Quasi-product-convolution Hessian approximation for ice sheet inverse problems

Nick Alger[1]

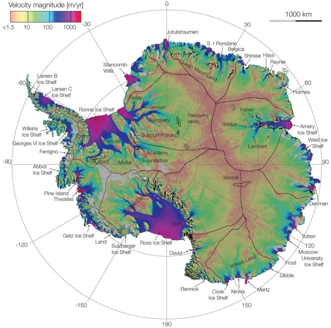
Joint work with:
Noémi Petra,[2] Tucker Hartland,[2] Omar Ghattas[1]

[1]Oden Institute
The University of Texas at Austin

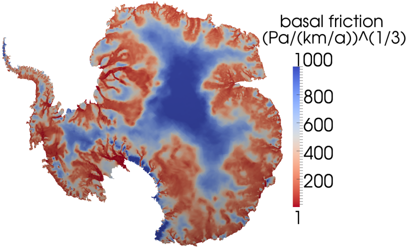
[2]Applied Mathematics, School of Natural Sciences
University of California, Merced

SIAM UQ22

Antarctic ice sheet



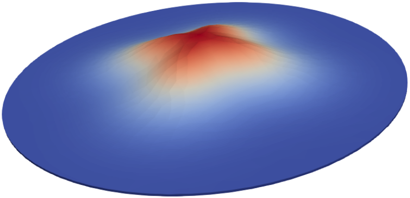
Observed surface flow velocity from InSAR (Rignot et. al, 2011)



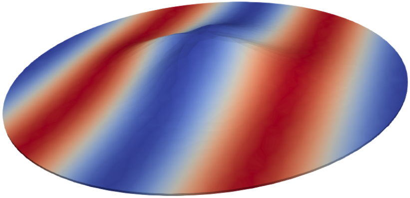
Antarctic ice sheet inversion for the basal friction parameter field from InSAR surface velocities

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).

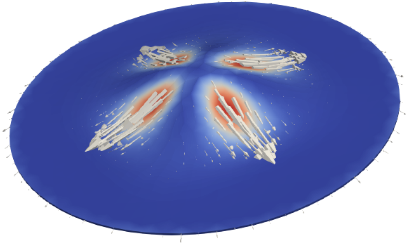
Model Problem (Ice mountain)



(a) Ice sheet elevation



(b) Basal friction



(c) Velocity

Ice sheet dynamics: forward and inverse

Balance of linear momentum, mass, and energy

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

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

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

We have: Satellite observations of surface velocity

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

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

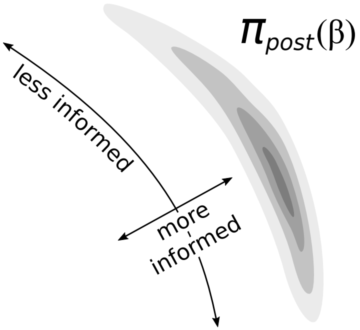
($\mathbf{T}$ is tangential component)

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.



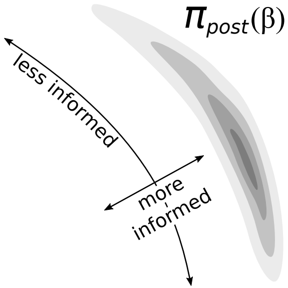
Hessian: local Gaussian approximation

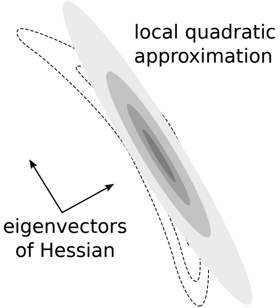
Hessian:

$$\mathbf{H} := -\frac{d^2 \log \pi_{\text{post}}}{d\beta^2}$$

Local Gaussian approximation proposal:

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





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)

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

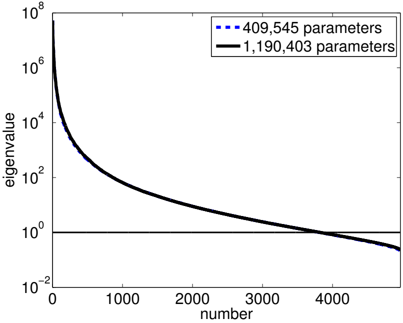
- Cost: **2** linearized Stokes **PDE solves** per matvec

Low rank Hessian approximation (extremely expensive!)

Low-rank approximation/Woodbury formula:

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

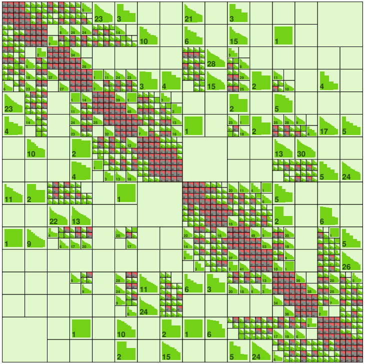
where $\boldsymbol{V}_r$ and $\boldsymbol{\Lambda}_r$ are the eigenvectors/values of $\boldsymbol{H}_d \boldsymbol{v}_i = \lambda_i \mathbf{H}_r \boldsymbol{v}_i$



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)

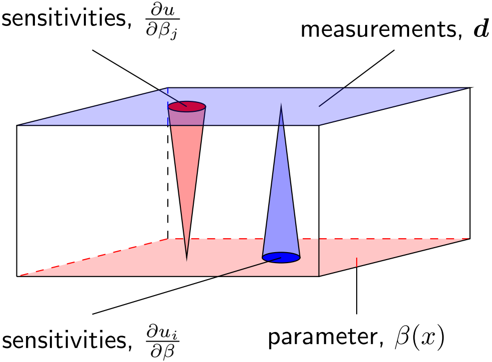
- Matrix is reordered and subdivided recursively
- Many off-diagonal blocks are low-rank
- Overall matrix may be high rank
- $O(N\,(\log N)^a)$ complexity matrix operations, $a \in \{0,1,2,3\}$
  - matrix-vector products, matrix-matrix addition, matrix-matrix multiplication, matrix factorization, matrix 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
  - T. Hartland, G. Stadler, M. Perego, K. Liégeois and N. Petra. *Hierarchical off-diagonal approximation of Hessians in inverse problems*. To be submitted.
  - Tucker Hartland is giving a talk about this later
  - 3:00-3:25 Thursday, MS140, Room: Augusta F - 7th Floor
- **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***

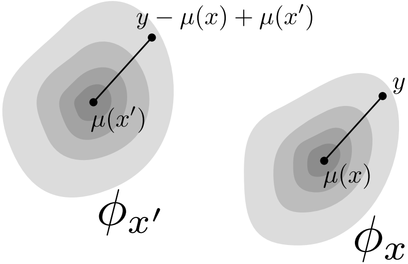
Local sensitivities



Local mean-displacement invariance

Impulse response ϕ_x:

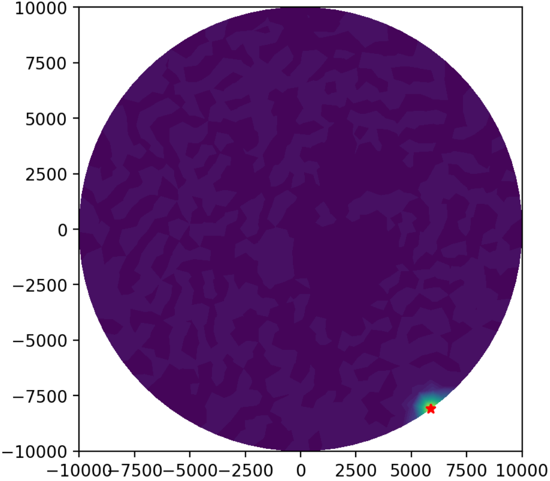
$$\phi_x := H_d\ \delta_x = \text{action of Hessian operator on delta distribution at x}$$
$$\mu(x) := \text{center of mass (mean) of } \phi_x$$



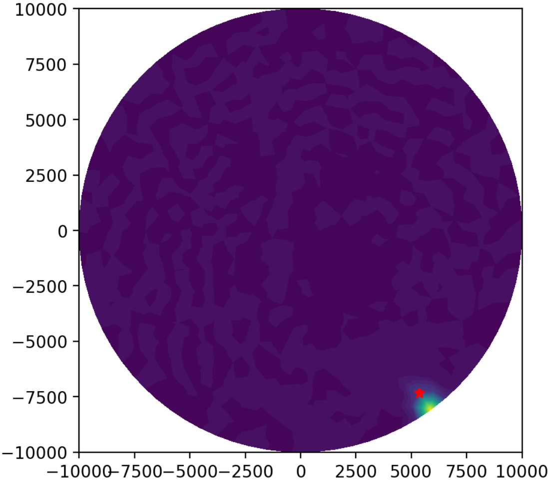
Local mean-displacement invariance:

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

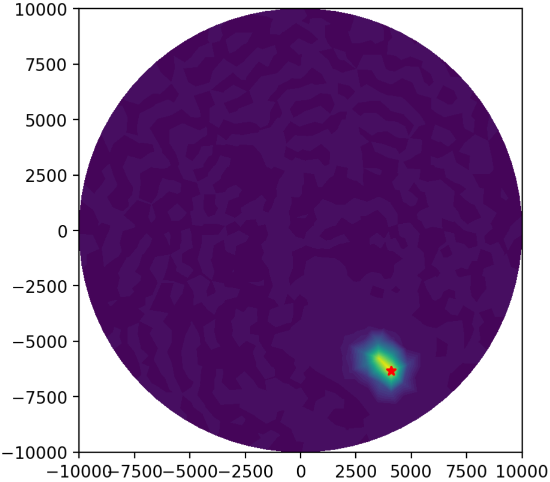
Gauss-Newton Hessian impulse responses (scaled)



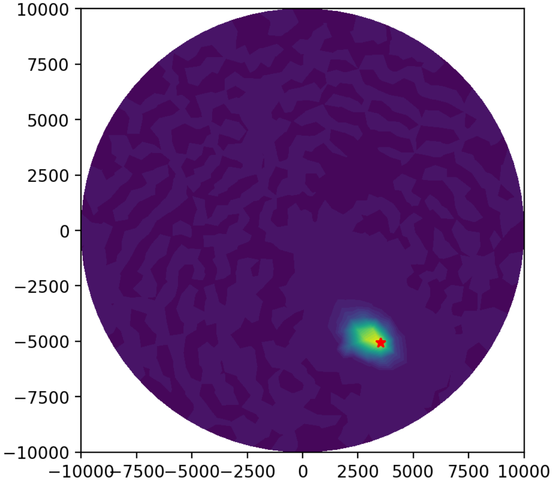
Gauss-Newton Hessian impulse responses (scaled)



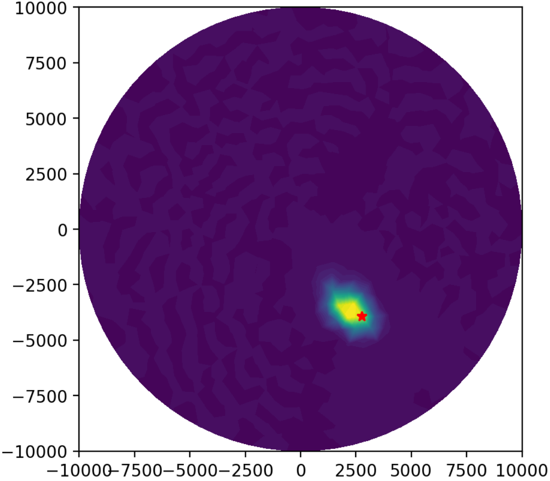
Gauss-Newton Hessian impulse responses (scaled)



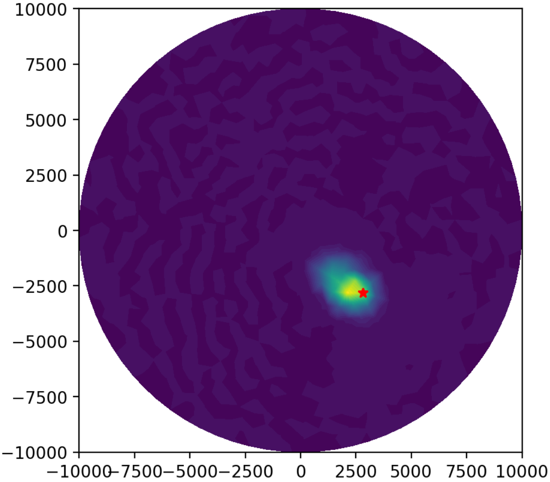
Gauss-Newton Hessian impulse responses (scaled)



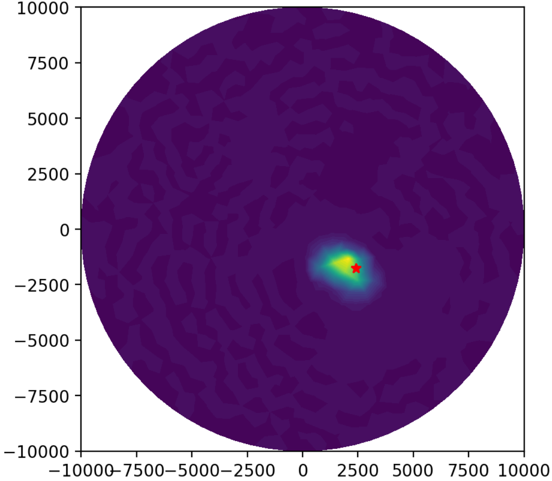
Gauss-Newton Hessian impulse responses (scaled)



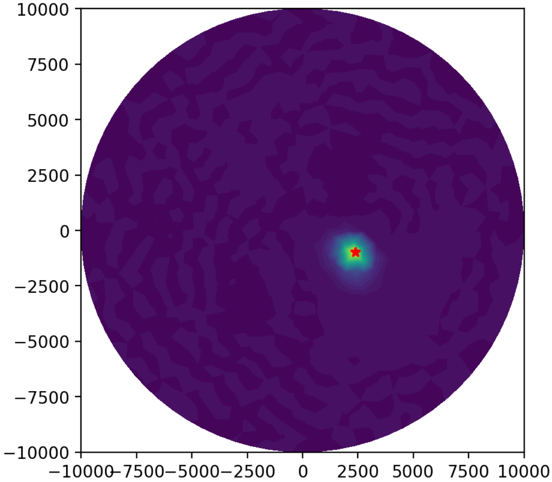
Gauss-Newton Hessian impulse responses (scaled)



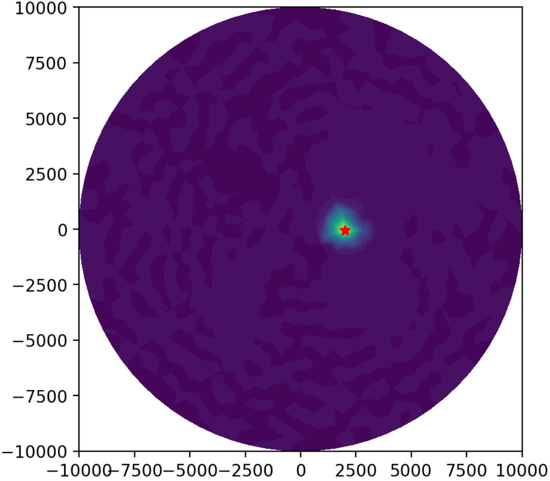
Gauss-Newton Hessian impulse responses (scaled)



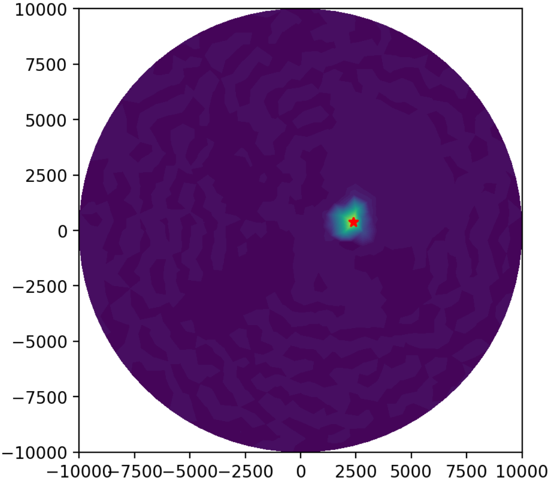
Gauss-Newton Hessian impulse responses (scaled)



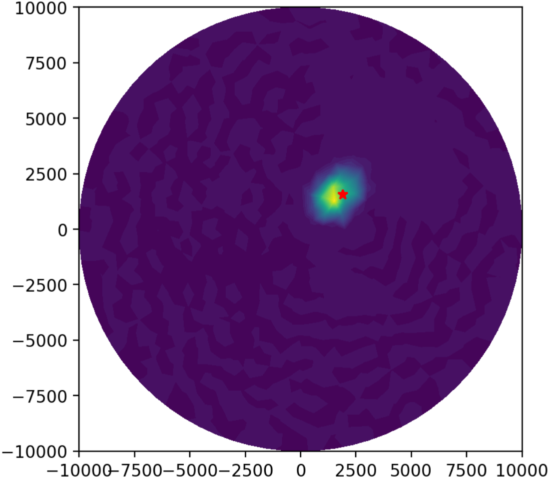
Gauss-Newton Hessian impulse responses (scaled)



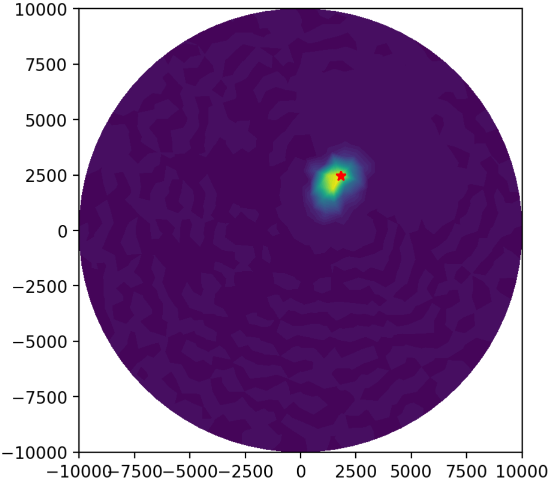
Gauss-Newton Hessian impulse responses (scaled)



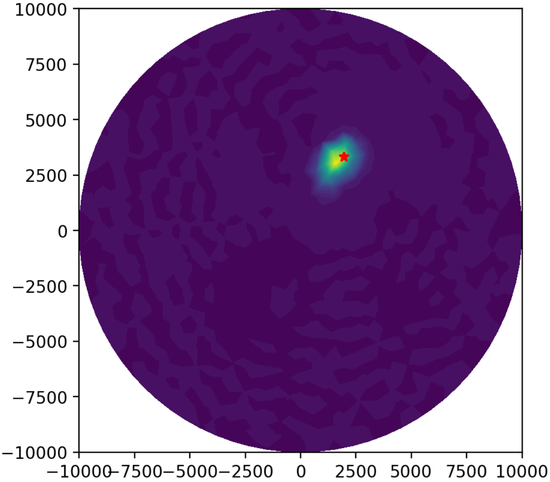
Gauss-Newton Hessian impulse responses (scaled)



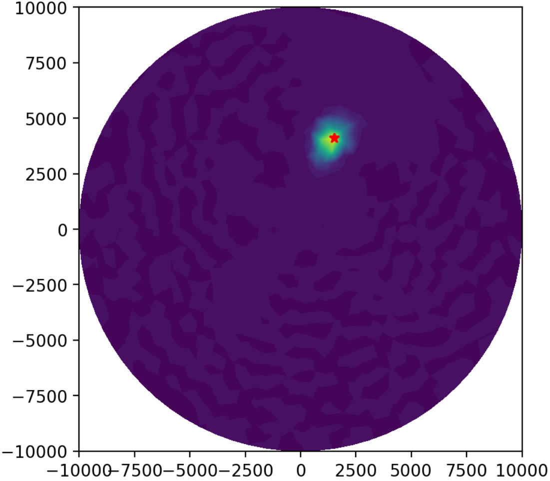
Gauss-Newton Hessian impulse responses (scaled)



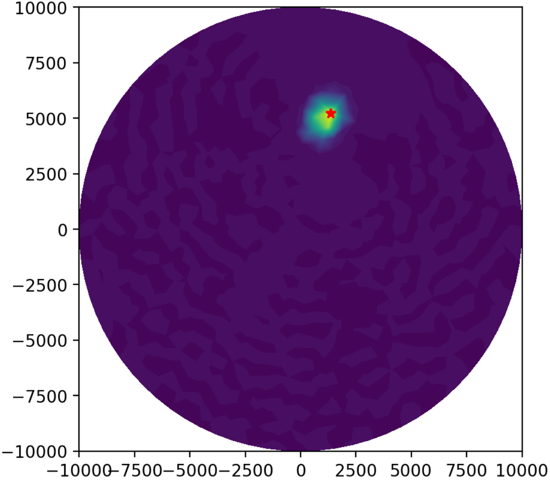
Gauss-Newton Hessian impulse responses (scaled)



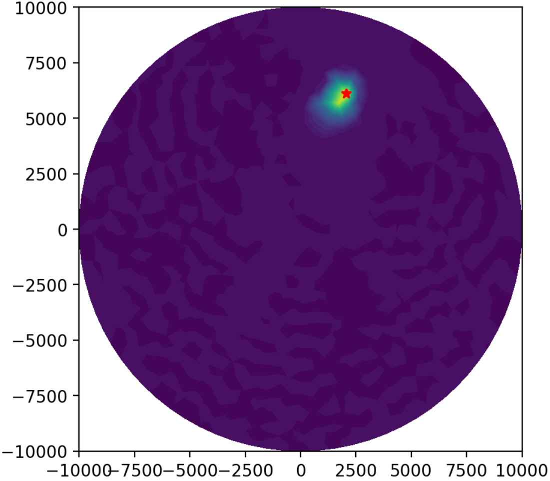
Gauss-Newton Hessian impulse responses (scaled)



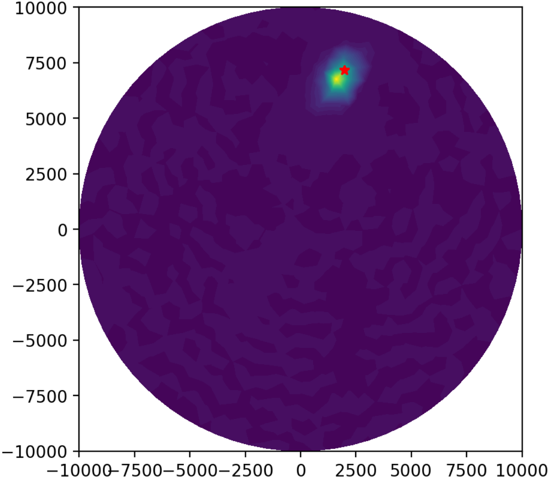
Gauss-Newton Hessian impulse responses (scaled)



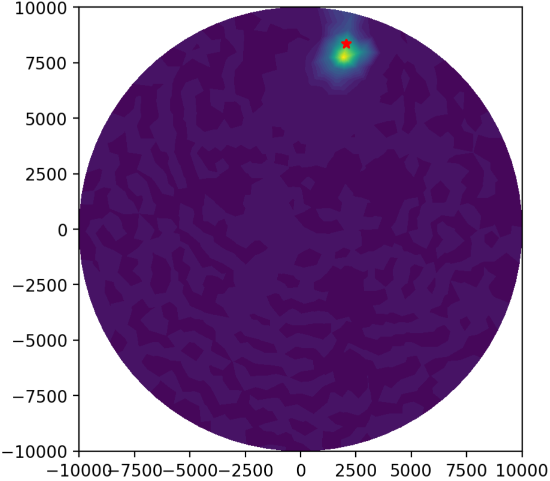
Gauss-Newton Hessian impulse responses (scaled)



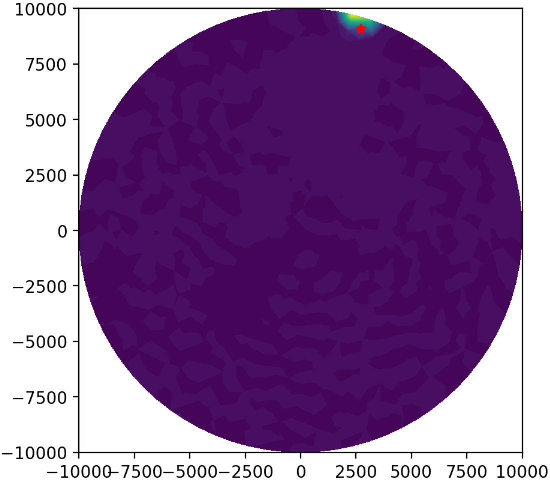
Gauss-Newton Hessian impulse responses (scaled)



Gauss-Newton Hessian impulse responses (scaled)

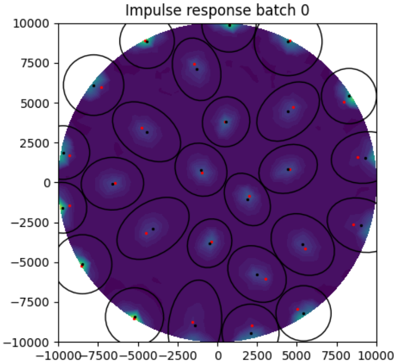


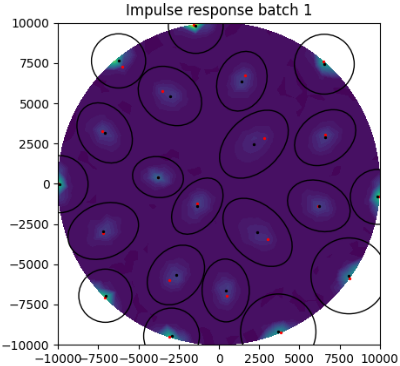
Gauss-Newton Hessian impulse responses (scaled)



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



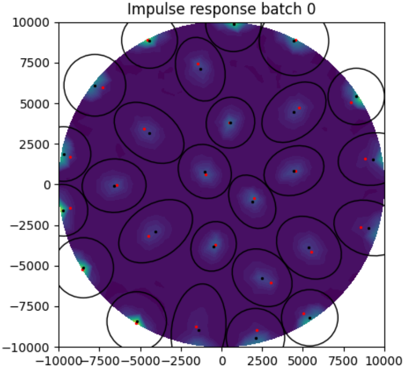


Hessian approximation: technical details

- 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

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

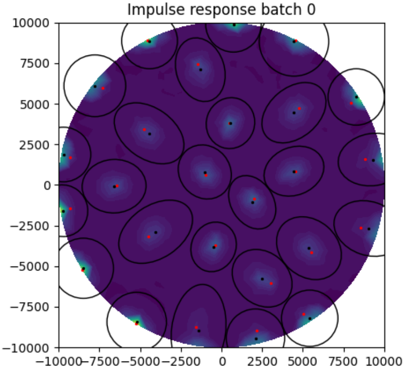
- Apply Hessian to constant, linear, and quadratic functions $\rightarrow$ get estimates of support for every impulse response

Impulse response support ellipsoids

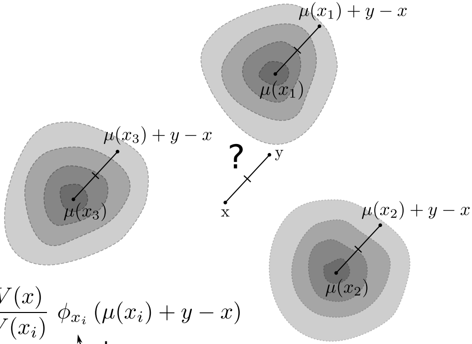
- ϕ_x is **approximately supported** in the ellipsoid

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

- Picking impulse response points becomes an **ellipsoid packing problem**
- Pack ellipsoids using **greedy algorithm**



Radial basis function interpolation



$$H_d(y, x) \approx \sum_i w_i \frac{V(x)}{V(x_i)} \; \phi_{x_i}\left(\mu(x_i) + y - x\right)$$

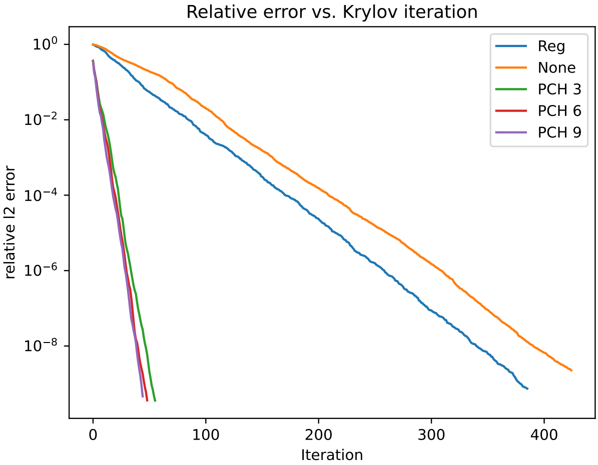
known

unknown

RBF weights

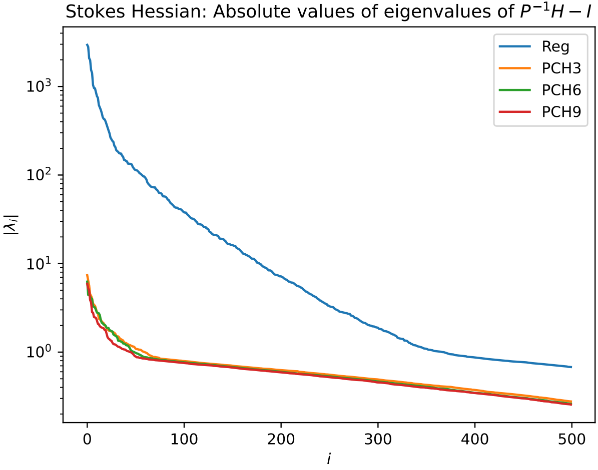
- Interpolate impulse responses using polyharmonic spline radial basis functions.
- Use only k-nearest neighbors (must solve $k \times k$ linear system)

CG Hessian solve with different preconditioners



Solving $Hp = -g$ with preconditioned conjugate gradient

Preconditioned spectrum



Summary

- Hessian approximations or preconditioners are essential for Bayesian sampling in 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 yield a highly efficient Hessian approximation.

N. Alger, T. Hartland, N. Petra, O. Ghattas. *Efficient matrix-free approximation of operators with locally supported non-negative integral kernels, with application to Hessians in PDE-constrained inverse problems*. To be submitted.