Using Bayesian inference to recover the material parameters of a heterogeneous hyperelastic body

Jack S. Hale, Patrick E. Farrell, Stéphane P. A. Bordas

UNIVERSITÉ DU LUXEMBOURG

UNIVERSITY OF OXFORD

EPSRC Engineering and Physical Sciences Research Council

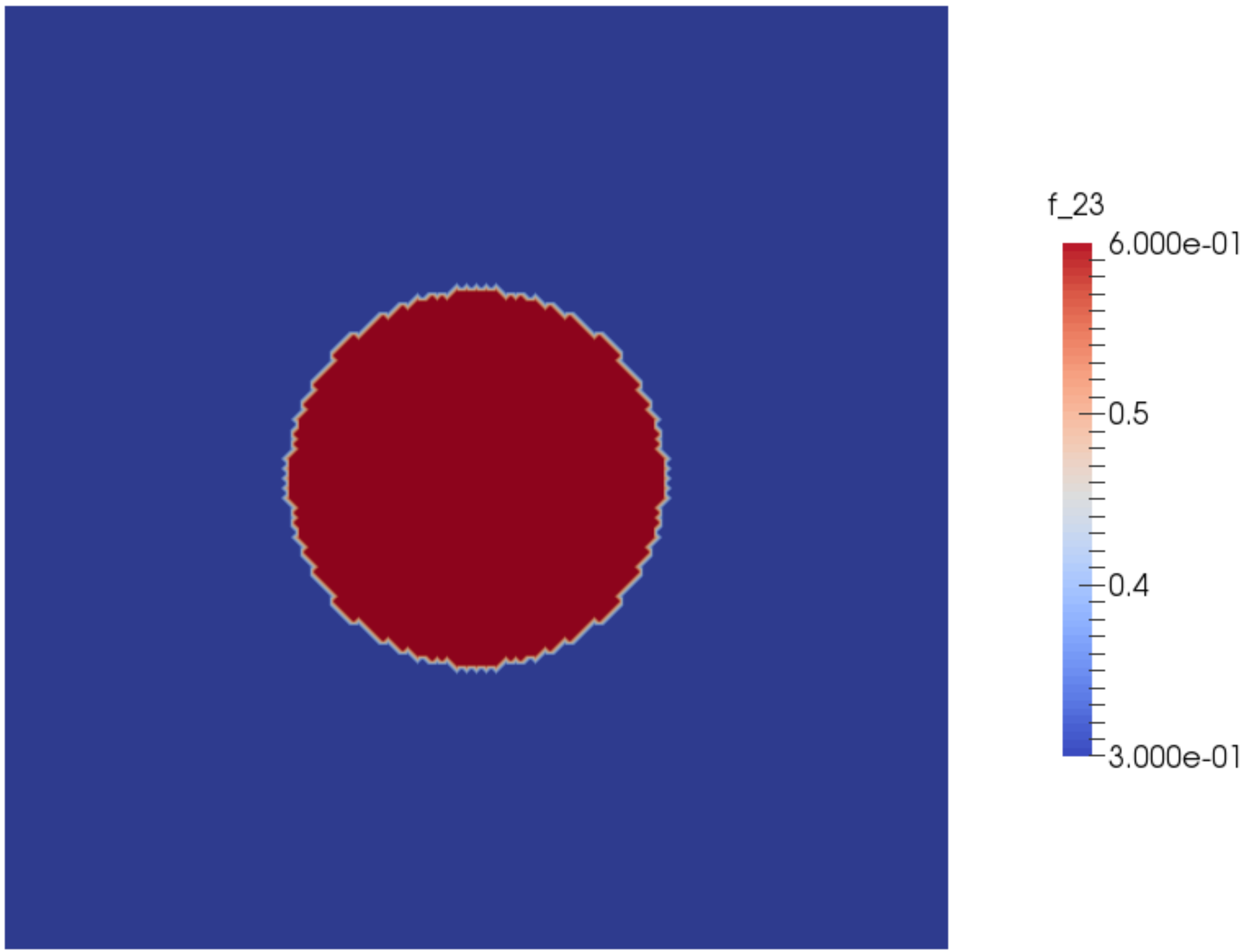
Fonds National de la Recherche Luxembourg

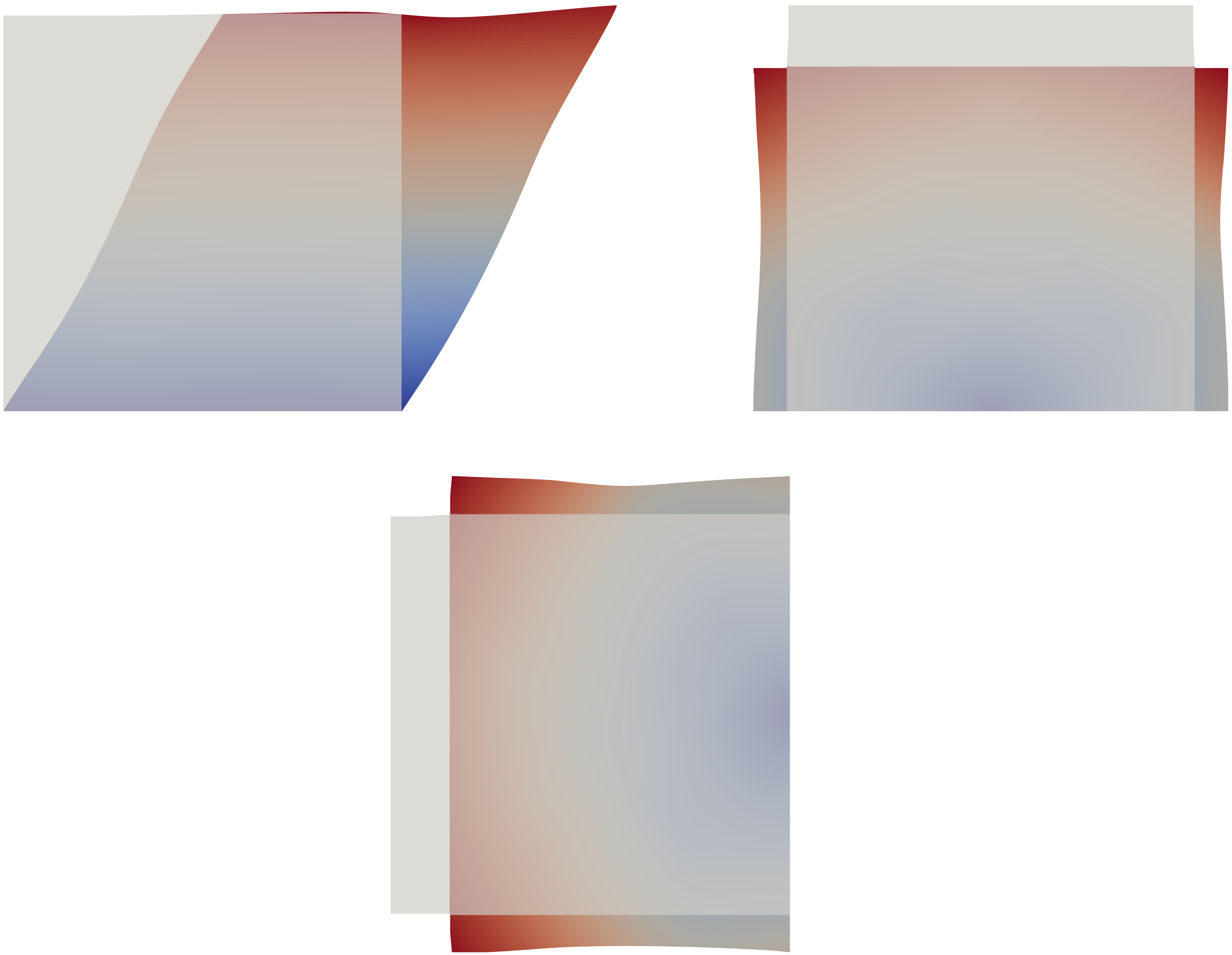
MARIE CURIE ACTIONS

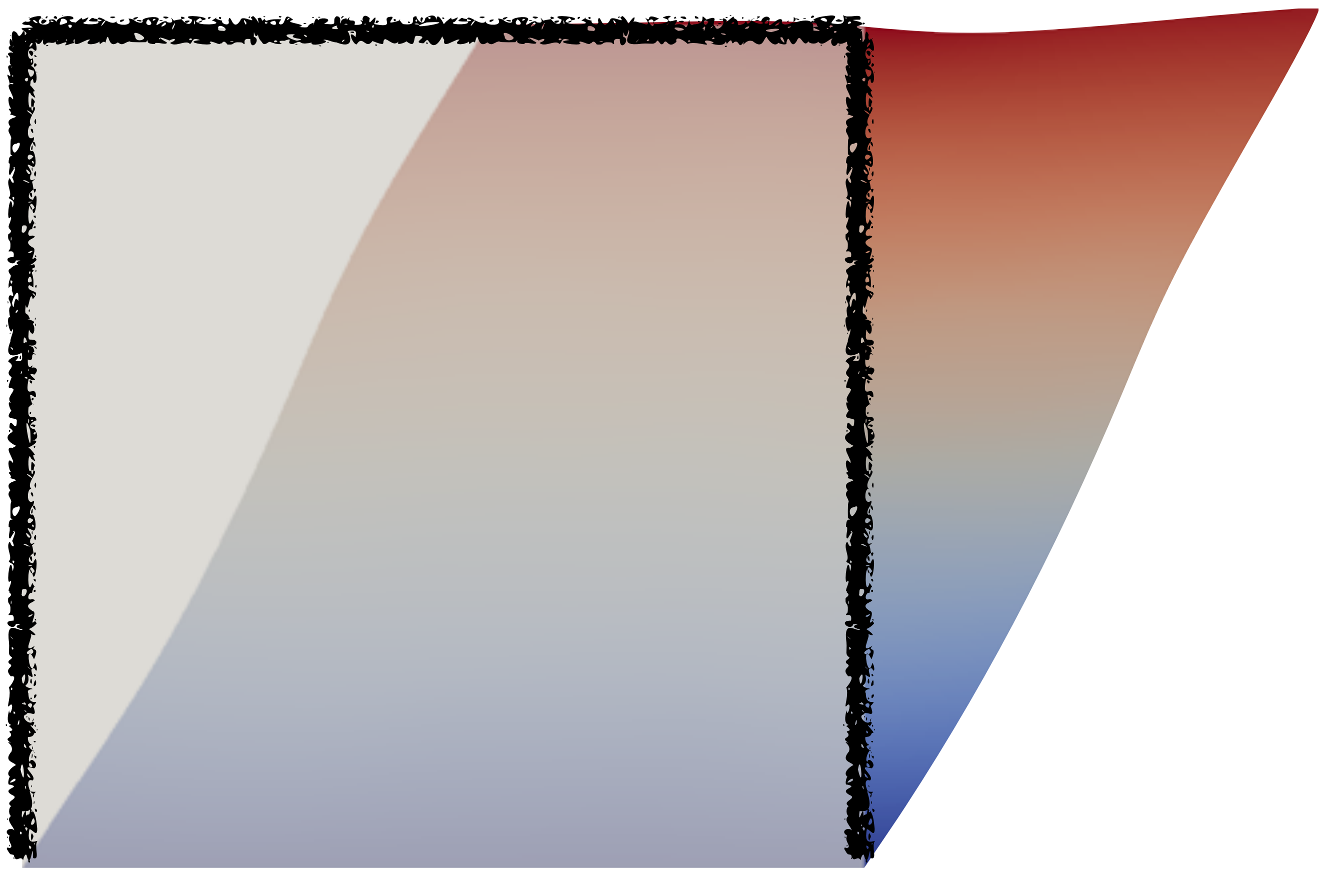
European Research Council

Overview

- Goals.
- Example uncertainty quantification problem: sparse surface observations of a geometrically non-linear hyperelastic block.
- Using dolfin-adjoint [Farrell et al.] and petsc4py [Dalcin] to solve the problem.
- Efficiently dealing with the high-dimensional posterior covariance approximation.


f_23
6.000e-01
0.5
0.4
3.000e-01





Q: What can we infer about the material parameters inside the domain, just from displacement observations on the outside?

Q: Which parameters am I most uncertain about?

Key Features

- Use Bayesian theory of inference to give statistical uncertainty quantification framework, over classical regularised PDE-constrained optimisation framework.

- dolfin-adjoint: can derive MPI-aware adjoint and higher-order adjoint equations from high-level Unified Form Language expressions.

- 'Fast' high parameter space uncertainty quantification ~ minutes for thousands of parameters on laptop.

- Parallel linear algebra solved using PETSc and SLEPc.

Terminology

- Observation. *Displacements.* y
- Parameter. *Material property.* x
- Parameter-to-observable map. *Finite deformation hyperelasticity.* $f(x)$

Bayes Theorem

$$\pi_{\text{posterior}}(x \mid y) \propto \pi_{\text{likelihood}}(y \mid x)\pi_{\text{prior}}(x)$$

Goal: Given the observations, find the posterior distribution of the unknown parameters.

Assumptions

1. I think my parameter is Gaussian (prior).

2. My parameter to observable map is linear and my noise model is Gaussian.

$$X \sim \mathcal{N}(x_0, \mathbf{\Gamma}_{\text{prior}}), X \in \mathbb{R}^n$$

$$Y = AX + E, Y \in \mathbb{R}^m, A \in \mathbb{R}^{m \times n}$$

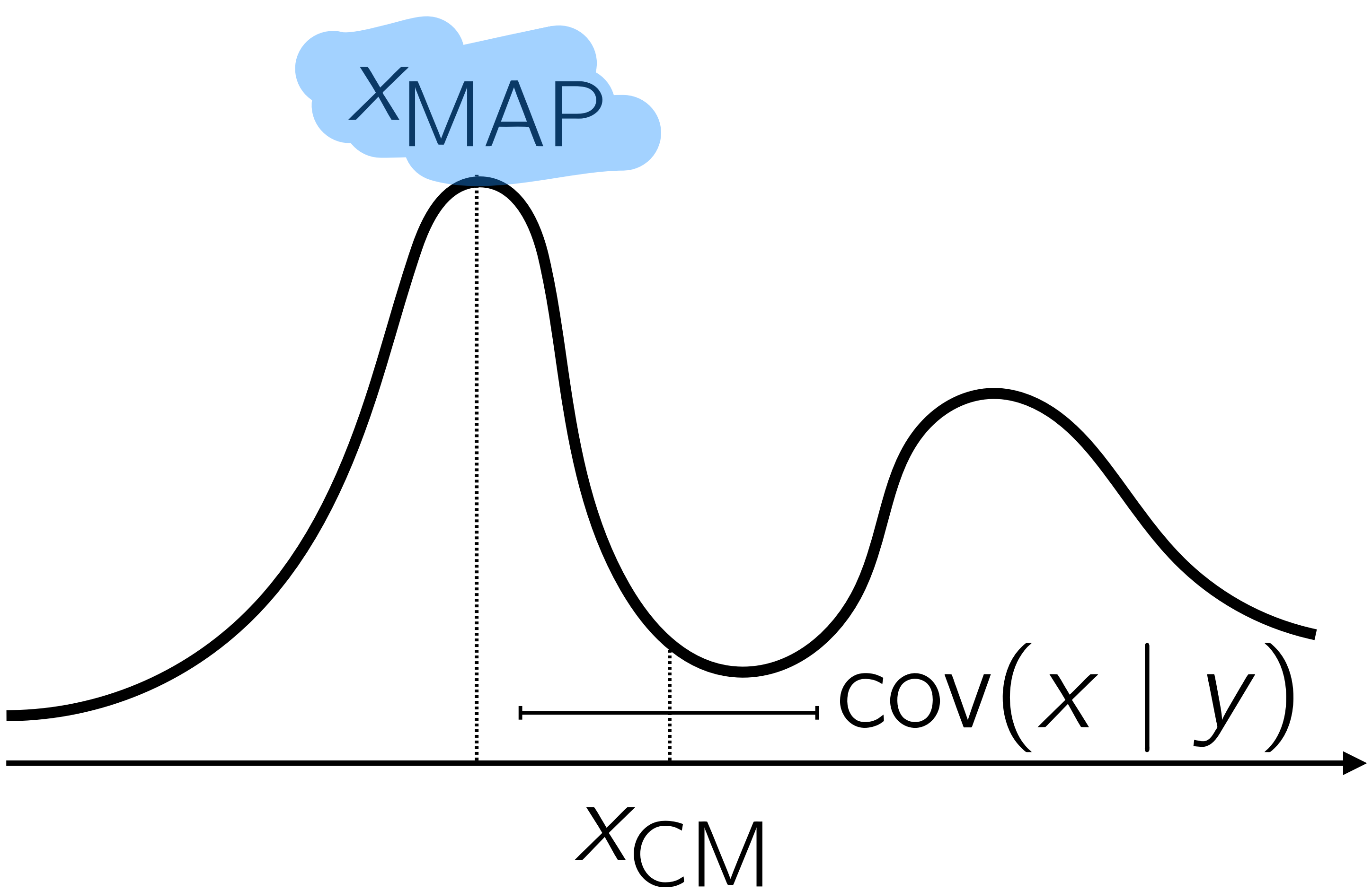
$$E \sim \mathcal{N}(0, \mathbf{\Gamma}_{\text{noise}}), Y \in \mathbb{R}^m$$

$$
\begin{aligned}
-\ln \pi_{\text{posterior}}(x \mid y) &= \frac{1}{2}\|y - \mathbf{A}x\|^2_{\mathbf{\Gamma}^{-1}_{\text{noise}}} + \frac{1}{2}\|x - x_0\|^2_{\mathbf{\Gamma}^{-1}_{\text{prior}}} \\
&= \frac{1}{2}\|x - x_{\text{MAP}}\|_{\mathbf{H}}
\end{aligned}
$$

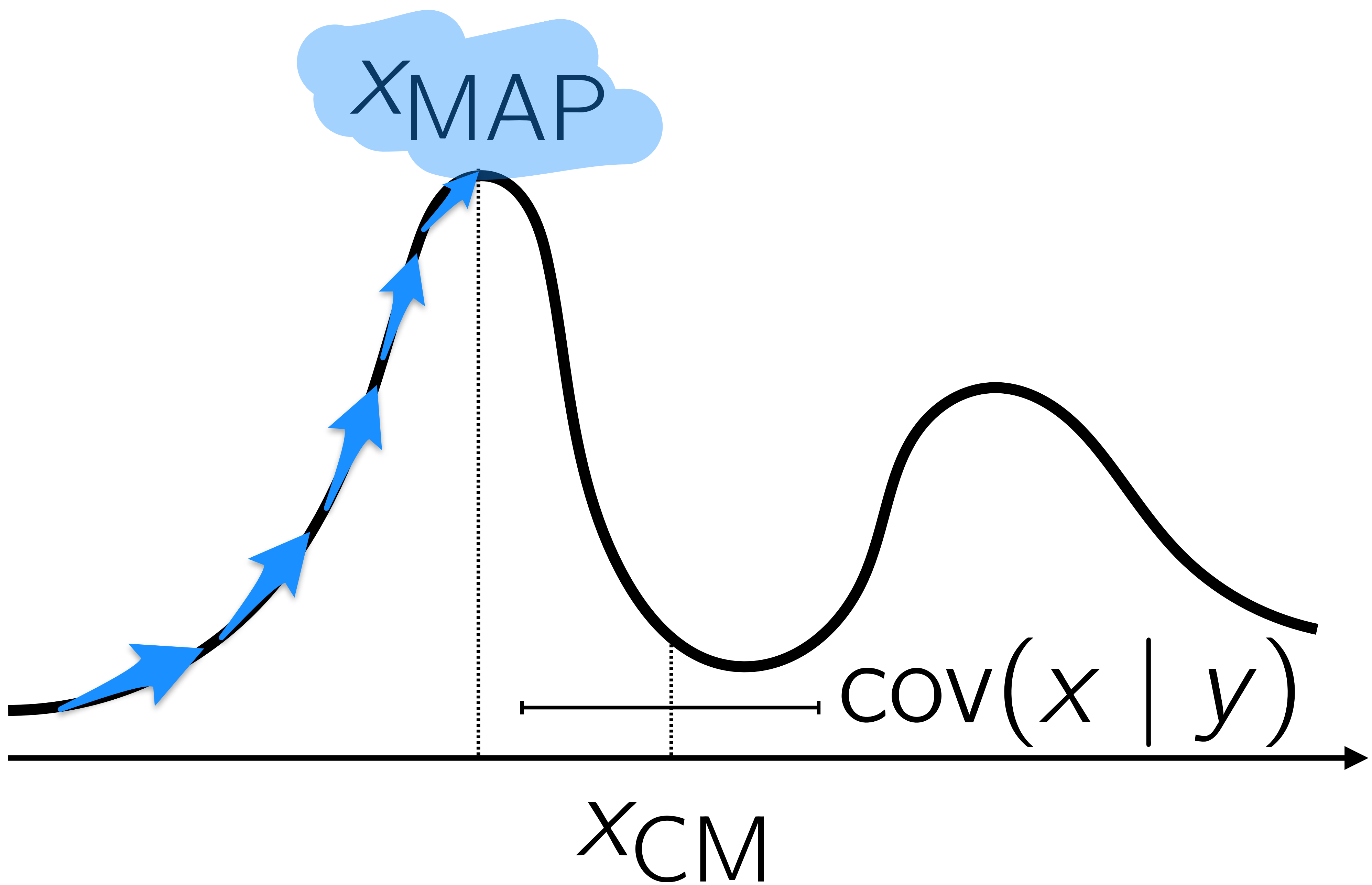
$$
\pi_{\text{posterior}} \sim \mathcal{N}(x_{\text{MAP}}, \mathbf{H}^{-1})
$$

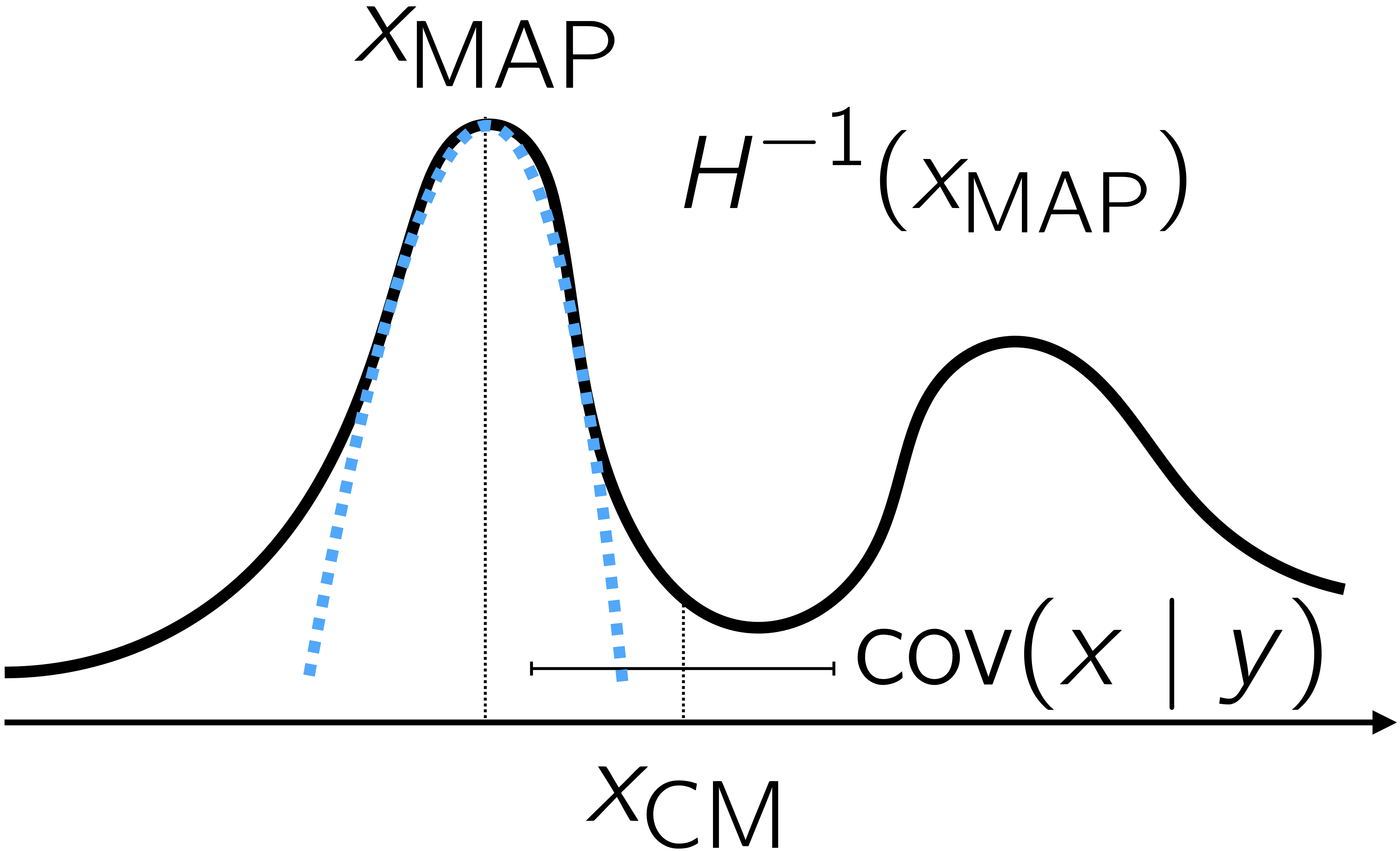
$$\pi_{\text{posterior}}(x \mid y) \propto \pi_{\text{likelihood}}(y \mid x)\pi_{\text{prior}}(x)$$

$$x_{\text{MAP}} = \underset{x \in \mathbb{R}^n}{\arg\max}\, \pi_{\text{posterior}}(x \mid y)$$



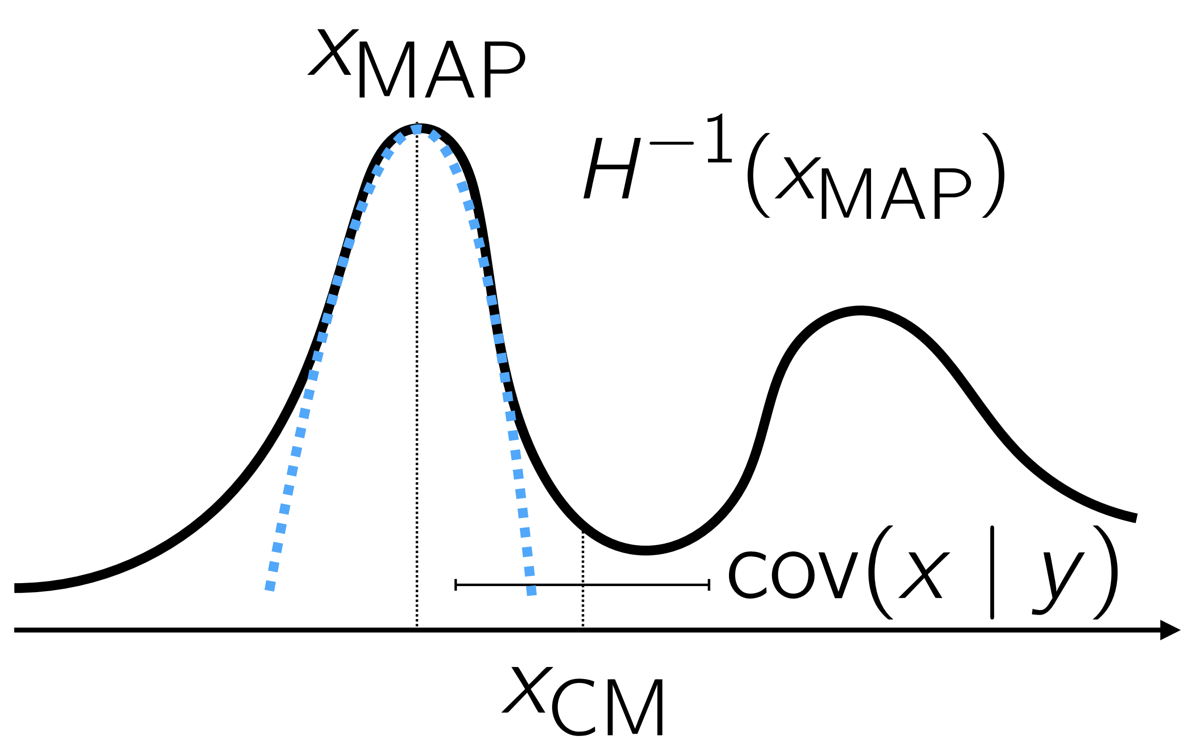
Strategy




x_{MAP}
$H^{-1}(x_{\mathrm{MAP}})$
$\mathrm{cov}(x \mid y)$
x_{CM}

$$\pi_{\text{posterior}} \sim \mathcal{N}(x_{\text{MAP}}, \mathbf{H}^{-1})$$

$$\pi^{\text{approx}}_{\text{posterior}} \sim \mathcal{N}(x_{\text{MAP}}, \mathbf{H}^{-1}(x_{\text{MAP}}))$$



Back to the problem…

Find x_{MAP} that satisfies:

$$\min_x \frac{1}{2}\|y - f(x)\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{noise}}} + \frac{1}{2}\|x - x_0\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{prior}}},$$

where the parameter-to-observable map $f : \mathbb{R}^n \to \mathbb{R}^m$ is is defined such that:

$$F(f(x), x) = D_v \int_\Omega \Pi(f(x), x)\; dx = 0 \quad \forall v \in H^1_D(\Omega),\; x \in L^2(\Omega),$$

where

$$\begin{aligned}
\Pi(u, x) &= \int_\Omega \psi(u, x)\; dx - \int_\Gamma t \cdot u\; ds, \\
\psi(u, x) &= \frac{x}{2}(I_C - d) - x\ln(J) + \frac{\lambda}{2}\ln(J)^2, \\
\mathbf{F} &= \frac{\partial \phi}{\partial X} = \mathbf{I} + \nabla u, \\
\mathbf{C} &= \mathbf{F}^T\mathbf{F}, \\
\mathbf{I_C} &= \text{tr}(C), \\
\mathbf{J} &= \det\mathbf{F}.
\end{aligned}$$

```python
from dolfin import *
mesh = UnitSquareMesh(32, 32)

U = VectorFunctionSpace(mesh, "CG", 1)
V = FunctionSpace(mesh, "CG", 1)
# solution
u = Function(U)
# test functions
v = TestFunction(U)
# incremental solution
du = TrialFunction(U)
mu = interpolate(Constant(1.0), V)
lmbda = interpolate(Constant(100.0), V)

dims = mesh.type().dim()
I = Identity(dims)
F = I + grad(u)
C = F.T*F
J = det(F)
Ic = tr(C)
```

```python
phi = (mu/2.0)*(Ic - dims) - mu*ln(J) + (lmbda/
2.0)*(ln(J))**2
Pi = phi*dx
# gateux derivative with respect to u in direction v
F = derivative(Pi, u, v)
# and with respect to u in direction du
J = derivative(F, u, du)

u_h = Function(U)
F_h = replace(F, {u: u_h})
J_h = replace(J, {u: u_h})
solve(F_h == 0, u_h, bcs, J=J_h)
```

```
u_obs << File("observations.xdmf")
J = Functional(inner(u - u_obs, u - u_obs)*dx + \
               inner(mu, mu))
m = Control(mu)
J_hat = ReducedFunctional(J, m)
...
dJdm = Jhat.derivative()[0]
H = Jhat.hessian(dm)[0]
```

Solving

Strategy: Use hooks in dolfin-adjoint to solve with petsc4py-based contexts.

- Parameter-to-observable map. Newton-Krylov method.
    - Inner solve with GAMG preconditioned GMRES (KSP) with near null-space set.
    - Newton with second-order backtracking line search (SNES).

- MAP estimator.
  - Bound constrained Quasi-Newton BLMVM with More-Thuente line search (TAO).
  - Reisz-Map aware solver - mesh independent convergence.

- Principal Component Analysis.
  - Leading: Krylov Schur (SLEPc).

Big

Dense

$$\mathbf{H} \in \mathbb{R}^{n \times n}$$

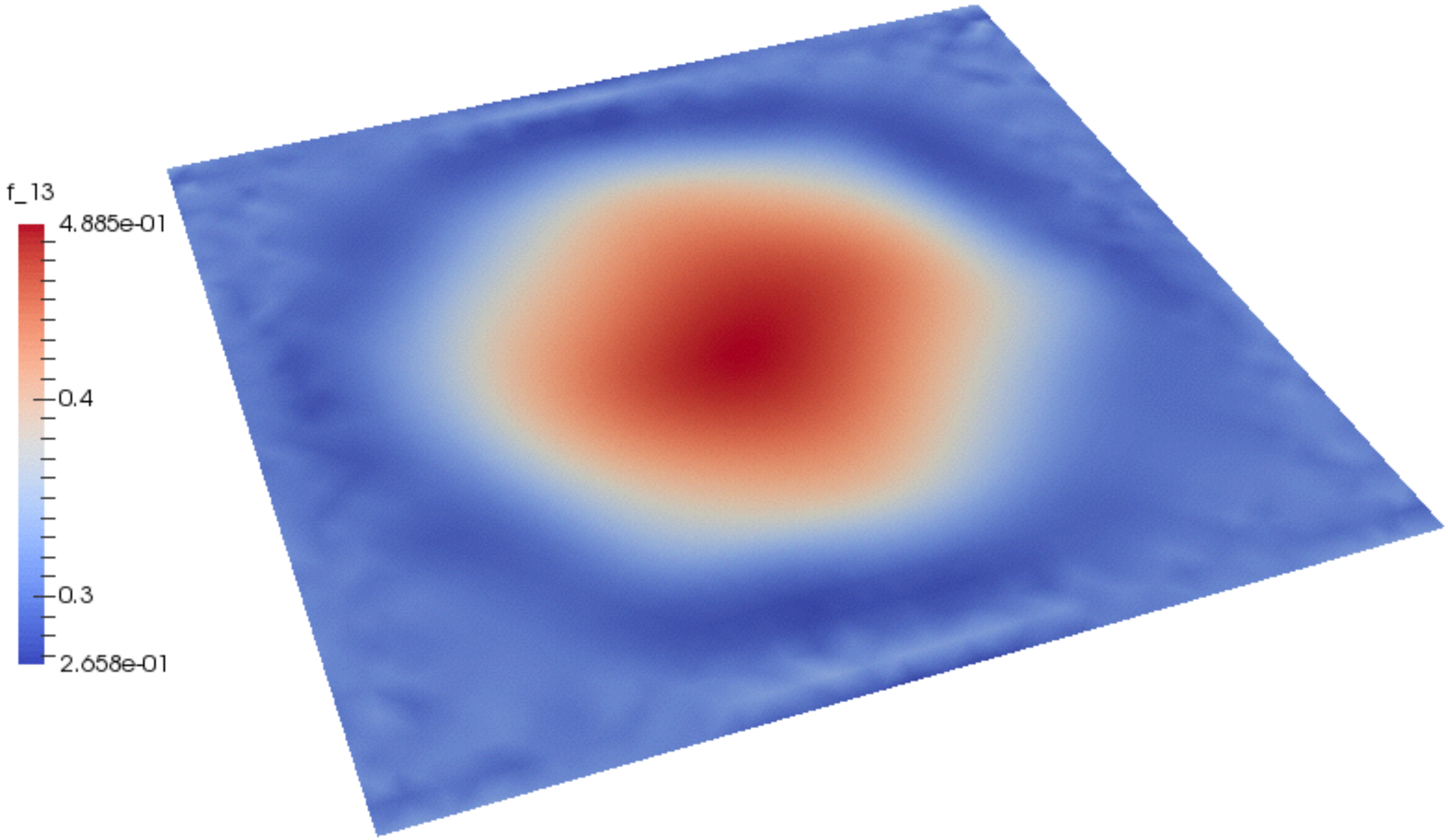
Expensive to calculate

Only have action

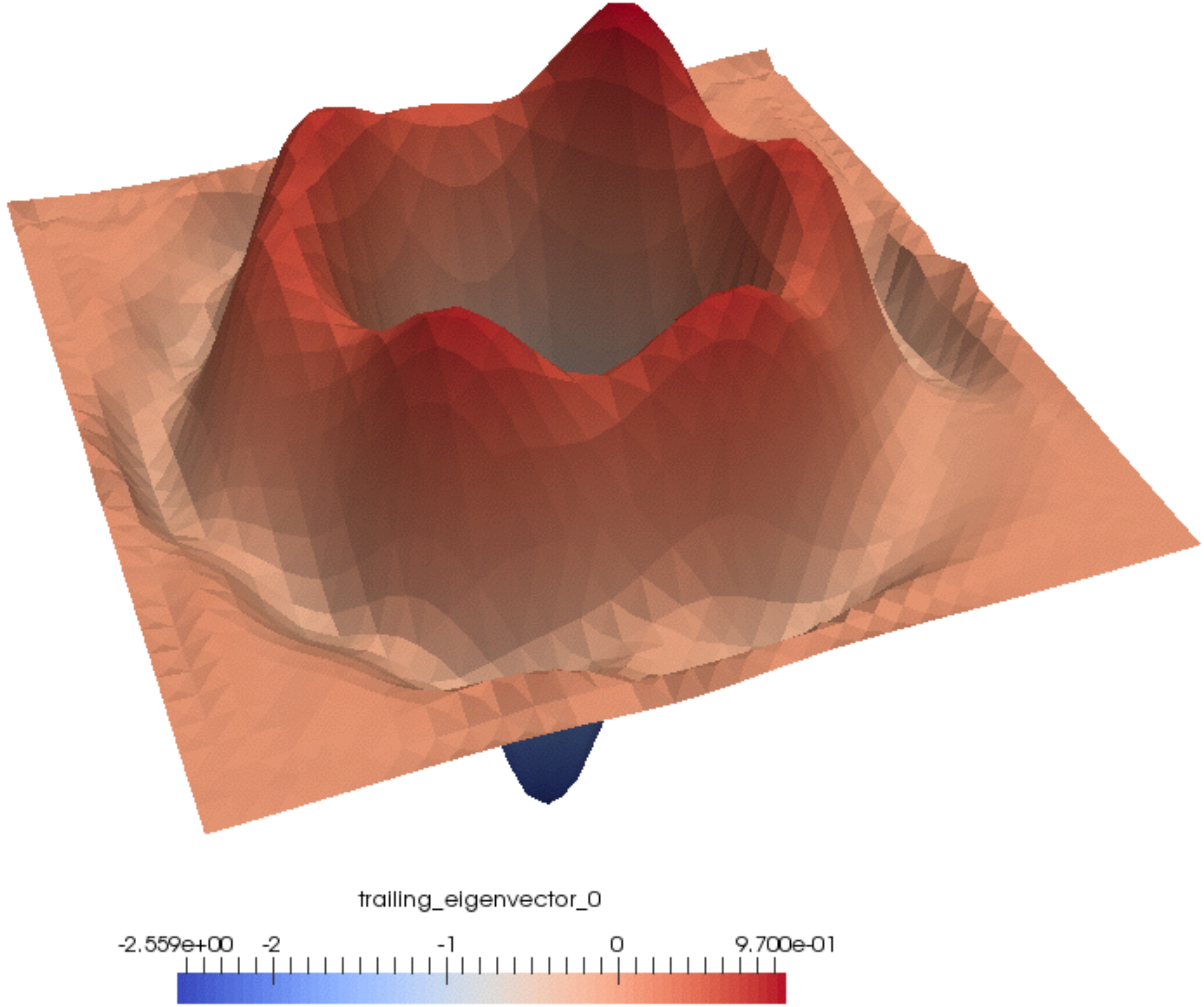
$$\mathbf{H} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^T$$

$$\boldsymbol{\Gamma}_{\text{post}} = \mathbf{H}^{-1} = \mathbf{Q}^T\boldsymbol{\Lambda}^{-1}\mathbf{Q}$$

In addition: Optimal low rank updates from Spantini et al. [arXiv 2016]



x_{map}

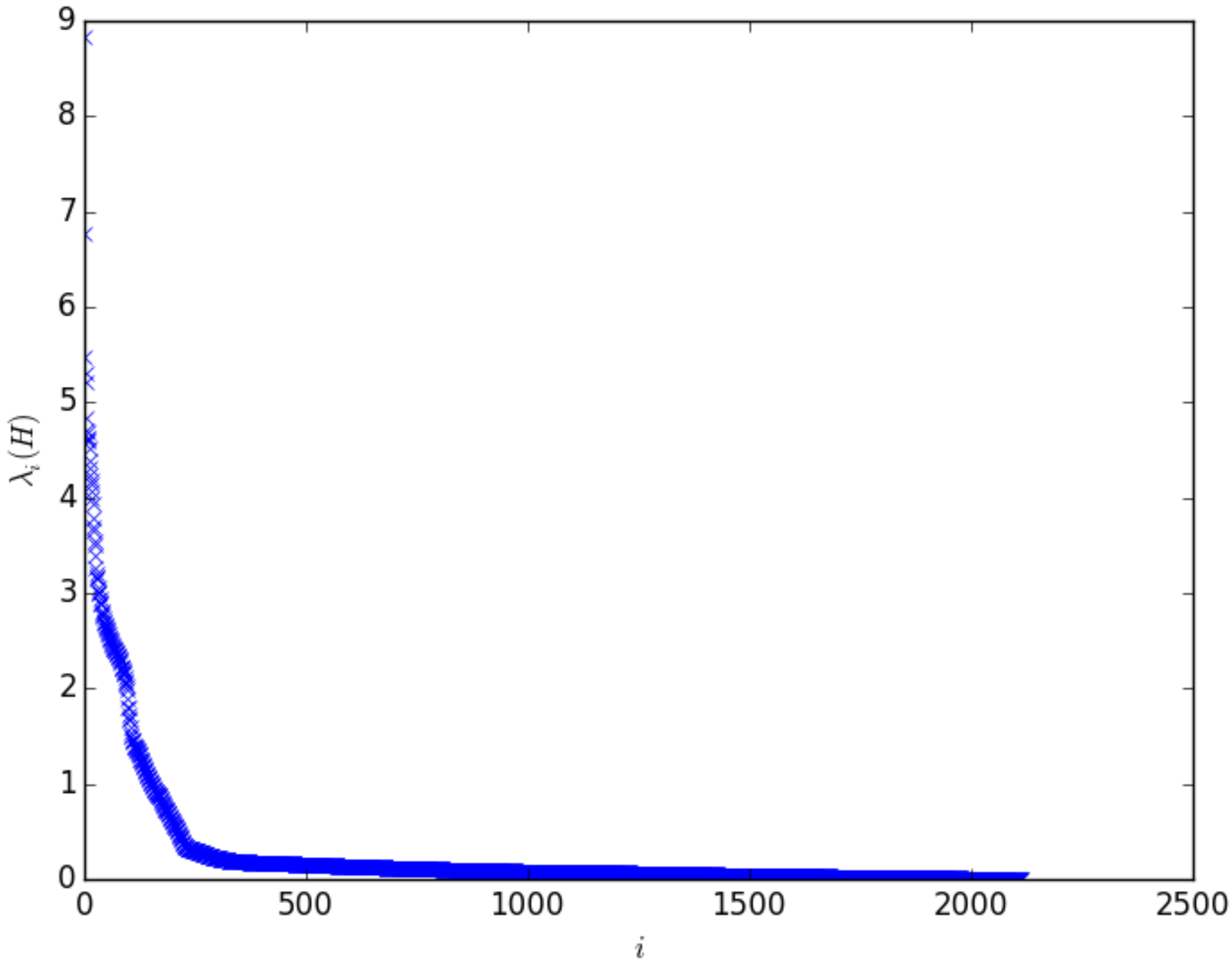

trailing_eigenvector_0
-2.559e+00
-2
-1
0
9.700e-01

Leading Eigenvectors

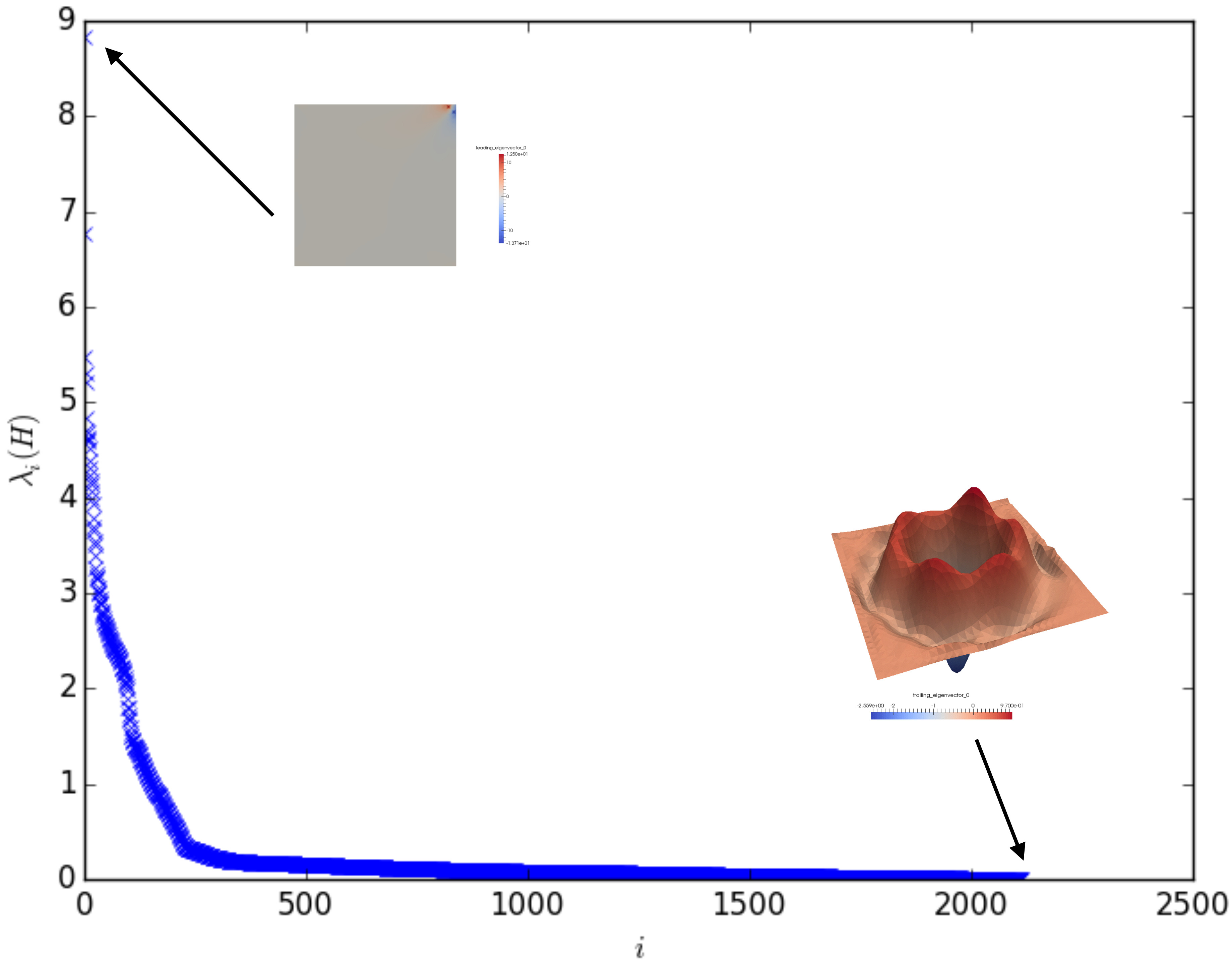
Direction in parameter space *most* constrained by the observations

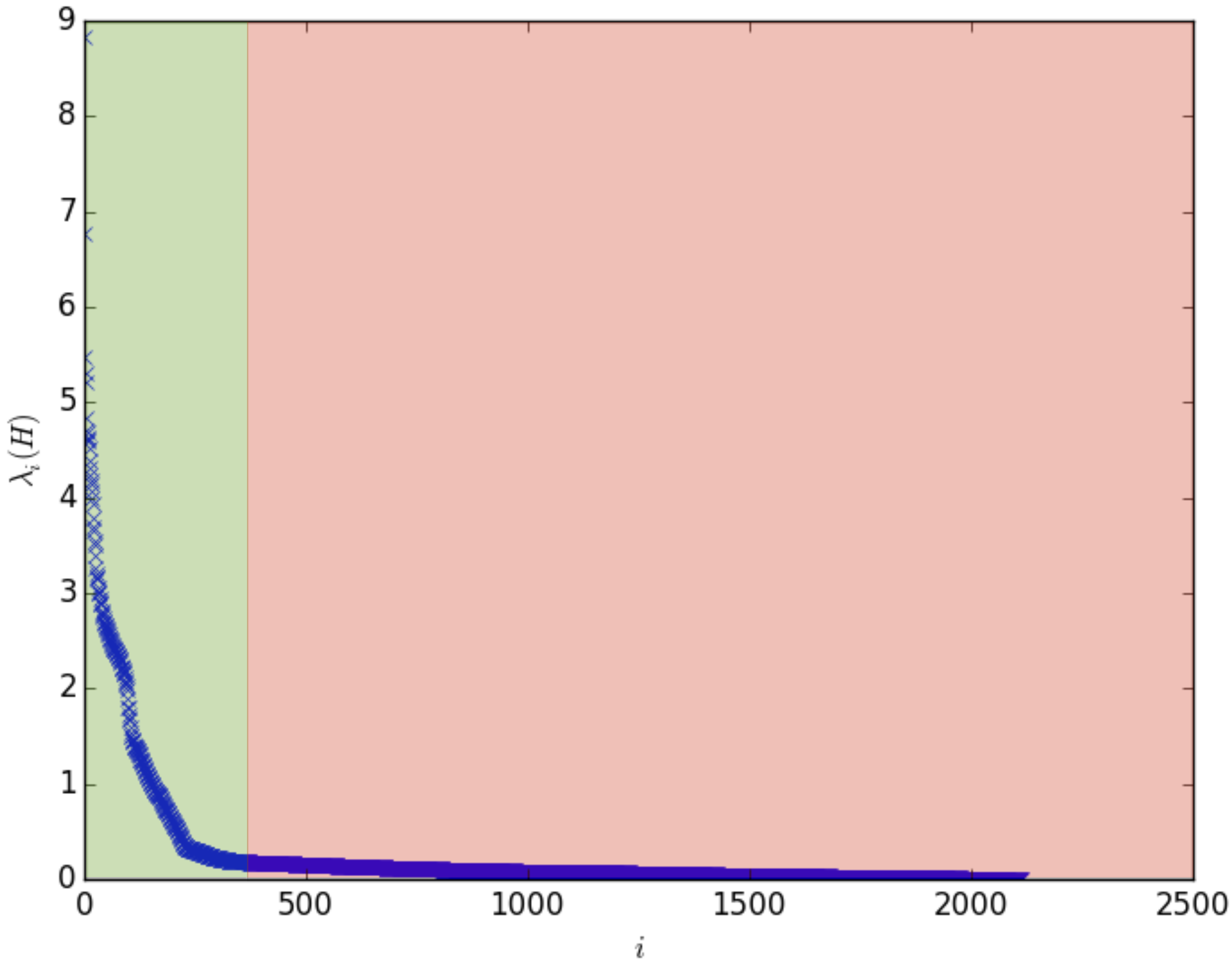

leading_eigenvector_0
1.250e+01
10
0
-10
-1.371e+01


leading_eigenvector_1
1.405e+01
10
0
-10
-1.377e+01

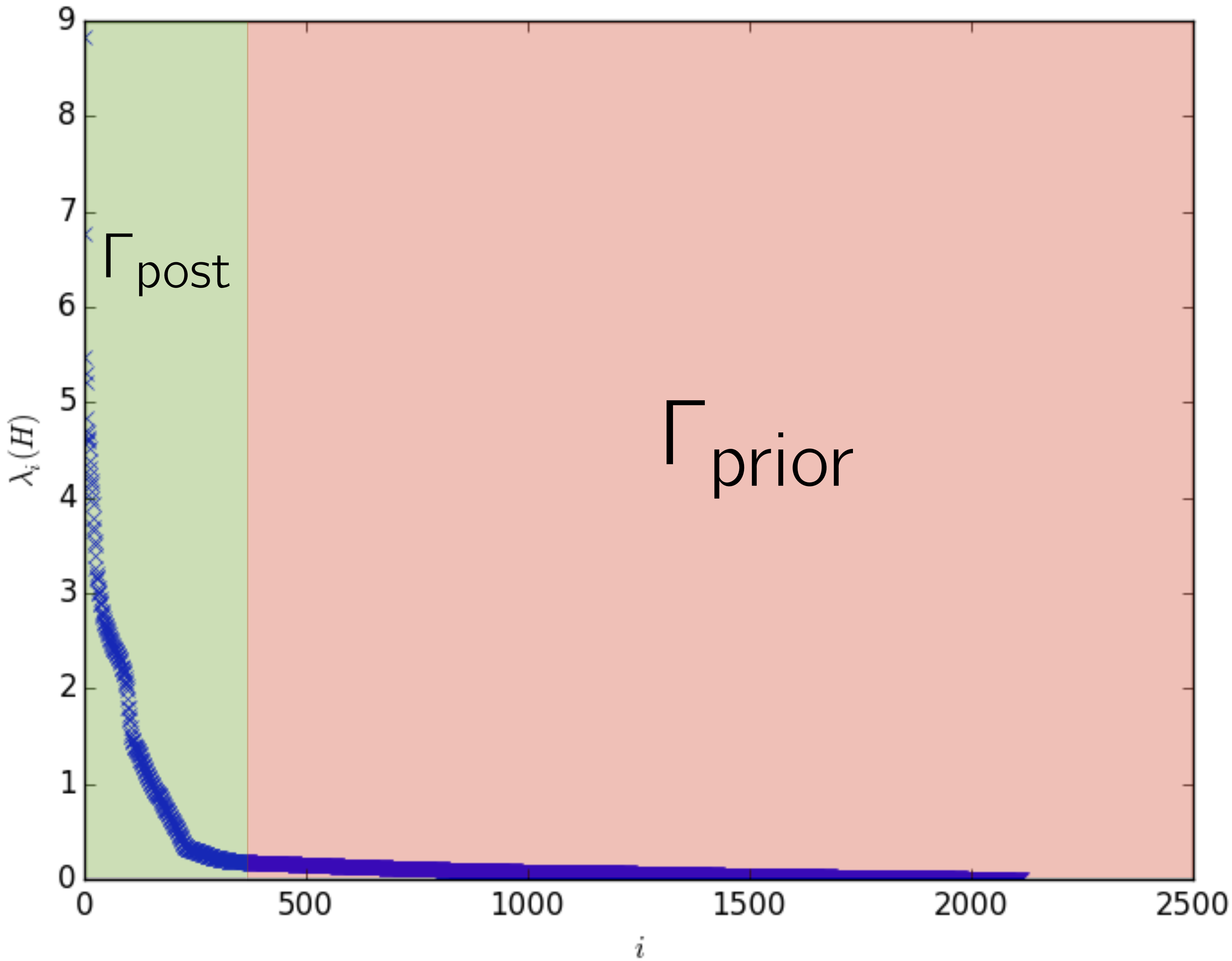


9
8
7
6
5
4
3
2
1
0
$\lambda_i(H)$
0
500
1000
1500
2000
2500
i


9
8
7
6
5
4
3
2
1
0
$\lambda_i(H)$
0
500
1000
1500
2000
2500
i



9
8
7
6
5
4
3
2
1
0
$\lambda_i(H)$
0
500
1000
1500
2000
2500
i



Matches trends from Flath et al. p424 for linear parameter to observable maps.

Full Hessian.
4000+ actions.
2000 to calculate H. 2000 to extract.

Partial Hessian.
501 actions for 292 for leading.
209 to calculate H. 292 to extract.

Huge savings in computational cost.
Scales with model dimension.

Summary

- We are developing methods to access uncertainty in the recovered parameters in hyperelastic materials.
- This is done within the framework of Bayesian inference.
- dolfin-adjoint makes assembling the equations relatively easy, solving them is tougher.
- Currently exploring efficient Monte Carlo methods to quickly calculate sample statistics of posterior.

Fonds National de la Recherche Luxembourg

COFUNDED BY MARIE CURIE ACTIONS

erc
European Research Council
Established by the European Commission

StG. No. 279578 RealTCut

EPSRC
Engineering and Physical Sciences Research Council