Introduction to Scientific ML : Day 1

By Juan Felipe Osorio Ramirez

MindLab
Department of Computer Systems and Industrial Engineering, UN
Summer 2023

UNIVERSITY of WASHINGTON

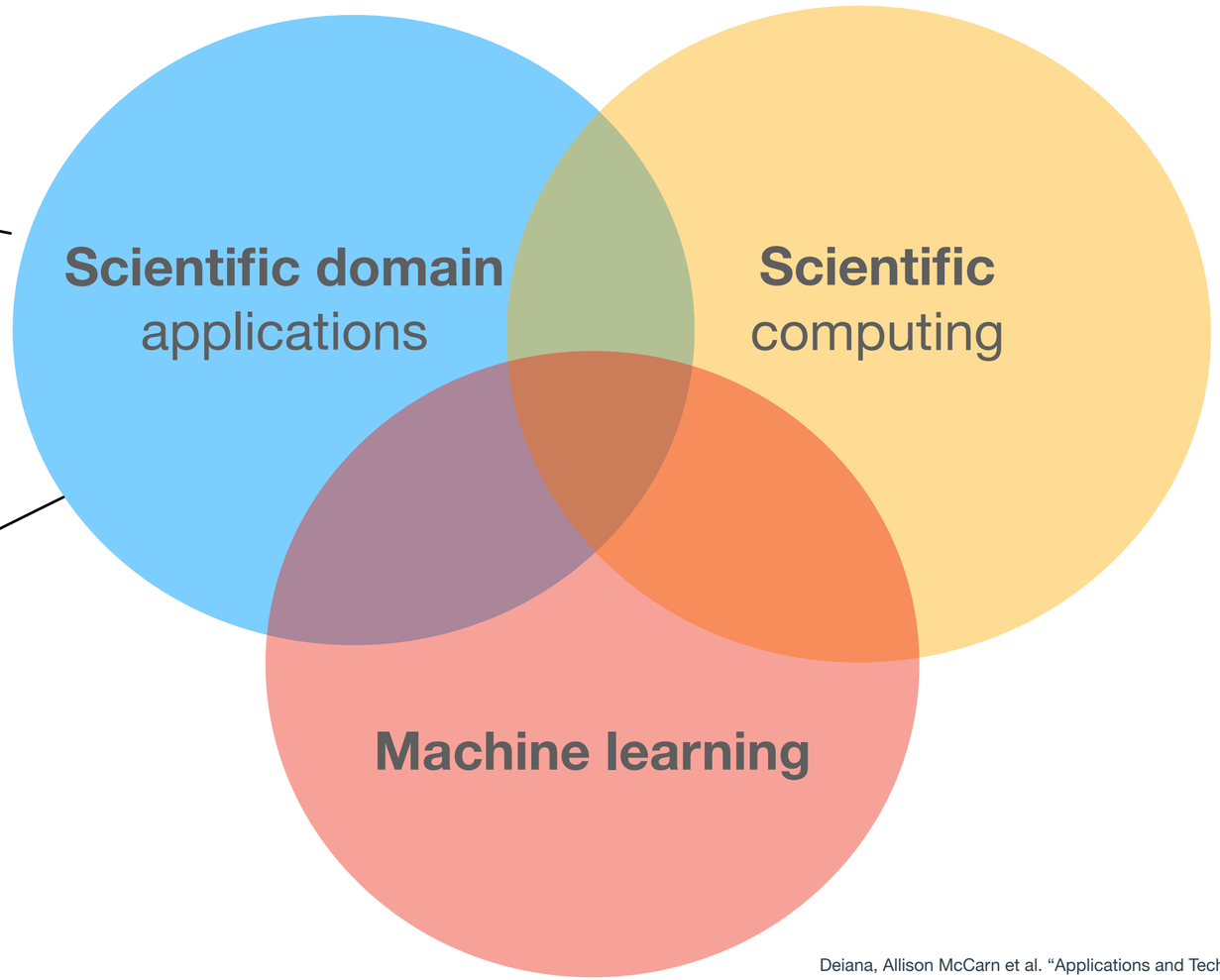
Today

- Introduction: What, why and examples.
- Agenda for the next days
- Theory: Introduction to kernel methods
- Theory: Kernel methods for solving PDEs
- Example: Solve Poisson’s equation in 1D
- Code: Implementation of the method

What is SciML ?

Emerging discipline within the data science community.

Computational physics
Genomics
Optics
Computer vision
Robotics
Material science
Medicine
Aerospace
…



Deiana, Allison McCarn et al. “Applications and Techniques for Fast Machine Learning in Science.” *Frontiers in Big Data* 5 (2021): n. pag.

Why to work on SciML ?

DOE: Basic research needs

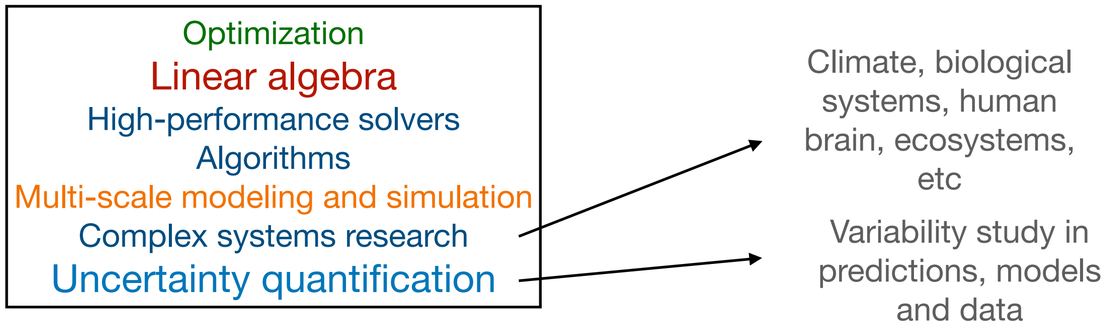
Foundational research

- Domain awareness
- Interpretability
- Robustness

Capability research

- ML-enhancing and simulation
- Intelligent automation
- Decision support for complex systems

Coherent + long-term research strategy in SciML and AI includes:



Workshop Report on Basic Research Needs for Scientific Machine Learning: Core Technologies for Artificial Intelligence. United States: N. p., 2019.

Some first applications

In the study of dynamics and model discovery

Hod Lipson, Columbia University

“Higher animals use some form of an "internal model" of themselves for planning complex actions and predicting their consequence, but it is not clear if and how these self-models are acquired or what form they take” Taken from Lipson’s website

Recent examples

SIAM 2023 Conference on Optimization, Seattle (WA)

Lu, Lu, et al. "Multifidelity deep neural operators for efficient learning of partial differential equations with application to fast inverse design of nanoscale heat transport." Physical Review Research 4.2 (2022): 023210.

ML for low fidelity data →

NNs learn operators between infinite dimensional spaces. They don't necessarily need large amounts of high fidelity data !

Chen, Mingkun, et al. "Physics-augmented deep learning for high-speed electromagnetic simulation and optimization." (2021).

ML for simulation in photonics design →

CNN to predict electromagnetic field distributions with ultra fast speeds and high accuracy for entire classes of dielectric photonic structures

Lu, Lu, et al. "Physics-informed neural networks with hard constraints for inverse design." SIAM Journal on Scientific Computing 43.6 (2021): B1105-B1132.

ML in a pipeline for inverse design →

In contrast to intuition-based approaches using heuristics, inverse design starts with the targeted functionality and sets it as an objective function to be optimized via partial-differential-equation-constrained (PDE-constrained) large-scale optimization

Concrete examples of SciML

Kernel methods

Day 1

Solving and learning non-linear PDEs with Gaussian processes

Chen, Y., Hosseini, B., Owhadi, H., & Stuart, A. M. JCP (2021)

Operator learning

Kernel methods are Competitive for Operator Learning

Battle, Darcy, Hosseini & Owhadi, arXiv:2304.13202 (2023)

A Kernel Approach for PDE Discovery and Operator Learning

Long, Mrvaljevic, Zhe & Hosseini, arXiv:2210.08140 (2022)

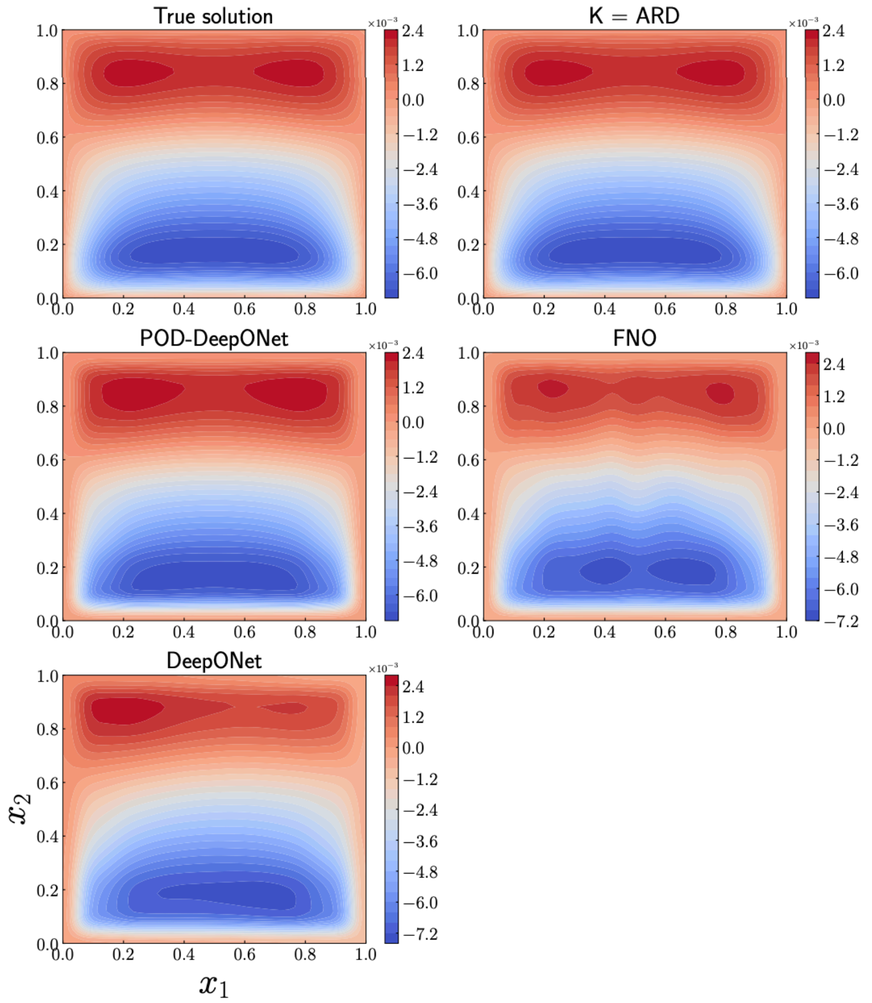


Image taken from: arXiv:2210.08140

Concrete examples of SciML

Sparse regression

Day 2

Discovering governing equations by
Sparse Identification of nonlinear dynamical systems

Brunton, Proctor & Kutz, National Academy of Sciences (2016)

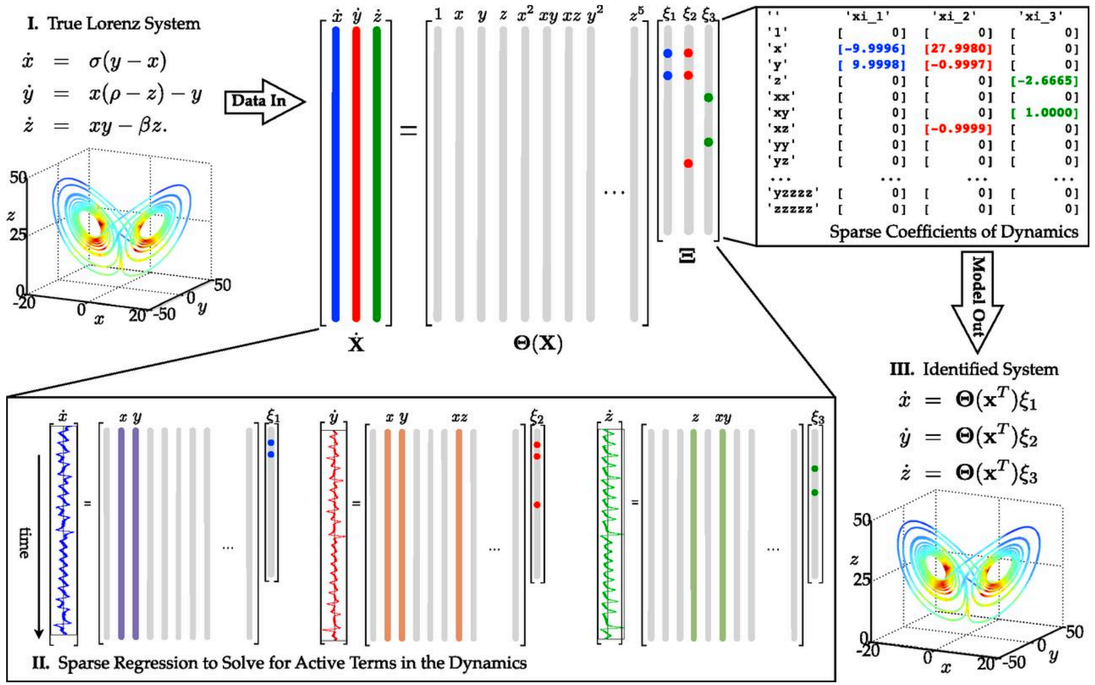
ML for PDEs

Data-driven discovery of PDEs

Rudy, Brunton, Proctor & Kutz, Science advances (2017)

Machine learning for PDEs

Brunton, Kutz, Science advances (2017)



Workshop Report on Basic Research Needs for Scientific Machine Learning: Core Technologies for Artificial Intelligence. United States: N. p., 2019.

Concrete examples of SciML

Neural networks

Day 3

Physics-informed neural networks

Raissi, Perdirakis & Karniadakis, JCP (2019)

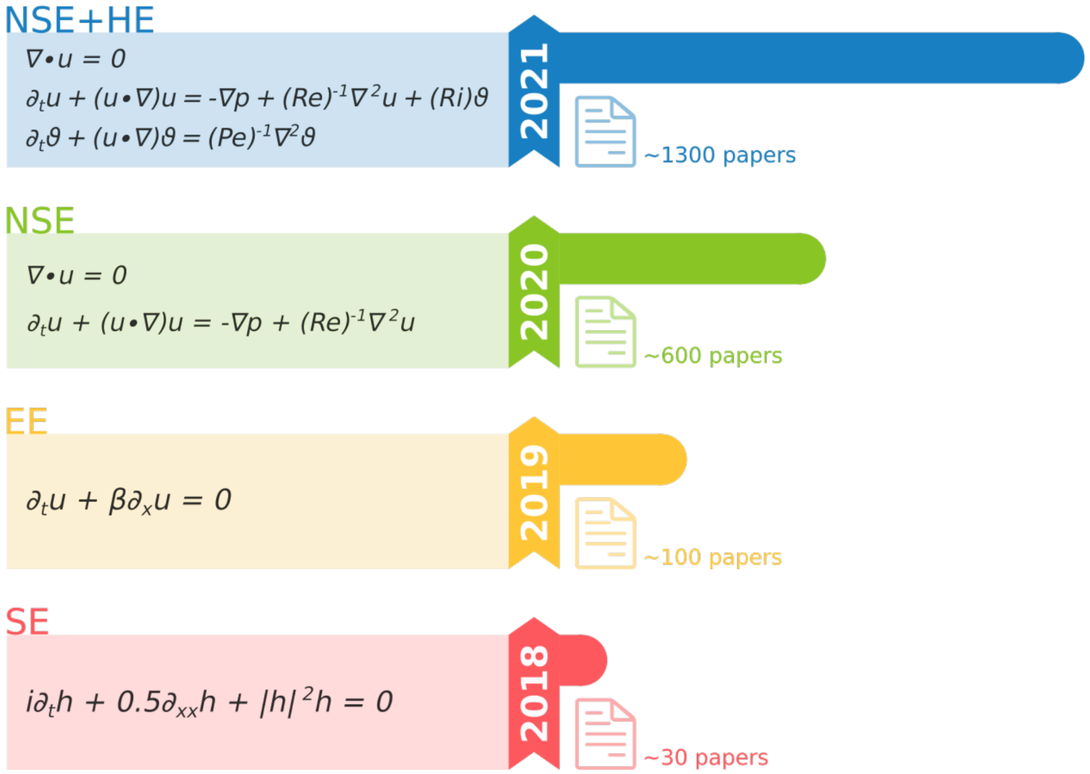
Operator learning

DeepONet

Lu, Pengzhan & Karniadakis, Nature machine intelligence (2019)

Fourier neural operator

Li, Kovachki, Azizzadenesheli, Liu, Bhattacharya, Stuart & Anandkumar, (2020)



Agenda + Logistics

Day 1

Goal: Solve Poisson's equation using kernels

Solving and learning non-linear PDEs with Gaussian processes
Chen, Y., Hosseini, B., Owhadi, H., & Stuart, A. M. JCP (2021)

poisson.ipynb

Day 2

Goal: Learn Lorentz system using SINDY

Discovering governing equations by Sparse Identification of nonlinear dynamical systems
Brunton, Proctor & Kutz, National Academy of Sciences (2016)

lorentz.ipynb

Day 3

Goal: Solve an ODE using PINNs

Physics-informed neural networks
Raissi, Perdirakis & Karniadakis, JCP (2019)

pde.ipynb

Please download your notebook here: https://drive.google.com/drive/folders/18_iKBiP_trJgI4SqM9iYpAmCM7PLpjuf?usp=sharing

Introduction to kernel methods

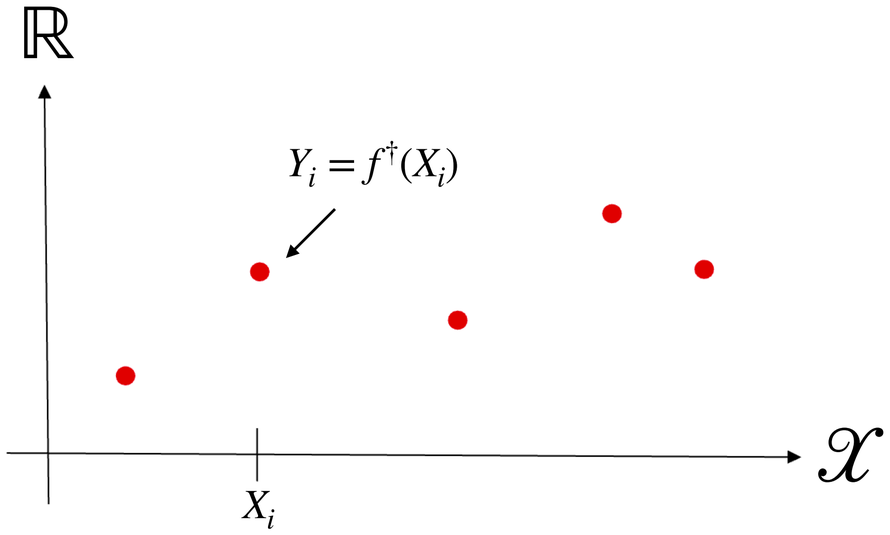
Problem

We want to approximate an unknown function

$$f^\dagger : \mathcal{X} \to \mathbb{R}$$

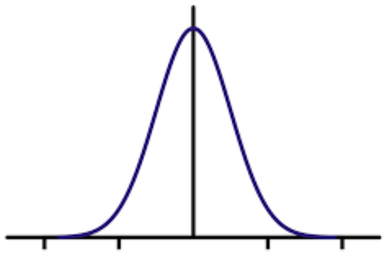
given the observed data $f^\dagger(X) = Y$ where $(X, Y) \in (\mathcal{X}^N, \mathbb{R}^N)$ and

$$\begin{aligned} X &:= (X_1, \ldots, X_N), \\ f^\dagger(X) &:= \big(f(X_1), \ldots, f(X_N)\big), \\ Y &:= (Y_1, \ldots, Y_N)\,. \end{aligned}$$



Four equivalent ways to think about kernels

1. Kernel. $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that $\forall m \in \mathbb{N} \wedge \{x_1, \ldots, x_m\} \subset \mathcal{X}$ we have the matrix $\left(K(x_i, x_j)\right)_{i,j=1}^{m}$ is positive definite and symmetric.



2. Feature map. $\exists$ a Hilbert space $\mathcal{F}$ and a map $\varphi : \mathcal{X} \to \mathcal{F}$ such that

$$K(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{F}} .$$

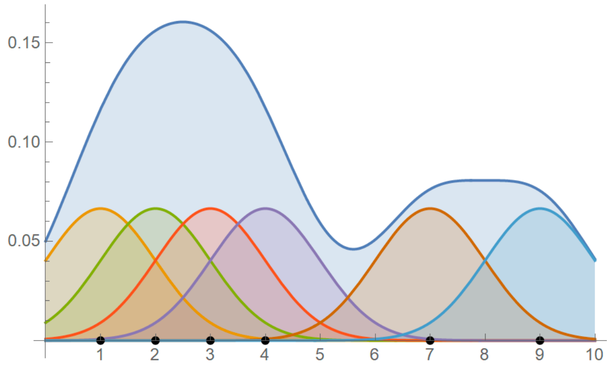
3. RKHS. $\exists$ a Hilbert space $\mathcal{H} := \{f : \mathcal{X} \to \mathbb{R}\}$ such that $f(x) = \langle f, K(x, \cdot) \rangle_{\mathcal{H}}$ for $x \in \mathcal{X}$ and $f \in \mathcal{H}$. Write $\|f\|_K := \|f\|_{\mathcal{H}}$.

4. GPs. $\exists$ a Gaussian process $\xi, \xi : \mathcal{X} \to$ Gaussian space , such that $K(x, x') = \mathbb{E}\left[\xi(x)\xi(x')\right]$

Write $\xi \sim \mathcal{N}(0, K)$.

Four equivalent ways to use kernels (to solve the problem)

Approximate $f^\dagger$ using:



1. Kernel. $f(x) = K(x, X)K(X, X)^{-1}Y$

2. Feature map. $f(x) = \langle \varphi(x), c \rangle_{\mathcal{F}}$ where $c \in \mathcal{F}, f(X) = Y$ and

   $\|c\|_{\mathcal{F}}$ is minimal.

3. RKHS(optimal recovery). Minimizer of

$$\begin{cases} \text{Minimize}_{f \in \mathcal{H}} & \|f\|_K \\ \text{subject to} & f(X) = Y \end{cases}$$

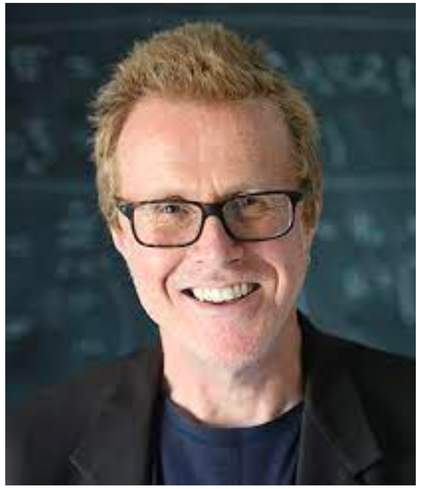
4. GP regression. $f(x) = \mathbb{E}\left[\xi(x) \,|\, \xi(X) = Y\right].$

Solving PDEs with kernels

Houman Owhadi

Yifan Chen

Bamdad Hosseini



Andrew Stuart

Caltech

UNIVERSITY of WASHINGTON

What is a partial differential equation?

Is an equation that involves two or more independent variables, an unknown function (dependent on those variables), and partial derivatives of the unknown function with respect to the independent variables.

$$Pu(\underline{x}) = -\operatorname{div} a(\underline{x}) \nabla u(\underline{x}) + \underline{b}(\underline{x})^{\top} \nabla u(\underline{x}) + c(\underline{x}) u(\underline{x}) = 0$$

We usually define a specific domain where we want to solve the equation, say $\Omega \subset \mathbb{R}^d$.

In order to find unique classical solutions we also need to define how the solution behaves at the boundary of Ω.

Dirichlet: $u(\underline{x}) = g(\underline{x}) \quad \forall \underline{x} \in \partial\Omega$

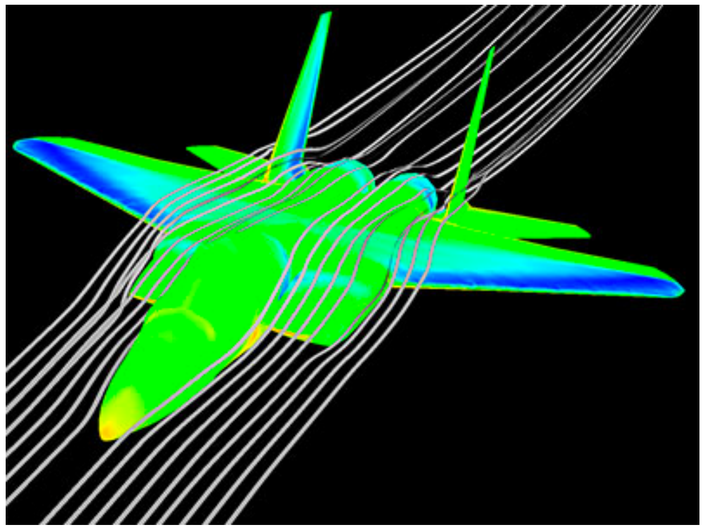
Neuman: $\partial_{\underline{n}} u(\underline{x}) = g(\underline{x}) \quad \forall \underline{x} \in \partial\Omega$

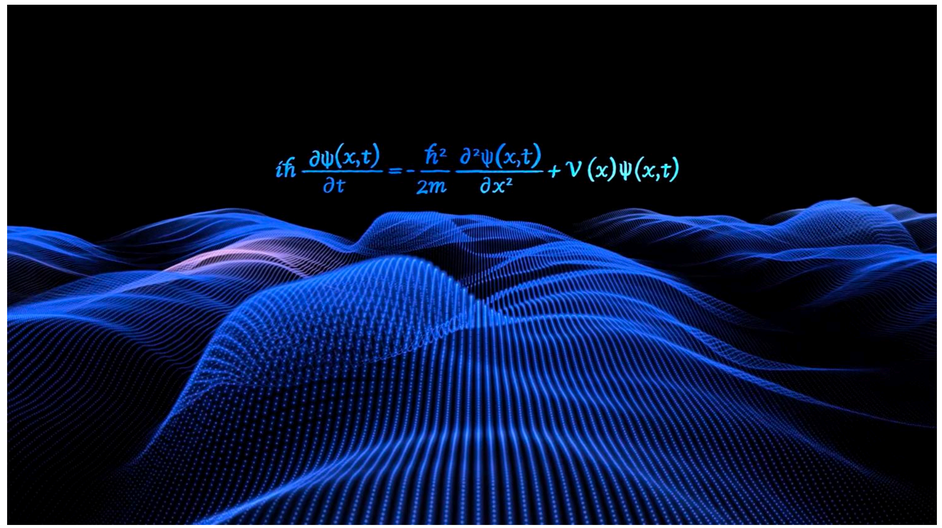
Neuman: $a u(\underline{x}) + b \partial_{\underline{n}} u(\underline{x}) = g(\underline{x}) \quad \forall \underline{x} \in \partial\Omega$

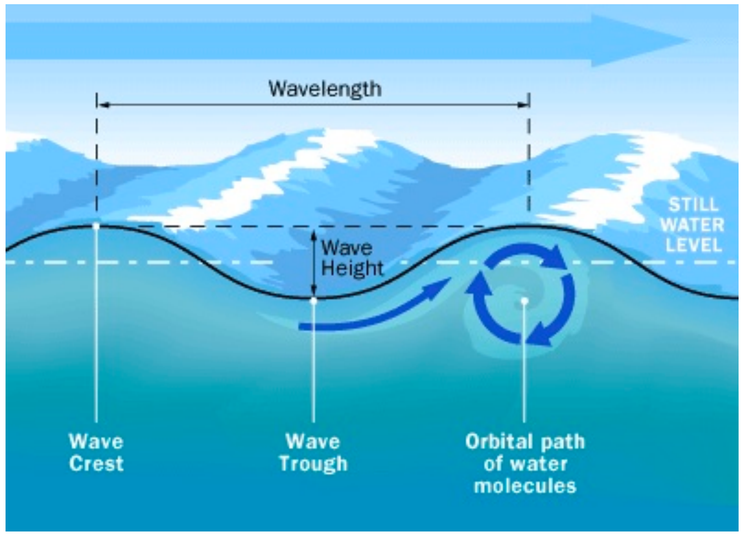
Evans, L. C. (2022). *Partial differential equations* (Vol. 19). American Mathematical Society.

PDEs are ubiquitous in science

- Equations control how the physical world behaves.
- PDEs are the basis for most scientific and engineering problems.







How to solve PDEs?

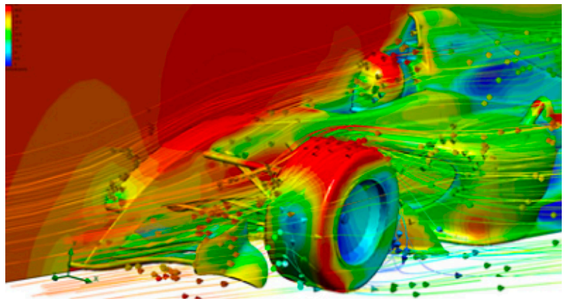
Analytical methods

- Integral transforms
- Green's functions
- Method of characteristics
- Separation of variables
- Eigenfunction expansions
- Power series
- Integral equations
- Coordinate transformations
- Fourier analysis methods
- …

Numerical methods

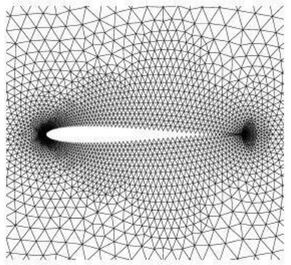
Mesh

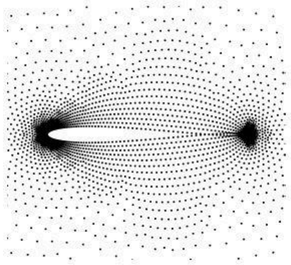
- Finite differences
- Finite volume
- Finite Elements
- …



Mesh-free

- Collocation methods
- Spectral methods
- Kernel methods
- …





Trefethen, L. N. (2000). *Spectral methods in MATLAB*. Society for industrial and applied mathematics.

Solving and learning PDEs with GPs

Chen, Hosseini, Owhadi, Stuart (2021) JCP

Physical problem (PDE) → Numerical solution

$$-\Delta u(x) = f(x) \text{ for all } x \in \Omega$$
$$u(x) = g(x) \text{ for all } x \in \partial\Omega$$

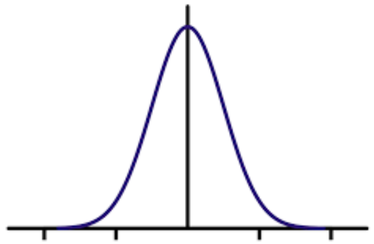
$$\longrightarrow \quad u^M \approx u^\dagger \text{ at } x \in \Omega$$

We assume that the problem above is well posed and has unique solution $u^\dagger$. Also, we take the source $f(x) = sin(x), g(x) = 0$ and $\overline{\Omega} = [0,1]$.

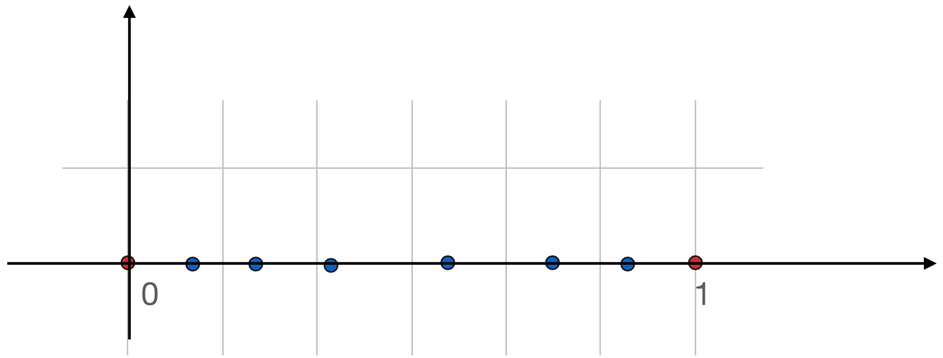
Method

To approximate numerically $u^\dagger$ we will first interpret the problem as an optimal recovery problem.

1. Choose a kernel $K : \overline{\Omega} \times \overline{\Omega} \to \mathbb{R}$ where $\overline{\Omega} = \Omega \cup \partial\Omega$ and we denote by $\mathcal{U}$ its RKHS with an associated norm $\| \cdot \|_{\mathcal{U}}$



2. Let $1 \leq M_\Omega < M < \infty,$ then we fix M points in $\overline{\Omega}$ where $X_{M_\Omega} = \{x_1, \ldots, x_{M_\Omega}\} \in \Omega$ and $X_{\partial M_\Omega} = \{x_{M_\Omega+1}, \ldots, x_M\} \in \partial\Omega$



Method

To approximate numerically $u^\dagger$ we will first interpret the problem as an optimal recovery problem.

The method approximates the solution $u^\dagger$ of the problem with a minimizer u^M of

$$\begin{cases} \text{Minimize}_{u \in \mathcal{U}} \|u\|_{\mathcal{U}} \\ \text{Subject to } -\Delta u(x_m) = f(x_m) \text{ for } m \in \{1,\dots,M_\Omega\}, \\ \qquad\qquad u(x_m) = 0 \text{ for } m \in \{M_{\Omega+1}, \dots, M\}\,. \end{cases}$$

Re-write the problem above as

$$\begin{cases} \text{Minimize } \|u\|_{\mathcal{U}} \\ \text{Subject to } \varphi_m(u) = f(x_m) \text{ for } m \in \{1,\dots,M_\Omega\}, \\ \qquad\qquad \varphi_m(u) = 0 \text{ for } m \in \{M_{\Omega+1}, \dots, M\}\,. \end{cases}$$

where

$$\varphi_m(u) = (\delta_{x_m} \circ -\Delta)(u) = -\Delta u(x_m) \text{ for } m \in \{1,\dots,M_\Omega\},$$
$$\varphi_m(u) = \delta_{x_m}(u) = u(x_m) \text{ for } m \in \{M_{\Omega+1}, \dots, M\}\,.$$

Define

$$\varphi(u) := (\varphi_1(u), \dots, \varphi_M(u))$$

to state the problem as

$$\begin{cases} \text{Minimize} \quad \|u\|_{\mathcal{U}} \\ \text{Subject to } \varphi(u) = \underline{y} \end{cases}$$

with $\underline{y} = \begin{pmatrix} f(X_{M_\Omega}) \\ 0_{M-M_\Omega} \end{pmatrix}$

Method

Use of a generalized representer theorem.

The problem

$$\begin{cases} \text{Minimize} & \|u\|_{\mathcal{U}} \\ \text{Subject to} & \varphi(u) = \underline{y} \end{cases}$$

with $\underline{y} = \begin{pmatrix} f(X_{M_\Omega}) \\ 0_{M-M_\Omega} \end{pmatrix}$ has a unique solution

$$u(\,\cdot\,) = K(\,\cdot\,, \varphi) K(\varphi, \varphi)^{-1} \underline{y}$$

See the proof of this result in Owhadi, H., & Scovel, C. (2019). Operator-Adapted Wavelets, Fast Solvers, and Numerical Homogenization: From a Game Theoretic Approach to Numerical Approximation and Algorithm Design (Vol. 35). Cambridge University Press. Ch. 12

where

$$K(\,\cdot\,, \varphi) = \begin{pmatrix} \vdots \\ -\Delta_x K(x, \cdot\,)\,|_{x=x_m} \\ \vdots \\ K(x_m, \cdot\,) \\ \vdots \end{pmatrix} \begin{matrix} 1 \\ \vdots \\ M_\Omega \\ M_\Omega + 1 \\ \vdots \\ M \end{matrix} \begin{matrix} \\ \Omega \\ \\ \\ \partial\Omega \\ \end{matrix}$$

and

$$K(\varphi, \varphi) = \begin{pmatrix} \Delta_x \Delta_y K(x, y)\,|_{x=x_m, y=x_{m'}} & -\Delta_x K(x, x_{m'})\,|_{x=x_m} \\ -\Delta_y K(x_m, y)\,|_{y=x_{m'}} & K(x_m, x_{m'}) \end{pmatrix}$$

Implementation

The condition number associated with the linear equation $Ax = b$ gives a bound on how inaccurate the solution x will be after approximation.

$$u(\,\cdot\,) = K(\,\cdot\,,\varphi)K(\varphi,\varphi)^{-1}\underline{y}$$

Here's the bottle neck !

The method relies in construting

$$K(\,\cdot\,,\varphi) = \begin{pmatrix} \vdots \\ -\Delta_x K(x,\,\cdot\,)\,|_{x=x_m} \\ \vdots \\ \vdots \\ K(x_m,\,\cdot\,) \\ \vdots \end{pmatrix} \begin{matrix} 1 \\ \vdots \\ M_\Omega \\ M_\Omega + 1 \\ \vdots \\ M \end{matrix}$$

and

$$K(\varphi,\varphi) = \begin{pmatrix} \Delta_x\Delta_y K(x,y)\,|_{x=x_m,y=x_{m'}} & -\Delta_x K(x,x_{m'})\,|_{x=x_m} \\ -\Delta_y K(x_m,y)\,|_{y=x_{m'}} & K(x_m,x_{m'}) \end{pmatrix}$$

Also, the matrix $K(\varphi,\varphi)$ can be ill-conditioned and needs regularization

$$K(\varphi,\varphi) \longleftarrow K(\varphi,\varphi) + \lambda R$$

where $R = \mathrm{Diag}(K(\varphi,\varphi))$

RBF kernel

$$u(\,\cdot\,) = K(\,\cdot\,, \varphi) K(\varphi, \varphi)^{-1} \underline{y}$$

Here's the bottle neck !

Also, the matrix $K(\varphi, \varphi)$ can be ill-conditioned and needs regularization. Consider the RBF kernel

$$K(x, y) := \exp\left(-\frac{|x - y|^2}{2\sigma^2}\right) \approx O(1)$$

$$\partial_x K(x, y) = -\frac{|x - y|}{2\sigma^2} k(x, y) \approx O\left(\frac{1}{\sigma^2}\right)$$

$$\vdots$$

Gets worst

Choice of kernels

Formulation of the method is independent beyond the choice of K beyond asking for sufficient regularity w.r.t. to its entries:

$-\Delta u(x) = f(x)$ for all $x \in \Omega$ $\longrightarrow$ $K(x, y)$

Second order PDE

At least twice differentiable

The ideal choice of a kernel would be the Green's function of the PDE, but then can solve analytically the PDE.

Usually, we don't expect polynomials as solutions of PDEs. So polynomial kernels are not a great choice.

RBF kernel may lead to bad conditioning and overly smooth solutions.

Choice of kernels

Matérn class

Using some theory from elliptic PDEs if we know the regularity of f (the RHS of the PDE),

then we know the regularity (in the Sobolev sense) of the solution u .

The family of kernels that is naturally adapted to such regularity classes is the Matérn kernel:

$$K(t) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{t}{\gamma} \right) K_\nu \left(\sqrt{2\nu} \frac{t}{\gamma} \right) .$$

Special cases:

$$K(t) = \exp\left(-\frac{t}{\gamma} \right)$$

$\nu = 1/2$

$$K(t) = \left(1 + \frac{\sqrt{3}t}{\gamma} \right) \exp\left(-\frac{\sqrt{3}t}{\gamma} \right)$$

$\nu = 3/2$

$$K(t) \to \exp\left(-\frac{t^2}{2\gamma^2} \right)$$

$\nu \to \infty$

Choice of kernels

Matérn class

The reason the Matérn kernels are desirable for PDEs and function approximation is that their RKHS, are equivalent to Sobolev spaces.

Theorem. Let $K(\underline{x}, \underline{y}) = K_\nu(\|\underline{x} - \underline{y}\|$ be a Matérn kernel with with index ν such that $s = \nu + \frac{d}{2}$ is an integer. Then the RKHS of K is norm equivalent with the Sobolev space $H^s\left(\mathbb{R}^d\right)$, i.e.,

$$C_1\|u\|_{H^s} \leqslant \|u\|_K \leqslant C_2\|u\|_{H^s}, \quad C_1, C_2 \geqslant 0$$

Kanagawa, et al. Gaussian Processes and Kernel Methods: A Review on Connections and Equivalences (2018)

Let’s run the demo