



The Future of Earth System Modeling: Polar Climates

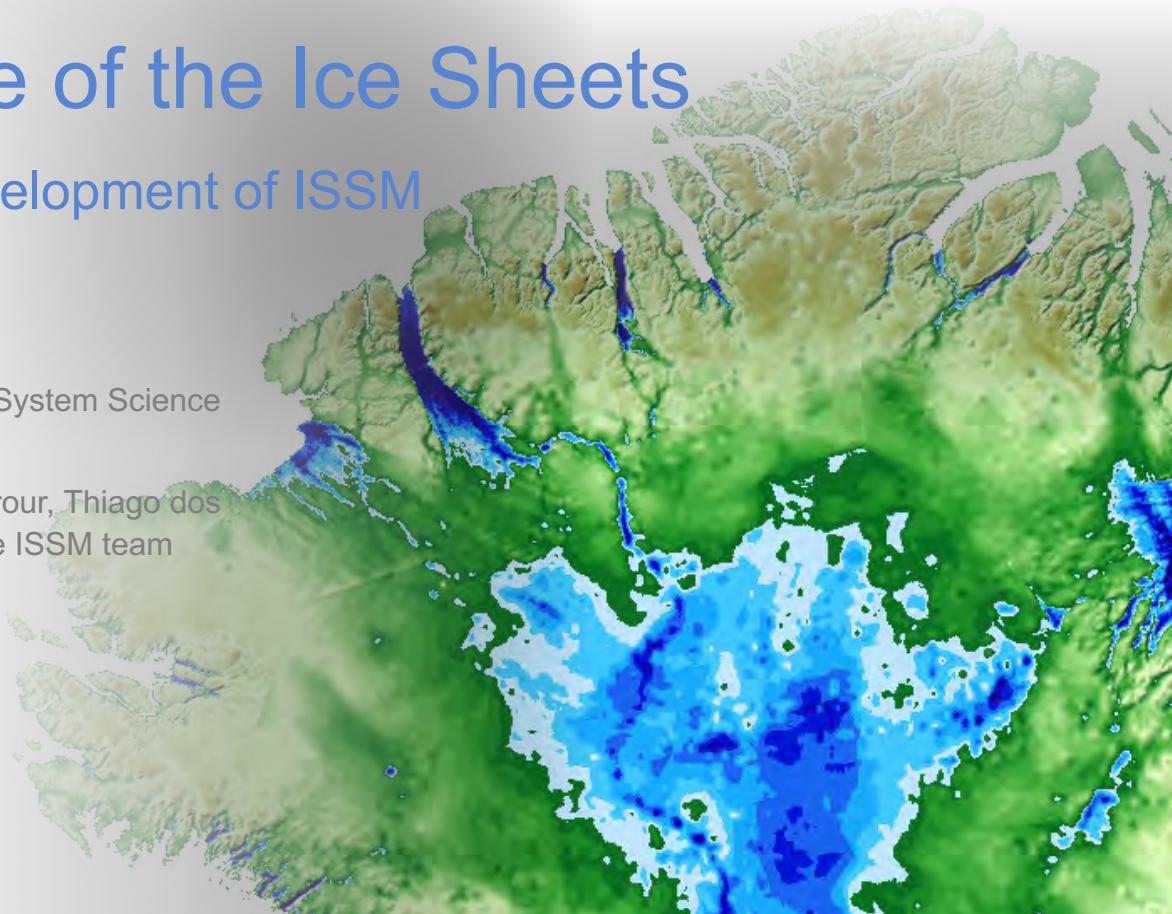
Modeling the Future of the Ice Sheets

Lessons Learned from the Development of ISSM

Mathieu MORLIGHEM*

University of California Irvine, Department of Earth System Science

*with contributions from Johannes Bondzio, Eric Larour, Thiago dos Santos, Helene Seroussi, and other members of the ISSM team



Observations

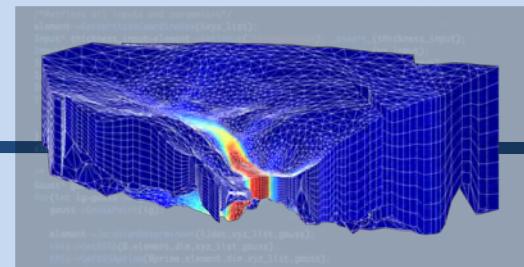


Discrepancy ?

Numerical model

Input parameters

- Bed topography
- Forcings
- Initial conditions
- ...



Model output

- Ice temperature
- Mass balance
- Surface velocities
- ...

Model physics

Parameterization of Physical processes

- Basal friction
- Iceberg Calving
- ...

Energy balance

- Heat transfer

$$\frac{\partial T}{\partial t} = -\mathbf{v} \cdot \nabla T + \frac{k_{th}}{\rho c} \Delta T + \frac{\Phi}{\rho c}$$

Stress balance

- Incompressible Stokes flow

$$\nabla \cdot \boldsymbol{\sigma}' - \nabla P + \rho \mathbf{g} = \mathbf{0}$$

Mass balance

- Incompressibility

$$\frac{\partial H}{\partial t} = -\nabla \cdot H \bar{\mathbf{v}} + \dot{M}_s - \dot{M}_b$$

Outline

1. Physics of ice sheet flow

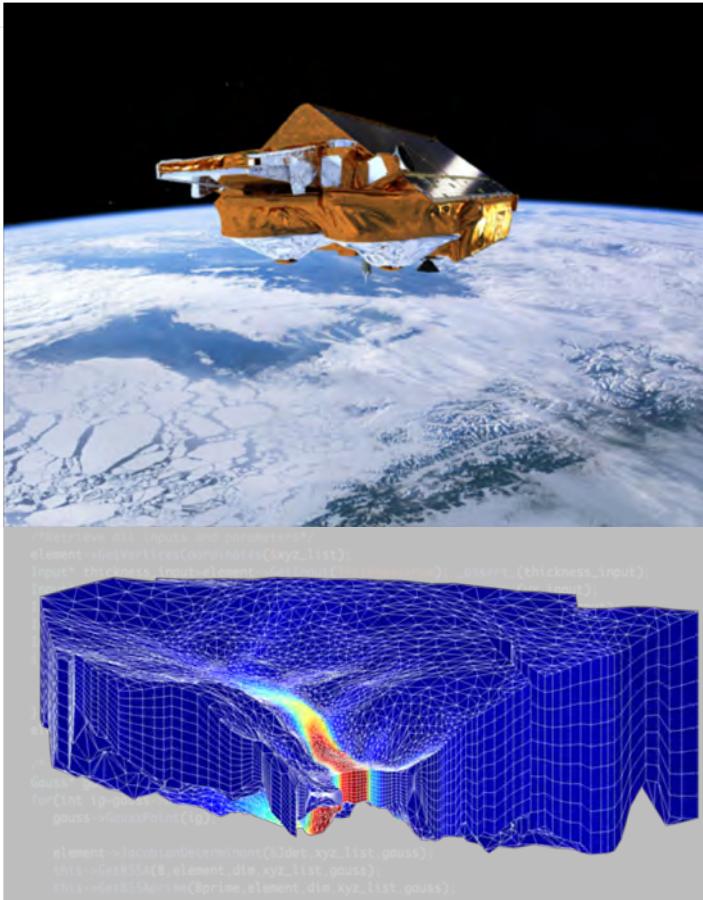
- Momentum balance and possible approximations
 - Key physical processes:
 - Grounding line dynamics
 - Calving dynamics

2. Numerical strategies

- Mesh refinement and resolution of PDEs
 - Verification and Validation
 - Dealing with moving boundaries

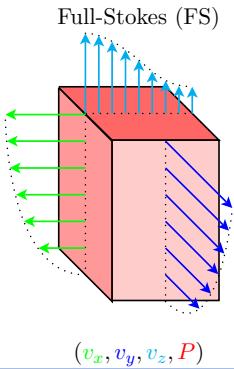
3. Project management

- Code development and architecture
 - Regression tests
 - Project management web interface



Hierarchy of ice flow models

$$\left\{ \begin{array}{l} \frac{\partial}{\partial x} \left(2\mu \frac{\partial v_x}{\partial x} \right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial v_x}{\partial y} + \mu \frac{\partial v_y}{\partial x} \right) + \frac{\partial}{\partial z} \left(\mu \frac{\partial v_x}{\partial z} + \mu \frac{\partial v_z}{\partial x} \right) - \frac{\partial p}{\partial x} = 0 \\ \\ \frac{\partial}{\partial x} \left(\mu \frac{\partial v_x}{\partial y} + \mu \frac{\partial v_y}{\partial x} \right) + \frac{\partial}{\partial y} \left(2\mu \frac{\partial v_y}{\partial y} \right) + \frac{\partial}{\partial z} \left(\mu \frac{\partial v_y}{\partial z} + \mu \frac{\partial v_z}{\partial y} \right) - \frac{\partial p}{\partial y} = 0 \\ \\ \frac{\partial}{\partial x} \left(\mu \frac{\partial v_x}{\partial z} + \mu \frac{\partial v_z}{\partial x} \right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial v_y}{\partial z} + \mu \frac{\partial v_z}{\partial y} \right) + \frac{\partial}{\partial z} \left(2\mu \frac{\partial v_z}{\partial z} \right) - \frac{\partial p}{\partial z} - \rho g = 0 \\ \\ \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} = 0 \end{array} \right.$$



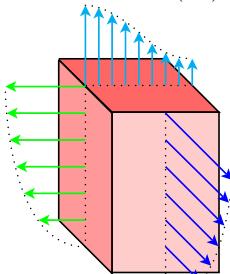
Hierarchy of ice flow models

$$\left\{ \begin{array}{l} \frac{\partial}{\partial x} \left(2\mu \frac{\partial v_x}{\partial x} \right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial v_x}{\partial y} + \mu \frac{\partial v_y}{\partial x} \right) + \frac{\partial}{\partial z} \left(\mu \frac{\partial v_x}{\partial z} + \mu \frac{\partial v_z}{\partial x} \right) - \frac{\partial p}{\partial x} = 0 \\ \\ \end{array} \right.$$

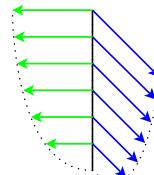
$$\mathbf{v}(z) = \mathbf{v}_b - 2(\rho g)^n \|\nabla s\|^{n-1} \nabla s \int_b^z (s-z)^n A \, dz$$

$$\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} = 0$$

Full-Stokes (FS)

 (v_x, v_y, v_z, P)

Hutter (SIA)

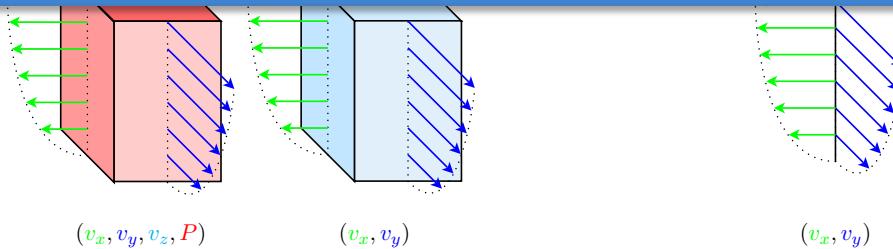
 (v_x, v_y) 

Hierarchy of ice flow models

$$\nabla \cdot (2\mu \dot{\varepsilon}_{HO1}) = \rho g \frac{\partial s}{\partial x}$$

$$\nabla \cdot (2\mu \dot{\varepsilon}_{HO2}) = \rho g \frac{\partial s}{\partial y}$$

$$\dot{\varepsilon}_{HO1} = \begin{bmatrix} 2\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} \\ \frac{1}{2} \left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x} \right) \\ \frac{1}{2} \frac{\partial v_x}{\partial z} \end{bmatrix} \quad \dot{\varepsilon}_{HO2} = \begin{bmatrix} \frac{1}{2} \left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x} \right) \\ \frac{\partial v_x}{\partial x} + 2\frac{\partial v_y}{\partial y} \\ \frac{1}{2} \frac{\partial v_y}{\partial z} \end{bmatrix}$$

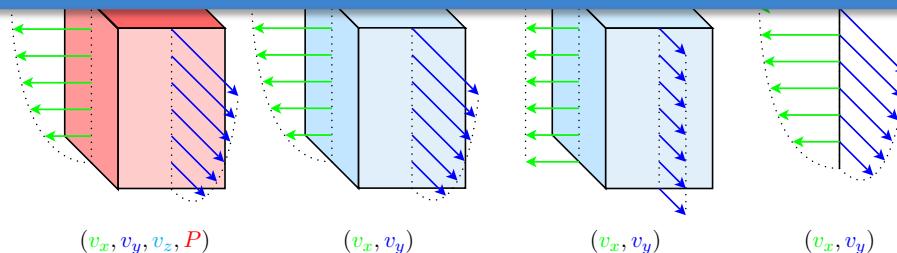


Hierarchy of ice flow models

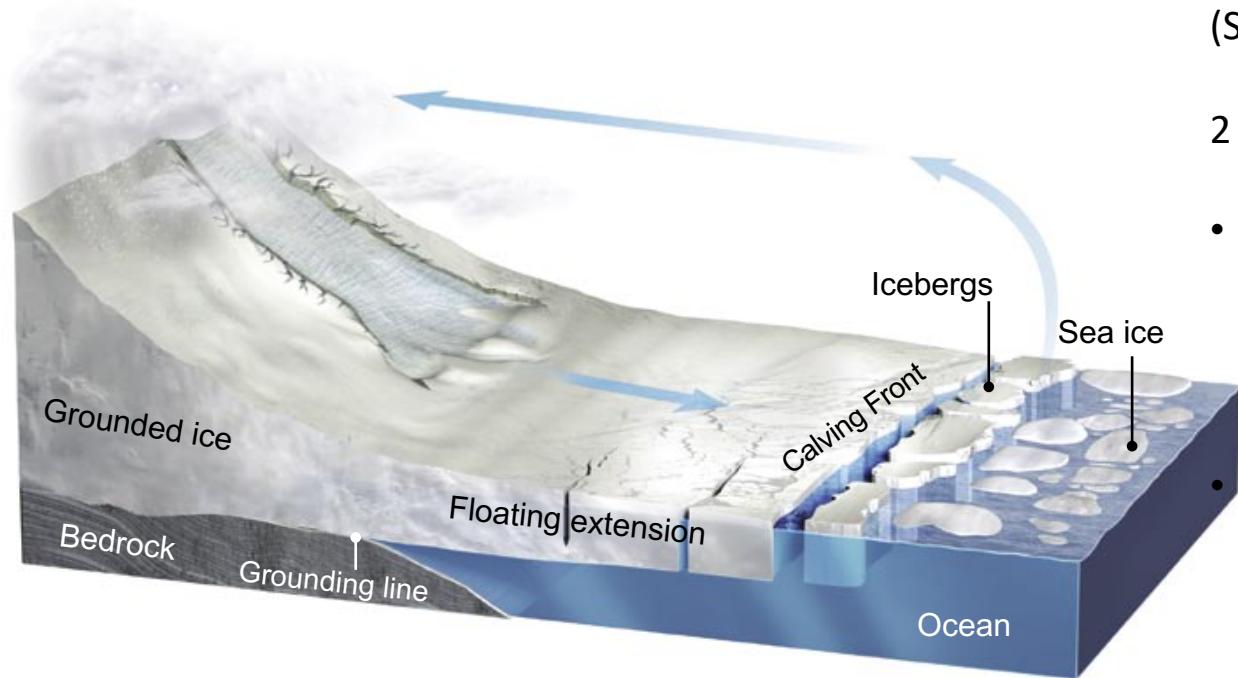
$$\nabla \cdot (2\bar{\mu}H\dot{\varepsilon}_{SSA1}) - \alpha^2 v_x = \rho g H \frac{\partial s}{\partial x}$$

$$\nabla \cdot (2\bar{\mu}H\dot{\varepsilon}_{SSA2}) - \alpha^2 v_y = \rho g H \frac{\partial s}{\partial y}$$

$$\dot{\varepsilon}_{SSA1} = \begin{bmatrix} 2\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} \\ \frac{1}{2}\left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right) \end{bmatrix} \quad \dot{\varepsilon}_{SSA2} = \begin{bmatrix} \frac{1}{2}\left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right) \\ \frac{\partial v_x}{\partial x} + 2\frac{\partial v_y}{\partial y} \end{bmatrix}$$



Grounding Line Dynamics (MISI)



Lots of research over the past 10 years
(See MISMIP, MISMIP3d, MISMIP+,...)

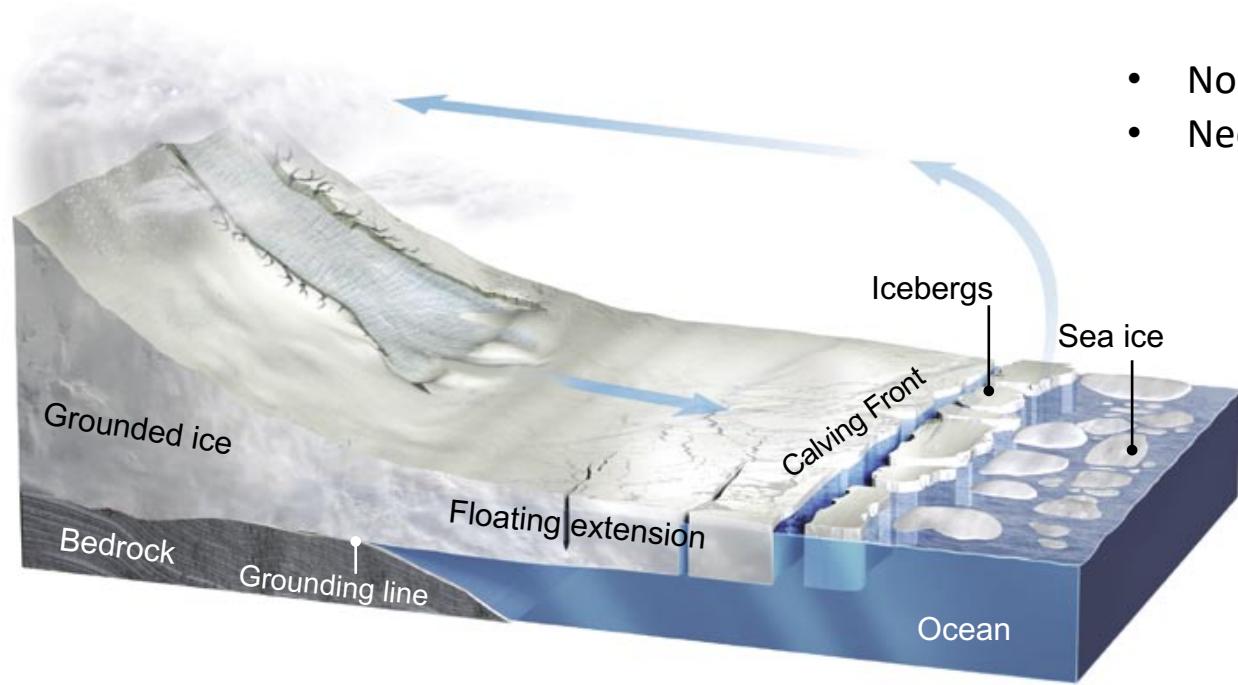
2 possible strategies:

- Based on hydrostatic equilibrium (SSA, HO, L1L2, ...)
 - Subelement parameterization
 - Mesh requirements: ~500 m
- Based on contact mechanics (FS)
 - No subelement param. parameterization
 - Mesh requirements: <100 m

Calving Dynamics (MICI?)

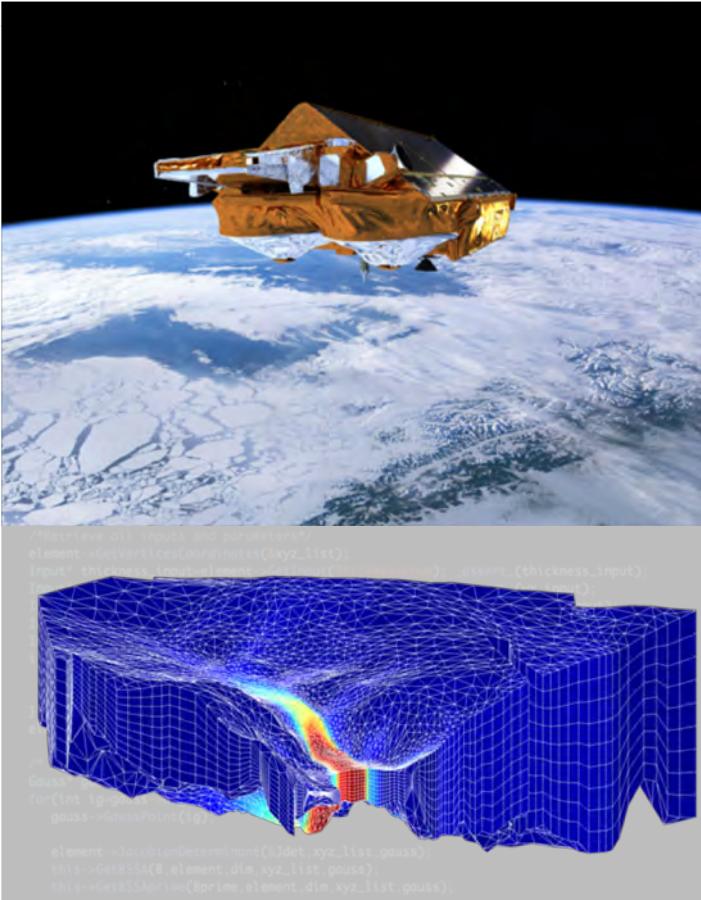
Currently very active area of research

- No “universal” calving law currently exists
- Needs to be included in ice sheet models

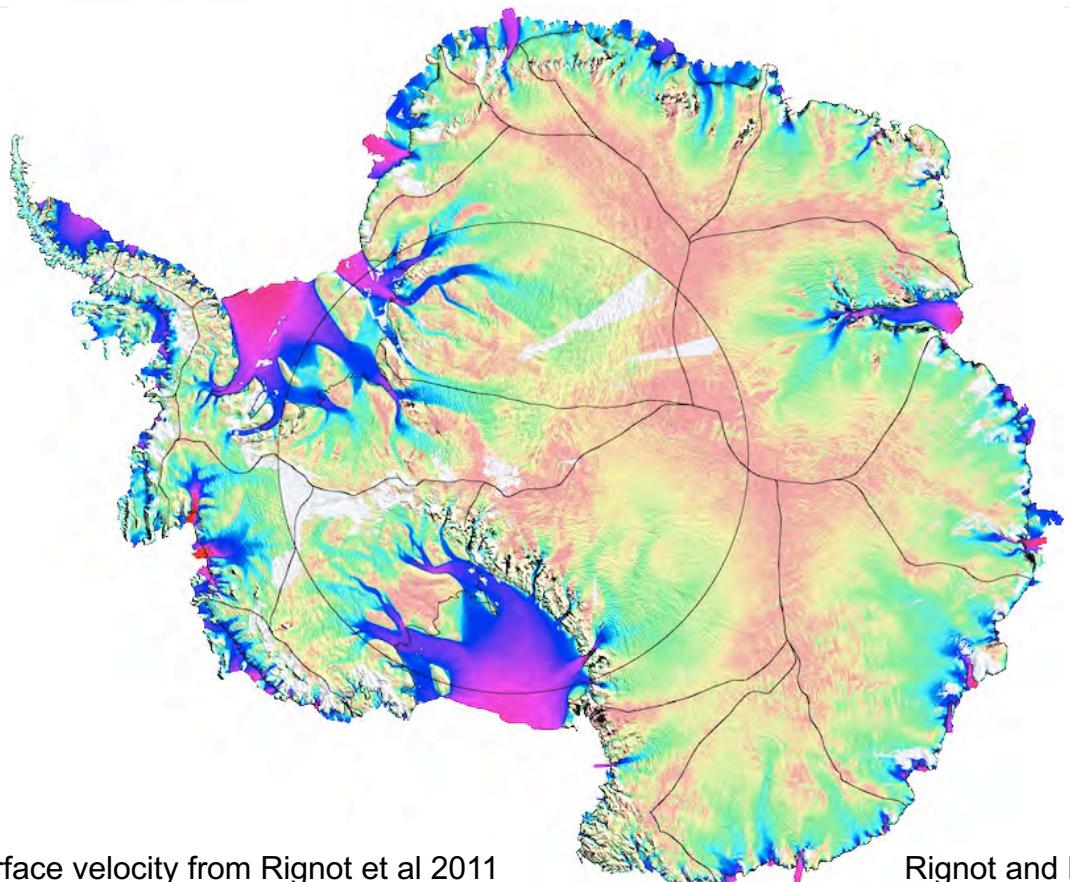


Outline

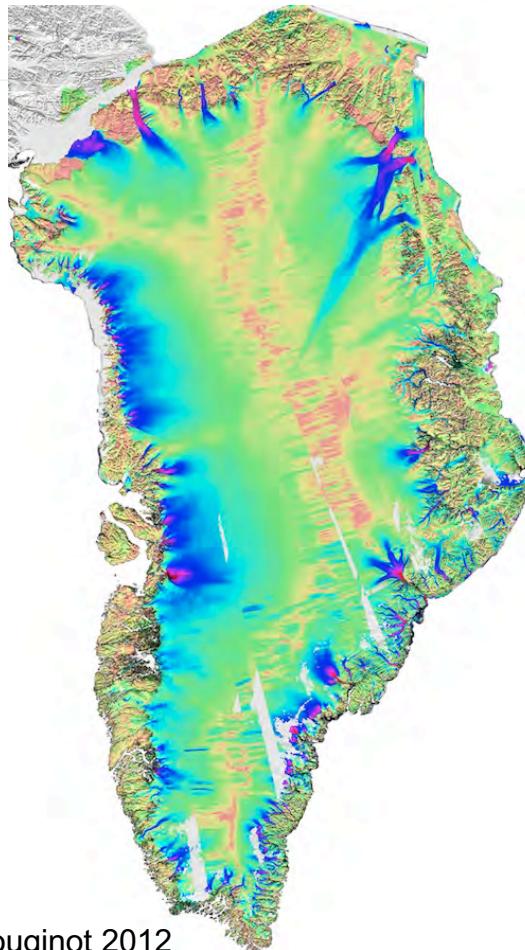
1. Physics of ice sheet flow
 - Momentum balance and possible approximations
 - Key physical processes:
 - Grounding line dynamics
 - Calving dynamics
2. Numerical strategies
 - Mesh refinement and resolution of PDEs
 - Verification and Validation
 - Dealing with moving boundaries
3. Project management
 - Code development and architecture
 - Regression tests
 - Project management web interface



Ice sheet flow

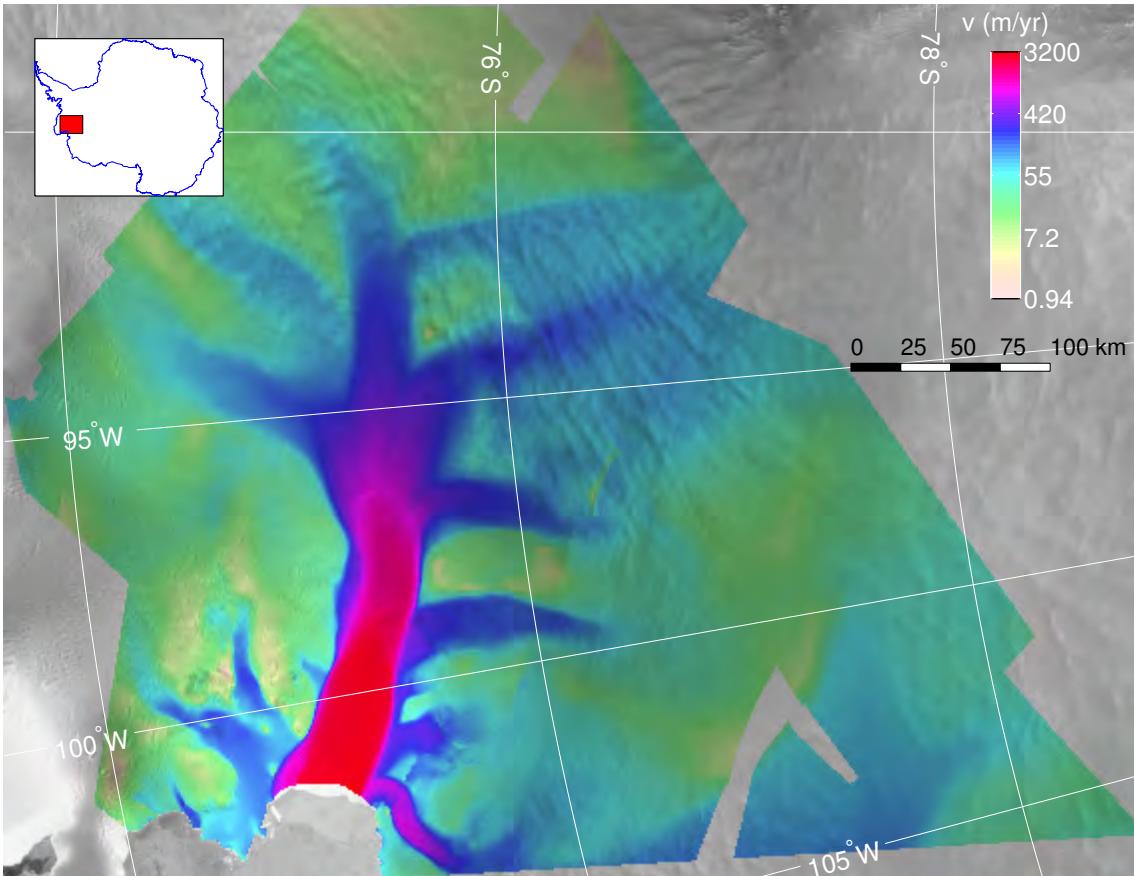


Surface velocity from Rignot et al 2011

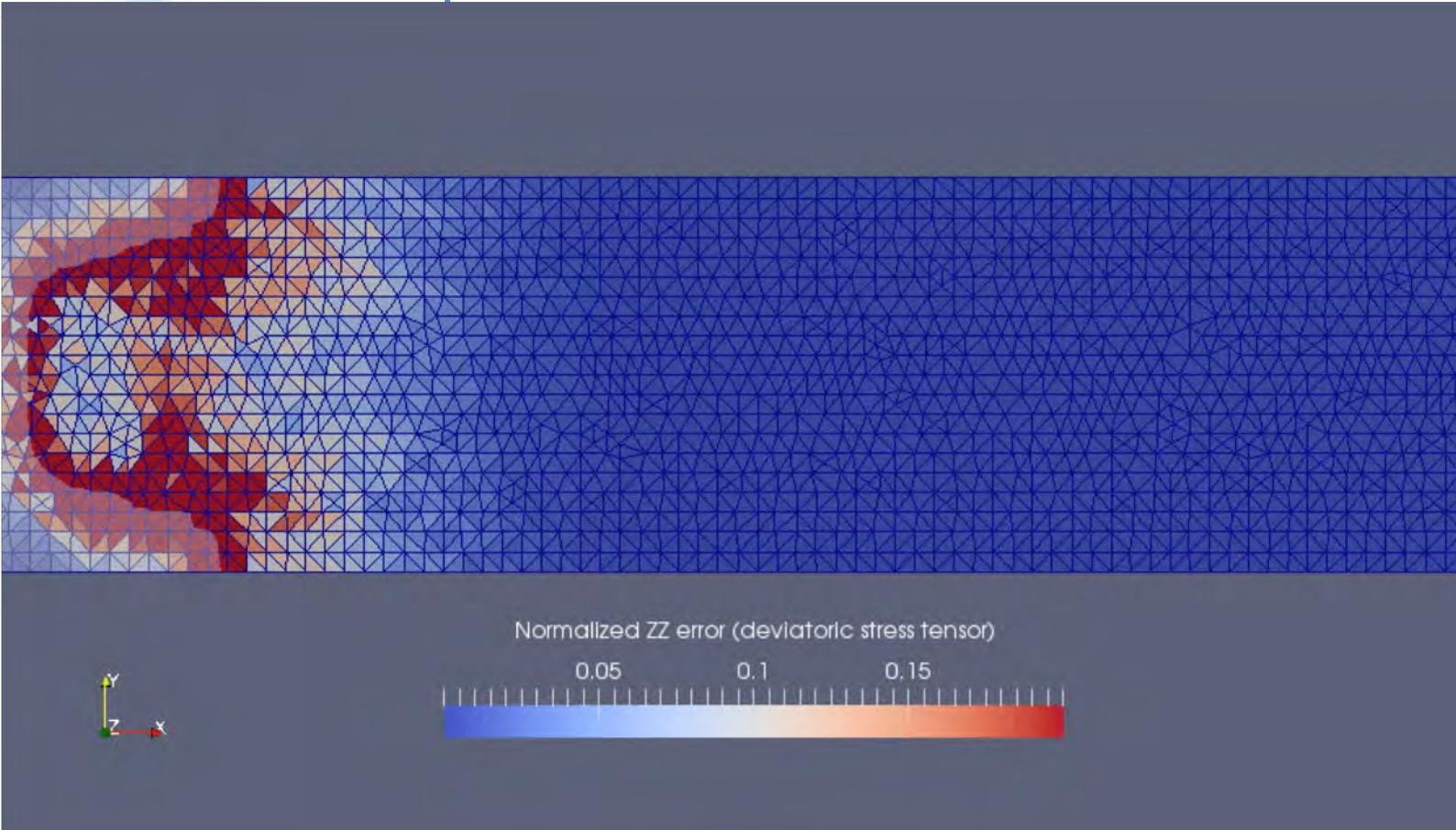


Rignot and Mouginot 2012

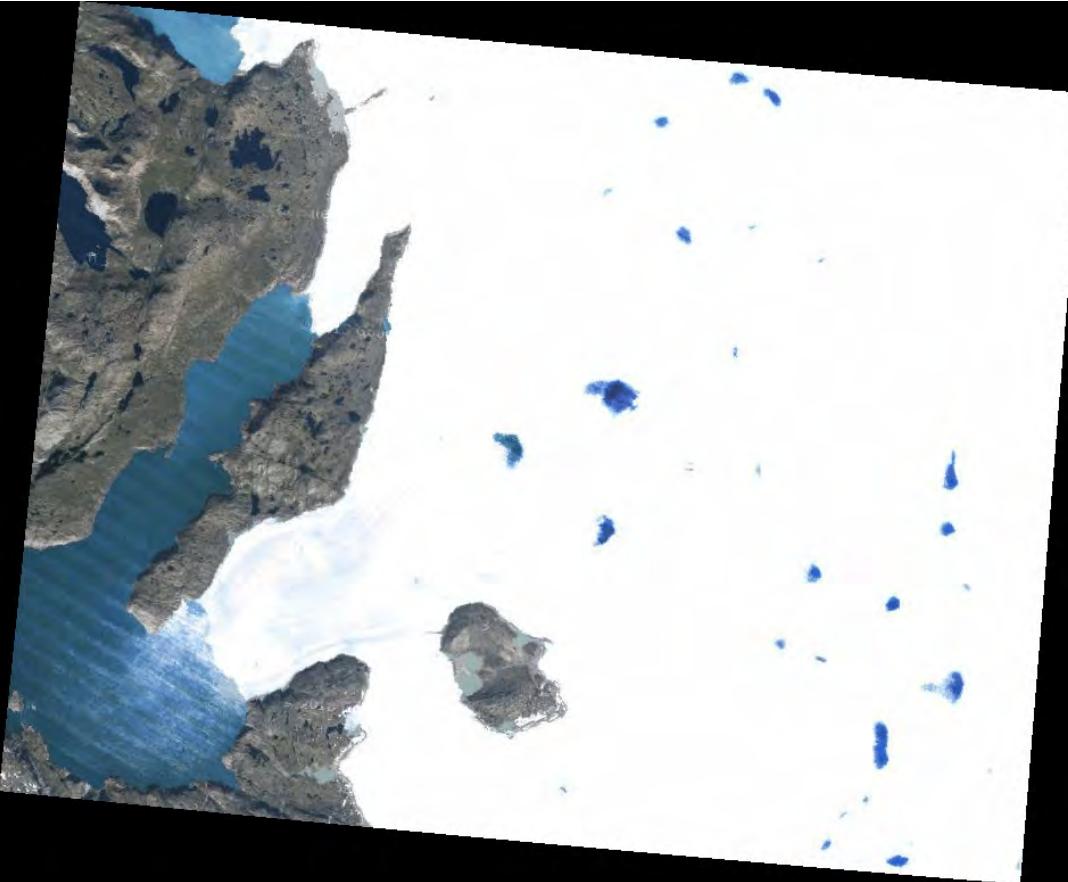
Anisotropic Mesh refinement



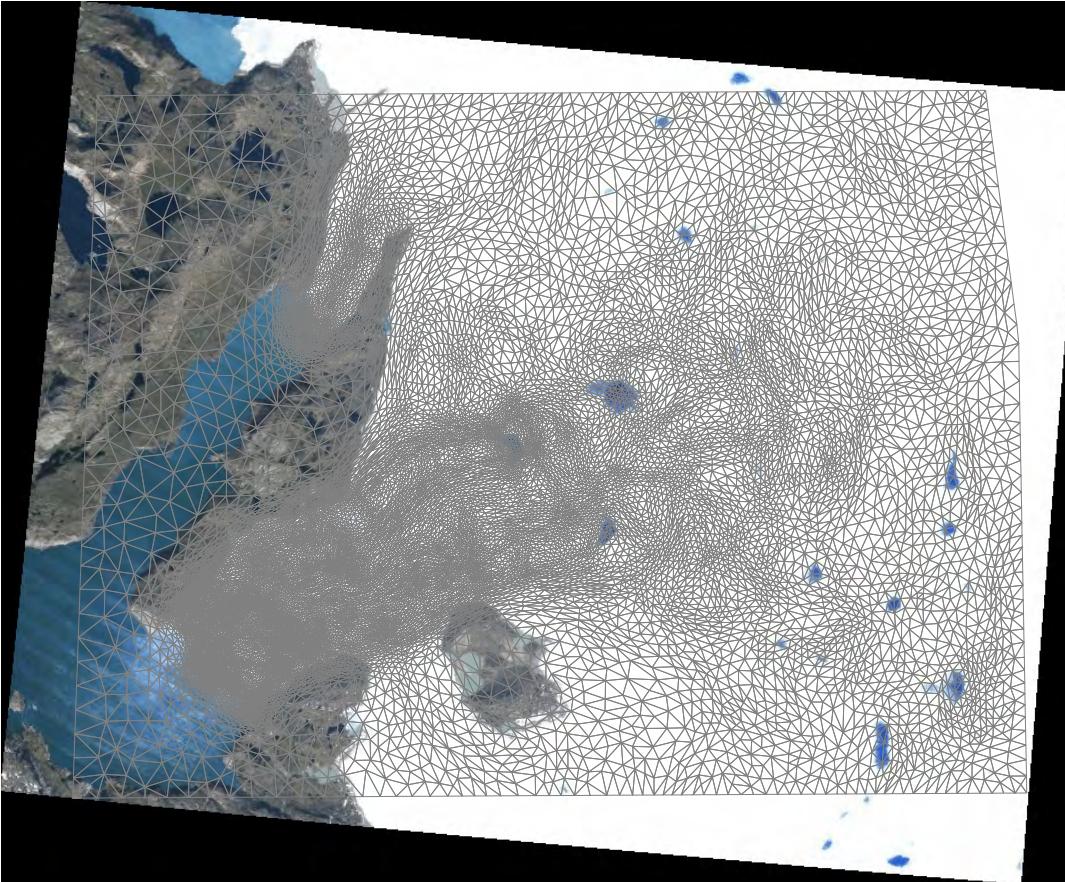
Adaptive Mesh Refinement



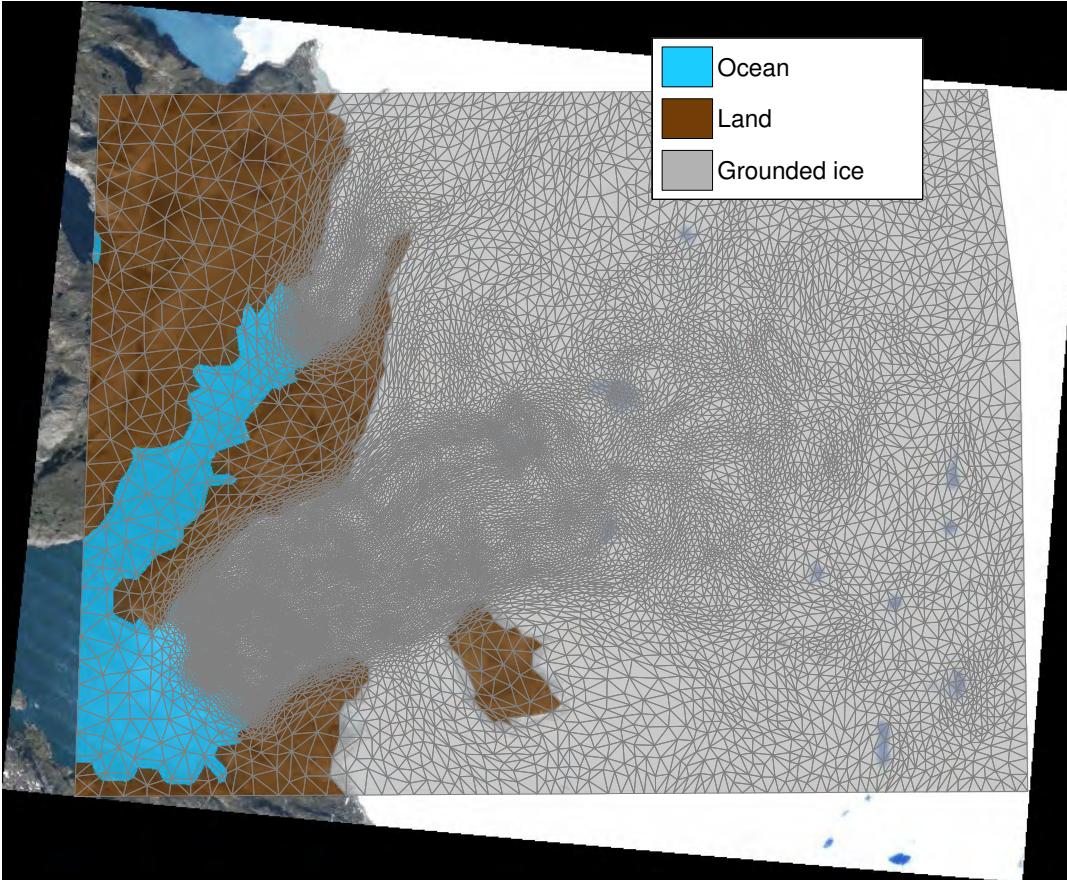
Store Glacier



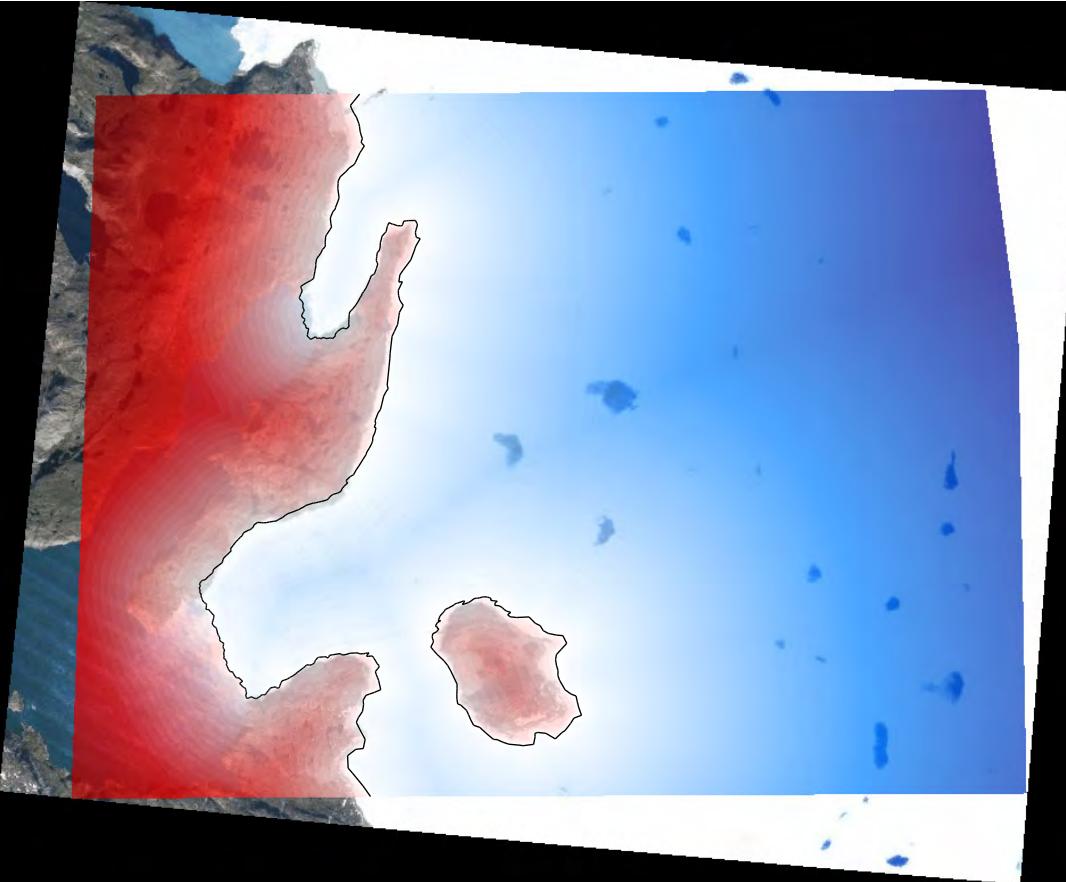
Store Glacier



Store Glacier



Store Glacier



Level set Method

- Tracks model boundary implicitly

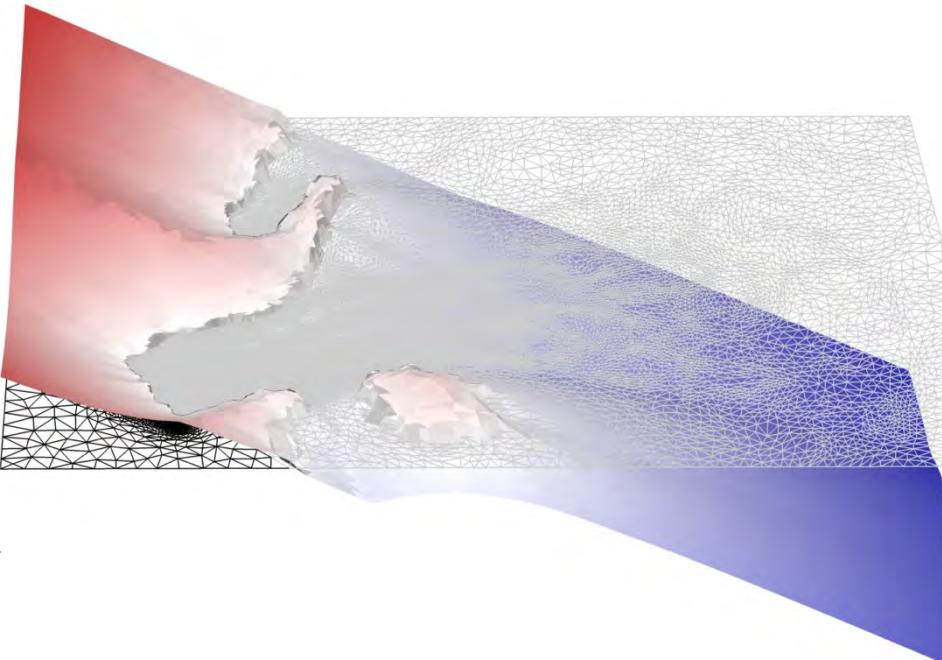
$$\begin{cases} \phi(x) < 0 & \text{If } x \text{ is on the ice} \\ \phi(x) > 0 & \text{If } x \text{ is not on the ice} \\ \phi(x) = 0 & \text{If } x \text{ is on the ice boundary} \end{cases}$$

- Ice front moves with a speed v_{front}

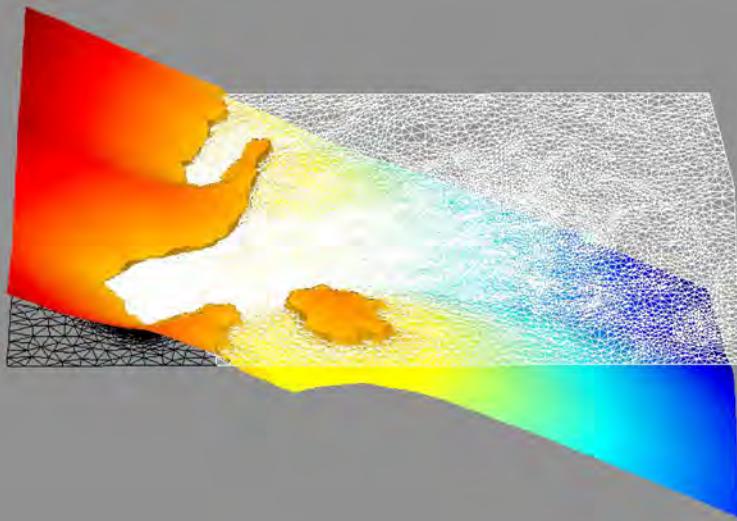
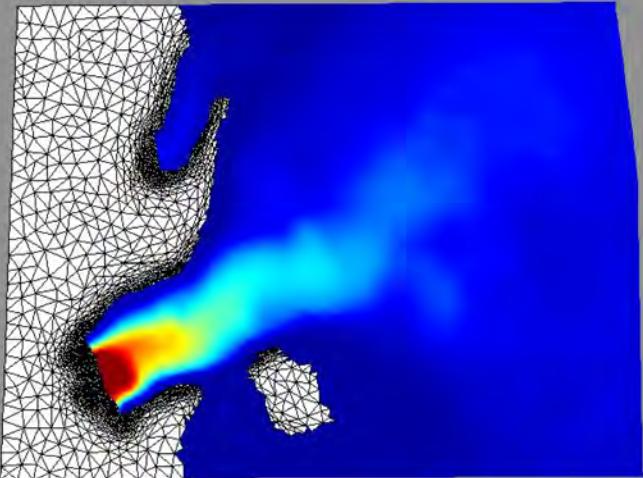
$$\mathbf{v}_{\text{front}} = \mathbf{v} - (c + \dot{M}) \mathbf{n}$$

- Level set equation

$$\frac{\partial \phi}{\partial t} = -\mathbf{v}_{\text{front}} \cdot \nabla \phi$$



Level set Method $c = 0, M = 10 \text{ km/yr}$



Example of ice front dynamics parameterization

$$\mathbf{v}_{\text{front}} = \mathbf{v} - \left(c + \dot{M} \right) \mathbf{n}$$

- Calving law (e.g., Morlighem et al. 2016)

$$c = \|\mathbf{v}\| \frac{\tilde{\sigma}}{\sigma_{\max}}$$

- Tensile von-Mises stress

$$\tilde{\sigma} = \sqrt{3} B \tilde{\varepsilon}_e^{1/n}$$

- Threshold: σ_{max} calibrated

- Melting rate (Rignot et al. 2016)

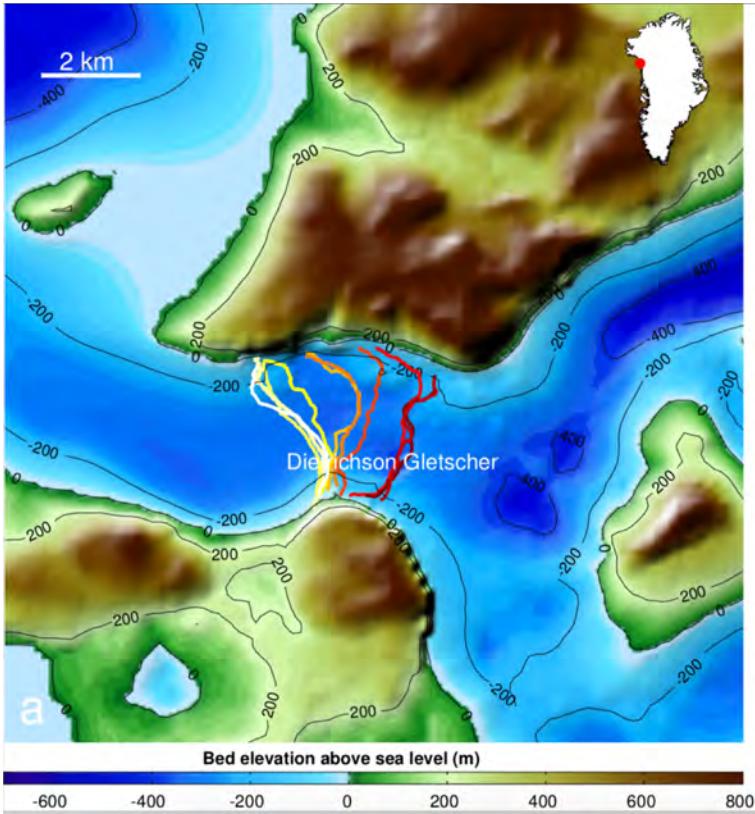
$$\dot{M} = (A h q_{sg}^\alpha + B) \text{TF}^\beta$$

- A : 3e-4 units B : 0.15 units
- α : 0.39 β : 1.18
- h : water depth (BedMachine v3)
- q_{sg} : subglacial water flux (RACMO)
- TF: thermal forcing (ECCO2)

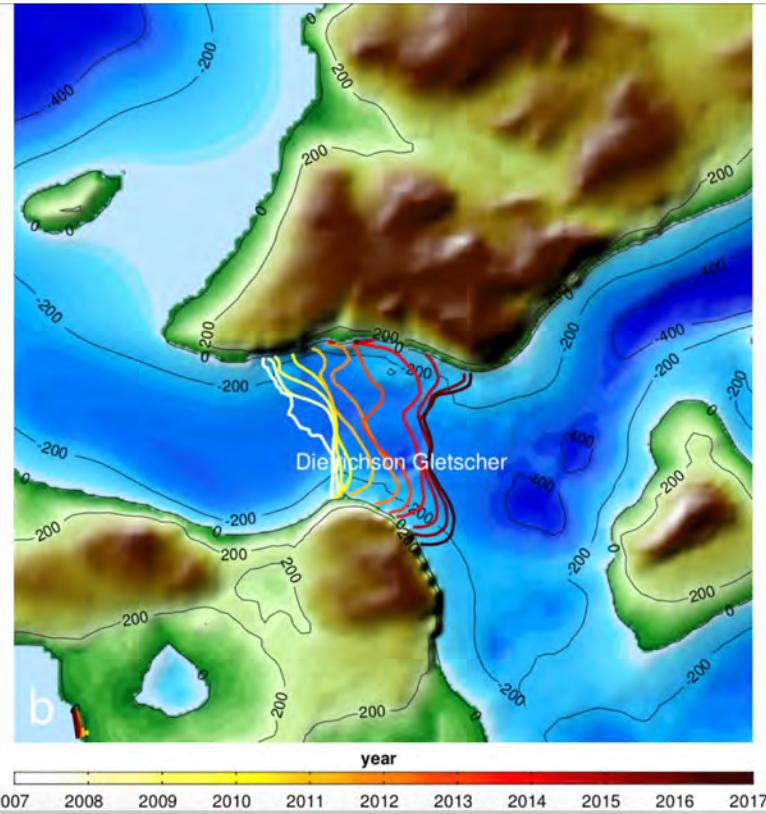


Threshold calibration

Observed ice front positions

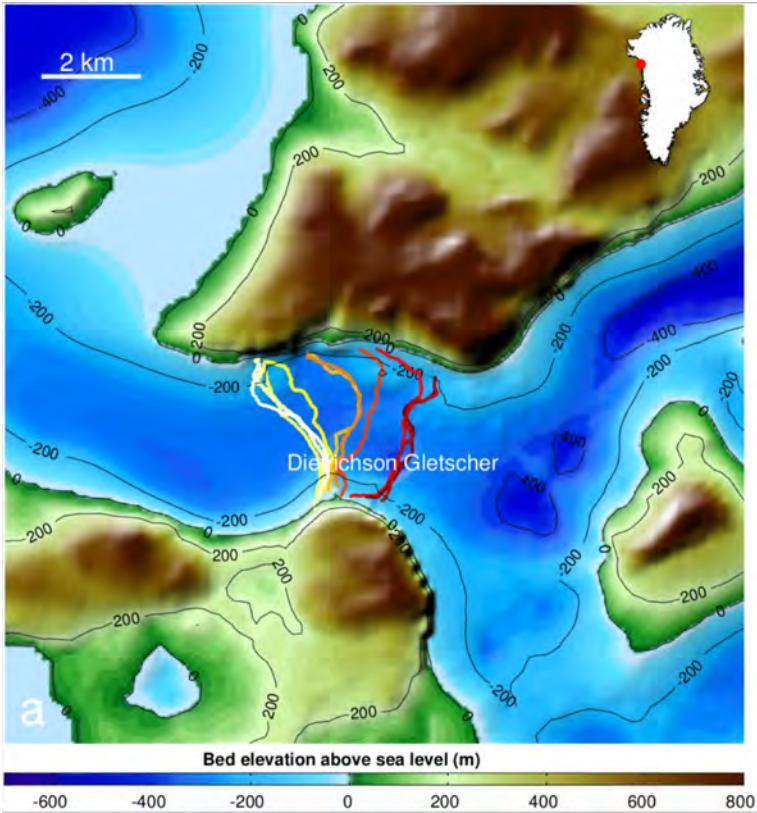


Modeled ice front positions

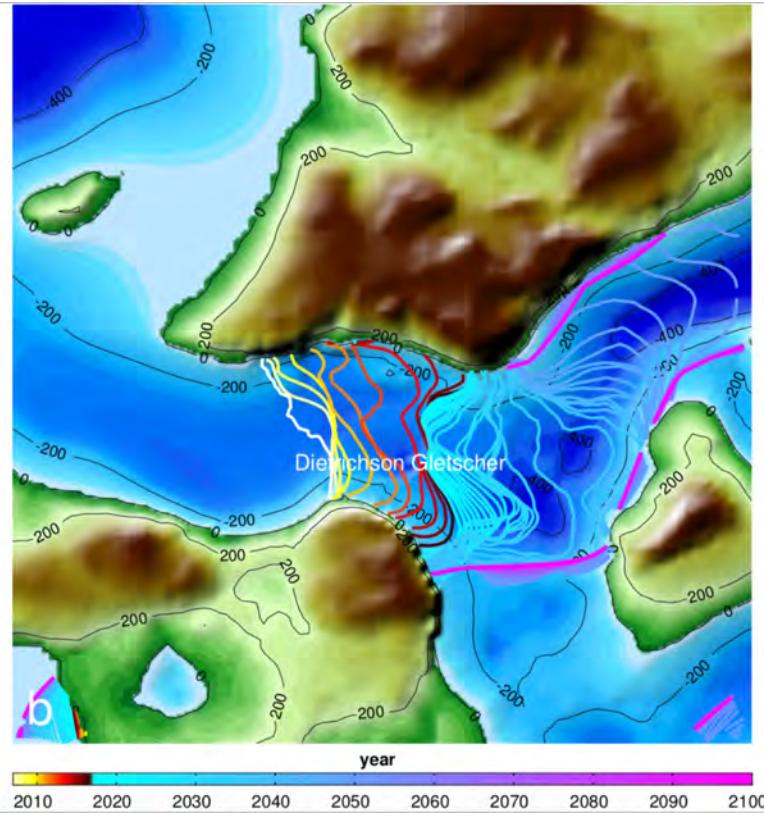


Threshold calibration

Observed ice front positions

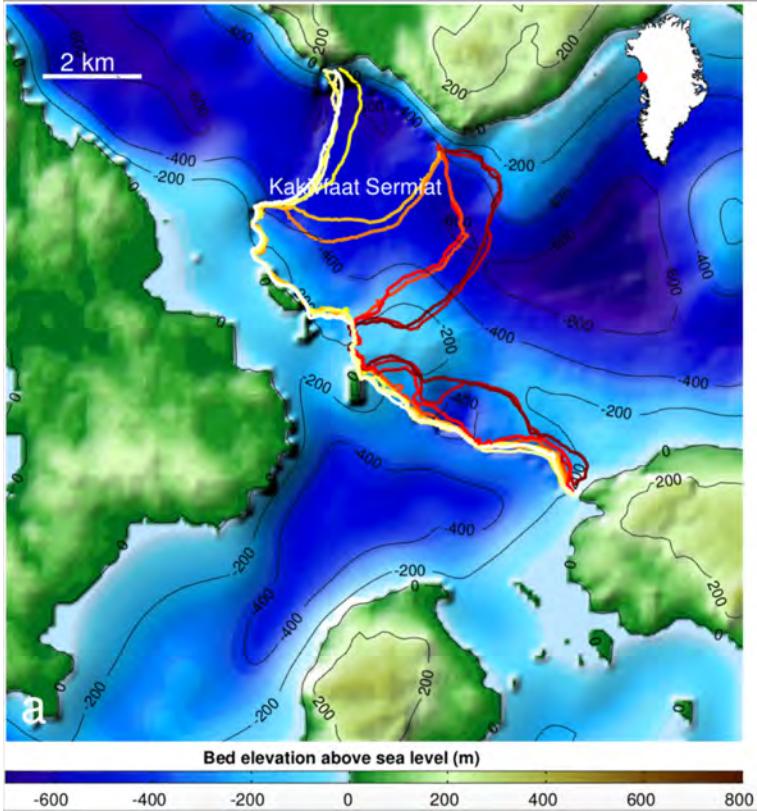


Modeled ice front positions

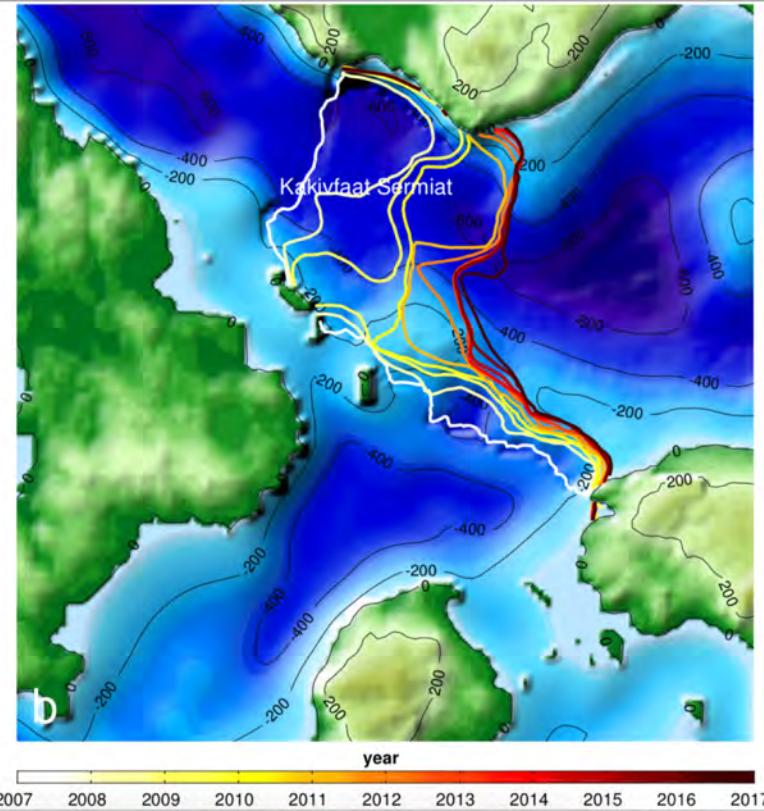


Threshold calibration

Observed ice front positions

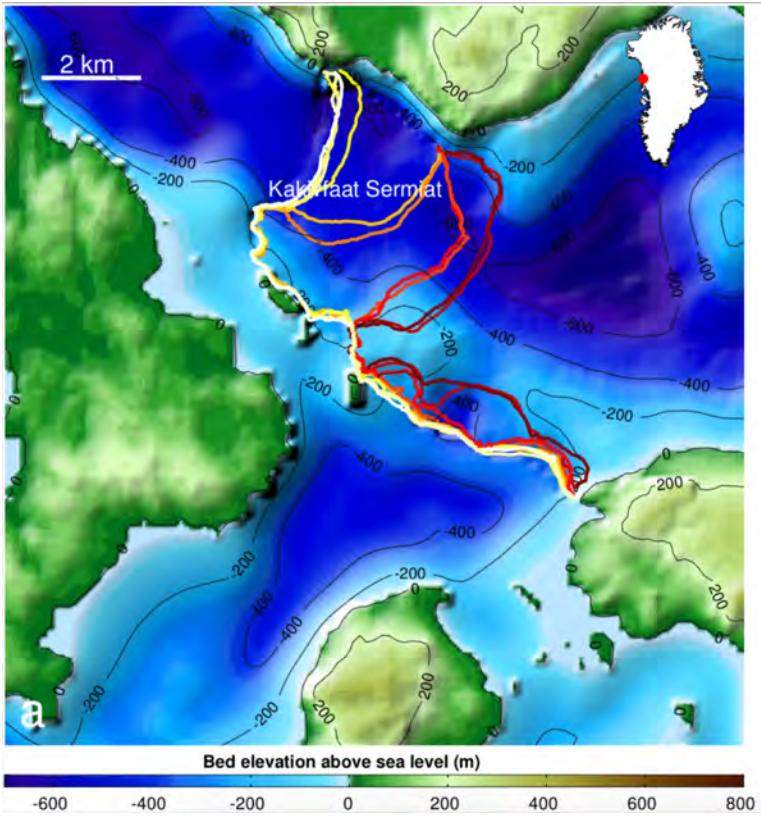


Modeled ice front positions

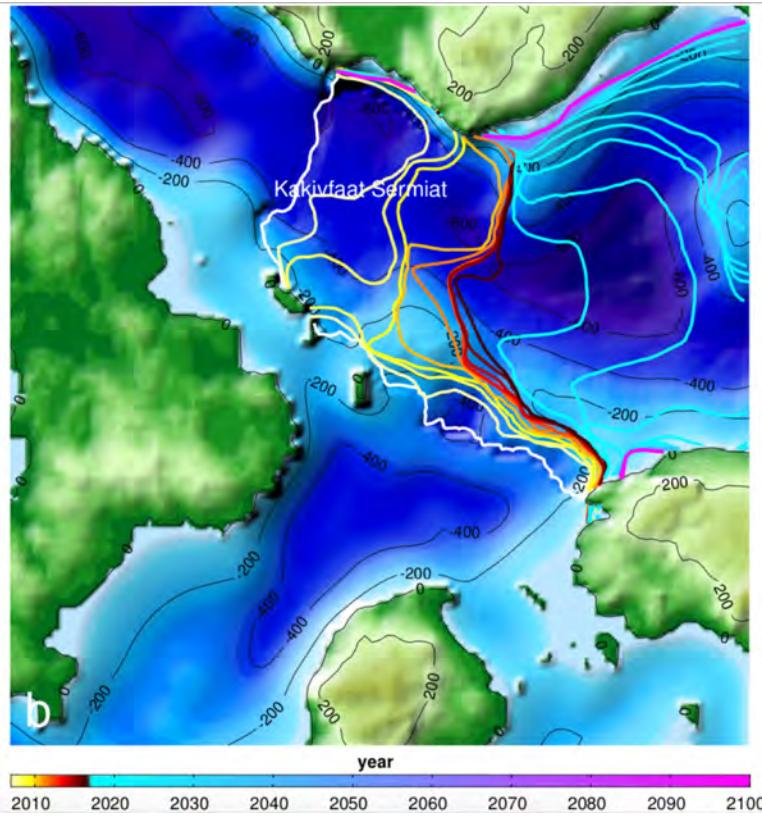


Threshold calibration

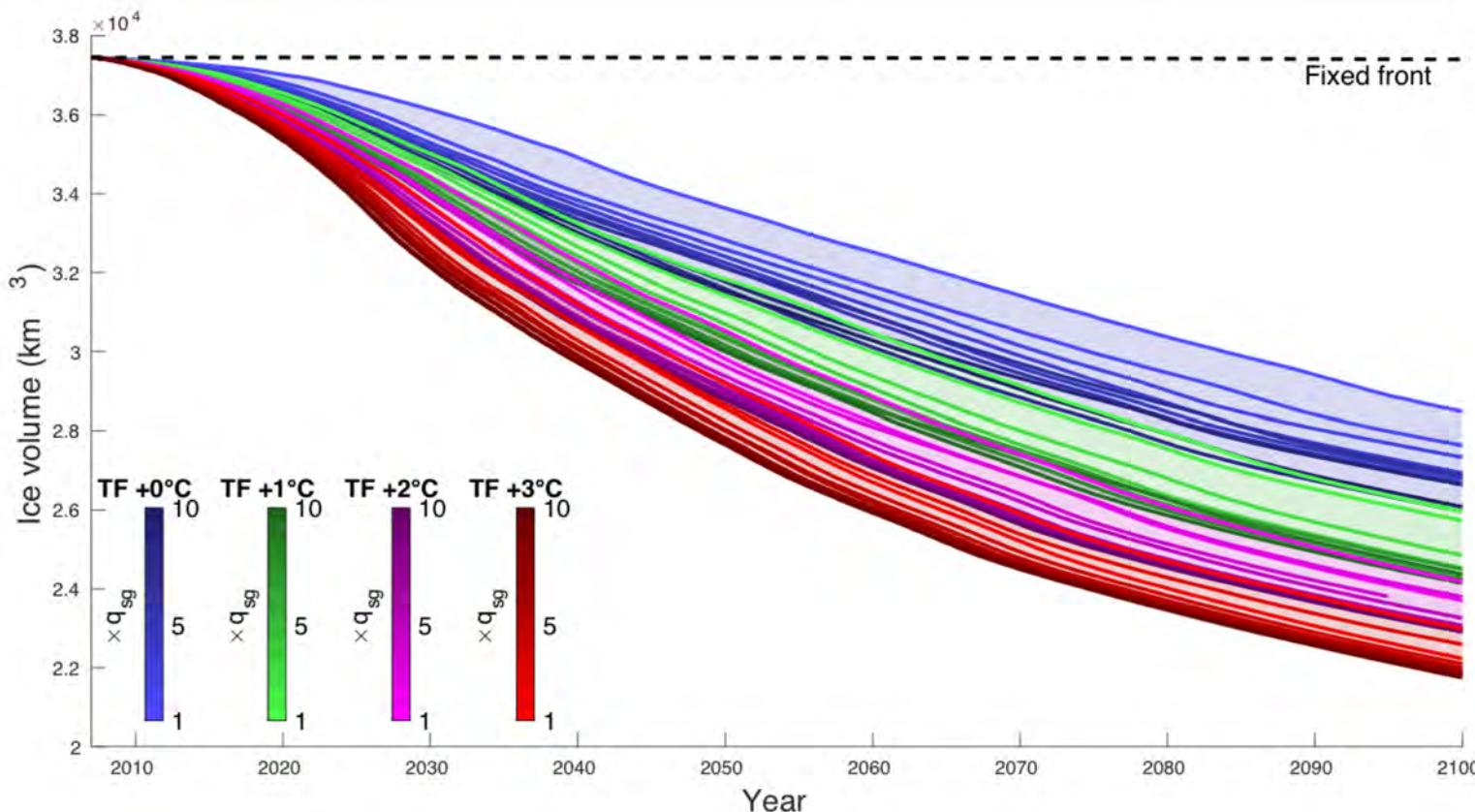
Observed ice front positions



Modeled ice front positions

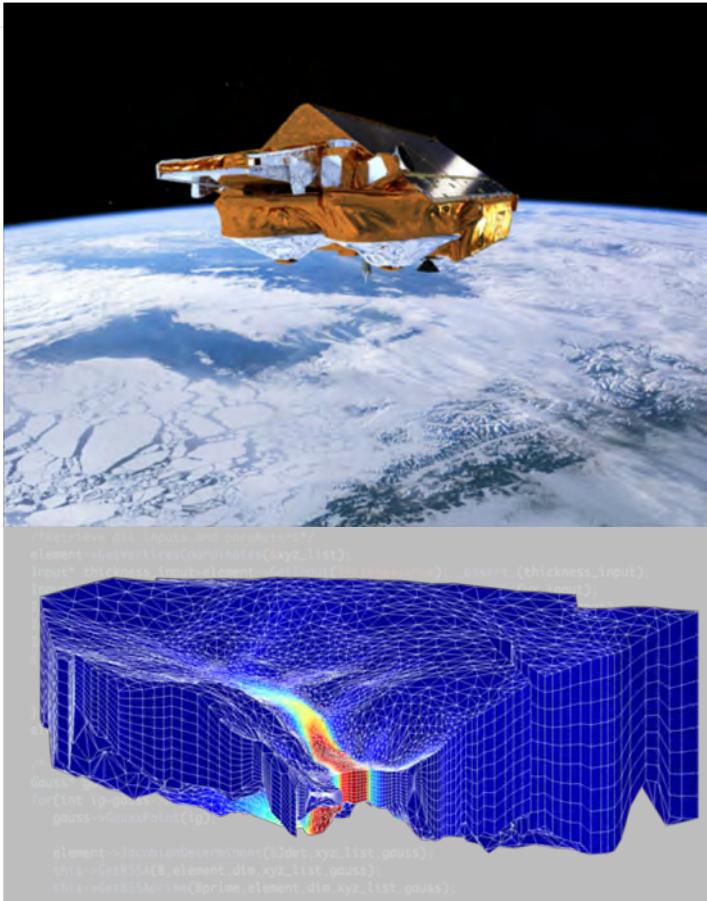


Ice volumes



Outline

1. Physics of ice sheet flow
 - Momentum balance and possible approximations
 - Key physical processes:
 - Grounding line dynamics
 - Calving dynamics
2. Numerical strategies
 - Mesh refinement and resolution of PDEs
 - Verification and Validation
 - Dealing with moving boundaries
3. Project management
 - Code development and architecture
 - Regression tests
 - Project management web interface



Architecture: some considerations

What Architecture?

- MPI (not just OpenMP)
- Hybrid (MPI/OpenMP)
- As modular as possible

What Language?

- **MATLAB/python/...** simple user interface
- **Fortran:** widely used in climate models, supports source-to-source (AD: OpenAD, Tapenade, TAF, etc)
- **C:** supports pointer, dynamic memory allocation, no (good) AD tool
- **C++:** same as C but additionally:
 - supports Object-Oriented programming
 - supports Object-Overloading for AD (CoDiPack, Sacado, ADOLC, etc)

Relying on external packages?

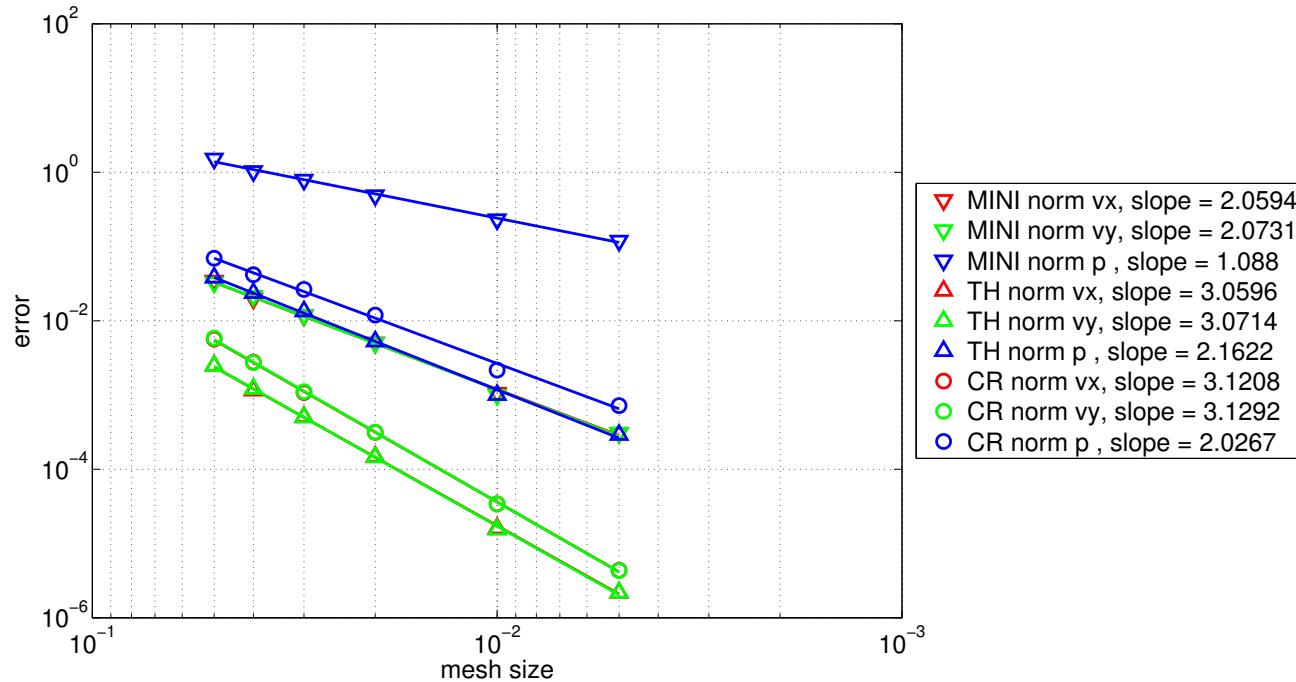
- The fewer the better
- in-house is always better

Team development & Coding:

- **Let scientists code!**
- Consistent coding style/clear rules
- **Open source** (svn/git)

Verification & Validation

- Inter-comparison: test against other models (ISMIP-HOM, MISMIP,...)
- Verification: test against analytical solutions
- Validation: test against “reality”



Automatic Regression tests

Ice Sheet System Model

Jenkins > All >

ENABLE AUTO REFRESH

People

Build History

Project Relationship

Check File Fingerprint

Credentials

Build Queue
No builds in the queue.

Build Executor Status

| S | W | Name ↓ | Last Success | Last Failure | Last Duration |
|---|---|---|---------------------|-----------------|---------------|
| | | Linux_Ubuntu | 7 hr 17 min - #3068 | N/A | 13 min |
| | | Linux_Ubuntu_ADOLC | 7 hr 4 min - #2885 | N/A | 8 min 51 sec |
| | | Linux_Ubuntu_Binaries | 17 days - #73 | 18 days - #71 | 1 hr 8 min |
| | | Linux_Ubuntu_CoDiPack | 7 hr 4 min - #115 | 14 days - #103 | 6 min 24 sec |
| | | Linux_Ubuntu_Compilation | 7 hr 14 min - #619 | N/A | 1.1 sec |
| | | Linux_Ubuntu_Dakota | 7 hr 4 min - #2028 | N/A | 14 min |
| | | Linux_Ubuntu_Externalpackages | 7 hr 15 min - #613 | 17 days - #598 | 1.2 sec |
| | | Linux_Ubuntu_IceOcean | 7 hr 4 min - #2253 | N/A | 15 min |
| | | Linux_Ubuntu_Javascript | 7 hr 16 min - #1366 | N/A | 3 min 19 sec |
| | | Linux_Ubuntu_Python | 7 hr 13 min - #1713 | N/A | 9 min 52 sec |
| | | Linux_Ubuntu_SolidEarth | 7 hr 4 min - #2439 | N/A | 15 min |
| | | Linux_Ubuntu SVN | 7 hr 15 min - #654 | N/A | 21 sec |
| | | Linux_Ubuntu_trunk | 13 days - #59 | N/A | 58 min |
| | | macOS | 7 hr 13 min - #2453 | 13 days - #2442 | 6 min 31 sec |
| | | macOS_binaries | 17 days - #116 | 17 days - #115 | 46 min |
| | | macOS_Dakota | 7 hr 6 min - #2176 | N/A | 24 min |
| | | macOS_Dakota_Binaries | 14 days - #39 | 15 days - #38 | 1 hr 18 min |
| | | macOS_Examples | 7 hr 6 min - #1717 | N/A | 10 min |

Project management website (Trac)

ISSM issm.ess.uci.edu/trac/issm/changeset/23293

Changeset 23293

Timestamp: Sep 16, 2018, 7:51:25 PM (2 months ago)
Author: morlighé

Message: BUG: fixing memory leaks

Location: issm/trunk-jpl/src/c
Files: 3 edited

cores/controladm1qn3_core.cpp (3 diffs)
modules/ModelProcessors/Control/UpdateElementsAndMaterialsControl.cpp (1 diff)
modules/SetControlInputsFromVectorx/SetControlInputsFromVectorx.cpp (1 diff)

Unmodified Added Removed Modified

issm/trunk-jpl/src/c/cores/controladm1qn3_core.cpp

Revision 23291

```

92 int num_dependents_old = num_dependents;
93 int num_independents_old = num_independents;

324 if(my_rank==0){
325     num_dependents = 0;
326     num_independents = 0;
327 }

328 #if defined(_HAVE_ADOOLC_)
329 /*Get gradient for ADOOLC {{*/
330 if(my_rank==0){
331     num_dependents = 0;
332     num_independents = 0;
333 }

334 /*get the EDF pointer:*/
335 ...
336 /*initialize direction index in the weights vector: */
337 aWeightVector=xNewZeroInit<IssmPDouble>(num_dependents);
338 if(my_rank==0) tape_codi.setGradient(codi_global.output_indices[aDepIndex],1.0);

339 tape_codi.evaluate();
340

```

Revision 23293

```

322 int num_dependents_old = num_dependents;
323 int num_independents_old = num_independents;

324 #if defined(_HAVE_ADOOLC_)
325 /*Get gradient for ADOOLC {{*/
326 if(my_rank==0){
327     num_dependents = 0;
328     num_independents = 0;
329 }

330 /*get the EDF pointer:*/
331 ...
332 /*initialize direction index in the weights vector: */
333 aWeightVector=xNewZeroInit<IssmPDouble>(num_dependents);
334 if(my_rank==0){
335     if(aDepIndex<0 || aDepIndex>=num_dependents || codi_global.output_indices.size() <= aDepIndex){
336         _error_("index value for AutodiffForaReverseIndexEnum should be in [0,num_dependents-1]");
337     }
338     tape_codi.setGradient(codi_global.output_indices[aDepIndex],1.0);
339 }

340 tape_codi.evaluate();

```

Conclusions

- Ice physics:
 - Do not rely on SIA
 - Should not focus on FS only
 - Model should be capable of capturing Ice front and Grounding line dynamics (at the minimum)
- Numerics:
 - FE or FV, do not use FD
 - Static mesh adaptation, or AMR
 - Handle Moving boundary
 - Check against existing benchmarks
- Code management:
 - Good project management platform
 - Open source
 - Let scientists code!



Thank you ! Questions ?

