Inference of Partial Differential Equations via Constrained Gaussian Processes

Zhaohui Li

Georgia Institute of Technology

February 7, 2023

Joint work with Shihao Yang (Georgia Tech) and C.F. Jeff Wu (Georgia Tech)

Introduction

- Partial differential equations (PDEs) are widely employed to describe the physical and engineering phenomenon.
- Some *parameters*, which are determined by material properties, engineering properties, etc., are very important for prediction of PDE.
- In real world applications, directly measuring of these parameters are sometimes impossible.
- Estimating these parameters from physical experiment data are important task, known as model calibration, inverse problems, etc.
- We propose a new method for PDE parameter inference, called **P**DE-**I**nformed **G**aussian **P**rocess **I**nference (PIGPI).

Motivation Example

- As a motivation example, we consider the long-range infrared light detection and ranging (LIDAR) equation.
- The received signal over time t and range z can be predicted by the PDE solution:

$$\frac{\partial u(t,s)}{\partial t} - \theta_D \frac{\partial^2 u(t,s)}{\partial s^2} - \theta_S \frac{\partial u(t,s)}{\partial s} = \theta_A u(t,s), 0 \le t \le 20, 0 \le s \le 40,$$

  with specified boundary and initial conditions.
- The task is to estimate the parameters $\theta_D, \theta_S, \theta_A$ from the observation data $y(\boldsymbol{x}_i) = u(\boldsymbol{x}_i) + \varepsilon_i, i = 1, \ldots, n$, where $\boldsymbol{x}_i = (t_i, s_i)$, $\varepsilon_i \sim N(0, \sigma_e^2)$ is random error.
- Let $\boldsymbol{\tau} = \{\boldsymbol{x}_i, i = 1, \ldots, n\}$, $y(\boldsymbol{\tau}) = (y(\boldsymbol{x}_i), i = 1, \ldots, n)$.

Problem Formulation

In general, let's start with a semi-linear partial differential equation (PDE):

$$\mathcal{L}_{\boldsymbol{x}}^{\boldsymbol{\theta}} u(\boldsymbol{x}) = f(\boldsymbol{x}, u(\boldsymbol{x}), \boldsymbol{\theta}),$$

where $\boldsymbol{x} = (x_1, \ldots, x_p)$, $\mathcal{L}_{\boldsymbol{x}}^{\boldsymbol{\theta}} u(\boldsymbol{x})$ denotes a *linear* differential operator on $u \in \mathcal{F}$ (some Hilbert space) of order a.
In the motivation example,

- $\mathcal{L}_{\boldsymbol{x}}^{\boldsymbol{\theta}} u(\boldsymbol{x}) = \frac{\partial u(\boldsymbol{x})}{\partial t} - \theta_D \frac{\partial^2 u(\boldsymbol{x})}{\partial s^2} - \theta_S \frac{\partial u(\boldsymbol{x})}{\partial s}$, where $\boldsymbol{x} = (t, s)$.
- $f(\boldsymbol{x}, u(\boldsymbol{x}), \boldsymbol{\theta}) = \theta_A u(\boldsymbol{x})$.

Problem Formulation

In general, we can assume that the ***PDE operator*** has the form

$$\mathcal{L}_{\boldsymbol{x}}^{\boldsymbol{\theta}} u(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}_i \in A} c_i(\boldsymbol{\theta}, \boldsymbol{x}) \frac{\partial^{|\boldsymbol{\alpha}_i|} u(\boldsymbol{x})}{\partial^{\alpha_{i1}} x_1 \cdots \partial^{\alpha_{ip}} x_p},$$

where $\boldsymbol{\alpha}_i = (\alpha_{i1}, \ldots, \alpha_{ip})$, $\alpha_{ij} = 0, 1, 2, \ldots$, and $|\boldsymbol{\alpha}_i| = \sum_{j=1}^{p} \alpha_{ij} > 0$. $A = \{\boldsymbol{\alpha}_i, i = 1, ..., l\}$.
The order of $\mathcal{L}_{\boldsymbol{x}}^{\boldsymbol{\theta}}$ is defined by $a = \max_i \|\boldsymbol{\alpha}_i\|_1$.
In the motivation example,

- $\mathcal{L}_{\boldsymbol{x}}^{\boldsymbol{\theta}} u(\boldsymbol{x}) = \frac{\partial u(\boldsymbol{x})}{\partial t} - \theta_D \frac{\partial^2 u(\boldsymbol{x})}{\partial s^2} - \theta_S \frac{\partial u(\boldsymbol{x})}{\partial s}$.
- $c_1 = 1, c_2 = -\theta_D, c_3 = -\theta_S$, $\boldsymbol{\alpha}_1 = (1, 0), \boldsymbol{\alpha}_2 = (0, 2), \boldsymbol{\alpha}_3 = (0, 1)$.
- $a = 2$, a second order PDE.

Basic Idea of PIGPI

- The task is to estimate the parameters $\boldsymbol{\theta}$ from the observation data $y(\boldsymbol{x}_i) = u(\boldsymbol{x}_i) + \varepsilon_i, i = 1, \ldots, n_{\boldsymbol{\tau}}$. Let $\boldsymbol{\tau} = \{\boldsymbol{x}_i, i = 1, \ldots, n_{\boldsymbol{\tau}}\}$.
- We assign a Gaussian process (GP) prior on $u(\boldsymbol{x})$ denoted by $U(\boldsymbol{x}) \sim \mathrm{GP}(\mu, \sigma^2\mathcal{K}(\cdot,\cdot))$.
- To incorporate PDE constraints into GP prior, define a random variable W quantifying the difference between GP $U(\boldsymbol{x})$ and the PDE structure with given $\boldsymbol{\theta}$, i.e.,

$$W = \sup_{\boldsymbol{x}\in\Omega} \|\mathcal{L}_{\boldsymbol{x}}^{\boldsymbol{\theta}} U(\boldsymbol{x}) - f(\boldsymbol{x}, U(\boldsymbol{x}), \boldsymbol{\theta})\|.$$

- $W \equiv 0$ **if and only if** U **is the solution of PDE with specified parameter** $\boldsymbol{\theta}$**.**

Basic Idea of PIGPI

- However, in reality W is not computable. We approximate W by finite discretization on the set $\boldsymbol{I} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n_I}\} \subset \Omega$ such that $\boldsymbol{\tau} \subset \boldsymbol{I} \subset \Omega$ and similarly define $W_{\boldsymbol{I}}$ as

$$W_{\boldsymbol{I}} = \sup_{\boldsymbol{x} \in \boldsymbol{I}} \|\mathcal{L}_{\boldsymbol{x}}^{\boldsymbol{\theta}} U(\boldsymbol{x}) - f(\boldsymbol{x}, U(\boldsymbol{x}), \boldsymbol{\theta})\|.$$

- When $\boldsymbol{I}$ is dense, $W_{\boldsymbol{I}}$ can well approximate W.
- Will discuss the choice of $\boldsymbol{I}$ later.

Basic Idea of PIGPI

- An important property for Gaussian process:
- If $U(\boldsymbol{x}) \sim \mathrm{GP}(\mu, \sigma^2 \mathcal{K}(\cdot, \cdot))$;
- Given enough order of differentiable to $\mathcal{K}$ ($2a$ order derivative exists).
- Then $\mathcal{L}_{\boldsymbol{x}}^{\boldsymbol{\theta}} U(\boldsymbol{x}) \sim \mathrm{GP}(\mathcal{L}_{\boldsymbol{x}}^{\boldsymbol{\theta}} \mu(\boldsymbol{x}), \mathcal{L}_{\boldsymbol{x}} \mathcal{L}_{\boldsymbol{x}'} \mathcal{K}(\boldsymbol{x}, \boldsymbol{x}'))$.
- Recall

$$W_{\boldsymbol{I}} = \sup_{\boldsymbol{x} \in \boldsymbol{I}} \| \mathcal{L}_{\boldsymbol{x}}^{\boldsymbol{\theta}} U(\boldsymbol{x}) - f(\boldsymbol{x}, U(\boldsymbol{x}), \boldsymbol{\theta}) \|.$$

- $W_{\boldsymbol{I}} = 0 \rightarrow \mathcal{L}_{\boldsymbol{x}}^{\boldsymbol{\theta}} U(\boldsymbol{x}) = f(\boldsymbol{x}, U(\boldsymbol{x}), \boldsymbol{\theta})$;
- $W_{\boldsymbol{I}} = 0 | (U(\boldsymbol{I}) = u(\boldsymbol{I})) \rightarrow \mathcal{L}_{\boldsymbol{x}}^{\boldsymbol{\theta}} U(\boldsymbol{x}) = f(\boldsymbol{x}, u(\boldsymbol{x}), \boldsymbol{\theta})$.

Basic Idea of PIGPI

- By treating $W_{\boldsymbol{I}}$ as an approximation of W and assigning a noniformative prior for $\boldsymbol{\theta}$, Jeffrey's prior on σ_e^2, the posterior is immediately obtained

$$
\begin{aligned}
& p_{\sigma_e^2,\boldsymbol{\Theta},U(\boldsymbol{I})|W_{\boldsymbol{I}},Y(\boldsymbol{\tau})}\left(\sigma_e^2,\boldsymbol{\theta},u(\boldsymbol{I})|W_{\boldsymbol{I}}=0,Y(\boldsymbol{\tau})=y(\boldsymbol{\tau})\right) \\
\propto & P\left(\sigma_e^2,\boldsymbol{\Theta}=\boldsymbol{\theta},U(\boldsymbol{I})=u(\boldsymbol{I}),W_I=0,Y(\boldsymbol{\tau})=y(\boldsymbol{\tau})\right) \\
= & \pi(\sigma_e^2)\times\pi_{\boldsymbol{\Theta}}\left(\boldsymbol{\theta}\right)\times P\left(U(\boldsymbol{I})=u(\boldsymbol{I})|\boldsymbol{\Theta}=\boldsymbol{\theta}\right) \\
& \times P\left(Y(\boldsymbol{\tau})=y(\boldsymbol{\tau})|\sigma_e^2,U(\boldsymbol{I})=u(\boldsymbol{I}),\boldsymbol{\Theta}=\boldsymbol{\theta}\right) \\
& \times P\left(W_{\boldsymbol{I}}=0|Y(\boldsymbol{\tau})=y(\boldsymbol{\tau}),U(\boldsymbol{I})=u(\boldsymbol{I}),\boldsymbol{\Theta}=\boldsymbol{\theta}\right). \\
= & \frac{1}{\sigma_e^2}\pi_{\boldsymbol{\Theta}}\left(\boldsymbol{\theta}\right)\exp\Big\{-\frac{1}{2}\big[n_{\boldsymbol{I}}\log(2\pi)+\log(|C|)+\|u(\boldsymbol{I})-\mu(\boldsymbol{I})\|_{C^{-1}} \\
& +n\log(2\pi)+n\log(\sigma_e^2)+\|u(\boldsymbol{\tau})-y(\boldsymbol{\tau})\|_{\sigma_e^{-2}} \\
& +n_{\boldsymbol{I}}\log(2\pi)+\log|K|+\|f(\boldsymbol{I},u(\boldsymbol{I}),\boldsymbol{\theta})-\mathcal{L}_{\boldsymbol{x}}\mu(\boldsymbol{I})-m\{u(\boldsymbol{I})-\mu(\boldsymbol{I})\}\|_{K^{-1}}\big]\Big\},
\end{aligned}
$$

- Posterior inference for both $\boldsymbol{\theta}$ and $u(\boldsymbol{I})$ can be done by sampling from/optimizing this (unnormalized) posterior density.

Basic Idea of PIGPI

where

$$\begin{cases} C = \mathcal{K}(\boldsymbol{I}, \boldsymbol{I}) \\ m = \mathcal{L}\mathcal{K}(\boldsymbol{I}, \boldsymbol{I})\mathcal{K}(\boldsymbol{I}, \boldsymbol{I})^{-1} \\ K = \mathcal{L}\mathcal{K}\mathcal{L}(\boldsymbol{I}, \boldsymbol{I}) - \mathcal{L}\mathcal{K}(\boldsymbol{I}, \boldsymbol{I})\mathcal{K}(\boldsymbol{I}, \boldsymbol{I})^{-1}\mathcal{K}\mathcal{L}(\boldsymbol{I}, \boldsymbol{I}) \end{cases},$$

- When $\mathcal{L}$ depends on $\boldsymbol{\theta}$, m and K need to be updated when $\boldsymbol{\theta}$ changes.
- Till now, we assume the PDE operator is linear. Although covers a group of nonlinear-PDE cases, parameter inference from complex nonlinear PDEs are very important and challenging.
- To solve these problems, we propose a novel method that
  - can decouple the dependence between parameter $\boldsymbol{\theta}$ and covariance matrix K . Thus K is fixed once $\boldsymbol{I}$ is given, i.e., no need to update when evaluating posterior density.
  - can deal with a wide range of nonlinear PDEs.

Handling Non-linear and Parameter-dependent Operators

To demonstrate, we consider a nonlinear PDE,

$$\frac{\partial u}{\partial t}(\boldsymbol{x}) = \theta_1 u(\boldsymbol{x}) \frac{\partial u}{\partial s}(\boldsymbol{x}) - \theta_2 \frac{\partial^2 u}{\partial s^2}(\boldsymbol{x}).$$

This PDE is called Burger's equation, the PDE operator is $\mathcal{L}u = \frac{\partial u}{\partial t} - \theta_1 u \frac{\partial u}{\partial s} + \theta_2 \frac{\partial^2 u}{\partial s^2}$. It is

- Nonlinear: $u(\boldsymbol{x}) \frac{\partial u}{\partial s}(\boldsymbol{x})$ term;
- Parameter-operator dependent: $-\theta_1 u \frac{\partial u}{\partial s} + \theta_2 \frac{\partial^2 u}{\partial s^2}$ term.

Handling Non-linear and Parameter-dependent Operators

Recall the Burger's equation,

$$\frac{\partial u}{\partial t}(\boldsymbol{x}) = \theta_1 u(\boldsymbol{x}) \frac{\partial u}{\partial s}(\boldsymbol{x}) - \theta_2 \frac{\partial^2 u}{\partial s^2}(\boldsymbol{x}).$$

First, we define an equivalent PDE system,

$$\begin{aligned}
\frac{\partial u_1}{\partial s}(\boldsymbol{x}) &= u_2(\boldsymbol{x}), \\
\frac{\partial u_2}{\partial s}(\boldsymbol{x}) &= u_3(\boldsymbol{x}), \\
\frac{\partial u_1}{\partial t}(\boldsymbol{x}) &= \theta_1 u_1(\boldsymbol{x}) u_2(\boldsymbol{x}) - \theta_2 u_3(\boldsymbol{x}),
\end{aligned}$$

- This system of PDEs is called augmented PDE.
- The augmented PDE system has a *linear, parameter independent* operator.
- PDE is still nonlinear.

Handling Non-linear and Parameter-dependent Operators

In general, we consider the nonlinear PDE of the form

$$\mathcal{L}_{\boldsymbol{x}} u(\boldsymbol{x}) = f(\boldsymbol{x}, u, \boldsymbol{\theta}) \overset{\text{rewrite}}{\Longleftrightarrow} \nabla^{\boldsymbol{\alpha}_1} u = \mathcal{L}_1(\boldsymbol{x}, \boldsymbol{\theta}, u, \nabla^{\boldsymbol{\alpha}_2} u, \ldots, \nabla^{\boldsymbol{\alpha}_l} u) + f(\boldsymbol{x}, u, \boldsymbol{\theta}) \tag{1}$$

where $\mathcal{L}_1$ is a nonlinear function of $(\boldsymbol{x}, \boldsymbol{\theta}, u, \nabla^{\boldsymbol{\alpha}_2} u, \ldots, \nabla^{\boldsymbol{\alpha}_l} u)$, which is the remaining part of $\mathcal{L}_{\boldsymbol{x}}$. $\mathcal{L}_1$ may contains parameter dependent components and nonlinear components. Then one method for defining augmented PDE is to rewrite (1) as

$$\nabla^{\boldsymbol{\alpha}_1} u = \mathcal{L}_1(\boldsymbol{x}, \boldsymbol{\theta}, \underbrace{u}_{u_1}, \underbrace{\nabla^{\boldsymbol{\alpha}_2} u}_{u_2}, \ldots, \underbrace{\nabla^{\boldsymbol{\alpha}_l} u}_{u_l}) + f(\boldsymbol{x}, u, \boldsymbol{\theta}),$$

such that

$$\begin{aligned} \nabla^{\boldsymbol{\alpha}_2} u_1(\boldsymbol{x}) &= u_2(\boldsymbol{x}), \\ &\cdots \\ \nabla^{\boldsymbol{\alpha}_l} u_1(\boldsymbol{x}) &= u_l(\boldsymbol{x}), \\ \nabla^{\boldsymbol{\alpha}_1} u_1(\boldsymbol{x}) &= f(\boldsymbol{x}, u_1(\boldsymbol{x}), \boldsymbol{\theta}) + \mathcal{L}_1(\boldsymbol{x}, \boldsymbol{\theta}, u_1, u_2, \ldots, u_l). \end{aligned} \tag{2}$$

Handling Non-linear and Parameter-dependent Operators

Several properties are helpful for constructing PIGPI method based on augmented PDE:

- It can be shown these two PDE (systems) are equivalent:
  - Classical solution of original PDE can generalize to a classical solution of augmented PDE;
  - Classical solution of augmented PDE is also a classical solution of original PDE;
- The augmented PDE contains no parameters in left hand side, i.e., the PDE operator is independent to parameters;
- PDE operator for augmented PDE is linear operator.

After augmentation, the idea is natural to apply the proposed PIGPI to the augmented multi-variate PDE.

Discussion - Non-uniqueness of Augmentation

Lowest degree of derivative (LDD) principal: It is easy to see that the augmentation is not unique. As an example, consider equation $u_t - \theta_1 u u_x - \theta_2 u_{xx} = 0$, the augmentation can be

$$\begin{aligned} u_t(\boldsymbol{x}) &= \theta_1 u(\boldsymbol{x}) u_1(\boldsymbol{x}) - \theta_2 u_2(\boldsymbol{x}), \\ u_x(\boldsymbol{x}) &= u_1(\boldsymbol{x}), \\ u_{xx}(\boldsymbol{x}) &= u_2(\boldsymbol{x}), \end{aligned}$$

or

$$\begin{aligned} u_t(\boldsymbol{x}) &= \theta_1 u(\boldsymbol{x}) u_1(\boldsymbol{x}) - \theta_2 u_2(\boldsymbol{x}), \\ u_x(\boldsymbol{x}) &= u_1(\boldsymbol{x}), \\ u_{1x}(\boldsymbol{x}) &= u_2(\boldsymbol{x}). \end{aligned}$$

While in this paper, we recommend to use the **second** augmentation. The main reasons are (a). the second augmented PDE system is a 1-order PDE while the first one is 2-order PDE. We prefer to use a lower order PDE. (b). the second augmented PDE system produce simpler covariance matrix K.

Construction of $\boldsymbol{I}$

For many PDE based problems, $\Omega \subset \mathbb{R}^p$, where $p = 2, 3$, or 4. It is vital to chose proper discretization $\boldsymbol{I}$:

- $\boldsymbol{I}$ should be dense in Ω, i.e., $\forall \boldsymbol{x} \in \Omega$, distance between $\boldsymbol{x}$ and $\boldsymbol{I}$ should be as small as possible. $\boldsymbol{I}$ should be space filling.
- From a computational point of view, $|\boldsymbol{I}|$ should be as small as possible.

In practical applications, it is a common situation that the observation data are collected ahead of data analysis. Thus, we assume that $\boldsymbol{\tau}$ is known and fixed. Method for constructing $\boldsymbol{I}$:

0 Construct a (large) candidate point set $\mathcal{D}$ of size N ($N >> n_{\boldsymbol{I}}$);

1 Start with $\boldsymbol{I} = \boldsymbol{\tau}$.

2 For $i = n_{\boldsymbol{\tau}} + 1, \ldots, n_{\boldsymbol{I}}$, repeat

2.1 Find $\boldsymbol{x}_i = \text{argmax}_{\boldsymbol{x} \in \mathcal{D}} d(\boldsymbol{x}, \boldsymbol{I})$;

2.2 $\boldsymbol{I} = \boldsymbol{I} \cup \boldsymbol{x}_i$.

3 Output $\boldsymbol{I}$.

Handling Boundary/Initial Conditions

In particular, we divide the IBCs into two categories, i.e., Dirichlet IBCs and Non-Dirichlet IBCs. Dirichlet IBCs are given by the known value of PDE solution on the specific boundary regions, i.e.,

$$u(\boldsymbol{x}) = b_1(\boldsymbol{x}), \boldsymbol{x} \in \Gamma_1. \tag{3}$$

In many applications, the initial conditions have the same form with Dirichlet boundary condition, i.e., known value of PDE solution at time $t = 0$. The Non-Dirichlet boundary condition can be represented by an differential operator

$$\mathcal{B}_{\boldsymbol{x},\boldsymbol{\theta}} u(\boldsymbol{x}) = b_2(\boldsymbol{x}, u(\boldsymbol{x}), \boldsymbol{\theta}), \boldsymbol{x} \in \Gamma_2, \tag{4}$$

where $\mathcal{B}$ is a differential operator with order $b > 0$, which has the similar form with $\mathcal{L}$ that we defined in previous sections.

Handling Boundary/Initial Conditions

For non-Dirichlet Boundary conditions, we define the comprehensive operator

$$\mathcal{L}_{\boldsymbol{x},\boldsymbol{\theta}} u(\boldsymbol{x}) = \begin{cases} \mathcal{L}_{\boldsymbol{x},\boldsymbol{\theta}} u(\boldsymbol{x}), & \boldsymbol{x} \in \Omega \backslash \Gamma_2 \\ \mathcal{B}_{\boldsymbol{x},\boldsymbol{\theta}} u(\boldsymbol{x}), & \boldsymbol{x} \in \Gamma_2 \end{cases}.$$

Similarly, define the comprehensive source term

$$f(\boldsymbol{x}, u(\boldsymbol{x}), \boldsymbol{\theta}) = \begin{cases} f(\boldsymbol{x}, u(\boldsymbol{x}), \boldsymbol{\theta}), & \boldsymbol{x} \in \Omega \backslash \Gamma_2 \\ b_2(\boldsymbol{x}, u(\boldsymbol{x}), \boldsymbol{\theta}), & \boldsymbol{x} \in \Gamma_2 \end{cases}.$$

For non-Dirichlet boundary condition, it is natural to incorporate the boundary information by including several proper chosen boundary points and replacing $\mathcal{L}$ and f with their comprehensive forms.

Handling Boundary/Initial Conditions

Let $\boldsymbol{I}_1 \subset \Gamma_1$ and $\boldsymbol{I}_2 \subset \Gamma_2$ denote the discretization subset of Γ_1 and Γ_2, respectively. For non-Dirichlet boundary condition, it is natural to incorporate the boundary information by replacing $\boldsymbol{I}$ with $\boldsymbol{I} \cup \boldsymbol{I}_2$ and replacing $\mathcal{L}$ and f with their comprehensive forms. For Dirichlet boundary condition, we assume the value of PDE solution is known a priori on Γ_1 ahead of inference.

Handling Boundary/Initial Conditions

The posterior now is modified as follows,

$$
\begin{aligned}
&p_{\sigma_e^2,\boldsymbol{\Theta},U(\boldsymbol{I})|W_{\boldsymbol{I}},Y(\boldsymbol{\tau})=y(\boldsymbol{\tau}),U(\boldsymbol{I}_1)=b_1(\boldsymbol{I}_1))}\left(\sigma_e^2,\boldsymbol{\theta},u(\boldsymbol{I})|W_{\boldsymbol{I}}=0,Y(\boldsymbol{\tau})=y(\boldsymbol{\tau}),U(\boldsymbol{I}_1)=b_1(\boldsymbol{I}_1)\right)\\
\propto& P\left(\sigma_e^2,\boldsymbol{\Theta}=\boldsymbol{\theta},U(\boldsymbol{I})=u(\boldsymbol{I}),W_I=0,Y(\boldsymbol{\tau})=y(\boldsymbol{\tau}),U(\boldsymbol{I}_1)=b_1(\boldsymbol{I}_1)\right)\\
=&\pi(\sigma_e^2)\times\pi_{\boldsymbol{\Theta}}\left(\boldsymbol{\theta}\right)\times P\left(U(\boldsymbol{I})=u(\boldsymbol{I})\right)\\
&\times P\left(Y(\boldsymbol{\tau})=y(\boldsymbol{\tau})|U(\boldsymbol{I})=u(\boldsymbol{I}),\pi_{\boldsymbol{\Theta}}\right)\\
&\times P\left(U(\boldsymbol{I}_1)=b_1(\boldsymbol{I}_1)|U(\boldsymbol{I})=u(\boldsymbol{I})\right)\\
&\times P\left(W_{\boldsymbol{I}}=0|U(\boldsymbol{I}_1)=b_1(\boldsymbol{I}_1),U(\boldsymbol{I})=u(\boldsymbol{I}),\boldsymbol{\Theta}=\boldsymbol{\theta}\right).\\
\propto&\frac{1}{\sigma_e^2}\times\pi_{\boldsymbol{\Theta}}(\boldsymbol{\theta})\exp\Big\{-\frac{1}{2}\big[n_{\boldsymbol{I}}\log(2\pi)+\log(|C|)+\|u(\boldsymbol{I})-\mu(\boldsymbol{I})\|_{C^{-1}}\\
&+n\log(2\pi)+n\log(\sigma_e^2)+\|y(\boldsymbol{\tau})-u(\boldsymbol{\tau})\|_{\sigma_e^{-2}}\\
&+n_{\boldsymbol{I}_1}\log(2\pi)+\log|C_b|+\left\|b_1(\boldsymbol{I}_1)-\mu(\boldsymbol{I}_1)-\mathcal{K}(\boldsymbol{I}_1,\boldsymbol{I})\mathcal{K}(\boldsymbol{I},\boldsymbol{I})^{-1}(u(\boldsymbol{I})-\mu(\boldsymbol{I}_1))\right\|_{C_b^{-1}}\\
&+n_{\boldsymbol{I}}\log(2\pi)+\log|K_b|+\|f(\boldsymbol{I},u(\boldsymbol{I}),\boldsymbol{\theta})-\mathcal{L}_{\boldsymbol{x}}\mu(\boldsymbol{I})-m_b\{u(\boldsymbol{I}\cup\boldsymbol{I}_1))-\mu(\boldsymbol{I}\cup\boldsymbol{I}_1))\}\|_{K_b^{-1}}\big]\Big\},
\end{aligned}
$$

where $C_b=\mathcal{K}(\boldsymbol{I}_1,\boldsymbol{I}_1)-\mathcal{K}(\boldsymbol{I}_1,\boldsymbol{I})\mathcal{K}(\boldsymbol{I},\boldsymbol{I})^{-1}\mathcal{K}(\boldsymbol{I},\boldsymbol{I}_{\boldsymbol{1}})$,
$K_b=\mathcal{LKL}(\boldsymbol{I},\boldsymbol{I})-\mathcal{LK}(\boldsymbol{I},\boldsymbol{I}\cup\boldsymbol{I}_1)\mathcal{K}(\boldsymbol{I}\cup\boldsymbol{I}_1,\boldsymbol{I}\cup\boldsymbol{I}_1)^{-1}\mathcal{KL}(\boldsymbol{I}\cup\boldsymbol{I}_1,\boldsymbol{I})$ and
$m_b=\mathcal{LK}(\boldsymbol{I},\boldsymbol{I}\cup\boldsymbol{I}_1))\mathcal{K}(\boldsymbol{I}\cup\boldsymbol{I}_1),\boldsymbol{I}\cup\boldsymbol{I}_1))^{-1}$.

Dimensional Reduction for $U(I)$

- The parameter space is of dimension $ln_{\boldsymbol{I}} + d$, l is the number of PDE components. Thus, when $n_{\boldsymbol{I}}$ is large, optimizing or sampling from posterior are challenging tasks.
- The Karhunen Loeve (KL) expansion to the GP $U(\boldsymbol{x})$ is given by

$$U(\boldsymbol{x}) = \sum_{i=1}^{\infty} Z_i \sqrt{\lambda_i} \psi_i(\boldsymbol{x}),$$

- $\sqrt{\lambda_i}$ are eigenvectors of kernel function of GP in decreasing order, we can choose an $M \in \mathbb{N}$ such that λ_i for $i > M$ are negligible, then the GP $U(\boldsymbol{x})$ is approximated by

$$U(\boldsymbol{x}) \approx \sum_{i=1}^{M} Z_i \sqrt{\lambda_i} \psi_i(\boldsymbol{x}).$$

M is chosen such that $\sum_{i=1}^{M} \lambda_i / \sum_{i=1}^{n_{\boldsymbol{I}}} \lambda_i \geq 99.99\%$.

- $U(\boldsymbol{x})$ is parametrized by $(Z_1, Z_2, \ldots, Z_M)$.

Prior Tempering - Balancing Contribution of Prior and Likelihood

- When $\boldsymbol{I}$, i.e., the discretization set, is a large set while $\boldsymbol{\tau}$ is relatively small, i.e., the observation is sparse.
- When $\boldsymbol{I}_1$, i.e., the discretization set for boundary region, is a large set while $\boldsymbol{\tau}$ is relatively small.

The contribution of prior of $U(\boldsymbol{x})$ becomes too large. To mitigate this imbalance, we provide an idea of **tempering**. In particular

- We replace $\log(|C|) + \|u(\boldsymbol{I}) - \mu(\boldsymbol{I})\|_{C^{-1}}$ with $\log(|C|)/\beta + \|u(\boldsymbol{I}) - \mu(\boldsymbol{I})\|_{C^{-1}}/\beta$, where $\beta = n/n_{\boldsymbol{I}}$.
- We replace $\|m(\boldsymbol{I}_1) - y(\boldsymbol{I}_1)\|_{C_b^{-1}}$ by $\frac{1}{n_1}\|m(\boldsymbol{I}_1) - y(\boldsymbol{I}_1)\|_{C_b^{-1}}$.
- In summary, the posterior is modified to

$$
\begin{aligned}
& p^{(\beta)}_{\sigma_e^2,\boldsymbol{\Theta},U(\boldsymbol{I})|W_{\boldsymbol{I}},Y(\boldsymbol{\tau}),U(\boldsymbol{I}_1)}\left(\sigma_e^2,\boldsymbol{\theta},u(\boldsymbol{I})|W_{\boldsymbol{I}}=0,Y(\boldsymbol{\tau})=y(\boldsymbol{\tau}),U(\boldsymbol{I}_1)=b(\boldsymbol{I}_1)\right) \\
\propto & \pi_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) \times \left[P\left(U(\boldsymbol{I})=u(\boldsymbol{I})\right) \times P\left(W_{\boldsymbol{I}}=0|U(\boldsymbol{I}_1)=b(\boldsymbol{I}_1),U(\boldsymbol{I})=u(\boldsymbol{I}),\boldsymbol{\Theta}=\boldsymbol{\theta}\right)\right]^{\frac{1}{\beta}} \\
& \times P\left(U(\boldsymbol{I}_1)=b(\boldsymbol{I}_1)|U(\boldsymbol{I})=u(\boldsymbol{I})\right)^{\frac{1}{n_1}} \\
& \times \pi(\sigma_e^2) \times P\left(Y(\boldsymbol{\tau})=y(\boldsymbol{\tau})|U(\boldsymbol{I})=u(\boldsymbol{I}),\boldsymbol{\Theta}\right)
\end{aligned}
$$

Hamiltonian Monte Carlo

- We apply the Hamiltonian Monte Carlo(HMC) algorithm to draw random sample from the posterior distribution.
- Compared to using a Gaussian random walk proposal distribution in the Metropolis–Hastings algorithm, Hamiltonian Monte Carlo reduces the correlation between successive sampled states by proposing moves to distant states which maintain a high probability of acceptance.
- Leapfrog method for HMC is adopted to draw proposal for parameters.

Summary of PIGPI Procedure

Algorithm 1 Posterior Inference for PIGPI Procedure

1: Input data $\mathcal{D} = \{(y_i, \boldsymbol{x}_i), i = 1, \ldots, n_{\boldsymbol{\tau}}\}$, selecting $\boldsymbol{I}$ using method proposed in previous slides.
2: Train Gaussian process model based on $\mathcal{D}$, obtain the posterior density.
3: Optimize the posterior density to obtain MAP estimation of $u(\boldsymbol{I})$, σ_e^2 and $\boldsymbol{\theta}$.
4: Take MAP estimation obtained from Step 3 as a initial state. Draw posterior sample for $\boldsymbol{\theta}$, σ_e^2 and $u(\boldsymbol{I})$ using HMC algorithm.

Numerical Illustration - Preparation

Proposed methods

- PIGPI

Benchmark methods

- Two-Stage Method (TSM).
- Automated PDE identification (API) method (Liu et. al 2021).
- Methods (BM and PC) proposed by Xun et. al.(2013)

Xun, X., Cao, J., Mallick, B., Maity, A., & Carroll, R. J. (2013). *Parameter estimation of partial differential equation models.* Journal of the American Statistical Association, 108(503), 1009-1020.

Liu, R., Bianco, M. J., & Gerstoft, P. (2021). *Automated partial differential equation identification.* The Journal of the Acoustical Society of America, 150(4), 2364-2374.

Numerical Illustration - Preparation

Evaluation metrics

- Bias, Root mean square error (RMSE) of MAP of $\boldsymbol{\theta}$ (component-wise): Evaluate the accuracy of parameter estimation.
- Coverage rate of 95% credible intervals for $\boldsymbol{\theta}$ (component-wise): Evaluate the accuracy of uncertainty quantification.
- RMSE of MAP of $u(\boldsymbol{x_I})$: Evaluate the ability of recovering PDE.
- Computation time for MAP optimization: Evaluate the efficiency of alternative methods.

Example1-Contaminant Source Identification

- Consider a dimensionless diffusion equation on a square domain $\boldsymbol{s} \in S = [0,1] \times [0,1]$ and time interval $t \in [0,1]$

$$\frac{\partial u}{\partial t} - \nabla^2 u = \frac{c}{2\pi\sigma^2} \exp\left(-\frac{|\boldsymbol{s} - \boldsymbol{\chi}|}{2\sigma^2}\right).$$

- The boundary and initial conditions are given by $u(\boldsymbol{s}, t) = 0, \boldsymbol{s} \in \partial S, t \in [0,1], u(\boldsymbol{s}, 0) = 0, \boldsymbol{s} \in S$, where u is the dissolved concentration of contaminant, t is time, $\boldsymbol{s}$ is the location.
- Observation: $y(\boldsymbol{x}_i) = u(\boldsymbol{x}_i) + \varepsilon_i, i = 1, \ldots, n$, where $\boldsymbol{x}_i = (t_i, \boldsymbol{s}_i)$, $\varepsilon_i \sim N(0, \sigma_e)$.
- The purpose is to estimate the parameter $\boldsymbol{\theta} = (c, \mathcal{X}_1, \mathcal{X}_2)$.

Example1-Contaminant Source Identification

By applying HMC, the posterior density and trace of posterior sample are shown in figure 1 and 2. Note that the corresponding true values for $\boldsymbol{\theta} = (5, 0.25, 0.75)$ and $\sigma_e = 0.001$.

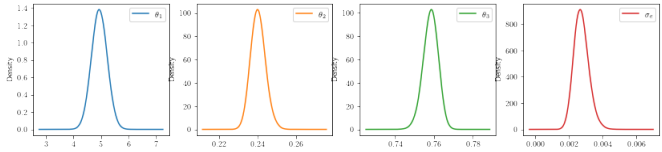

Figure 1: The density estimation of $\boldsymbol{\theta}$ and σ_e.

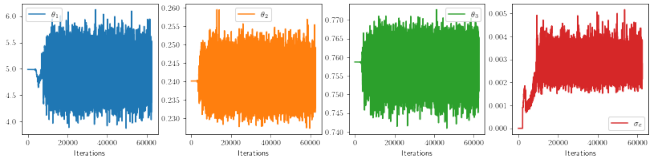

Figure 2: The trace plot of posterior sample for $\boldsymbol{\theta}$ and σ_e.

Example1-Contaminant Source Identification

We compare the computational time of MAP optimization between using and not using KL expansion.

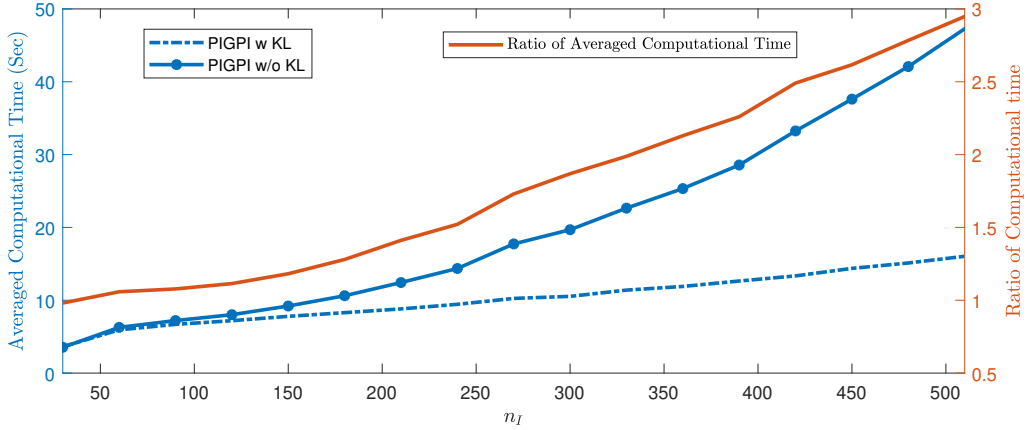

Figure 3: Comparison of computational time

Example2-Long-Range Infrared Light Detection and Ranging

Recall the motivation example

$$\frac{\partial u(t,s)}{\partial t} - \theta_D \frac{\partial^2 u(t,s)}{\partial s^2} - \theta_S \frac{\partial u(t,s)}{\partial s} = \theta_A u(t,s),\ t \in [0,20],\ s \in [0,40].$$

- The boundary condition is given by $u(t,0) = 0$ and initial condition is $u(0,s) = \{1 + 0.1 * (20-s)^2\}^{-1}$.
- Observation: $y(\boldsymbol{x}_i) = u(\boldsymbol{x}_i, \boldsymbol{\theta}_0) + \varepsilon_i, i = 1, \ldots, n$, where $\varepsilon_i \sim N(0, \sigma_e)$. The true value for $\boldsymbol{\theta}_0$ is $(1, 0.1, 0.1)$.
- θ_D and θ_S are involved in the linear PDE operator.;
- This example can be found in the JASA paper:
  Xun, X., Cao, J., Mallick, B., Maity, A., & Carroll, R. J. (2013). *Parameter estimation of partial differential equation models.* Journal of the American Statistical Association, 108(503), 1009-1020.

Example2-Long-Range Infrared Light Detection and Ranging

- PIGPI method can be applied but is possibly time consuming;
- Compare the computational time of PIGPI and PIGPI + PDE augmentation;
- Use Adam algorithm, 2500 iterations for each optimization.

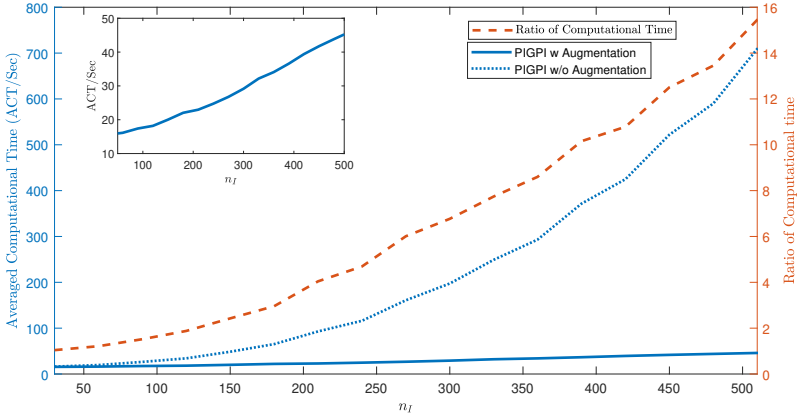

Figure 4: Comparison of computational time

Example2-Long-Range Infrared Light Detection and Ranging

- Comparing with methods proposed in JASA paper:
  Xun, X., Cao, J., Mallick, B., Maity, A., & Carroll, R. J. (2013). *Parameter estimation of partial differential equation models.* Journal of the American Statistical Association, 108(503), 1009-1020.
- $\boldsymbol{\tau} = \{(i, j), i = 1, 2, \ldots, 20, j = 1, 2, \ldots, 40\}$. $\boldsymbol{I} = \boldsymbol{\tau}$.
- Two methods are proposed in this paper, Bayesian method (BM) and parameter cascading method (PC).
- We comapre with BM, PC and TSM.

Example2-Long-Range Infrared Light Detection and Ranging

		$\sigma_e = 0.02$			$\sigma_e = 0.05$		
		θ_D	θ_A	θ_S	θ_D	θ_A	θ_S
Bias $\times 10^{-3}$	**PIGPI**	**-14.00**	**-0.20**	**-0.12**	**-27.35**	**-0.34**	**-0.29**
	BM	-16.50	-0.40	-0.20	-35.60	1.00	0.60
	PC	-29.70	-0.10	-0.30	-55.90	-0.20	-0.50
	TSM	-105.33	-2.69	-1.28	-140.12	-4.05	-2.12
SD $\times 10^{-3}$	**PIGPI**	**9.37**	**1.63**	**0.21**	**20.31**	**3.74**	**0.48**
	BM	9.10	1.60	0.20	22.20	3.80	0.50
	PC	24.90	3.80	0.50	40.50	6.20	0.80
	TSM	29.42	3.82	0.52	49.00	7.29	1.03
RMSE $\times 10^{-3}$	**PIGPI**	**16.85**	**1.64**	**0.24**	**34.06**	**3.75**	**0.56**
	BM	18.81	1.66	0.27	42.00	3.90	1.00
	PC	38.96	3.75	0.54	69.10	6.20	2.20
	TSM	109.35	4.67	1.38	148.43	8.34	2.36
CR %	**PIGPI**	**98.6**	**100**	**99.2**	**79.7**	**95.9**	**92.2**
	BM	93.9	99.9	98.8	74	97.8	93.5
	PC	84.3	96.7	94.9	78.1	96.5	93.8

Example3-Burger's Equation

We consider the viscous Burgers' equation given by

$$\frac{\partial u}{\partial t} - \theta_1 u \frac{\partial u}{\partial s} + \theta_2 \frac{\partial^2 u}{\partial s^2} = 0, s \in [0,1], t \in [0, 0.1],$$

where $\theta_2 > 0$ is the viscosity. Burgers' equation is a nonlinear equation. The boundary conditions and initial conditions are given by

$$\begin{aligned} \frac{\partial u(t,0)}{\partial s} = \frac{\partial u(t,1)}{\partial s} = 0, & \qquad t \in [0, 0.1] \\ u(0,s) = \exp\{-100(s-0.5)^2\}, & \qquad s \in [0,1] \end{aligned}$$

- Observation: $y(\boldsymbol{x}_i) = u(\boldsymbol{x}_i, \boldsymbol{\theta}_0) + \varepsilon_i, i = 1, \ldots, n$, where $\varepsilon_i \sim N(0, \sigma_e)$. The true value for $\boldsymbol{\theta}_0$ is $(1, 0.1)$.
- $\boldsymbol{\tau} = \{(i/20, j/20), i = 1, 2, \ldots, 20, j = 1, 2, \ldots, 20\}$. $\boldsymbol{I} = \boldsymbol{\tau}$.

Example3-Burger's Equation

- Compare with PDE identification method (API method):

Liu, R., Bianco, M. J., and Gerstoft, P. (2021). *Automated partial differential equation identification.* The Journal of the Acoustical Society of America, 150(4):2364–2374

		$\sigma_e = 0.001$		$\sigma_e = 0.01$	
		θ_1	θ_2	θ_1	θ_2
Bias $\times 10^{-3}$	**PIGPI w IBC**	**-4.60**	**-0.05**	**-15.49**	**-0.27**
	PIGPI w/o IBC	**-3.15**	**-0.19**	**-23.24**	**-1.60**
	API	10.76	-6.69	108.59	85.97
	TSM	-10.44	-2.19	-50.61	-8.55
SD $\times 10^{-3}$	**PIGPI w IBC**	**2.49**	**0.19**	**11.31**	**0.79**
	PIGPI w/o IBC	**2.84**	**0.22**	**19.21**	**1.34**
	API	6.15	0.51	299.73	258.97
	TSM	4.01	0.35	27.24	2.72
RMSE $\times 10^{-3}$	**PIGPI w IBC**	**5.23**	**0.20**	**19.18**	**0.83**
	PIGPI w/o IBC	**4.24**	**0.29**	**30.15**	**2.09**
	API	12.39	6.71	318.66	272.74
	TSM	11.18	2.22	57.46	8.97
CR %	**PIGPI w IBC**	**100**	**100**	**82**	**96.2**
	PIGPI w/o IBC	**100**	**100**	**81.4**	**81.5**

Example3-Burger's Equation

- The improvement of taking advantages of boundary/initial conditions;
- The boundary conditions can significantly reduce the error of posterior inference of PDE solution.

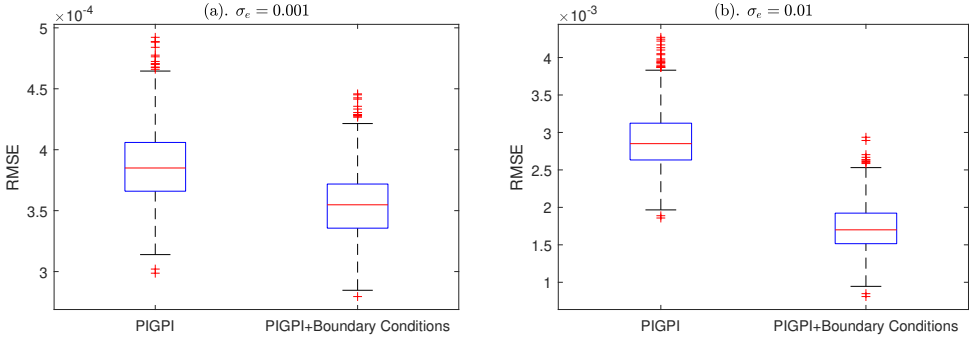

Figure 5: Comparison of RMSEs, PIGPI without boundary conditions v.s. PIGPI with boundary conditions

Summary

- We propose a new method for *parameter inference* involves complex PDE models.
- The proposed method doesn't require time-consuming PDE numerical solver such as finite element methods or finite difference methods.
- We present methods for choosing discretization set and dimensional reduction to the parameters.
- Numerical examples are employed to illustrate the performance of the proposed method.

Thank You!

zhaohui.li@gatech.edu