Numerics of Machine Learning
Lecture 08
Partial Differential Equations

Marvin Pförtner
8 December 2022

Eberhard Karls Universität Tübingen
Faculty of Science
Department of Computer Science
Chair for the Methods of Machine Learning

Outlook

- What are PDEs? Why are they important?
- How to integrate PDEs into probabilistic ML models?
- Practical Modeling Example

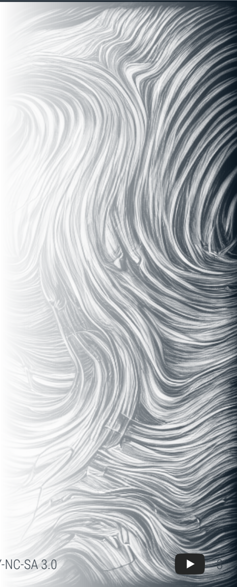
EBERHARD KARLS UNIVERSITÄT TÜBINGEN

PDEs are the language of mechanistic knowledge

- PDEs are extremely precise *mechanistic models* of the real world

Motivation

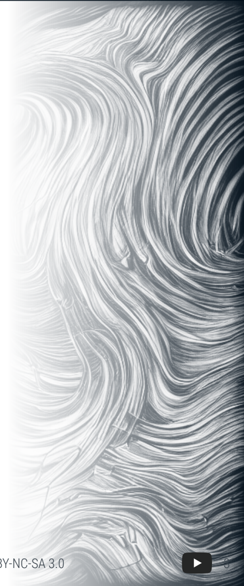
Why PDEs?



- PDEs are extremely precise *mechanistic models* of the real world
- example: fluid mechanics

Motivation

Why PDEs?



- PDEs are extremely precise *mechanistic models* of the real world
- example: fluid mechanics $\Rightarrow$ Navier Stokes equations
- PDE problems are typically large-scale and difficult to solve

Motivation

Why PDEs?

- PDEs are extremely precise *mechanistic models* of the real world
- example: fluid mechanics $\Rightarrow$ Navier Stokes equations
- PDE problems are typically large-scale and difficult to solve
- theory and practice of PDEs are a highly active field of research

Motivation

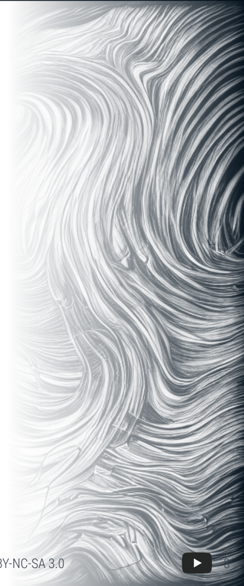
Why PDEs?

- PDEs are extremely precise *mechanistic models* of the real world
- example: fluid mechanics $\Rightarrow$ Navier Stokes equations
- PDE problems are typically large-scale and difficult to solve
- theory and practice of PDEs are a highly active field of research

A fundamental problem in analysis is to decide whether such smooth, physically reasonable solutions exist for the Navier–Stokes equations. To give reasonable leeway to solvers while retaining the heart of the problem, we ask for a proof of one of the following four statements.

(A) Existence and smoothness of Navier–Stokes solutions on $\mathbb{R}^3$. Take $\nu > 0$ and $n = 3$. Let $u^\circ(x)$ be any smooth, divergence-free vector field satisfying (4). Take $f(x,t)$ to be identically zero. Then there exist smooth functions $p(x,t), u_i(x,t)$ on $\mathbb{R}^3 \times [0,\infty)$ that satisfy (1), (2), (3), (6), (7).

(from the official Clay Mathematics Institute Problem Description for the Navier Stokes Millennium Problem)

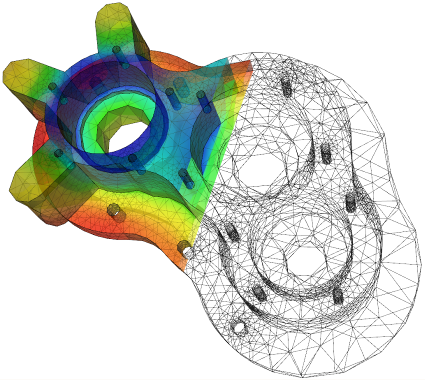


- general (nonlinear) PDEs are difficult to understand and solve $\Rightarrow$ focus on linear PDEs in this lecture

Motivation

Why linear PDEs?

- general (nonlinear) PDEs are difficult to understand and solve $\Rightarrow$ focus on linear PDEs in this lecture
- models of many physical processes are based on linear PDEs
  - thermal conduction (heat equation)

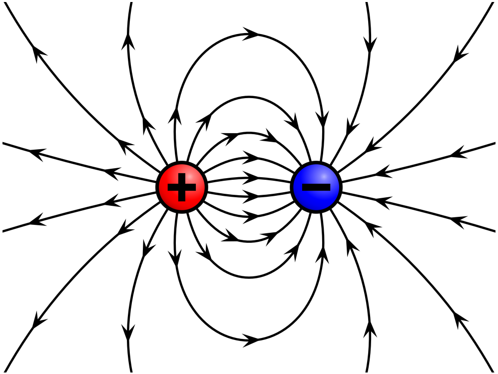


©User A1, Wikimedia Commons, CC BY-SA 3.0

Motivation

Why linear PDEs?

- general (nonlinear) PDEs are difficult to understand and solve ⇒ focus on linear PDEs in this lecture
- models of many physical processes are based on linear PDEs
  - thermal conduction (heat equation)
  - electromagnetism (Maxwell's equations)



Motivation

Why linear PDEs?

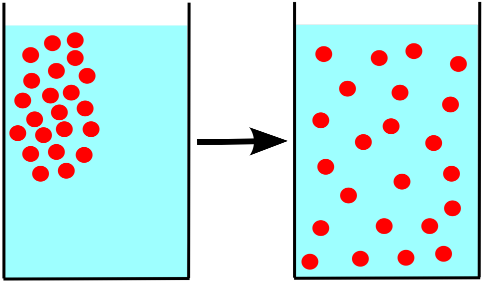
- general (nonlinear) PDEs are difficult to understand and solve $\Rightarrow$ focus on linear PDEs in this lecture
- models of many physical processes are based on linear PDEs
  - thermal conduction (heat equation)
  - electromagnetism (Maxwell's equations)
  - wave mechanics (wave equation)

©Roger McLassus, Wikimedia Commons, CC BY-SA 3.0

Motivation

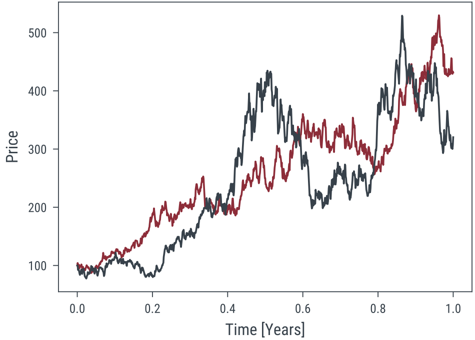
Why linear PDEs?

- general (nonlinear) PDEs are difficult to understand and solve $\Rightarrow$ focus on linear PDEs in this lecture
- models of many physical processes are based on linear PDEs
  - thermal conduction (heat equation)
  - electromagnetism (Maxwell's equations)
  - wave mechanics (wave equation)
  - Brownian motion (Fokker-Planck/Kolmogorov forward equation)

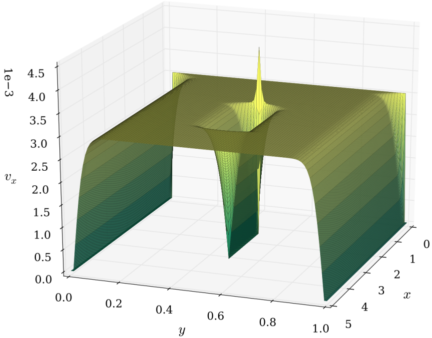


©JrPol, Wikimedia Commons, CC BY-SA 3.0

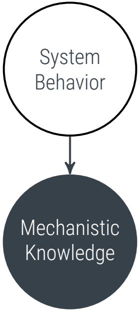
- general (nonlinear) PDEs are difficult to understand and solve $\Rightarrow$ focus on linear PDEs in this lecture
- models of many physical processes are based on linear PDEs
    - thermal conduction (heat equation)
    - electromagnetism (Maxwell's equations)
    - wave mechanics (wave equation)
    - Brownian motion (Fokker-Planck/Kolmogorov forward equation)
- the famous Black-Scholes equation in mathematical finance is a linear PDE



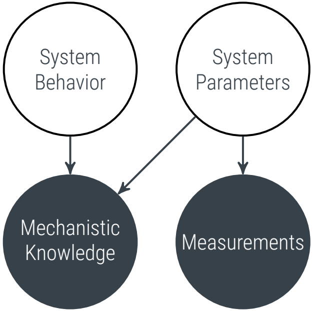
- general (nonlinear) PDEs are difficult to understand and solve $\Rightarrow$ focus on linear PDEs in this lecture
- models of many physical processes are based on linear PDEs
  - thermal conduction (heat equation)
  - electromagnetism (Maxwell's equations)
  - wave mechanics (wave equation)
  - Brownian motion (Fokker-Planck/Kolmogorov forward equation)
- the famous Black-Scholes equation in mathematical finance is a linear PDE
- linear approximations are an important tool in the analysis and numerical solution of nonlinear PDEs



- PDEs usually depend on *parameters* which can only be determined through *noisy measurements*



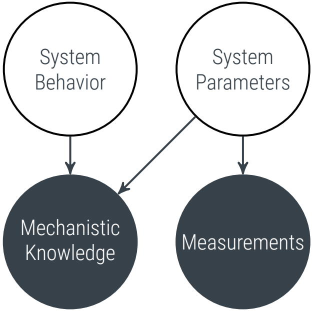
- PDEs usually depend on *parameters* which can only be determined through *noisy measurements*



Goal

Fusing Probabilistic and Mechanistic Models

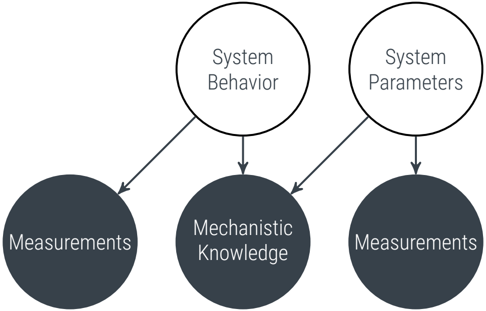
- PDEs usually depend on *parameters* which can only be determined through *noisy measurements*
- examples:
  - strength and distribution of heat sources
  - charge distribution
  - material parameters
  - forces



Goal

Fusing Probabilistic and Mechanistic Models

- PDEs usually depend on *parameters* which can only be determined through *noisy measurements*
- examples:
  - strength and distribution of heat sources
  - charge distribution
  - material parameters
  - forces

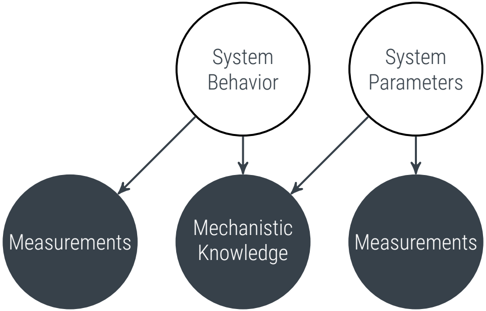


Goal

Fusing Probabilistic and Mechanistic Models

- PDEs usually depend on *parameters* which can only be determined through *noisy measurements*
- examples:
  - strength and distribution of heat sources
  - charge distribution
  - material parameters
  - forces

⇒ use Bayesian *statistical estimation* to fuse (exact) *mechanistic knowledge* and (noisy/uncertain) *measurement data*



We look for a function $u : D \subset \mathbb{R}^d \to \mathbb{R}^n$ which solves the equation

$$\mathcal{D}[u] = f$$

on the interior of D, where $\mathcal{D}$ is a linear differential operator.

What is a linear PDE?

We look for a function $u : D \subset \mathbb{R}^d \to \mathbb{R}^n$ which solves the equation

$$\mathcal{D}[u] = f$$

on the interior of D, where $\mathcal{D}$ is a linear differential operator.

Example: Linear Differential Operators

- Laplacian ($n = 1$)

$$\mathcal{D} = \Delta = \sum_{i=1}^{d} \frac{\partial^2}{\partial x_i^2}$$

What is a linear PDE?

Linear Systems in Function Spaces

We look for a function $u : D \subset \mathbb{R}^d \to \mathbb{R}^n$ which solves the equation

$$\mathcal{D}[u] = f$$

on the interior of D, where $\mathcal{D}$ is a linear differential operator.

Example: Linear Differential Operators

- Laplacian ($n = 1$)

$$\mathcal{D} = \Delta = \sum_{i=1}^{d} \frac{\partial^2}{\partial x_i^2}$$

- Affine ODE ($d = 1, n \geq 1$)

$$\mathcal{D}[u] = \frac{\mathrm{d}u}{\mathrm{d}t} - A(t)u(t) \qquad \Rightarrow \qquad \frac{\mathrm{d}u}{\mathrm{d}t} = A(t)u(t) + f(t) =: \tilde{f}(u(t), t)$$

What is a linear PDE?

Linear Systems in Function Spaces

We look for a function $u : D \subset \mathbb{R}^d \to \mathbb{R}^n$ which solves the equation

$$\mathcal{D}[u] = f$$

on the interior of D, where $\mathcal{D}$ is a linear differential operator.

Problems

- usually no analytic solution $\Rightarrow$ numerical solvers necessary $\Rightarrow$ *discretization error*

What is a linear PDE?

Linear Systems in Function Spaces

We look for a function $u : D \subset \mathbb{R}^d \to \mathbb{R}^n$ which solves the equation

$$\mathcal{D}[u] = f$$

on the interior of D, where $\mathcal{D}$ is a linear differential operator.

Problems

- usually no analytic solution $\Rightarrow$ numerical solvers necessary $\Rightarrow$ *discretization error*
- *parameters* of the PDE (diffop parameters, right-hand side, etc.) are usually *not known exactly*

What is a linear PDE?

Linear Systems in Function Spaces

We look for a function $u : D \subset \mathbb{R}^d \to \mathbb{R}^n$ which solves the equation

$$\mathcal{D}[u] = f$$

on the interior of D, where $\mathcal{D}$ is a linear differential operator.

Problems

- usually no analytic solution $\Rightarrow$ numerical solvers necessary $\Rightarrow$ *discretization error*
- *parameters* of the PDE (diffop parameters, right-hand side, etc.) are usually *not known exactly*
- classical solvers sometimes difficult to embed in computational pipelines

Boundary Value Problems

Because simulating the entire universe turns out to be difficult...

- PDEs by themselves don't have a unique solution

Boundary Value Problems

Because simulating the entire universe turns out to be difficult...

- PDEs by themselves don't have a unique solution
- example: Poisson equation ($\Delta u = f$)

$$\Delta(Ax + b) = \operatorname{tr}\left(\nabla\nabla^\top (Ax + b)\right) = \operatorname{tr}(0) = 0$$

Boundary Value Problems

Because simulating the entire universe turns out to be difficult...

- PDEs by themselves don't have a unique solution
- example: Poisson equation ($\Delta u = f$)

$$\Delta(Ax + b) = \operatorname{tr}\left(\nabla\nabla^\top(Ax + b)\right) = \operatorname{tr}(0) = 0$$

$\Rightarrow$ for any solution $u^\star$ to the Poisson equation, $u^\star + Ax + b$ is also a solution

Boundary Value Problems

Because simulating the entire universe turns out to be difficult...

- PDEs by themselves don't have a unique solution
- example: Poisson equation ($\Delta u = f$)

$$\Delta(Ax + b) = \operatorname{tr}\left(\nabla\nabla^\top(Ax + b)\right) = \operatorname{tr}(0) = 0$$

$\Rightarrow$ for any solution $u^\star$ to the Poisson equation, $u^\star + Ax + b$ is also a solution

- uniqueness is typically achieved by requiring (linear) boundary conditions to hold

$$\mathcal{B}[u] = g, \qquad \text{where } \mathcal{B} \text{ linear and } g \colon \partial D \to \mathbb{R}^n$$

Boundary Value Problems

Because simulating the entire universe turns out to be difficult...

- PDEs by themselves don't have a unique solution
- example: Poisson equation ($\Delta u = f$)

$$\Delta(Ax + b) = \operatorname{tr}\left(\nabla\nabla^\top(Ax + b)\right) = \operatorname{tr}(0) = 0$$

$\Rightarrow$ for any solution $u^\star$ to the Poisson equation, $u^\star + Ax + b$ is also a solution

- uniqueness is typically achieved by requiring (linear) boundary conditions to hold

$$\mathcal{B}[u] = g, \qquad \text{where } \mathcal{B} \text{ linear and } g : \partial D \to \mathbb{R}^n$$

- physical intuition: summary of all influences on the outside of the simulation domain to a single known boundary function g

Boundary Value Problems

Because simulating the entire universe turns out to be difficult...

- PDEs by themselves don't have a unique solution
- example: Poisson equation ($\Delta u = f$)

$$\Delta(Ax + b) = \operatorname{tr}\left(\nabla\nabla^\top (Ax + b)\right) = \operatorname{tr}(0) = 0$$

$\Rightarrow$ for any solution $u^\star$ to the Poisson equation, $u^\star + Ax + b$ is also a solution

- uniqueness is typically achieved by requiring (linear) boundary conditions to hold

$$\mathcal{B}[u] = g, \qquad \text{where } \mathcal{B} \text{ linear and } g \colon \partial D \to \mathbb{R}^n$$

- physical intuition: summary of all influences on the outside of the simulation domain to a single known boundary function g
- PDE + Boundary Conditions = Boundary Value Problem (BVP)

Boundary Value Problems

Because simulating the entire universe turns out to be difficult...

- PDEs by themselves don't have a unique solution
- example: Poisson equation ($\Delta u = f$)

$$\Delta(Ax + b) = \operatorname{tr}\left(\nabla\nabla^\top (Ax + b)\right) = \operatorname{tr}(0) = 0$$

$\Rightarrow$ for any solution $u^\star$ to the Poisson equation, $u^\star + Ax + b$ is also a solution

- uniqueness is typically achieved by requiring (linear) boundary conditions to hold

$$\mathcal{B}[u] = g, \qquad \text{where } \mathcal{B} \text{ linear and } g \colon \partial D \to \mathbb{R}^n$$

- physical intuition: summary of all influences on the outside of the simulation domain to a single known boundary function g
- PDE + Boundary Conditions = Boundary Value Problem (BVP)
- example: Dirichlet boundary conditions $\mathcal{B}[u] = u|_{\partial D}$

PDEs are statements about functions and functions are (typically) infinite-dimensional objects.

Functions are Vectors

A Crash Course on Function Spaces

Vector space	$V = \mathbb{R}^n$	$V \subseteq \mathbb{R}^{\mathcal{X}}$
Addition and Scalar Multiplication	$(\alpha v + \beta w)_i = \alpha v_i + \beta w_i$	$(\alpha f + \beta g)(x) = \alpha f(x) + \beta g(x)$

Functions are Vectors

A Crash Course on Function Spaces

Vector space	$V = \mathbb{R}^n$	$V \subseteq \mathbb{R}^{\mathcal{X}}$
Addition and Scalar Multiplication	$(\alpha v + \beta w)_i = \alpha v_i + \beta w_i$	$(\alpha f + \beta g)(x) = \alpha f(x) + \beta g(x)$
Bases	$v = \sum_{i=1}^{n} v_i e_i = \sum_{i=1}^{n} v'_i b_i$	(sometimes) $f = \sum_{i=1}^{\infty} \alpha_i \phi_i$

Functions are Vectors

A Crash Course on Function Spaces

Vector space	$V = \mathbb{R}^n$	$V \subseteq \mathbb{R}^{\mathcal{X}}$
Addition and Scalar Multiplication	$(\alpha v + \beta w)_i = \alpha v_i + \beta w_i$	$(\alpha f + \beta g)(x) = \alpha f(x) + \beta g(x)$
Bases	$v = \sum_{i=1}^n v_i e_i = \sum_{i=1}^n v_i' b_i$	(sometimes) $f = \sum_{i=1}^\infty \alpha_i \phi_i$
Linear Maps	Matrix $M \in \mathbb{R}^{m \times n}$ $M(\alpha v + \beta w) = \alpha M v + \beta M w$	Linear Operator $\mathcal{L}: V \mapsto W$ $\mathcal{L}[\alpha f + \beta g] = \alpha \mathcal{L}[f] + \beta \mathcal{L}[g]$

Functions are Vectors

A Crash Course on Function Spaces

Vector space	$V = \mathbb{R}^n$	$V \subseteq \mathbb{R}^{\mathcal{X}}$
Addition and Scalar Multiplication	$(\alpha v + \beta w)_i = \alpha v_i + \beta w_i$	$(\alpha f + \beta g)(x) = \alpha f(x) + \beta g(x)$
Bases	$v = \sum_{i=1}^n v_i e_i = \sum_{i=1}^n v_i' b_i$	(sometimes) $f = \sum_{i=1}^\infty \alpha_i \phi_i$
Linear Maps	Matrix $M \in \mathbb{R}^{m \times n}$ $M(\alpha v + \beta w) = \alpha M v + \beta M w$	Linear Operator $\mathcal{L} \colon V \mapsto W$ $\mathcal{L}[\alpha f + \beta g] = \alpha \mathcal{L}[f] + \beta \mathcal{L}[g]$
Norms $\Rightarrow$ Banach spaces	e.g. $\|v\|_\infty := \max_{i=1,\dots,n} \lvert v_i \rvert$ $\Rightarrow (\mathbb{R}^n, \|\cdot\|_\infty)$	e.g. $\|f\|_\infty := \sup_{x \in \mathcal{X}} \lvert f(x) \rvert$ $\Rightarrow (C^k([a, b]), \|\cdot\|_\infty)$

Functions are Vectors

A Crash Course on Function Spaces

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Vector space	$V = \mathbb{R}^n$	$V \subseteq \mathbb{R}^{\mathcal{X}}$
Addition and Scalar Multiplication	$(\alpha v + \beta w)_i = \alpha v_i + \beta w_i$	$(\alpha f + \beta g)(x) = \alpha f(x) + \beta g(x)$
Bases	$v = \sum_{i=1}^n v_i e_i = \sum_{i=1}^n v_i' b_i$	(sometimes) $f = \sum_{i=1}^\infty \alpha_i \phi_i$
Linear Maps	Matrix $M \in \mathbb{R}^{m \times n}$ $M(\alpha v + \beta w) = \alpha M v + \beta M w$	Linear Operator $\mathcal{L} \colon V \mapsto W$ $\mathcal{L}[\alpha f + \beta g] = \alpha \mathcal{L}[f] + \beta \mathcal{L}[g]$
Norms $\Rightarrow$ Banach spaces	e.g. $\|v\|_\infty := \max_{i=1,\dots,n} \lvert v_i \rvert$ $\Rightarrow (\mathbb{R}^n, \|\cdot\|_\infty)$	e.g. $\|f\|_\infty := \sup_{x \in \mathcal{X}} \lvert f(x) \rvert$ $\Rightarrow (C^k([a,b]), \|\cdot\|_\infty)$
Inner Products $\Rightarrow$ Hilbert spaces	e.g. $\langle v, w \rangle_2 := \sum_{i=1}^n v_i w_i$ $\Rightarrow$ Euclidean space $(\mathbb{R}^n, \langle \cdot, \cdot \rangle_2)$	e.g. $\langle f, g \rangle_{L_2} := \int_{\mathcal{X}} f(x) g(x)\, \mu(dx)$

$\Rightarrow$ A linear PDE $\mathcal{D}[u] = f$ is a linear system in infinite-dimensional vector spaces of functions.

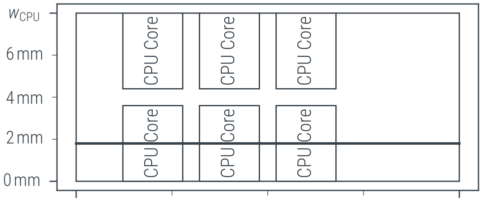
Toy Example of a Physical Model: The Heat Distribution in a CPU

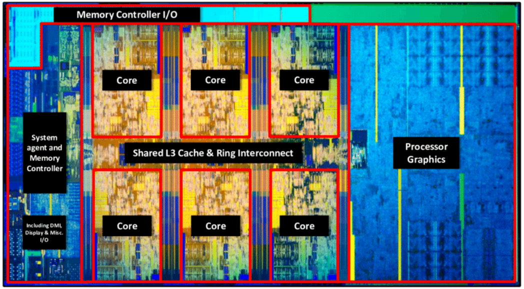
Spatial Domain: $D_{\text{CPU}} = [0, l_{\text{CPU}}] \times [0, w_{\text{CPU}}] \times [0, d_{\text{CPU}}] \subset \mathbb{R}^3$

©ElooKoN, Wikimedia Commons, CC BY-SA 4.0

©Ser Amantio di Nicolao, Wikimedia Commons, CC BY-SA 2.0

Spatial Domain: $D_{\text{CPU,2D}} = [0, l_{\text{CPU}}] \times [0, w_{\text{CPU}}] \subset \mathbb{R}^2$



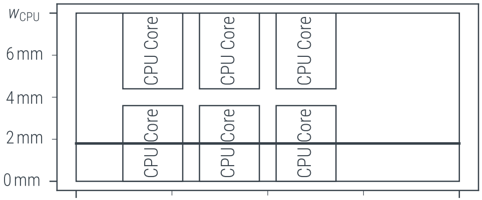


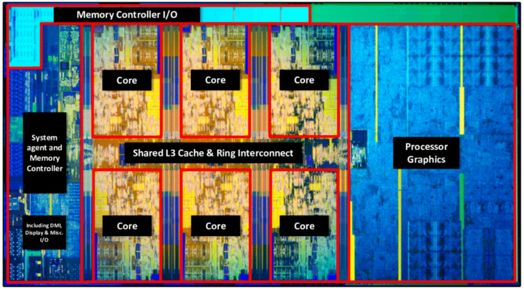
from Hebbar [2018]

A Linear PDE for Computer Scientists

The Heat Distribution in a CPU

Spatial Domain: $D_{\text{CPU,1D}} = [0, l_{\text{CPU}}] \subset \mathbb{R}$

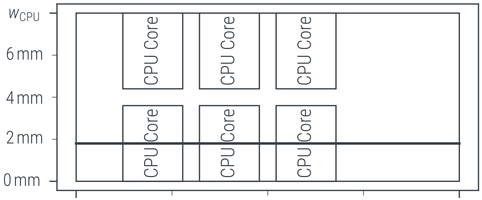


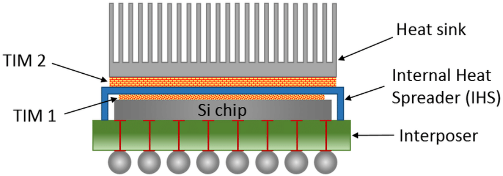


from Hebbar [2018]

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Spatial Domain: $D_{\text{CPU},1\text{D}} = [0, l_{\text{CPU}}] \subset \mathbb{R}$



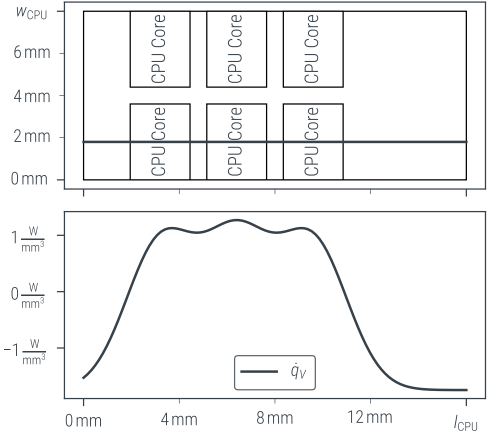


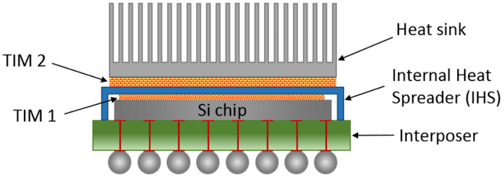
from Nylander [2018]

A Linear PDE for Computer Scientists

The Heat Distribution in a CPU

Spatial Domain: $D_{\text{CPU,1D}} = [0, l_{\text{CPU}}] \subset \mathbb{R}$





from Nylander [2018]

A Linear PDE for Computer Scientists II

The Heat Distribution in a CPU

Lienhard and Lienhard [2020]

Heat Equation

$$c_p \rho \frac{\partial u}{\partial t} - \kappa \Delta u = \dot{q}_V$$

where

- $u\colon [0, T] \times D \rightharpoonup \mathbb{R}$ *temperature*
- c_p, ρ, κ material parameters
- $\dot{q}_V\colon [0, T] \times D \rightharpoonup \mathbb{R}$ *heat source*

Heat Equation

$$c_p \rho \frac{\partial u}{\partial t} - \kappa \Delta u = \dot{q}_V$$

where

- $u \colon [0, T] \times D \to \mathbb{R}$ *temperature*
- c_p, ρ, κ material parameters
- $\dot{q}_V \colon [0, T] \times D \to \mathbb{R}$ *heat source*

Stationary Heat Equation

$$-\kappa \Delta u = \dot{q}_V$$

where

- $u \colon D \to \mathbb{R}$ *temperature*
- κ material parameter
- $\dot{q}_V \colon D \to \mathbb{R}$ *heat source*

Heat Equation

$$c_p \rho \frac{\partial u}{\partial t} - \kappa \Delta u = \dot{q}_V$$

where

- $u : [0, T] \times D \rightharpoonup \mathbb{R}$ *temperature*
- c_p, ρ, κ material parameters
- $\dot{q}_V : [0, T] \times D \rightharpoonup \mathbb{R}$ *heat source*

Stationary Heat Equation

$$-\kappa \frac{\mathrm{d}^2 u}{\mathrm{d}x^2} = \dot{q}_V$$

where

- $u : \mathbb{R} \rightharpoonup \mathbb{R}$ *temperature*
- κ material parameter
- $\dot{q}_V : \mathbb{R} \rightharpoonup \mathbb{R}$ *heat source*

(Linear) PDEs are Indirect Observations of Their Solution

Conservation Laws and Information Operators

- *conservation laws* are among the most fundamental laws of physics (conservation of energy, mass, momentum, charge, . . .) $\Rightarrow$ usually expressed as PDEs

(Linear) PDEs are Indirect Observations of Their Solution

Conservation Laws and Information Operators

- *conservation laws* are among the most fundamental laws of physics (conservation of energy, mass, momentum, charge, ...) $\Rightarrow$ usually expressed as PDEs
- example: the heat equation states conservation of (heat) energy

$$c_p \rho \frac{\partial u}{\partial t} = \kappa \Delta u + \dot{q}_V$$

(Linear) PDEs are Indirect Observations of Their Solution

Conservation Laws and Information Operators

- *conservation laws* are among the most fundamental laws of physics (conservation of energy, mass, momentum, charge, ...) $\Rightarrow$ usually expressed as PDEs
- example: the heat equation states conservation of (heat) energy

$$c_p \rho \frac{\partial u}{\partial t} = \kappa \Delta u + \dot{q}_V$$

- assuming a PDE to hold often amounts to the *observation* that a quantity is conserved locally

(Linear) PDEs are Indirect Observations of Their Solution

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

- *conservation laws* are among the most fundamental laws of physics (conservation of energy, mass, momentum, charge, . . .) $\Rightarrow$ usually expressed as PDEs
- example: the heat equation states conservation of (heat) energy

$$c_p \rho \frac{\partial u}{\partial t} = \kappa \Delta u + \dot{q}_V$$

- assuming a PDE to hold often amounts to the *observation* that a quantity is conserved locally
- generally, PDEs are *observations* of some *local mathematical property* of the unknown solution function

$$\mathcal{I}[u] := \mathcal{D}[u] - f = 0$$

(Linear) PDEs are Indirect Observations of Their Solution

Conservation Laws and Information Operators

- *conservation laws* are among the most fundamental laws of physics (conservation of energy, mass, momentum, charge, ...) $\Rightarrow$ usually expressed as PDEs
- example: the heat equation states conservation of (heat) energy

$$c_p \rho \frac{\partial u}{\partial t} = \kappa \Delta u + \dot{q}_V$$

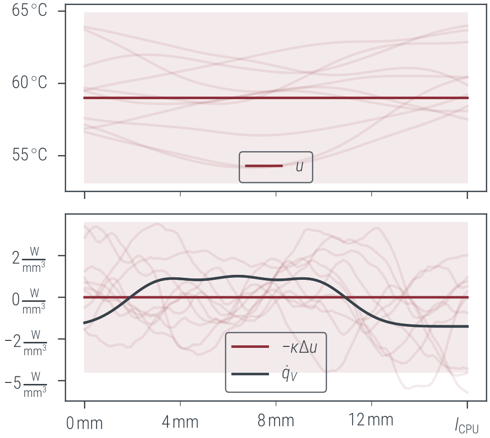
- assuming a PDE to hold often amounts to the *observation* that a quantity is conserved locally
- generally, PDEs are *observations* of some *local mathematical property* of the unknown solution function

$$\mathcal{I}[u] := \mathcal{D}[u] - f = 0$$

$\Rightarrow$ *information operator* (Cockayne et al. [2019], Tronarp et al. [2019])

Prior

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

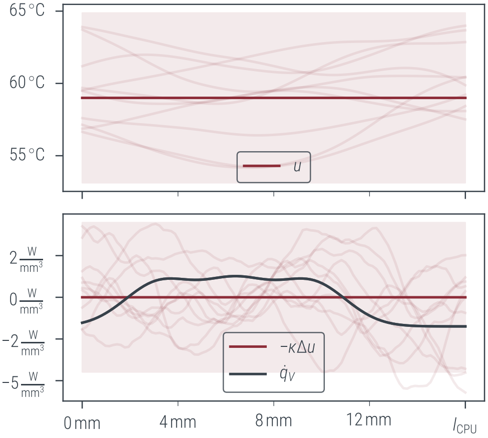


Prior

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

Observations

$$\mathcal{I}_{\mathrm{PDE}}[u] = \underbrace{-\kappa \Delta u}_{=:\mathcal{D}[u]} - \dot{q}_V = 0$$



Prior

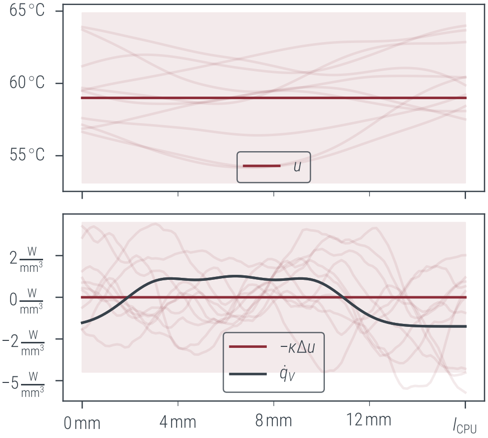
$$u \sim \mathcal{GP}(m, k)$$

Observations

$$\mathcal{I}_{\mathrm{PDE}}[u] = \underbrace{-\kappa \Delta u}_{=:\mathcal{D}[u]} - \dot{q}_V = 0$$

(Prior) Predictive

$$\mathcal{D}[u] \sim ?$$



Prior

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

Observations

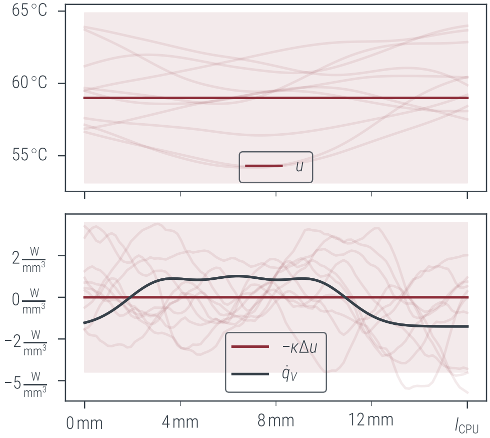
$$\mathcal{I}_{\text{PDE}}[u] = \underbrace{-\kappa \Delta u}_{=:\mathcal{D}[u]} - \dot{q}_V = 0$$

(Prior) Predictive

$$\mathcal{D}[u] \sim ?$$

Posterior

$$u \mid \mathcal{D}[u] - \dot{q}_V = 0 \sim ?$$



GP Inference with PDE Observations

Prior

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

Observations

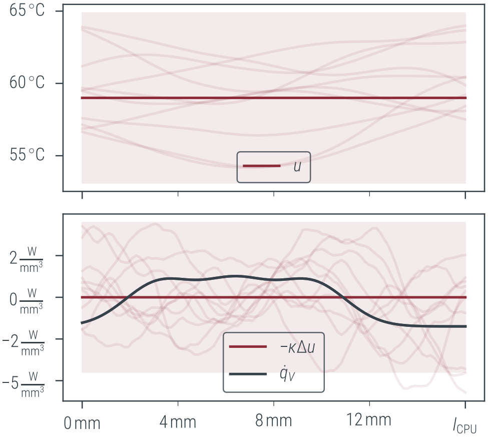
$$\mathcal{I}_{\text{PDE}}[u] = \underbrace{-\kappa \Delta u}_{=:\mathcal{D}[u]} - \dot{q}_V = 0$$

(Prior) Predictive

$$\mathcal{D}[u] \sim \mathcal{GP}$$

Posterior

$$u \mid \mathcal{D}[u] - \dot{q}_V = 0 \sim \mathcal{GP}$$



Prior

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

Observations

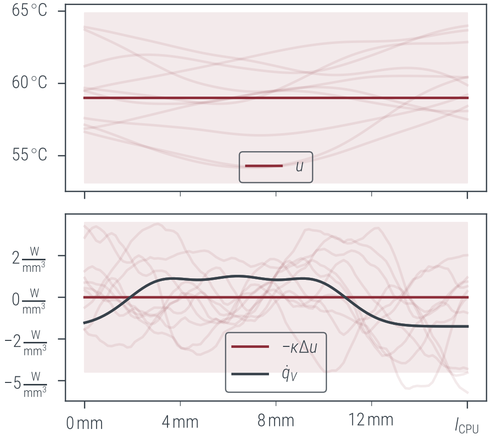
$$\mathcal{I}_{\text{PDE}}[u] = \underbrace{-\kappa \Delta u}_{=:\mathcal{D}[u]} - \dot{q}_V = 0$$

(Prior) Predictive

$$\mathcal{D}[u] \sim \mathcal{GP}$$

Posterior

$$u \mid \mathcal{D}[u](X) - \dot{q}_V(X) = 0 \sim \mathcal{GP}$$



Linear Gaussian Process Inference I

	$\mathbb{R}^d$	
Prior	$x \sim \mathcal{N}(\mu_x, \Sigma_x)$	
Observation Model	$Ax + \epsilon = y$, where ▶ $A \in \mathbb{R}^{n \times d}$ and ▶ $\epsilon \sim \mathcal{N}(\mu_\epsilon, \Sigma_\epsilon)$ with $\epsilon \perp\!\!\!\perp x$.	
Prior Predictive	$Ax + \epsilon \sim \mathcal{N}\left(A\mu_x + \mu_\epsilon, A\Sigma_x A^\top + \Sigma_\epsilon\right)$	
Posterior	$x \mid Ax + \epsilon = y \sim \mathcal{N}(\mu_{x\mid y}, \Sigma_{x\mid y})$	
	$\mu_{x\mid y} := \mu_x + G_{x\mid y}(y - (A\mu_x + \mu_\epsilon))$ $\Sigma_{x\mid y} := \Sigma_x - G_{x\mid y} A \Sigma_x$ $G_{x\mid y} := \Sigma_x A^\top (A \Sigma_x A^\top + \Sigma_\epsilon)^\dagger$	

Linear Gaussian Process Inference I

	$\mathbb{R}^d$	$\mathbb{R}^{\mathcal{X}}$
Prior	$x \sim \mathcal{N}(\mu_x, \Sigma_x)$	$f \sim \mathcal{GP}(m_f, k_f)$
Observation Model	$Ax + \epsilon = y$, where ▶ $A \in \mathbb{R}^{n \times d}$ and ▶ $\epsilon \sim \mathcal{N}(\mu_\epsilon, \Sigma_\epsilon)$ with $\epsilon \perp\!\!\!\perp x$.	
Prior Predictive	$Ax + \epsilon \sim \mathcal{N}\left(A\mu_x + \mu_\epsilon, A\Sigma_x A^\top + \Sigma_\epsilon\right)$	
Posterior	$x \mid Ax + \epsilon = y \sim \mathcal{N}(\mu_{x\mid y}, \Sigma_{x\mid y})$	
	$\mu_{x\mid y} := \mu_x + G_{x\mid y}(y - (A\mu_x + \mu_\epsilon))$ $\Sigma_{x\mid y} := \Sigma_x - G_{x\mid y} A \Sigma_x$ $G_{x\mid y} := \Sigma_x A^\top (A\Sigma_x A^\top + \Sigma_\epsilon)^\dagger$	

Linear Gaussian Process Inference I

	$\mathbb{R}^d$	$\mathbb{R}^{\mathcal{X}}$
Prior	$x \sim \mathcal{N}(\mu_x, \Sigma_x)$	$f \sim \mathcal{GP}(m_f, k_f)$
Observation Model	$Ax + \epsilon = y$, where ▶ $A \in \mathbb{R}^{n \times d}$ and ▶ $\epsilon \sim \mathcal{N}(\mu_\epsilon, \Sigma_\epsilon)$ with $\epsilon \perp\!\!\!\perp x$.	$\mathcal{L}[u] + \epsilon = y$, where ▶ $\mathcal{L}:$ paths$(f) \to \mathbb{R}^n$ is linear and ▶ $\epsilon \sim \mathcal{N}(\mu_\epsilon, \Sigma_\epsilon)$ with $\epsilon \perp\!\!\!\perp f$.
Prior Predictive	$Ax + \epsilon \sim \mathcal{N}\left(A\mu_x + \mu_\epsilon, A\Sigma_x A^\top + \Sigma_\epsilon\right)$	
Posterior	$x \mid Ax + \epsilon = y \sim \mathcal{N}(\mu_{x\|y}, \Sigma_{x\|y})$	
	$\mu_{x\|y} := \mu_x + G_{x\|y}(y - (A\mu_x + \mu_\epsilon))$ $\Sigma_{x\|y} := \Sigma_x - G_{x\|y} A \Sigma_x$ $G_{x\|y} := \Sigma_x A^\top (A\Sigma_x A^\top + \Sigma_\epsilon)^\dagger$	

Linear Gaussian Process Inference I

	$\mathbb{R}^d$	$\mathbb{R}^{\mathcal{X}}$
Prior	$x \sim \mathcal{N}(\mu_x, \Sigma_x)$	$f \sim \mathcal{GP}(m_f, k_f)$
Observation Model	$Ax + \epsilon = y$, where ▶ $A \in \mathbb{R}^{n \times d}$ and ▶ $\epsilon \sim \mathcal{N}(\mu_\epsilon, \Sigma_\epsilon)$ with $\epsilon \perp\!\!\!\perp x$.	$\mathcal{L}[u] + \epsilon = y$, where ▶ $\mathcal{L}:$ paths$(f) \rightarrow \mathbb{R}^n$ is linear and ▶ $\epsilon \sim \mathcal{N}(\mu_\epsilon, \Sigma_\epsilon)$ with $\epsilon \perp\!\!\!\perp f$.
Prior Predictive	$Ax + \epsilon \sim \mathcal{N}\left(A\mu_x + \mu_\epsilon, A\Sigma_x A^\top + \Sigma_\epsilon\right)$	$\mathcal{L}[u] + \epsilon \sim \mathcal{N}(\mathcal{L}[m_f] + \mu_\epsilon, \mathcal{L}k_f\mathcal{L}^* + \Sigma_\epsilon)$, where $(\mathcal{L}k_f\mathcal{L}^*)_{ij} := \mathcal{L}[x \mapsto \mathcal{L}[k(\cdot, x)]_i]_j$
Posterior	$x \mid Ax + \epsilon = y \sim \mathcal{N}(\mu_{x\mid y}, \Sigma_{x\mid y})$	
	$\mu_{x\mid y} := \mu_x + G_{x\mid y}(y - (A\mu_x + \mu_\epsilon))$ $\Sigma_{x\mid y} := \Sigma_x - G_{x\mid y} A \Sigma_x$ $G_{x\mid y} := \Sigma_x A^\top (A\Sigma_x A^\top + \Sigma_\epsilon)^\dagger$	

Linear Gaussian Process Inference I

	$\mathbb{R}^d$	$\mathbb{R}^{\mathcal{X}}$
Prior	$x \sim \mathcal{N}(\mu_x, \Sigma_x)$	$f \sim \mathcal{GP}(m_f, k_f)$
Observation Model	$Ax + \epsilon = y$, where ▶ $A \in \mathbb{R}^{n \times d}$ and ▶ $\epsilon \sim \mathcal{N}(\mu_\epsilon, \Sigma_\epsilon)$ with $\epsilon \perp\!\!\!\perp x$.	$\mathcal{L}[u] + \epsilon = y$, where ▶ $\mathcal{L}\colon \text{paths}(f) \to \mathbb{R}^n$ is linear and ▶ $\epsilon \sim \mathcal{N}(\mu_\epsilon, \Sigma_\epsilon)$ with $\epsilon \perp\!\!\!\perp f$.
Prior Predictive	$Ax + \epsilon \sim \mathcal{N}(A\mu_x + \mu_\epsilon, A\Sigma_x A^\top + \Sigma_\epsilon)$	$\mathcal{L}[u] + \epsilon \sim \mathcal{N}(\mathcal{L}[m_f] + \mu_\epsilon, \mathcal{L}k_f\mathcal{L}^* + \Sigma_\epsilon)$, where $(\mathcal{L}k_f\mathcal{L}^*)_{ij} := \mathcal{L}[x \mapsto \mathcal{L}[k(\cdot, x)]_i]_j$
Posterior	$x \mid Ax + \epsilon = y \sim \mathcal{N}(\mu_{x\mid y}, \Sigma_{x\mid y})$	$f \mid \mathcal{L}[f] + \epsilon = y \sim \mathcal{GP}(m_{f\mid y}, k_{f\mid y})$
	$\mu_{x\mid y} := \mu_x + G_{x\mid y}(y - (A\mu_x + \mu_\epsilon))$ $\Sigma_{x\mid y} := \Sigma_x - G_{x\mid y}A\Sigma_x$ $G_{x\mid y} := \Sigma_x A^\top (A\Sigma_x A^\top + \Sigma_\epsilon)^\dagger$	

Linear Gaussian Process Inference I

	$\mathbb{R}^d$	$\mathbb{R}^{\mathcal{X}}$
Prior	$x \sim \mathcal{N}(\mu_x, \Sigma_x)$	$f \sim \mathcal{GP}(m_f, k_f)$
Observation Model	$Ax + \epsilon = y$, where ▶ $A \in \mathbb{R}^{n \times d}$ and ▶ $\epsilon \sim \mathcal{N}(\mu_\epsilon, \Sigma_\epsilon)$ with $\epsilon \perp\!\!\!\perp x$.	$\mathcal{L}[u] + \epsilon = y$, where ▶ $\mathcal{L}:$ paths$(f) \rightarrow \mathbb{R}^n$ is linear and ▶ $\epsilon \sim \mathcal{N}(\mu_\epsilon, \Sigma_\epsilon)$ with $\epsilon \perp\!\!\!\perp f$.
Prior Predictive	$Ax + \epsilon \sim \mathcal{N}(A\mu_x + \mu_\epsilon, A\Sigma_x A^\top + \Sigma_\epsilon)$	$\mathcal{L}[u] + \epsilon \sim \mathcal{N}(\mathcal{L}[m_f] + \mu_\epsilon, \mathcal{L}k_f\mathcal{L}^* + \Sigma_\epsilon)$, where $(\mathcal{L}k_f\mathcal{L}^*)_{ij} := \mathcal{L}[x \mapsto \mathcal{L}[k(\cdot, x)]_i]_j$
Posterior	$x \mid Ax + \epsilon = y \sim \mathcal{N}(\mu_{x\mid y}, \Sigma_{x\mid y})$	$f \mid \mathcal{L}[f] + \epsilon = y \sim \mathcal{GP}(m_{f\mid y}, k_{f\mid y})$
	$\mu_{x\mid y} := \mu_x + G_{x\mid y}(y - (A\mu_x + \mu_\epsilon))$ $\Sigma_{x\mid y} := \Sigma_x - G_{x\mid y}A\Sigma_x$ $G_{x\mid y} := \Sigma_x A^\top (A\Sigma_x A^\top + \Sigma_\epsilon)^\dagger$	$m_{f\mid y}(x) := m_f(x) + G_{f\mid y}(x)(y - (\mathcal{L}[\mu_x] + \mu_\epsilon))$ $k_{f\mid y}(x_1, x_2) := k_f(x_1, x_2) - G_{f\mid y}(x)\mathcal{L}[k_f(\cdot, x_2)]$ $G_{f\mid y}(x) := \mathcal{L}[k_f(\cdot, x)]^\top (\mathcal{L}k_f\mathcal{L}^* + \Sigma_\epsilon)^\dagger$

Linear Gaussian Process Inference II

- example: derivative of a GP at a point $x \in \mathcal{X} \subset \mathbb{R}^d$

$$\mathcal{L}: \text{ paths}(f) \to \mathbb{R}, h \mapsto \left.\frac{\mathrm{d}h(t)}{\mathrm{d}t}\right|_{t=x}$$

then $\mathcal{L}k_f\mathcal{L}^* \in \mathbb{R}$ and

$$\mathcal{L}k_f\mathcal{L}^* = \mathcal{L}\left[t_2 \mapsto \mathcal{L}\left[k(\cdot, t_2)\right]\right] =$$

- example: derivative of a GP at a point $x \in \mathcal{X} \subset \mathbb{R}^d$

$$\mathcal{L}: \text{ paths}(f) \to \mathbb{R}, h \mapsto \left. \frac{\mathrm{d}h(t)}{\mathrm{d}t} \right|_{t=x}$$

then $\mathcal{L} k_f \mathcal{L}^* \in \mathbb{R}$ and

$$\mathcal{L} k_f \mathcal{L}^* = \mathcal{L}\left[t_2 \mapsto \mathcal{L}\left[k(\cdot, t_2)\right]\right] = \mathcal{L}\left[t_2 \mapsto \left. \frac{\mathrm{d}k(t_1, t_2)}{\mathrm{d}t_1} \right|_{t_1=x}\right]$$

- example: derivative of a GP at a point $x \in \mathcal{X} \subset \mathbb{R}^d$

$$\mathcal{L}: \text{ paths}(f) \rightarrow \mathbb{R}, h \mapsto \left.\frac{\mathrm{d}h(t)}{\mathrm{d}t}\right|_{t=x}$$

then $\mathcal{L}k_f\mathcal{L}^* \in \mathbb{R}$ and

$$\mathcal{L}k_f\mathcal{L}^* = \mathcal{L}\left[t_2 \mapsto \mathcal{L}\left[k(\cdot, t_2)\right]\right] = \mathcal{L}\left[t_2 \mapsto \left.\frac{\mathrm{d}k(t_1, t_2)}{\mathrm{d}t_1}\right|_{t_1=x}\right] = \left.\frac{\partial^2 k_f(t_1, t_2)}{\partial t_2 \partial t_1}\right|_{(t_1,t_2)=(x,x)}$$

Linear Gaussian Process Inference II

- example: derivative of a GP at a point $x \in \mathcal{X} \subset \mathbb{R}^d$

$$\mathcal{L}: \text{paths}(f) \to \mathbb{R}, h \mapsto \left.\frac{dh(t)}{dt}\right|_{t=x}$$

then $\mathcal{L}k_f\mathcal{L}^* \in \mathbb{R}$ and

$$\mathcal{L}k_f\mathcal{L}^* = \mathcal{L}\left[t_2 \mapsto \mathcal{L}\left[k(\cdot, t_2)\right]\right] = \mathcal{L}\left[t_2 \mapsto \left.\frac{dk(t_1, t_2)}{dt_1}\right|_{t_1=x}\right] = \left.\frac{\partial^2 k_f(t_1, t_2)}{\partial t_2 \partial t_1}\right|_{(t_1,t_2)=(x,x)}$$

- for $\mathcal{L}: \text{paths}(f) \to \mathbb{R}^{\mathcal{X}'}$, we also define

$$(\mathcal{L}k_f)(x_1', x_2) := \mathcal{L}\left[k_f(\cdot, x_2)\right](x_1')$$
$$(k_f\mathcal{L}^*)(x_1, x_2') := \mathcal{L}\left[k_f(x_1, \cdot)\right](x_2')$$
$$(\mathcal{L}k_f\mathcal{L}^*)(x_1', x_2') := \mathcal{L}\left[(\mathcal{L}k_f)(x_1', \cdot)\right](x_2')$$

- example: derivative of a GP at a point $x \in \mathcal{X} \subset \mathbb{R}^d$

$$\mathcal{L}\colon \text{paths}(f) \to \mathbb{R}, h \mapsto \left.\frac{\mathrm{d}h(t)}{\mathrm{d}t}\right|_{t=x}$$

then $\mathcal{L}k_f\mathcal{L}^* \in \mathbb{R}$ and

$$\mathcal{L}k_f\mathcal{L}^* = \mathcal{L}\left[t_2 \mapsto \mathcal{L}\left[k(\cdot, t_2)\right]\right] = \mathcal{L}\left[t_2 \mapsto \left.\frac{\mathrm{d}k(t_1, t_2)}{\mathrm{d}t_1}\right|_{t_1=x}\right] = \left.\frac{\partial^2 k_f(t_1, t_2)}{\partial t_2 \partial t_1}\right|_{(t_1,t_2)=(x,x)}$$

- for $\mathcal{L}\colon \text{paths}(f) \to \mathbb{R}^{\mathcal{X}'}$, we also define

$$(\mathcal{L}k_f)(x_1', x_2) := \mathcal{L}\left[k_f(\cdot, x_2)\right](x_1')$$
$$(k_f\mathcal{L}^*)(x_1, x_2') := \mathcal{L}\left[k_f(x_1, \cdot)\right](x_2')$$
$$(\mathcal{L}k_f\mathcal{L}^*)(x_1', x_2') := \mathcal{L}\left[(\mathcal{L}k_f)(x_1', \cdot)\right](x_2')$$

$$\Rightarrow \mathcal{L}[f] \sim \mathcal{GP}\left(\mathcal{L}[m], \mathcal{L}k_f\mathcal{L}^*\right)$$

Prior

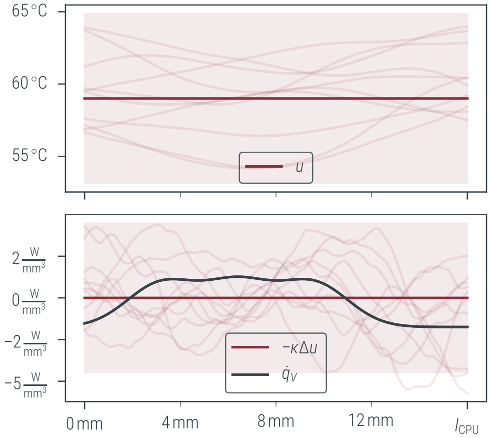
$$u \sim \mathcal{GP}(m, k)$$

Observations

$$\mathcal{I}_{\text{PDE}}[u] = \underbrace{-\kappa \Delta u}_{=:\mathcal{D}[u]} - \dot{q}_V = 0$$

(Prior) Predictive

$$\mathcal{D}[u] \sim \mathcal{GP}$$



Prior

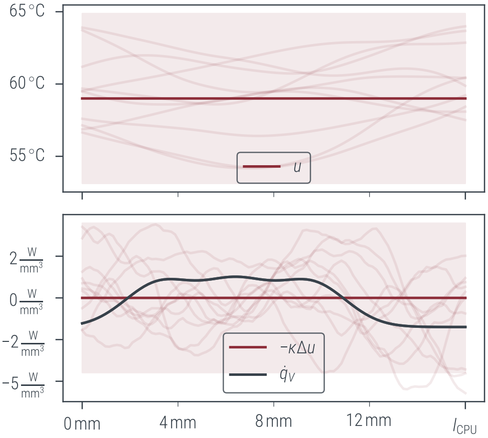
$$u \sim \mathcal{GP}(m, k)$$

Observations

$$\mathcal{I}_{\text{PDE}}[u] = \underbrace{-\kappa \Delta u}_{=:\mathcal{D}[u]} - \dot{q}_V = 0$$

(Prior) Predictive

$$\mathcal{D}[u] \sim \mathcal{GP}(\mathcal{D}[m], \mathcal{D}k\mathcal{D}^*)$$



Prior

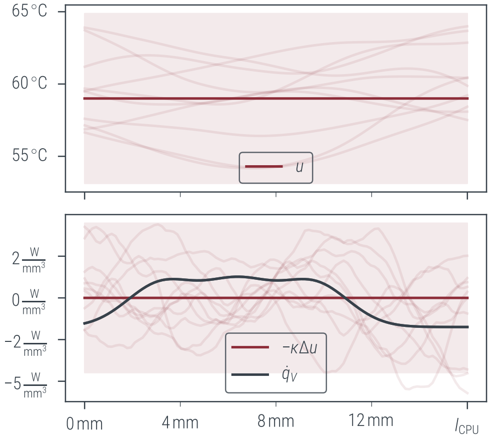
$$u \sim \mathcal{GP}(m, k)$$

Observations

$$\mathcal{I}_{\mathrm{PDE}}[u] = \underbrace{-\kappa \Delta u}_{=:\mathcal{D}[u]} - \dot{q}_V = 0$$

Posterior

$$u \mid \mathcal{D}[u](X) - \dot{q}_V(X) = 0 \sim \mathcal{GP}$$



Prior

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

Observations

$$\mathcal{I}_{\mathrm{PDE}}[u] = \underbrace{-\kappa \Delta u}_{=:\mathcal{D}[u]} - \dot{q}_V = 0$$

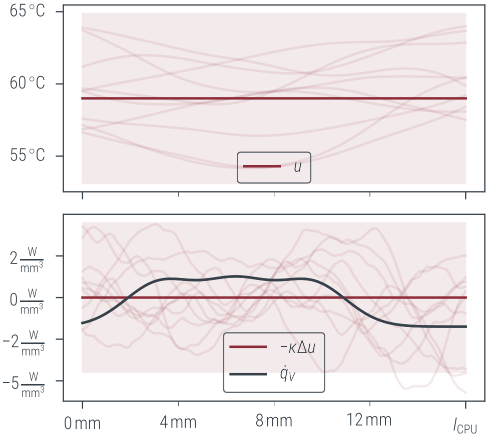
Posterior

$$u \mid \mathcal{D}[u](X) - \dot{q}_V(X) = 0 \sim \mathcal{GP}(m_{\mathrm{PDE}}, k_{\mathrm{PDE}})$$

$$m_{\mathrm{PDE}}(x) = m(x) + G_{\mathrm{PDE}}(x)(\dot{q}_V(X) - \mathcal{D}[m](X))$$

$$k_{\mathrm{PDE}}(x_1, x_2) = k(x_1, x_2) - G_{\mathrm{PDE}}(x_1)(\mathcal{D}k)(X, x_2)$$

$$G_{\mathrm{PDE}}(x) := (k\mathcal{D}^*)(x, X)(\mathcal{D}k\mathcal{D}^*)(X, X)^\dagger$$



GP Inference with PDE Observations

Prior

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

Observations

$$\mathcal{I}_{\text{PDE}}[u] = \underbrace{-\kappa \Delta u}_{=:\mathcal{D}[u]} - \dot{q}_V = 0$$

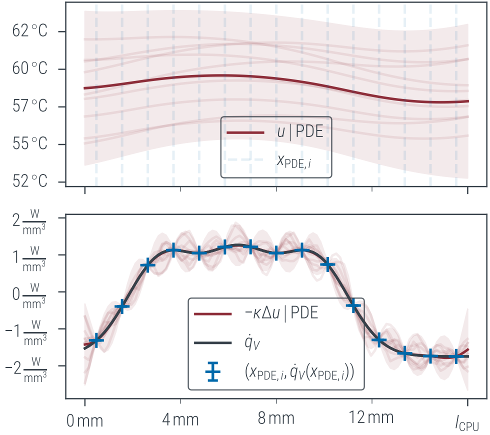
Posterior

$$u \mid \mathcal{D}[u](X) - \dot{q}_V(X) = 0 \sim \mathcal{GP}(m_{\text{PDE}}, k_{\text{PDE}})$$

$$m_{\text{PDE}}(x) = m(x) + G_{\text{PDE}}(x)(\dot{q}_V(X) - \mathcal{D}[m](X))$$

$$k_{\text{PDE}}(x_1, x_2) = k(x_1, x_2) - G_{\text{PDE}}(x_1)(\mathcal{D}k)(X, x_2)$$

$$G_{\text{PDE}}(x) := (k\mathcal{D}^*)(x, X)(\mathcal{D}k\mathcal{D}^*)(X, X)^\dagger$$



GP Inference with PDE and Boundary Observations

Probabilistic Symmetric RKHS Collocation for the Dirichlet Problem

Cockayne et al. [2017]

Prior

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

Observations

$$\mathcal{I}_{\mathrm{PDE}}[u] = -\kappa \Delta u - \dot{q}_V = 0$$
$$\mathcal{I}_{\mathrm{BC}}[u] = u|_{\partial D} - g = 0$$

Posterior

$$u \mid \begin{array}{l} -\kappa \Delta u(X_{\mathrm{PDE}}) - \dot{q}_V(X_{\mathrm{PDE}}) = 0, \\ u(X_{\mathrm{BC}}) - g(X_{\mathrm{BC}}) = 0 \end{array} \sim \mathcal{GP}$$

Prior

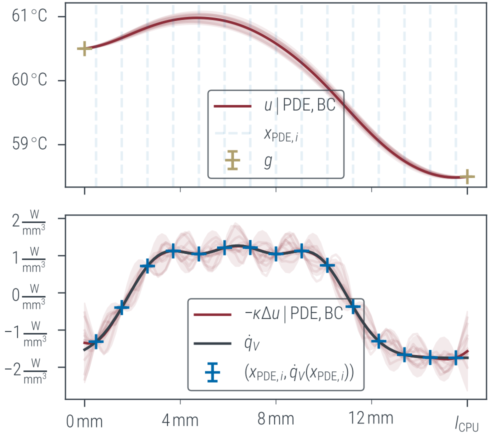
$$u \sim \mathcal{GP}(m, k)$$

Observations

$$\mathcal{I}_{\text{PDE}}[u] = -\kappa \Delta u - \dot{q}_V = 0$$
$$\mathcal{I}_{\text{BC}}[u] = u|_{\partial D} - g = 0$$

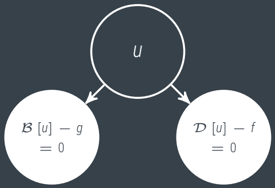
Posterior

$$u \mid \begin{array}{l} -\kappa \Delta u(X_{\text{PDE}}) - \dot{q}_V(X_{\text{PDE}}) = 0, \\ u(X_{\text{BC}}) - g(X_{\text{BC}}) = 0 \end{array} \sim \mathcal{GP}$$



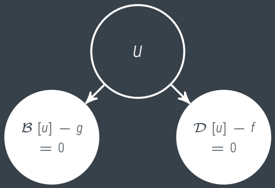
We have seen that GP inference can produce

- an approximate solution of the BVP, and



We have seen that GP inference can produce

- an approximate solution of the BVP, and
- an estimate of the approximation error.

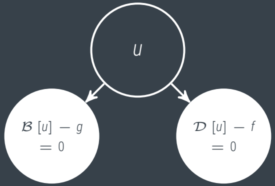


We have seen that GP inference can produce

- an approximate solution of the BVP, and
- an estimate of the approximation error.

Unfortunately,

- the boundary values are unknown in deployment, and

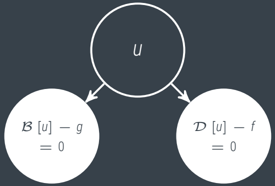


We have seen that GP inference can produce

- an approximate solution of the BVP, and
- an estimate of the approximation error.

Unfortunately,

- the boundary values are unknown in deployment, and
- the values of the heat source distribution are uncertain.



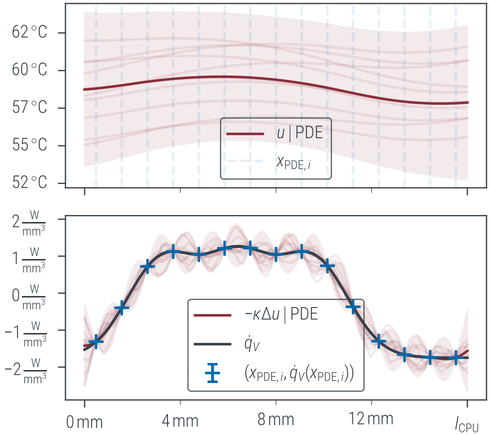
EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Prior

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

Information Operators

$$\mathcal{I}_{\mathrm{PDE}}[u, \dot{q}_V] = -\kappa \Delta u\,(X_{\mathrm{PDE}}) - \dot{q}_V(X_{\mathrm{PDE}}) = 0$$



Prior

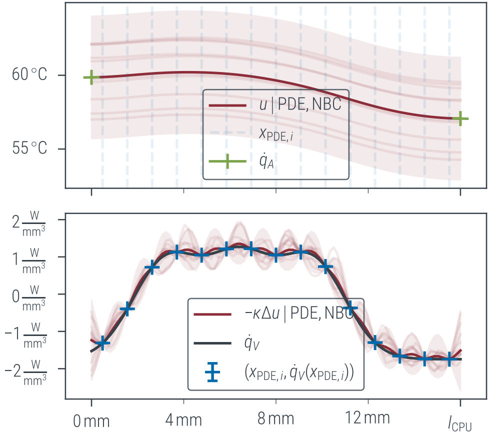
$$u \sim \mathcal{GP}(m, k)$$

$$\dot{q}_A \sim \mathcal{GP}(m_{\dot{q}_A}, k_{\dot{q}_A})$$

Information Operators

$$\mathcal{I}_{\text{PDE}}[u, \dot{q}_V] = -\kappa \Delta u(X_{\text{PDE}}) - \dot{q}_V(X_{\text{PDE}}) = 0$$

$$\mathcal{I}_{\text{BC}}[u, \dot{q}_A] = -\kappa \partial_{\nu(X_{\text{BC}})} u(X_{\text{BC}}) - \dot{q}_A(X_{\text{BC}}) = 0$$



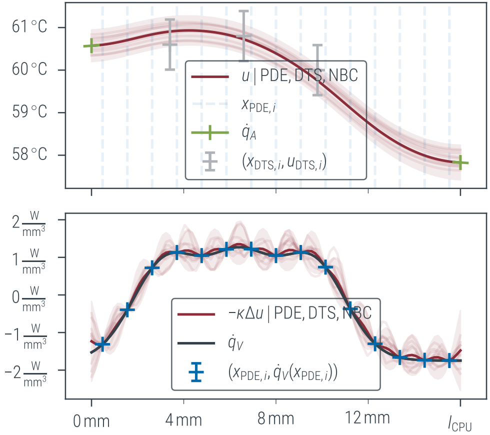
Prior

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

$$\dot{q}_A \sim \mathcal{GP}(m_{\dot{q}_A}, k_{\dot{q}_A})$$
$$\epsilon_{\text{DTS}} \sim \mathcal{N}(0, \Sigma_{\text{DTS}})$$

Information Operators

$$\mathcal{I}_{\text{PDE}}[u, \dot{q}_V] = -\kappa \Delta u(X_{\text{PDE}}) - \dot{q}_V(X_{\text{PDE}}) = 0$$
$$\mathcal{I}_{\text{BC}}[u, \dot{q}_A] = -\kappa \partial_{\nu(X_{\text{BC}})} u(X_{\text{BC}}) - \dot{q}_A(X_{\text{BC}}) = 0$$
$$\mathcal{I}_{\text{DTS}}[u, \epsilon_{\text{DTS}}] = u(X_{\text{DTS}}) + \epsilon_{\text{DTS}} = u_{\text{DTS}}$$



Prior

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

$$\dot{q}_V \sim \mathcal{GP}(m_{\dot{q}_V}, k_{\dot{q}_V})$$

$$\dot{q}_A \sim \mathcal{GP}(m_{\dot{q}_A}, k_{\dot{q}_A})$$

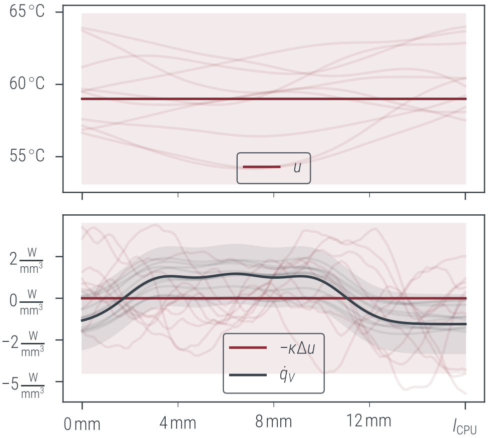
$$\epsilon_{\text{DTS}} \sim \mathcal{N}(0, \Sigma_{\text{DTS}})$$

Information Operators

$$\mathcal{I}_{\text{PDE}}[u, \dot{q}_V] = -\kappa \Delta u(X_{\text{PDE}}) - \dot{q}_V(X_{\text{PDE}}) = 0$$

$$\mathcal{I}_{\text{BC}}[u, \dot{q}_A] = -\kappa \partial_{\nu(X_{\text{BC}})} u(X_{\text{BC}}) - \dot{q}_A(X_{\text{BC}}) = 0$$

$$\mathcal{I}_{\text{DTS}}[u, \epsilon_{\text{DTS}}] = u(X_{\text{DTS}}) + \epsilon_{\text{DTS}} = u_{\text{DTS}}$$

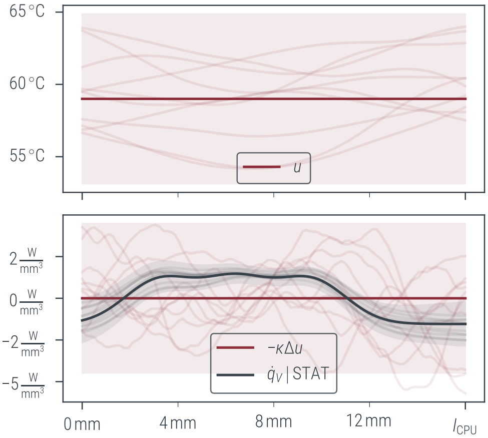


Prior

$$u \sim \mathcal{GP}(m, k)$$
$$\dot{q}_V \sim \mathcal{GP}(m_{\dot{q}_V}, k_{\dot{q}_V})$$
$$\dot{q}_A \sim \mathcal{GP}(m_{\dot{q}_A}, k_{\dot{q}_A})$$
$$\epsilon_{\text{DTS}} \sim \mathcal{N}(0, \Sigma_{\text{DTS}})$$

Information Operators

$$\mathcal{I}_{\text{STAT}}[\dot{q}_V, \dot{q}_A] = d_{\text{CPU}} \int_D \dot{q}_V \, \mathrm{d}x - \int_{\partial D} \dot{q}_A \, \mathrm{d}S = 0$$
$$\mathcal{I}_{\text{PDE}}[u, \dot{q}_V] = -\kappa \Delta u(X_{\text{PDE}}) - \dot{q}_V(X_{\text{PDE}}) = 0$$
$$\mathcal{I}_{\text{BC}}[u, \dot{q}_A] = -\kappa \partial_{\nu(X_{\text{BC}})} u(X_{\text{BC}}) - \dot{q}_A(X_{\text{BC}}) = 0$$
$$\mathcal{I}_{\text{DTS}}[u, \epsilon_{\text{DTS}}] = u(X_{\text{DTS}}) + \epsilon_{\text{DTS}} = u_{\text{DTS}}$$



Uncertainty from Observational Data

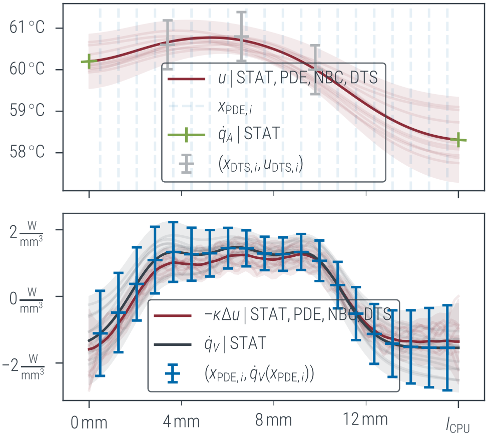
Uncertain Right-Hand Side

Prior

$$u \sim \mathcal{GP}(m, k)$$
$$\dot{q}_V \sim \mathcal{GP}(m_{\dot{q}_V}, k_{\dot{q}_V})$$
$$\dot{q}_A \sim \mathcal{GP}(m_{\dot{q}_A}, k_{\dot{q}_A})$$
$$\epsilon_{\text{DTS}} \sim \mathcal{N}(0, \Sigma_{\text{DTS}})$$

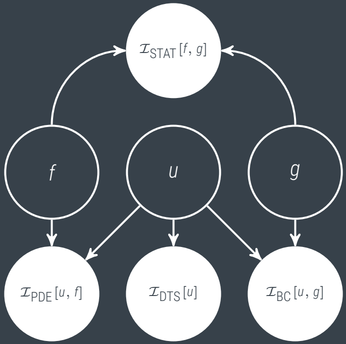
Information Operators

$$\mathcal{I}_{\text{STAT}}[\dot{q}_V, \dot{q}_A] = d_{\text{CPU}} \int_D \dot{q}_V \, dx - \int_{\partial D} \dot{q}_A \, dS = 0$$
$$\mathcal{I}_{\text{PDE}}[u, \dot{q}_V] = -\kappa \Delta u(X_{\text{PDE}}) - \dot{q}_V(X_{\text{PDE}}) = 0$$
$$\mathcal{I}_{\text{BC}}[u, \dot{q}_A] = -\kappa \partial_{\nu(X_{\text{BC}})} u(X_{\text{BC}}) - \dot{q}_A(X_{\text{BC}}) = 0$$
$$\mathcal{I}_{\text{DTS}}[u, \epsilon_{\text{DTS}}] = u(X_{\text{DTS}}) + \epsilon_{\text{DTS}} = u_{\text{DTS}}$$



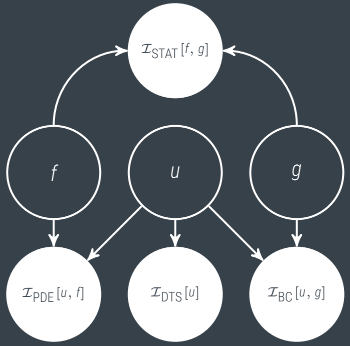
The GP approach integrates enables integration of

- **prior knowledge** about the solution,



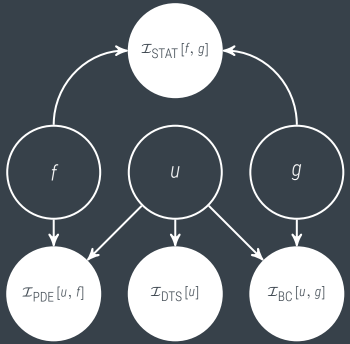
The GP approach integrates enables integration of

- **prior knowledge** about the solution,
- **mechanistic knowledge** in the form of a linear PDE,



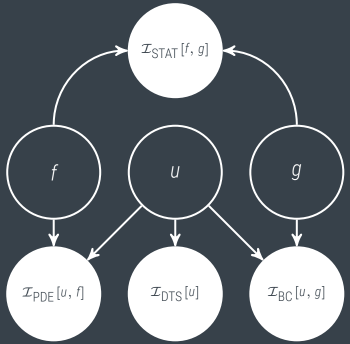
The GP approach integrates enables integration of

- **prior knowledge** about the solution,
- **mechanistic knowledge** in the form of a linear PDE,
- **uncertain** boundary conditions and right-hand sides, and



The GP approach integrates enables integration of

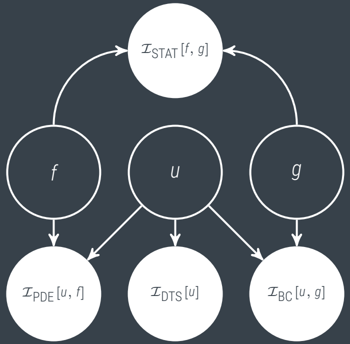
- **prior knowledge** about the solution,
- **mechanistic knowledge** in the form of a linear PDE,
- **uncertain** boundary conditions and right-hand sides, and
- noisy empirical **measurements**,



The GP approach integrates enables integration of

- **prior knowledge** about the solution,
- **mechanistic knowledge** in the form of a linear PDE,
- **uncertain** boundary conditions and right-hand sides, and
- noisy empirical **measurements**,

all while providing

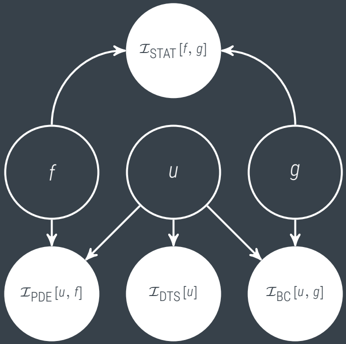


The GP approach integrates enables integration of

- **prior knowledge** about the solution,
- **mechanistic knowledge** in the form of a linear PDE,
- **uncertain** boundary conditions and right-hand sides, and
- noisy empirical **measurements**,

all while providing

- quantification of approximation error,

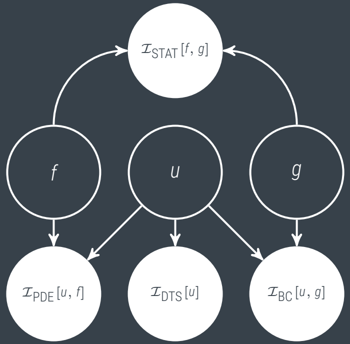


The GP approach integrates enables integration of

- **prior knowledge** about the solution,
- **mechanistic knowledge** in the form of a linear PDE,
- **uncertain** boundary conditions and right-hand sides, and
- noisy empirical **measurements**,

all while providing

- quantification of approximation error,
- error propagation from uncertain system parameters.

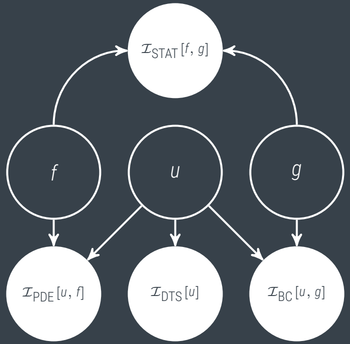


The GP approach integrates enables integration of

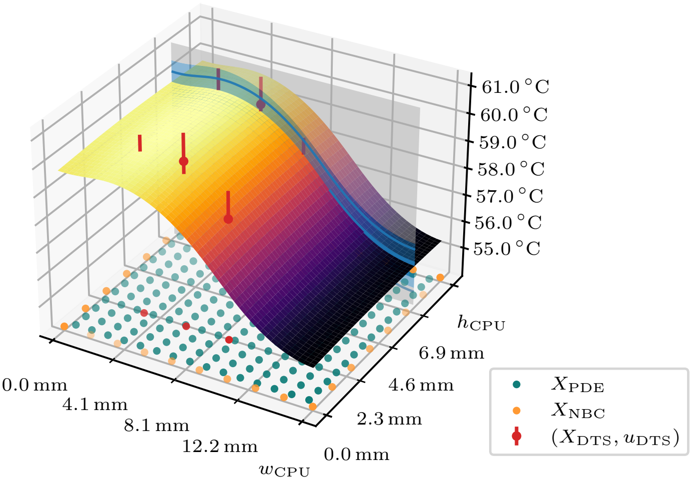
- **prior knowledge** about the solution,
- **mechanistic knowledge** in the form of a linear PDE,
- **uncertain** boundary conditions and right-hand sides, and
- noisy empirical **measurements**,

all while providing

- quantification of approximation error,
- error propagation from uncertain system parameters.

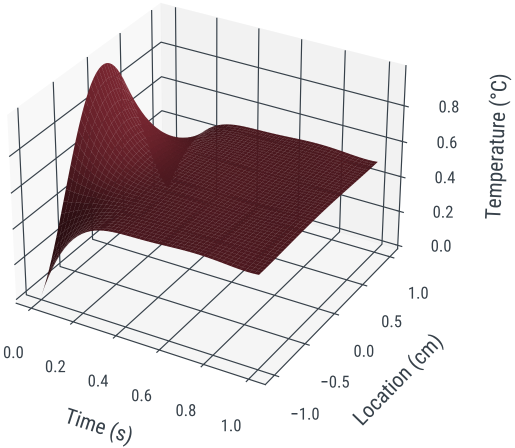


All this is only possible because we give up on trying to identify a single unique solution in favor of a probability measure over *infinitely many solution candidates*.

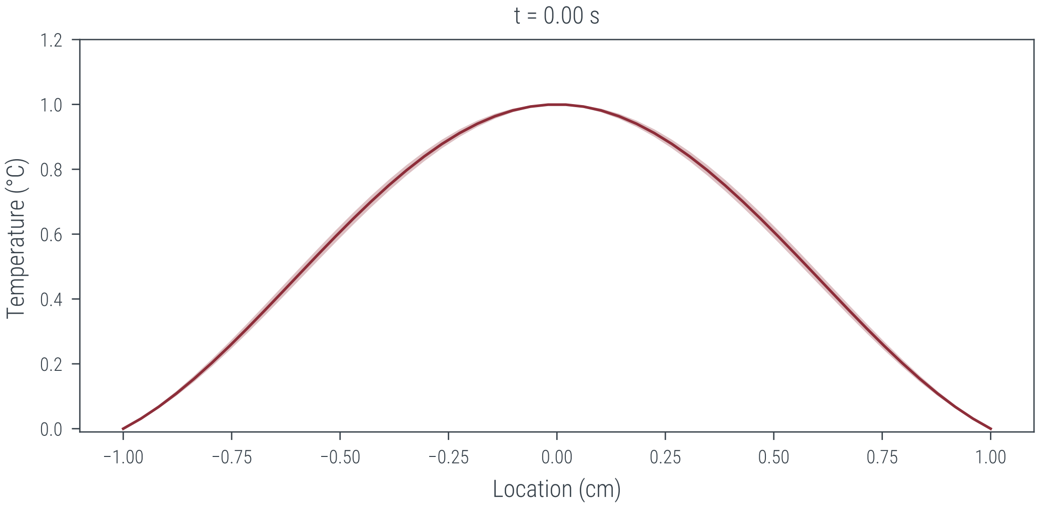


61.0 °C
60.0 °C
59.0 °C
58.0 °C
57.0 °C
56.0 °C
55.0 °C
h_{CPU}
6.9 mm
4.6 mm
2.3 mm
0.0 mm
0.0 mm
4.1 mm
8.1 mm
12.2 mm
w_{CPU}
X_{PDE}
X_{NBC}
$(X_{\mathrm{DTS}}, u_{\mathrm{DTS}})$

GP-based Simulation of the 1D Heat Equation



GP-based Simulation of the 1D Heat Equation



Relationship to Classical Methods

Weighted Residual Methods Through the Lens of GP Inference

- What about the vast number of classical numerical methods for (linear) PDEs developed over the past century?

Relationship to Classical Methods

Weighted Residual Methods Through the Lens of GP Inference

- What about the vast number of classical numerical methods for (linear) PDEs developed over the past century?
- the posterior mean of the GP-based method we have seen today is equal to the estimate produced by a classical method: symmetric collocation [Fasshauer, 1997]

Relationship to Classical Methods

Weighted Residual Methods Through the Lens of GP Inference

- What about the vast number of classical numerical methods for (linear) PDEs developed over the past century?
- the posterior mean of the GP-based method we have seen today is equal to the estimate produced by a classical method: symmetric collocation [Fasshauer, 1997]
- more generally, one can show that all weighted residual methods [Fletcher, 1984] can be realized as posterior means of GPs in a similar fashion
  - parametric and nonparametric collocation methods

Relationship to Classical Methods

Weighted Residual Methods Through the Lens of GP Inference

- What about the vast number of classical numerical methods for (linear) PDEs developed over the past century?
- the posterior mean of the GP-based method we have seen today is equal to the estimate produced by a classical method: symmetric collocation [Fasshauer, 1997]
- more generally, one can show that all weighted residual methods [Fletcher, 1984] can be realized as posterior means of GPs in a similar fashion
  - parametric and nonparametric collocation methods
  - finite-volume methods

Relationship to Classical Methods

Weighted Residual Methods Through the Lens of GP Inference

- What about the vast number of classical numerical methods for (linear) PDEs developed over the past century?
- the posterior mean of the GP-based method we have seen today is equal to the estimate produced by a classical method: symmetric collocation [Fasshauer, 1997]
- more generally, one can show that all weighted residual methods [Fletcher, 1984] can be realized as posterior means of GPs in a similar fashion
  - parametric and nonparametric collocation methods
  - finite-volume methods
  - (Petrov-)Galerkin methods
    - finite-element methods
    - spectral methods

Relationship to Classical Methods

Weighted Residual Methods Through the Lens of GP Inference

- What about the vast number of classical numerical methods for (linear) PDEs developed over the past century?
- the posterior mean of the GP-based method we have seen today is equal to the estimate produced by a classical method: symmetric collocation [Fasshauer, 1997]
- more generally, one can show that all weighted residual methods [Fletcher, 1984] can be realized as posterior means of GPs in a similar fashion
  - parametric and nonparametric collocation methods
  - finite-volume methods
  - (Petrov-)Galerkin methods
    - finite-element methods
    - spectral methods
- the finite-difference discretization can also be generalized via GP inference [Krämer et al., 2022]

Relationship to Classical Methods

Weighted Residual Methods Through the Lens of GP Inference

- What about the vast number of classical numerical methods for (linear) PDEs developed over the past century?
- the posterior mean of the GP-based method we have seen today is equal to the estimate produced by a classical method: symmetric collocation [Fasshauer, 1997]
- more generally, one can show that all weighted residual methods [Fletcher, 1984] can be realized as posterior means of GPs in a similar fashion
  - parametric and nonparametric collocation methods
  - finite-volume methods
  - (Petrov-)Galerkin methods
    - finite-element methods
    - spectral methods
- the finite-difference discretization can also be generalized via GP inference [Krämer et al., 2022]

$\Rightarrow$ GP-based approaches as uncertainty-aware drop-in replacements for classical methods

Quick Summary

- We have now seen how GPs can be used to solve linear PDEs.
- More generally, we showed that GPs are an elegant language for regression from heterogeneous information sources provided by linear (or affine) information operators.

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Quick Summary

- We have now seen how GPs can be used to solve linear PDEs.
- More generally, we showed that GPs are an elegant language for regression from heterogeneous information sources provided by linear (or affine) information operators.

But where is all the hard math that was mentioned in the beginning? So far we just needed derivatives and some linear algebra.

The Sharp Bits...

...because all these infinities require some care

[Pförtner et al., 2022] (in preparation)

Theorem (Linear Gaussian Process Inference)

Let $f \sim \mathcal{GP}(m_f, k_f)$ be a Gaussian process with index set $\mathcal{X}$, whose mean function and sample paths lie in a real separable RKHS $\mathcal{H} \supset \mathcal{H}_{k_f}$. Let $\mathcal{L} \colon \mathcal{H} \to \mathbb{R}^n$ be a bounded (i.e. continuous) linear operator. Further, let $\epsilon \sim \mathcal{N}(\mu_\epsilon, \Sigma_\epsilon)$ be an $\mathbb{R}^n$-valued Gaussian random variable with $\epsilon \perp\!\!\!\perp f$. Then

$$\mathcal{L}[f] + \epsilon \sim \mathcal{N}\left(\mathcal{L}[m_f] + \mu_\epsilon, \mathcal{L} k_f \mathcal{L}^* + \Sigma_\epsilon\right)$$

and

$$f \mid \mathcal{L}[f] + \epsilon = y \sim \mathcal{GP}\left(m_{f|y}, k_{f|y}\right)$$

for any $y \in \mathbb{R}^n$ with conditional mean and covariance function given by

$$m_{f|y}(x) = m_f(x) + \left\langle \mathcal{L}[k_f(x, \cdot)], (\mathcal{L} k_f \mathcal{L}^* + \Sigma_\epsilon)^\dagger (y - (\mathcal{L}[m_f] + \mu_\epsilon)) \right\rangle_{\mathbb{R}^n},$$

and

$$k_{f|y}(x_1, x_2) = k_f(x_1, x_2) - \left\langle \mathcal{L}[k_f(x_1, \cdot)], (\mathcal{L} k_f \mathcal{L}^* + \Sigma_\epsilon)^\dagger \mathcal{L}[k_f(\cdot, x_2)] \right\rangle_{\mathbb{R}^n}.$$

Sample Paths and Path Spaces of GPs

Why we use GPs to model unknown functions

Definition (Gaussian Process)

A *Gaussian process* is a family of random variables $\{\omega \mapsto f(x, \omega)\}_{x \in \mathcal{X}}$ on a common Borel probability space $(\Omega, \mathcal{B}(\Omega), \mathrm{P})$ such that every finite combination $f(x_1, \cdot), \dots, f(x_n, \cdot)$ of the random variables follows a multivariate normal distribution.

Definition (Gaussian Process)

A *Gaussian process* is a family of random variables $\{\omega \mapsto f(x, \omega)\}_{x \in \mathcal{X}}$ on a common Borel probability space $(\Omega, \mathcal{B}(\Omega), \mathrm{P})$ such that every finite combination $f(x_1, \cdot), \dots, f(x_n, \cdot)$ of the random variables follows a multivariate normal distribution.

- for a fixed $\omega \in \Omega$, the function $f(\cdot, \omega) \colon \mathcal{X} \to \mathbb{R}, x \mapsto f(x, \omega)$ is called a sample path of the GP

Sample Paths and Path Spaces of GPs

Why we use GPs to model unknown functions

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Definition (Gaussian Process)

A *Gaussian process* is a family of random variables $\{\omega \mapsto f(x, \omega)\}_{x \in \mathcal{X}}$ on a common Borel probability space $(\Omega, \mathcal{B}(\Omega), \mathrm{P})$ such that every finite combination $f(x_1, \cdot), \ldots, f(x_n, \cdot)$ of the random variables follows a multivariate normal distribution.

- for a fixed $\omega \in \Omega$, the function $f(\cdot, \omega) \colon \mathcal{X} \to \mathbb{R}, x \mapsto f(x, \omega)$ is called a sample path of the GP
- the set $\mathrm{paths}(f) := \{f(\cdot, \omega) \colon \omega \in \Omega\}$ containing all possible sample paths is called the path space of the GP

Sample Paths and Path Spaces of GPs

Why we use GPs to model unknown functions

Definition (Gaussian Process)

A *Gaussian process* is a family of random variables $\{\omega \mapsto f(x, \omega)\}_{x \in \mathcal{X}}$ on a common Borel probability space $(\Omega, \mathcal{B}(\Omega), \mathrm{P})$ such that every finite combination $f(x_1, \cdot), \dots, f(x_n, \cdot)$ of the random variables follows a multivariate normal distribution.

- for a fixed $\omega \in \Omega$, the function $f(\cdot, \omega) \colon \mathcal{X} \to \mathbb{R}, x \mapsto f(x, \omega)$ is called a sample path of the GP
- the set $\mathrm{paths}(f) := \{f(\cdot, \omega) \colon \omega \in \Omega\}$ containing all possible sample paths is called the path space of the GP
- for our purposes, we need to know whether the sample paths of the GP are sufficiently differentiable

Sample Paths and Path Spaces of GPs

Why we use GPs to model unknown functions

Definition (Gaussian Process)

A *Gaussian process* is a family of random variables $\{\omega \mapsto f(x, \omega)\}_{x \in \mathcal{X}}$ on a common Borel probability space $(\Omega, \mathcal{B}(\Omega), \mathrm{P})$ such that every finite combination $f(x_1, \cdot), \dots, f(x_n, \cdot)$ of the random variables follows a multivariate normal distribution.

- for a fixed $\omega \in \Omega$, the function $f(\cdot, \omega) \colon \mathcal{X} \to \mathbb{R}, x \mapsto f(x, \omega)$ is called a sample path of the GP
- the set $\mathrm{paths}(f) := \{f(\cdot, \omega) \colon \omega \in \Omega\}$ containing all possible sample paths is called the path space of the GP
- for our purposes, we need to know whether the sample paths of the GP are sufficiently differentiable
- given a GP f and a linear operator $\mathcal{L} \colon \mathrm{paths}(f) \to \mathbb{R}^n$, the notation $\mathcal{L}[f]$ refers to the function

$$\omega \mapsto \mathcal{L}[f(\cdot, \omega)],$$

i.e. the application of $\mathcal{L}$ to the sample paths of

Definition (Gaussian Process)

A *Gaussian process* is a family of random variables $\{\omega \mapsto f(x, \omega)\}_{x \in \mathcal{X}}$ on a common Borel probability space $(\Omega, \mathcal{B}(\Omega), \mathrm{P})$ such that every finite combination $f(x_1, \cdot), \dots, f(x_n, \cdot)$ of the random variables follows a multivariate normal distribution.

- for a fixed $\omega \in \Omega$, the function $f(\cdot, \omega) \colon \mathcal{X} \to \mathbb{R}, x \mapsto f(x, \omega)$ is called a sample path of the GP
- the set $\mathrm{paths}(f) := \{f(\cdot, \omega) \colon \omega \in \Omega\}$ containing all possible sample paths is called the path space of the GP
- for our purposes, we need to know whether the sample paths of the GP are sufficiently differentiable
- given a GP f and a linear operator $\mathcal{L} \colon \mathrm{paths}(f) \to \mathbb{R}^n$, the notation $\mathcal{L}[f]$ refers to the function

  $$\omega \mapsto \mathcal{L}[f(\cdot, \omega)],$$

  i.e. the application of $\mathcal{L}$ to the sample paths of
- to do Bayesian inference, $\omega \mapsto \mathcal{L}[f(\cdot, \omega)]$ must be a random variable, i.e. measurable

Definition (Reproducing Kernel Hilbert Space)

A Hilbert space $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ of real-valued functions on an arbitrary set $\mathcal{X}$ is called a reproducing kernel Hilbert space (RKHS) if there is a function $k \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that

1. $k(x, \cdot) \in \mathcal{H}$ for all $x \in \mathcal{X}$, and
2. for all $f \in \mathcal{H}$ and $x \in \mathcal{X}$

$$f(x) = \langle k(x, \cdot), f \rangle_{\mathcal{H}} \qquad \text{(reproducing property).}$$

The function k is called the reproducing kernel of $\mathcal{H}$.

Matérn Kernels

Controlling Sample Path Differentiability

see [Kanagawa et al., 2018]

$$k_{\nu=p+1/2,l}(x_1, x_2) := \exp\left(-\frac{\sqrt{2p+1}\,|x_1 - x_2|}{l}\right) \frac{p!}{(2p)!} \sum_{i=0}^{p} \frac{(p+i)!}{i!(p-i)!} \left(\frac{2\sqrt{2p+1}\,|x_1 - x_2|}{l}\right)^{p-i}$$

Matérn Kernels

Controlling Sample Path Differentiability

see [Kanagawa et al., 2018]

$$k_{\nu=p+1/2,l}(x_1, x_2) := \exp\left(-\frac{\sqrt{2p+1}\,|x_1 - x_2|}{l}\right) \frac{p!}{(2p)!} \sum_{i=0}^{p} \frac{(p+i)!}{i!(p-i)!} \left(\frac{2\sqrt{2p+1}\,|x_1 - x_2|}{l}\right)^{p-i}$$

- functions in the RKHS of $k_{\nu=p+1/2,l}$ are p-times differentiable

Matérn Kernels

Controlling Sample Path Differentiability

see [Kanagawa et al., 2018]

$$k_{\nu=p+1/2,l}(x_1, x_2) := \exp\left(-\frac{\sqrt{2p+1}\,|x_1 - x_2|}{l}\right) \frac{p!}{(2p)!} \sum_{i=0}^{p} \frac{(p+i)!}{i!(p-i)!} \left(\frac{2\sqrt{2p+1}\,|x_1 - x_2|}{l}\right)^{p-i}$$

- functions in the RKHS of $k_{\nu=p+1/2,l}$ are p-times differentiable
- however, the paths of a GP f with kernel $k_{\nu=p+1/2,l}$ do not lie in this RKHS

Matérn Kernels

Controlling Sample Path Differentiability

see [Kanagawa et al., 2018]

$$k_{\nu=p+1/2,l}(x_1, x_2) := \exp\left(-\frac{\sqrt{2p+1}\,|x_1 - x_2|}{l}\right) \frac{p!}{(2p)!} \sum_{i=0}^{p} \frac{(p+i)!}{i!(p-i)!} \left(\frac{2\sqrt{2p+1}\,|x_1 - x_2|}{l}\right)^{p-i}$$

- functions in the RKHS of $k_{\nu=p+1/2,l}$ are p-times differentiable
- however, the paths of a GP f with kernel $k_{\nu=p+1/2,l}$ do not lie in this RKHS
- the sample paths of f are rougher (informally, they admit strictly less than $p - \frac{d}{2}$ partial derivatives)

Matérn Kernels

Controlling Sample Path Differentiability

see [Kanagawa et al., 2018]

$$k_{\nu=p+1/2,l}(x_1, x_2) := \exp\left(-\frac{\sqrt{2p+1}\,|x_1 - x_2|}{l}\right) \frac{p!}{(2p)!} \sum_{i=0}^{p} \frac{(p+i)!}{i!(p-i)!} \left(\frac{2\sqrt{2p+1}\,|x_1 - x_2|}{l}\right)^{p-i}$$

- functions in the RKHS of $k_{\nu=p+1/2,l}$ are p-times differentiable
- however, the paths of a GP f with kernel $k_{\nu=p+1/2,l}$ do not lie in this RKHS
- the sample paths of f are rougher (informally, they admit strictly less than $p - \frac{d}{2}$ partial derivatives)
- rule of thumb: if partial derivatives of order at most m appear in the equation, choose

$$p = m + \left\lceil \frac{d+1}{2} \right\rceil$$

$\Rightarrow$ differential operator on paths is bounded

Matérn Kernels

Controlling Sample Path Differentiability

see [Kanagawa et al., 2018]

$$k_{\nu=p+1/2,l}(x_1, x_2) := \exp\left(-\frac{\sqrt{2p+1}\,|x_1 - x_2|}{l}\right)\frac{p!}{(2p)!}\sum_{i=0}^{p}\frac{(p+i)!}{i!(p-i)!}\left(\frac{2\sqrt{2p+1}\,|x_1 - x_2|}{l}\right)^{p-i}$$

- functions in the RKHS of $k_{\nu=p+1/2,l}$ are p-times differentiable
- however, the paths of a GP f with kernel $k_{\nu=p+1/2,l}$ do not lie in this RKHS
- the sample paths of f are rougher (informally, they admit strictly less than $p - \frac{d}{2}$ partial derivatives)
- rule of thumb: if partial derivatives of order at most m appear in the equation, choose

$$p = m + \left\lceil \frac{d+1}{2} \right\rceil$$

  $\Rightarrow$ differential operator on paths is bounded
- more flexible choice of d-dimensional kernel via products of 1D Matérns (see exercise sheet)

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Summary

- PDEs are important and powerful language for modeling the real world
- PDEs can be solved via GP inference
- More generally, GPs provide a rigorous framework for probabilistic **inference on functions** with **heterogeneous information sources** provided by affine information operators
- Some mathematical care must be taken so as not to make mistakes in prior construction

Please cite this course, as

```
@techreport{NoML22,
    title = {Numerics of Machine Learning},
    author = {N. Bosch and J. Grosse
    and P. Hennig and A. Kristiadi
    and M. Pförtner and J. Schmidt
    and F. Schneider and L. Tatzel
    and J. Wenger},
    series = {Lecture Notes in Machine Learning},
    year = {2022},
    institution = {Tübingen AI Center},
}
```

References I

Alain Berlinet and Christine Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Springer, first edition, 2004. ISBN 978-1-4613-4792-7. doi: 10.1007/978-1-4419-9096-9.

Jon Cockayne, Chris Oates, Tim Sullivan, and Mark Girolami. Probabilistic numerical methods for PDE-constrained Bayesian inverse problems. In Geert Verdoolaege, editor, *Proceedings of the 36th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, volume 1853 of *AIP Conference Proceedings*, pages 060001–1 – 060001–8, 2017. doi: 10.1063/1.4985359.

Jon Cockayne, Chris J. Oates, T. J. Sullivan, and Mark Girolami. Bayesian probabilistic numerical methods. *SIAM Review*, 61(4):756–789, 2019. doi: 10.1137/17M1139357.

Gregory E. Fasshauer. Solving partial differential equations by collocation with radial basis functions. In Alain Le Méhauté, Christophe Rabut, and Larry L. Schumaker, editors, *Surface Fitting and Multiresolution Methods*, pages 131–138. Vanderbilt University Press, Nashville, TN, 1997. ISBN 9780826512949.

References II

C. A. J. Fletcher. *Computational Galerkin Methods*. Scientific Computation. Springer, Berlin, Heidelberg, 1 edition, 1984. ISBN 978-3-642-85949-6. doi: 10.1007/978-3-642-85949-6.

Mark Girolami, Eky Febrianto, Yin Ge, and Fehmi Cirak. The statistical finite element method (statFEM) for coherent synthesis of observation data and model predictions. *Computer Methods in Applied Mechanics and Engineering*, 275:113533, 2021. doi: 10.1016/j.cma.2020.113533.

Thore Graepel. Solving noisy linear operator equations by Gaussian processes: Application to ordinary and partial differential equations. In *Proceedings of the 20th International Conference on Machine Learning*, pages 234–241. AAAI Press, 2003.

Ranjan Hebbar. *SPEC CPU2017: PERFORMANCE, ENERGY AND EVENT CHARACTERIZATION ON MODERN PROCESSORS*. PhD thesis, 06 2018.

Philipp Hennig, Michael A. Osborne, and Mark Girolami. Probabilistic numerics and uncertainty in computations. *Proceedings of the Royal Society A*, 471(2179), 2015. doi: 10.1098/rspa.2015.0142.

Motonobu Kanagawa, Philipp Hennig, Dino Sejdinovic, and Bharath K. Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv preprint arXiv:1807.02582*, 2018.

References III

Nicholas Krämer, Jonathan Schmidt, and Philipp Hennig. Probabilistic numerical method of lines for time-dependent partial differential equations. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151, pages 625–639. PMLR, 2022. URL `https://proceedings.mlr.press/v151/kramer22a.html`.

John H. Lienhard, IV and John H. Lienhard, V. *A Heat Transfer Textbook*. Phlogiston Press, Cambridge, MA, 5th edition, 2020. URL `http://ahtt.mit.edu`.

Stefania Maniglia and Abdelaziz Rhandi. Gaussian measures on separable Hilbert spaces and applications, January 2004.

Andreas Nylander. *Fabrication and Characterisation of Carbon Nanotube Array Thermal Interface Materials*. PhD thesis, 11 2018.

Chris J. Oates and Tim J. Sullivan. A modern retrospective on probabilistic numerics. *Statistics and Computing*, 29:1335–1351, 2019. doi: 10.1007/s11222-019-09902-z.

References IV

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Machine learning of linear differential equations using Gaussian processes. *Journal of Computational Physics*, 348:683–693, 2017. ISSN 0021-9991. doi: 10.1016/j.jcp.2017.07.050.

Simo Särkkä. Linear operators and stochastic partial differential equations in Gaussian process regression. In Timo Honkela, Włodzisław Duch, Mark Girolami, and Samuel Kaski, editors, *Artificial Neural Networks and Machine Learning – ICANN 2011*, pages 151–158, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. doi: 10.1007/978-3-642-21738-8_20.

Ingo Steinwart. Convergence types and rates in generic Karhunen-Loève expansions with applications to sample path properties. *Potential Analysis*, 51:361–395, 2019. doi: 10.1007/s11118-018-9715-5.

Filip Tronarp, Hans Kersting, Simo Särkkä, and Philipp Hennig. Probabilistic solutions to ordinary differential equations as non-linear Bayesian filtering: A new perspective. *Statistics and Computing*, 29:1297–1315, 2019. doi: 10.1007/s11222-019-09900-1.

Junyang Wang, Jon Cockayne, Oksana Chkrebtii, Tim J. Sullivan, and Chris J. Oates. Bayesian numerical methods for nonlinear partial differential equations. *Statistics and Computing*, 31(55), 2021. doi: 10.1007/s11222-021-10030-w.