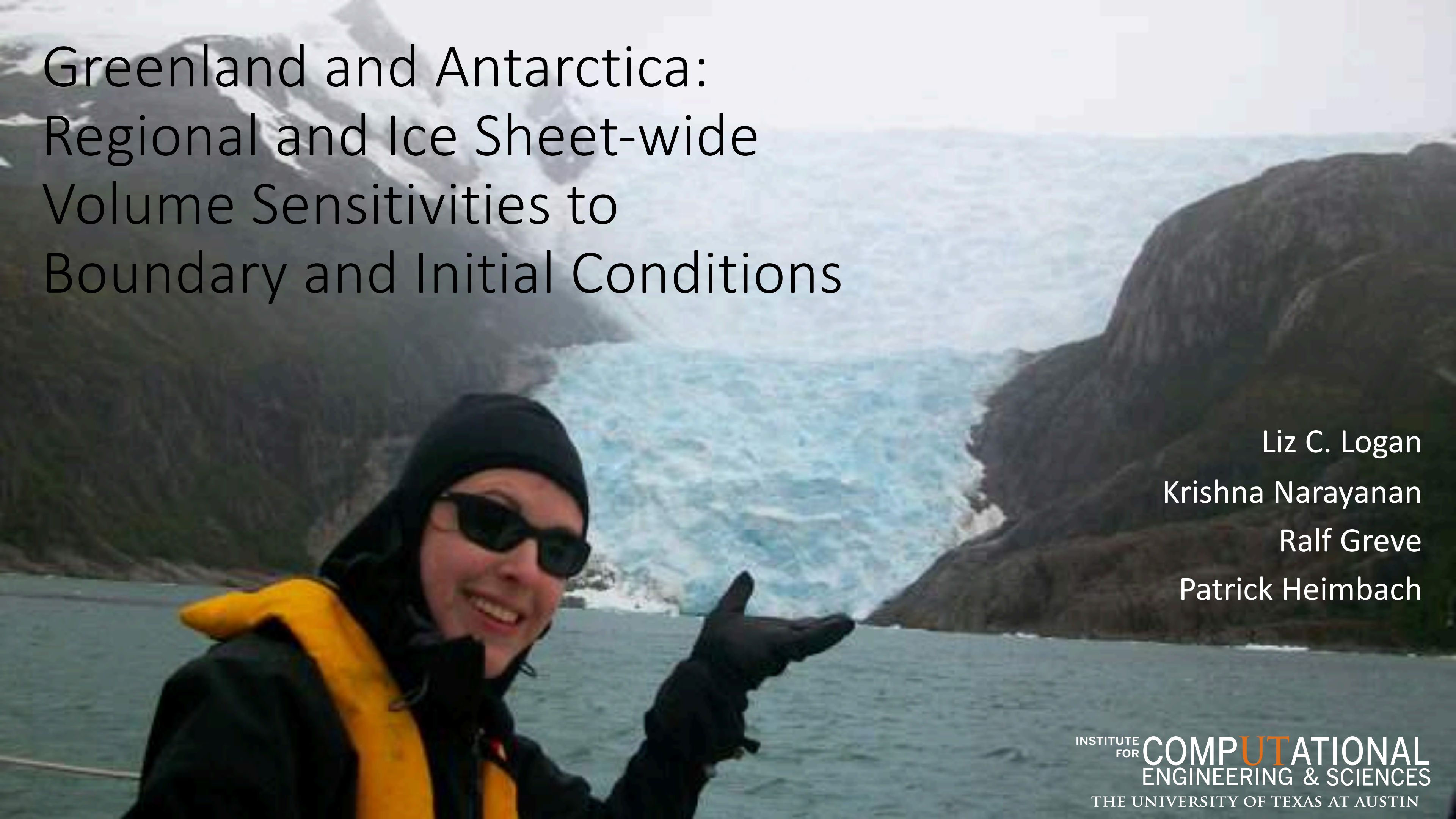
Greenland and Antarctica: Regional and Ice Sheet-wide Volume Sensitivities to Boundary and Initial Conditions



Liz C. Logan
Krishna Narayanan
Ralf Greve
Patrick Heimbach

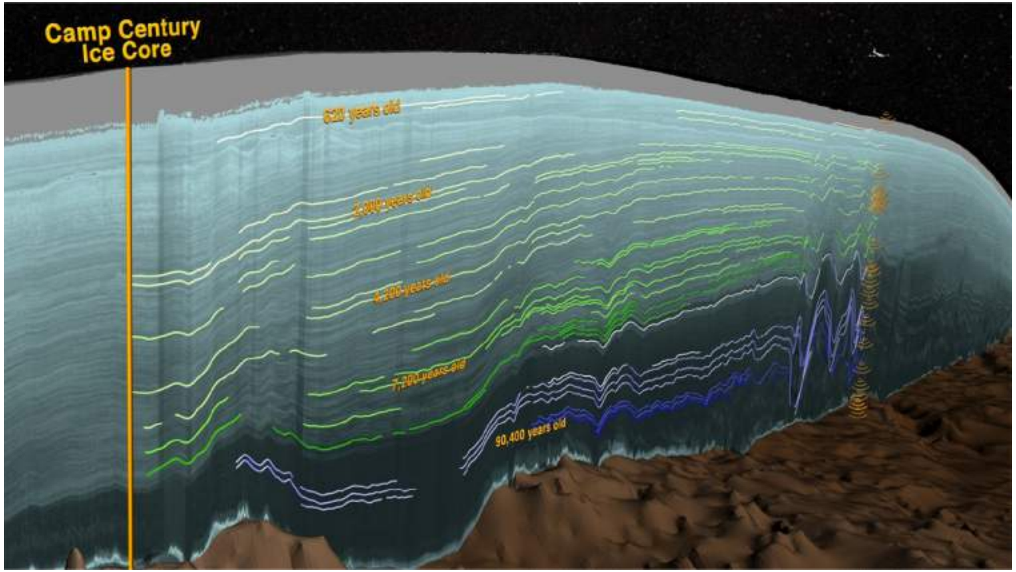
INSTITUTE FOR COMPUTATIONAL ENGINEERING & SCIENCES
THE UNIVERSITY OF TEXAS AT AUSTIN

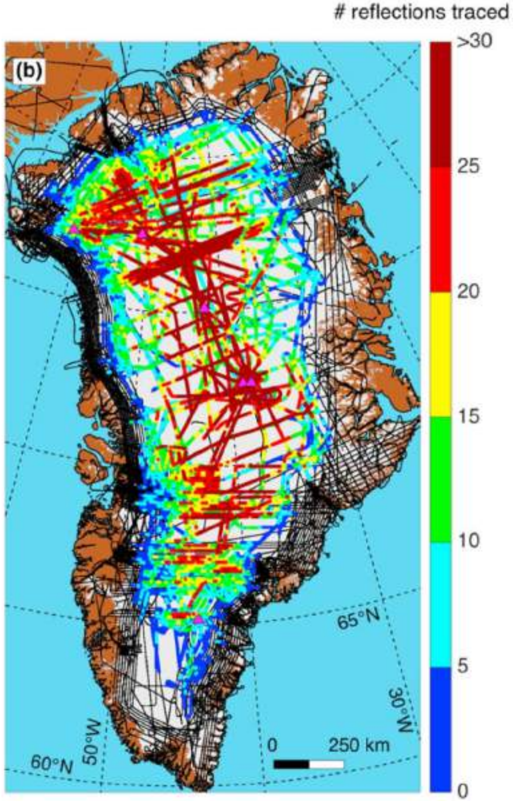
Road map:

1. Motivating problem: Greenland Ice Sheet state / parameter estimate
2. Development of ice sheet adjoint model
3. Sensitivity exploration

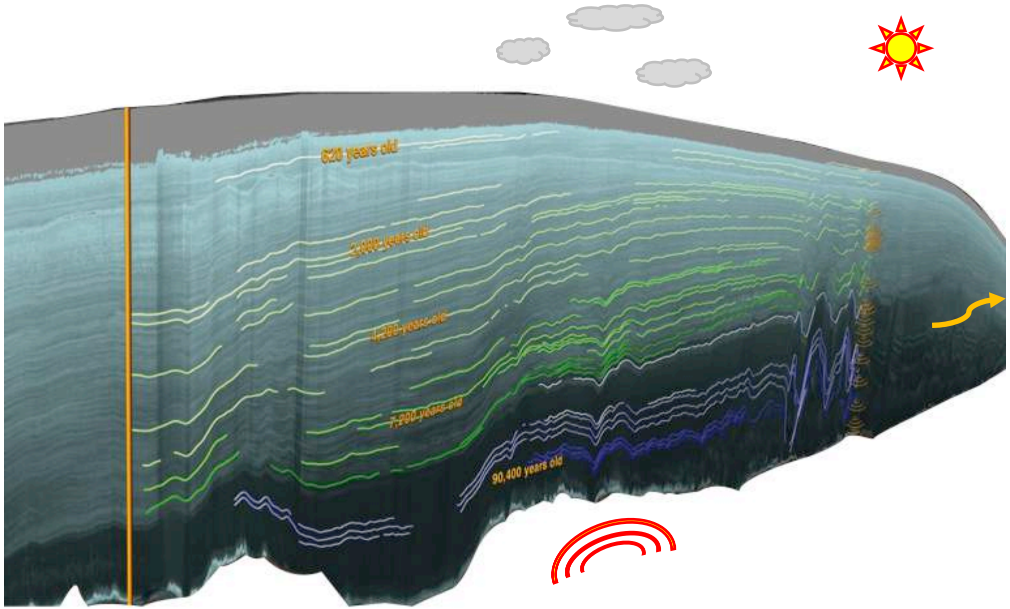
1. Motivating problem: Greenland state estimate

History of the ice sheet?





1. Motivating problem: Greenland state estimate



2. Development of SICOPOLIS adjoint

Ice-sheet model SICOPOLIS

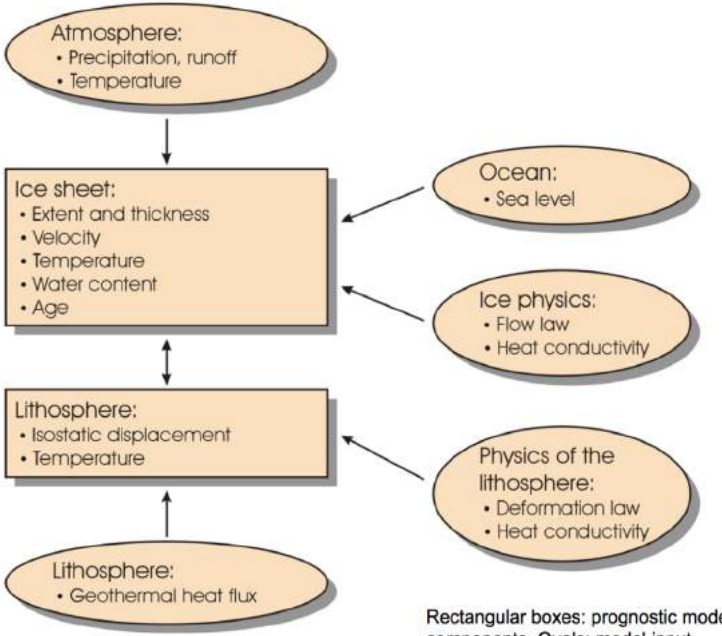
HAL 9000's iconic camera eye

Ralf Greve

Institute of Low Temperature Science,
Hokkaido University, Sapporo, Japan

Presentation held at the *Workshop on Ice Flow Modelling*,
Niels Bohr Institute for Astronomy, Physics and Geophysics, University of Copenhagen, Denmark, 2007-05-09.

SICOPOLIS



Rectangular boxes: prognostic model components. Ovals: model input.

2. Development of SICOPOLIS adjoint

Field equations (SIA)

- Non-linear viscous flow law (Glen):

$$\mathrm{D} = EA(T',\omega)\, f(\sigma)\, \mathrm{t}^{\mathrm{D}}, \qquad f(\sigma) = \sigma^{n-1}, \quad n = 3$$

- Stresses:

$$p = \rho g(h - z), \quad (t_{xz}, t_{yz}) = -p\,\nabla h, \quad \sigma = \sqrt{t_{xz}^2 + t_{yz}^2}$$

- Horizontal velocity:

$$\mathbf{v}_\mathrm{h} = \mathbf{v}_\mathrm{b} - \left(2(\rho g)^n |\nabla h|^{n-1} \int_b^z A(T',\omega)\,(h - z')^n\,\mathrm{d}z' \right) \times \nabla h$$

- Ice thickness equation:

$$\frac{\partial H}{\partial t} = -\nabla \cdot \mathbf{q} + a_\mathrm{s} - a_\mathrm{b} \quad \Rightarrow \quad \frac{\partial h}{\partial t} = \nabla \cdot (D \nabla h) + a_\mathrm{s} - a_\mathrm{b} + \frac{\partial b}{\partial t}$$

- Temperature equation (cold ice):

$$\frac{\partial T}{\partial t} + \mathbf{v} \cdot \mathrm{grad}\, T = \frac{1}{\rho c} \frac{\partial}{\partial z} \left(\kappa \frac{\partial T}{\partial z} \right) + \frac{2}{\rho c} EA(T')\, \sigma^{n+1}$$

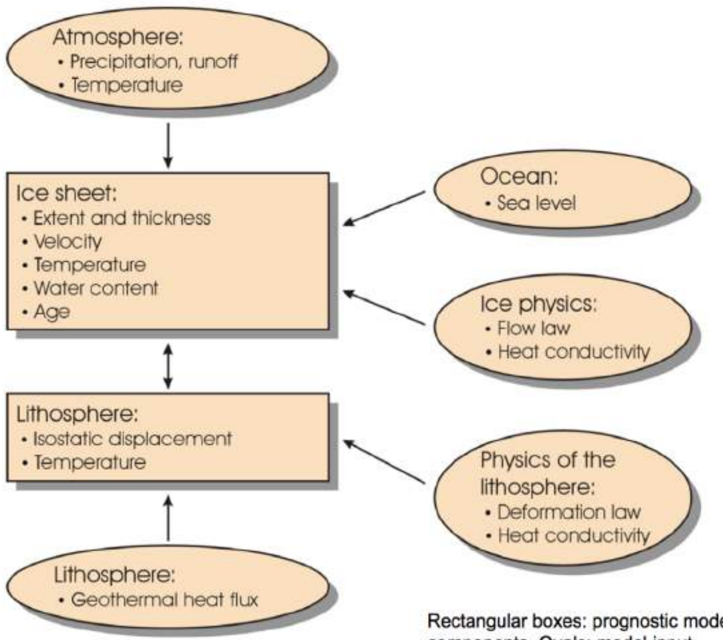
- Water-content equation (temperate ice):

$$\frac{\partial \omega}{\partial t} + \mathbf{v} \cdot \mathrm{grad}\, \omega = \frac{2}{\rho L} EA(\omega)\, \sigma^{n+1} - D(\omega) + \mathrm{CCC}$$

- Age:

$$\frac{\partial A}{\partial t} + \mathbf{v} \cdot \mathrm{grad}\, A = 1$$

SICOPOLIS

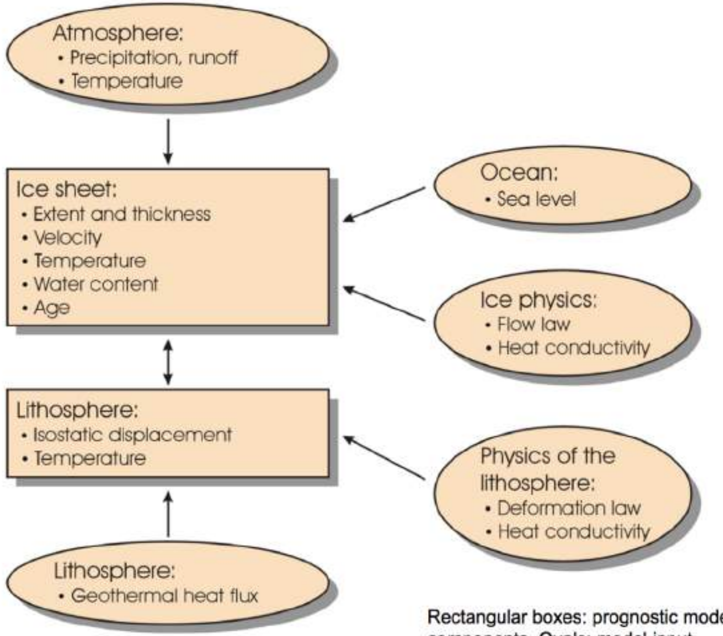


Rectangular boxes: prognostic model components. Ovals: model input.

2. Development of SICOPOLIS adjoint

I. Greenland
II. Antarctica
III. Laurentide
IV. Fennoscandia
V. Austfanna
VI. Tibet
VII. Mars!

SICOPOLIS



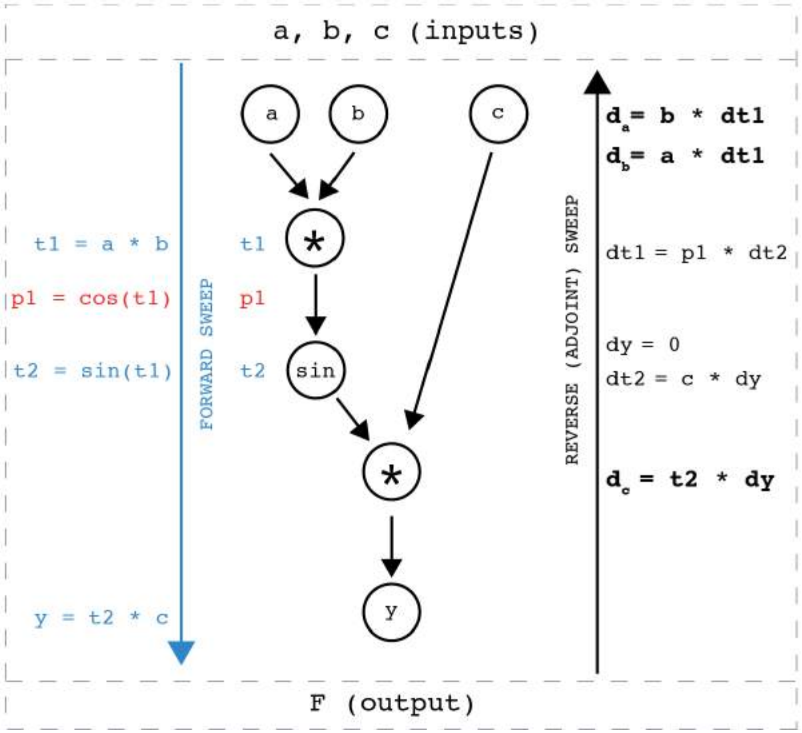
Rectangular boxes: prognostic model components. Ovals: model input.

2. Development of SICOPOLIS adjoint

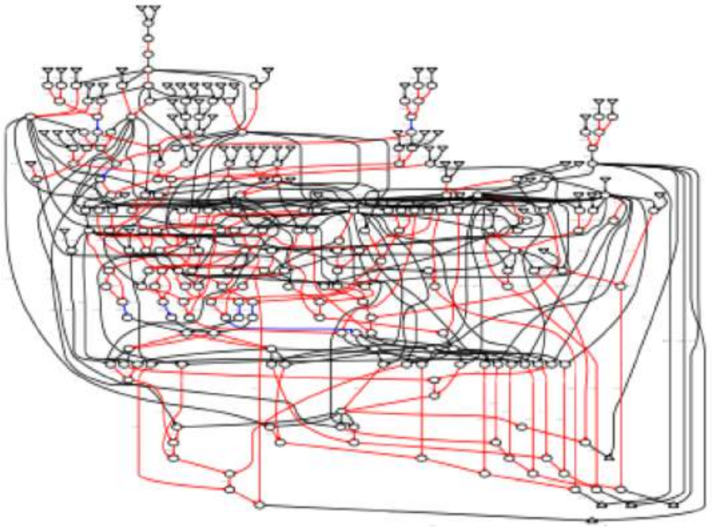
```fortran
!   ——— d_help_b, c_drag

    if (n_cts(j,i) == -1_i1b) then   ! cold ice base

       if (sub_melt_flag(j,i)) then
          temp_diff = max((temp_c_m(0,j,i)-temp_c(0,j,i)), 0.0_dp)
          cvxy1a    = exp(-gamma_slide_inv(j,i)*temp_diff)  ! sub-melt sliding
          ctau1a    = 1.0_dp/(cvxy1a+eps_dp)**p_weert_inv(j,i)
       else
          cvxy1a    = 0.0_dp    ! no sub-melt sliding
          ctau1a    = 1.0_dp/eps_dp**p_weert_inv(j,i)    ! dummy value
       end if

       d_help_b(j,i) = cvxy1*cvxy1a
       c_drag(j,i)   = ctau1*ctau1a

    else if (n_cts(j,i) == 0_i1b) then   ! temperate ice base

       d_help_b(j,i) = cvxy1   ! basal sliding
       c_drag(j,i)   = ctau1   ! (pressure-melting conditions)

    else   ! n_cts(j,i) == 1_i1b, temperate ice layer

       d_help_b(j,i) = cvxy1   ! basal sliding
       c_drag(j,i)   = ctau1   ! (pressure-melting conditions)

    end if

!   ——— Contribution of the basal water layer

#if (BASAL_HYDROLOGY==1)

#if (HYDRO_SLIDE_SAT_FCT==0)

    ratio_Hw_slide = max(H_w(j,i)*Hw0_slide_inv, 0.0_dp)
                     ! constrain to interval [0,infty)

    cvxy1b = 1.0_dp + c_Hw_slide*(1.0_dp-exp(-ratio_Hw_slide))
                     ! exponential saturation function
                     ! by Kleiner and Humbert (2014, J. Glaciol. 60)

#elif (HYDRO_SLIDE_SAT_FCT==1)

    ratio_Hw_slide = max(min(H_w(j,i)*Hw0_slide_inv, 1.0_dp), 0.0_dp)
                     ! constrain to interval [0,1]

    cvxy1b = 1.0_dp + c_Hw_slide*ratio_Hw_slide
                     ! linear saturation function
```

F : y = sin (a * b) * c

F : = [d_a , d_b , d_c] ?

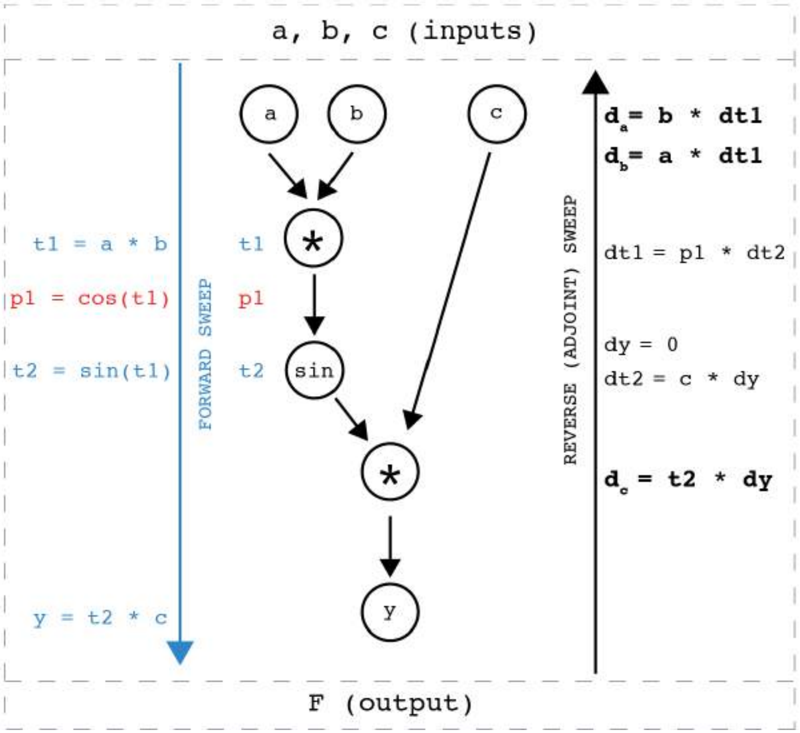


2. Development of SICOPOLIS adjoint



F : y = sin (a * b) * c

F : = [d_a , d_b , d_c] ?



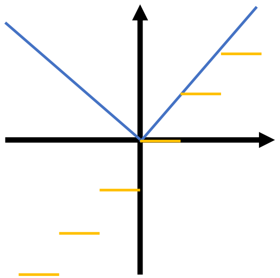
2. Development of SICOPOLIS adjoint

There can be problems, however:

- Non-smoothness

  abs, ceiling, and floor

  $\sqrt{(x)}$



- Unstructured code
- Solvers!
- Memory issues

2. Development of SICOPOLIS adjoint

There can be problems, however:

- Non-smoothness
- Unstructured code
- Solvers!
- Memory issues

```
#ifndef ALLOW_OPENAD /* Normal */

do kc=1, KCMAX-1
   if (omega_c_neu(kc,j,i) > eps_omega) then
      kc_cts_neu(j,i) = kc
   else
      exit
   end if
end do

#else /* OpenAD */

kcdone = .false.

do kc=1, KCMAX-1
   if (kcdone.eqv..false.) then
      if (omega_c_neu(kc,j,i) > eps_omega) then
         kc_cts_neu(j,i) = kc
      else
         kcdone = .true.
      end if
   end if
end do

#endif /* Normal vs. OpenAD */
```

2. Development of SICOPOLIS adjoint

There can be problems, however:

- Non-smoothness
- Unstructured code
- Solvers!
- Memory issues

$$A \cdot x = b \rightarrow x := solve(A, b)$$

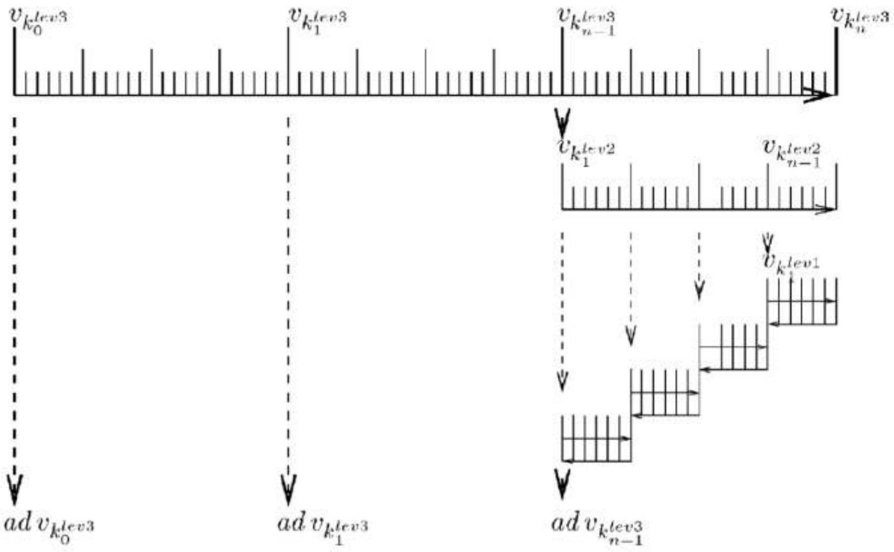
$$A^T \cdot \bar{b} = \bar{x} \rightarrow \bar{b} := solve(A^T, \bar{x})$$

$$\bar{A} := -x^T \cdot \bar{b}$$

2. Development of SICOPOLIS adjoint

There can be problems, however:

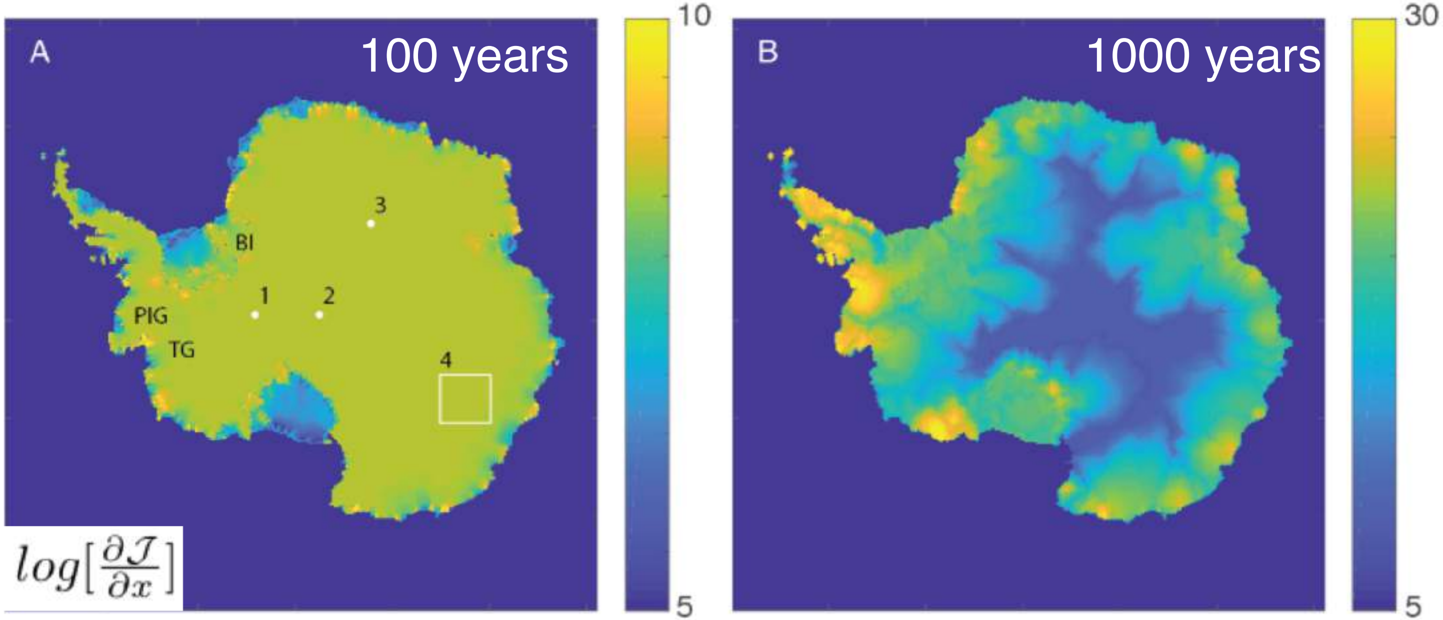
- Non-smoothness
- Unstructured code
- Solvers!
- Memory issues
    - Binomial checkpointing (not this →)



3. Adjoint model!

Looking at adjoint sensitivities

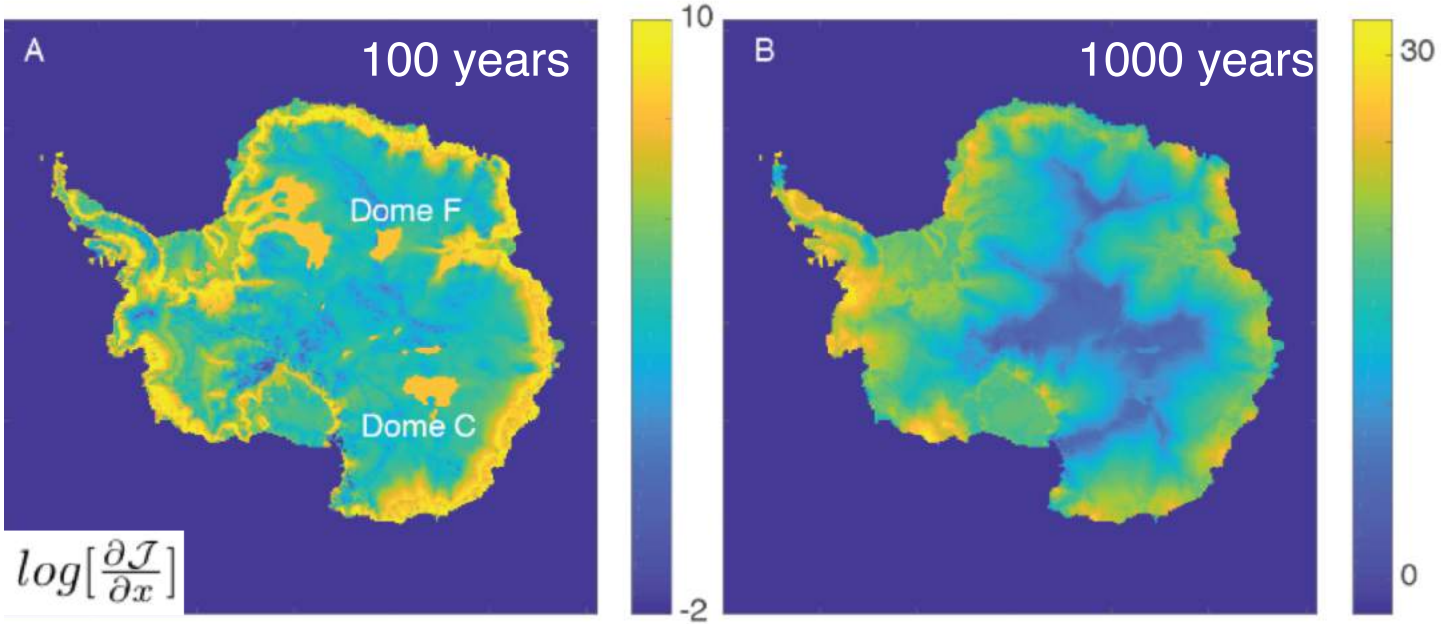
$$\frac{\partial \mathcal{J}}{\partial x} = \frac{\partial Volume}{\partial IceThickness}$$



3. Adjoint model!

Looking at adjoint sensitivities

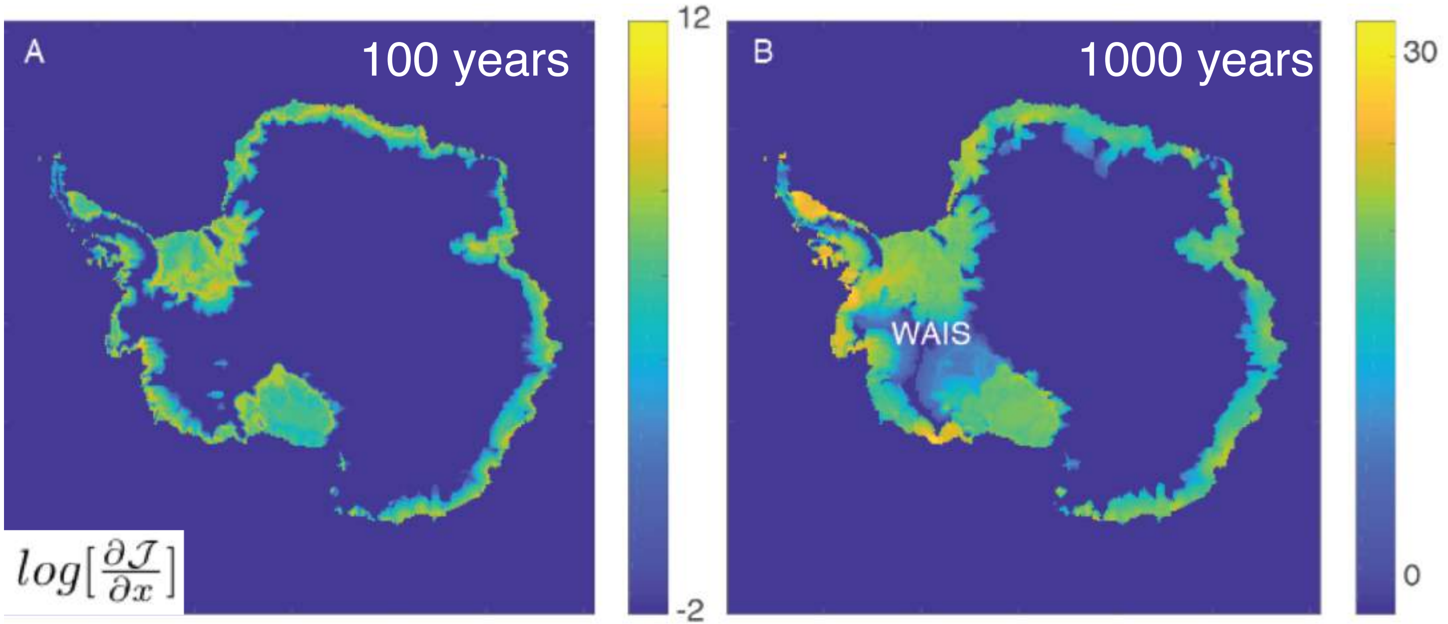
$$\frac{\partial \mathcal{J}}{\partial x} = \frac{\partial Volume}{\partial BasalTemperature}$$



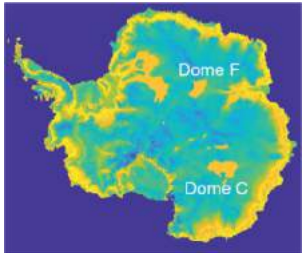
3. Adjoint model!

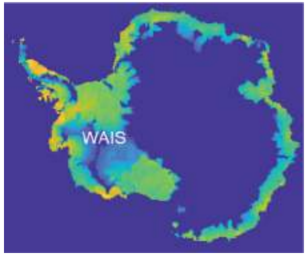
Looking at adjoint sensitivities

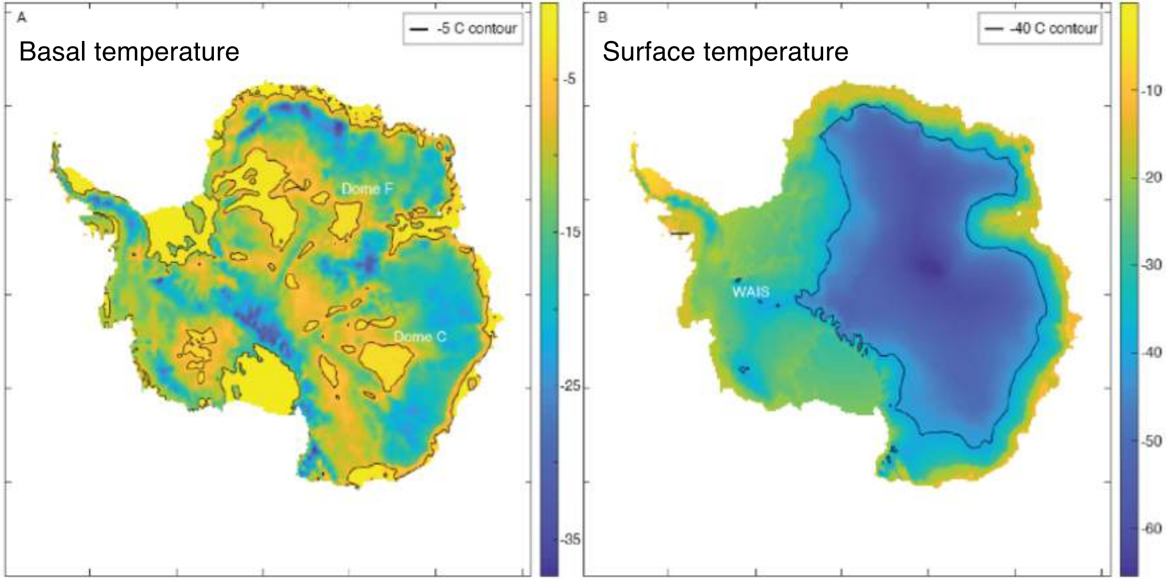
$$\frac{\partial \mathcal{J}}{\partial x} = \frac{\partial Volume}{\partial SurfaceTemperature}$$



3. Adjoint model!

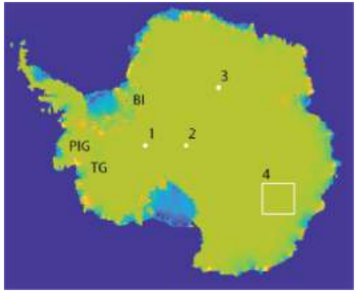






3. Adjoint model!

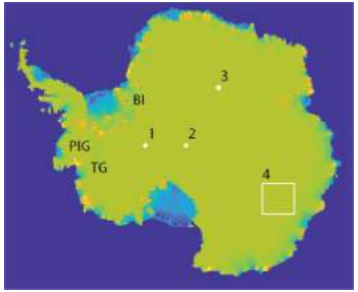
$$\frac{\Delta \mathcal{J}}{\Delta x_{\varepsilon}} = \frac{\mathcal{J}(x+\varepsilon) - \mathcal{J}(x-\varepsilon)}{2\varepsilon} \approx \frac{\partial \mathcal{J}}{\partial x} \forall \varepsilon \ll \| x \|$$



Variable (1)	Region (from Fig. 1A) (2)	$\frac{\delta J}{\delta \text{Variable}}$ (3)	$\frac{\Delta J}{\Delta \text{Variable}}$ (4)	ΔV_{adj} (5)	ΔV_{fd} (6)	% Deviation (7)
ice thickness	2	3.99915×10^8	3.99916×10^8	–	–	2.51×10^{-3}
July precipitation	1	2.1053×10^{16}	2.0709×10^{16}	–	–	1.6
basal temperature	3	2.1527×10^9	1.5422×10^8	–	–	93
surface temperature	TG	-1.5016×10^6	-4.2937×10^7	–	–	2.7593×10^3
ice thickness	Greenland	-9.0837×10^6	-9.1133×10^6	–	–	3.2448×10^{-1}
ice thickness	4	–	–	1.6411×10^5	1.2626×10^7	7.59×10^3
July precipitation	4	–	–	8.6987×10^{23}	1.7337×10^{23}	80
basal temperature	4	–	–	-1.1493×10^9	-1.5200×10^9	32

3. Adjoint model!

$$\frac{\Delta \mathcal{J}}{\Delta x_\varepsilon} = \frac{\mathcal{J}(x+\varepsilon) - \mathcal{J}(x-\varepsilon)}{2\varepsilon} \approx \frac{\partial \mathcal{J}}{\partial x} \forall \varepsilon \ll \| x \|$$



Variable (1)	Region (from Fig. 1A) (2)	$\frac{\delta J}{\delta \text{Variable}}$ (3)	$\frac{\Delta J}{\Delta \text{Variable}}$ (4)	ΔV_{adj} (5)	ΔV_{fd} (6)	% Deviation (7)
ice thickness	2	3.99915×10^8	3.99916×10^8	–	–	2.51×10^{-3}
July precipitation	1	2.1053×10^{16}	2.0709×10^{16}	–	–	1.6
basal temperature	3	2.1527×10^9	1.5422×10^8	–	–	93
surface temperature	TG	-1.5016×10^6	-4.2937×10^7	–	–	2.7593×10^3
ice thickness	Greenland	-9.0837×10^6	-9.1133×10^6	–	–	3.2448×10^{-1}
ice thickness	4	–	–	1.6411×10^5	1.2626×10^7	7.59×10^3
July precipitation	4	–	–	8.6987×10^{23}	1.7337×10^{23}	80
basal temperature	4	–	–	-1.1493×10^9	-1.5200×10^9	32

3. Adjoint model!

$$\frac{\Delta\mathcal{J}}{\Delta x_{\varepsilon}} = \frac{\mathcal{J}(x+\varepsilon)-\mathcal{J}(x-\varepsilon)}{2\varepsilon} \approx \frac{\partial\mathcal{J}}{\partial x} \forall\varepsilon \ll \| x \|$$

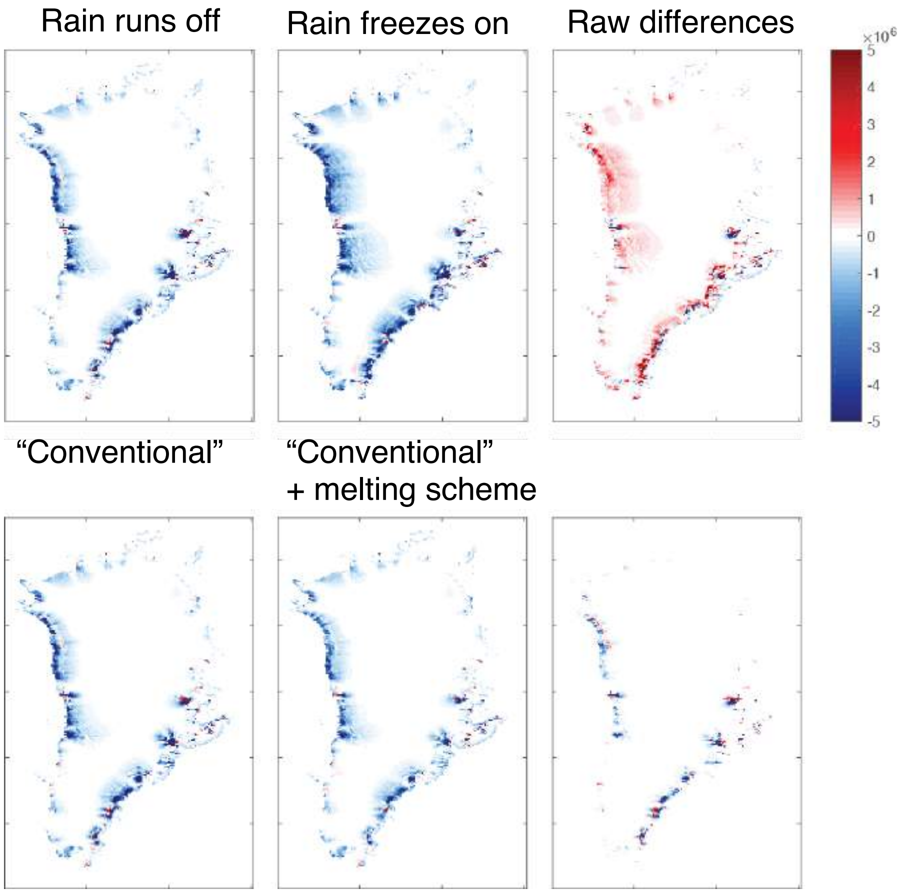
Variable (1)	Region (from Fig. 1A) (2)	$\frac{\delta J}{\delta \text{Variable}}$ (3)	$\frac{\Delta J}{\Delta \text{Variable}}$ (4)	ΔV_{adj} (5)	ΔV_{fd} (6)	% Deviation (7)
ice thickness	2	3.99915×10^8	3.99916×10^8	–	–	2.51×10^{-3}
July precipitation	1	2.1053×10^{16}	2.0709×10^{16}	–	–	1.6
basal temperature	3	2.1527×10^9	1.5422×10^8	–	–	93
surface temperature	TG	-1.5016×10^6	-4.2937×10^7	–	–	2.7593×10^3
surface temperature	Greenland	-9.0837×10^6	-9.1133×10^6	–	–	3.2448×10^{-1}
ice thickness	4	–	–	1.6411×10^5	1.2626×10^7	7.59×10^3
July precipitation	4	–	–	8.6987×10^{23}	1.7337×10^{23}	80
basal temperature	4	–	–	-1.1493×10^9	-1.5200×10^9	32

3. Adjoint model!

$$\frac{\partial \mathcal{J}}{\partial x} = \frac{\partial Volume}{\partial IceThickness}$$

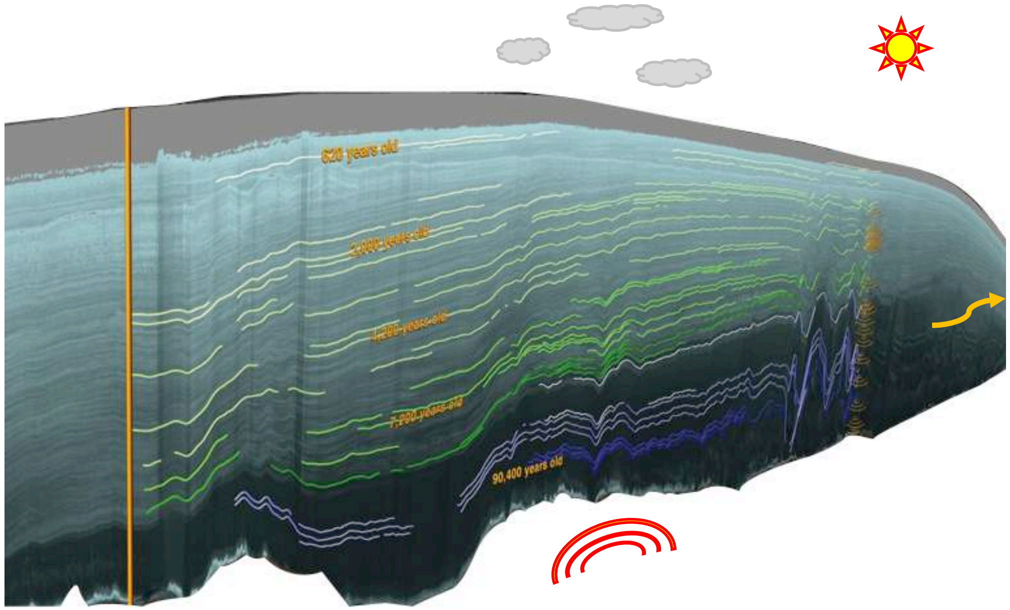
Ablation Scheme

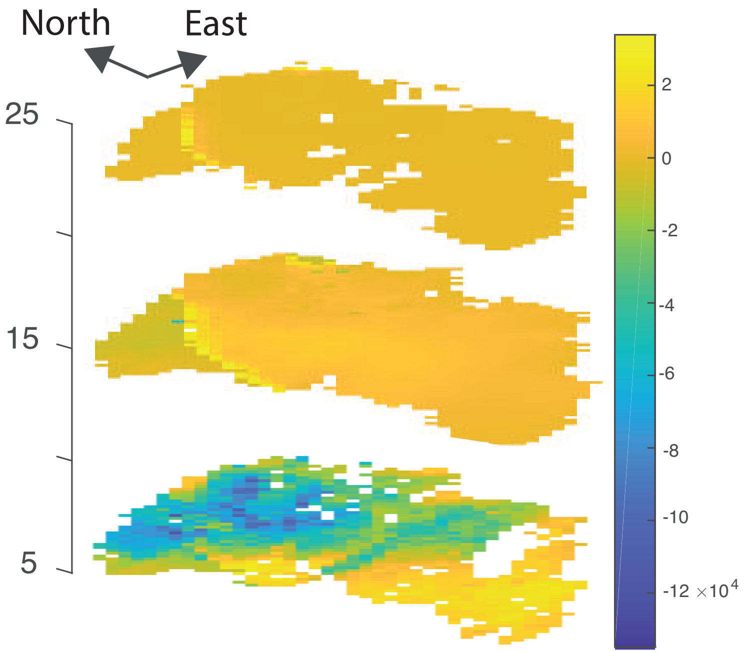
Enthalpy Formulation



3. Motivating problem: Greenland state estimate

Cost function: $\mathcal{J}(age_{ice}) = age_{observed} - age_{SICOPOLIS}$





Summary

- “abandon all non-differentiables, ye who adjoint here”
- Still have some work to do with Antarctic external library solver
- On the way to the Paleo State Estimate of Greenland