Probabilistic PDE Solvers

Jon Cockayne, ProbNum School 2024

Overview

1. Get Motivated
2. Generalising GP Regression
3. Probabilistic PDE Solvers
4. A Tiny Bit of Theory
5. Inverse Problems

1: Motivation

Partial Differential Equations

- PDEs are everywhere…
- …but they are very hard to solve.

What is a PDE?

The Heat Equation

Unknown function

$$u : [0,1] \times [0,T] \to \mathbb{R}$$

$$u = u(x, t)$$

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \qquad x \in (0,1),\ t \in (0,T]$$

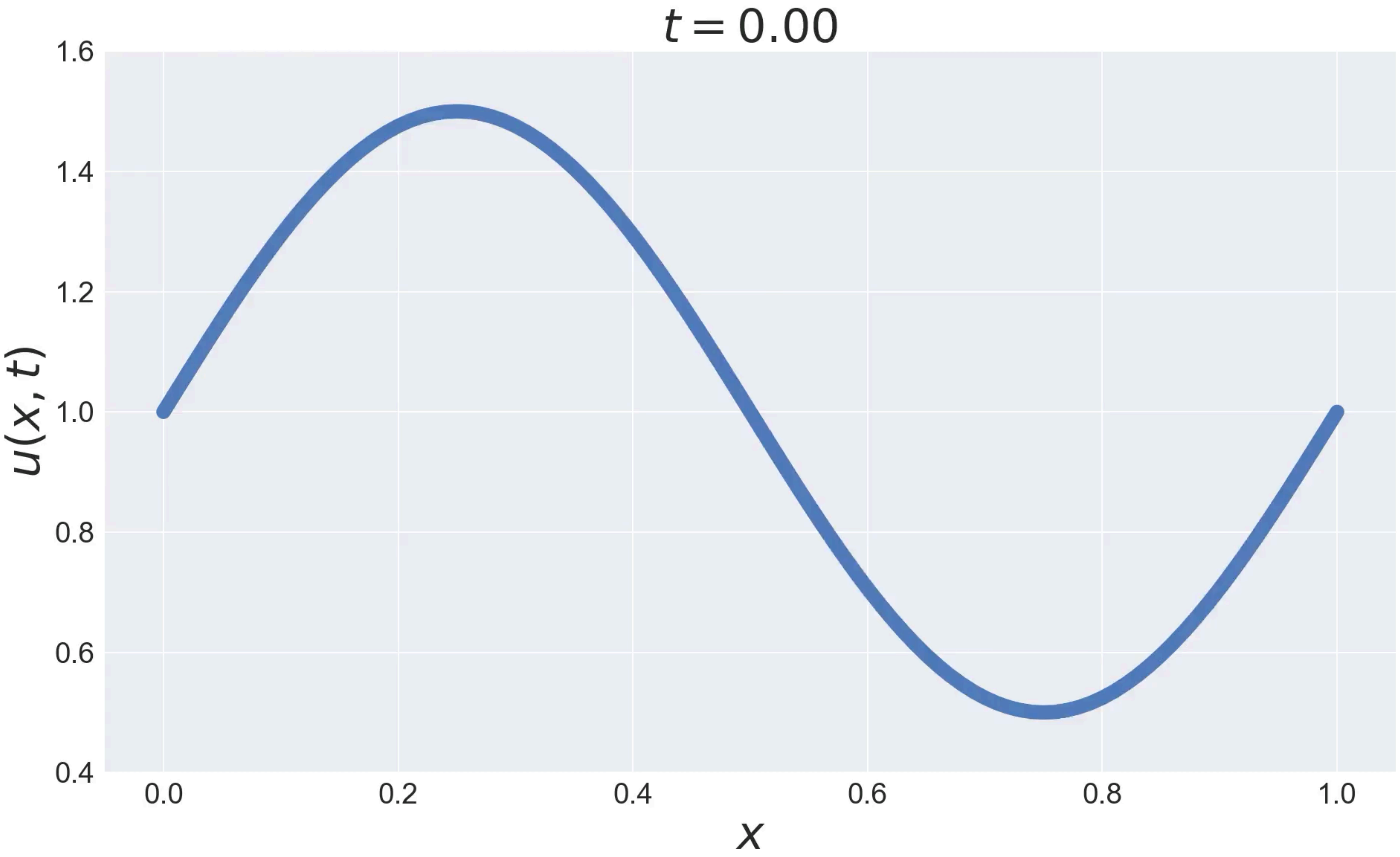
Initial Condition

$$u(x,0) = u_0(x)$$

$$u(0) = a$$

$$u(1) = b$$

(Dirichlet)
Boundary Condition


$t = 0.00$
1.6
1.4
1.2
1.0
0.8
0.6
0.4
$u(x, t)$
0.0
0.2
0.4
0.6
0.8
1.0
x

1.1: PDEs are Everywhere in Scientific Computing

Example 1: Engineering

- Elasticity equations
- Mechanical behaviour of buildings and structures.

$$\nabla \cdot \sigma + F = \rho \ddot{u}$$

$$\epsilon = \frac{1}{2}[\nabla u + \nabla u^{\top}]$$

$$\sigma = C : \epsilon$$

Displacement u

Strain ϵ

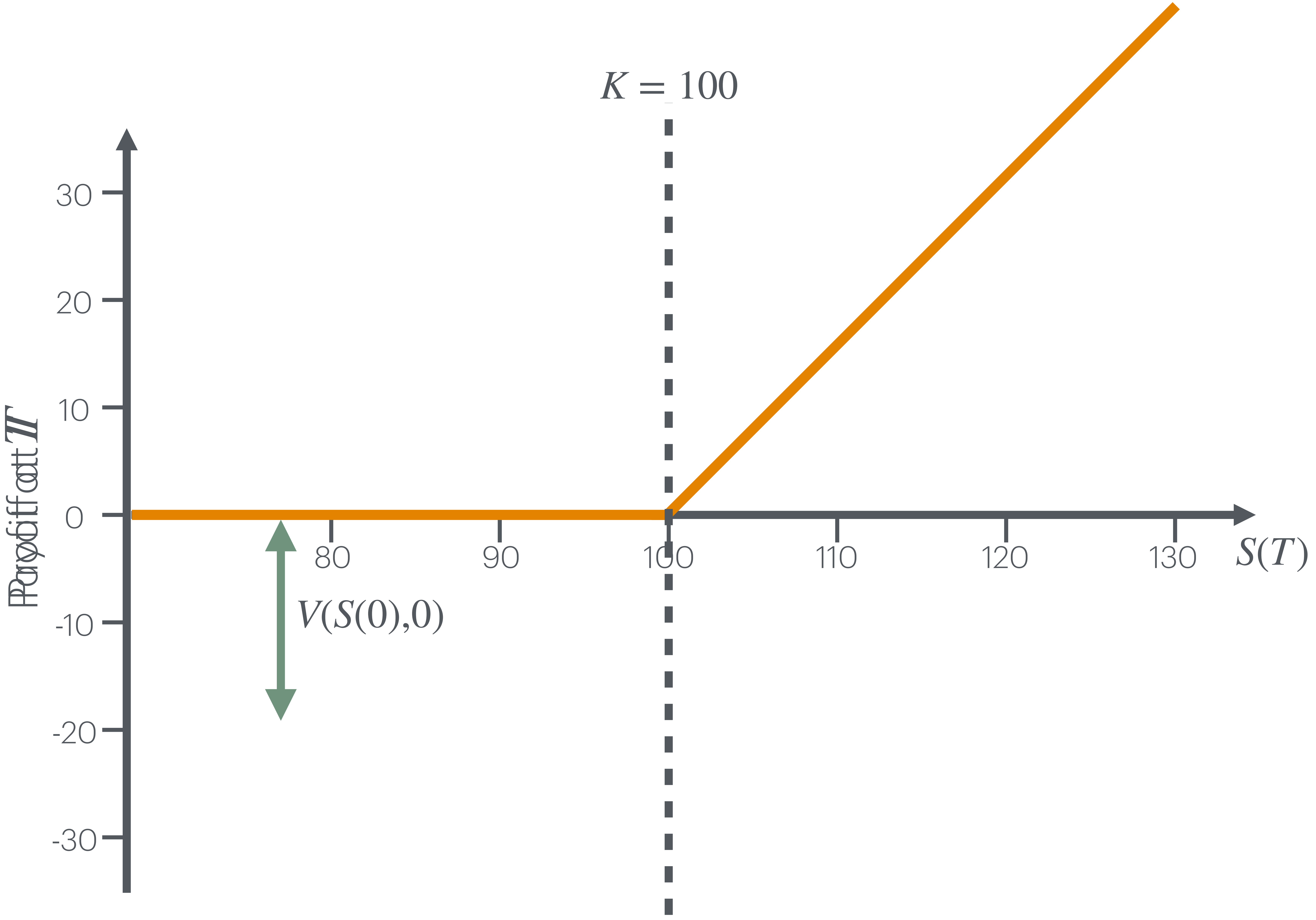
Stress σ

Stiffness C

Force/volume F

Example 2: Finance

- Call option on underlying asset S
  - Expires at T, “Strike Price” K
  - Pays off $\max(S(T) - K, 0)$.
- Denote the “Value” of the call as $V(S, t)$ for any $0 \leq t \leq T$.
- What is $V(S(0), 0)$?



$K = 100$
30
20
10
0
-10
-20
-30
80
90
100
110
120
130
$S(T)$
$V(S(0),0)$

Example 2: Finance

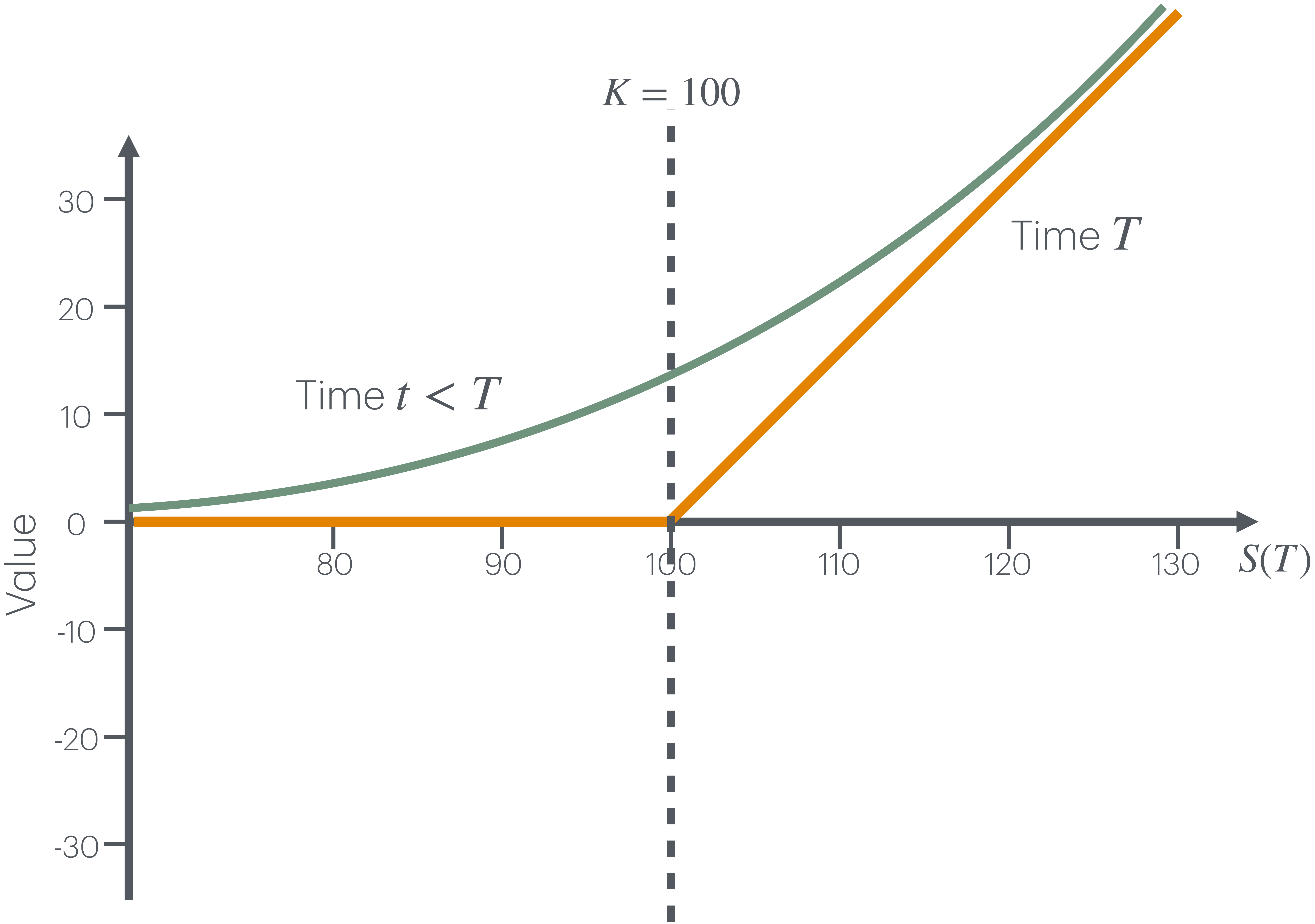
- What is the (expected) value of this asset $V(S, t)$ for $t \in [0,T)$?
- Assume the price of a stock follows geometric Brownian motion:

$$dS = \mu S\, dt + \sigma S\, dW_t$$

- Black-Scholes Formula:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0$$

$$V(T, S) = V_T(S)$$



$K = 100$
30
20
10
0
-10
-20
-30
Value
80
90
100
110
120
130
$S(T)$
Time $t < T$
Time T

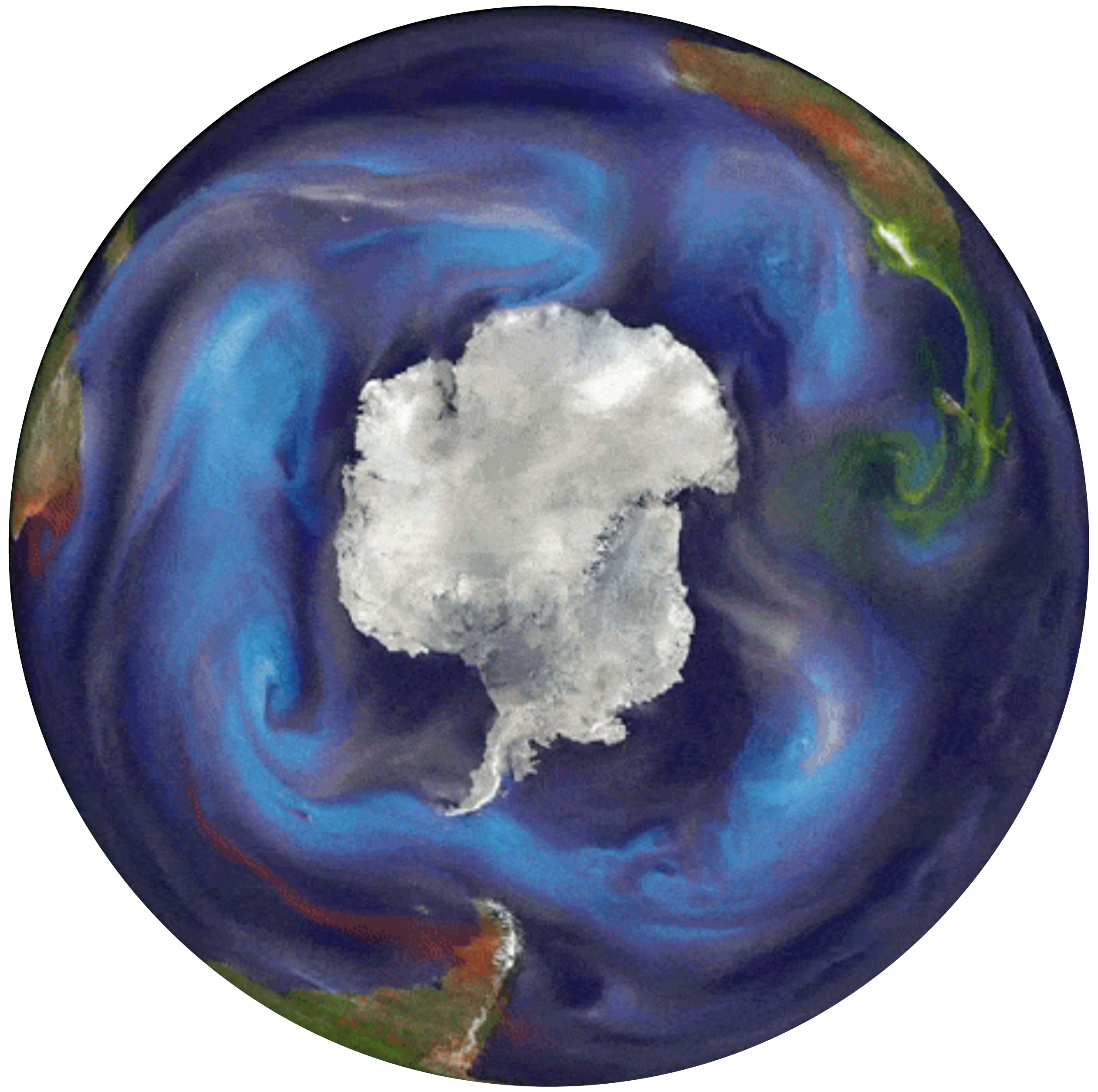
Example 3: Weather and Climate Modelling

- Ocean + Air modelled as coupled fluids using Navier-Stokes:

$$\rho\left(\frac{\partial v}{\partial t} + v \cdot \nabla v\right) = -\nabla p + \nabla \cdot T + f$$

- Numerical implementation is “Computational Fluid Dynamics”.

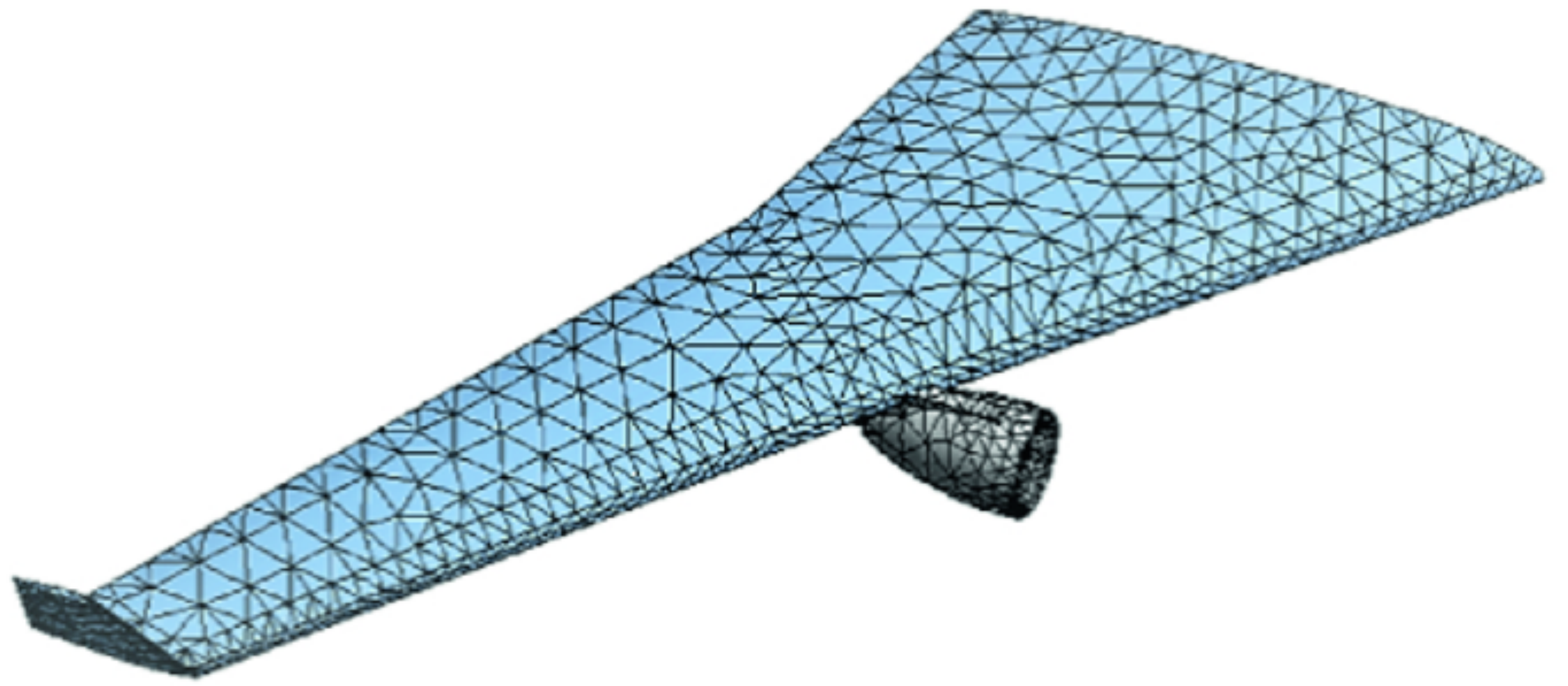
- Simulate forward to predict weather and climate.



1.2: PDEs are Hard to Solve

How are PDEs Usually Solved?

- Represent the domain with a grid or mesh.
- Represent a solution on the mesh.
- Approximate the PDE.
- Solve the resulting (discrete) equations
- This incurs "discretisation error" $\|u - u_N\|$

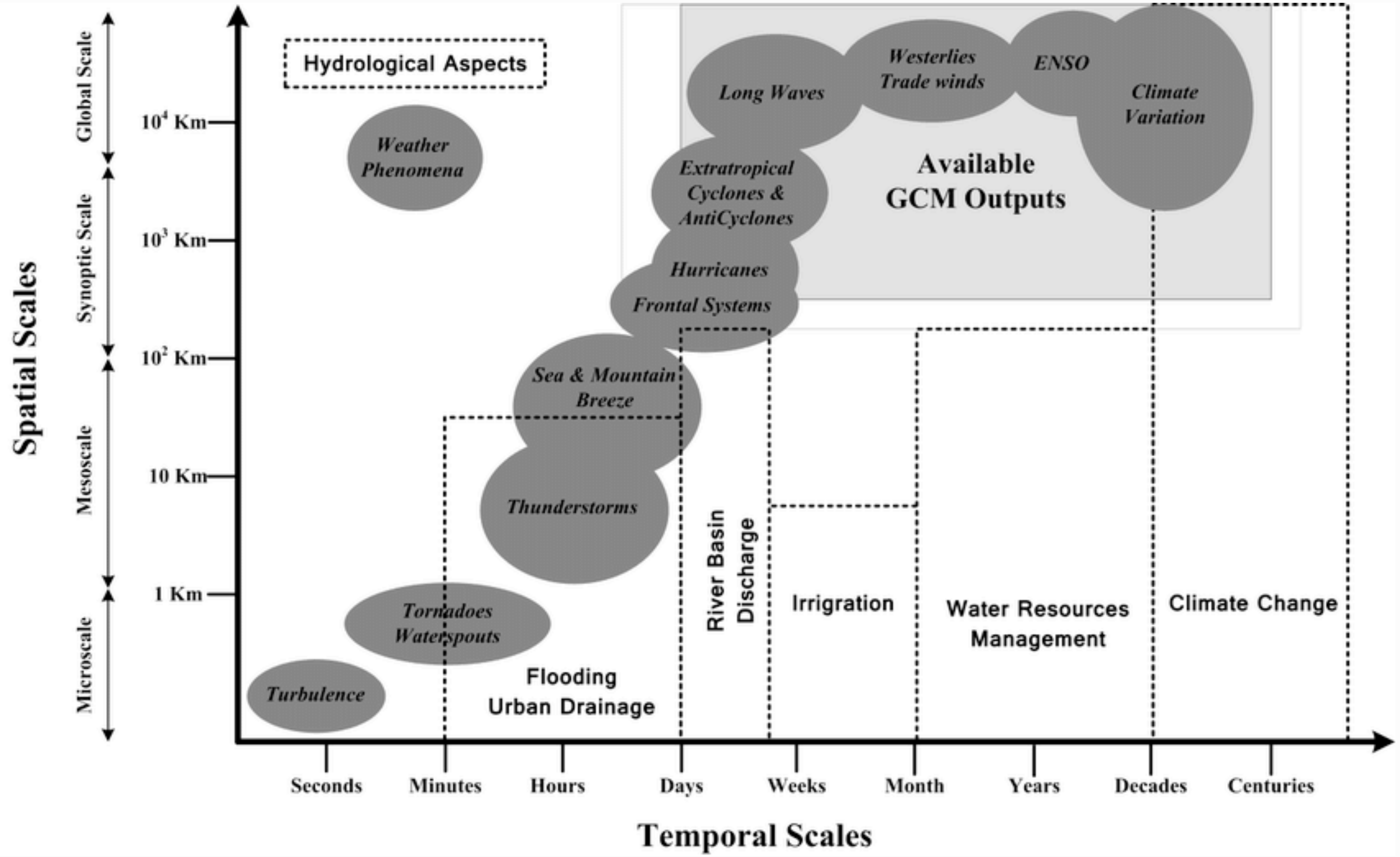


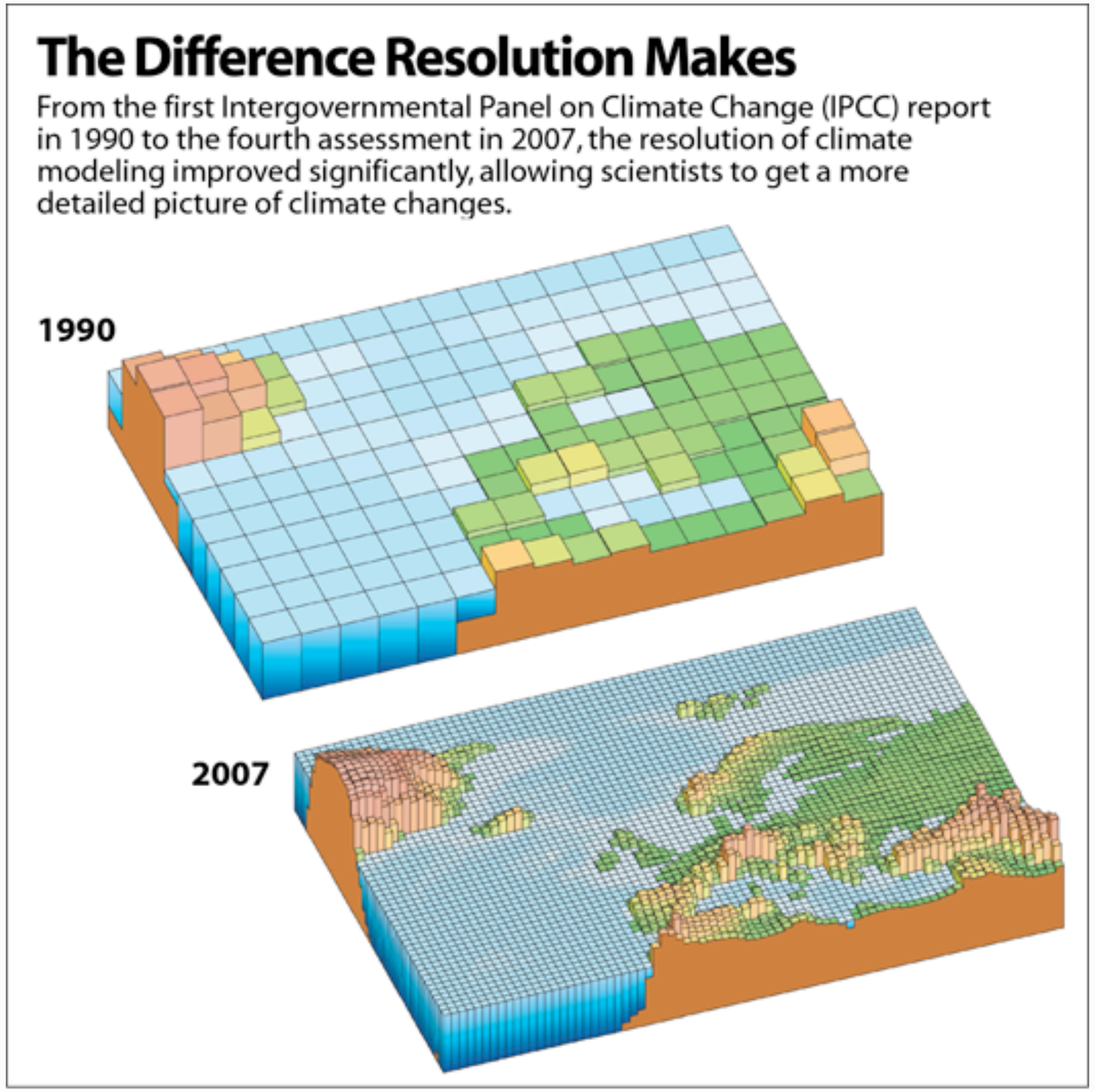
$$u \approx u_N := \sum_{i=1}^{N} w_i \phi_i(x)$$

Linear System / Objective Function

Calculate "best" w_i

Discretisation Error





SOURCE: University Corporation for Atmospheric Research (UCAR) InsideClimate News

This is what PN is for!

(But there is a huge gap)

2. Generalising GP Regression

Recap: GP Regression

- We suppose we have a GP Prior:

$$u \sim \mathcal{GP}(m, k)$$

- Condition on observations $u(x_i) = u_i$. Let $\mathcal{D} := \{(x_i, u_i)\}_{i=1}^n$:

$$\begin{aligned} u \mid \mathcal{D} &\sim \mathcal{GP}(\bar{m}, \bar{k}) \\ \bar{m}(x) &= m(x) + k(x, X)K(X, X)^{-1}(\mathbf{u} - m(X)) \\ \bar{k}(x, x') &= k(x, x') - k(x, X)K(X, X)^{-1}k(X, x') \end{aligned}$$

Under the Hood

- We can construct the conditional distribution because of **joint Gaussianity**:

$$\begin{bmatrix} u(X) \\ u(X') \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(X) \\ m(X') \end{bmatrix}, \begin{pmatrix} k(X,X) & k(X,X') \\ k(X',X) & k(X',X') \end{pmatrix} \right)$$

- This is **multivariate** Gaussian, so we can use the multivariate Gaussian conditioning formula:

$$\begin{bmatrix} U \\ Y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m_U \\ m_Y \end{bmatrix}, \begin{pmatrix} \Sigma_{UU} & \Sigma_{UY} \\ \Sigma_{UY}^\top & \Sigma_{YY} \end{pmatrix} \right) \implies \begin{aligned} &U \mid Y = y \sim \mathcal{N}(\bar{m}, \bar{\Sigma}) \\ &\quad \bar{m} = m_U + \Sigma_{UY}\Sigma_{YY}^{-1}(y - \mu_Y) \\ &\quad \bar{\Sigma} = \Sigma_{UU} - \Sigma_{UY}\Sigma_{YY}^{-1}\Sigma_{UY}^\top \end{aligned}$$

Key Observation:
We can do this anywhere we get joint Gaussianity.

Generalising Observations

- Encapsulate information provided in an "information operator" $\mathscr{A} : \mathscr{U} \to \mathbb{R}^n$
- If $\mathscr{A}$ is a (suitable*) linear operator we have:

$$\begin{bmatrix} u(X) \\ \mathscr{A}u \end{bmatrix} \sim \mathscr{N}\left(\begin{bmatrix} m(X) \\ \mathscr{A}m \end{bmatrix}, \begin{pmatrix} k(X, X) & k(X, \cdot\,)\mathscr{A}^\dagger \\ \mathscr{A}k(\,\cdot\,, X) & \mathscr{A}k\mathscr{A}^\dagger \end{pmatrix} \right)$$

* Matsumoto, T., & Sullivan, T. J. (2023). Images of Gaussian and other stochastic processes under closed, densely-defined, unbounded linear operators (Version 5). arXiv. https://doi.org/10.48550/ARXIV.2305.03594

The Adjoint in the Room

- The previous slide had terms like $k(X, \cdot\,)\mathscr{A}^\dagger$ and $\mathscr{A}k\mathscr{A}^\dagger$
- Technically $\mathscr{A}^\dagger$ is the adjoint of $\mathscr{A}$.
- We don't need to worry about that - it just "operates on the second argument".
- E.g....
  - Considering $k(x, x')$...
  - $\mathscr{A}u = \frac{du}{dx}(0.5) \implies k(X, \cdot\,)\mathscr{A}^\dagger = \frac{dk}{dx'}(X,0.5)$

Linking to GP Regression

- E.g. in GP regression:

$$\mathscr{A}u = \begin{bmatrix} \delta_{x_1} u \\ \vdots \\ \delta_{x_n} u \end{bmatrix} = \begin{bmatrix} u(x_1) \\ \vdots \\ u(x_n) \end{bmatrix}$$

- As a result, $\mathscr{A}k\mathscr{A}^\dagger = k(X, X)$ as expected.

General Conditional Distribution

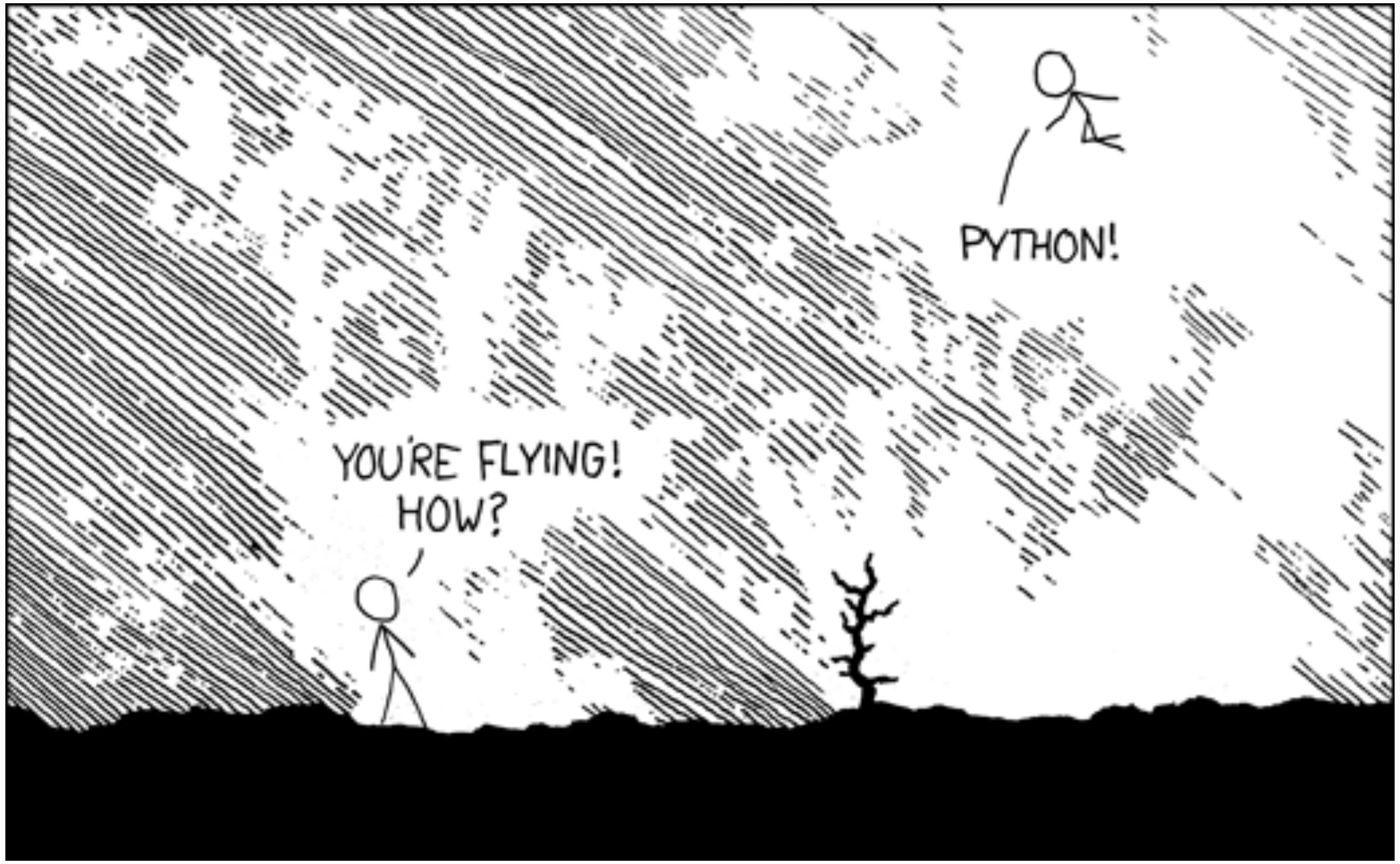
$$u(X) \mid \mathcal{A}u = y \sim \mathcal{N}(\bar{m}(X), \bar{k}(X, X))$$

$$\bar{m}(x) = m(x) + k(x, \cdot\,)\mathcal{A}^{\dagger}[\mathcal{A}k\mathcal{A}^{\dagger}]^{-1}(y - \mathcal{A}m)$$

$$\bar{k}(x, x') = k(x, x') + k(x, \cdot\,)\mathcal{A}^{\dagger}[\mathcal{A}k\mathcal{A}^{\dagger}]^{-1}\mathcal{A}k(\,\cdot\,, x')$$

Pförtner, M., Steinwart, I., Hennig, P., & Wenger, J. (2022). Physics-Informed Gaussian Process Regression Generalizes Linear PDE Solvers (Version 5). arXiv. https://doi.org/10.48550/ARXIV.2212.12474

Illustration: Conditioning on Derivatives



3: Probabilistic PDE Solvers

We just need to adapt $\mathscr{A}$ to the problem at hand.

Let's Solve a PDE

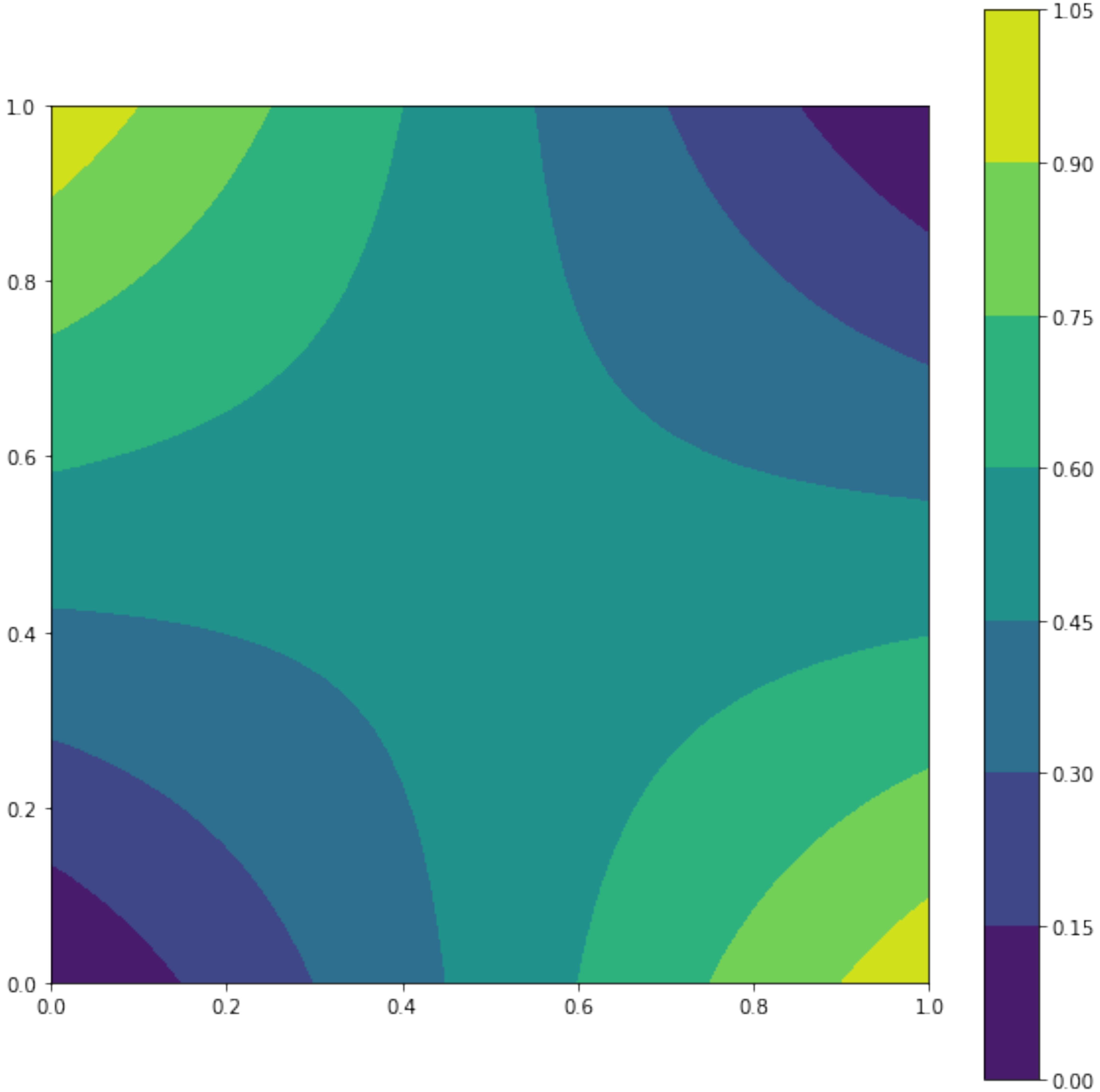
- Consider the canonical linear elliptic PDE with Dirichlet Boundary Conditions:

Given

$$-\nabla \cdot \left(\kappa(x)\nabla u(x)\right) = f(x) \qquad x \in D$$

$$u(x) = b(x) \qquad x \in \partial D$$

Unknown


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
1.05
0.90
0.75
0.60
0.45
0.30
0.15
0.00

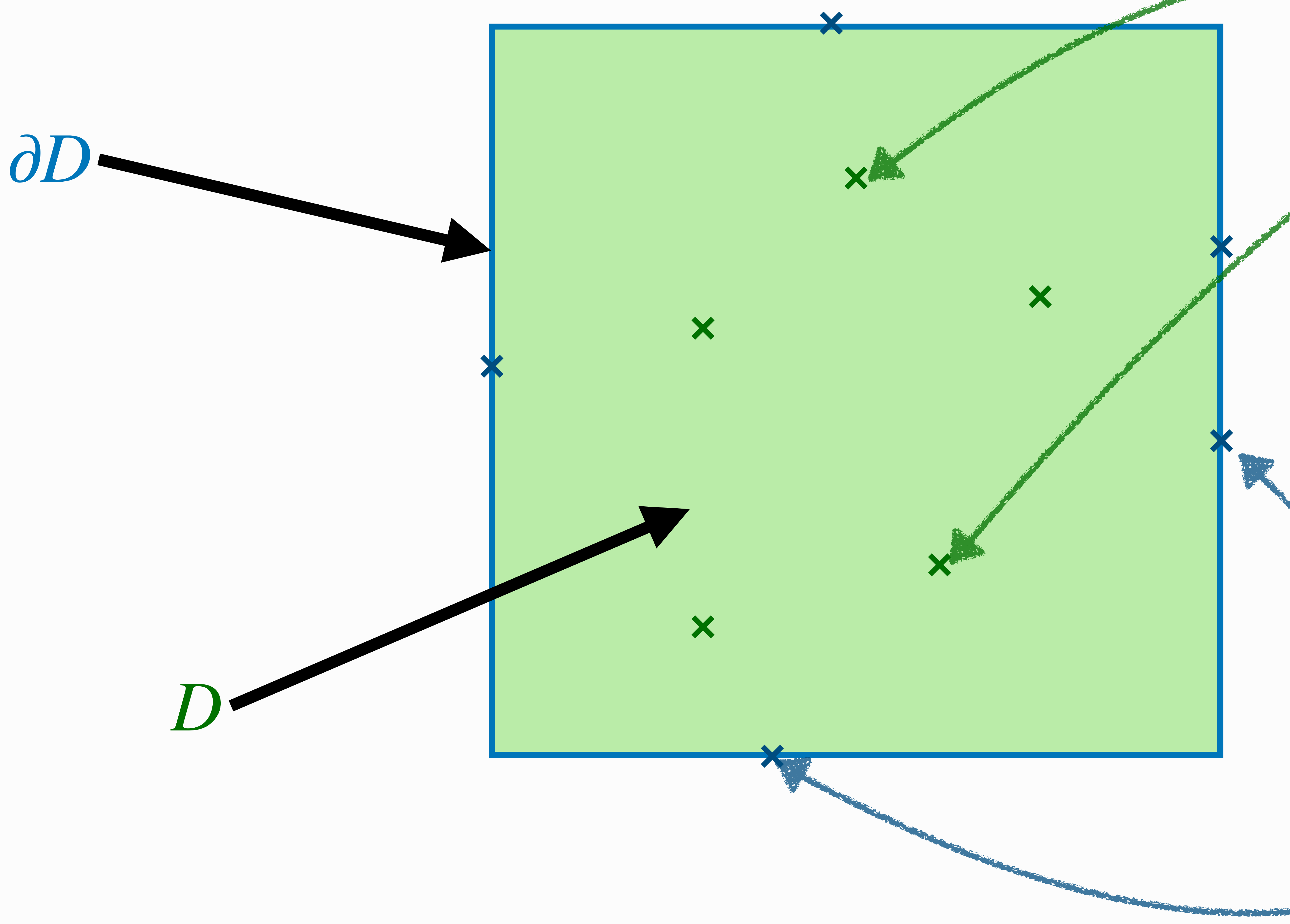
Let's Solve a PDE

- Consider the canonical linear elliptic PDE with Dirichlet Boundary Conditions:

$$
\begin{aligned}
\mathscr{L}_1 u &= f \\
\mathscr{L}_2 u &= b
\end{aligned}
\qquad
\begin{pmatrix}
x \in D & \mathscr{L}_1 u = -\nabla \cdot \big(\kappa(x)\nabla u(x)\big) \\
x \in \partial D & \mathscr{L}_2 u = u
\end{pmatrix}
$$

Let's Solve a PDE

A Natural Information Operator



$$\mathscr{D} = \left\{ \left(x_1^1, f(x_1^1)\right), \ldots, \left(x_{N_1}^1, f(x_{N_1}^1)\right) \right\}$$

$$\bigcup$$

$$\left\{ \left(x_1^2, b(x_1^2)\right), \ldots, \left(x_{N_2}^2, b(x_{N_2}^2)\right) \right\}$$

Information Operator

$$\mathscr{L}_1 u = f \qquad x \in D$$
$$\mathscr{L}_2 u = b \qquad x \in \partial D$$

$$\mathscr{A} := \begin{bmatrix} \delta_{x_1^1} \circ \mathscr{L}_1 \\ \vdots \\ \delta_{x_{N_1}^1} \circ \mathscr{L}_1 \\ \delta_{x_1^2} \circ \mathscr{L}_2 \\ \vdots \\ \delta_{x_{N_2}^2} \circ \mathscr{L}_2 \end{bmatrix} = \begin{bmatrix} \mathscr{A}_1 \\ \\ \mathscr{A}_2 \end{bmatrix}$$

$$u(X) \mid \mathscr{A}u = y \sim \mathcal{N}(\bar{m}(X), \bar{k}(X, X))$$

$$\bar{m}(x) = m(x) + k(x, \cdot\,)\mathscr{A}^\dagger[\mathscr{A}k\mathscr{A}^\dagger]^{-1}(y - \mathscr{A}m)$$

$$\bar{k}(x, x') = k(x, x') + k(x, \cdot\,)\mathscr{A}^\dagger[\mathscr{A}k\mathscr{A}^\dagger]^{-1}\mathscr{A}k(\,\cdot\,, x')$$

$$\begin{bmatrix} u(X) \\ \mathscr{A}u \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(X) \\ \mathscr{A}m \end{bmatrix}, \begin{pmatrix} k(X, X) & k(X, \cdot\,)\mathscr{A}^\dagger \\ \mathscr{A}k(\,\cdot\,, X) & \mathscr{A}k\mathscr{A}^\dagger \end{pmatrix}\right)$$

$$\begin{bmatrix} u(X) \\ \mathscr{A}_1u \\ \mathscr{A}_2u \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(X) \\ \mathscr{A}_1m \\ \mathscr{A}_2m \end{bmatrix}, \begin{pmatrix} k(X, X) & k(X, \cdot\,)\mathscr{A}_1^\dagger & k(X, \cdot\,)\mathscr{A}_2^\dagger \\ \mathscr{A}_1k(\,\cdot\,, X) & \mathscr{A}_1k\mathscr{A}_1^\dagger & \mathscr{A}_1k\mathscr{A}_2^\dagger \\ \mathscr{A}_2k(\,\cdot\,, X) & \mathscr{A}_1k\mathscr{A}_2^\dagger & \mathscr{A}_2k\mathscr{A}_2^\dagger \end{pmatrix}\right)$$

Probabilistic PDE Solver

- We then have the following:

$$
\begin{aligned}
u(X) \mid \mathscr{A}u = \mathbf{y} \sim \mathcal{N}(\bar{m}(X), \bar{k}(X, X)) \\
\bar{m}(X) = m(X) + k(X, \cdot\,)\mathscr{A}^{\dagger}[\mathscr{A}k\mathscr{A}^{\dagger}]^{-1}(\mathbf{y} - \mathscr{A}m) \\
\bar{k}(X, X) = k(X, X) - k(X, \cdot\,)\mathscr{A}^{\dagger}[\mathscr{A}k\mathscr{A}^{\dagger}]^{-1}\mathscr{A}k(\,\cdot\,, X)
\end{aligned}
$$

- The above is our probabilistic PDE solver…!

Illustration: Probabilistic PDE Solver

- See also https://github.com/marvinpfoertner/linpde-gp.

4: A Tiny Bit of Theory

What about UQ?

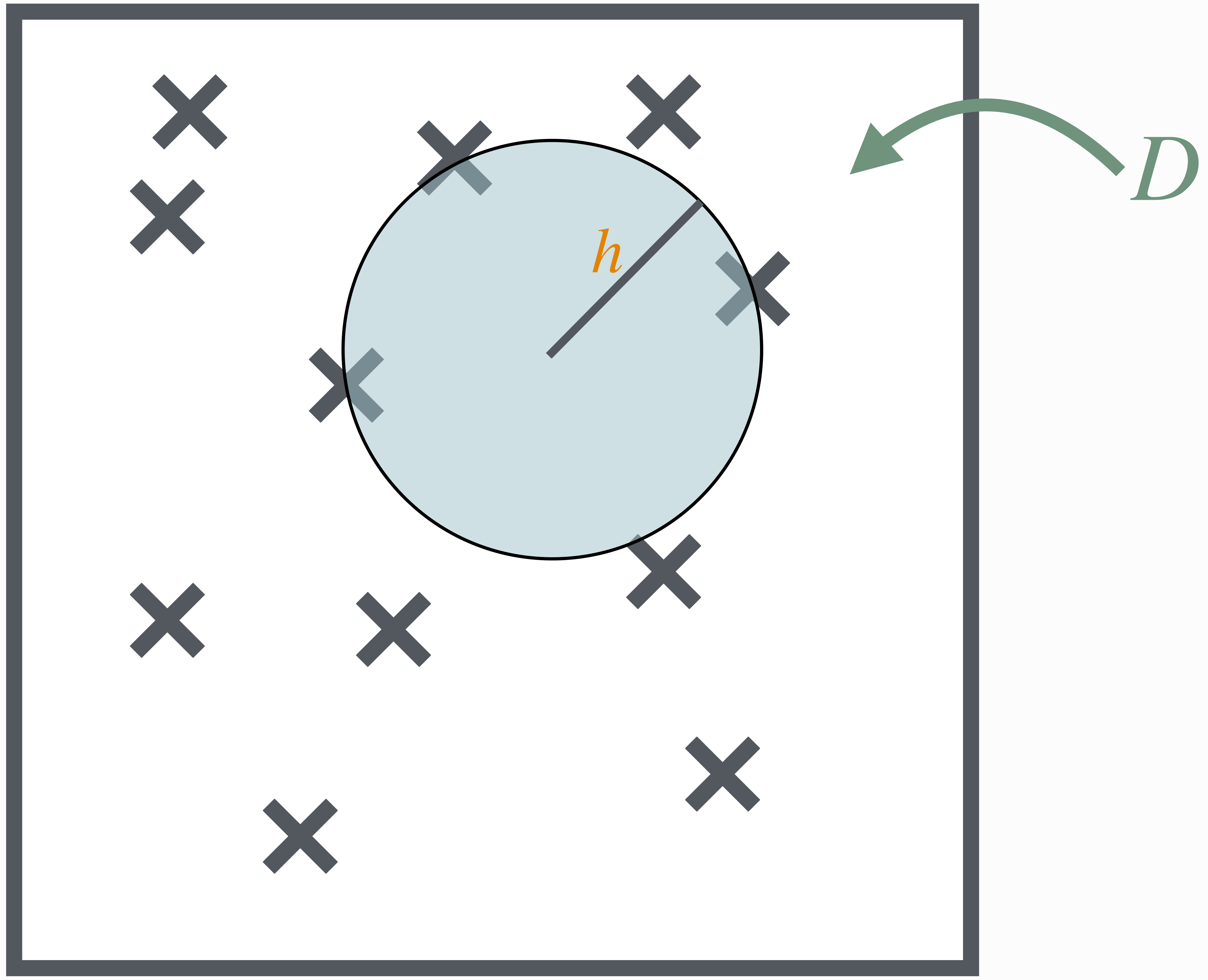
Suppose that (in addition to some technical assumptions):

- $u \in \mathbb{H}^{\beta}(D)$ for some $\beta > d/2$
- The RKHS $H_k(D)$ is equivalent to $\mathbb{H}^{\beta}(D)$

Then we have the error bound:

$$|u(x) - \bar{m}(x)| \leq \bar{k}(x,x)^{\frac{1}{2}} \|u - m\|_{H_k(D)}$$

Fill Distance



What About Convergence?

Consider the fill distance

$$h = \sup_{x \in D} \min_{x' \in \mathscr{D}} \|x - x'\|_2$$

Let $\rho < \beta - d/2$ denote the differential order of the PDE. Then it holds that

$$\bar{k}(x, x)^{\frac{1}{2}} \leq Ch^{\beta - \rho - d/2}$$

Generalising

- We focussed on point evaluation. Things can be made (much) more general.
- The first result holds much more generally.
- Fill-distance-based bounds are much trickier to derive for non-point-evaluation based $\mathscr{A}$.

Pförtner, M., Steinwart, I., Hennig, P., & Wenger, J. (2022). Physics-Informed Gaussian Process Regression Generalizes Linear PDE Solvers (Version 5). arXiv. https://doi.org/10.48550/ARXIV.2212.12474

5: Inverse Problems

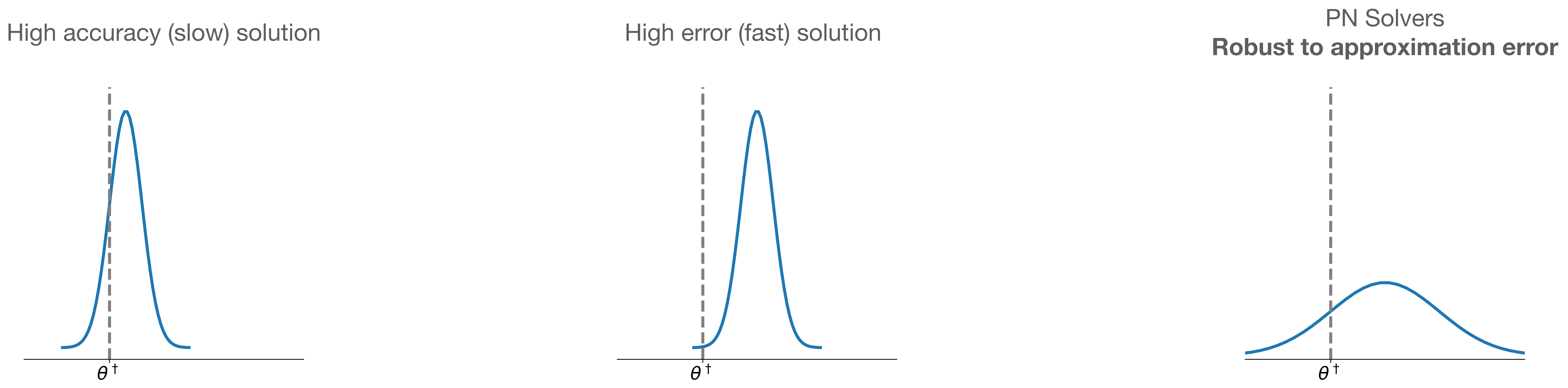
Bayesian Inverse Problems

- Suppose we have data:

$$y = \mathcal{G}(\theta) + \zeta$$

- Calculate / approximate the posterior distribution:

$$p(\theta \mid y) = \frac{p(y \mid \mathcal{G}(\theta))p(\theta)}{p(y)}$$

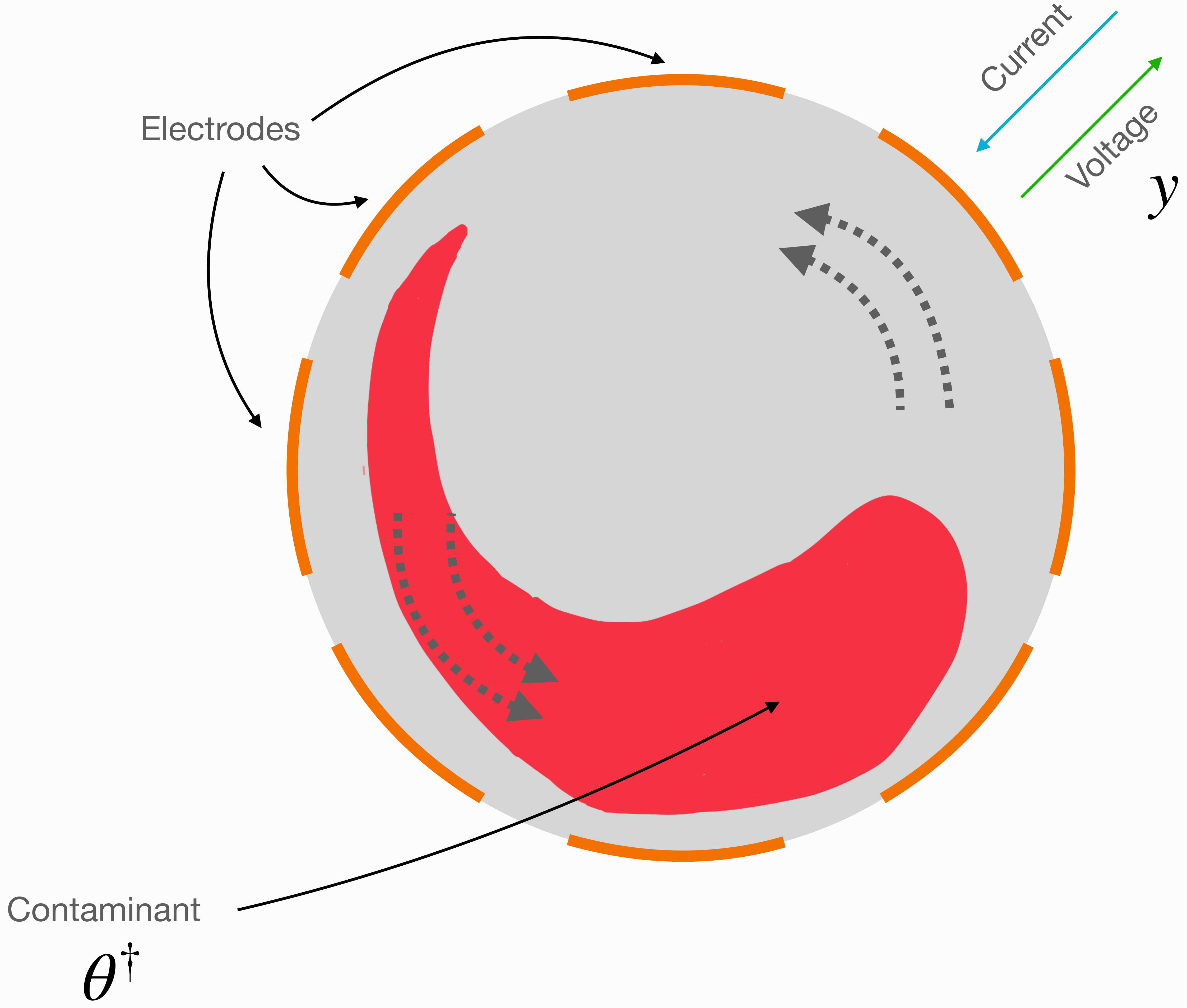

High accuracy (slow) solution
High error (fast) solution
PN Solvers
Robust to approximation error
$\theta^\dagger$
$\theta^\dagger$
$\theta^\dagger$

Hydrocyclones

Clean fluid

Contaminants

Electrical Impedance Tomography



"Data-generating model"

$$y = \mathscr{G}(\theta^\dagger) + \zeta$$

A (more complex) linear elliptic PDE

Putting PNM into Inference Problems

Latent solution to PDE in $\mathscr{G}(\theta)$

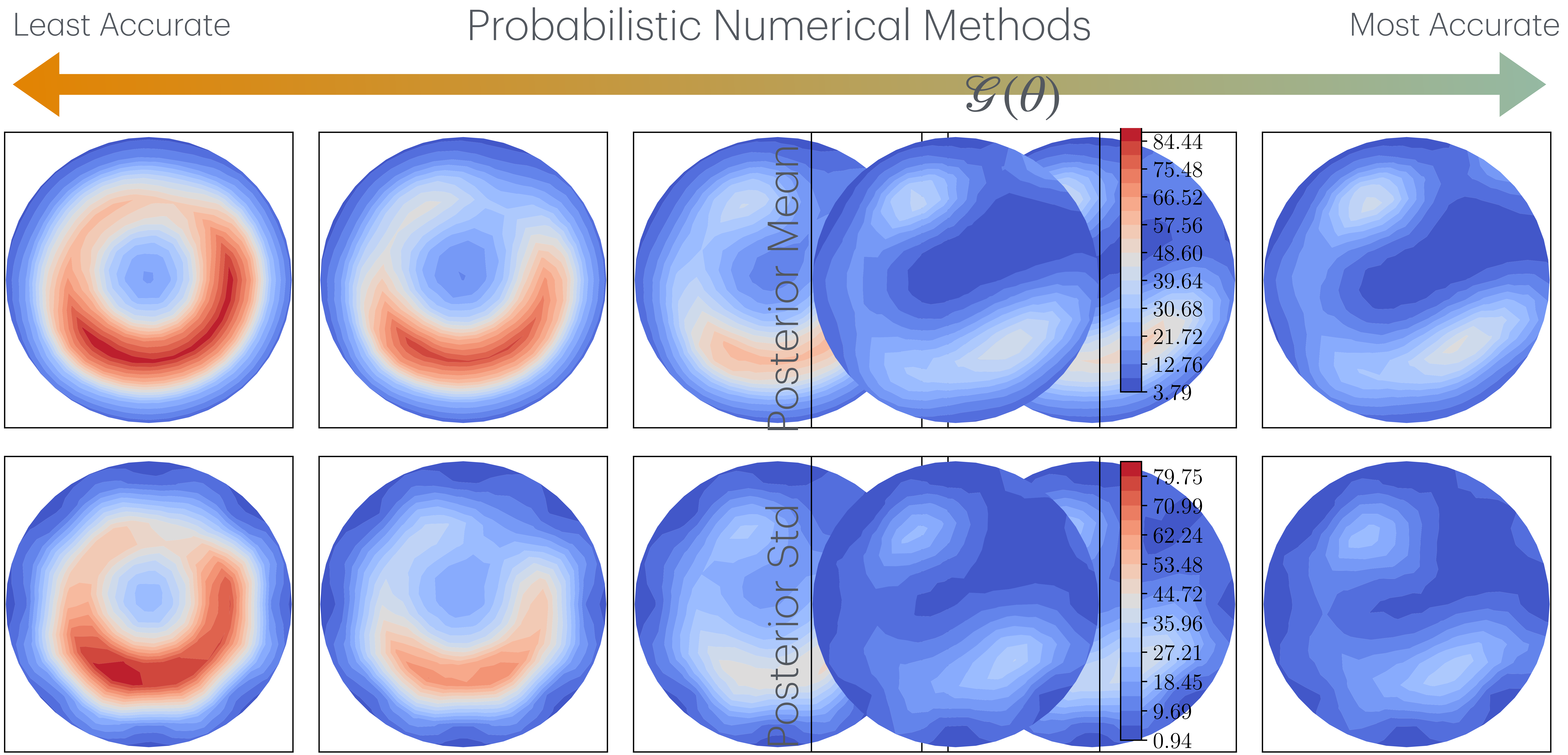
$$p(y \mid \theta, u_\theta^\dagger)$$

PN solution to PDE

$$p_{\mathrm{PN}}(y \mid \theta) = \int p(y \mid \theta, u)\, p(u \mid \mathscr{D}, \theta)\, \mathrm{d}u$$

"Inflate" likelihood by error in PNM.

In some cases can be done explicitly.



Bayesian Probabilistic Numerical Methods in Time-Dependent State Estimation for Industrial Hydrocyclone Equipment Oates, Cockayne, Aykroyd & Girolami, Journal of the American Statistical Association (2019)

Conclusions

A Blueprint for Bayesian PNM?

- "We just need to specify $\mathscr{A}$."
- For nonlinear $\mathscr{A}$ this is (ridiculously) harder…!

Thanks!