Moment based point spread function (PSF) Hessian approximation for large scale ice sheet inverse problems

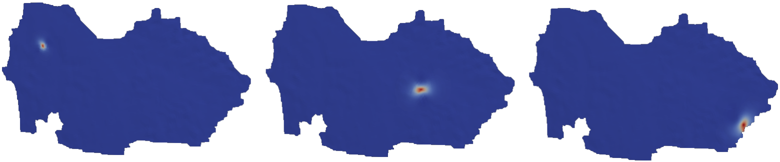
Nick Alger,[1] Tucker Hartland,[2]
Toby Isaac,[3] Noémi Petra,[2] Omar Ghattas[1]

[1]Oden Institute
The University of Texas at Austin

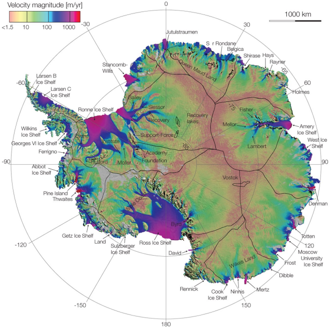
[2]Applied Mathematics, School of Natural Sciences
University of California, Merced

[2]Mathematics and Computer Science Division
Argonne National Laboratory

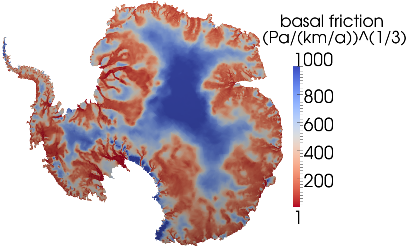
SIAM UQ24, Feb. 27, 2024



Antartic ice sheet



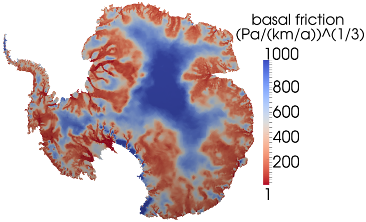
Observed surface flow velocity from InSAR (Rignot et. al, 2011)



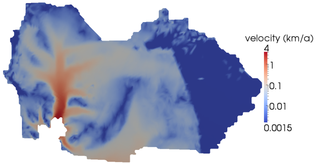
Antarctic ice sheet inversion for the basal friction parameter field from InSAR surface velocities

Details in: T. Isaac, N. Petra, G. Stadler, and O. Ghattas. *Scalable and efficient algorithms for the propagation of uncertainty from data through inference to prediction for large-scale problems, with application to flow of the Antarctic ice sheet*, Journal of Computational Physics, 296, 348-368 (2015).

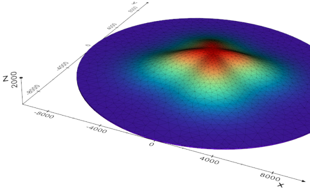
Model problems



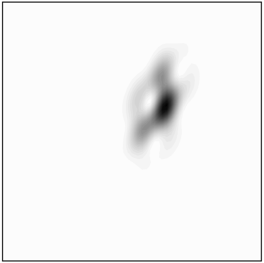
(a) Antartica
(Ymir)



(b) Pine Island Glacier
(Ymir)



(c) Ice Mountain (Hippylib)



(d) Spatially variant blur

Ice sheet dynamics: forward and inverse

Balance of linear momentum, mass, and energy

$$-\boldsymbol{\nabla} \cdot [\eta(\theta, \boldsymbol{u})\, \dot{\boldsymbol{\varepsilon}} - \boldsymbol{I}p] = \rho \boldsymbol{g}, \qquad [\dot{\boldsymbol{\varepsilon}} = \tfrac{1}{2}(\boldsymbol{\nabla u} + \boldsymbol{\nabla u}^T)]$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0,$$

$$\rho c \left(\frac{\partial \theta}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \theta \right) - \boldsymbol{\nabla} \cdot (K \boldsymbol{\nabla} \theta) = 2\, \eta\, \mathrm{tr}(\dot{\boldsymbol{\varepsilon}}^2)$$

We have: Satellite observations of surface velocity

We want: The sliding/friction coefficient β in Robin boundary condition

$$\mathbf{T}(\boldsymbol{\sigma}\mathbf{n}) + \beta(x)\mathbf{T}\mathbf{u} = 0$$

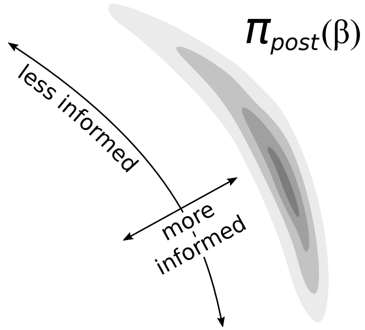
($\mathbf{T}$ is tangential component)

Bayesian approach

Inverse problem: given noisy data $\boldsymbol{d}$ and a model $\boldsymbol{f}$, infer parameters $\boldsymbol{\beta}$ that characterize the model, i.e.,

$$\boldsymbol{f}(\boldsymbol{\beta}) + \boldsymbol{e} = \boldsymbol{d}$$

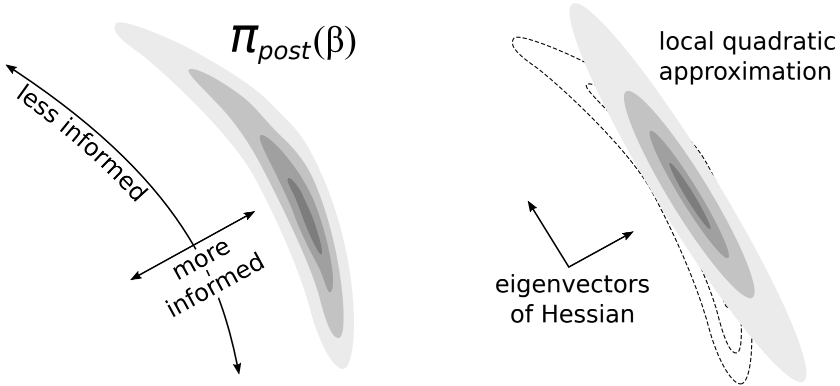
Interpret $\boldsymbol{\beta}$, $\boldsymbol{d}$ as random variables; solution of inverse problem is the "posterior" probability density function $\pi_{\text{post}}(\boldsymbol{\beta})$ found via Bayes' theorem.



Hessian: local Gaussian approximation

Local Gaussian approximation proposal:

$$\pi_{\mathsf{prop}}(\beta) := \frac{\det \boldsymbol{H}^{1/2}}{(2\pi)^{n/2}} \exp\left(-\frac{1}{2}\left(\boldsymbol{y} - \boldsymbol{\beta}_k + \boldsymbol{H}^{-1}\boldsymbol{g}\right)^T \boldsymbol{H}\left(\boldsymbol{y} - \boldsymbol{\beta}_k + \boldsymbol{H}^{-1}\boldsymbol{g}\right)\right)$$



Matrix-free

$$\mathbf{H} = \underbrace{\mathbf{H}_d}_{\text{data misfit Hessian}} + \underbrace{\mathbf{H}_r}_{\text{Prior Hessian}}$$

- Data misfit Hessian, and therefore the whole Hessian, are only available matrix-free
- Cannot access $\mathbf{H}_{ij}$ easily
- Can compute matrix-vector products (matvecs)

$$\mathbf{u} \mapsto \mathbf{H}\mathbf{u}$$

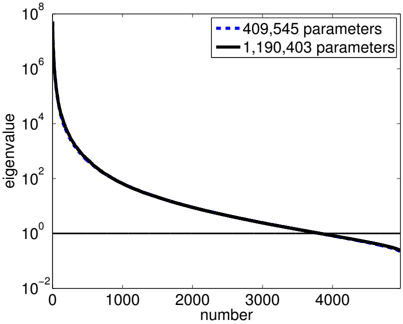
- Cost: **2** linearized Stokes **PDE solves** per matvec

Low rank Hessian approximation (extremely expensive!)

Low-rank approximation/Woodbury formula:

$$\boldsymbol{H}^{-1} = (\boldsymbol{H}_d + \mathbf{H}_r)^{-1} \approx \mathbf{H}_r^{1/2}(\boldsymbol{V}_r \boldsymbol{\Lambda}_r \boldsymbol{V}_r^T + \boldsymbol{I})^{-1}\mathbf{H}_r^{1/2}$$

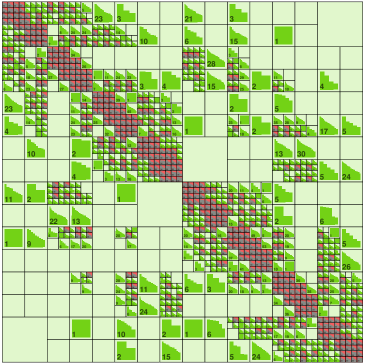
where $\boldsymbol{V}_r$ and $\boldsymbol{\Lambda}_r$ are the eigenvectors/values of $\boldsymbol{H}_d \boldsymbol{v}_i = \lambda_i \mathbf{H}_r \boldsymbol{v}_i$



Details in: T. Isaac, N. Petra, G. Stadler, and O. Ghattas. *Scalable and efficient algorithms for the propagation of uncertainty from data through inference to prediction for large-scale problems, with application to flow of the Antarctic ice sheet*, Journal of Computational Physics, 296, 348-368 (2015).

Hierarchical matrices ($\mathcal{H}$-matrices)

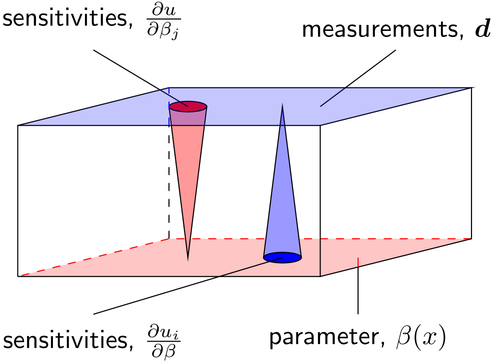
- Matrix is reordered and subdivided recursively
- Many off-diagonal blocks are low-rank
- Overall matrix may be high rank
- $O(N\,(\log N)^a)$ complexity matrix operations, $a \in \{0, 1, 2, 3\}$
  - matrix-vector products, matrix-matrix addition, matrix-matrix multiplication, matrix factorization, matrix inversion, ...



Hierarchical matrix vs. matrix free

- Classical methods for building $\mathcal{H}$-matrix require matrix entries $\mathbf{H}_{ij}$
- New algebraic methods based on "peeling process" can build $\mathcal{H}$-matrix from matrix-vector products
  - Ambartsumyan, Boukaram, Bui-Thanh, Ghattas, Keyes, Stadler, Turkiyyah, and Zampini, "Hierarchical matrix approximations of Hessians arising in inverse problems governed by PDEs", SISC, 2020.
  - Hartland, Tucker Andrew, et al. "Hierarchical off-diagonal low-rank approximation of Hessians in inverse problems, with application to ice sheet model initialization." Inverse Problems (2023).
- **Problem:** peeling process better than low rank, but **still expensive**
- Here we build the $\mathcal{H}$-matrix faster by taking advantage of the problem structure
  - **Local sensitivities**
  - **Local mean-displacement invariance**
  - **Non-negative impulse responses***

Local sensitivities

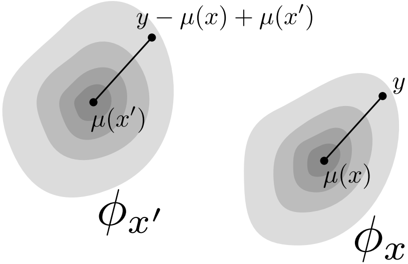


Local mean-displacement invariance

Impulse response ϕ_x:

$\phi_x := H_d \; \delta_x =$ action of Hessian operator on delta distribution at x

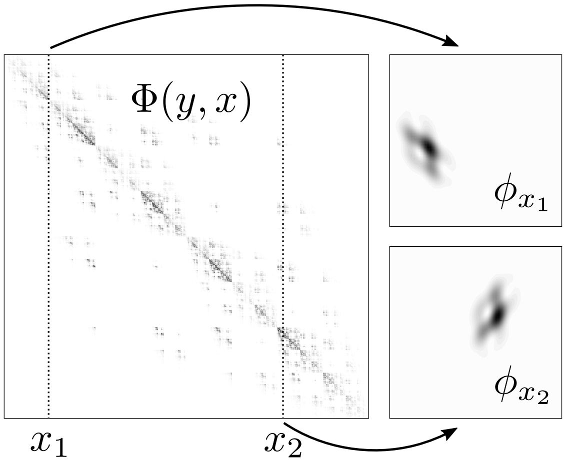
$\mu(x) :=$ center of mass (mean) of ϕ_x



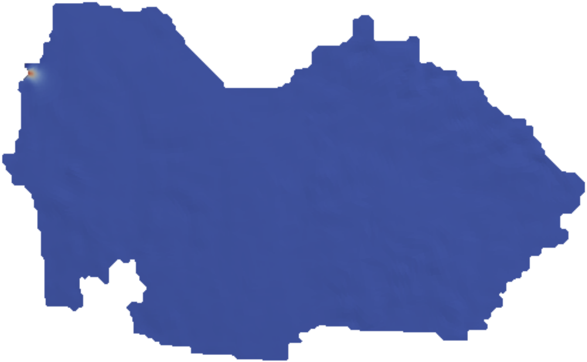
Local mean-displacement invariance:

$$\phi_x(y) \approx \phi_{x'}(y - \mu(x) + \mu(x'))$$

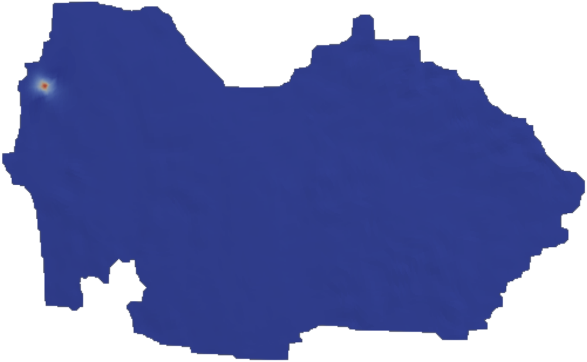
Impulse responses are "columns" of the integral kernel



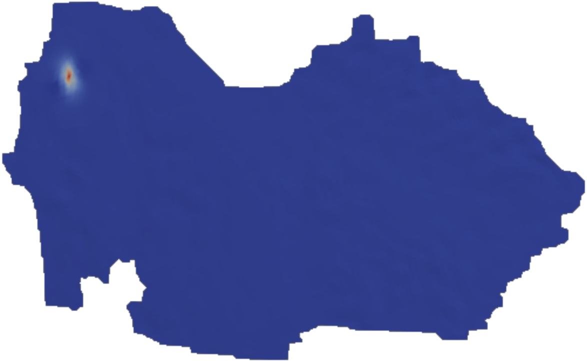
Hessian impulse responses (Pine Island Glacier)



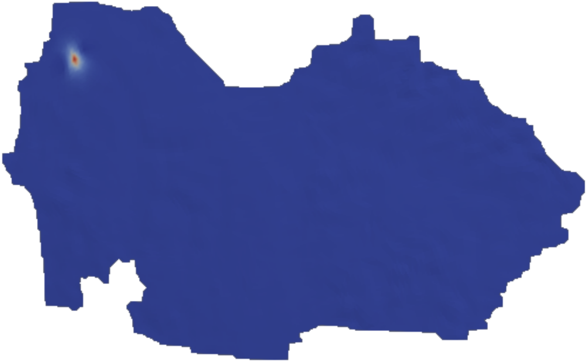
Hessian impulse responses (Pine Island Glacier)



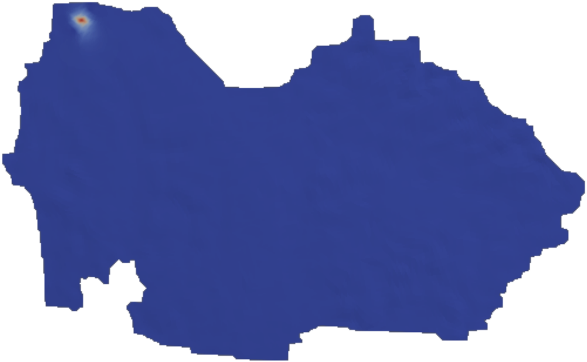
Hessian impulse responses (Pine Island Glacier)



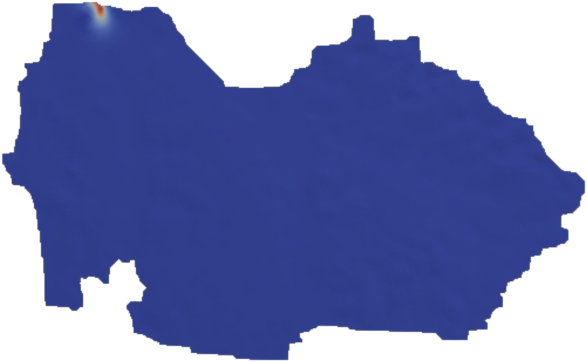
Hessian impulse responses (Pine Island Glacier)



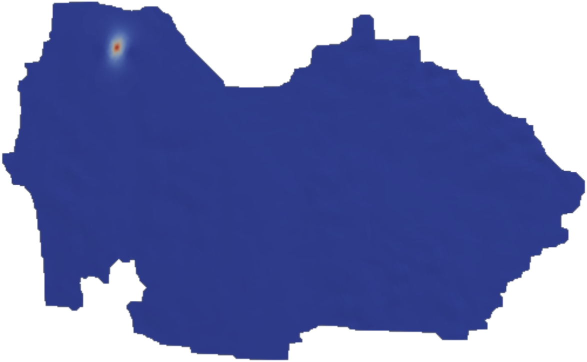
Hessian impulse responses (Pine Island Glacier)



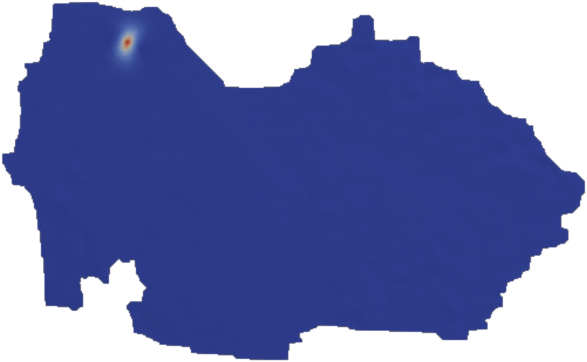
Hessian impulse responses (Pine Island Glacier)



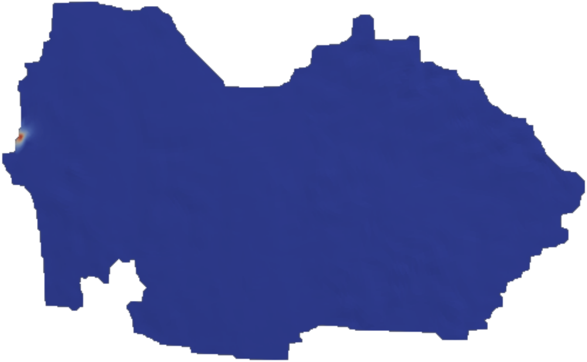
Hessian impulse responses (Pine Island Glacier)



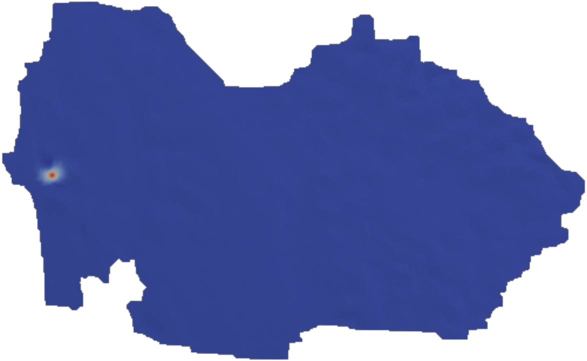
Hessian impulse responses (Pine Island Glacier)



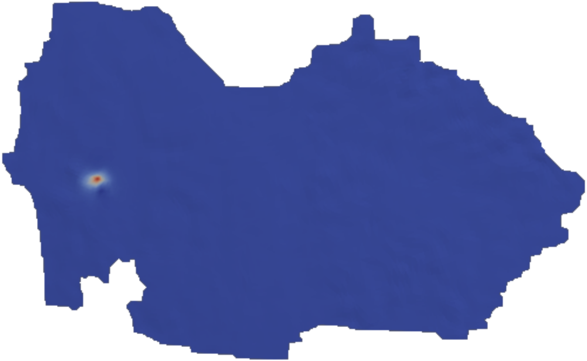
Hessian impulse responses (Pine Island Glacier)



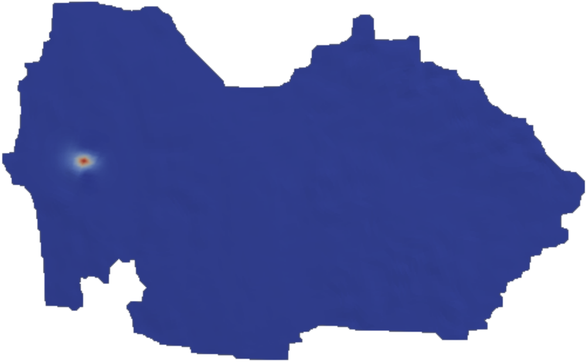
Hessian impulse responses (Pine Island Glacier)



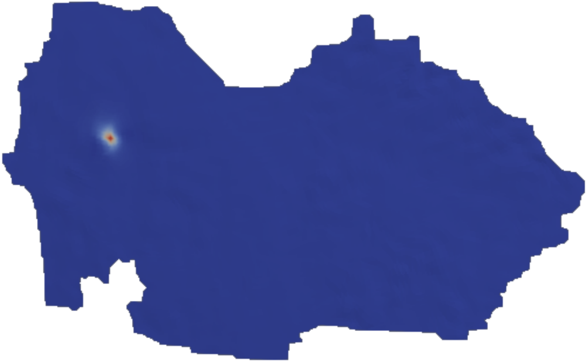
Hessian impulse responses (Pine Island Glacier)



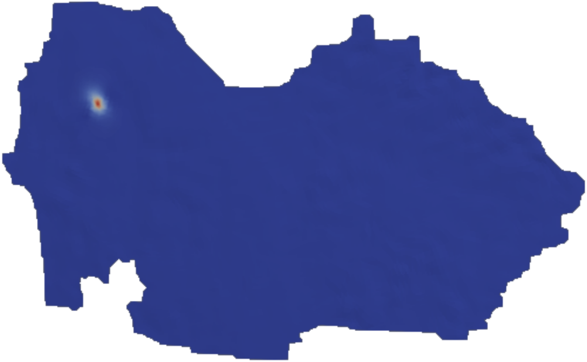
Hessian impulse responses (Pine Island Glacier)



Hessian impulse responses (Pine Island Glacier)

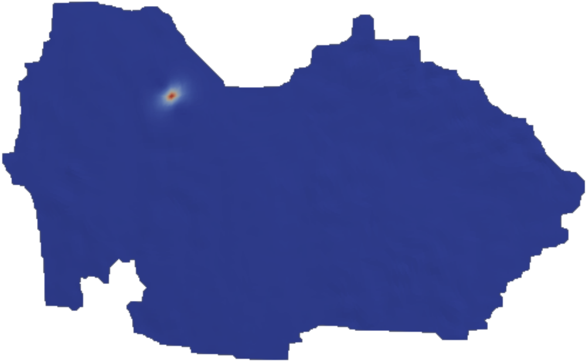


Hessian impulse responses (Pine Island Glacier)

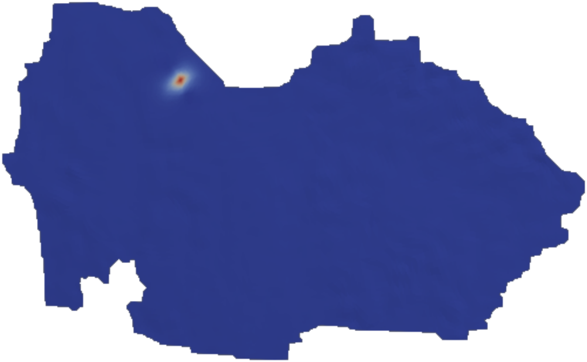


Hessian impulse responses (Pine Island Glacier)

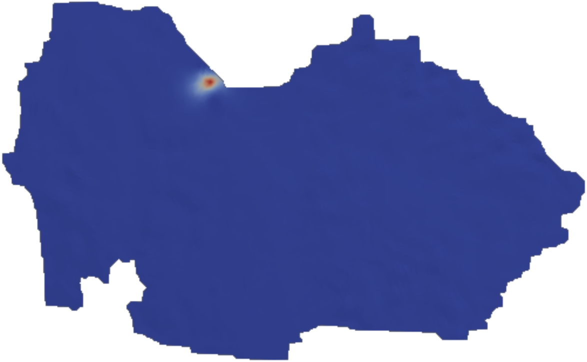
Hessian impulse responses (Pine Island Glacier)



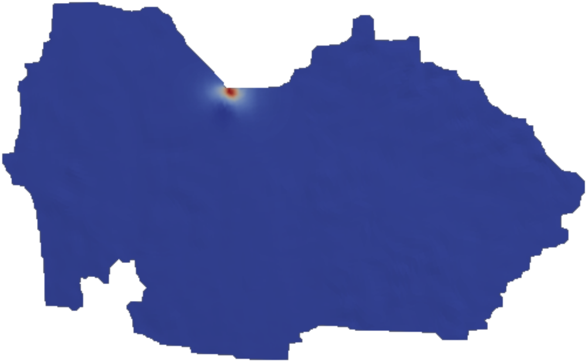
Hessian impulse responses (Pine Island Glacier)



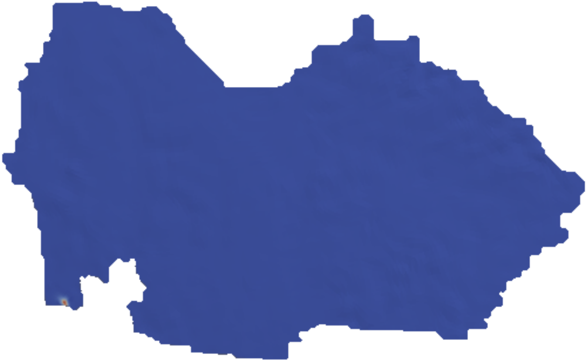
Hessian impulse responses (Pine Island Glacier)



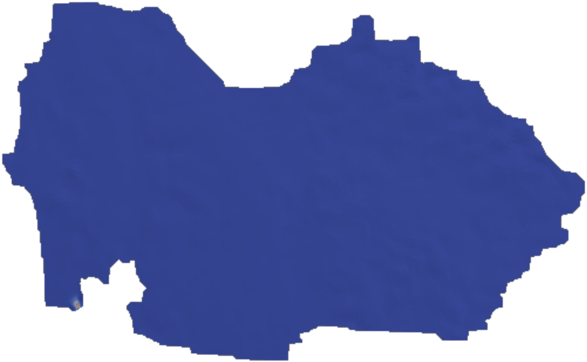
Hessian impulse responses (Pine Island Glacier)



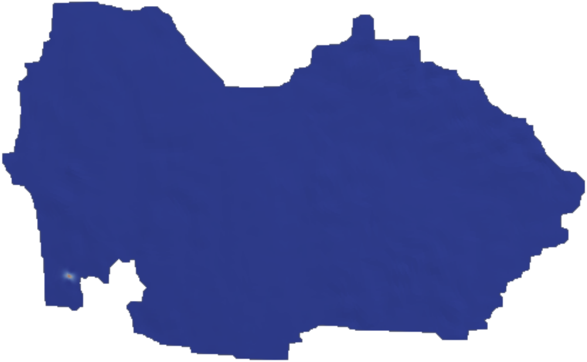
Hessian impulse responses (Pine Island Glacier)



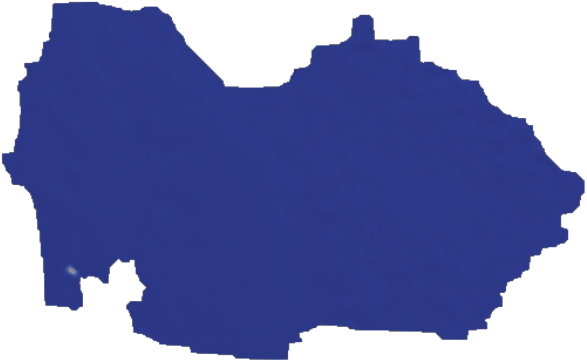
Hessian impulse responses (Pine Island Glacier)



Hessian impulse responses (Pine Island Glacier)

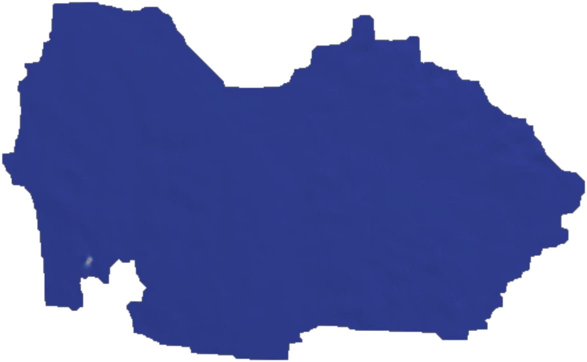


Hessian impulse responses (Pine Island Glacier)

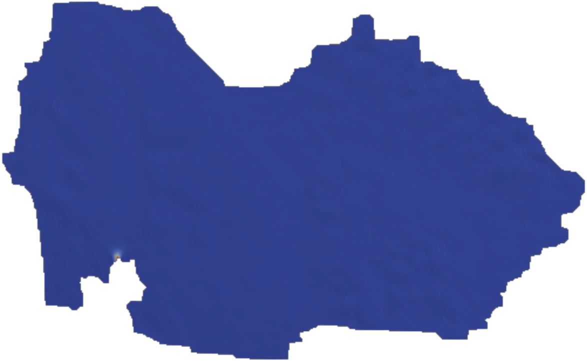


Hessian impulse responses (Pine Island Glacier)

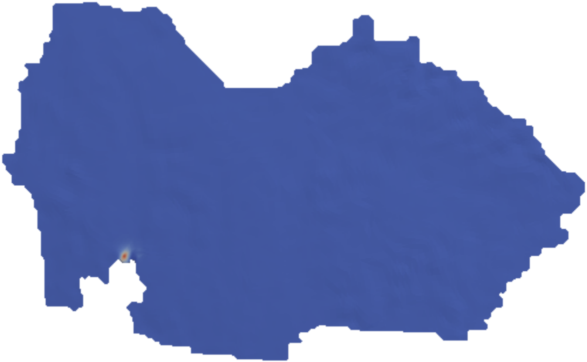
Hessian impulse responses (Pine Island Glacier)



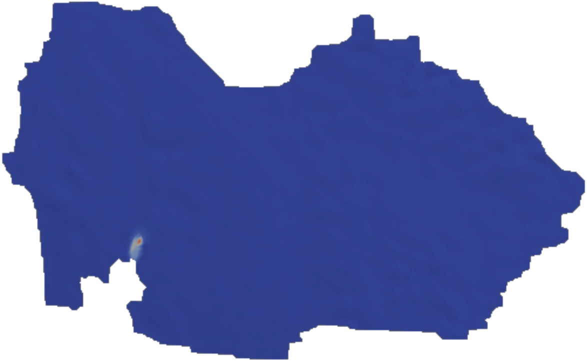
Hessian impulse responses (Pine Island Glacier)



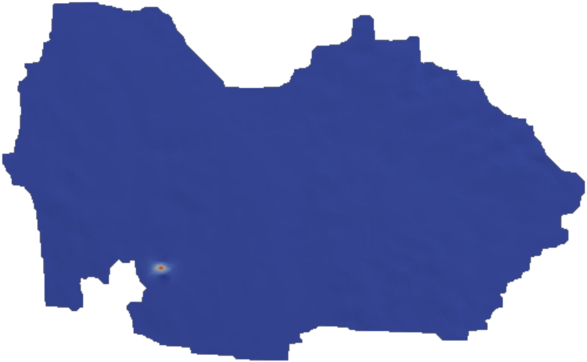
Hessian impulse responses (Pine Island Glacier)



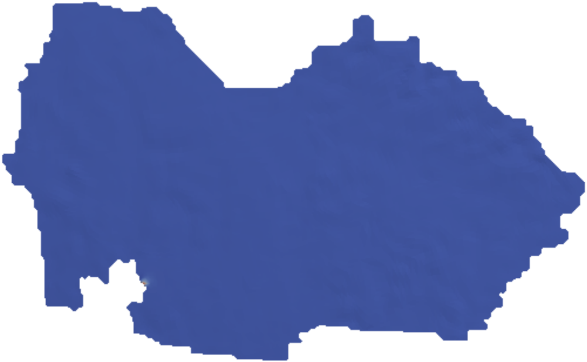
Hessian impulse responses (Pine Island Glacier)



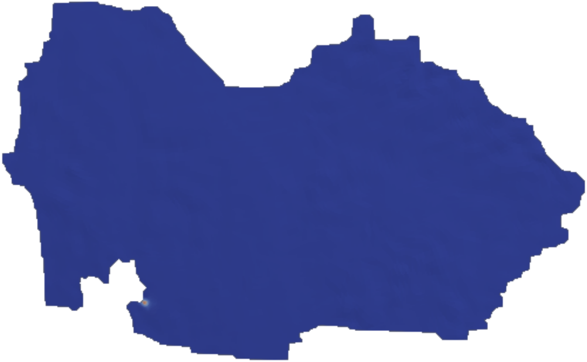
Hessian impulse responses (Pine Island Glacier)



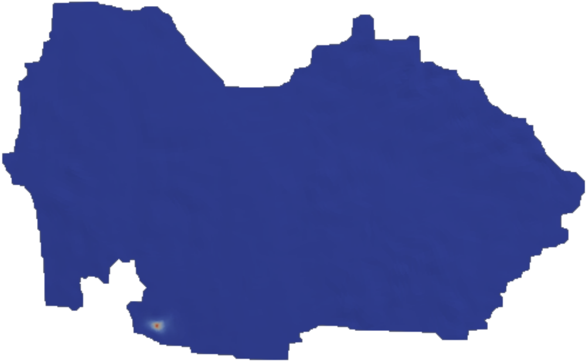
Hessian impulse responses (Pine Island Glacier)



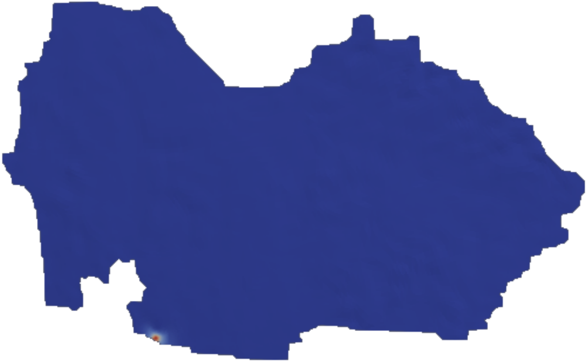
Hessian impulse responses (Pine Island Glacier)



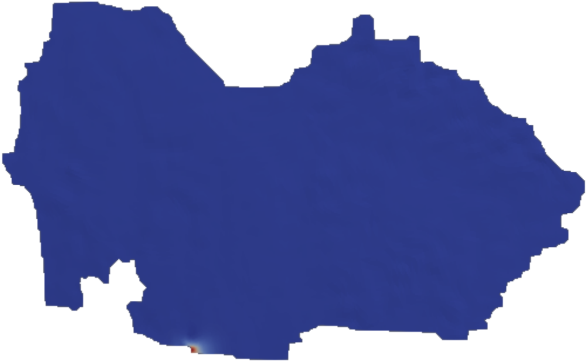
Hessian impulse responses (Pine Island Glacier)



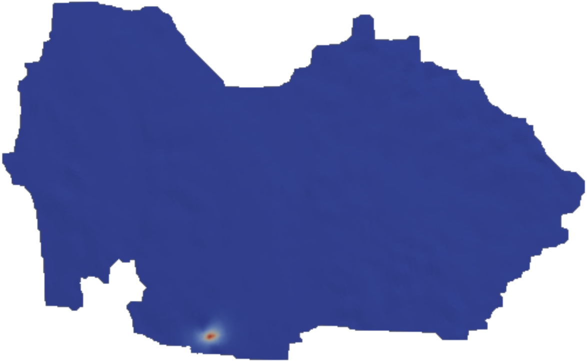
Hessian impulse responses (Pine Island Glacier)



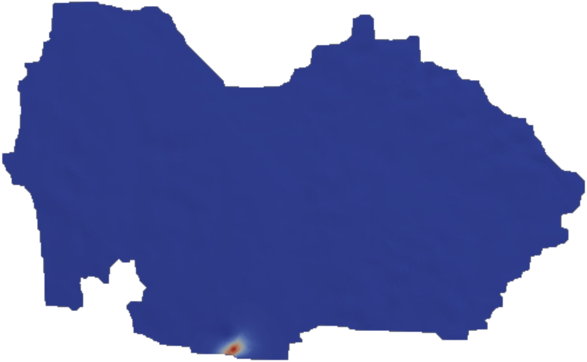
Hessian impulse responses (Pine Island Glacier)



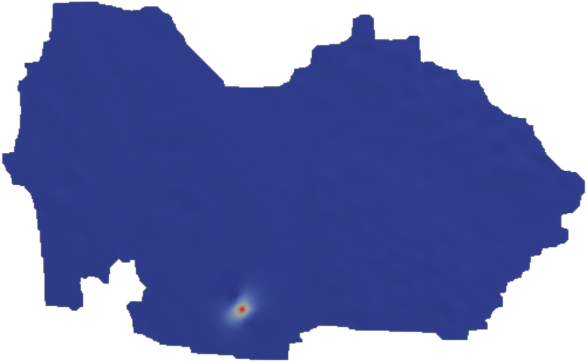
Hessian impulse responses (Pine Island Glacier)



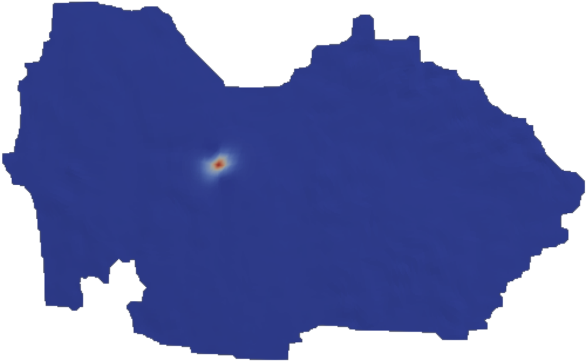
Hessian impulse responses (Pine Island Glacier)



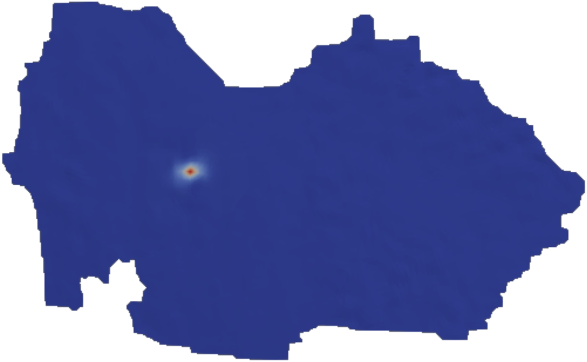
Hessian impulse responses (Pine Island Glacier)



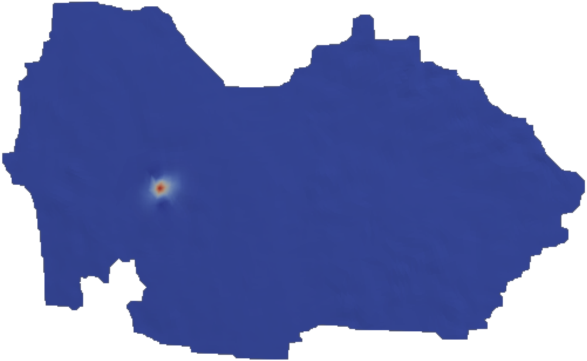
Hessian impulse responses (Pine Island Glacier)



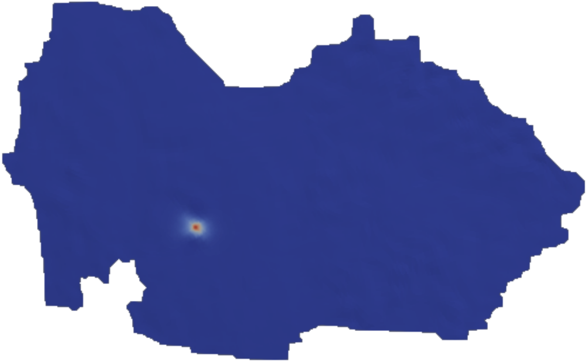
Hessian impulse responses (Pine Island Glacier)



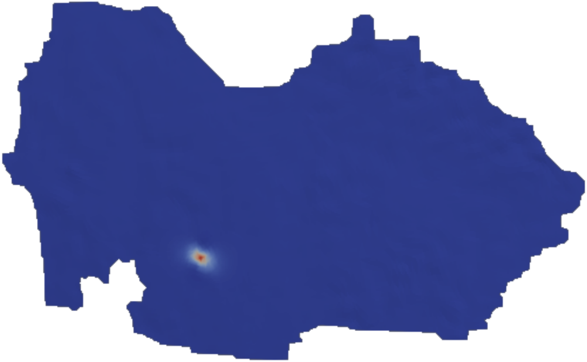
Hessian impulse responses (Pine Island Glacier)



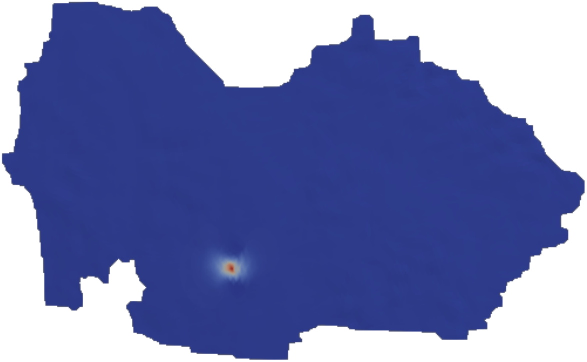
Hessian impulse responses (Pine Island Glacier)



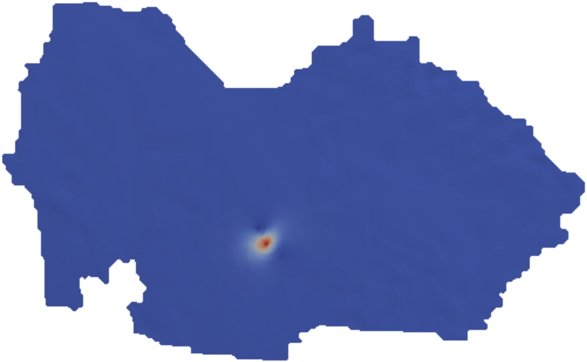
Hessian impulse responses (Pine Island Glacier)



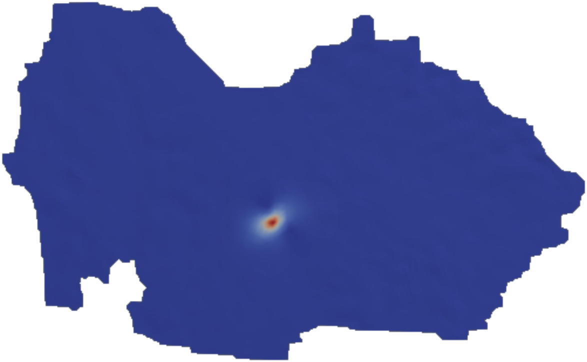
Hessian impulse responses (Pine Island Glacier)



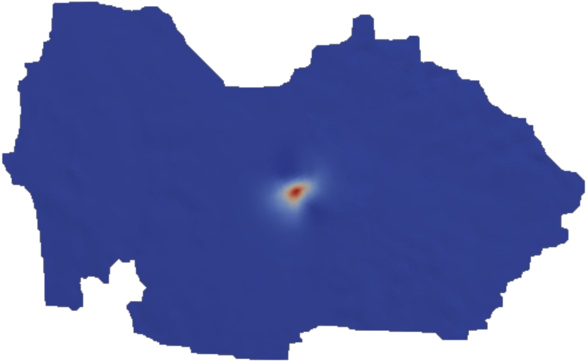
Hessian impulse responses (Pine Island Glacier)



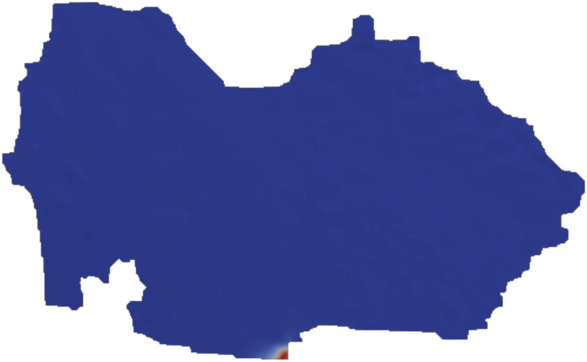
Hessian impulse responses (Pine Island Glacier)



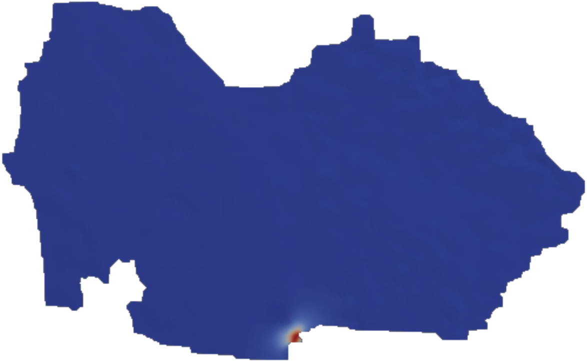
Hessian impulse responses (Pine Island Glacier)



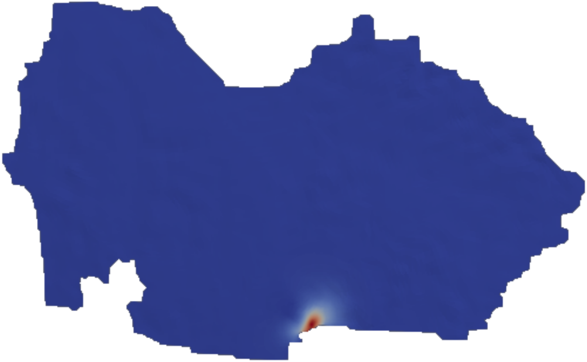
Hessian impulse responses (Pine Island Glacier)



Hessian impulse responses (Pine Island Glacier)

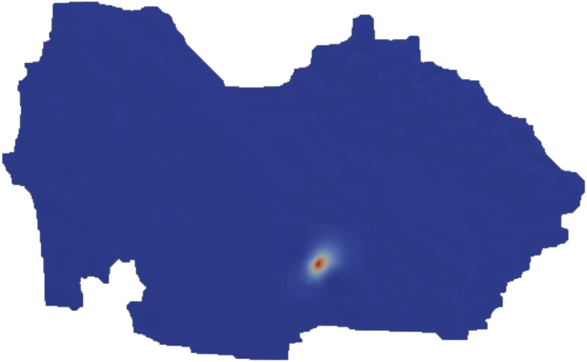


Hessian impulse responses (Pine Island Glacier)

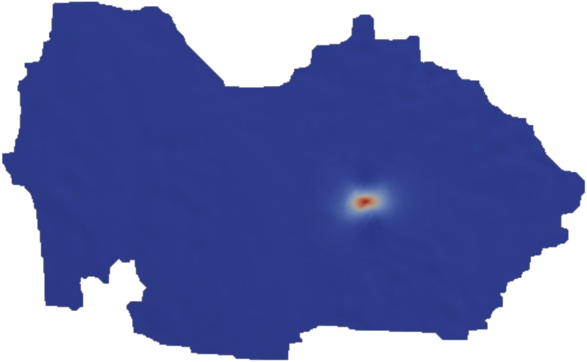


Hessian impulse responses (Pine Island Glacier)

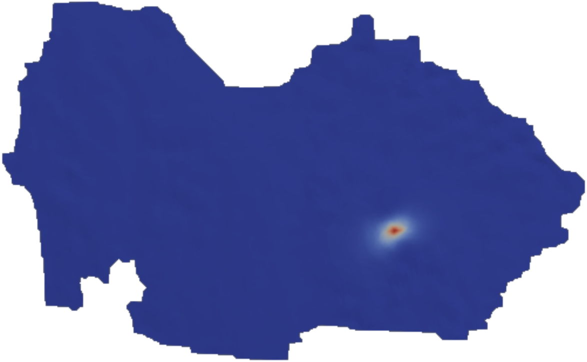
Hessian impulse responses (Pine Island Glacier)



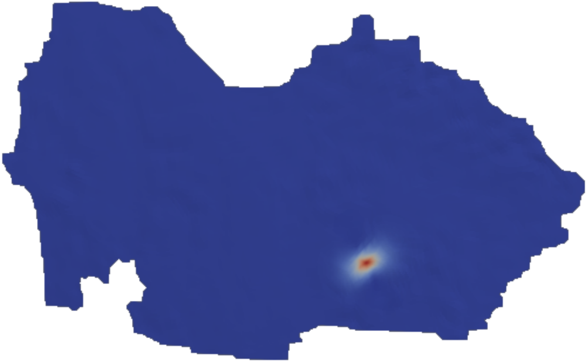
Hessian impulse responses (Pine Island Glacier)



Hessian impulse responses (Pine Island Glacier)

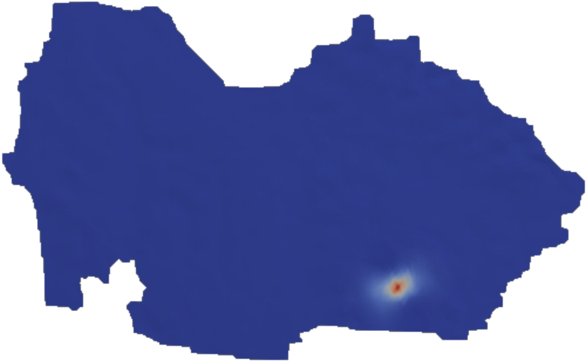


Hessian impulse responses (Pine Island Glacier)



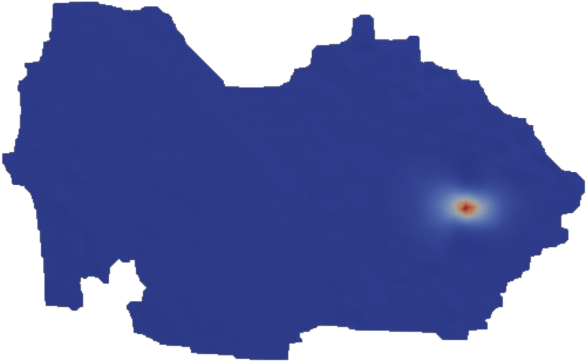
Hessian impulse responses (Pine Island Glacier)

Hessian impulse responses (Pine Island Glacier)

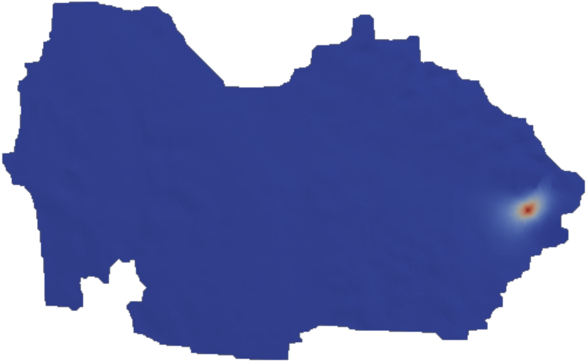


Hessian impulse responses (Pine Island Glacier)

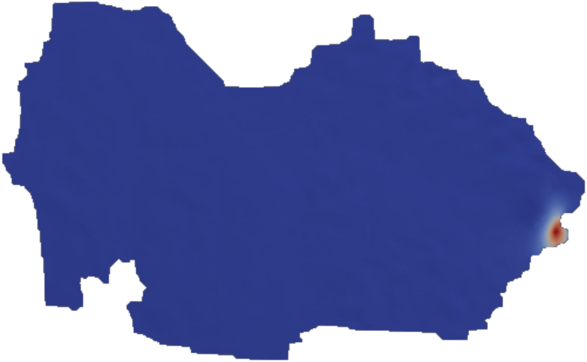
Hessian impulse responses (Pine Island Glacier)



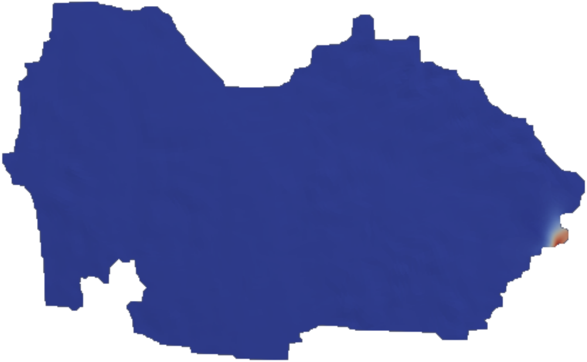
Hessian impulse responses (Pine Island Glacier)



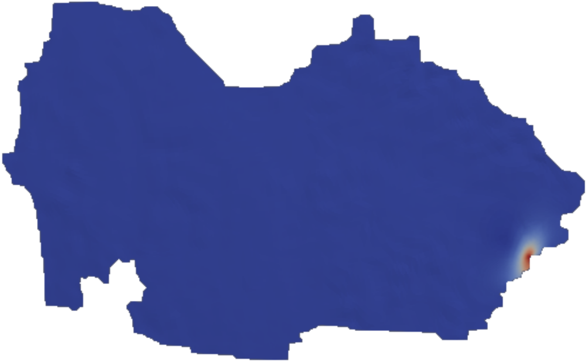
Hessian impulse responses (Pine Island Glacier)



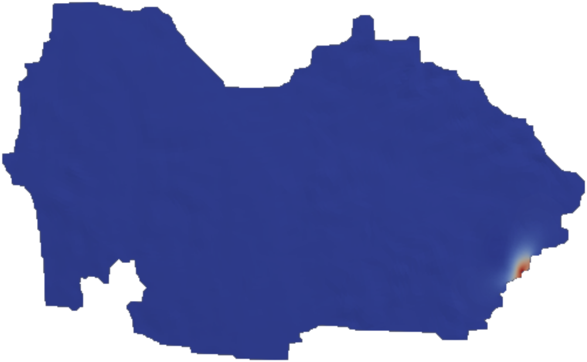
Hessian impulse responses (Pine Island Glacier)



Hessian impulse responses (Pine Island Glacier)



Hessian impulse responses (Pine Island Glacier)

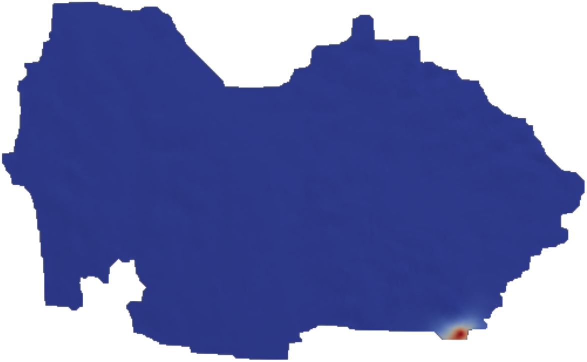


Hessian impulse responses (Pine Island Glacier)

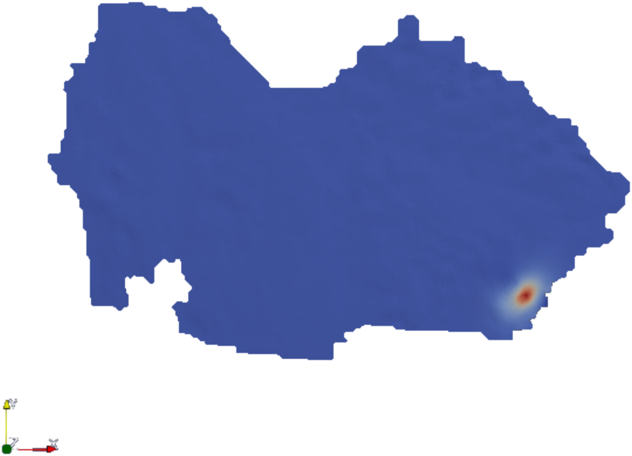
Hessian impulse responses (Pine Island Glacier)

Hessian impulse responses (Pine Island Glacier)

Hessian impulse responses (Pine Island Glacier)



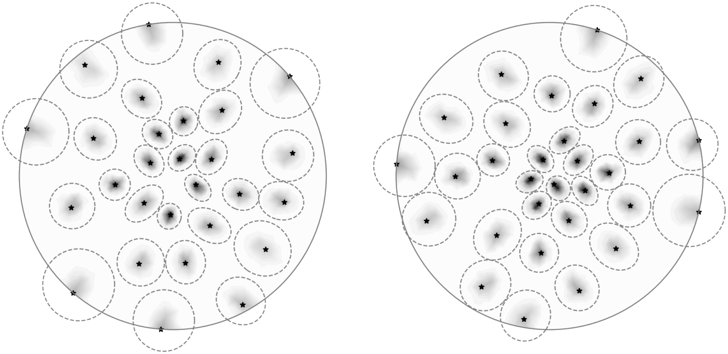
Hessian impulse responses (Pine Island Glacier)



Hessian impulse responses (Pine Island Glacier)

Hessian approximation method: big idea

- **Step 1:** Compute "batches" of impulse responses by applying Hessian to Dirac combs
- **Step 2:** Interpolate known impulse responses to approximate unknown Hessian entries $\mathbf{H}_{ij}$
- **Step 3:** Convert to $\mathcal{H}$-matrix to do linear algebra



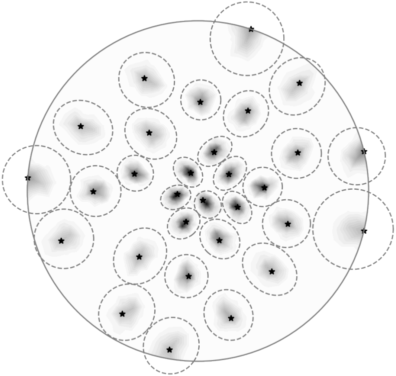
*Impulse response batches from Ice Mountain

Technical details

- How do we choose the impulse response points?
  - How do we make sure they don't overlap?
- How do we interpolate the impulse responses?
  - What about boundary issues?

How to choose impulse response points?

One hessian matrix-vector product $\rightarrow$ many impulse responses



- **Goal:** choose as many points as possible, such that the impulse response supports don't overlap
- **Dilemma:** How can we know the impulse response supports before we compute them?

Matrix analogy: getting all row sums

Matrix: let $\mathbf{A} \in \mathbb{R}^{N \times N}$. Then

$$\mathbf{A}^T \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} \text{sum of } \mathbf{A} \text{ col } 1 \\ \text{sum of } \mathbf{A} \text{ col } 2 \\ \vdots \\ \text{sum of } \mathbf{A} \text{ col } N \end{bmatrix}$$

Apply matrix to vector of ones $\rightarrow$ get row sums for all rows

Operator: let $C(x) = 1$ be the constant function. Then

$$(H_d^T C)^*(y) = \int_\Omega (H_d \delta_y)(x) dx$$

Apply Hessian to constant function $\rightarrow$ get volumes of every impulse response

Mean and standard deviations of impulse responses

- Let $V(x)$, $\mu(x)$, and $\Sigma(x)$ be the "volume", "mean", and "variance" of ϕ_x

- Let C, L^i, and Q^{ij} be the following functions:

$$C(x) := 1, \qquad L^i(x) := x^i, \qquad Q^{ij}(x) = x^i x^j$$

- Then

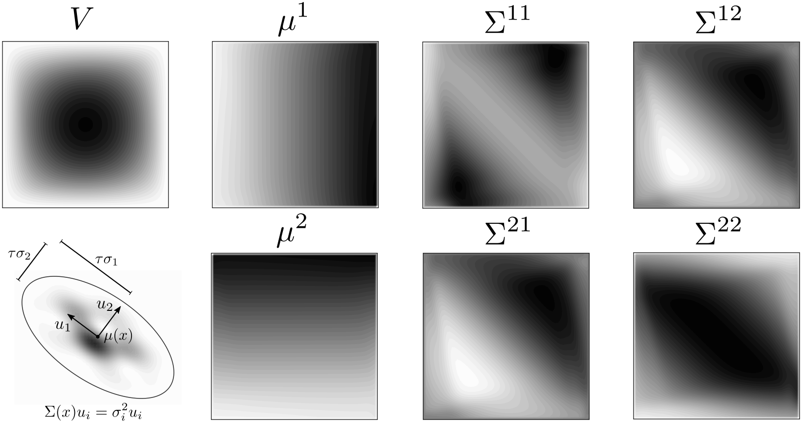
$$\begin{aligned} V &= \left(H_d^T C\right)^* \\ \mu^i &= \left(H_d^T L^i\right)^* / V \\ \Sigma^{ij} &= \left(H_d^T Q^{ij}\right)^* / V - \mu^i \cdot \mu^j \end{aligned}$$

- Apply Hessian to constant, linear, and quadratic functions $\rightarrow$ get estimates of support for every impulse response

Impulse response moments

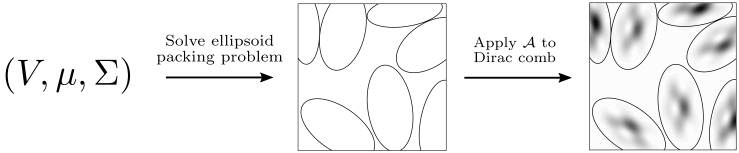
ϕ_x is **approximately supported** in the ellipsoid:

$$E = \{y : (y - \mu(x))^T \Sigma(x)^{-1} (y - \mu(x)) \leq \tau^2\}$$

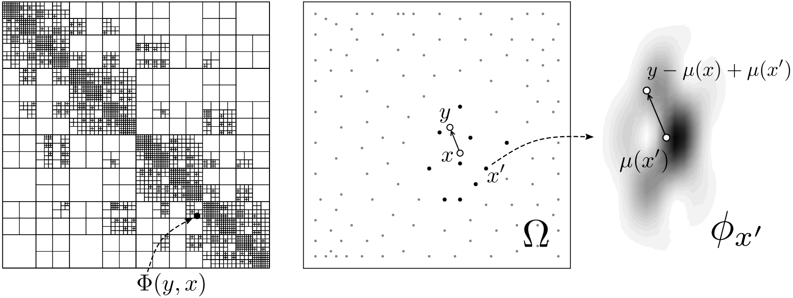


Impulse response batches via ellipsoid packing

- Picking impulse response points becomes an **ellipsoid packing problem**
- Pack ellipsoids using **greedy algorithm**
- Get batch of impulse responses by applying Hessian to **Dirac comb** (weighted sum of delta functions associated with ellipsoids)
- **Repeat** to get more batches



Radial basis function interpolation



- Must compute $O(N\log N)$ kernel entries $\Phi(y,x)$ to construct H-matrix
- For each entry, interpolate impulse responses using radial basis functions.
- Use only k-nearest neighbors (must solve $k\times k$ linear system)

Computational cost

- **Hessian-vector products:**

$$6 + n_{\text{batches}}$$

E.g., 11 Hessian-vector products for 5 batches of impulse responses

- **Ellipsoid intersection tests:**

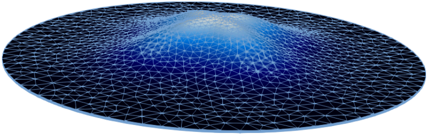
$$O(Nm)$$

where m is total number of impulse responses in all batches

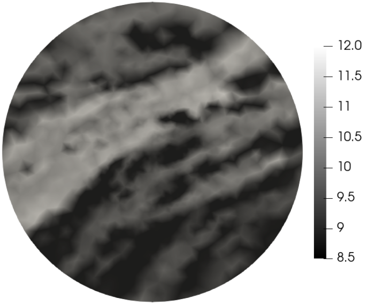
- **Elementary operations** to build and use the H-matrix:

$$O(N \log N)$$

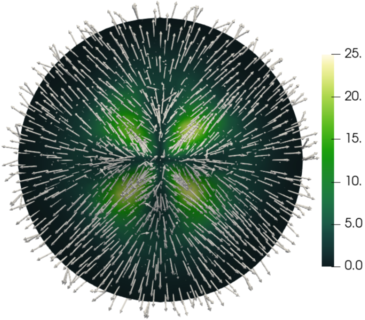
Ice Mountain: Setup



(a) Ice sheet model geometry



(b) β_{true}



(c) v_{true}

Ice Mountain: Reconstructions

(a) 25% noise (b) 5% noise (c) 1% noise

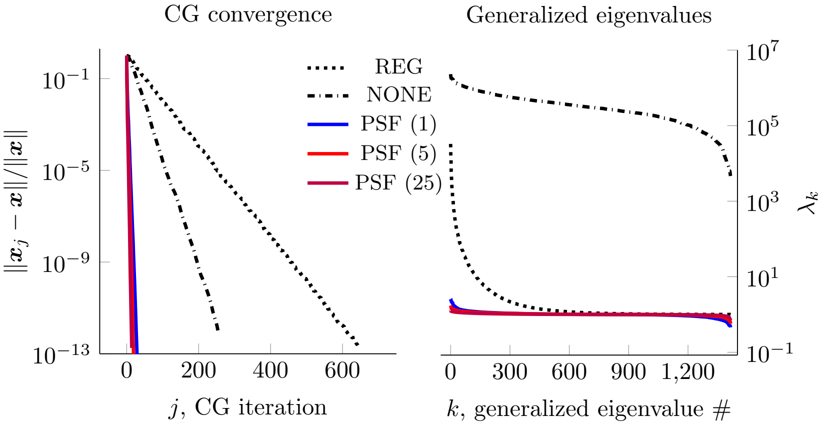
- Deterministic reconstruction / MAP point for varying noise levels
- Bi-Laplacian regularization / prior
- Regularization / “prior” strength chosen via Morozov discrepancy principle

Ice Mountain: inexact Newton-CG convergence

- 5% noise
- Preconditioner build at 3rd Newton iteration and re-used for all subsequent iterations
- PSF (5): our preconditioner with 5 batches.
- REG: regularization preconditioning.
- NONE: no preconditioning.

	PSF (5)			REG			NONE		
Iter	#CG	#Stokes	$\|\mathbf{g}\|$	#CG	#Stokes	$\|\mathbf{g}\|$	#CG	#Stokes	$\|\mathbf{g}\|$
0	1	4	1.9e+7	3	8	1.9e+7	1	4	1.9e+7
1	2	6	6.1e+6	8	18	8.4e+6	2	6	6.1e+6
2	4	10	2.6e+6	16	34	4.1e+6	4	10	2.6e+6
3	2	6+22	6.9e+5	34	70	1.8e+6	14	30	6.9e+5
4	3	8	4.4e+4	52	106	5.6e+5	29	60	1.3e+5
5	5	12	2.2e+3	79	160	9.4e+4	38	78	1.0e+4
6	0	2	1.1e+1	102	206	6.5e+3	58	118	1.8e+2
7	—	—	—	151	304	1.2e+2	0	2	5.5e-1
8	—	—	—	0	2	2.9e-1	—	—	—
Total	17	70	—	445	908	—	146	308	—

Ice Mountain: preconditioned Hessian spectral properties



- Hessian evaluated at MAP point for 5% noise.
- Left: solving $\mathbf{H}\mathbf{x} = -\mathbf{b}$ via preconditioned conjugate gradient
- Right: eigenvalues for generalized eigenvalue problem $\mathbf{H}\mathbf{u} = \lambda\widetilde{\mathbf{H}}\mathbf{u}$

Ice Mountain: preconditioned Hessian condition number

noise level	COND($\widetilde{\mathbf{H}}^{-1}\mathbf{H}$)				
	REG	NONE	PSF (1)	PSF (5)	PSF (25)
25%	1.01e+3	2.96e+3	1.34e+0	1.30e+0	1.18e+0
11%	7.40e+3	1.05e+3	2.27e+0	1.55e+0	1.31e+0
5.0%	3.29e+4	4.96e+2	5.61e+0	3.06e+0	1.92e+0
2.2%	1.66e+5	8.89e+2	1.58e+1	8.07e+0	4.03e+0
1.0%	5.36e+5	1.61e+3	7.17e+1	1.93e+1	9.19e+0

Summary

- Hessian approximations or preconditioners are essential for solution of Bayesian inverse problems governed by partial differential equations.

- Low-rank approximations of the Hessian become prohibitive as the data becomes more informative, as is the case for ice sheet inverse problems.

- Local point spread function interpolation combined with Hierarchical matrix representations promise a more efficient Hessian approximation.

Alger, N., Hartland, T., Petra, N., Ghattas, O. (2023). Point spread function approximation of high rank Hessians with locally supported non-negative integral kernels. To appear in SISC.