# hIPPYlib: An Extensible Software Framework for PDE-constrained Deterministic and Linearized Bayesian Inverse Problems

Noémi Petra[1]

joint work with:
Omar Ghattas[2], Umberto Villa[2]

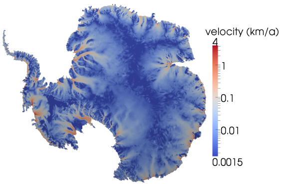
[1]School of Natural Sciences, University of California, Merced
[2]ICES, The University of Texas at Austin

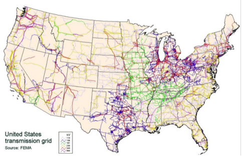
samsi
Optimization Program Summer School
August 8-12, 2016

# Collaborators and acknowledgments

- Alen Alexanderian (NC State)
- Toby Isaac (U Chicago)
- James Martin (ICES, UT Austin)
- Georg Stadler (Courant, NYU)
- Hongyu Zhu (ICES, UT Austin)

- Mihai Anitescu (ANL)
- Emil Constantinescu (ANL)
- Cosmin Petra (LLNL)




- NSF (pending) support: SSI Collaborative Research: Integrating Data with Complex Predictive Models under Uncertainty: An Extensible Software Framework for Large-Scale Bayesian Inversion
  - Joint with: Omar Ghattas (UT Austin), Umberto Villa (UT Austin), Youssef Marzouk (MIT) and Matthew Parno (CRREL)

Thank you!

# Outline

1. Intro and roadmap
   - What you will hopefully learn today . . .
   - Inverse problems: overview and examples
   - Motivation and challenges
   - Inference of basal sliding parameter field in ice sheets

2. Deterministic Inversion: formulation, solution methods and examples
   - Newton-type descent algorithms with line search
   - Calculus of variations, weak forms and computing derivatives via adjoints
   - Example: Coefficient field inversion in an elliptic PDE

3. Bayesian approach to inverse problems
   - Why is a statistical perspective useful in inverse problems?
   - Bayesian inversion
   - Challenges for large-scale Bayesian inverse problems
   - Example: Inversion for the initial condition in an advection-diffusion PDE

4. hIPPYlib: Inverse Problem PYthon library

5. Conclusions, discussion and references

# A little bit about my background

- B.S and M.S., Mathematics and Computer Science, University of Babeș-Bolyai, Cluj-Napoca, Romania
  (*founded in 1872 as "The Franz Joseph Hungarian Royal University of Kolozsvár". World famous mathematicians at UBB: Frigyes Riesz, Lipót Fejér, Alfréd Haar, Gyula Farkas*)


# A little bit about my background

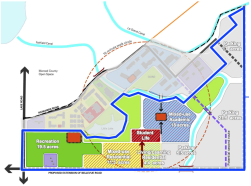
- B.S and M.S., Mathematics and Computer Science, University of Babeș-Bolyai, Cluj-Napoca, Romania
  (*founded in 1872 as "The Franz Joseph Hungarian Royal University of Kolozsvár". World famous mathematicians at UBB: Frigyes Riesz, Lipót Fejér, Alfréd Haar, Gyula Farkas*)

# A little bit about my background

- M.S. and Ph.D. Applied Mathematics, University of Maryland, Baltimore County, USA
- Postdoc and research associate, Center for Computational Geosciences and Optimization (CCGO), Institute for Computational Engineering and Sciences (ICES), The University of Texas at Austin
- Visiting Faculty (Summers 2015 and 2016), Argonne National Laboratory (funded by DOE)
- Assistant Professor of Applied Mathematics, University of California, Merced

# A little bit about my background

- M.S. and Ph.D. Applied Mathematics, University of Maryland, Baltimore County, USA
- Postdoc and research associate, Center for Computational Geosciences and Optimization (CCGO), Institute for Computational Engineering and Sciences (ICES), The University of Texas at Austin
- Visiting Faculty (Summers 2015 and 2016), Argonne National Laboratory (funded by DOE)
- Assistant Professor of Applied Mathematics, University of California, Merced

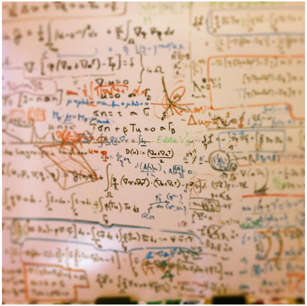



# A little bit about my background

- M.S. and Ph.D. Applied Mathematics, University of Maryland, Baltimore County, USA
- Postdoc and research associate, Center for Computational Geosciences and Optimization (CCGO), Institute for Computational Engineering and Sciences (ICES), The University of Texas at Austin
- Visiting Faculty (Summers 2015 and 2016), Argonne National Laboratory (funded by DOE)
- Assistant Professor of Applied Mathematics, University of California, Merced

# What you will hopefully learn today . . .

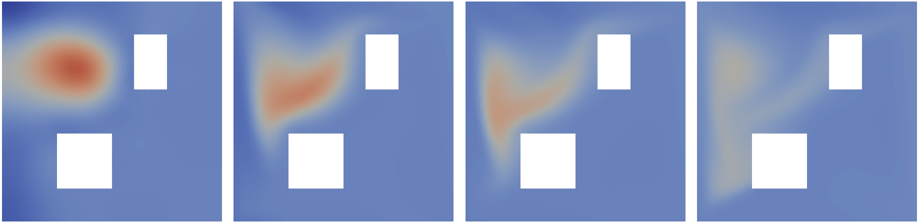
- (Most likely review) what inverse problems (governed by PDEs) are and why they are important
- The regularization versus statistical approach to inverse problems
- You will see several model problems but also examples of challenging, large-scale real-world inverse problems
- How to approach these problems, i.e. how to solve these problems efficiently; will learn about hIPPYlib



# References

1. **Theory and computational methods for inverse problems:**
   - Heinz Engl, Michael Hanke, and Andreas Neubauer, *Regularization of Inverse Problems*, Dordrecht, 2nd ed., 1996.
   - Curtis R. Vogel, *Computational Methods for Inverse Problems*, SIAM, 2002.
   - Guy Chavent, *Nonlinear Least Squares for Inverse Problems*, Springer, 2009.
2. **Numerical optimization background:**
   - Jorge Nocedal and Stephen J. Wright, *Numerical Optimization*, Springer-Verlag, 1999.
   - C. Tim Kelley, *Iterative Methods of Optimization*, SIAM, 1999.
3. **Optimization of systems governed by PDEs:**
   - A. Borzi and V. Schulz, *Computational Optimization of Systems Governed by Partial Differential Equations*, SIAM, 2012
   - Fredi Tröltzsch, *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, Graduate Studies in Mathematics Vol. 112, AMS, 2010.
   - M. Hinze, R. Pinnau, M. Ulbich, and S. Ulbrich, *Optimization with PDE constraints*, Springer, 09.
4. **Finite element background and FEniCS:**
   - Eric B. Becker, Graham F. Carey, and J. Tinsley Oden, *Finite Elements: Volume I, An Introduction*, Prentice Hall, 1981.
   - Mark S. Gockenbach, *Understanding and Implementing the Finite Element Method*, SIAM, 2006.
   - Howard Elman, David Silvester, and Andrew Wathen, *Finite Elements and Fast Iterative Solvers*, Oxford University Press, 2005.
   - A. Logg, K. A. Mardal, and G. Wells, *Automated solution of differential equations by the finite element method: The FEniCS book*, vol. 84, Springer Science & Business Media, 2012. *You can download the electronic copy from here.*
5. **Probabilistic approach to inverse problems:**
   - Albert Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*, SIAM, 2005.
   - Jari Kaipio and Erkki Somersalo, *Statistical and Computational Inverse Problems*, Springer, 2005.
   - Andrew Stuart, Inverse problems, *A Bayesian approach*, Acta Numerica, 2010.

- Implements state-of-the-art scalable adjoint-based algorithms for PDE-based deterministic and Bayesian inverse problems.
- Builds on FEniCS for the discretization of the PDE and on PETSc for scalable and efficient linear algebra operations and solvers.
- Release: hIPPYlib 1.0 is now available at http://hippylib.github.io



# A little bit of history of `hIPPYlib`

- `VIP-PDE` (Sandbox for **V**ariational **I**nverse **P**roblems governed by **PDE**s), a computational sandbox that contains and allows easy extensions of state-of-the-art scalable algorithms for PDE-based deterministic inverse problems in COMSOL with matlab (open source, co-developed with Georg Stadler, NYU)

  Details in: N. Petra and G. Stadler, *Model variational inverse problems governed by partial differential equations*. ICES Technical Report, The University of Texas at Austin, 2011

- `hIPPYlib` (**I**nverse **P**roblem **PY**thon **lib**rary), a computational sandbox that contains and allows easy extensions of state-of-the-art scalable algorithms for PDE-based deterministic and Bayesian inverse problems in python (open source, co-developed with Umberto Villa, ICES UT Austin)

  Details in: U. Villa, N. Petra, and O. Ghattas, *hIPPYlib: An Extensible Software Framework for Large-Scale Deterministic and Linearized Bayesian Inverse Problems*. To be submitted.

- Both VIPPDE and `hIPPYlib` have been used in numerous classes at UT Austin, UC Merced and NYU. These *prototyping* environments have led to three SISC, two glaciology, and one SIAM UQ papers.

# Regularization approach to inverse problems

Parameter-to-observable map: often unique solution, well-posed

$$\boxed{\boldsymbol{m}} \xrightarrow{\text{forward PDE}} \boxed{\boldsymbol{f}(\boldsymbol{m})} \xrightarrow{\text{Observations}} \boxed{\boldsymbol{d} = f(\boldsymbol{m}) + \boldsymbol{\varepsilon}}$$

Inverse problem: ill-posed, under-determined

Given observations $\boldsymbol{d}$, find the parameters $\boldsymbol{m}$ such that $\boldsymbol{d} \approx \boldsymbol{f}(\boldsymbol{m})$ through solution of the optimization problem

$$\min_{\boldsymbol{m}} \mathcal{J}(\boldsymbol{m}) := \frac{1}{2} \parallel \boldsymbol{f}(\boldsymbol{m}) - \boldsymbol{d} \parallel_{\mathbf{W}}^2 + \frac{\alpha}{2} \parallel \boldsymbol{m} - \boldsymbol{m}_0 \parallel_{\boldsymbol{R}}^2$$

where the forward problem maps the parameter $\boldsymbol{m}$ into the observable $\boldsymbol{f}(\boldsymbol{m})$, and $\alpha$ is a properly chosen regularization parameter.

The solution of this optimization problem depends on weights and regularization.

# Example I: Image deblurring & denoising

forward problem does not involve a PDE

Original image

$$\xrightarrow{F}$$

Blurred image

# Example I: Image deblurring & denoising

forward problem does not involve a PDE

Original image

$\xrightarrow{F}$

Blurred image

- $F$ ... blurring operator (no PDE in forward problem)
- parameters/image: left image (not known in inverse problem)
- data/measurements: right (blurred) image

# Example I: Image deblurring & denoising

forward problem does not involve a PDE

Original image

$\xrightarrow{F}$

Blurred image

- $F$ . . . blurring operator (no PDE in forward problem)
- parameters/image: left image (not known in inverse problem)
- data/measurements: right (blurred) image

$\longleftarrow$ (inverse problem) (?) $\longleftarrow$

# Example II: Parameter identification in elliptic PDE

## Model for single phase flow, ground water filtration

$$-\nabla \cdot (a\nabla u) = f \quad \text{in } \Omega \subset \mathbb{R}^d \quad \text{with bdry conditions}$$

$f \ldots$ given source function

$a \ldots$ permeability, hydraulic permittivity

$u \ldots$ unknown pressure

# Example II: Parameter identification in elliptic PDE

## Model for single phase flow, ground water filtration

$$-\nabla \cdot (a\nabla u) = f \quad \text{in } \Omega \subset \mathbb{R}^d \quad \text{with bdry conditions}$$

$f\ldots$ given source function

$a\ldots$ permeability, hydraulic permittivity

$u\ldots$ unknown pressure

- forward problem $F$ is the parameter-to-solution mapping $a \to u_a$ (i.e., solve the PDE)
- data/measurements: pressure $u$ at points/parts of $\Omega$
- parameter field/image: $a = a(x)$
- inverse problem: given (measurements of) $u$, find $a$

# Example III: Inverse scattering

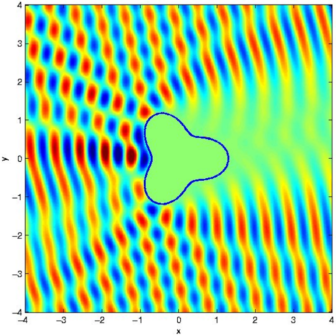
Use (acoustic/elastic/electromagnetic) waves scattered by object to learn about its shape

For instance, the acoustic wave equation is given by:

$$u_{tt} - \frac{1}{s(x)^2}\Delta u = 0 \quad \text{in } \Omega \subset \mathbb{R}^d \quad \text{with bdry. \& initial cond.}$$

$s(x)$... spatially varying wave speed, $s(x) = 1$ outside object and $s(x) \neq 1$ inside the (unknown) object

$u = u(x,t)$... wave field



# Example III: Inverse scattering

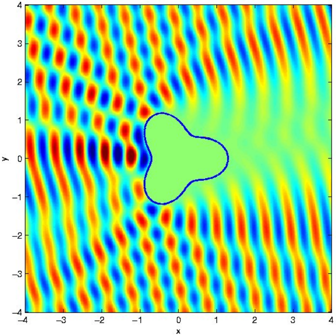
Use (acoustic/elastic/electromagnetic) waves scattered by object to learn about its shape

For instance, the acoustic wave equation is given by:

$$u_{tt} - \frac{1}{s(x)^2}\Delta u = 0 \quad \text{in } \Omega \subset \mathbb{R}^d \quad \text{with bdry. \& initial cond.}$$

$s(x)\ldots$spatially varying wave speed, $s(x) = 1$ outside object and $s(x) \neq 1$ inside the (unknown) object
$u = u(x,t)\ldots$wave field

- parameters/image: region where $s(x) \neq 1$
- forward operator $F$: given wave speed $s(x)$, compute the wave field $u$, i.e., solve the PDE
- measurements of wave field $u$ at points for time instances/intervals



# Example III: Inverse scattering

Use (acoustic/elastic/electromagnetic) waves scattered by object to learn about its shape

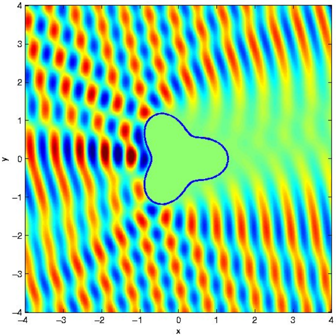
For instance, the acoustic wave equation is given by:

$$u_{tt} - \frac{1}{s(x)^2}\Delta u = 0 \quad \text{in } \Omega \subset \mathbb{R}^d \quad \text{with bdry. \& initial cond.}$$

$s(x)\ldots$ spatially varying wave speed, $s(x) = 1$ outside object and $s(x) \neq 1$ inside the (unknown) object

$u = u(x,t)\ldots$ wave field

- parameters/image: region where $s(x) \neq 1$
- forward operator $F$: given wave speed $s(x)$, compute the wave field $u$, i.e., solve the PDE
- measurements of wave field $u$ at points for time instances/intervals




Applications: radar, submarine detection, ocean floor, TSA body scans...

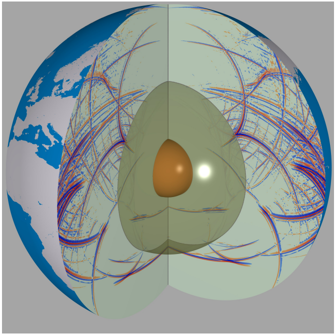
# Example IV: Wave-based material inversion

Propagate acoustic/elastic/electromagnetic waves through unknown medium

Similar as scattering problem

$$u_{tt} - \frac{1}{s(x)^2}\Delta u = 0 \quad \text{in } \Omega \subset \mathbb{R}^d \quad \text{with bdry. \& initial cond.}$$

$s(x)$... spatially varying wave speed
$u = u(x,t)$... wave field



# Example IV: Wave-based material inversion

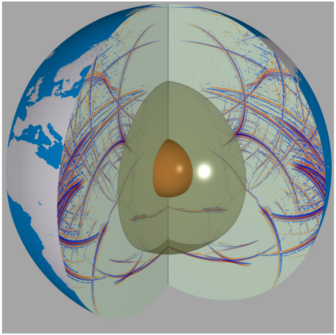
Propagate acoustic/elastic/electromagnetic waves through unknown medium

Similar as scattering problem

$$u_{tt} - \frac{1}{s(x)^2}\Delta u = 0 \quad \text{in } \Omega \subset \mathbb{R}^d \quad \text{with bdry. \& initial cond.}$$

$s(x)\ldots$ spatially varying wave speed
$u = u(x,t)\ldots$ wave field

- parameter/image: wave speed $s(x)$
- forward operator $F$: given the wave speed $s(x)$, compute the wave field $u$, i.e., solve the PDE
- data/measurements: wave field $u$ at points in space and time



# Example IV: Wave-based material inversion

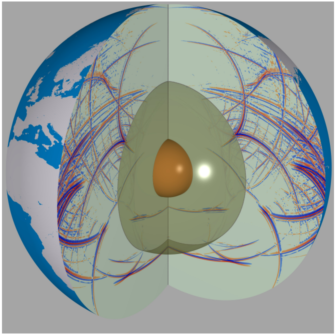
Propagate acoustic/elastic/electromagnetic waves through unknown medium

Similar as scattering problem

$$u_{tt} - \frac{1}{s(x)^2}\Delta u = 0 \quad \text{in } \Omega \subset \mathbb{R}^d \quad \text{with bdry. \& initial cond.}$$

$s(x)$... spatially varying wave speed
$u = u(x,t)$... wave field

- parameter/image: wave speed $s(x)$
- forward operator $F$: given the wave speed $s(x)$, compute the wave field $u$, i.e., solve the PDE
- data/measurements: wave field $u$ at points in space and time

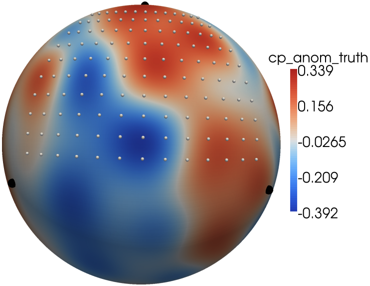


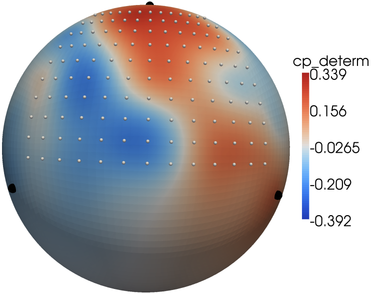
Applications: study of earth interior, oil exploration,...

# Example V: Wave-based material inversion

Propagate acoustic/elastic/electromagnetic waves through unknown medium

Recent results in global seismic inversion: "Truth" wave speed anomaly on the left and solution of inversion on the right. Black dots correspond to earthquake sources, white dots are receiver measurement points.



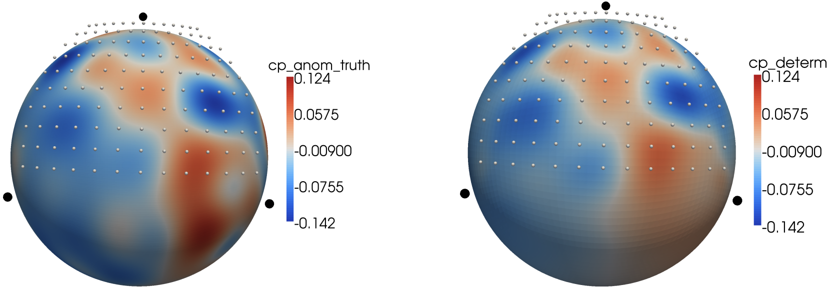

Wave speed anomaly in a depth of 70km.

Details in:. Bui-Thanh, T.; Ghattas, O.; Martin, J. & Stadler, G. A Computational Framework for Infinite-Dimensional Bayesian Inverse Problems: Part I. The Linearized Case, with Application to Global Seismic Inversion, SIAM Journal on Scientific Computing, 2013, 35, A2494-A2523

# Example V: Wave-based material inversion

Propagate acoustic/elastic/electromagnetic waves through unknown medium

Recent results in global seismic inversion: "Truth" wave speed anomaly on the left and solution of inversion on the right. Black dots correspond to earthquake sources, white dots are receiver measurement points.



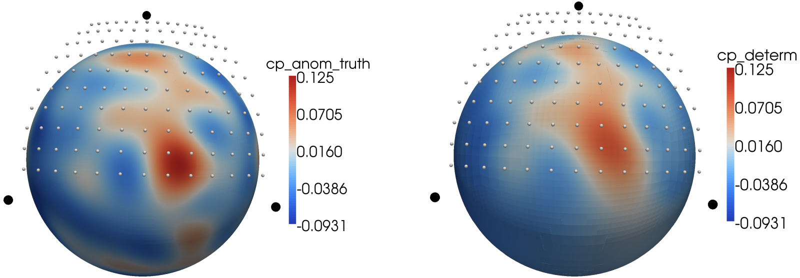

Wave speed anomaly in a depth of 700km.

Details in:. Bui-Thanh, T.; Ghattas, O.; Martin, J. & Stadler, G. A Computational Framework for Infinite-Dimensional Bayesian Inverse Problems: Part I. The Linearized Case, with Application to Global Seismic Inversion, SIAM Journal on Scientific Computing, 2013, 35, A2494-A2523

# Example V: Wave-based material inversion

Propagate acoustic/elastic/electromagnetic waves through unknown medium

Recent results in global seismic inversion: “Truth” wave speed anomaly on the left and solution of inversion on the right. Black dots correspond to earthquake sources, white dots are receiver measurement points.




Wave speed anomaly in a depth of 1400km.

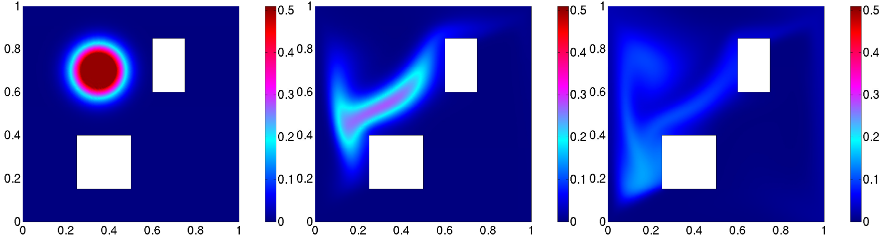
Details in:. Bui-Thanh, T.; Ghattas, O.; Martin, J. & Stadler, G. A Computational Framework for Infinite-Dimensional Bayesian Inverse Problems: Part I. The Linearized Case, with Application to Global Seismic Inversion, SIAM Journal on Scientific Computing, 2013, 35, A2494-A2523

# Example VI: Inversion for an initial condition

## Advection-diffusion equation

Transport of a concentration filed $u$ by diffusion and an advective field $\mathbf{v}$. Use measurements at boundaries of white squares to infer the initial concentration $u_0$.

$$\begin{aligned} u_t - \kappa\Delta u + \mathbf{v}\cdot\nabla u &= 0 && \text{in } \Omega\times[0,T], \\ \kappa\nabla u\cdot\mathbf{n} &= 0 && \text{on } \partial\Omega\times[0,T], \\ u(0,x) &= u_0 && \text{in } \Omega. \end{aligned}$$

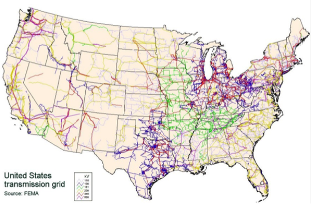



Applications: Detecting contamination source in transport in ground water, ocean or air.

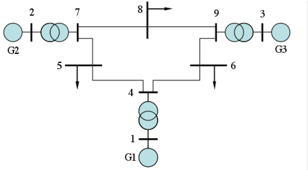
# Example VII: Parameter Estimation for Power Grid

## Differential-algebraic (DAE) driven inverse problems

$$\dot{x} = h(x, y, m, t)$$
$$0 = g(x, y, t)$$
$$x(0) = x_0$$




- Estimate/infer model parameters $m$ (e.g., inertia of the generators)
- Observational data (e.g., bus voltages)




Goal: Enable real-time power grid operational tasks such as monitoring fault-detection, dynamic stability assessment, etc.

Details in: Petra, N.; Petra, C.; Zhang, Z.; Constantinescu, E. and Anitescu, M. A Bayesian Approach for Parameter Estimation with Uncertainty for Dynamic Power Systems. In review.

# Example VIII: Dynamics of the Antarctic ice sheet

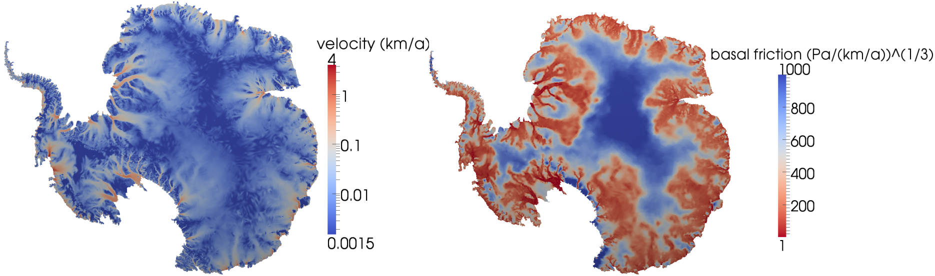
Glaciers flow thousands of miles from the continent's deep interior to its coast

Credit: NASA Goddard Space Flight Center/JPL-Caltech

# Example VIII: Inference of basal friction in Antarctica

Creeping, viscous, incompressible, non-Newtonian flow:

$$
\begin{aligned}
-\boldsymbol{\nabla} \cdot \left[2\eta(\boldsymbol{u})\left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T\right) - \boldsymbol{I}p\right] = \rho \boldsymbol{g} \quad & \text{in } \Omega \\
\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0 \quad & \text{in } \Omega \\
\boldsymbol{\sigma_u n} = \boldsymbol{0} \quad & \text{on } \Gamma_t \\
\boldsymbol{u} \cdot \boldsymbol{n} = 0,\ \boldsymbol{T \sigma_u n} + \exp(\beta) \boldsymbol{T u} = \boldsymbol{0} \quad & \text{on } \Gamma_b
\end{aligned}
$$

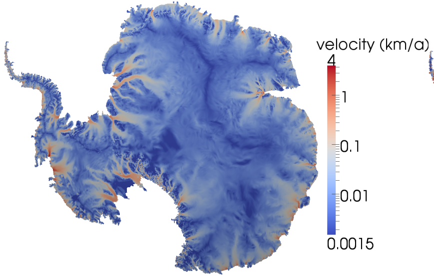


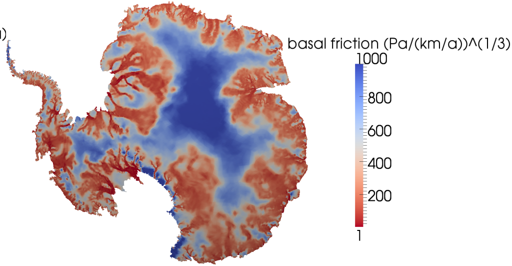

Left: InSAR-based Antarctica ice surface velocity observations
Right: Inferred basal friction field

# Example VIII: Inference of basal friction in Antarctica

Creeping, viscous, incompressible, non-Newtonian flow:

$$
\begin{aligned}
-\boldsymbol{\nabla} \cdot \left[2\eta(\boldsymbol{u})\left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T\right) - \boldsymbol{I}p\right] = \rho \boldsymbol{g} \quad & \text{in } \Omega \\
\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0 \quad & \text{in } \Omega \\
\boldsymbol{\sigma_u n} = \boldsymbol{0} \quad & \text{on } \Gamma_t \\
\boldsymbol{u} \cdot \boldsymbol{n} = 0,\ \boldsymbol{T\sigma_u n} + \exp(\beta)\boldsymbol{Tu} = \boldsymbol{0} \quad & \text{on } \Gamma_b
\end{aligned}
$$




Left: Recovered ice surface velocity observations

Right: Inferred basal friction field

# Example VIII: Inference of basal friction in Antarctica

Creeping, viscous, incompressible, non-Newtonian flow:

- Spatial discretization:
  - velocity (state, adjoint, incremental state, incremental adjoint): $\mathbb{Q}_2$
  - pressure (state, adjoint, incremental state, incremental adjoint): $\mathbb{Q}_0^{\text{disc}}$
  - basal friction, incremental basal friction: $\mathbb{Q}_2$ (biquadratic)

Stampede, Texas Advanced Computing Center (TACC)

- # state parameters: 4,085,841
- # inversion parameters: 409,545
- # elements: 99,984
- # of cores: 1024
- inexact Newton-CG, prec. L-BFGS

- real data
- reduction in norm of gradient: $10^{-3}$
- # of Newton iterations: 213
- average # of CG iterations per Newton iteration: 239
- total # of (linearized) Stokes: 107,578

Details in: Isaac, T.; Petra, N.; Stadler, G. & Ghattas, O. 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, 2015, 296, 348-368

# Outline

1. Intro and roadmap
   - What you will hopefully learn today . . .
   - Inverse problems: overview and examples
   - Motivation and challenges
   - Inference of basal sliding parameter field in ice sheets
2. Deterministic Inversion: formulation, solution methods and examples
   - Newton-type descent algorithms with line search
   - Calculus of variations, weak forms and computing derivatives via adjoints
   - Example: Coefficient field inversion in an elliptic PDE
3. Bayesian approach to inverse problems
   - Why is a statistical perspective useful in inverse problems?
   - Bayesian inversion
   - Challenges for large-scale Bayesian inverse problems
   - Example: Inversion for the initial condition in an advection-diffusion PDE
4. hIPPYlib: **I**nverse **P**roblem **PY**thon **lib**rary
5. Conclusions, discussion and references

# Continuous unconstrained optimization

Consider $f(\cdot) \in C^2$, $f : \mathbb{R}^n \to R$, and solve

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} f(\boldsymbol{x}).$$

A point $\boldsymbol{x}^*$ is a global solution if

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}^*) \tag{1}$$

for all $\boldsymbol{x} \in \mathbb{R}^n$, and a local solution if (1) for all $\boldsymbol{x}$ in a neighborhood of $\boldsymbol{x}^*$.

Similar methods as presented here for $\boldsymbol{x} \in \mathbb{R}^n$ can be worked out in infinite dimensions, where $\boldsymbol{x}$ is replaced by a function $u = u(x)$.

Optimization slides courtesy of Georg Stadler.

# Continuous unconstrained optimization

## Necessary conditions

At a local minimum $\boldsymbol{x}^*$ holds the first-order necessary condition

$$\mathbb{R}^n \ni \nabla f(\boldsymbol{x}^*) = 0$$

and the second-order (necessary) sufficient condition

$$\mathbb{R}^{n\times n} \ni \nabla^2 f(\boldsymbol{x}^*) \quad \text{is positive (semi-) definite.}$$

To find a candidate for a minimum, we could solve the nonlinear equation for a stationary point: Find $\boldsymbol{x} \in \mathbb{R}^n$ such that

$$G(\boldsymbol{x}) := \nabla f(\boldsymbol{x}) = 0,$$

for instance with Newton's method. Note that the Jacobian of $G(\boldsymbol{x})$ is $\nabla^2 f(\boldsymbol{x})$.

# Descent algorithm

It is often preferrable to use an iterative descent algorithms, which takes into account the optimization structure.

Basic descent algorithm:

1. Initialize starting point $\boldsymbol{x}^0$, set $k = 1$.
2. For $k = 0, 1, 2, \ldots$, find a descent direction $\boldsymbol{d}^k$
3. Find a step length $\alpha_k > 0$ for the update

$$\boldsymbol{x}^{k+1} := \boldsymbol{x}^k + \alpha_k \boldsymbol{d}^k$$

such that $f(\boldsymbol{x}^{k+1}) < f(\boldsymbol{x}^k)$. Set $k := k + 1$ and repeat.

# Descent algorithm

Instead of solving an $n$-dim. minimization problem, (approximately) solve a sequence of 1-dim. problems:

- Initialization: As close as possible to $\boldsymbol{x}^*$.
- Descent direction: Direction in which function decreases locally.
- Step length: Want to make large, but not too large steps.
- Check for descent: Make sure you make progress towards a (local) minimum.

# Descent algorithm

Initialization: Ideally close to the minimizer. Solution depends, in general, on initialization (in the presence of multiple local minima).

# Descent algorithm

Directions, in which the function decreases (locally) are called descent directions.

- Steepest descent direction:

$$\boldsymbol{d}^k = -\nabla f(\boldsymbol{x}^k)$$

- When $B_k \in \mathbb{R}^{n\times n}$ is positive definite, then

$$\boldsymbol{d}^k = -B_k^{-1}\nabla f(\boldsymbol{x}^k)$$

  is the quasi-Newton descent direction.
- When $H_k = H(\boldsymbol{x}^k) = \nabla^2 f(\boldsymbol{x}^k)$ is positive definite, then

$$\boldsymbol{d}^k = -H_k^{-1}\nabla f(\boldsymbol{x}^k)$$

  is the Newton descent direction. At a local minimum, $H(\boldsymbol{x}^*)$ is positive (semi)definite.

Idea behind Newton's method in optimization: Instead of finding minimum of $f$, find minimum of quadratic approximation of $f$ around current point:

$$q_k(\boldsymbol{d}) = f(\boldsymbol{x}^k) + \nabla f(\boldsymbol{x}^k)^T \boldsymbol{d} + \frac{1}{2}\boldsymbol{d}^T \nabla^2 f(\boldsymbol{x}^k)\boldsymbol{d}$$

Minimum is (provided $\nabla^2 f(\boldsymbol{x}^k)$ is spd):

$$\boldsymbol{d} = -\nabla^2 f(\boldsymbol{x}^k)^{-1} \nabla f(\boldsymbol{x}^k).$$

is the Newton search direction. Since this is the minimum of the quadratic approximation, $\alpha_k = 1$ is the "optimal" step length.

# Descent algorithm

Step length: Need to choose step length $\alpha_k > 0$ in

$$\boldsymbol{x}^{k+1} := \boldsymbol{x}^k + \alpha_k \boldsymbol{d}^k$$

Ideally: Find minimum $\alpha$ of 1-dim. problem

$$\min_{\alpha > 0} f(\boldsymbol{x}^k + \alpha \boldsymbol{d}^k).$$

It is not necessary to find the exact minimum.

# Descent algorithm

Step length (continued): Find $\alpha_k$ that satisfies the Armijo condition:

$$f(\boldsymbol{x}^k + \alpha_k \boldsymbol{d}^k) \leq f(\boldsymbol{x}^k) + c_1 \alpha_k \nabla f(\boldsymbol{x}^k)^T \boldsymbol{d}^k, \tag{2}$$

where $c_1 \in (0, 1)$ (usually chosen rather small, e.g., $c_1 = 10^{-4}$).

Additionally, one often uses the gradient condition

$$\nabla f(\boldsymbol{x}^k + \alpha_k \boldsymbol{d}^k)^T \boldsymbol{d}^k \geq c_2 \nabla f(\boldsymbol{x}^k)^T \boldsymbol{d}^k \tag{3}$$

with $c_2 \in (c_1, 1)$.

The two conditions (2) and (3) are called Wolfe conditions.

Denote the angle between $\boldsymbol{d}^k$ and $-\nabla f(\boldsymbol{x}^k)$ by $\Theta_k$:

$$\cos(\Theta_k) = \frac{-\nabla f(\boldsymbol{x}^k)^T \boldsymbol{d}^k}{\|\nabla f(\boldsymbol{x}^k)\| \|\boldsymbol{d}^k\|}.$$

Assumptions on $f : \mathbb{R}^n \to \mathbb{R}$: continuously differentiable, derivative is Lipschitz-continuous, $f$ is bounded from below.
Method: descent algorithm with Wolfe-conditions.
Then:

$$\sum_{k \geq 0} \cos^2(\Theta_k) \|\nabla f(\boldsymbol{x}^k)\|^2 < \infty.$$

Denote the angle between $\boldsymbol{d}^k$ and $-\nabla f(\boldsymbol{x}^k)$ by $\Theta_k$:

$$\cos(\Theta_k) = \frac{-\nabla f(\boldsymbol{x}^k)^T \boldsymbol{d}^k}{\|\nabla f(\boldsymbol{x}^k)\| \|\boldsymbol{d}^k\|}.$$

Assumptions on $f : \mathbb{R}^n \to \mathbb{R}$: continuously differentiable, derivative is Lipschitz-continuous, $f$ is bounded from below.
Method: descent algorithm with Wolfe-conditions.
Then:

$$\sum_{k \geq 0} \cos^2(\Theta_k) \|\nabla f(\boldsymbol{x}^k)\|^2 < \infty.$$

In particular: If $\cos(\Theta_k) \geq \delta > 0$, then $\lim_{k\to\infty} \|\nabla f(\boldsymbol{x}^k)\| = 0,$.

# Descent algorithm

Alternative to Wolfe step length: Find $\alpha_k$ that satisfies the Armijo condition:

$$f(\boldsymbol{x}^k + \alpha_k \boldsymbol{d}^k) \leq f(\boldsymbol{x}^k) + c_1 \alpha_k \nabla f(\boldsymbol{x}^k)^T \boldsymbol{d}^k, \tag{4}$$

where $c_1 \in (0, 1)$.

# Descent algorithm

Alternative to Wolfe step length: Find $\alpha_k$ that satisfies the Armijo condition:

$$f(\boldsymbol{x}^k + \alpha_k \boldsymbol{d}^k) \leq f(\boldsymbol{x}^k) + c_1 \alpha_k \nabla f(\boldsymbol{x}^k)^T \boldsymbol{d}^k, \tag{4}$$

where $c_1 \in (0, 1)$.

Use backtracking linesearch to find a step length that is large enough:

- Start with (large) step length $\alpha_k^0 > 0$.
- If it satisfies (4), accept the step length.
- Else, compute $\alpha_k^{i+1} := \rho \alpha_k^i$ with $\rho < 1$ (usually, $\rho = 0.5$) and go back to previous step.

This also leads to a globally converging method to a stationary point.

Let us consider a simple case, where $f$ is quadratic:

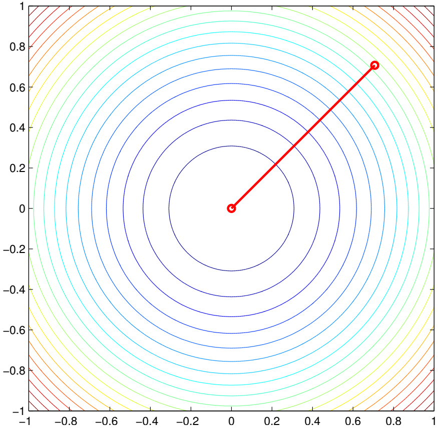
$$f(\boldsymbol{x}) := \frac{1}{2}\boldsymbol{x}^T Q \boldsymbol{x} - \boldsymbol{b}^T \boldsymbol{x},$$

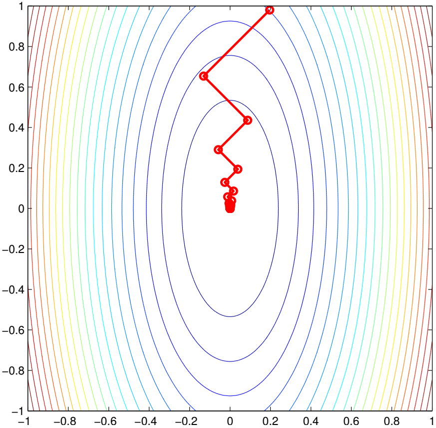
where $Q$ is spd. The gradient is $\nabla f(x) = Q\boldsymbol{x} - b$, and minimizer $\boldsymbol{x}^*$ is solution to $Q\boldsymbol{x} = \boldsymbol{b}$. Using exact line search, the convergence is:

$$\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\|_Q^2 \leq \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} \|\boldsymbol{x}^k - \boldsymbol{x}^*\|_Q^2$$

(linear convergence with rate depending on eigenvalues of $Q$)

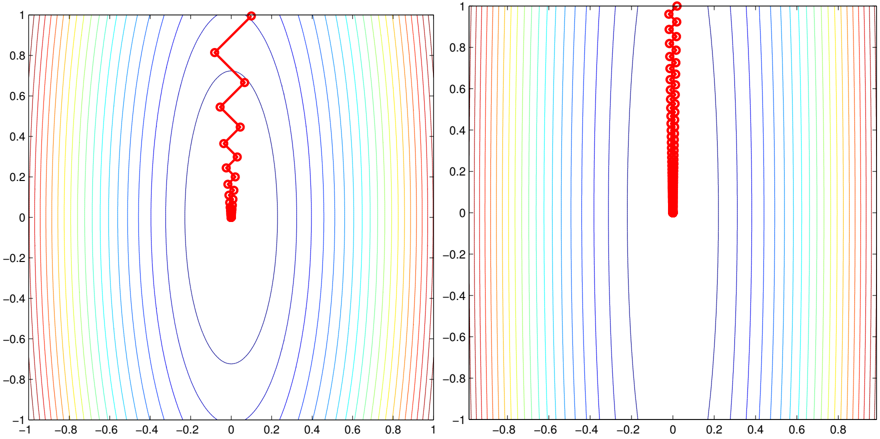
# Descent algorithms
## Convergence of steepest descent



# Descent algorithms

## Convergence of steepest descent



Newton's method: Assumptions on f: $2\times$differentiable with Lipschitz-continuous Hessian $\nabla^2 f(\boldsymbol{x}^k)$. Hessian is positive definite in a neighborhood around solution $\boldsymbol{x}^*$.

Assumptions on starting point: $\boldsymbol{x}^0$ sufficient close to $\boldsymbol{x}^*$.

Then: Quadratic convergence of Newton's method with $\alpha_k = 1$, and $\|\nabla f(\boldsymbol{x}^k)\| \to 0$ quadratically.

Newton's method: Assumptions on f: 2×differentiable with Lipschitz-continuous Hessian $\nabla^2 f(\boldsymbol{x}^k)$. Hessian is positive definite in a neighborhood around solution $\boldsymbol{x}^*$.

Assumptions on starting point: $\boldsymbol{x}^0$ sufficient close to $\boldsymbol{x}^*$.

Then: Quadratic convergence of Newton's method with $\alpha_k = 1$, and $\|\nabla f(\boldsymbol{x}^k)\| \to 0$ quadratically.

Equivalent to Newton's method for solving $\nabla f(\boldsymbol{x}) = 0$, if Hessian is positive.

# Descent algorithm

## Newton's method

At a point $\boldsymbol{x}^k$ further away from the minimizer, the Hessian $\nabla^2 f(\boldsymbol{x}^k)$ is not necessarily positive definite, and thus

$$\boldsymbol{d}^k = -\nabla^2 f(\boldsymbol{x}^k)^{-1} \nabla f(\boldsymbol{x}^k)$$

might not be defined. That is,

$$\nabla^2 f(\boldsymbol{x}^k) \boldsymbol{d}^k = -\nabla f(\boldsymbol{x}^k)$$

cannot be solved.

At a point $\boldsymbol{x}^k$ further away from the minimizer, the Hessian $\nabla^2 f(\boldsymbol{x}^k)$ is not necessarily positive definite, and thus

$$\boldsymbol{d}^k = -\nabla^2 f(\boldsymbol{x}^k)^{-1} \nabla f(\boldsymbol{x}^k)$$

might not be defined. That is,

$$\nabla^2 f(\boldsymbol{x}^k) \boldsymbol{d}^k = -\nabla f(\boldsymbol{x}^k)$$

cannot be solved. Remedies:

- Modify $\nabla^2 f(\boldsymbol{x}^k)$ by only taking the part that is guaranteed to be spd. An example is the Gauss-Newton method for nonlinear least squares problems
- Add a sufficiently positive matrix to the Hessian, e.g., use $B_k := \nabla^2 f(\boldsymbol{x}^k) + \lambda^k I$, with $\lambda^k > \geq 0$.
- Newton-conjugate gradient (Newton-CG) method (see next slide)

# Descent algorithm

## Newton-CG method

Compute descent direction $\boldsymbol{d}$ by approximately solving

$$\nabla^2 f(\boldsymbol{x}^k)\boldsymbol{d}^k = -\nabla f(\boldsymbol{x}^k)$$

using the conjugate gradient (CG) method. Initialize CG iteration with $0$ and terminate when "a negative curvature is found" (a parameter in the CG method becomes negative).

- Always results in a descent direction
- Only requires the application of the Hessian to vectors (rather than the explicit Hessian)—advantage for PDE-constrained (inverse) problems!

# Calculus of variations and weak forms

- The theory of *calculus of variations* concerns the minimization of functionals (i.e., mappings from a set of functions to the real numbers).
- These optimization problems can be seen as the infinite-dimensional version of the finite-dimensional optimization problem.
- We will use examples to demonstrate basic principles in variational calculus using prototype model problems.

- Suppose we are interested in finding solutions $u^\star$ to the minimization problem:

$$\min_{u \in \mathcal{V}_0} \Pi(u), \quad \mathcal{V}_0 = \{v \in H^1(\Omega) : v\big|_{\Gamma_D} = 0\}$$

where

$$\Pi(u) := \frac{1}{2} \int_\Omega k \nabla u \cdot \nabla u \, dx - \int_\Omega f u \, dx - \int_{\Gamma_N} h u \, ds$$

  - $H^1(\Omega)$ is the Sobolev space of functions in $L^2(\mathcal{D})$ with square integrable derivatives
  - $\Omega \subset \mathbb{R}^2$ is an open bounded domain with sufficiently smooth boundary $\Gamma = \Gamma_D \cup \Gamma_N$, $\Gamma_D \cap \Gamma_N = \emptyset$
  - $u(\boldsymbol{x}) \in \mathcal{V}_0$ (it could describe for instance the transverse deflection of a membrane at a point $\boldsymbol{x} \in \Omega$)
  - $f \in L^2(\Omega)$ and $h \in L^2(\Gamma_N)$ given functions
  - $k = k(\boldsymbol{x})$ is assumed to be a strictly positive and bounded function (it could for instance describe the stiffness of a membrane)

- A minimum $u^\star$ is characterized by

$$\Pi(u^\star + \varepsilon \tilde{u}) \geq \Pi(u^\star), \forall \tilde{u} \in \mathcal{V}_0, \text{ and } \varepsilon > 0 \text{ with } u^\star + \varepsilon \tilde{u} \in \mathcal{V}_0.$$

- Thus, a minimum $u^\star$ must satisfy the *Euler-Lagrange* conditions for stationarity, namely

$$\frac{\partial \Pi(u^\star + \varepsilon \tilde{u})}{\partial \varepsilon}\Big|_{\varepsilon=0} = 0 \text{ for all } \tilde{u} \in \mathcal{V}_0.$$

- Differentiating with respect to $\varepsilon$ we obtain

$$\frac{\partial \Pi(u^\star + \varepsilon \tilde{u})}{\partial \varepsilon} = \int_\Omega k \nabla (u^\star + \varepsilon \tilde{u}) \cdot \nabla \tilde{u} \, dx - \int_\Omega f \tilde{u} \, dx - \int_{\Gamma_N} h \tilde{u} \, dx$$

- Setting $\varepsilon = 0$, we obtain the *weak* (or *variational*) form: Find $u^\star \in \mathcal{V}_0$ such that

$$\int_\Omega k \nabla u^\star \cdot \nabla \tilde{u} \, dx - \int_\Omega f \tilde{u} \, dx - \int_{\Gamma_N} h \tilde{u} \, dx = 0, \quad \text{for all } \tilde{u} \in \mathcal{V}_0.$$

- Use Green's identity, which is a multidimensional version of integration-by-parts. The identity states that for all $u, v \in H^1(\Omega)$ holds

$$\int_\Omega k\nabla u \cdot \nabla v \, dx = -\int_\Omega u \nabla \cdot (k \nabla v)\, dx + \int_{\partial\Omega} uk\nabla v \cdot \boldsymbol{n}\, ds$$

- Using this identity for the first term in the weak form we obtain

$$\begin{aligned} 0 &= \int_\Omega -\nabla \cdot (k\nabla u^\star)\tilde{u}\, dx + \int_{\partial\Omega} (k\nabla u^\star \cdot \boldsymbol{n})\tilde{u}\, ds \\ &- \int_\Omega f\tilde{u}\, dx - \int_{\Gamma_N} h\tilde{u}\, dx \\ &= \int_\Omega -\left[f + \nabla \cdot (k\nabla u^\star)\right] \tilde{u}\, dx + \int_{\Gamma_N} \left[(k\nabla u^\star \cdot \boldsymbol{n}) - h\right] \tilde{u}\, ds \end{aligned}$$

for all $\tilde{u} \in \mathcal{V}_0$.

- Since $\tilde{u}$ is arbitrary, this implies that the factors multiplying $\tilde{u}$ must vanish. (This is a very common argument in variational calculus.)
- Since $\tilde{u}$ is arbitrary in $\Omega$

$$-\nabla \cdot (k\nabla u^\star) = f \text{ on } \Omega.$$

- Since $u^\star$ is in $\mathcal{V}_0$, it satisfies the Dirichlet boundary condition

$$u^\star = 0 \text{ on } \Gamma_D.$$

- Since $\tilde{u} \in \mathcal{V}_0$ is arbitrary on $\Gamma_N$, it also satisfies the Neumann boundary condition

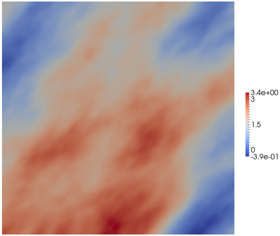
$$k\nabla u^\star \cdot \boldsymbol{n} = h \text{ on } \Gamma_N.$$

- These equations are the *strong form* for the variational problem.

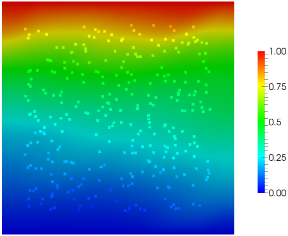
# Coefficient field inversion in an elliptic PDE

## The elliptic equation in strong form

$$
\begin{aligned}
-\nabla \cdot (e^m \nabla u) &= f \quad \text{in } \mathcal{D}\\
u &= g \quad \text{on } \Gamma_D\\
e^m \nabla u \cdot \boldsymbol{n} &= h \quad \text{on } \Gamma_N
\end{aligned}
$$




- $u$ is the state variable (e.g., pressure field)
- $m$ is the inversion parameter (e.g., the log permeability field))
- $f \in L^2(\mathcal{D})$ is a source term
- $g \in H^{1/2}(\Gamma_D)$ and $h \in L^2(\Gamma_N)$ are Dirichlet and Neumann boundary data
- $\mathcal{D} \subset \mathbb{R}^2$ is an open bounded domain with sufficiently smooth boundary $\Gamma = \Gamma_D \cup \Gamma_N$, $\Gamma_D \cap \Gamma_N = \emptyset$



# Coefficient field inversion in an elliptic PDE

## The elliptic equation in weak form

- Define the spaces

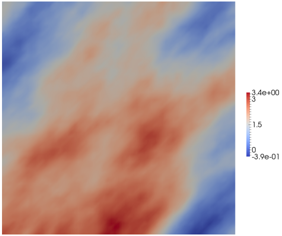
$$\mathcal{V}_g = \{v \in H^1(\mathcal{D}) : v\big|_{\Gamma_D} = g\}$$
$$\mathcal{V}_0 = \{v \in H^1(\mathcal{D}) : v\big|_{\Gamma_D} = 0\},$$

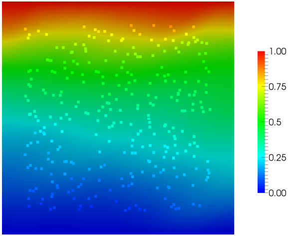
  where $H^1(\mathcal{D})$ is the Sobolev space of functions in $L^2(\mathcal{D})$ with square integrable derivatives.

- The weak form: find $u \in \mathcal{V}_g$ such that

$$\langle e^m \nabla u, \nabla p\rangle = \langle f, p\rangle + \langle h, p\rangle_{\Gamma_N}, \quad \forall p \in \mathcal{V}_0,$$

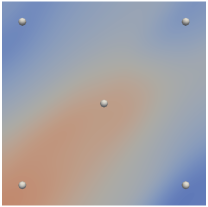
  where $\langle \cdot, \cdot \rangle$ and $\langle \cdot, \cdot \rangle_{\Gamma_N}$ denote the standard inner products in $L^2(\mathcal{D})$ and $L^2(\Gamma_N)$, respectively.

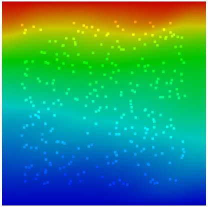


# The inverse problem

$$\min_m \mathcal{J}(m) := \frac{1}{2} \left\| \mathcal{B}u(m) - \boldsymbol{d} \right\|^2_{\boldsymbol{\Gamma}_{\text{noise}}} + \frac{1}{2} \left\| m - m_{\text{pr}} \right\|^2_{\mathcal{C}^{-1}_{\text{prior}}}$$





- $u$ solves the elliptic PDE for given parameter $m$
- $\mathcal{B}$ is a linear observation operator that extracts measurements from $u$
- $m_{\text{pr}}$ is the mean of the log coefficient field
- $\boldsymbol{d} \in \mathbb{R}^q$ is a given data vector
- $\boldsymbol{\Gamma}_{\text{noise}}$ and $\mathcal{C}^{-1}_{\text{prior}}$ appropriately chosen weights (to account for noise in measurements and properties of the parameter)

# Gradient computation

## using standard variational approach

- The Lagrangian functional:

$$\mathcal{L}(u,m,p) := \mathcal{J}(m) + \langle e^m \nabla u, \nabla p \rangle - \langle f, p \rangle - \langle p, h \rangle_{\Gamma_N},$$

where $p \in \mathcal{V}_0$ is the Lagrange multiplier.

- Optimality system:

$$\begin{aligned}
\mathcal{L}_p[\tilde{p}] &= \langle e^m \nabla u, \nabla \tilde{p} \rangle - \langle f, \tilde{p} \rangle - \langle \tilde{p}, h \rangle_{\Gamma_N} &&= 0 \quad \text{(state)}\\
\mathcal{L}_u[\tilde{u}] &= \langle e^m \nabla \tilde{u}, \nabla p \rangle + \langle \mathcal{B}^* \boldsymbol{\Gamma}_{\text{noise}} (\mathcal{B}u - \boldsymbol{d}), \tilde{u} \rangle &&= 0 \quad \text{(adjoint)}\\
\mathcal{L}_m[\tilde{m}] &= \langle m - m_{\text{pr}}, \tilde{m} \rangle_{\mathcal{C}_{\text{prior}}^{-1}} + \langle \tilde{m} e^m \nabla u, \nabla p \rangle &&= 0 \quad \text{(gradient)}
\end{aligned}$$

for all variations $(\tilde{u}, \tilde{p}) \in \mathcal{V}_0 \times \mathcal{V}_0$.

- The gradient (also in weak form):

$$\mathcal{G}(m)(\tilde{m}) = \langle m - m_{\text{pr}}, \tilde{m} \rangle_{\mathcal{C}_{\text{prior}}^{-1}} + \langle \tilde{m} e^m \nabla u, \nabla p \rangle,$$

where $u$ and $p$ are solutions to the state and adjoint equations, respectively

# The action of the Hessian in a direction

- The *meta*-Lagrangian functional

$$
\begin{aligned}
\mathcal{L}^{\mathcal{G}}(u,m,p;\hat{u},\hat{m},\hat{p}) :=\ & \mathcal{G}(m)(\hat{m}) && \text{(gradient)}\\
&+ \langle e^m \nabla u, \nabla \hat{p}\rangle - \langle f,\hat{p}\rangle - \langle \hat{p}, h\rangle_{\Gamma_N} && \text{(state)}\\
&+ \langle e^m \nabla \hat{u}, \nabla p\rangle + \left\langle \mathcal{B}^* \boldsymbol{\Gamma}_{\text{noise}}^{-1}(\mathcal{B}u - \boldsymbol{d}), \hat{u}\right\rangle && \text{(adjoint)}
\end{aligned}
$$

- The Hessian in a direction $\hat{m}$ as the variation of $\mathcal{L}^{\mathcal{G}}$ with respect to $m$:

$$
\mathcal{H}(m,\tilde{m})(\hat{m}) = \langle \tilde{m} e^m \nabla \hat{u}, \nabla p\rangle + \langle \hat{m}, \tilde{m}\rangle_{\mathcal{C}_{\text{prior}}^{-1}} + \langle \tilde{m}\hat{m} e^m \nabla u, \nabla p\rangle + \langle \tilde{m} e^m \nabla u, \nabla \hat{p}\rangle
$$

- $u$ and $p$ are the solutions of the state and adjoint equations
- $\hat{u}$, $\hat{p}$ are the *incremental state and adjoint* variables

# The action of the Hessian in a direction

## The incremental ("second order") state and adjoint

- The Hessian in a direction $\hat{m}$ as the variation of $\mathcal{L}^{\mathcal{G}}$ with respect to $m$:

$$\mathcal{H}(m,\tilde{m})(\hat{m}) = \langle \tilde{m} e^m \nabla \hat{u}, \nabla p\rangle + \langle \hat{m}, \tilde{m}\rangle_{\mathcal{C}_{\text{prior}}^{-1}} + \langle \tilde{m}\hat{m} e^m \nabla u, \nabla p\rangle + \langle \tilde{m} e^m \nabla u, \nabla \hat{p}\rangle$$

- $u$ and $p$ are the solutions of the state and adjoint equations
- $\hat{u}$, $\hat{p}$ are the *incremental state and adjoint* variables, and are obtained by taking variations of $\mathcal{L}^{\mathcal{G}}$ with respect to $p$ and $u$:

$$\langle e^m \nabla \hat{u}, \nabla \tilde{p}\rangle + \langle \hat{m} e^m \nabla u, \nabla \tilde{p}\rangle = 0,\ \forall \tilde{p} \in \mathcal{V}_0 \quad \text{(inc. state)}$$

$$\left\langle \mathcal{B}^* \mathbf{\Gamma}_{\text{noise}}^{-1} \mathcal{B}\hat{u}, \tilde{u}\right\rangle + \langle \hat{m} e^m \nabla \tilde{u}, \nabla p\rangle + \langle e^m \nabla \tilde{u}, \nabla \hat{p}\rangle = 0,\ \forall \tilde{u} \in \mathcal{V}_0 \quad \text{(inc. adjoint).}$$

# The inexact Newton-conjugate gradient algorithm

**Algorithm 1** The inexact Newton-CG algorithm.

$i \leftarrow 0$
Given $\boldsymbol{m}_0$ solve the state equation to obtain $\boldsymbol{u}_0$.
Given $\boldsymbol{m}_0$ and $\boldsymbol{u}_0$ compute the cost $c_0$.
**while** not converged and $i <$ max_iter **do**
  Given $\boldsymbol{m}_i$ and $\boldsymbol{u}_i$ solve the adjoint equation to obtain $\boldsymbol{p}_i$.
  Given $\boldsymbol{m}_i$, $\boldsymbol{u}_i$ and $\boldsymbol{p}_i$ evaluate the reduced gradient $\boldsymbol{g}_i$.
  **if** $\|\boldsymbol{g}_i\| \le \tau$ **then**
    **break**
  **end if**
  Given $\|\boldsymbol{g}_i\|$ compute the tolerance for the conjugate gradient.
  Solve the linear system $\mathbf{H}_i \widehat{\boldsymbol{m}}_i = -\boldsymbol{g}_i$ using conjugate gradient.
  $j \leftarrow 0$, $\alpha \leftarrow 1$
  **while** $j <$ max_backtracking_iter **do**
    Set $\boldsymbol{m}_{i+1} = \boldsymbol{m}_i + \alpha \widehat{\boldsymbol{m}}_i$.
    Given $\boldsymbol{m}_{i+1}$ solve the state equation to obtain $\boldsymbol{u}_{i+1}$
    Given $\boldsymbol{m}_{i+1}$ and $\boldsymbol{u}_{i+1}$ compute the cost $c_{i+1}$.
    **if** $c_{i+1} < c_i - c_{\text{armijo}}(\boldsymbol{g}_i, \widehat{\boldsymbol{m}}_i)$ **then**
      **break**
    **end if**
    $\alpha \leftarrow \alpha/2, \quad j \leftarrow j+1$
  **end while**
  $i \leftarrow i+1$
**end while**

# Scalability of the inverse solver

## Inexact Newton-CG method applied to an ice sheet inverse problem

| #sdof | #mdof | #N | #CG | avgCG | #PDE solves |
|---|---|---|---|---|---|
| 95,796 | 10,371 | 42 | 2718 | 65 | 7031 |
| 233,834 | 25,295 | 39 | 2342 | 60 | 6440 |
| 848,850 | 91,787 | 39 | 2577 | 66 | 6856 |
| 3,372,707 | 364,649 | 39 | 2211 | 57 | 6193 |
| 22,570,303 | 1,456,225 | 40 | 1923 | 48 | 5376 |

- **#sdof**: number of degrees of freedom for the state variables;
- **#mdof**: number of degrees of freedom for the inversion parameter field;
- **#N**: number of Newton iterations;
- **#CG**, **avgCG**: total and average (per Newton iteration) number of CG iterations;
- **#forward/state solves**: total number of PDE solves (from forward, adjoint, and incremental forward and adjoint problems)

The cost of solving the inverse problem by the inexact Newton-CG method, measured by the number of forward/state solves, is independent of the parameter dimension as well as the data dimension.

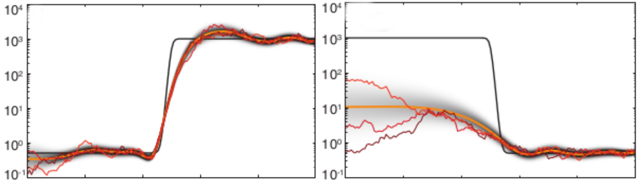
# Numerical results

. . . show the PoissonDeterministic.ipynb ipython notebook . . .

# Outline

# Why is a statistical perspective useful in inverse problems?

- To characterize the *uncertainty* in the solution of the inverse problem
  - To understand how this uncertainty depends on the number and quality of observations, features of the forward model, prior knowledge about the parameters, etc.
- To make probabilistic *predictions*
- To choose "useful" (for inversion) observations and design experiments
- To address questions of model error, model selection, etc.

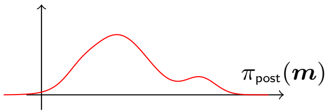


# Bayesian approach to inverse problems

Inverse problem

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

Interpret $\boldsymbol{m}$, $\boldsymbol{d}$ as random variables; solution of inverse problem is a probability density function $\pi_{\text{post}}(\boldsymbol{m})$ for $\boldsymbol{m}$:




## Remarks:

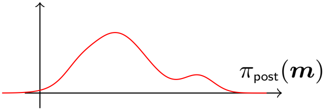
- systematic method to quantify measurement errors and incorporate prior knowledge
- allows quantification of uncertainty in reconstruction
- related to regularization approach (think of $\pi_{\text{post}}(\boldsymbol{m})$ as $\exp(-\mathcal{J}(\boldsymbol{m}))$)
- high-dimensional probability density

# Bayesian approach to inverse problems

Inverse problem

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

Interpret $\boldsymbol{m}$, $\boldsymbol{d}$ as random variables; solution of inverse problem is a probability density function $\pi_{\text{post}}(\boldsymbol{m})$ for $\boldsymbol{m}$:




## Target:

- characterize $\pi_{\text{post}}(\boldsymbol{m})$ statistically (mean, covariance, MAP point...)
- for functions $\boldsymbol{m}$ (large vectors after discretization)
- for expensive $\boldsymbol{f}(\cdot)$
- exploit connection to PDE-constrained optimization

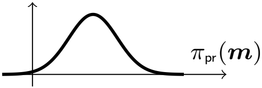
# Bayes formula (finite dimensions)

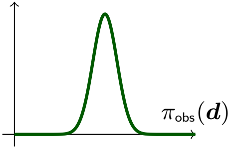
Given:

$$\begin{aligned} \pi_{\text{pr}}(\boldsymbol{m}) &: \text{ prior p.d.f. of model parameters } \boldsymbol{m} \\ \pi_{\text{obs}}(\boldsymbol{d}) &: \text{ prior p.d.f. of measurement error } \boldsymbol{d} \\ \pi_{\text{model}}(\boldsymbol{d}|\boldsymbol{m}) &: \text{ conditional p.d.f. combining } \boldsymbol{d} \text{ and } \boldsymbol{m} \text{ (model)} \end{aligned}$$

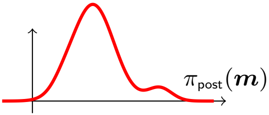
Then, the *posterior p.d.f. of the model parameters* is given by:

$$\begin{aligned} \pi_{\text{post}}(\boldsymbol{m}|\boldsymbol{d}) &\propto \pi_{\text{pr}}(\boldsymbol{m}) \int_{\mathcal{D}} \frac{\pi_{\text{obs}}(\boldsymbol{d})\pi_{\text{model}}(\boldsymbol{d}|\boldsymbol{m})}{\mu(\boldsymbol{d})} d\boldsymbol{m} \\ &\propto \pi_{\text{pr}}(\boldsymbol{m})\, \pi_{\text{like}}(\boldsymbol{d}|\boldsymbol{m}) \end{aligned}$$




Model relating $\boldsymbol{m}$ and $\boldsymbol{d}$




Bayes' formula:

$\pi_{\text{post}}(\boldsymbol{m}) \propto \pi_{\text{pr}}(\boldsymbol{m})\ \pi_{\text{like}}(\boldsymbol{y}|\boldsymbol{m})$

# Bayes formula (infinite dimensions)

(A. Stuart, Acta Numerica, (2010))

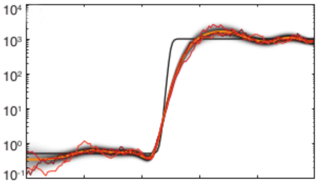
- invert for $m \in L^2(\Omega)$
- prior measure of parameters $\mu_0 := \mathcal{N}(m_0, \mathcal{A}^{-2})$ on $L^2(\Omega)$
- covariance operator is given by the inverse of differential operator $\mathcal{A}^2$, where $\mathcal{A}(m) := -\alpha \boldsymbol{\nabla} \cdot (\boldsymbol{\Theta} \boldsymbol{\nabla} m) + \alpha m$
- Bayesian solution of the inverse problem defined as conditional measure $\mu^d$ of $m$ given the data $\boldsymbol{d} \in \mathbb{R}^q$, where

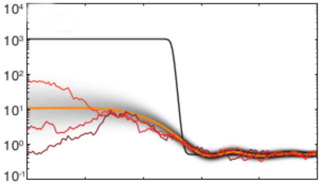
$$\frac{d\mu^d}{d\mu_0} = \frac{1}{Z}\pi_{\text{like}}(\boldsymbol{d}|m) \propto \exp\left(-\frac{1}{2}\|\boldsymbol{f}(m) - \boldsymbol{d}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{noise}}}\right)$$

  is the Radon-Nikodym derivative w.r.t. $\mu_0$ and $\boldsymbol{f}(m)$ is the parameter-to-observable map and $\boldsymbol{\Gamma}_{\text{noise}}$ the noise covariance operator
- leads to well-posed Bayesian inverse problem (in 2D and 3D); exploits fast elliptic solvers

# What information to extract from the posterior?

- Posterior mean of $m$; *maximum a posteriori (MAP) estimate of* $m$
- Posterior covariance or higher moments of $m$
- Credible intervals (i.e., higher posterior density regions)
- Posterior realizations: for direct assessment, or to estimate posterior predictions or other posterior expectations



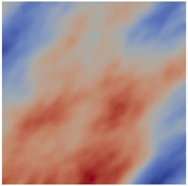
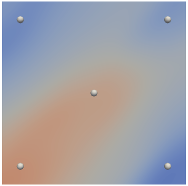
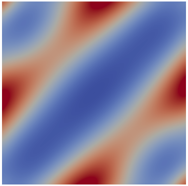



# Prior distributions

- In ill-posed inverse problems, prior plays an essential role
- Intuitive idea when building a prior: assign lower probability to values of $m$ that you don't expect to see, higher probability to values of $m$ that you do expect to see
- Examples of priors:
  - Gaussian priors derived from differential operators
  - Gaussian processes with specified covariance kernel
  - Hierarchical priors
  - Besov space priors

# Prior distributions

- In ill-posed inverse problems, prior plays an essential role
- Intuitive idea when building a prior: assign lower probability to values of $m$ that you don't expect to see, higher probability to values of $m$ that you do expect to see
- Examples of priors:
  - Gaussian priors derived from differential operators
  - Gaussian processes with specified covariance kernel
  - Hierarchical priors
  - Besov space priors

# Example of a prior model

- We consider the prior as a Gaussian random field $\mathcal{N}(m_{\text{pr}}, \mathcal{C}_{\text{prior}})$ on $L^2(\Omega)$
- The covariance operator $\mathcal{C}_{\text{prior}}$ given by the inverse of differential operator $\mathcal{A}^2$

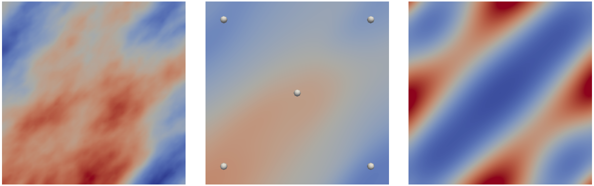
$$\mathcal{A} = \widetilde{\mathcal{A}} + p\mathcal{M}, \; \widetilde{\mathcal{A}}(m) := -\gamma \boldsymbol{\nabla} \cdot (\boldsymbol{\Theta} \boldsymbol{\nabla} m) + \alpha m$$

  - $\gamma$, $\alpha$, $p$, and $\boldsymbol{\Theta}$, prior parameters
  - $\mathcal{M} := \sum_{i=1}^{5} \delta_i I$, i.e., assume that we can measure the parameter (e.g., log permeability field) at five points and use a *mollifier function*

$$\delta_i(x) = \exp\left( -\frac{\gamma^2}{\alpha^2} \|x - x_i\|^2_{\boldsymbol{\Theta}^{-1}} \right), \quad i = 1, \ldots, 5$$

- Compute $m_{\text{pr}}$ as a regularized least-squares fit of these point observations:

$$m_{\text{pr}} = \arg\min_{m} \frac{1}{2} \langle m, m \rangle_{\widetilde{\mathcal{A}}} + \frac{p}{2} \langle m_{\text{true}} - m, m_{\text{true}} - m \rangle_{\mathcal{M}}$$



# Gaussian noise model

We assume additive Gaussian noise in the measurements:

$$\boldsymbol{d} = \boldsymbol{f}(\boldsymbol{m}) + \boldsymbol{e}, \quad \boldsymbol{e} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Gamma}_{\text{noise}})$$

Thus:

$$\pi_{\text{like}}(\boldsymbol{d}|\boldsymbol{m}) = \exp\left(-\tfrac{1}{2}(\boldsymbol{f}(\boldsymbol{m}) - \boldsymbol{d})^T \boldsymbol{\Gamma}_{\text{noise}}^{-1}(\boldsymbol{f}(\boldsymbol{m}) - \boldsymbol{d})\right)$$

- $\boldsymbol{d} \in \mathbb{R}^q$ the vector of (noisy) point-wise observations of the state $u$
- Assume the measurement noise at each location to be a multivariate Gaussian variable with mean $0$ and covariance $\boldsymbol{\Gamma}_{\text{noise}} = \sigma^2 \mathbf{I}$
- $\boldsymbol{f}(\boldsymbol{m}) = \mathcal{B}u(m)$ is the *parameter-to-observable* map

# The posterior pdf

If the prior is Gaussian with mean $\boldsymbol{m}_{\text{pr}}$ and covariance $\boldsymbol{\Gamma}_{\text{prior}}$, then we obtain for the posterior pdf:

$$\pi_{\text{post}}(\boldsymbol{m}) \propto \exp\left(-\tfrac{1}{2} \parallel \boldsymbol{f}(\boldsymbol{m}) - \boldsymbol{d} \parallel^2_{\boldsymbol{\Gamma}^{-1}_{\text{noise}}} - \tfrac{1}{2} \parallel \boldsymbol{m} - \boldsymbol{m}_{\text{pr}} \parallel^2_{\boldsymbol{\Gamma}^{-1}_{\text{prior}}}\right)$$

The "maximum a posteriori" point is

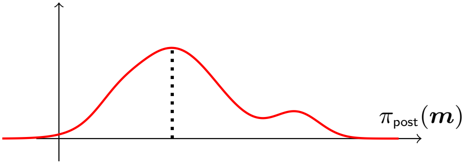
$$\begin{aligned} \boldsymbol{m}_{\text{MAP}} &\stackrel{\text{def}}{=} \arg\ \max_{\boldsymbol{m}}\ \pi_{\text{post}}(\boldsymbol{m}) \\ &= \arg\ \min_{\boldsymbol{m}}\ \frac{1}{2}\,\|\boldsymbol{f}(\boldsymbol{m}) - \boldsymbol{d}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{noise}}} + \frac{1}{2}\,\|\boldsymbol{m} - \boldsymbol{m}_{\text{pr}}\|^2_{\boldsymbol{\Gamma}^{-1}_{\text{prior}}} \end{aligned}$$

$\Rightarrow$ deterministic inverse problem with appropriate weighted norms!

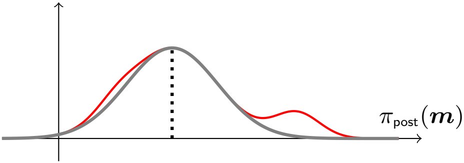
# Challenges for large-scale Bayesian inverse problems

- Method of choice is to sample the posterior density using Markov chain Monte Carlo (MCMC).
- For inverse problems characterized by high-dimensional parameter spaces and expensive forward simulations, standard MCMC methods become prohibitive.
- Standard MCMC methods view the parameter-to-observable map as a black-box.
- Goals:
  - overcome bottlenecks of MCMC: develop MCMC methods that reduce effective problem dimension by exploiting the structure of the problem (as has been done successfully in PDE-constrained optimization)
  - apply to large-scale Bayesian inverse problems
  - approximate the posterior pdf

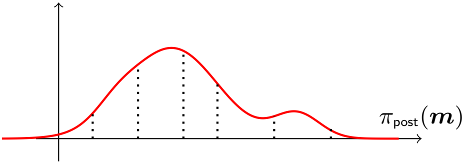
# Approximation of the posterior pdf




MAP estimation




Gaussian approximation around MAP

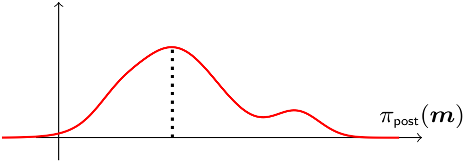



MCMC sampling

# MAP estimation

Despite the explicit form of $\pi_{\text{post}}(\boldsymbol{m}|\boldsymbol{d})$, its exploration is difficult due to:

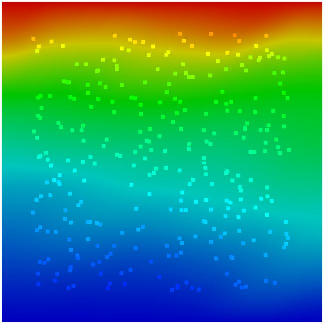
- the high/infinite dimension of $\boldsymbol{m}$
- the expensive PDE-based parameter-to-observable map $\boldsymbol{f}$




Find the maximum a posteriori (MAP) point.

- requires solution of PDE-constrained optimization problem $\sim$ deterministic inversion
- use, e.g., reduced space Newton-conjugate gradient method, or a Newton-type full space method.
- requires gradients and Hessians, computed through adjoint equations

# MAP estimation



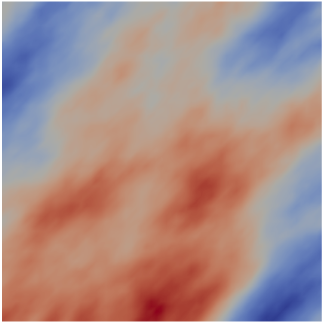


Figure: Left: The pressure field $u$ (left) obtained by solving the state equation with $m_{\text{true}}$ (right).

# MAP estimation

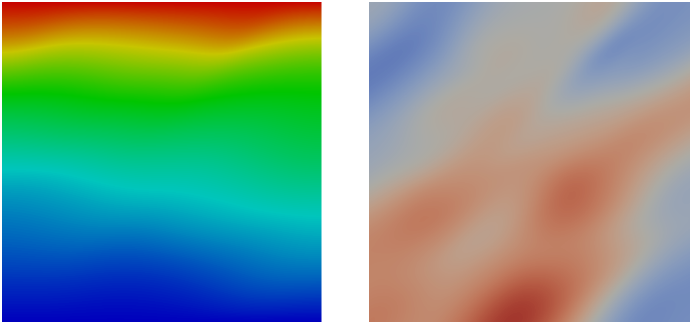


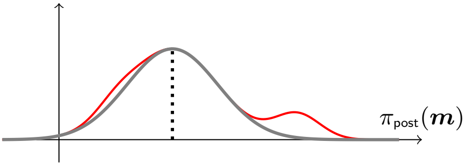
Figure: Left: The pressure field $u$ (left) obtained by solving the state equation with the reconstructed parameter $m$ (right).

# Gaussian approximation around MAP

Assume linear(ized) parameter-to-observable map

$$\boldsymbol{d} = \boldsymbol{F}\boldsymbol{m}.$$

Then, the posterior p.d.f. is:




$$\pi_{\text{post}}(\boldsymbol{m}) \propto \exp\left(-\frac{1}{2} \parallel \boldsymbol{F}\boldsymbol{m} - \boldsymbol{d} \parallel^2_{\boldsymbol{\Gamma}^{-1}_{\text{noise}}} - \frac{1}{2} \parallel \boldsymbol{m} - \boldsymbol{m}_{\text{pr}} \parallel^2_{\boldsymbol{\Gamma}^{-1}_{\text{prior}}}\right)$$
$$= \exp\left(-\frac{1}{2}(\boldsymbol{m} - \boldsymbol{m}_{\text{MAP}})^T(\boldsymbol{F}^T\boldsymbol{\Gamma}^{-1}_{\text{noise}}\boldsymbol{F} + \boldsymbol{\Gamma}^{-1}_{\text{prior}})(\boldsymbol{m} - \boldsymbol{m}_{\text{MAP}})\right)$$

Thus, the posterior is also Gaussian, i.e., $\boldsymbol{m} \sim \mathcal{N}(m_{\text{MAP}}, \boldsymbol{\Gamma}_{\text{post}})$. The covariance matrix is the inverse Hessian of $\mathcal{J}(\cdot)$, i.e.,

$$\boldsymbol{\Gamma}^{-1}_{\text{post}} = \boldsymbol{H} = \boldsymbol{F}^T\boldsymbol{\Gamma}^{-1}_{\text{noise}}\boldsymbol{F} + \boldsymbol{\Gamma}^{-1}_{\text{prior}}$$
$$= \nabla^2_{\boldsymbol{m}}(-\log \pi_{\text{post}})$$

# Low rank approximation of data misfit Hessian

Under the assumption of Gaussian noise and linearized parameter-to-observables mapping

- Covariance estimate given by the inverse of the Hessian:

$$\begin{aligned}
\boldsymbol{\Gamma}_{\text{post}} = \boldsymbol{H}^{-1} &= \left( \boldsymbol{F}^T \boldsymbol{\Gamma}_{\text{noise}}^{-1} \boldsymbol{F} + \boldsymbol{\Gamma}_{\text{prior}}{}^{-1} \right)^{-1} \\
&= \boldsymbol{\Gamma}_{\text{prior}}^{1/2} \left( \boldsymbol{\Gamma}_{\text{prior}}^{1/2} \boldsymbol{F}^T \boldsymbol{\Gamma}_{\text{noise}}^{-1} \boldsymbol{F} \, \boldsymbol{\Gamma}_{\text{prior}}^{1/2} + \boldsymbol{I} \right)^{-1} \boldsymbol{\Gamma}_{\text{prior}}^{1/2} \\
&\approx \boldsymbol{\Gamma}_{\text{prior}}^{1/2} \left( \boldsymbol{V}_r \boldsymbol{\Lambda}_r \boldsymbol{V}_r^T + \boldsymbol{I} \right)^{-1} \boldsymbol{\Gamma}_{\text{prior}}^{1/2} \\
&= \boldsymbol{\Gamma}_{\text{prior}}^{1/2} \left[ \boldsymbol{I} - \boldsymbol{V}_r \boldsymbol{D}_r \boldsymbol{V}_r^T + \mathcal{O}\left( \sum_{i=r+1}^{n} \frac{\lambda_i}{\lambda_i + 1} \right) \right] \boldsymbol{\Gamma}_{\text{prior}}^{1/2}
\end{aligned}$$

  where $\boldsymbol{V}_r, \boldsymbol{\Lambda}_r$ are truncated eigenvector/values of prior-preconditioned data misfit Hessian, and $\boldsymbol{D}_r = \mathrm{diag}(\lambda_i/(\lambda_i + 1))$, and used the Sherman-Morrison-Woodbury th to invert/factor.

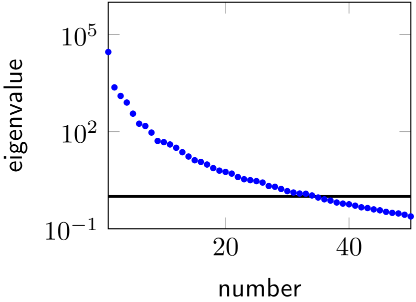
- Low-rank approximation of prior-preconditioned data misfit Hessian $\boldsymbol{\Gamma}_{\text{prior}}^{1/2} \boldsymbol{F}^T \boldsymbol{\Gamma}_{\text{noise}}^{-1} \boldsymbol{F} \, \boldsymbol{\Gamma}_{\text{prior}}^{1/2}$ is obtained with dimension-independent number of matvecs (using the randomized SVD or Lanczos iterative methods).

# Low-rank-based posterior covariance

- Posterior covariance is given by prior covariance less information gained from data:

$$\boldsymbol{\Gamma}_{\text{post}} \approx \boldsymbol{\Gamma}_{\text{prior}} - \boldsymbol{\Gamma}_{\text{prior}}^{1/2} \boldsymbol{V}_r \boldsymbol{D}_r \boldsymbol{V}_r^T \boldsymbol{\Gamma}_{\text{prior}}^{1/2}$$

- Log-linear plot of spectrum of prior-preconditioned data misfit Hessian. The low rank approximation captures the dominant, data-informed portion of the spectrum. (The eigenvalues are truncated at 0.2.):



# Spectrum of the Reduced Hessian

. . . show the HessianSpectrum.ipynb notebook . . .

# Sampling from large scale Gaussian random fields

- To sample from a small-scale multivariate Gaussian distribution, it is common to resort to a Cholesky factorization of the covariance matrix

$$\mathbf{\Gamma} = \mathbf{C}\mathbf{C}^T.$$

- If $\boldsymbol{\eta}$ is a vector of independent identically distributed Gaussian variables $\eta_i$ with zero mean ($\mathsf{E}\left[\boldsymbol{\eta}\right] = \mathbf{0}$) and unit variance ($\mathrm{cov}(\boldsymbol{\eta}) = \mathbf{I}$), then $\boldsymbol{x} = \mathbf{C}\boldsymbol{\eta}$ is such that

$$\mathrm{cov}(\boldsymbol{x}) = \mathsf{E}\left[\boldsymbol{x}\boldsymbol{x}^T\right] = \mathsf{E}\left[\mathbf{C}\boldsymbol{\eta}\boldsymbol{\eta}^T\mathbf{C}^T\right] = \mathbf{C}\mathsf{E}\left[\boldsymbol{\eta}\boldsymbol{\eta}^T\right]\mathbf{C}^T = \mathbf{C}\mathbf{C}^T = \mathbf{\Gamma}.$$

- Since a linear function of a Gaussian vector is still Guassian, we have that

$$\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma}).$$

- However, this approach is not feasible for large scale problems since it requires to compute a Cholesky factorization of the covariance matrix.
- We exploit a technique based on rectangular decompositions of the underlying finite element matrix to efficiently generate large scale samples from priors.

- Compute a *rectangular decomposition* of the form
$$\mathbf{A} = \mathbf{C}\mathbf{C}^T,$$
for any symmetric positive definite finite element matrix $\mathbf{A}$.
- The matrix $\mathbf{A}$ stems from finite element discretization of bilinear form $a(u_h, v_h)$:
$$A_{i,j} = a\left(\phi_i, \phi_j\right), \quad i, j = 1, \dots, n,$$
where $\{\phi_i\}_{i=1}^n$ is the finite element basis of the space $V_h$.
- Compute the local matrices $\mathbf{A}_e = \mathbf{B}^T \mathbf{D}_e \mathbf{B}$ which correspond to the restriction of the bilinear form $a$ to each element $e$ in the mesh.
- Using the *global-to-local* mapping of the degrees of freedom (dof) $\mathbf{G}_e$, the global matrix $\mathbf{A}$ is computed by summing all the local contributions as follows
$$\mathbf{A} = \sum_e \mathbf{G}_e^T \mathbf{A}_e \mathbf{G}_e = \sum_e \mathbf{G}_e^T \mathbf{B}^T \mathbf{D}_e \mathbf{B} \mathbf{G}_e.$$
- Here $\mathbf{B}$ is the element-independent dof-to-quadrature point basis evaluation matrix and $\mathbf{D}_e \in \mathbb{R}^{q \times q}$ is a (block) diagonal diagonal matrix at the quadrature points ($q$ denotes #quadrature nodes over all elements).

- Since, for any two elements $e_i$ and $e_j$ ( $i \neq j$) in the mesh, the sets of quadrature nodes relative to the elements $e_i$ and $e_j$ are disjoint we have that

$$\mathbf{C}_{e_i}\mathbf{C}_{e_j}^T = \delta_{ij}\mathbf{A}_{e_i}.$$

- Then the rectangular matrix $\mathbf{C} \in \mathbb{R}^{n \times q}$ defined as

$$\mathbf{C} = \sum_e \mathbf{G}_e^T \mathbf{B}^T \mathbf{D}_e^{\frac{1}{2}} = \sum_e \mathbf{C}_e$$

- Finally we have

$$\mathbf{C}\mathbf{C}^T = \left(\sum_e \mathbf{C}_e\right)\left(\sum_e \mathbf{C}_e\right)^T = \sum_e \mathbf{C}_e\mathbf{C}_e^T = \sum_e \mathbf{G}_e^T\mathbf{B}^T\mathbf{D}_e\mathbf{B}\mathbf{G}_e = \mathbf{A}.$$

# Sampling from the prior

## Using rectangular decomposition

- Key idea: exploit the structure of the covariance matrix and the assembly procedure of finite element matrices.
- For *Laplacian* priors, the inverse of the covariance matrix admits a sparse representation as a finite element matrix $\mathbf{R}$ stemming from the finite element discretization of a coercive symmetric differential operator.
- Then to sample from the distribution $\mathcal{N}\left(\mathbf{0}, \mathbf{R}^{-1}\right)$ we compute $\boldsymbol{x}$ as the solution of the linear system

$$\mathbf{R}\boldsymbol{x} = \mathbf{C}\boldsymbol{\eta}, \text{ where } \boldsymbol{\eta} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}_q\right),$$

  where $\mathbf{C} \in \mathbb{R}^{n \times q}$ $(q \geq n)$ is the rectangular factor of $\mathbf{R}$.
- We have

$$\mathsf{E}[\boldsymbol{x}\boldsymbol{x}^T] = \mathbf{R}^{-1}\mathbf{C}\mathsf{E}[\boldsymbol{\eta}\boldsymbol{\eta}^T]\mathbf{C}^T\mathbf{R}^{-1} = \mathbf{R}^{-1}\,\mathbf{R}\,\mathbf{R}^{-1} = \mathbf{R}^{-1},$$

  where we exploited the fact that $\mathsf{E}[\boldsymbol{\eta}\boldsymbol{\eta}^T] = \mathbf{I}_q$ and $\mathbf{C}\mathbf{C}^T = \mathbf{R}$.

This method is particularly efficient at large scale since:

- the matrix $\mathbf{C}$ is sparse and can be computed efficiently by exploiting the finite element assembly routine;
- it only involves the solution of a linear system in $\mathbf{R}$ for which efficient and scalable methods are available (conjugate gradient and algebraic multigrid preconditioner); and
- the stochastic dimension of $\boldsymbol{\eta}$ also scales linearly with the size of the problem.

# Sampling from the Gaussian approximation of the posterior

- Given a sample from the prior distribution $\boldsymbol{x} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}^{-1}\right)$ a sample from the Gaussian approximation of the posterior $\mathcal{N}\left(\mathbf{0}, (\mathbf{H}_{\text{misfit}} + \mathbf{R})^{-1}\right)$ can be computed as follow

$$\boldsymbol{y} = \left(\mathbf{I}_n - \mathbf{U}_r \mathbf{S}_r \mathbf{U}_r^T \mathbf{R}\right) \boldsymbol{x}, \ \mathbf{S}_r = \mathbf{I}_r - (\boldsymbol{\Lambda}_r + \mathbf{I}_r)^{-\frac{1}{2}}.$$

- Thanks to the $\mathbf{R}$-orthogonality of the eigenvectors matrix $\mathbf{U}_r$ (i.e. $\mathbf{U}_r^T \mathbf{R} \mathbf{U}_r = \mathbf{I}_r$), a straightforward calculation gives:

$$\begin{aligned}
\operatorname{cov}(\boldsymbol{y}) = \mathsf{E}\left[\boldsymbol{y}\boldsymbol{y}^T\right] &= \mathsf{E}\left[\left(\mathbf{I}_n - \mathbf{U}_r \mathbf{S}_r \mathbf{U}_r^T \mathbf{R}\right) \boldsymbol{x}\boldsymbol{x}^T \left(\mathbf{I}_n - \mathbf{U}_r \mathbf{S}_r \mathbf{U}_r^T \mathbf{R}\right)^T\right] \\
&= \left(\mathbf{I}_n - \mathbf{U}_r \mathbf{S}_r \mathbf{U}_r^T \mathbf{R}\right) \mathsf{E}\left[\boldsymbol{x}\boldsymbol{x}^T\right] \left(\mathbf{I}_n - \mathbf{U}_r \mathbf{S}_r \mathbf{U}_r^T \mathbf{R}\right)^T \\
&= \left(\mathbf{I}_n - \mathbf{U}_r \mathbf{S}_r \mathbf{U}_r^T \mathbf{R}\right) \mathbf{R}^{-1} \left(\mathbf{I}_n - \mathbf{U}_r \mathbf{S}_r \mathbf{U}_r^T \mathbf{R}\right)^T \\
&= \mathbf{R}^{-1} - \mathbf{U}_r (2\mathbf{S}_r - \mathbf{S}_r^2) \mathbf{U}_r^T \\
&= \mathbf{R}^{-1} - \mathbf{U}_r \mathbf{D}_r \mathbf{U}_r^T \approx \mathbf{H}^{-1}
\end{aligned}$$

# Samples from the prior and posterior

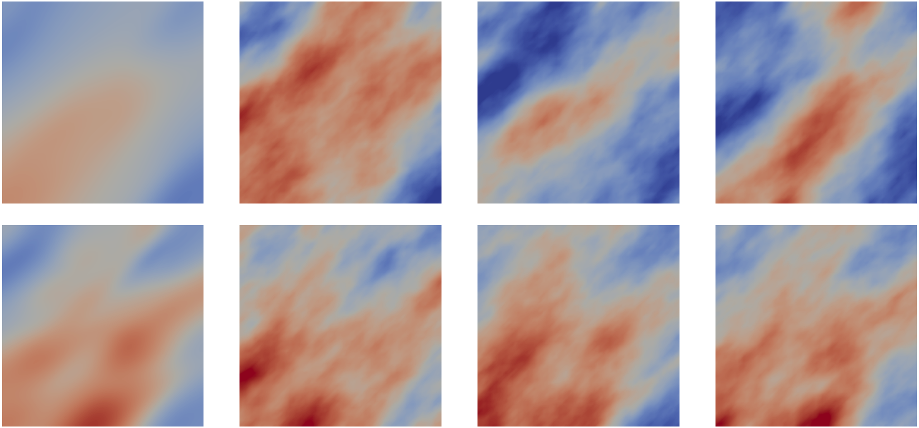


Figure: Top: Prior mean log permeability $m_{\text{pr}}$ (left), and samples drawn from the prior distribution. Bottom: The MAP point (left) and samples drawn from the Gaussianized posterior distribution.

# Pointwise variance of the prior

- For the prior we use an unbiased stochastic estimator to estimate the diagonal of the inverse of $\mathbf{R}$ of the form

$$\operatorname{diag}(\mathbf{R}^{-1}) \approx \left[\sum_{i=1}^{s} \boldsymbol{z}_j \odot \boldsymbol{w}_j\right] \oslash \left[\sum_{i=1}^{s} \boldsymbol{z}_j \odot \boldsymbol{z}_j\right],$$

  - where $\boldsymbol{w}_j$ solves $\mathbf{R}\boldsymbol{w}_j = \boldsymbol{z}_j$ and $\boldsymbol{z}_j$ are random i.i.d. vectors.
  - Here $\odot$ and $\oslash$ represent the element-wise multiplication and division operators of vectors, respectively.
  - The convergence of the method is independent of the size of the problem, but convergence is in general slow.

Details in: Costas Bekas, Effrosyni Kokiopoulou, and Yousef Saad. "An estimator for the diagonal of a matrix", Applied Numerical Mathematics, 57 (2007), pp. 1214-1229.

# Pointwise variance of the posterior

- For the posterior we use the low rank factorization of the Hessian misfit and the Woodbury formula to obtain the approximation:

$$\delta(\mathbf{H}^{-1}) \approx \delta(\mathbf{R}^{-1} - \mathbf{U}_r\mathbf{D}_r\mathbf{U}_r^T) = \delta(\mathbf{R}^{-1}) - \delta(\mathbf{U}_r\mathbf{D}_r\mathbf{U}_r^T).$$

- The first term is approximated by either using stochastic or probing methods, while the data-informed correction $\delta(\mathbf{U}_r\mathbf{D}_r\mathbf{U}_r^T)$ can be explicitly computed in $\mathcal{O}(n)$ operations as follows

$$\delta(\mathbf{U}_r\mathbf{D}_r\mathbf{U}_r^T) = \sum_{i=1}^{r} \left[ \left( \frac{\lambda_i}{1+\lambda_i} \boldsymbol{u}_i \right) \odot \boldsymbol{u}_i \right].$$

# Reduction in variance due to observations

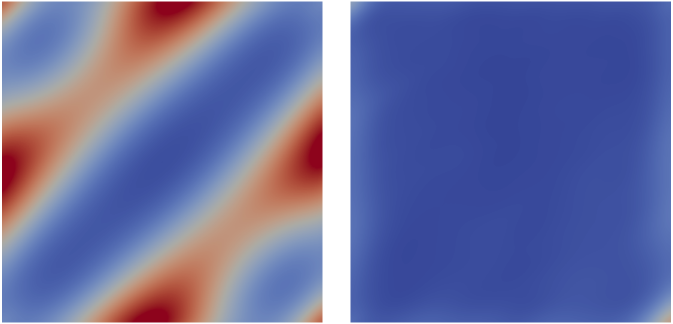


Figure: The pointwise variance of the prior distribution (left) and of (the Gaussian approximation of) the posterior distribution (right).

# Example: Inversion for the initial condition in an advection-diffusion PDE

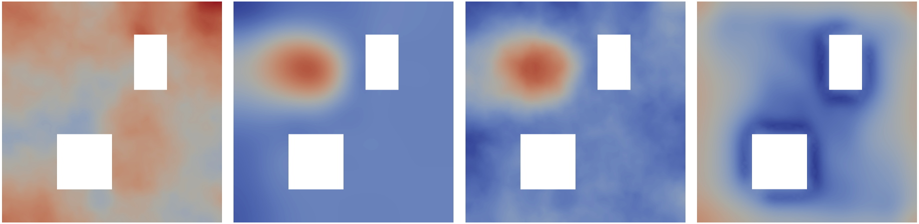
. . . show ipython notebook . . .

# Outline

1. Intro and roadmap
   - What you will hopefully learn today . . .
   - Inverse problems: overview and examples
   - Motivation and challenges
   - Inference of basal sliding parameter field in ice sheets
2. Deterministic Inversion: formulation, solution methods and examples
   - Newton-type descent algorithms with line search
   - Calculus of variations, weak forms and computing derivatives via adjoints
   - Example: Coefficient field inversion in an elliptic PDE
3. Bayesian approach to inverse problems
   - Why is a statistical perspective useful in inverse problems?
   - Bayesian inversion
   - Challenges for large-scale Bayesian inverse problems
   - Example: Inversion for the initial condition in an advection-diffusion PDE
4. `hIPPYlib`: **I**nverse **P**roblem **PY**thon **lib**rary
5. Conclusions, discussion and references

# hIPPYlib: Inverse Problem PYthon library

- Implements state-of-the-art scalable adjoint-based algorithms for PDE-based deterministic and Bayesian inverse problems.
- Builds on FEniCS for the discretization of the PDE and on PETSc for scalable and efficient linear algebra operations and solvers.
- Release: `hIPPYlib` 1.0 is now available at `http://hippylib.github.io`



# hIPPYlib: Inverse Problem PYthon library

- Features:
  - Friendly, compact, near-mathematical FEniCS notation to express the PDE and likelihood in weak form
  - Automatic generation of efficient code for the discretization of weak forms using FEniCS
  - Symbolic differentiation of weak forms to generate derivatives and adjoint information
  - Globalized Inexact Newton-CG method to solve the inverse problem
  - Randomized algorithm for the generalized eigenvalue problem
  - Low rank representation of the posterior covariance using randomized algorithms
  - Sampling from the prior and from the Gaussian approximation of the posterior
  - Extract pointwise variance of prior and posterior
- Release:
  - `hIPPYlib` 1.0 is now available at: `http://hippylib.github.io`

Details in: U. Villa, N. Petra, and O. Ghattas, *hIPPYlib: An Extensible Software Framework for Large-Scale Deterministic and Linearized Bayesian Inverse Problems*. To be submitted.

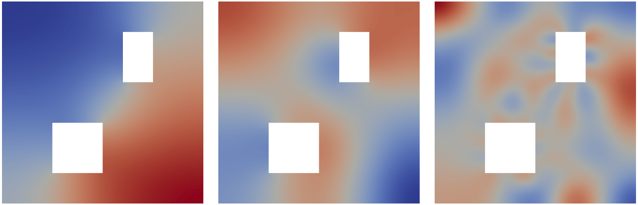
# Other existing libraries

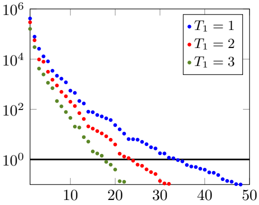
Several existing libraries address:

- finite element computations based on variational forms
  - COMSOL Multiphysics (COMSOL Inc.), deal.II (Texas A&M), dune, the FEniCS project (Simula Research Laboratory), and Sundance (Sandia and Texas Tech), a package from the Trilinos project
- PDE-constrained optimization
  - dolfin-adjoint: builds also on FEniCS and derives discrete adjoints from a forward model written in the Python interface to DOLFIN using a combination of symbolic and automatic differentiation
  - jInv: a flexible Julia package for PDE parameter estimation
- UQ and Bayesian inverse problems:
  - QUESO (UT Austin), DAKOTA (Sandia), PSUADE (LLNL), Uq toolkit (Sandia), MUQ (MIT)

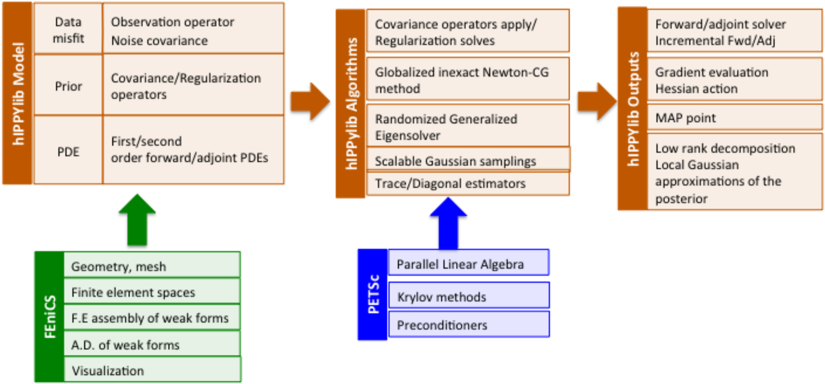
# Extensions in hIPPYlib:

- exploits the structure of the underlying infinite-dimensional PDE forward problem (and the corresponding adjoint and incremental problems)
- straightforward implementation of second order adjoints, needed by the Hessian apply (efficient operation such as the inverse and square-root inverse of the Hessian applied to a vector)
- incorporates modifications of state-of-the-art algorithms to ensure consistency with the infinite-dimensional settings
- allows exploring and testing various priors, observation operators, noise covariance models



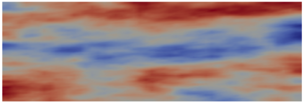
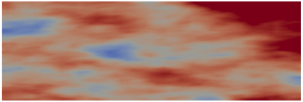
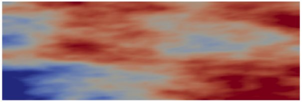


# Design and software components of hIPPYlib



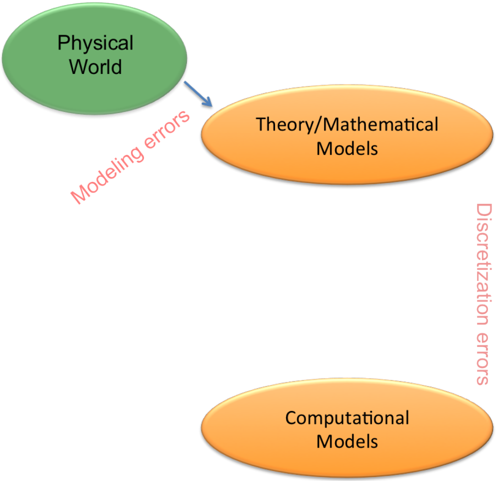
# hIPPYlib used at

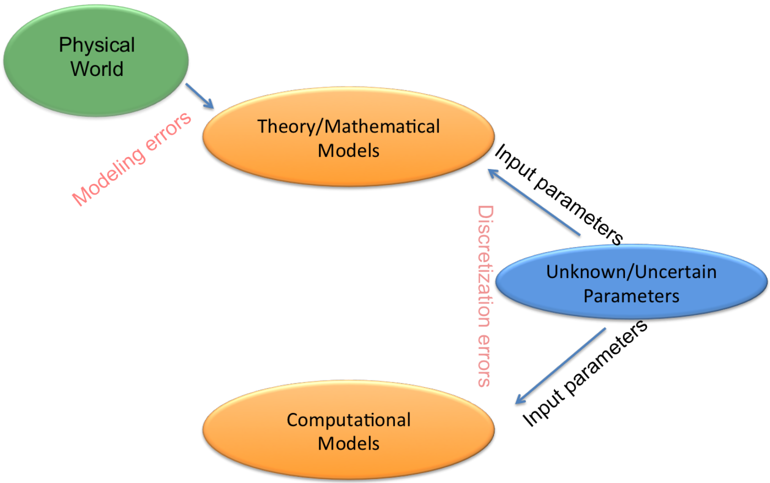
- Inverse problem classes at UT Austin, UC Merced and NYU/Courant
- Various workshops and summer schools (e.g., ICERM and SAMSI)
- Research on ice sheet modeling, RANS turbulence model, poroelasticity, melting problems, turbulent flows, Markov chain Monte Carlo Sampling, Bayesian Calibration of inadequate model

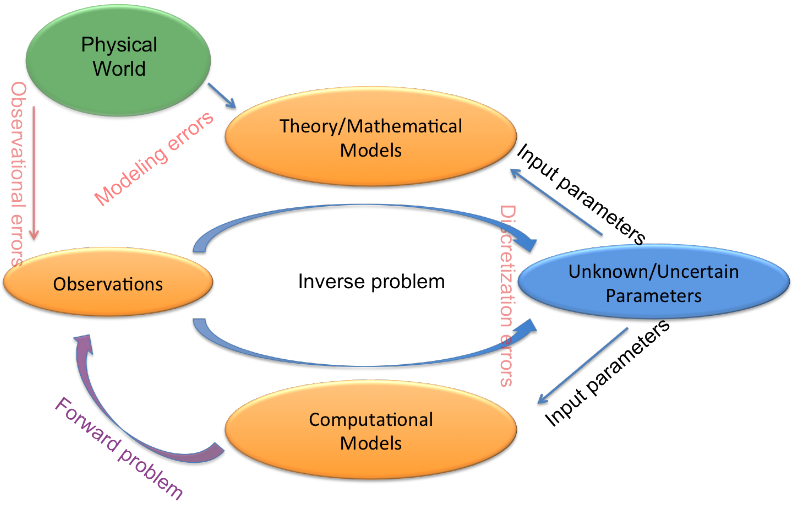




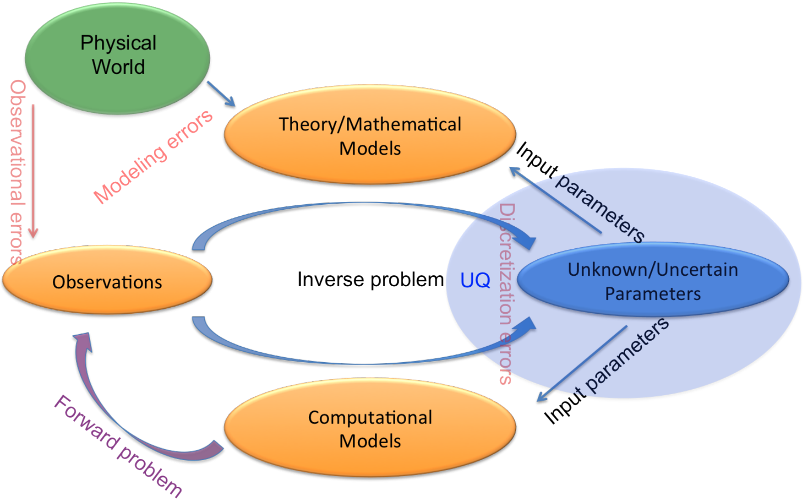
# Outline

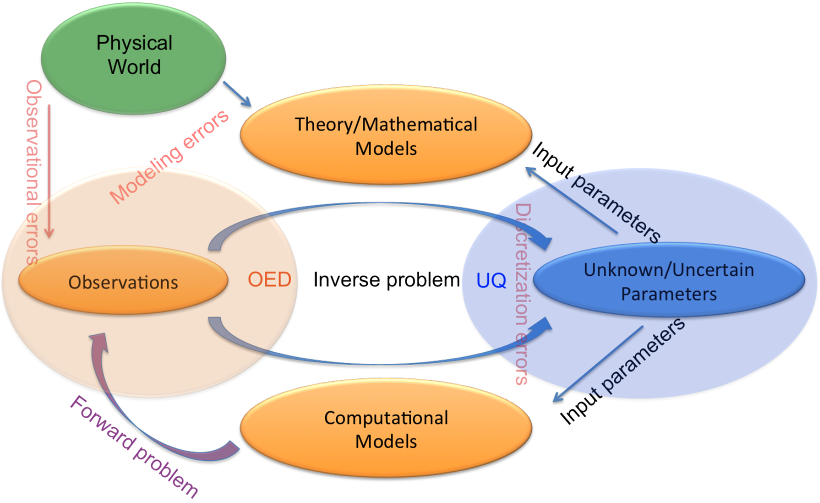
1. Intro and roadmap
   - What you will hopefully learn today . . .
   - Inverse problems: overview and examples
   - Motivation and challenges
   - Inference of basal sliding parameter field in ice sheets
2. Deterministic Inversion: formulation, solution methods and examples
   - Newton-type descent algorithms with line search
   - Calculus of variations, weak forms and computing derivatives via adjoints
   - Example: Coefficient field inversion in an elliptic PDE
3. Bayesian approach to inverse problems
   - Why is a statistical perspective useful in inverse problems?
   - Bayesian inversion
   - Challenges for large-scale Bayesian inverse problems
   - Example: Inversion for the initial condition in an advection-diffusion PDE
4. `hIPPYlib`: **I**nverse **P**roblem **PY**thon **lib**rary
5. Conclusions, discussion and references


Physical World
Modeling errors
Theory/Mathematical Models
Discretization errors
Computational Models

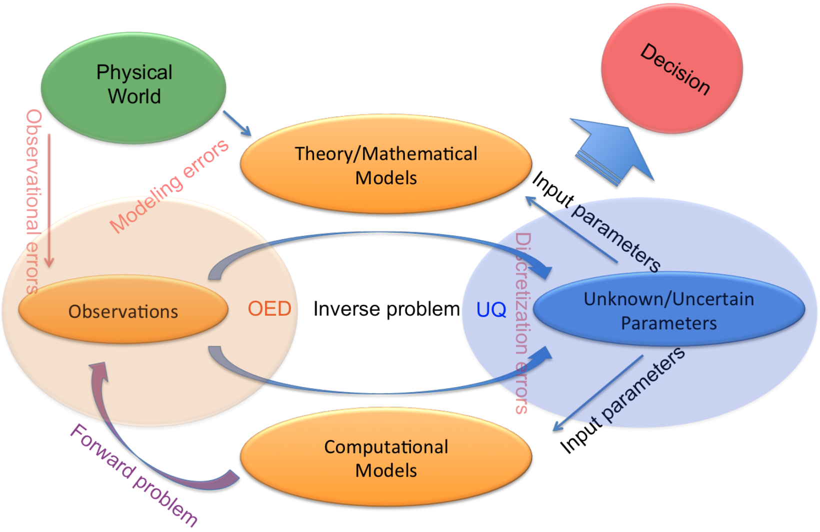

Physical World
Modeling errors
Theory/Mathematical Models
Input parameters
Discretization errors
Unknown/Uncertain Parameters
Input parameters
Computational Models


Physical World
Theory/Mathematical Models
Modeling errors
Observational errors
Observations
Inverse problem
Discretization errors
Unknown/Uncertain Parameters
Input parameters
Input parameters
Computational Models
Forward problem


Physical World
Theory/Mathematical Models
Modeling errors
Observational errors
Observations
Inverse problem
UQ
Discretization errors
Input parameters
Unknown/Uncertain Parameters
Input parameters
Computational Models
Forward problem



Physical World
Theory/Mathematical Models
Observational errors
Modeling errors
Input parameters
Discretization errors
Observations
OED
Inverse problem
UQ
Unknown/Uncertain Parameters
Computational Models
Input parameters
Forward problem



Physical World
Theory/Mathematical Models
Decision
Observational errors
Modeling errors
Input parameters
Discretization errors
Observations
OED
Inverse problem
UQ
Unknown/Uncertain Parameters
Input parameters
Computational Models
Forward problem