Automatic Differentiation of current ice sheet models
(with some context)

Daniel Goldberg (U. of Edinburgh School of Geoscience)
S.H. Krishna Narayanan (Argonne)
Laurent Hascoet (INRIA)
Jean Utke (Allstate)
Model adjoints in climate sciences

Adjoint have come into use for models with high degrees of complexity

- Atmosphere/Weather Models
- Ocean Circulation Models
- Ice Dynamic (Glacial flow) models
4DVar for weather models

Developed to improve numerical weather prediction


- Variational approach to optimising fit to 4-dimensional \((x,y,z,t)\) atmospheric data
Global Cost function:

\[ J = (x - x_b)^T C_{pr} (x - x_b) + \sum_{i=0}^{N} (y_i - F_i(x))^T C_{err} (y_i - F_i(x)) \]

Minimized to find best-fit of state

- x: control variable (state at initial time)
- y: observation at time level i
- \( F_i \): NWP map to \( y_i \)
- \( C_{pr} \): Prior Covariance matrix
- \( C_{err} \): Observational covariance
4DVar for weather models

Compare with 3DVar

\[ J = (x - x_b)^T C_{pr} (x - x_b) + (y - H(x))^T C_{err} (y - H(x)) \]

- \( H \) is now a time-independent map from state variables to observables
- No \textit{dynamics} used in assimilation
- Intended as “stepping stone” to 4DVar

3DVar (or filter)

4DVar
4DVar versus 3DVar

Lorenc and Rawlins, *QJRMS* (2005):

- In comparison, 4DVar outperformed 3DVar in forecast skill
- *But underperformed in data-fit within analysis window*

A better instantaneous fit to observations does not mean improved forecast skill

Could be that transient nature of 4DVar optimisation better constrains the relevant *modes of variability* (Swanson et al., 1998, *Tellus*)
Minimization of misfit

Just in case...

• Minimisation of cost function $J$ seeks a stationary point of $J$ in control space $x$

• General approach is (a) gradient descent, (b) Quasi-Newton, or (c) Newton

• (a,b) require the gradient of $J$. (c) requires its Hessian (which is more difficult than the gradient)
AD in atmospheric 4DVar

Atmospheric models have millions of degrees of freedom, billions of outputs → Adjoint models are required for optimisation

<To my knowledge> AD less extensive than e.g. in variational assimilation of ocean data

• 4DVar “grew up” before AD was extensively used

• In many approaches, continuous (Courtier and Talagrand, 1990) or discrete (Thepaut and Courtier, 1991) adjoints developed “by hand”
Adjoint of Ocean Models

- Ocean governed by similar equations of motion to the atmosphere
- Early ocean model-data syntheses used analytic adjoints (Tziperman and Thacker, 1989):

\[
\nabla^2 \psi_t + \psi_x + RJ(\psi, \nabla^2 \psi) \\
= -\epsilon_b \nabla^2 \psi + \epsilon_h \nabla^4 \psi + \text{curl} \tau(x, y), \\
\nabla^2 \lambda_t + \lambda_x + R[J(\lambda, \nabla^2 \psi) - \nabla^2 J(\lambda, \psi)] \\
= \epsilon_b \nabla^2 \lambda - \epsilon_h \nabla^4 \lambda + \Phi,
\]
Adjoint of Ocean Models

\[
\begin{align*}
\frac{D\vec{v}_h}{Dt} + f \hat{k} \times \vec{v}_h + \frac{1}{\rho_c} \nabla_z p &= \vec{F} \\
\epsilon_{n h} \frac{D w}{Dt} + \frac{g \rho}{\rho_c} + \frac{1}{\rho_c} \frac{\partial p}{\partial z} &= \epsilon_{n h} F_w \\
\nabla_z \cdot \vec{v}_h + \frac{\partial w}{\partial z} &= 0 \\
\rho &= \rho(\theta, S) \\
\frac{D \theta}{Dt} &= Q_{\theta} \\
\frac{DS}{Dt} &= Q_s
\end{align*}
\]

But complexity of modern ocean circulation models requires a more comprehensive, flexible approach.

http://mitgcm.org
Ocean State Estimation

- State Estimation: The equivalent (sort of) to 4DVar for the ocean
- Data record for the ocean grows, but still much remains unobserved
- Need for diverse data products and flexible assimilation platform
• State Estimation: The equivalent (sort of) to 4DVar for the ocean

• Data record for the ocean grows, but still much remains unobserved

• Need for diverse data products and flexible assimilation platform
Ocean State Estimation

• Difference in *timescales* between atmosphere and ocean

• Deep ocean has timescale of centuries – cannot have “jumps” every 6 hours like in 4DVar

• This would lead to transients that would take decades to dissipate
MITgcm adjoint/state estimate framework

\[ \frac{D\tilde{v}_h}{Dt} + f \mathbf{k} \times \nabla h + \frac{1}{\rho_c} \nabla z p = \bar{F} \]
\[ \epsilon_{n, h} \frac{D w}{D t} + \frac{g \rho}{\rho_c} + \frac{1}{\rho_c} \frac{\partial p}{\partial z} = \epsilon_{n, h} \bar{F}_w \]
\[ \nabla z \cdot \tilde{v}_h + \frac{\partial w}{\partial z} = 0 \]
\[ \rho = \rho(\theta, S) \]
\[ \frac{D \theta}{D t} = \bar{Q}_\theta \]
\[ \frac{D S}{D t} = \bar{Q}_s \]

Adjoint model
- State estimates
- Sensitivities

TAF
or
OpenAD
or
Tapenade

ECCO-GODAE Surface speed
Wunsch et al (2009)


Arctic State Estimate (courtesy P Heimbach)
Utility of State Estimates

What use is a state estimate if it can’t be used to predict?

• Outputs allow diagnostics not possible with measurements alone
• Provides boundary conditions or inputs to regional or biogeochemical studies

Wunsch et al, 2007
Adjoint sensitivities have been used extensively to understand variability and controlling factors of global ocean system.

Adjoint sensitivities of global production to iron and radiation (Dutkiewicz et al, 2006)

Submarine ice shelf melt sensitivity to ocean conditions (Heimbach and Losch, 2012)
What about ice sheets?

From this point I’ll discuss:

- Ice sheet modelling: approach and intuition
- Previous use of adjoint models
- AD in ice modelling, applications, new developments
Glacial flow: equations of motion

It is perhaps best to give the characteristics of ice flow before showing equations:

**Glacial bodies:**

1. **Are controlled by internal stresses**
   - Rather than inertia or coriolis

2. **Are non-Newtonian**
   - On timescales > 1 hour – power-law viscous

3. **Are not turbulent**

4. **Are thin** (small aspect ratio)

5. **Have longer time scales than the ocean or atmosphere**
   - Decades (glaciers and ice shelves) to millennia (ice sheets)
Glacial flow: equations of motion

Beginning with a general momentum balance and crossing out terms, the difference with the ocean becomes obvious:

\[ \rho \left( \frac{D\vec{u}}{Dt} + f \times \vec{u} \right) = -\nabla p + \nabla \cdot \sigma - g\vec{k} \]

\[ \nabla \cdot \vec{u} = 0 \]

Water:
\[ \sigma_{ij} = \mu \varepsilon_{ij} \equiv \frac{\mu}{2} \left( u_{i,j} + u_{j,i} \right) \]

Glacial ice:
\[ \sigma_{ij} = B(T) \left( \sqrt{2\varepsilon : \varepsilon} \right)^{-2/3} \varepsilon_{ij} \]

(Not as well-supported)

(Well-supported)

However, advection of scalars (e.g. temperature \( T \)) remain similar
Approximation to Full Stokes

Over most of Antarctica and Greenland, flow is dominated by either vertical shearing (interior) or horizontal deformation (streams/shelves).

In this case, approximations (including small aspect ratio and hydrostasy) can be made:

L1L2 (Hindmarsh, 2004): 2D (non-saddle) elliptic equation for depth-averaged velocities
“Hybrid” approximation (L1L2)

Approximation to Stokes, yields \(u, v\) (Goldberg 2011)

\[
\tau_{b,x} = \rho g H \partial_x s - \partial_x \left( 2H \bar{\mu}(2\bar{u}_x + \bar{v}_y) \right) - \partial_y \left( H \bar{\mu}(\bar{u}_y + \bar{v}_x) \right)
\]

\[
\tau_{b,y} = \rho g H \partial_y s - \partial_x \left( H \bar{\mu}(\bar{u}_y + \bar{v}_x) \right) - \partial_y \left( 2H \bar{\mu}(\bar{u}_x + 2\bar{v}_y) \right)
\]

\[
\mu = \frac{B(T)}{2} \left( 2\bar{u}_x^2 + 2\bar{v}_y^2 + 2\bar{u}_x \bar{v}_y + \frac{1}{4}(\bar{u}_y - \bar{v}_x)^2 + \frac{1}{4}(u_z^2 + v_z^2) \right)^{\frac{1-n}{2n}}
\]

Additionally, a closure is required for \(\tau_b\) (basal stress) – generally a linear sliding law, i.e.

\[
\bar{\tau} = \beta^2 \bar{u}(z_b)
\]

Evolution for thickness \(H\)

\[
H_t + \nabla \cdot (\bar{u}_h H) = a, \quad \bar{u}_h = (\bar{u}, \bar{v})
\]
“Hybrid” approximation (L1L2)

Approximation to Stokes, yields $u,v$ (Goldberg 2011)

\[
\begin{align*}
\tau_{b,x} &= \rho g H \partial_x s - \partial_x \left( 2H \bar{\mu}(2\bar{u}_x + \bar{v}_y) \right) - \partial_y \left( H \bar{\mu}(\bar{u}_y + \bar{v}_x) \right) \\
\tau_{b,y} &= \rho g H \partial_y s - \partial_x \left( H \bar{\mu}(\bar{u}_y + \bar{v}_x) \right) - \partial_y \left( 2H \bar{\mu}(\bar{u}_x + 2\bar{v}_y) \right)
\end{align*}
\]

\[
\mu = \frac{B(T)}{2} \left( 2\bar{u}_x^2 + 2\bar{v}_y^2 + 2\bar{u}_x \bar{v}_y + \frac{1}{4} (\bar{u}_y + \bar{v}_x)^2 + \frac{1}{4} (\bar{u}_z^2 + \bar{v}_z^2) \right) \frac{1-n}{2n}
\]

Additionally, a closure is required for $\tau_b$ (basal stress) – generally a linear sliding law, i.e.

\[
\bar{\tau} = \beta^2 \bar{u}(z_b)
\]

Evolution for thickness $H$

\[
H_t + \nabla \cdot (\bar{u}_h H) = a, \quad \bar{u}_h = (\bar{u}, \bar{v})
\]
How ice dynamics “work”

- Ice geometry (through driving stress) induces ice flow.
- On land, horizontal stresses distribute basal stress and driving stress nonlocally.
- Ice shelves transmit forces, affecting flow within the ice sheet.
  - This is how the ocean is able to affect ice sheets!
A common adjoint application: Basal shear inversion

$$\tau = \beta(x, y)^2 \tilde{u}(z_b)$$

Bed friction

Glacier

$$\tilde{u} = \text{Forward model for } u,v$$

$$\begin{align*}
\partial_x \left( 2H \mu (2u_x + v_y) \right) + \partial_y \left( H \mu (u_y + v_x) \right) - \beta^2 u &= \rho g H s_x \\
\partial_x \left( H \mu (u_y + v_x) \right) + \partial_y \left( 2H \mu (u_x + 2v_y) \right) - \beta^2 v &= \rho g H s_y
\end{align*}$$

$$\begin{align*}
\partial_x \left( 2H \mu (2\lambda_x + \eta_y) \right) + \partial_y \left( H \mu (\lambda_y + \eta_x) \right) - \beta^2 \lambda &= (u - u_{\text{obs}}) \\
\partial_x \left( H \mu (\lambda_y + \eta_x) \right) + \partial_y \left( 2H \mu (\lambda_x + 2\eta_y) \right) - \beta^2 \eta &= (v - v_{\text{obs}})
\end{align*}$$

A “static” or “snapshot” assimilation of surface velocity

A “static” or “snapshot” assimilation of surface velocity

Forward model for $u,v$

Adjoint model (no need for AD!)
A common adjoint application: Basal shear inversion

Bed friction

\[ \vec{t} = \beta(x, y)^2 \vec{u}(z) \]

Forward model for \( u,v \)

Shear pattern under PIG (Morlighem et al 2010)

A “static” or “snapshot” assimilation of surface velocity

\[
\begin{align*}
\partial_x \left( 2H \mu(2\lambda_x + \eta_y) \right) &+ \partial_y \left( 2H \mu(\lambda_y + 2\eta_x) \right) - \beta^2 \eta = (v - v_{\text{obs}}) \\
\partial_x \left( H \mu(\lambda_y + \eta_x) \right) &+ \partial_y \left( 2H \mu(\lambda_x + 2\eta_y) \right) - \beta^2 \eta = (v - v_{\text{obs}})
\end{align*}
\]

Adjoint model (no need for AD!)

\[
\begin{align*}
\partial_x \left( H \mu(\lambda_y + \eta_x) \right) - \beta^2 u & = \rho g H s_x \\
\partial_y \left( 2H \mu(\lambda_x + 2\eta_y) \right) - \beta^2 v & = \rho g H s_y
\end{align*}
\]
Adding the time dimension

- But the amount of available data is growing exponentially
  - Time-dependent altimetry and velocities, internal layers
- By 3DVar/4DVar analogy: our predictions will improve if we incorporate the time dimension
- By State Estimate analogy: balanced model state that is useful to further investigation

It starts with a time-dependent adjoint – and for that we need AD
STREAMICE: MITgcm glacial dynamics package

- MITgcm: initially developed to solve primitive equations for ocean
- But physical “packages” since developed (e.g. sea ice, chemistry)

STREAMICE package:
- Glacial flow dynamics with hybrid stress balance
- Q1 Finite element solve for velocity, Finite Volume for thickness
- “Borrows” grid, domain decomposition and I/O from main code
- Therefore, application of AD did not need to reinvent the wheel
FOR n = initialTimeStep TO finalTimeStep

    // Constructs \( \hat{a} \) from \( H^{[n]} \):
    CALL CALC_DRIVING_STRESS(\( H^{[n]} \))
    \( m = 0 \)
    REPEAT UNTIL CONVERGENCE OF \( \mathbf{u} \)
        \( \mathbf{u} = \Phi(\mathbf{u}, \hat{a}) \)
        \( m = m+1 \)
        store \( L, \mathbf{u} \) and other variables
        \( \text{last}\!m^{[n]} = m \)
    // Finds \( H^{[n+1]} \) from continuity equation with \( u \):
    CALL ADVECT_THICKNESS()
Algorithm

FOR $n = \text{initialTimeStep}$ TO $\text{finalTimeStep}$

// Constructs $\hat{a}$ from $H^{[n]}$:

CALL CALC_DRIVING_STRESS($H^{[n]}$)
$m = 0$
REPEAT UNTIL CONVERGENCE OF $u$

$u = \Phi(u, \hat{a})$
$m = m+1$
store $L$, $u$ and other variables
$\text{lastm}^{[n]} = m$

// Finds $H^{[n+1]}$ from continuity equation with $u$:
CALL ADVECT_THICKNESS()

FOR $n = \text{finalTimeStep}$ DOWNTO $\text{initialTimeStep}$

// Constructs $\delta^*H^{[n]}$ and $\delta^*u^{[n]}$ from $\delta^*H^{[n+1]}$
// via the adjoint of the continuity equation:

CALL AD_ADVECT_THICKNESS()
REPEAT $\text{lastm}^{[n]}$ TIMES

restore $L$, $u$ and other variables

$\delta^*\hat{a} = \delta^*\hat{a} + \delta^*u(\frac{\partial \Phi}{\partial \hat{a}})^T$

$\delta^*u = \delta^*u(\frac{\partial \Phi}{\partial u})^T$

// Updates $\delta^*H^{[n]}$ from $\delta^*\hat{a}$:
CALL AD_CALC_DRIVING_STRESS($\delta^*H^{[n]}$)

---

Forward

$A_{n-1} = F(u_{n-1})$
$u_n = A_{n-1}^{-1}b$

Adjoint

$\delta^*b = \delta^*b + A_{n-1}^{-T}\delta^*u_n$
$\delta^*A_{n-1} = -\delta^*b(u_n^T)$

.....
1. Adjoint calculation has own fixed-point iteration – may be underconverged (inaccurate) or overconverged (wasteful)
2. Requires that intermediate variables in calculation of $\Phi$ (there are lots) be stored or recomputed

(2) is specific issue with OpenAD
• OpenAD more “aggressive” in defining state and has larger memory/storage needs
• A method where storage does not grow with # of fixed-point iterations is critical
Special treatment of fixed point problems

Christianson, B. (1994), Reverse accumulation and attractive fixed points: an algorithm for calculating adjoint of generic fixed point problem

Mathematical basis of algorithm:

Converged state

\[ u_* = \Phi(u_*, a) \]

Differential of conv state

\[ \delta u_* = \Phi_u(u_*, a)\delta u_* + \Phi_a(u_*, a)\delta a \]

Rearranged:

\[ \delta u_* = \left[I - \Phi_u(u_*, a)\right]^{-1}\Phi_a(u_*, a)\delta a \]

\[ = \left[I + \Phi_u + \Phi_u^2 + \Phi_u^3 + \ldots\right]\Phi_a(u_*, a)\delta a \]

- The algorithm essentially constructs the adjoint of this operator
- Note: If the forward problem converges, the series above converges
Fixed-point adjoint AD model (Store-All)

\[
\begin{align*}
  x &= y \cdot \frac{\partial f}{\partial x}(z,x) \\
  z &= y \cdot \frac{\partial f}{\partial z}(z,x) \\
  \bar{w} &= \bar{z} \\
  \bar{w} &= \bar{w} \cdot \frac{\partial \phi}{\partial z}(z,x) + \bar{z}
\end{align*}
\]

\[
\begin{align*}
  z &= g(x) \\
  \text{Repeat until } z \text{ stable} \\
  z &= \phi(z,x) \\
  \text{end} \\
  \text{push}(z) \\
  z &= \phi(z,x) \\
  y &= f(z,x)
\end{align*}
\]

\[
\begin{align*}
  \text{Repeat until } \bar{w} \text{ stable} \\
  \text{look}(z) \\
  \bar{w} &= \bar{w} \cdot \frac{\partial \phi}{\partial z}(z,x) + \bar{z} \\
  \text{end} \\
  \text{pop}(z) \\
  \bar{x} &= \bar{w} \cdot \frac{\partial \phi}{\partial x}(z,x) \\
  \bar{z} &= 0.0
\end{align*}
\]

OpenAD Fixed Point Modification

Modified Algorithm

FOR n = initialTimeStep TO finalTimeStep
  // Constructs \( \hat{a} \) from \( H^{[n]} \):
  CALL CALC_DRIVING_STRESS(\( H^{[n]} \))
  \( u = \) initial guess
  CALL PHISTAGE(PRELOOP, \( w \), \( u \), \( \hat{a} \))
  REPEAT UNTIL CONVERGENCE OF \( u \)
    CALL PHISTAGE(INLOOP, \( w \), \( u \), \( \hat{a} \))
  CALL PHISTAGE(POSTLOOP, \( w \), \( u \), \( \hat{a} \))
  // Finds \( H^{[n+1]} \) from continuity equation with \( u \):
  CALL ADVECT_THICKNESS()

SUBROUTINE PHISTAGE(phase, \( w \), \( u \), \( \hat{a} \))
  IF (phase==PRELOOP)
    // do nothing
  ELSE IF (phase==INLOOP)
    save tape pointer
    \( u = \Phi(u, \hat{a}) \)
    // Makes sure no storage is done:
    restore tape pointer
  ELSE IF (phase==POSTLOOP)
    \( u = \Phi(u, \hat{a}) \)
    store \( L \), \( u \) and other variables

Fixed-point Adjoint

FOR n = finalTimeStep DOWNTO initialTimeStep
  // Constructs \( \delta^*H^{[n]} \) and \( \delta^*u \) from \( \delta^*H^{[n+1]} \)
  // via the adjoint of the continuity equation:
  CALL AD_ADVECT_THICKNESS()
  CALL AD_PHISTAGE(POSTLOOP, \( \delta^*w \), \( \delta^*u \), \( \delta^*\hat{a} \))
  REPEAT UNTIL CONVERGENCE OF \( \delta^*w \)
    CALL AD_PHISTAGE(INLOOP, \( \delta^*w \), \( \delta^*u \), \( \delta^*\hat{a} \))
  CALL AD_PHISTAGE(PRELOOP, \( \delta^*w \), \( \delta^*u \), \( \delta^*\hat{a} \))
  \( \delta^*u = 0.0 \)
  // Updates \( \delta^*H^{[n]} \) from \( \delta^*\hat{a} \):
  CALL AD_CALC_DRIVING_STRESS(\( \delta^*H^{[n]} \))

SUBROUTINE AD_PHISTAGE(phase, \( \delta^*w \), \( \delta^*u \), \( \delta^*\hat{a} \))
  IF (phase==POSTLOOP)
    \( \delta^*w = \delta^*u \)
  ELSE IF (phase==INLOOP)
    save tape pointer
    restore \( L \), \( u \) and other variables
    \( \delta^*w = \delta^*w \left( \frac{\partial \Phi}{\partial u} \right)^T + \delta^*u \)
    // Makes sure converged state is reused:
    restore tape pointer
  ELSE IF (phase==PRELOOP)
    \( \delta^*\hat{a} = \delta^*w \left( \frac{\partial \Phi}{\partial u} \right)^T \)

Goldberg, Narayanan, Hascoet and Utke, Geoscientific Model Dev (2016)
OpenAD Fixed Point Modification

Time and Memory performance

<table>
<thead>
<tr>
<th>Grid size</th>
<th>Plain (untouched)</th>
<th>Mechanical adjoint</th>
<th>BC94 algorithm</th>
<th>BC94 algorithm with LU optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>total</td>
<td>avg iter</td>
<td>total</td>
<td>avg iter</td>
</tr>
<tr>
<td>40 × 20</td>
<td>6.1 s</td>
<td>34</td>
<td>26 s</td>
<td>9.8 s</td>
</tr>
<tr>
<td></td>
<td>reverse 21 s</td>
<td>adj iter 37</td>
<td>total 22 s</td>
<td>forward 94 s</td>
</tr>
<tr>
<td></td>
<td>dbl tape 392 MB</td>
<td></td>
<td>total 22 s</td>
<td>forward 94 s</td>
</tr>
<tr>
<td>0 × 40</td>
<td>69 s</td>
<td>37</td>
<td>278 s</td>
<td>122 s</td>
</tr>
<tr>
<td></td>
<td>reverse 193 s</td>
<td>adj iter 37</td>
<td>total 278 s</td>
<td>forward 94 s</td>
</tr>
<tr>
<td></td>
<td>dbl tape 1.42 GB</td>
<td></td>
<td>total 278 s</td>
<td>forward 94 s</td>
</tr>
<tr>
<td>160 × 80</td>
<td>591 s</td>
<td>51</td>
<td>1553 s</td>
<td>944 s</td>
</tr>
<tr>
<td></td>
<td>reverse 1514 s</td>
<td>adj iter 39</td>
<td>total 1553 s</td>
<td>forward 608 s</td>
</tr>
<tr>
<td></td>
<td>dbl tape 2.95 GB</td>
<td></td>
<td>total 1553 s</td>
<td>forward 608 s</td>
</tr>
</tbody>
</table>

- Treatment greatly reduces memory load
- Does not reduce timing significantly
- **However**, repeat inversion of identical matrix allows for LU factorisation, giving ~30% performance boost
• **Forward model:** 10-year simulation of model (beginning from steady state)

• **Cost function:** Volume above floatation loss after $t=10$yr

• **Controls:** (steady) melt rate and ice strength $B$

---

Adjoint sensitivities of a marine ice sheet (ideal)

Note: vulnerability of shear margin!

*Goldberg and Heimbach, Cryosphere, 2013*
Adjoint sensitivities of a marine ice sheet (ideal)

- 10-year simulation of model (beginning from steady state)
- Cost function: Volume above floating loss after \( t=10\text{yr} \)
- Controls: (steady) melt rate and ice strength \( B \)

Note: Vulnerability of shear margin!

Goldberg and Heimbach, Cryosphere, 2013
Adjoint sensitivities of a marine ice sheet, realistic

A similar experiment, but more realistic setting: **Smith Glacier and Crosson Ice shelf**, Amundsen Embayment, West Antarctica

- **Cost function**: Sensitivity of grounded ice loss (i.e. sea level contribution) over 10 years
- **Control**: Steady melt rate field

- Ice speed from MEaSUREs (Rignot et al, 2011)
Adjoint sensitivities of a marine ice sheet, realistic

A similar experiment, but more realistic setting: **Smith Glacier and Crosson Ice shelf**, Amundsen Embayment, West Antarctica

- **Cost function**: Sensitivity of grounded ice loss (i.e. sea level contribution) over 10 years
- **Control**: Steady melt rate field

Again, sensitivity is highest in **shear margin** between Crosson and Dotson

![Image showing sensitivity and speed fields with key performance metrics]

- ~400k DoF in solver
- 80 time steps (~3000 linear solves)
- 60 cores

**Performance Metrics**
- Forward: 1150 s
- Reverse: 2765 s
- Reverse with LU opt: 1778 s
Assimilation of time-dependent ice sheet data

Goldberg et al., The Cryosphere, 2015
Assimilation of time-dependent ice sheet data

- **Cost function:** (Weighted) misfit to thinning and velocity data
- **Control:** Basal sliding parameters and *boundary stresses*

2011 thinning misfit using
- 2001 **Snapshot Calibration** (left)
- 2001-2010 **Transient Calibration** (right)
• Prediction of grounded ice loss

• Note: this is not really a “3DVar versus 4DVar” test, because I don’t know how good the “4DVar” prediction is!
• But it does demonstrate the effect of incorporating time-dependent data into ice model calibration
Assimilation of time-dependent ice sheet data

• Can time-dependent controls be constrained?

• Constraining transient behaviour could give strong insight into hidden physics

• But is there sufficient data to do so?

• Cost function should go down as control space dimension increases – but how much should it decrease? (e.g. Bayes Information Criterion)
Additional applications of AD to ice models

- Heimbach and Bugnion, 2009
- OpenAD applied to SICOPOLIS – shallow ice approximation
- Application to Greenland Ice Sheet

- Larour and Utke et al, 2014
- ADOL-C applied to Shallow Shelf version of ISSM
- Adjoinable MPI developed (Larour and Utke et al, in press)
1. What are the uncertainties of the inferred parameters, and how do they affect forward projections?

2. Which observations most strongly influence predictions? *Can we use this to observe more efficiently?*

3. How long does the assimilation window need to be for a given prediction window?
“2nd Order” sensitivities – Hessian example

- First-order adjoint gives sensitivity of model to input parameters.
- But parameters are just products of inversion.
- We should be asking – how sensitive is the model to the data?

\[
\frac{\partial c_i}{\partial \hat{u}_k} = \left( \frac{\partial^2 J}{\partial c_i \partial c_j} \right)^{-1} \left( \frac{\partial^2 J}{\partial \hat{u}_k \partial c_j} \right)
\]

\[
\frac{\partial F}{\partial \hat{u}_k} = H^T_{\text{ext}} \left( \frac{\partial F}{\partial c_i} \right)^T
\]

\( u_k \): data  \quad \( c_i \): controls

\( J \): (static) cost fcn

\( F \): (time-dep) quantity of interest
Coupled ice-ocean adjoints?

As an MITgcm package, STREAMICE allows online (synchronous) ice-ocean coupling.

Though wetting/drying issues need to be sorted out!

In principle, coupled model will be adjoinable.

Asay-Davis et al, 2016

Synchronous simulation (no thermo)
Summary

• Ice sheet model-data synthesis has a long way to go to catch up with weather and ocean assimilation.

• But we should draw lessons from the latter two!

• Time-dependent adjoints are a good start.

• But more work is needed to:
  • Incorporate diverse data sources
  • Constrain posterior uncertainties of parameters
  • Optimise data collection strategy
  • Identify necessary assimilation windows
Assimilation of 'me-dependent ice sheet data
Assimilation of time-dependent ice sheet data