Inference of Nonlinear Partial Differential Equations via Constrained Gaussian Processes

Zhaohui Li

H. Milton Stewart School of Industrial and Systems Engineering (ISyE),
Georgia Institute of Technology

June 8, 2023

Joint work with Shihao Yang (ISyE) and Jeff Wu (ISyE)

1. Introduction

2. Methodology
   - Motivating Example
   - Problem Formulation
   - Methodology

3. Numerical Illustrations
   - Example-Long-Range Infrared Light Detection and Ranging Equation
   - Example-Burger's Equation

4. Summary

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.
- In real-world applications, directly measuring these parameters is sometimes impossible.
- Estimating these parameters from experimental data is an important task, known as model calibration, inverse problems, etc.
- In this work, we focus on inferring unknown *parameters* in complex PDE models from **sparse**, **noisy** observation data.

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.
- In real-world applications, directly measuring these parameters is sometimes impossible.
- Estimating these parameters from experimental data is an important task, known as model calibration, inverse problems, etc.
- In this work, we focus on inferring unknown *parameters* in complex PDE models from **sparse**, **noisy** observation data.

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.
- In real-world applications, directly measuring these parameters is sometimes impossible.
- Estimating these parameters from experimental data is an important task, known as model calibration, inverse problems, etc.
- In this work, we focus on inferring unknown *parameters* in complex PDE models from **sparse**, **noisy** observation data.

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.
- In real-world applications, directly measuring these parameters is sometimes impossible.
- Estimating these parameters from experimental data is an important task, known as model calibration, inverse problems, etc.
- In this work, we focus on inferring unknown *parameters* in complex PDE models from **sparse**, **noisy** observation data.

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.
- In real-world applications, directly measuring these parameters is sometimes impossible.
- Estimating these parameters from experimental data is an important task, known as model calibration, inverse problems, etc.
- In this work, we focus on inferring unknown *parameters* in complex PDE models from **sparse**, **noisy** observation data.

Motivating Example

- The long-range infrared light detection and ranging (LIDAR) equation:

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

  with specified boundary and initial conditions.

- Objective: estimate the parameters $\boldsymbol{\theta} = (\theta_S, \theta_D, \theta_A)$ from the observation data $y(\boldsymbol{x}_i) = u(\boldsymbol{x}_i) + \varepsilon$, where $\boldsymbol{x}_i = (t_i, s_i)$, $\varepsilon_i \sim N(0, \sigma_e^2), i = 1, \ldots, n$ are random errors.

Problem Formulation

Start with a semi-linear partial differential equation (PDE):

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

where $\boldsymbol{x} = (x_1, \ldots, x_p)$, $\mathcal{L}u(\boldsymbol{x})$ denotes a *linear* differential operator of order a:

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

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 $\mathcal{L}$ is defined by $a = \max_i |\boldsymbol{\alpha}_i|$.
For example, in the LIDAR equation,

- $\mathcal{L}u(\boldsymbol{x}) = \frac{\partial u(\boldsymbol{x})}{\partial t} - \theta_S \frac{\partial^2 u(\boldsymbol{x})}{\partial s^2} - \theta_D \frac{\partial u(\boldsymbol{x})}{\partial s}$, $f(\boldsymbol{x}, u(\boldsymbol{x}), \boldsymbol{\theta}) = \theta_A u(\boldsymbol{x})$.
- $c_1 = 1, c_2 = -\theta_1, c_3 = -\theta_2$, $\boldsymbol{\alpha}_1 = (1, 0), \boldsymbol{\alpha}_2 = (0, 2), \boldsymbol{\alpha}_3 = (0, 1)$.

Problem Formulation

Start with a semi-linear partial differential equation (PDE):

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

where $\boldsymbol{x} = (x_1, \ldots, x_p)$, $\mathcal{L}u(\boldsymbol{x})$ denotes a *linear* differential operator of order a:

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

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 $\mathcal{L}$ is defined by $a = \max_i |\boldsymbol{\alpha}_i|$.
For example, in the LIDAR equation,

- $\mathcal{L}u(\boldsymbol{x}) = \frac{\partial u(\boldsymbol{x})}{\partial t} - \theta_S \frac{\partial^2 u(\boldsymbol{x})}{\partial s^2} - \theta_D \frac{\partial u(\boldsymbol{x})}{\partial s}$, $f(\boldsymbol{x}, u(\boldsymbol{x}), \boldsymbol{\theta}) = \theta_A u(\boldsymbol{x})$.
- $c_1 = 1, c_2 = -\theta_1, c_3 = -\theta_2$, $\boldsymbol{\alpha}_1 = (1, 0), \boldsymbol{\alpha}_2 = (0, 2), \boldsymbol{\alpha}_3 = (0, 1)$.

Basic Idea

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

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

- $W = 0 \Leftrightarrow U$ **is the solution of PDE with specified parameter** θ**.**

Basic Idea

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

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

- $W = 0 \Leftrightarrow U$ **is the solution of PDE with specified parameter** θ**.**

Basic Idea

- The task is to estimate the parameters $\boldsymbol{\theta}$ from the observation data $y(\boldsymbol{x}_i) = u(\boldsymbol{x}_i) + \varepsilon_i, i = 1, \ldots, n$. Let $\boldsymbol{\tau} = \{\boldsymbol{x}_i, i = 1, \ldots, n\}$.
- Assign a Gaussian process (GP) prior on $u(\boldsymbol{x})$ denoted by $U(\boldsymbol{x}) \sim \text{GP}(\mu, \sigma^2\mathcal{K}(\cdot,\cdot))$.
- 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.,

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

- $W = 0 \Leftrightarrow U$ **is the solution of PDE with specified parameter** θ.

Basic Idea

- The task is to estimate the parameters $\boldsymbol{\theta}$ from the observation data $y(\boldsymbol{x}_i) = u(\boldsymbol{x}_i) + \varepsilon_i, i = 1, \ldots, n$. Let $\boldsymbol{\tau} = \{\boldsymbol{x}_i, i = 1, \ldots, n\}$.
- 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.,

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

- $W = 0 \Leftrightarrow U$ **is the solution of PDE with specified parameter $\boldsymbol{\theta}$.**

Basic Idea

- W is not computable!
- 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

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

- As $\boldsymbol{I}$ getting denser and denser, we expect that $W_{\boldsymbol{I}}$ provides a good approximation to W, this needs
  1. Proper smoothness condition on the function space $\mathcal{H}$ (PDE solution provides good smoothness properties.).
  2. $\boldsymbol{I}$ should be space-filling 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.
  3. From a computational point of view, $|\boldsymbol{I}|$ should be as small as possible.

Basic Idea

- W is not computable!
- 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

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

- As I getting denser and denser, we expect that W_I provides a good approximation to W, this needs
  1. Proper smoothness condition on the function space $\mathcal{H}$ (PDE solution provides good smoothness properties.).
  2. I should be space-filling in Ω, i.e., $\forall x \in \Omega$, distance between x and I should be as small as possible. I should be space filling.
  3. From a computational point of view, $|I|$ should be as small as possible.

Basic Idea

- W is not computable!
- 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

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

- As $\boldsymbol{I}$ getting denser and denser, we expect that $W_{\boldsymbol{I}}$ provides a good approximation to W, this needs
  1. Proper smoothness condition on the function space $\mathcal{H}$ (PDE solution provides good smoothness properties.).
  2. $\boldsymbol{I}$ should be space-filling 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.
  3. From a computational point of view, $|\boldsymbol{I}|$ should be as small as possible.

Basic Idea

Important properties of GP:

- Assume constant variance σ^2, $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}')$ has enough degree of smoothness, i.e.$\mathcal{L}_{\boldsymbol{x}} \mathcal{L}_{\boldsymbol{x}'} \mathcal{K}(\boldsymbol{x}, \boldsymbol{x}')$ exists and continuous.
- Then for given parameter $\boldsymbol{\theta}$, $\mathcal{L}U(\boldsymbol{x})$ is also GP: $\mathcal{L}_{\boldsymbol{x}} U(\boldsymbol{x}) \sim \text{GP}(\mathcal{L}_{\boldsymbol{x}} \mu(\boldsymbol{x}), \mathcal{L}_{\boldsymbol{x}} \mathcal{L}_{\boldsymbol{x}'} \mathcal{K}(\boldsymbol{x}, \boldsymbol{x}'))$.
- Correlation: $\text{corr}(\mathcal{L}_{\boldsymbol{x}} U(\boldsymbol{x}), U(\boldsymbol{x}')) = \mathcal{L}_{\boldsymbol{x}} \mathcal{K}(\boldsymbol{x}, \boldsymbol{x}')$.

We employ the product Matérn kernel:

- $\mathcal{K}(x, x') = \phi_1 \prod_{i=1}^{p} \frac{2^{1-\nu}}{\Gamma(\nu)} (\sqrt{2\nu} \frac{d_i}{\phi_{2i}})^{\nu} B_{\nu}(\sqrt{2\nu} \frac{d_i}{\phi_{2i}})$, where $d_i = |x_i - x_i'|, i = 1, \ldots, p$, Γ: Gamma function, B_{ν}: the modified Bessel function of the second kind.
- Degree of freedom ν is set to be $2a + \delta$ to ensure that the $2a$-th order derivatives of the kernel with respect to any coordinate x_i exists, where δ is a small positive number.

Basic Idea

Important properties of GP:

- Assume constant variance σ^2, $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}')$ has enough degree of smoothness, i.e.$\mathcal{L}_{\boldsymbol{x}}\mathcal{L}_{\boldsymbol{x}'}\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}')$ exists and continuous.
- Then for given parameter $\boldsymbol{\theta}$, $\mathcal{L}U(\boldsymbol{x})$ is also GP: $\mathcal{L}_{\boldsymbol{x}}U(\boldsymbol{x}) \sim \text{GP}(\mathcal{L}_{\boldsymbol{x}}\mu(\boldsymbol{x}), \mathcal{L}_{\boldsymbol{x}}\mathcal{L}_{\boldsymbol{x}'}\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}'))$.
- Correlation: $\text{corr}(\mathcal{L}_{\boldsymbol{x}}U(\boldsymbol{x}), U(\boldsymbol{x}')) = \mathcal{L}_{\boldsymbol{x}}\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}')$.

We employ the product Matérn kernel:

- $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \phi_1 \prod_{i=1}^{p} \frac{2^{1-\nu}}{\Gamma(\nu)}(\sqrt{2\nu}\frac{d_i}{\phi_{2i}})^{\nu} B_{\nu}(\sqrt{2\nu}\frac{d_i}{\phi_{2i}})$, where $d_i = |x_i - x_i'|, i = 1, \dots, p$, Γ: Gamma function, B_{ν}: the modified Bessel function of the second kind.
- Degree of freedom ν is set to be $2a + \delta$ to ensure that the $2a$-th order derivatives of the kernel with respect to any coordinate x_i exists, where δ is a small positive number.

Basic Idea

- Treating $W_{\boldsymbol{I}}$ as an approximation of W and assigning a noninformative prior for $\boldsymbol{\theta}$, the posterior is immediately obtained

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

- Posterior inference for both $\boldsymbol{\theta}$ and $u(\boldsymbol{I})$ can be done by sampling from/optimizing this (unnormalized) posterior density.
- The method is called **P**DE-**I**nformed **G**aussian **P**rocess **I**nference (**PIGPI**).

Calculation of GP components:

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

- $\mathcal{K}(\boldsymbol{I}, \boldsymbol{I})$: an $n_I \times n_I$ matrix with (i, j) element $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$;
- $\mathcal{L}\mathcal{K}(\boldsymbol{I}, \boldsymbol{I})$, : an $n_I \times n_I$ matrix with (i, j) element $\mathcal{L}_{\boldsymbol{x}}(\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j))$;
- $\mathcal{K}\mathcal{L}(\boldsymbol{I}, \boldsymbol{I})$: an $n_I \times n_I$ matrix with (i, j) element $\mathcal{L}_{\boldsymbol{x}'}(\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j))$;
- $\mathcal{L}\mathcal{K}\mathcal{L}(\boldsymbol{I}, \boldsymbol{I})$: an $n_I \times n_I$ matrix with (i, j) element $\mathcal{L}_{\boldsymbol{x}}(\mathcal{L}_{\boldsymbol{x}'}(\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)))$;

Handling Non-linear and Parameter-dependent Operators

Two limitations of original PIGPI:

- **Computationally Expensive for Parameter Dependent Operator:** When $\mathcal{L}_{\boldsymbol{x}}^{\boldsymbol{\theta}}$ depends on $\boldsymbol{\theta}$, the updating of $\boldsymbol{\theta}$ requires the updating of $\mathcal{L}\mathcal{K}\mathcal{L}$ and K^{-1}.
- **Non-Flexible to Non-Linear Operator:** For a non-linear PDE $\mathcal{A}(u, \boldsymbol{x}) = f$, $\mathcal{A}(U, \boldsymbol{x})$ may not be Gaussian.

To solve these problems, we propose a novel method that

- can decouple the dependence between parameter $\boldsymbol{\theta}$ and covariance matrix K .
- can handle nonlinear PDEs.

Handling Non-linear and Parameter-dependent Operators

Two limitations of original PIGPI:

- **Computationally Expensive for Parameter Dependent Operator:** When $\mathcal{L}_{\boldsymbol{x}}^{\boldsymbol{\theta}}$ depends on $\boldsymbol{\theta}$, the updating of $\boldsymbol{\theta}$ requires the updating of $\mathcal{L}\mathcal{K}\mathcal{L}$ and K^{-1}.
- **Non-Flexible to Non-Linear Operator:** For a non-linear PDE $\mathcal{A}(u, \boldsymbol{x}) = f$, $\mathcal{A}(U, \boldsymbol{x})$ may not be Gaussian.

To solve these problems, we propose a novel method that

- can decouple the dependence between parameter $\boldsymbol{\theta}$ and covariance matrix K .
- can handle nonlinear PDEs.

Handling Non-linear and Parameter-dependent Operators

A nonlinear PDE, Burger's equation:

$$\frac{\partial u}{\partial t}(\boldsymbol{x}) - \theta_1 u(\boldsymbol{x}) \frac{\partial u}{\partial s}(\boldsymbol{x}) + \theta_2 \frac{\partial^2 u}{\partial s^2}(\boldsymbol{x}) = 0.$$

The PDE operator $\mathcal{A}(u, \boldsymbol{x}) = \frac{\partial u}{\partial t} - \theta_1 u \frac{\partial u}{\partial s} + \theta_2 \frac{\partial^2 u}{\partial s^2}$ is

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

Handling Non-linear and Parameter-dependent Operators

Burger's equation,

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

Define an equivalent PDE system,

$$\begin{cases} \frac{\partial u}{\partial s}(\boldsymbol{x}) & = u_2(\boldsymbol{x}), \\ \frac{\partial u_2}{\partial s}(\boldsymbol{x}) & = u_3(\boldsymbol{x}), \\ \frac{\partial u}{\partial t}(\boldsymbol{x}) & = \theta_1 u(\boldsymbol{x}) u_2(\boldsymbol{x}) - \theta_2 u_3(\boldsymbol{x}), \end{cases} \tag{1}$$

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

Handling Non-linear and Parameter-dependent Operators

Properties of augmented PDE:

- **Equivalence**:
  - 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;
- **Decoupling PDE operator and parameter**: Augmented PDE operator (LFH) is independent of parameters;
- **Linearity**: Augmented PDE operator is linear.

It is natural to apply the proposed PIGPI to the augmented PDE.

Discussion - Non-uniqueness of Augmentation

Lowest degree of derivative (LDD) principal:

- The augmentation is not unique.
- Another augmentation for Burger's equation:

$$\begin{cases} \frac{\partial u}{\partial s}(\boldsymbol{x}) & = u_2(\boldsymbol{x}), \\ \frac{\partial^2 u}{\partial s^2}(\boldsymbol{x}) & = u_3(\boldsymbol{x}), \\ \frac{\partial u}{\partial t}(\boldsymbol{x}) & = \theta_1 u(\boldsymbol{x}) u_2(\boldsymbol{x}) - \theta_2 u_3(\boldsymbol{x}), \end{cases}$$

Our recommendation: (1).

Reasons:

- PDE (1) is a 1-order PDE while the second one is 2-order PDE. We prefer to use a lower-order PDE.
- PDE (1) produces a simpler covariance matrix K.

Discussion - Non-uniqueness of Augmentation

Lowest degree of derivative (LDD) principal:

- The augmentation is not unique.
- Another augmentation for Burger's equation:

$$\begin{cases} \frac{\partial u}{\partial s}(\boldsymbol{x}) & = u_2(\boldsymbol{x}), \\ \frac{\partial^2 u}{\partial s^2}(\boldsymbol{x}) & = u_3(\boldsymbol{x}), \\ \frac{\partial u}{\partial t}(\boldsymbol{x}) & = \theta_1 u(\boldsymbol{x}) u_2(\boldsymbol{x}) - \theta_2 u_3(\boldsymbol{x}), \end{cases}$$

Our recommendation: (1).
Reasons:

- PDE (1) is a 1-order PDE while the second one is 2-order PDE. We prefer to use a lower-order PDE.
- PDE (1) produces a simpler covariance matrix K.

Handling Non-linear and Parameter-dependent Operators

Name	Original Form	Augmented Form
Fisher's Equation	$\frac{\partial u}{\partial t} - D\frac{\partial^2 u}{\partial s^2} = ru(1-u)$	$\frac{\partial^2 u_1}{\partial s^2} = u_2$ $\frac{\partial u_1}{\partial t} = Du_2 + ru_1(1-u_1)$
Telegraph Equation	$\frac{\partial^2 u}{\partial t^2} + k\frac{\partial u}{\partial t} = a^2\frac{\partial^2 u}{\partial x^2} + bu$	$\frac{\partial u_1}{\partial t} = u_2$ $\frac{\partial^2 u_1}{\partial x^2} = u_3$ $\frac{\partial u_2}{\partial t} = a^2 u_3 + bu_1 - ku_2$
Nonlinear Heat Equation	$\frac{\partial u}{\partial t} = a\frac{\partial}{\partial s}\left(e^{\lambda u}\frac{\partial u}{\partial s}\right) + b + c_1 e^{\beta u} + c_2 e^{\gamma u}$	$\frac{\partial u_1}{\partial s} = u_2$ $\frac{\partial u_2}{\partial s} = u_3$ $\frac{\partial u_1}{\partial t} = a\lambda e^{\lambda u_1} u_2^2 + ae^{\lambda u_1} u_3$ $+b + c_1 e^{\beta u_1} + c_2 e^{\gamma u_1}$
Generalized Korteweg–de Vries Equation	$\frac{\partial u}{\partial t} + \frac{\partial^3 u}{\partial s^3} + g(u)\frac{\partial u}{\partial s} = 0$	$\frac{\partial u_1}{\partial s} = u_2$ $\frac{\partial u_1}{\partial t} + \frac{\partial^3 u_2}{\partial s^2} = g(u_1)u_2$
Reaction-Diffusion System	$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial s^2} + F(u,v),$ $\frac{\partial v}{\partial t} = a\frac{\partial^2 v}{\partial s^2} + G(u,v)$	$\frac{\partial^2 u_1}{\partial s^2} = u_2$ $\frac{\partial^2 v_1}{\partial s^2} = v_2$ $\frac{\partial u_1}{\partial t} = au_2 + F(u_1, v_1)$ $\frac{\partial v_1}{\partial t} = av_2 + G(u_1, v_1)$

Imperfect Augmentation Idea - Unable to Handle Arbitrary PDEs

Example (Eikonal Equation)

To end this subsection, we give an example to which our framework is not applicable. The PDE is one kind of the well-known Eikonal equation,

$$(\frac{\partial u}{\partial x_1})^2 + (\frac{\partial u}{\partial x_2})^2 = f(u, \boldsymbol{x}, \boldsymbol{\theta}), \tag{2}$$

where $f(u, \boldsymbol{x}, \boldsymbol{\theta})$ is a positive valued function. By simple algebra, we get two PDEs, i.e.,

$$\frac{\partial u}{\partial x_1} = \sqrt{f(u, \boldsymbol{x}, \theta) - (\frac{\partial u}{\partial x_2})^2},$$
$$\frac{\partial u}{\partial x_1} = -\sqrt{f(u, \boldsymbol{x}, \theta) - (\frac{\partial u}{\partial x_2})^2},$$

whose solutions are both solutions of (2). Thus, without additional information (eg. u is a function increasing with x_1), there is no unique augmentation form that is equivalent to (2).

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 space filling in Ω.
- 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 $i = n+1, \ldots, n_{\boldsymbol{I}}$, 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 Initial/Boundary Conditions

Two types of Initial/Boundary Conditions (IBCs): Dirichlet IBCs and Non-Dirichlet IBCs.

- Dirichlet IBCs: known value of PDE solution on the specific boundary regions, i.e.,

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

- The initial conditions are typically Dirichlet types;
- Non-Dirichlet IBCs

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

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

Handling Initial/Boundary Conditions

For non-Dirichlet IBCs, we define the comprehensive operator

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

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 IBCs, it is natural to incorporate the boundary information by including several properly chosen boundary points and replacing $\mathcal{L}$ and f with their comprehensive forms.

Handling Initial/Boundary Conditions

Dirichlet IBCs is treated as a set of noiseless observation on the boundary: $\boldsymbol{I}_1$ is the discretization for boundary Γ_1.

$$\begin{aligned}
&p_{\boldsymbol{\Theta},U(\boldsymbol{I})|W_{\boldsymbol{I}},Y(\boldsymbol{\tau})=y(\boldsymbol{\tau}),U(\boldsymbol{I}_1)=u(\boldsymbol{I}_1))}\left(\boldsymbol{\theta},u(\boldsymbol{I})|W_{\boldsymbol{I}}=0,Y(\boldsymbol{\tau})=y(\boldsymbol{\tau}),U(\boldsymbol{I}_1)=b(\boldsymbol{I}_1)\right)\\
\propto&P\left(\boldsymbol{\Theta}=\boldsymbol{\theta},U(\boldsymbol{I})=u(\boldsymbol{I}),W_I=0,Y(\boldsymbol{\tau})=y(\boldsymbol{\tau}),U(\boldsymbol{I}_1)=b(\boldsymbol{I}_1)\right)\\
=&\pi_{\boldsymbol{\Theta}}\left(\boldsymbol{\theta}\right)\times P\left(U(\boldsymbol{I})=u(\boldsymbol{I})\right)\\
&\times P\left(Y(\boldsymbol{\tau})=y(\boldsymbol{\tau})|U(\boldsymbol{I})=u(\boldsymbol{I})\right)\times P\left(U(\boldsymbol{I}_1)=u(\boldsymbol{I}_1)|U(\boldsymbol{I})=u(\boldsymbol{I})\right)\\
&\times P\left(W_{\boldsymbol{I}}=0|U(\boldsymbol{I}_1)=u(\boldsymbol{I}_1),U(\boldsymbol{I})=u(\boldsymbol{I}),\boldsymbol{\Theta}=\boldsymbol{\theta}\right).\\
\propto&\exp\Bigg\{-\frac{1}{2}\Big[|\boldsymbol{I}|\log(2\pi)+\log(|C|)+\|u(\boldsymbol{I})-\mu(\boldsymbol{I})\|_{C^{-1}}\\
&+n\log(2\pi\sigma_e^2)+\|u(\boldsymbol{\tau})-y(\boldsymbol{\tau})\|_{\sigma_e^{-2}}+\|m(\boldsymbol{I}_1)-y(\boldsymbol{I}_1)\|_{C_b^{-1}}\\
&+|\boldsymbol{I}|\log(2\pi\sigma_u^2)+\log|K|+\|f(\boldsymbol{I},u(\boldsymbol{I}),\boldsymbol{\theta})-\mathcal{L}\mu(\boldsymbol{I})-m\{u(\boldsymbol{I})-\mu(\boldsymbol{I})\}\|_{K_b^{-1}}\Big]\Bigg\},
\end{aligned}\tag{5}$$

where $C_b=\mathcal{K}(\boldsymbol{I}_1,\boldsymbol{I}_1)-\mathcal{K}(\boldsymbol{I}_1,\boldsymbol{I})\mathcal{K}(\boldsymbol{I},\boldsymbol{I})^{-1}\mathcal{K}(\boldsymbol{I},\boldsymbol{I_1})$,
$K_b=\mathcal{L}\mathcal{K}\mathcal{L}(\boldsymbol{I},\boldsymbol{I})-\mathcal{L}\mathcal{K}(\boldsymbol{I},\boldsymbol{I}\cup\boldsymbol{I}_1)\mathcal{K}(\boldsymbol{I}\cup\boldsymbol{I}_1,\boldsymbol{I}\cup\boldsymbol{I}_1)^{-1}\mathcal{K}\mathcal{L}(\boldsymbol{I}\cup\boldsymbol{I}_1,\boldsymbol{I})$.

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 $(Z_1, Z_2, \ldots, Z_M)$.

Numerical Illustration - Preparation

Evaluation metrics

- Root means square error (RMSE) or mean absolute percentage error(MAPE) of MAP of $\boldsymbol{\theta}$: Evaluate the accuracy of parameter estimation.
- RMSE of MAP of $u(\boldsymbol{x_I})$: Evaluate the accuracy of PDE solution estimation.
- Computation time for MAP optimization: Evaluate the efficiency of alternative methods.

Benchmark methods

- Two-Stage method (TSM) (Rai and Tripathi,2019).
- Automated PDE identification (API) method (Liu et. al 2021).
- Methods (BM and PC) proposed by Xun et. al.(2013)

Rai P.K & Tripathi, S. (2019) Gaussian process for estimating parameters of partial differential equations918 and its application to the Richards equation, Stochastic Environmental Research and Risk Assessment,33, pp. 1629–1649.

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.

Example-Long-Range Infrared Light Detection and Ranging

Revisit Motivating Example:

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

- IBCs: $u(t,0) = u(t,40) = 0$; $u(0,s) = (1 + 0.1 * (20 - s)^2)^{-1}$.
- 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 $(\theta_D, \theta_S, \theta_A) = (1, 0.1, 0.1)$.
- Linear PDE operator depends on θ_D, θ_S.
  - PIGPI without augmentation
  - PIGPI with augmentation

Example-Long-Range Infrared Light Detection and Ranging

Augmented PDE satisfies LOD principal:

$$\frac{\partial u_1(t,s)}{\partial s} = u_2(t,s),$$
$$\frac{\partial u_2(t,s)}{\partial s} = u_3(t,s),$$
$$\frac{\partial u_1(t,s)}{\partial t} = \theta_D u_3(t,s) + \theta_S u_2(t,s) + \theta_A u_1(t,s).$$

Example-Long-Range Infrared Light Detection and Ranging

- Both PIGPI with augmentation and PIGPI without augmentation can be applied.
- Use Adam algorithm, 2500 iterations for each method.

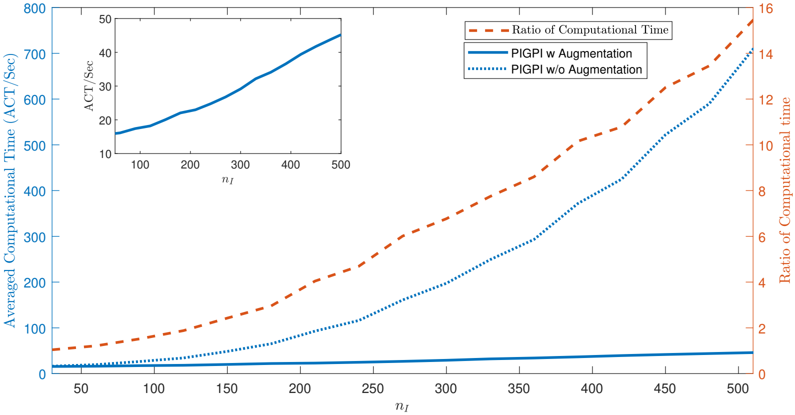

Figure 1: Comparison of computational time

Example-Long-Range Infrared Light Detection and Ranging

- Two methods are proposed in this paper, the Bayesian method (BM) and the parameter cascading method (PC): 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.
- We compare with BM, PC, and TSM(Two-stage method).
- $n = 800$, two cases for variance of random error $\sigma_e = 0.02$ or $\sigma_e = 0.05$.

Example-Long-Range Infrared Light Detection and Ranging

		$\sigma_e = 0.02$			$\sigma_e = 0.05$		
		θ_1	θ_2	θ_3	θ_1	θ_2	θ_3
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

Example-Burger's Equation

Revisit Nonlinear Burgers' equation:

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

IBCs:

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

- Compared with automated PDE identification method (API):
  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

Example-Burger's Equation

		$\sigma_e = 0.001$		$\sigma_e = 0.01$	
		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**

Example-Burger's Equation

The improvement of taking advantage of IBCs:

- IBCs are helpful to improve the estimation of parameters (Left).
- IBCs can significantly reduce the error of posterior inference of PDE solution (Right).

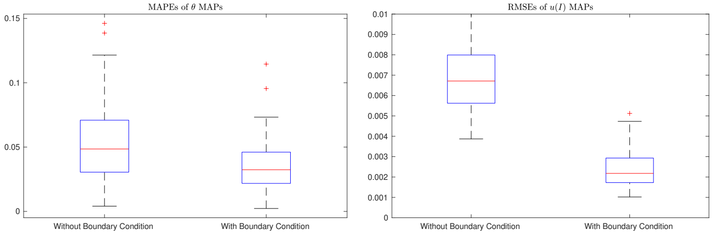

Figure 2: Comparison of MAPEs, PIGPI without IBCs v.s. PIGPI with IBCs

Summary

- We propose a new method for *parameter inference* involves complex PDE models, called PDE-Informed Gaussian Process Inference (PIGPI):
  - **Bypasses the requirement of a time-consuming PDE solver such as the finite element method.**
  - **Flexible to Nonlinear PDE and PDE systems with unobserved components.**
  - **Scalable to large data set:**
    - ⋆ Dimensional reduction method that is helpful for reducing the computational complexity when the discretization set is large.
  - **Ability to incorporate initial/boundary conditions.**
  - Numerical examples are employed to illustrate the performance of the proposed method.

Q & A

Thank You!

zhaohui.li@gatech.edu