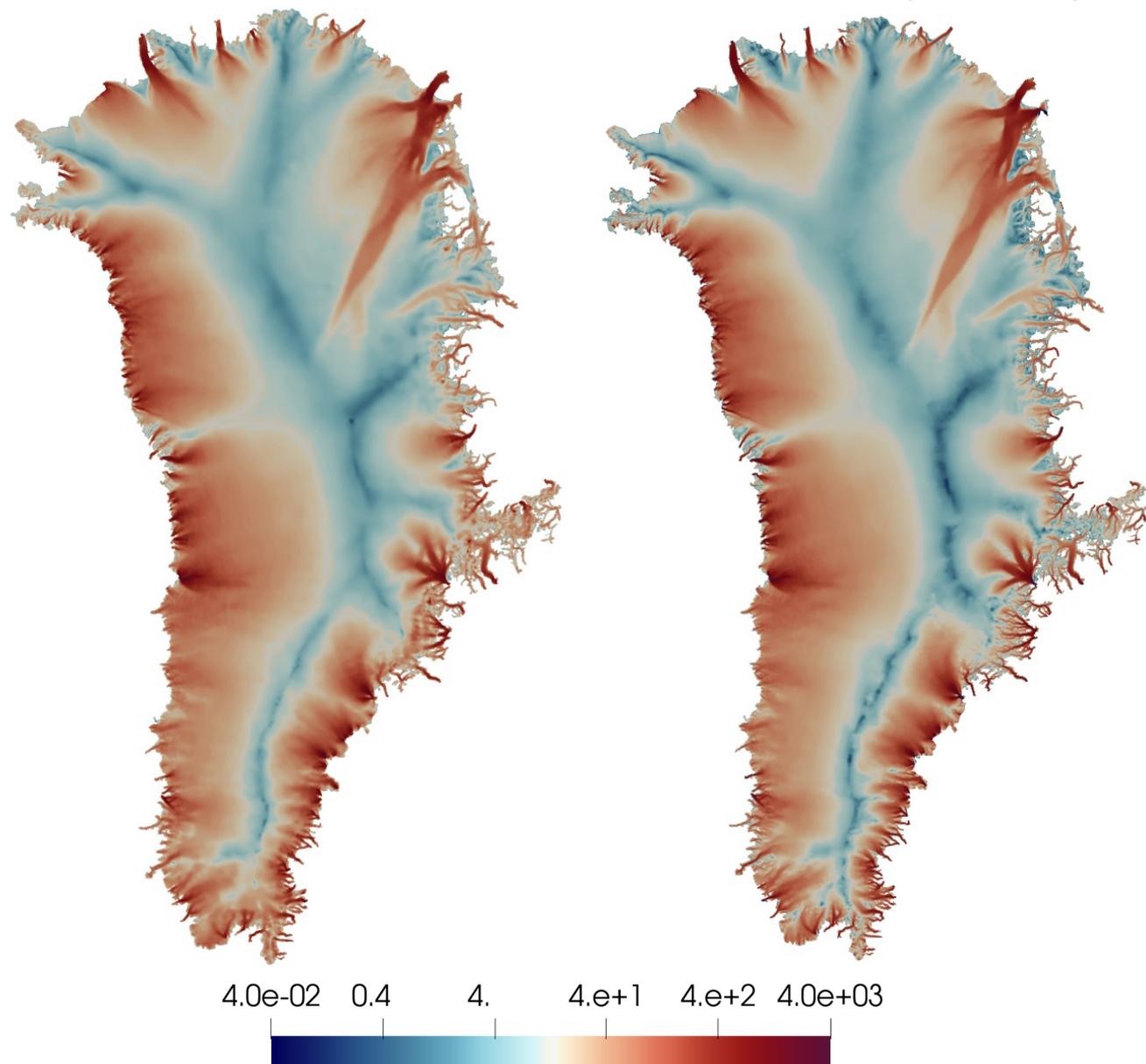


modeled ice speed [m/yr]

observed ice speed [m/yr]

modeled basal friction [kPa yr/m]

modeled temperature [K]



Variable resolution 1-10km mesh, **300K parameters, 14M unknowns.**

The optimization is constrained by the **coupled velocity-temperature** solvers. As a byproduct of the optimization we get an initial temperature field that is consistent with the velocity.

ICE-SHEET MODEL CALIBRATION



Numerical Optimization approach:

- **Trust Region method (Lin-Moré)**, using truncated CG for solving the quadratic subproblem. Requires computation of reduced gradient ($\nabla_{\theta} \mathcal{J}$, first total derivative of \mathcal{J}) and reduced Hessian ($\nabla_{\theta\theta} \mathcal{J}$, second total derivative of \mathcal{J}) to create a quadratic approximation of the objective \mathcal{J} .
- All first and second order derivatives are computed using **Automatic Differentiation** and **adjoints**.

ROL requires

- Definition of a Vector (and operations like sum, norm, dot product)
we use `Tpetra::MultiVector`
- For a given guess for θ , ROL requires $\mathcal{J}(\theta)$ and its first and second derivatives.





ICE-SHEET MODEL CALIBRATION – HESSIAN COMPUTATION WITH AD

PDE-constrained optimization methods benefit from Gradients and Hessian mat-vec products, computed with Automatic Differentiation (AD) using Sacado package:

➤ Gradient $\partial_{\theta} \mathcal{J}(\theta)$

Sacado derivative type (value + derivs)

\mathcal{J}	\mathcal{J}_{θ_0}	\mathcal{J}_{θ_1}	\mathcal{J}_{θ_2}	...
---------------	--------------------------	--------------------------	--------------------------	-----

➤ Hessian $\partial_{\theta\theta} \mathcal{J}(\theta) \mathbf{v} = \partial_r (\partial_{\theta} \mathcal{J}(\theta + r \mathbf{v})) \Big|_{r=0}$

Sacado nested derivatives (2d array)

\mathcal{J}	\mathcal{J}_{θ_0}	\mathcal{J}_{θ_1}	\mathcal{J}_{θ_2}	...
\mathcal{J}_r	$\mathcal{J}_{\theta_0,r}$	$\mathcal{J}_{\theta_1,r}$	$\mathcal{J}_{\theta_2,r}$...

Sacado provides specializations for Kokkos views to efficiently handle derivative types. Layout of Sacado type can be chosen to effectively use architectures and boost parallelism*

*Phipps, Pawlowski and Trott; ACM Transactions on Mathematical Software, 2022

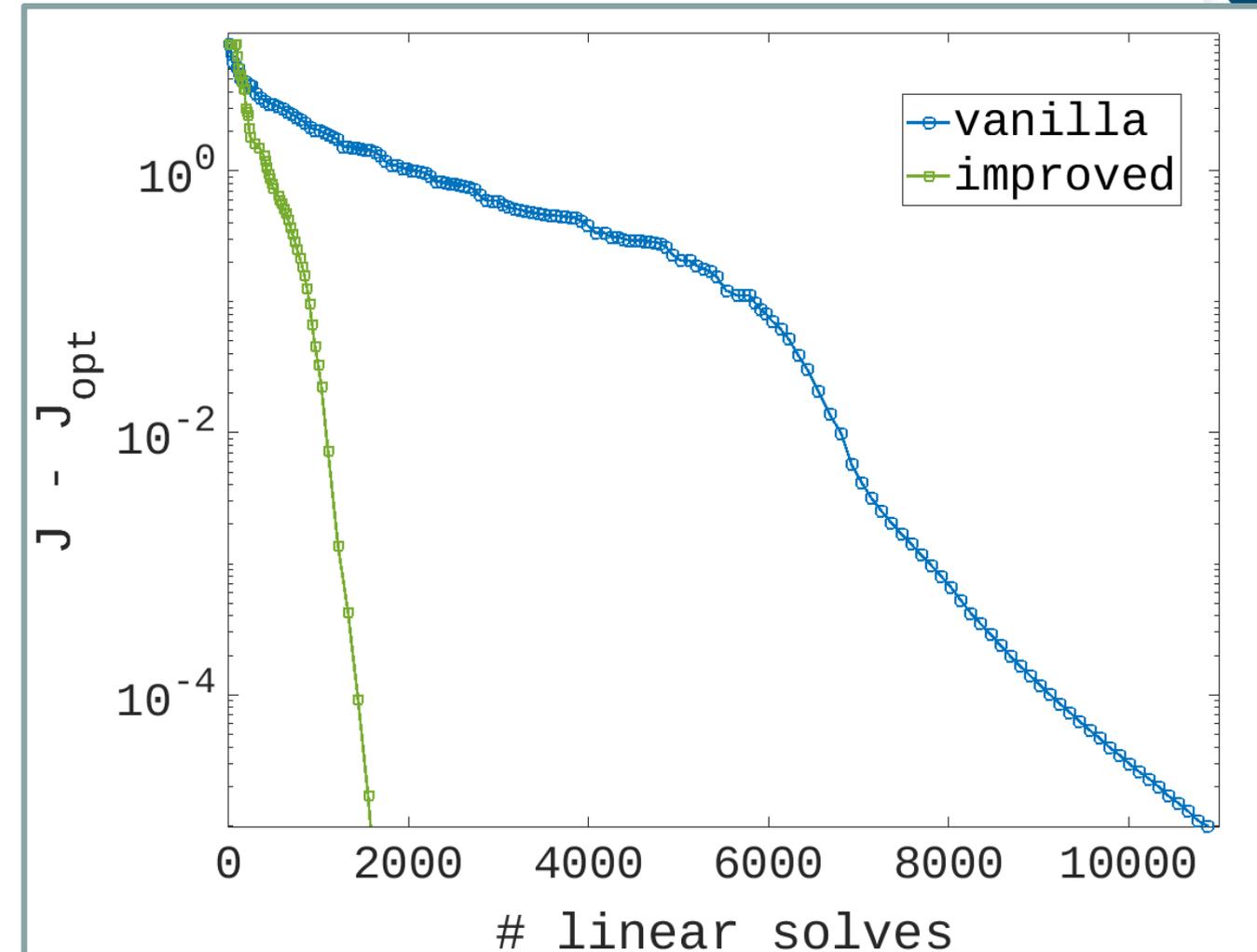
ICE SHEET CALIBRATION: PDE CONSTRAINED OPTIMIZATION



➤ Cost is often dominated by evaluation of reduced Hessian (and in particular by the assembly phase)

Improved Strategy using ROL advanced features:

- Use **mass-based dot-product** to define the gradient. Gradient \mathcal{G} is given by $\nabla_{\theta} \mathcal{J} v = (\mathcal{G}, v)$. Instead of $(u, v)_{L^2} = u^t v$, use $(u, v)_{L^2} = u^t M v$. Here M is the lumped mass matrix. Hence $\mathcal{G} := M^{-1} \nabla_{\theta} \mathcal{J}$.
- Use **preconditioner** for the reduced Hessian:
 - Use a low-rank approximation of the reduced Hessian (e.g., BFGS) as preconditioner,
 - Initialize BFGS with $\nabla_{pp} \mathcal{R}(p)$

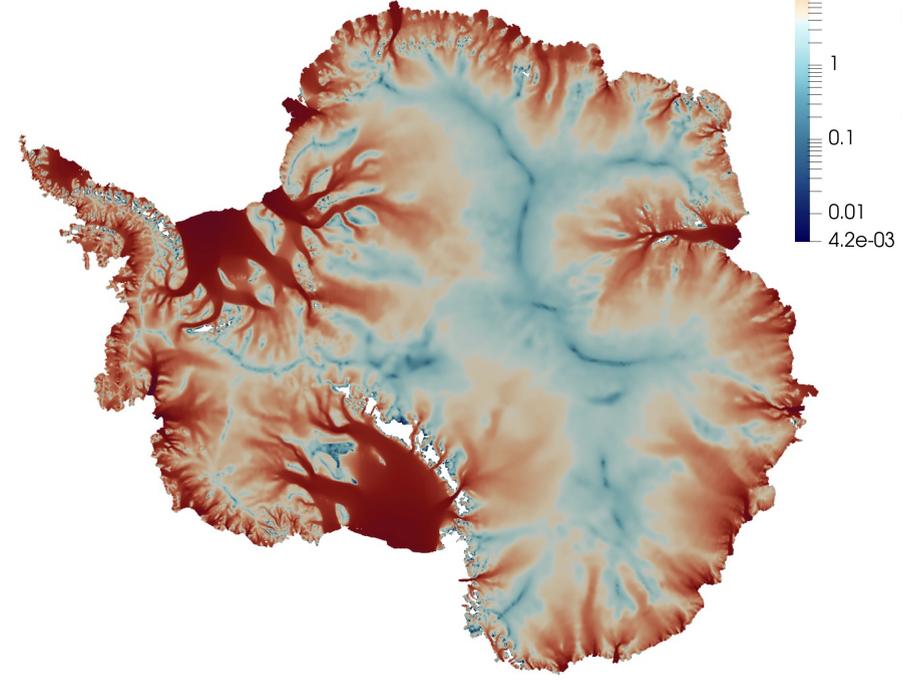


Problem: Optimization to initialize Humboldt glacier
Ice velocity depends on friction parameter. Ice temperature held constant.

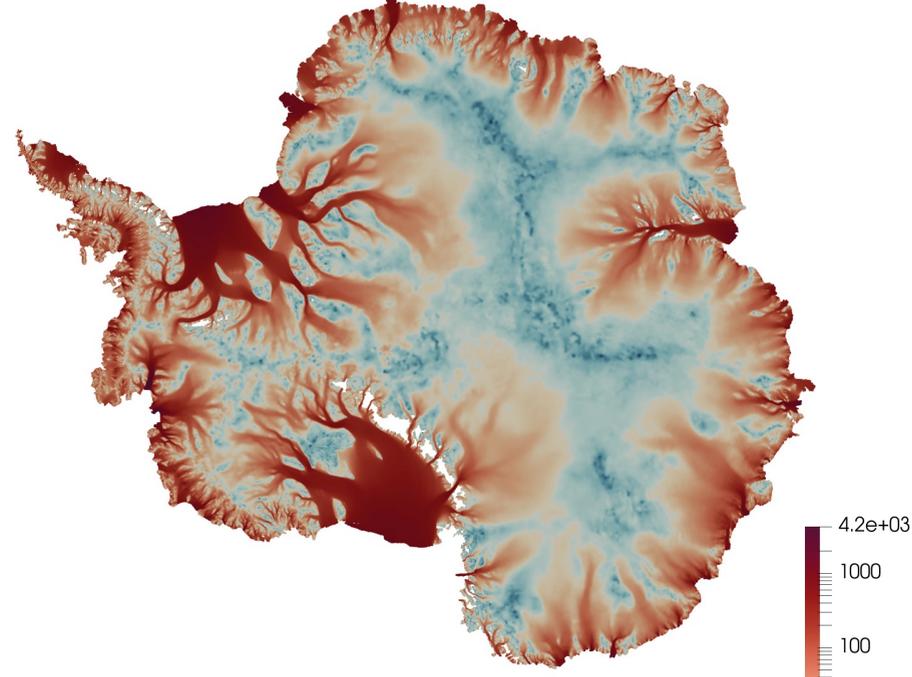
ICE SHEET CALIBRATION: ANTARCTIC ICE SHEET



modeled ice speed [m/yr]



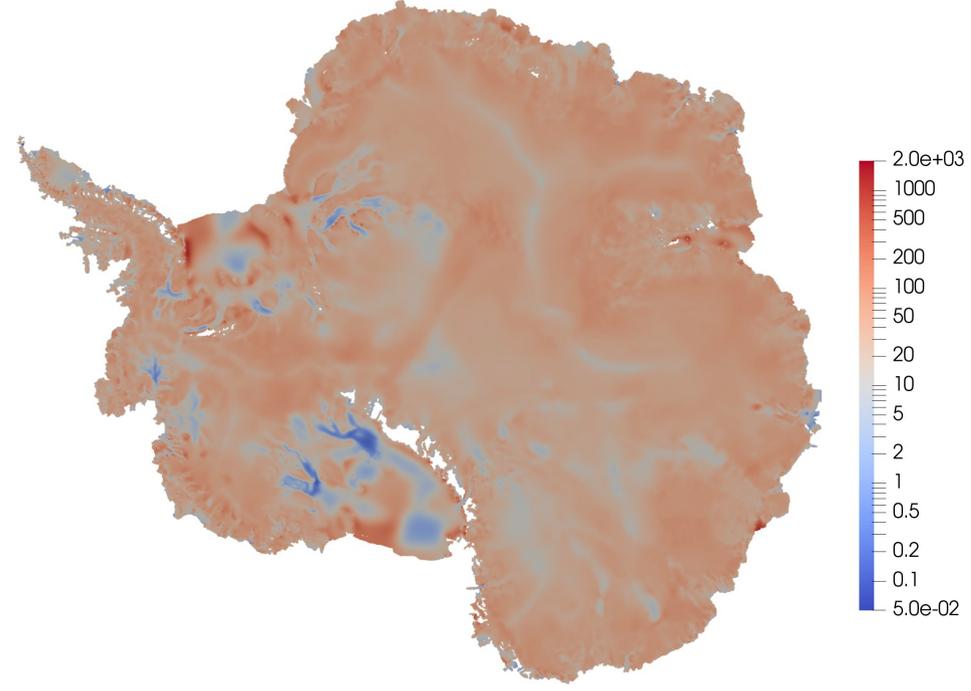
observed ice speed [m/yr]



Stiffening factor



basal friction



ICE SHEET CALIBRATION: PDE CONSTRAINED OPTIMIZATION



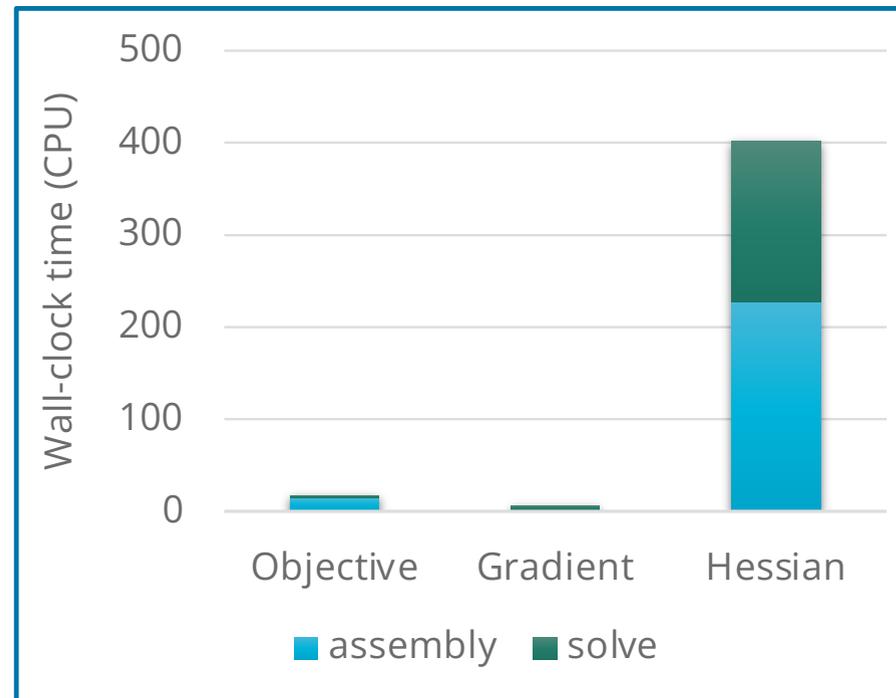
- Cost is often dominated by evaluation of reduced Hessian (specifically by the assembly phase)

Perlmutter architecture

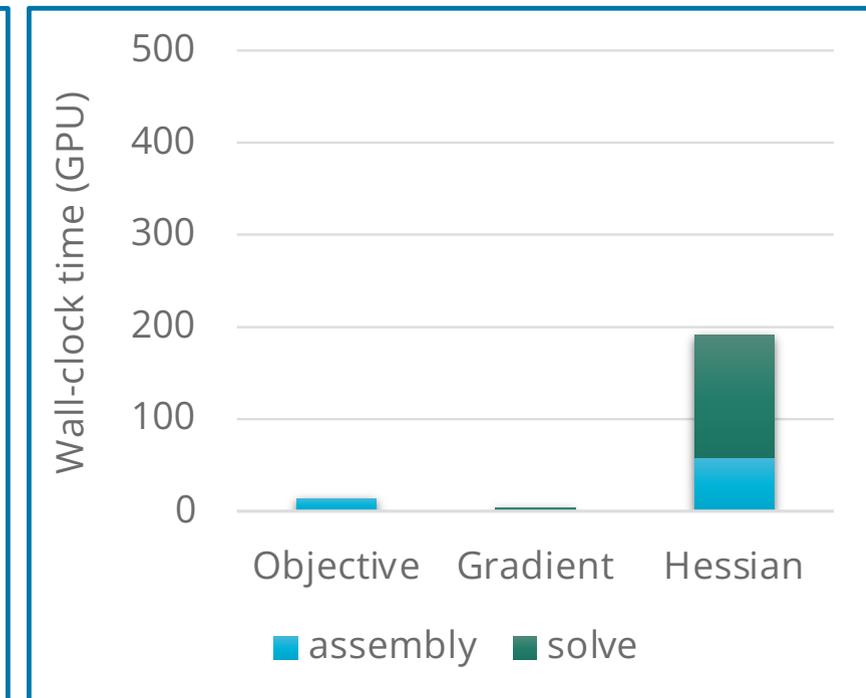
CPU node: 2 AMD Milan (64 cores each)

GPU node: 1 AMD Milan + 4 NVIDIA A100

CPU node



GPU node



Problem: Optimization to initialize Antarctic glacier
Ice velocity depends on friction parameter and stiffening. Ice temperature held constant.

HOW TO USE
TRILINOS TOOLS
IN YOUR APPLICATION



ADOPTING TRILINOS TOOLS IN FINITE ELEMENT APPLICATIONS



Your Case	You can use	Notes
I'd like to add advanced capabilities to my application with not-too-intrusive changes	Intrepid2 w/ Dof Manager, (py)ROL	Approach used by <i>Camellia, Sierra, Plato, ..</i> Easy to interface w/ (py)ROL
I like the Trilinos model evaluator concept and AD. I'm happy to restructure my application.	Intrepid2, Sacado, Phalanx, Tempus (Py)ROL	Approach used by <i>Albany</i> Application in charge of mesh, assembly, solvers.
I love everything Trilinos, I'm starting from scratch.	Panzer	Panzer provides Mesh I/O, space and time discretization, linear and nonlinear solvers, sensitivities and optimization solvers <i>Drekar, Empire</i> use this approach.

INCOMPLETE LIST OF CONTRIBUTORS AND SPONSOR



Contributors:

Ross Bartlett,
Pavel, Bochev,
Brian Carnes,
Eric Cyr,
Carter Edward,
Nathan Ellingwood,
Christian Glusa,
Kyungjoo Kim,
Drew Kouri,
Paul Kuberry
Kim Liegeois,
David Noble,
Roger Pawlowski,
Mauro Perego,
Kara Peterson,
Eric Phipps,
Curt Ober,
Bryan Reuter,
Denis Ridzal,

Nathan Roberts,
Andy Salinger,
Chris Siefert,
Irina Tezaur,
Greg Von Winckel,
Alan Williams

Sponsors:

