

## Statistica Sinica Preprint No: SS-2022-0061

<b>Title</b>	Identification of Partial-Differential-Equations-Based Models from Noisy Data with Splines
<b>Manuscript ID</b>	SS-2022-0061
<b>URL</b>	<a href="http://www.stat.sinica.edu.tw/statistica/">http://www.stat.sinica.edu.tw/statistica/</a>
<b>DOI</b>	10.5705/ss.202022.0061
<b>Complete List of Authors</b>	Yujie Zhao, Xiaoming Huo and Yajun Mei
<b>Corresponding Authors</b>	Yujie Zhao
<b>E-mails</b>	yujie.zhao@merck.com
Notice: Accepted version subject to English editing.	

# Identification of Partial-Differential-Equations-Based Models from Noisy Data via Splines

Yujie Zhao<sup>1</sup>, Xiaoming Huo<sup>2</sup>, Yajun Mei<sup>2</sup>

<sup>1</sup> *Biostatistics and Research Decision Sciences Department, Merck & Co., Inc*

<sup>2</sup> *H. Milton Stewart School of Industrial and Systems Engineering, Georgia Tech*

*Abstract:* We propose a two-stage method called *Spline Assisted Partial Differential Equation based Model Identification (SAPDEMI)* to identify partial differential equation (PDE)-based models from noisy data. In the first stage, we employ the cubic splines to estimate unobservable derivatives. The underlying PDE is based on a subset of these derivatives. This stage is computationally efficient: its computational complexity is a product of a constant with the sample size; this is the lowest possible order of computational complexity. In the second stage, we apply the Least Absolute Shrinkage and Selection Operator (Lasso) to identify the underlying PDE-based model. Statistical properties are developed, including the model identification accuracy. We validate our theory through various numerical examples and a real data case study. The case study is based on an National Aeronautics and Space Administration (NASA) data set.

*Key words and phrases:* partial differential equations, model identification, cubic splines, Lasso.

## 1. Introduction

Partial differential equations (PDE) are widely adopted to model physical processes in engineering (Wang et al., 2019), physics (Xun et al., 2013), biology (Lagergren et al., 2020), and more. In these applications, there are two classes of technical issues: the *forward problem* and the *inverse problem*. The forward problem studies the properties of functions that PDEs determine. It has been extensively studied by mathematicians (Olver, 2014; Wang et al., 2014). Different from forward problems, inverse problems aim at identifying PDE-based models from the observed noisy data. The research on the inverse problem is relatively sparse. The corresponding statistical property is notably less known. In this paper, we propose a method to solve the inverse problem. The inverse problem will be called a *PDE identification problem* and will be formulated.

Given the rise of the big data, the PDE identification problem becomes indispensable. A good PDE identification approach generates at least the following two benefits. First, one can predict future trends by the identified PDE model, conditioning such a model reflects the underlying processes. Second, interpretable PDE models allows scientists to validate/reexamine the underlying physical/biological laws governing the process.

We propose a new method – *Spline Assisted Partial Differential Equa-*

tion based Model Identification (SAPDEMI) – for the PDE identification problem. SAPDEMI can efficiently identify the underlying PDE model from the noisy data  $\mathcal{D}$ :

$$\begin{aligned} \mathcal{D} = \{ & (x_i, t_n, u_i^n) : x_i \in (0, X_{\max}) \subseteq \mathbb{R}, \forall i = 0, \dots, M - 1, \\ & t_n \in (0, T_{\max}) \subseteq \mathbb{R}, \forall n = 0, \dots, N - 1\} \in \Omega, \end{aligned} \quad (1.1)$$

where variable  $x_i \in \mathbb{R}$  is the spatial variable with  $x_i \in (0, X_{\max})$  for  $i = 0, 1, \dots, M - 1$ , and we call  $M$  the *spatial resolution*; variable  $t_n \in \mathbb{R}$  is the temporal variable with  $t_n \in (0, T_{\max})$  for  $n = 0, 1, \dots, N - 1$ , and we call  $N$  the *temporal resolution*; we use  $T_{\max}, X_{\max}$  to denote the upper bound of the temporal variable and the spatial variable, respectively; variable  $u_i^n$  is a representation of ground truth  $u(x_i, t_n)$  contaminated by noises that follow the normal distribution with mean zero and stand deviation  $\sigma$ :

$$u_i^n = u(x_i, t_n) + \epsilon_i^n, \quad \epsilon_i^n \stackrel{i.i.d.}{\sim} N(0, \sigma^2). \quad (1.2)$$

Here  $u(x, t)$  is the ground truth function, which is determined by an underlying PDE model. And it is assumed to satisfy the following equation:

$$\begin{aligned} \frac{\partial}{\partial t} u(x, t) = & \beta_{00}^* + \sum_{k=0}^{q_{\max}} \sum_{i=1}^{p_{\max}} \beta_{ki}^* \left[ \frac{\partial^k}{\partial x^k} u(x, t) \right]^i + \\ & \sum_{\substack{i+j \leq p_{\max} \\ i, j > 0}} \sum_{\substack{0 < k < l \\ l \leq q_{\max}}} \beta_{k^i, l^j}^* \left[ \frac{\partial^k}{\partial x^k} u(x, t) \right]^i \left[ \frac{\partial^l}{\partial t^l} u(x, t) \right]^j. \end{aligned} \quad (1.3)$$

The left-hand side of the above equation is the first-order partial derivative of the underlying function with respect to the temporal variable  $t$ ,

---

while the right-hand side is the  $p_{\max}$ th order polynomial of the derivatives with respect to the spatial variable  $x$  up to the  $q_{\max}$ th total order. For notational simplicity, we denote the ground truth coefficient vector,  $\beta^* = (\beta_{00}^*, \beta_{01}^*, \beta_{11}^*, \dots, \beta_{q_{\max} p_{\max}}^*)$ , as  $\beta^* = (\beta_1^*, \beta_2^*, \beta_3^*, \dots, \beta_K^*)^\top$ , where  $K = 1 + (p_{\max} + 1)q_{\max} + \frac{1}{2}q_{\max}(q_{\max} + 1)(p_{\max} - 1)!$  is the total number of coefficients on the right hand side. It is noted that, in practice, the majority of the entries in  $\beta^*$  are zero. For instance, in the transport equation  $\frac{\partial}{\partial t}u(x, t) = a \frac{\partial}{\partial x}u(x, t)$  with any  $a \neq 0$ , we only have  $\beta_3^* \neq 0$  and  $\beta_i^* = 0$  for any  $i \neq 3$  (see Olver, 2014, Section 2.2). Therefore, it is reasonable to assume that the coefficient  $\beta^*$  in (1.3) is sparse.

To identify the above model, one needs to overcome two technical challenges. First, the derivatives in (1.3) are unobservable and need to be estimated from the noisy observations. Second, the underlying model – which is presumably simple (i.e., sparse) – need to be identified.

We design our proposed SAPDEMI method to be a two-stage method to identify the underlying PDE models from the noisy data  $\mathcal{D}$ . The first stage is called a *functional estimation stage*, where we estimate all the derivatives from the noisy data  $\mathcal{D}$ , including  $\frac{\partial}{\partial t}u(x, t)$ ,  $\frac{\partial}{\partial x}u(x, t)$ , and so on. In this stage, we first use the cubic splines (Shridhar and Balatoni, 1974) to fit the noisy data  $\mathcal{D}$ , and then we approximate the derivatives of the underlying

## 1.1 Literature Review

---

function via the derivatives of the estimated cubic splines. The second stage is called the *model identification stage*, where we apply the Least Absolute Shrinkage and Selection Operator (Lasso) (Tibshirani, 1996) to identify the derivatives (or their combinations) that should be included in the PDE-based models. To ensure the accuracy, we develop sufficient conditions for correct identification and the asymptotic properties of the identified models. The main tool used in our theoretical analysis is the primal-dual witness (PDW) method (see Hastie et al., 2015, Chapter 11).

The rest of this section is organized as follows. In Section 1.1, we review the existing methods that relates to the PDE identification problem. In Section 1.2, we summarize our contributions.

### 1.1 Literature Review

A pioneering work of identifying the underlying dynamic models from noisy data is Liang and Wu (2008). This method is also a two-stage method, where in the functional estimation stage, Liang and Wu (2008) use the local polynomial regression to estimate the value of the function and its derivatives. Subsequently, in the model identification stage, Liang and Wu (2008) use the least squares method. Following this pioneering work, various extensions have been proposed.

## 1.1 Literature Review

---

The first class of extensions is to modify the functional estimation stage of Liang and Wu (2008), and we can divide these existing extensions into three categories. **(F1)**. In the numerical differentiation category (Wu et al., 2012), the derivative  $\frac{\partial}{\partial x}u(x, t)$  is simply approximated as  $\frac{\partial}{\partial x}u(x, t) \approx \frac{u(x+\Delta x, t)-u(x-\Delta x, t)}{2\Delta x}$ , where  $(x + \Delta x, t)$ ,  $(x - \Delta x, t)$  are two closest points of  $(x, t)$  in  $x$ -domain. The essence of numerical differentiation is to approximate the first order derivative as the slope of a nearby secant line. Although the implementation is easy, the approximation results could be highly biased, because its accuracy highly depends on  $\Delta x$ : a small value of  $\Delta x$  yield large rounding errors in the subtraction (Ueberhuber, 2012); while a large value of  $\Delta x$  leads to poor performance in the estimation of tangent slope by secants. Thus, this naive numerical differentiation is not the preferred due to its bias. **(F2)**. In the basis expansion category, researchers first approximate the unknown functions by the basis expansion methods and then approximate the derivatives of underlying function as the derivatives of the approximated functions. As for the choice of bases, there are multiple options in the existing literature. The most popular basis is the local polynomial basis (see Liang and Wu, 2008; Bär et al., 1999; Schaeffer, 2017; Rudy et al., 2017; Parlitz and Merkwirth, 2000). Another popular choice of basis is the spline basis (see Wu et al., 2012; Xun

## 1.1 Literature Review

---

et al., 2013; Wang et al., 2019). Our proposed method belongs to here. In this category, the major limitation of the existing approaches is the potentially high computational complexity. For instance, the local polynomial basis requires computational complexity of order  $\max\{O(M^2N), O(MN^2)\}$  in the functional estimation stage. However, we will show that our proposed SAPDEMI method only requires  $O(MN)$ . And notice the sample size of the dataset  $\mathcal{D}$  is  $MN$ , so it takes at least  $MN$  numerical operations to read  $\mathcal{D}$ . Consequently, the lowest possible bound in theory is  $O(MN)$ , and this is what achieved by our proposed SAPDEMI method. **(F3)**. In the machine or deep learning category, researchers first fit unknown functions by certain machine/deep learning methods and then approximate the derivatives of underlying functions as the derivatives of the approximated functions. One of the popular machine/deep learning methods is the neural network (NN) approach. For instance, Srivastava et al. (2020) use the artificial neural network (ANN). Its limitation is the potential overfitting as well as the selection of the hyper-parameters.

The second class of extension is to modify the model identification stage of Liang and Wu (2008). The existing methods fall in the framework of the (penalized) least squares method, and we can again divide them into three categories. **(M1)**. In the least squares category, research is done in or-

## 1.1 Literature Review

---

dinary differential equation (ODE) identification (Miao et al., 2009) and PDE identification (Bär et al., 1999; Wu et al., 2012) with problems in reported overfitting issues. **(M2)**. In the  $\ell_2$ -penalized least squares category, Xun et al. (2013); Wang et al. (2019) penalize the smoothness of the unknown function, which is assumed to be in a prescribed reproducing kernel Hilbert space (RKHS). Essentially, this method falls in the framework of the  $\ell_2$ -penalized least squares method. Although this method helps to avoid overfitting by introducing the  $\ell_2$ -penalty, it has limited power to do “model selection”. **(M3)**. In the  $\ell_1$ -penalized least squares method category, Schaeffer (2017) identifies the unknown dynamic models (i.e., functions) through the  $\ell_1$ -penalized least squares method, and later the author discusses the design of an efficient algorithm, which is based on the proximal mapping method. However, the author does not discuss the statistical property of the identified model. Recently, Kang et al. (2019) utilize the similar method as Schaeffer (2017) and demonstrate empirical successes. However, the derivation of the statistical theory is still missing; and this paper fills the gap.

In addition to the  $\ell_2$  or  $\ell_1$  penalized least-squares methods, there are some other methods that have been proposed in the model selection stage. However, these methods are not as widely used as the  $\ell_2/\ell_1$  penalized least-squares method. Some examples are the Akaike information criterion (AIC)

## 1.1 Literature Review

---

in Mangan et al. (2017), smoothly clipped absolute deviation (SCAD) in Lu et al. (2011), and hard-thresholding in Rudy et al. (2017). The first two approaches may lead to NP-hard problems in the numerical implementation. The last one is ad-hoc, and may be hard to analytically analyze. So this paper won't focus on these alternative approaches.

In this paper, our proposed SAPDEMI method applies to the PDE model. Yet, we also acknowledge that in addition to PDEs, there are other non-parametric models. Here, we take PDEs as an initial research project majorly because it is deterministic. Thus, we can compare our identified model with the true model, and show the model notification accuracy. As our initial research project, we prefer PDE over machine learning (ML) models (e.g., neural network, random forest) because PDE offers insight into the physical law. Yet, the ML models usually are black-box methods (Loyola-Gonzalez, 2019). We also prefer PDE over the time series models because it behaves like a “continuous version” of the time series model (Perona et al., 2000; Chen et al., 2018) at a high level. Furthermore, we prefer PDE over the Gaussian process (GP) model because GP restricts its response variables to follow Gaussian distribution (Liu et al., 2020; Wei et al., 2018). Again, although we take PDE as our initial research project, we are open to the aforementioned non-parametric models, as promising

future work.

## 1.2 Our Contributions

We summarize the contributions of our proposed method. **(1)** In the functional estimation stage, our proposed SAPDEMI method is computationally efficient. Specifically, we require computational complexity of order  $O(MN)$ , which is the lowest possible order in this stage. In comparison, the aforementioned local polynomial regression would require the computational complexity of order  $\max\{O(M^2N), O(MN^2)\}$ , which is higher. **(2)** In this paper, for our proposed SAPDEMI method, we established a theoretical guarantee of model identification accuracy. And we didn't find another occurrence of such a result. **(3)** We extend to PDE-based model identification, comparing to ODE-based model identification. The latter has more related work, while the former is not well understood in the literature.

The rest of the paper is organized as follows. In Section 2, we describe the technical details of our proposed SAPDEMI method. In Section 3, we present our main theory, including the sufficient conditions for correct identification, and the statistical properties of the identified models. In Section 4, we conduct numerical experiments to validate the theory from Section 3. In Section 5, we apply SAPDEMI to a real-world case study with

---

data downloaded from the National Aeronautics and Space Administration (NASA). In Section 6, we summarize and discuss some future research.

## 2. Proposed Method: SAPDEMI

The proposed SAPDEMI method is a two-stage method to identify the underlying PDE model from the noisy data  $\mathcal{D}$ . The first stage is called a *functional estimation stage*, where we estimate the function and their derivatives from the noisy data  $\mathcal{D}$  in (1.1) and input them in the second stage. The second stage is called a *model identification stage*, where we identify the underlying PDE-based model.

For notations throughout the paper, scalars are denoted by lowercase letters (e.g.,  $\beta$ ). Vectors are denoted by lowercase bold face letters (e.g.,  $\boldsymbol{\beta}$ ), and its  $i$ th entry is denoted as  $\beta_i$ . Matrices are denoted by uppercase bold-face letter (e.g.,  $\mathbf{B}$ ), and its  $(i, j)$ th entry is denoted as  $B_{ij}$ . For the vector  $\boldsymbol{\beta} \in \mathbb{R}^p$ , its  $k$ th norm is defined as  $\|\boldsymbol{\beta}\|_k := (\sum_{i=1}^p |\beta_i|^k)^{1/k}$ . For the matrix  $\mathbf{B} \in \mathbb{R}^{m \times n}$ , its Frobenius norm is defined as  $\|\mathbf{B}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |B_{ij}|^2}$ . We write  $f(n) = O(g(n))$ , if there exists a  $G \in \mathbb{R}^+$  and a  $n_0$  such that  $|f(n)| \leq Gg(x)$  for all  $n > n_0$ .

This section is organized as follows. In Section 2.1, we introduce the functional estimation stage, and in Section 2.2, we articulate the model

identification stage.

## 2.1 Functional Estimation Stage

In this section, we describe the functional estimation stage of our proposed SAPDEMI method. In this stage, we estimate the functional values and their derivatives from the noisy data  $\mathcal{D}$  in (1.1). These derivatives include the derivatives with respect to the spatial/temporal variable  $x/t$ . We take derivatives with respect to spatial variable  $x$  as an example, and the derivatives with respect to the temporal variable  $t$  can be derived similarly.

The main tool that we use is the cubic spline. Suppose there is a cubic spline  $s(x)$  over the knots  $\{(x_i, u_i^n)\}_{i=0,1,\dots,M-1}$  satisfying the properties in McKinley and Levine (1998): **(1)**.  $s(x) \in C^2[x_0, x_{M-1}]$ , where  $C^2[x_0, x_{M-1}]$  denotes the sets of function whose 0th, first and second derivatives are continuous in  $[x_0, x_{M-1}]$ ; **(2)**. For any  $i = 1, \dots, M-1$ ,  $s(x)$  is a polynomial of degree 3 in  $[x_{i-1}, x_i]$ ; **(3)**. For the two end-points,  $x_0, x_{M-1}$ , we have  $s''(x_0) = s''(x_{M-1}) = 0$ , where  $s''(x)$  is the second derivative of  $s(x)$ .

By fitting data  $\{(x_i, u_i^n)\}_{i=0,1,\dots,M-1}$  (with a general fixed  $n \in \{0, 1, \dots, N-1\}$ ) into the above cubic spline  $s(x)$ , one can solve  $s(x)$  as the minimizer of the following optimization problem:

$$J_\alpha(s) = \alpha \sum_{i=0}^{M-1} w_i [u_i^n - s(x_i)]^2 + (1 - \alpha) \int_{x_0}^{x_{M-1}} s''(x)^2 dx, \quad (2.4)$$

## 2.1 Functional Estimation Stage

where the first term  $\alpha \sum_{i=0}^{M-1} w_i [u_i^n - s(x_i)]^2$  is the weighted sum of squares for residuals, and we take the weight  $w_0 = w_1 = \dots = w_{M-1} = 1$  in our paper. In the second term  $(1 - \alpha) \int_{x_0}^{x_{M-1}} s''(x)^2 dx$ , function  $s''(x)$  is the second derivative of  $s(x)$ , and this term is the penalty of the smoothness.

In the above optimization problem, the parameter  $\alpha \in (0, 1]$  controls the trade off between the goodness of fit and the smoothness of the cubic spline.

By minimizing the above optimization problem, we can get the estimate of  $s(x)$ , together with its first derivative  $s'(x)$  and its second derivative  $s''(x)$ . If the cubic spline approximates the underlying PDE curves well,

we can declare that the derivatives of the underlying dynamic system can be approximated by the derivatives of the cubic spline  $s(x)$ , i.e., we have

$$\widehat{u(x, t_n)} \approx \widehat{s(x)}, \frac{\partial}{\partial x} \widehat{u(x, t_n)} \approx \widehat{s'(x)}, \frac{\partial^2}{\partial x^2} \widehat{u(x, t_n)} \approx \widehat{s''(x)} \quad (\text{Ahlberg et al., 1967;}$$

Rubin and Graves Jr, 1975; Rashidinia and Mohammadi, 2008). Following

the similar procedure to get the derivatives with respect to the spatial

variable  $x$ , we can get the derivatives with respect to the temporal variable

$$t, \text{ i.e., } \frac{\partial}{\partial t} \widehat{u(x_i, t_n)} \text{ for any } i = 0, \dots, M - 1, n = 0, \dots, N - 1.$$

A nice property of the cubic spline is that, there is a closed-form solution for (2.4). First of all, the value of cubic spline  $s(x)$  at the point

$$\{x_0, x_1, \dots, x_{M-1}\}, \text{ i.e., } \widehat{\mathbf{s}} = \left( \widehat{s(x_0)}, \widehat{s(x_1)}, \dots, \widehat{s(x_{M-1})} \right)^\top, \text{ can be solved as}$$

$$\widehat{\mathbf{s}} = [\alpha \mathbf{W} + (1 - \alpha) \mathbf{A}^\top \mathbf{M} \mathbf{A}]^{-1} \alpha \mathbf{W} \mathbf{u}^n. \quad (2.5)$$

## 2.1 Functional Estimation Stage

The above closed-form estimation can be used to approximate the function

that corresponds to the underlying PDE model, i.e.,  $\widehat{\mathbf{s}} \approx \widehat{\mathbf{f}}$

$= \left( \widehat{u(x_0, t_n)}, \widehat{u(x_1, t_n)}, \dots, \widehat{u(x_{M-1}, t_n)} \right)^\top$ . Here,  $\mathbf{W} = \text{diag}(w_0, \dots, w_{M-1}) \in \mathbb{R}^{M \times M}$ , vector  $\mathbf{u}^n = (u_0^n, \dots, u_{M-1}^n)^\top \in \mathbb{R}^M$ , and matrix  $\mathbf{A} \in \mathbb{R}^{(M-2) \times M}$ ,  $\mathbf{M} \in \mathbb{R}^{(M-2) \times (M-2)}$  are

$$\mathbf{A} = \begin{pmatrix} \frac{1}{h_0} & \frac{-1}{h_0} - \frac{1}{h_1} & \frac{1}{h_1} & \dots & 0 & 0 & 0 \\ 0 & \frac{1}{h_1} & \frac{-1}{h_1} - \frac{1}{h_2} & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \frac{1}{h_{M-3}} & \frac{-1}{h_{M-3}} - \frac{1}{h_{M-2}} & \frac{1}{h_{M-2}} \end{pmatrix}, \quad (2.6)$$

$$\mathbf{M} = \begin{pmatrix} \frac{h_0+h_1}{3} & \frac{h_1}{6} & 0 & \dots & 0 & 0 \\ \frac{h_1}{6} & \frac{h_1+h_2}{3} & \frac{h_2}{6} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \frac{h_{M-3}}{6} & \frac{h_{M-3}+h_{M-2}}{3} \end{pmatrix}, \quad (2.7)$$

respectively, with  $h_i = x_{i+1} - x_i$  for  $i = 0, 1, \dots, M-2$ .

For the mathematical details on the derivation of (2.5) from (2.4), and the derivation of first and second order derivatives, please refer to the supplementary material.

The advantage of the cubic spline is that its computational complexity is only a linear polynomial of the sample size  $MN$ .

## 2.1 Functional Estimation Stage

---

**Proposition 2.1.** Given data  $\mathcal{D}$  in (1.1), if we use the cubic spline in (2.4) in the functional estimation stage, the computation complexity is of order

$$\max\{O(p_{\max}MN), O(K^3)\},$$

where  $p_{\max}$  is the highest polynomial order in (1.3),  $M/N$  is the spatial/temporal resolution and  $K$  is the number of covariates in (1.3).

The proof can be found in the online supplementary material.

As suggested by Proposition 2.1, when  $p_{\max}, K \ll M, N$  (which is often the case in practice), it only requires  $O(MN)$  numerical operations in the functional estimation stage. This is the lowest possible order of complexity in this stage because  $MN$  is exactly the number of the sample size of  $\mathcal{D}$  and reading the data is a task of order  $O(MN)$ . So it can be concluded that it is very efficient to use cubic spline because its computational complexity achieves the lowest possible order of complexity.

For comparisons, we discuss the computational complexity of the local polynomial regression, which is widely used in the existing literature (Liang and Wu, 2008; Bär et al., 1999; Schaeffer, 2017; Rudy et al., 2017; Parlitz and Merkwirth, 2000). It turns out that its computational complexity is  $\max\{O(M^2N), O(MN^2), O(p_{\max}MN), O(K^3)\}$ , which is much higher than ours for a generalized polynomial order  $p_{\max}$ . Specifically, if one restricts

## 2.2 Model Identification Stage

the local polynomial regression method to the same order as that of the cubic spline, its computational complexity is

$$\max\{O(M^2N), O(MN^2), O(K^3)\},$$

which is still higher than the cubic spline method in Proposition 2.1. The related proposition and proof is available in the supplementary materials S3 and S8.2. We also summarize the pros and cons of the cubic spline and the local polynomial regression in Table 1.

Table 1: Pros and cons of the cubic spline and the local polynomial regression in the functional estimation stage ( assume that  $p_{\max}, q_{\max}, K \ll M, N$ )

Method	Cubic spline	Local polynomial regression
Pros	Computational complexity is $O(MN)$	Derivatives up to any order
Cons	If higher-than-2 order is required, need extensions beyond cubic splines.	Computational complexity is $\max\{(M^2N), O(MN^2)\}$

## 2.2 Model Identification Stage

In this section, we discuss the model identification stage of our proposed SAPDEMI method. In this stage, we will identify the PDE model in (1.3).

Note that the model in (1.3) can be regarded as a linear regression model whose response variable is the first order derivative with respect to

## 2.2 Model Identification Stage

the temporal variable  $t$ , i.e.,  $\frac{\partial u(x,t)}{\partial t}$ , and the covariates are the derivative(s) with respect to the spatial variable  $x$ , including  $\frac{\partial}{\partial x}u(x_i, t_n), \frac{\partial^2}{\partial x^2}u(x_i, t_n), \dots, \left(\frac{\partial^2}{\partial x^2}u(x_i, t_n)\right)^{p_{\max}}$ . Because we have  $MN$  observations in the dataset  $\mathcal{D}$  in (1.1), the response vector is of length  $MN$ :

$$\nabla_t \mathbf{u} = \left( \widehat{\frac{\partial u(x_0, t_0)}{\partial t}}, \dots, \widehat{\frac{\partial u(x_{M-1}, t_0)}{\partial t}}, \dots, \widehat{\frac{\partial u(x_{M-1}, t_{N-1})}{\partial t}} \right)^\top, \quad (2.8)$$

and design matrix is of dimension  $MN \times K$ :

$$\mathbf{X} = \left( \widehat{\mathbf{x}}_0^0, \widehat{\mathbf{x}}_1^0, \dots, \widehat{\mathbf{x}}_{M-1}^0, \widehat{\mathbf{x}}_1^0, \dots, \widehat{\mathbf{x}}_{M-1}^{N-1} \right)^\top \in \mathbb{R}^{MN \times K}. \quad (2.9)$$

For the above design matrix  $\mathbf{X}$ , its  $(nN + i + 1)$ st row is  $\widehat{\mathbf{x}}_i^n = \left( 1, u(x_i, t_n), \frac{\partial}{\partial x} \widehat{u}(x_i, t_n), \frac{\partial^2}{\partial x^2} \widehat{u}(x_i, t_n), \left( u(x_i, t_n) \right)^2, \dots, \left( \frac{\partial^2}{\partial x^2} \widehat{u}(x_i, t_n) \right)^{p_{\max}} \right)^\top$ . The  $K$  components of  $\widehat{\mathbf{x}}_i^n$  are candidate terms in the PDE model. And all the derivatives listed in (2.8), (2.9) are estimated from the functional estimation stage that is described in Section 2.1.

Next, we use the Lasso to identify the non-zero coefficients in (1.3):

$$\widehat{\boldsymbol{\beta}} = \arg \min_{\boldsymbol{\beta}} \frac{1}{2MN} \|\nabla_t \mathbf{u} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 \quad (2.10)$$

where  $\lambda > 0$  is a turning parameter that controls the trade off of the sparsity of  $\boldsymbol{\beta}$  and the goodness of fit. Given the  $\ell_1$  penalty in (2.10),  $\widehat{\boldsymbol{\beta}}$  will be sparse, i.e., only a few of its entries will likely be non-zero. Accordingly, we can identify the underlying PDE model as

$$\frac{\partial}{\partial t} u(x, t) = \mathbf{x}^\top \widehat{\boldsymbol{\beta}}. \quad (2.11)$$

---

where  $\mathbf{x} = \left(1, u(x, t), \frac{\partial}{\partial x}u(x, t), \frac{\partial^2}{\partial x^2}u(x, t), (u(x, t))^2, \dots, \left(\frac{\partial^2}{\partial x^2}u(x, t)\right)^{p_{\max}}\right)^\top \in \mathbb{R}^K$ . To solve equation (2.10), one can use the coordinate descent method (Beck and Tetrushvili, 2013; Tseng, 2001) and we articulate its details in the online supplementary material.

### 3. Theory on Statistical Properties

The theoretical evaluation is done from two aspects. **(S1)**. First, we check if our identified PDE model contains derivatives that are included in the ‘true’ underlying PDE model. This is called *support set recovery* property. Mathematically, it is to check if  $\text{supp}(\hat{\boldsymbol{\beta}}) \subseteq \text{supp}(\boldsymbol{\beta}^*)$ , where  $\hat{\boldsymbol{\beta}}$  is the minimizer of (2.10),  $\boldsymbol{\beta}^*$  is the ground truth, and  $\text{supp}(\boldsymbol{\beta}) = \{i : \beta_i \neq 0, \forall i, 1 \leq i \leq K\}$  for a general vector  $\boldsymbol{\beta} \in \mathbb{R}^K$ . However, the support recovery depends on the choice of the penalty parameter  $\lambda$ : a large value of  $\lambda$  leads to  $\text{supp}(\hat{\boldsymbol{\beta}}) = \emptyset$  (empty set); while a small value of  $\lambda$  results in a non-sparse  $\hat{\boldsymbol{\beta}}$ . A proper selection of  $\lambda$  hopefully leads to correct recovery of the support set recovery, i.e., we have  $\text{supp}(\hat{\boldsymbol{\beta}}) \subseteq \text{supp}(\boldsymbol{\beta}^*)$ . We will discuss the selection of  $\lambda$  to achieve the above goal in Theorem 3.1. **(S2)**. Second, we are interested in an upper bound of the estimation error of our estimator. Specifically, we consider  $\left\|\hat{\boldsymbol{\beta}}_{\mathcal{S}} - \boldsymbol{\beta}_{\mathcal{S}}^*\right\|_{\infty}$ , where  $\mathcal{S} = \text{supp}(\boldsymbol{\beta}^*)$ , vector  $\hat{\boldsymbol{\beta}}_{\mathcal{S}}$  and  $\boldsymbol{\beta}_{\mathcal{S}}^*$  are sub-vectors of  $\hat{\boldsymbol{\beta}}$  and  $\boldsymbol{\beta}^*$  and only contain elements whose indices are in  $\mathcal{S}$ . An

### 3.1 Conditions in the Theorems

---

upper bound of the above estimation error is discussed in Theorem 3.2.

This section is organized as follows. In Section 3.1, we present conditions for theorems. In Section 3.2, we state two theorems.

#### 3.1 Conditions in the Theorems

In this section, we introduce conditions used in our paper. We begin with three frequently used conditions in  $\ell_1$ -regularized regression models; these conditions provide sufficient conditions for exact sparse recovery (see Hastie et al., 2015, Chapter 11). Subsequently, we introduce three conditions that are widely used in cubic splines-based functional estimation (see Silverman, 1984, (2.5)-(2.8)).

**Condition 3.1** (Mutual Incoherence Condition). For some *incoherence parameter*  $\mu \in (0, 1]$  and  $P_\mu \in [0, 1]$ , we have  $\mathbb{P} \left( \|\mathbf{X}_{S^c}^\top \mathbf{X}_S (\mathbf{X}_S^\top \mathbf{X}_S)^{-1}\|_\infty \leq 1 - \mu \right) \geq P_\mu$ , where matrix  $\mathbf{X}_{S^c}$  is the complement of  $\mathbf{X}_S$ .

**Condition 3.2** (Minimal Eigenvalue Condition). There exists some constant  $C_{\min} > 0$  such that  $\Lambda_{\min} \left( \frac{1}{NM} \mathbf{X}_S^\top \mathbf{X}_S \right) \geq C_{\min}$ , almost surely. Here  $\Lambda_{\min}(\mathbf{A})$  denotes the minimal eigenvalue of a square matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ . This condition can be considered as a strengthened version of invertibility condition (see Hastie et al., 2015, Chapter 11).

### 3.1 Conditions in the Theorems

**Condition 3.3** (Knots c.d.f. Convergence Condition). Suppose for the sequence of the empirical distribution function over the design points  $a = x_0 < \dots < x_{M-1} = b$  with different sample size  $M$  is denoted as  $F_M(x)$ , i.e., we have  $F_M(x) = \frac{1}{M} \sum_{i=0}^{M-1} \mathbb{1}\{x_i \leq x\}$ , there exists an absolutely continuous distribution function  $F$  on  $[a, b]$  such that  $F_M \rightarrow F$  uniformly as  $M \rightarrow +\infty$ . Here  $\mathbb{1}\{A\}$  is the indicator of event  $A$ . A similar condition holds for the temporal variable: suppose the sequence of empirical distribution function over the design points  $\bar{a} = t_0 < \dots < t_{N-1} = \bar{b}$  with different sample size  $N$  is denoted as  $G_N(x)$ , there exists an absolutely continuous distribution function  $G$  on  $[\bar{a}, \bar{b}]$  such that  $G_N \rightarrow G$  uniformly as  $N \rightarrow +\infty$ .

**Condition 3.4** (Knots p.d.f. Convergence Condition). Suppose the first derivative of the function  $F, G$  (defined in Condition 3.3) is denoted as  $f, g$ , respectively, then we have

$$0 < \inf_{[x_0, x_{M-1}]} f \leq \sup_{[x_0, x_{M-1}]} f < +\infty \quad \text{and} \quad 0 < \inf_{[t_0, t_{N-1}]} g \leq \sup_{[t_0, t_{N-1}]} g < +\infty,$$

and  $f, g$  also have bounded first derivatives on  $[x_0, x_{M-1}], [t_0, t_{N-1}]$ .

**Condition 3.5** (Gentle Decrease of Smoothing Parameter Condition). Suppose that  $\zeta(M) = \sup_{[x_0, x_{M-1}]} |F_M - F|$ , The smoothing parameter  $\alpha$  in (2.4) depends on  $M$  in such a way that  $\alpha \rightarrow 0$  and  $\alpha^{-1/4} \zeta(M) \rightarrow 0$  as  $M \rightarrow +\infty$ .

A similar condition also hold for the temporal variable.

### 3.2 Main Theory

In the first theorem, we develop the lower bound of  $\lambda$  to realize the correct recovery of the support set, i.e.,  $\mathcal{S}(\hat{\beta}) \subseteq \mathcal{S}(\beta^*)$ .

**Theorem 3.1.** Provided with the data in (1.1) and suppose the conditions in Lemma S6.1 and Corollary S6.1 (see details in the online supplementary material) hold and Condition 3.1 - 3.5 also hold, if we take  $M = O(N)$ , then there exists a constant  $\mathcal{C}_{(\sigma, \|u\|_{L^\infty(\Omega)})} > 0$ , which is independent of spatial resolution  $M$  and temporal resolution  $N$ , such that if we set the cubic spline smoothing parameter with the spatial variable  $x$  in (2.4) as  $\alpha = O\left((1 + M^{-4/7})^{-1}\right)$ , set the cubic spline smoothing parameter with temporal variable  $t$  as  $\bar{\alpha} = O\left((1 + N^{-4/7})^{-1}\right)$ , and set turning parameter

$$\lambda \geq \mathcal{C}_{(\sigma, \|u\|_{L^\infty(\Omega)})} \frac{\sqrt{K} \log(N)}{\mu N^{3/7-r}}, \quad (3.12)$$

to identify the PDE model in (2.10) for some  $r \in (0, \frac{3}{7})$  with sufficient large  $N$ , then with probability greater than  $P_\mu - \underbrace{O(Ne^{-N^r})}_{P'}$ , we can have  $\mathcal{S}(\hat{\beta}) \subseteq \mathcal{S}(\beta^*)$ . Here  $K$  is the number of columns of the design matrix  $\mathbf{X}$  in (2.10), and  $\mu, P_\mu$  are defined in Condition 3.1.

The proof of the above theorem can be found in the online supplementary material. For the interest of page limitation, we put some lemmas in supplementary material, whose conditions is standardized in cubic splines.

## 3.2 Main Theory

---

The above theorem provides the lower bound of  $\lambda$  to realize the correct recovery of the support set. As indicated by (3.12), the lower bound is affected by several factors. First, it is affected by the temporal resolution  $N$ : as  $N$  increases, there is more flexibility in tuning this penalty parameter  $\lambda$ . Second, the lower bound in (3.12) is affected by the incoherence parameter  $\mu$ : if  $\mu$  is small, then the lower bound increases. This is because small  $\mu$  means that the group of feature variable candidates are similar to each other. This phenomenon is called *multicollinearity*. If that happens, we will have very limited choice to select  $\lambda$ . However, we can not enlarge the value of  $\mu$ , since it is decided by the dataset  $\mathcal{D}$  itself (see Condition 3.1). Third, this lower bound in (3.12) is affected by the number of columns of the matrix  $\mathbf{X}$ . If its number of columns is very large, then it requires larger  $\lambda$  to identify significant feature variables among potential feature variables.

We also want to point out that, the probability  $P_\mu - P'$  converges to  $P_\mu$  as  $N \rightarrow +\infty$ . This limiting probability  $P_\mu$  is determined by the data  $\mathcal{D}$  (see Condition (3.1)). So when  $N$  is very large, our proposed SAPDEMI method can realize  $\mathcal{S}(\hat{\beta}) \subseteq \mathcal{S}(\beta^*)$  with probability close to  $P_\mu$ .

In the second theorem, we develop upper bound of estimating error.

**Theorem 3.2.** Suppose the conditions in Theorem 3.1 hold, then with probability greater than  $1 - O(Ne^{-Nr}) \rightarrow 1$ , there exist a  $\dot{N} > 0$ , such that

when  $N > \dot{N}$ , we have

$$\left\| \widehat{\boldsymbol{\beta}}_{\mathcal{S}} - \boldsymbol{\beta}_{\mathcal{S}}^* \right\|_{\infty} \leq \sqrt{K} C_{\min} \left( \sqrt{K} \mathcal{C}_{(\sigma, \|u\|_{L^{\infty}(\Omega)})} \frac{\log(N)}{N^{3/7-r}} + \lambda \right),$$

where  $K$  is the number of columns of the matrix  $\mathbf{X}$ ,  $\mathcal{S} := \{i : \beta_i^* \neq 0, \forall i = 1, \dots, K\}$  and vectors  $\widehat{\boldsymbol{\beta}}_{\mathcal{S}}$  and  $\boldsymbol{\beta}_{\mathcal{S}}^*$  are the subvectors of  $\widehat{\boldsymbol{\beta}}$  and  $\boldsymbol{\beta}^*$  that only contain elements whose indices are in  $\mathcal{S}$ . Viewing from this theorem, we can see that when  $N \rightarrow +\infty$ , the error bound will convergence to 0.

The proof can be found in the online supplementary material. From the above theorem, we can see that, the estimation error bound for the  $\ell_{\infty}$ -norm of the coefficient error in (3.13) consists of two components. The first component is affected by the temporal resolution  $N$ , and the number of feature variable candidates  $K$ . As  $N \rightarrow +\infty$ , this first component convergence to 0 without explicit dependence on the choice of feature variable selected from (2.10). The second component is  $\sqrt{K} C_{\min} \lambda$ . When  $N$  increases to  $+\infty$ , this second component will also converge to 0. This is because, as stated in Theorem 3.1, we find that when  $N \rightarrow +\infty$ , the lower bound of  $\lambda$  — which realizes correct support recovery — converges to 0. So the accuracy of the coefficient estimation will improve if we increase  $N$ .

By combining Theorems 3.1 and 3.2, we find that when the minimum absolute value of the nonzero entries of  $\boldsymbol{\beta}^*$  is large enough, with the adequate choice of  $\lambda$ , the exact recovery can be guaranteed. Mathematically

---

speaking, when  $\min_{i \in \mathcal{S}} |(\boldsymbol{\beta}_{\mathcal{S}}^*)_i| > \sqrt{K} C_{\min} \left( \sqrt{K} \mathcal{C}_{(\sigma, \|u\|_{L^\infty(\Omega)})} \frac{\log(N)}{N^{3/7-r}} + \lambda \right)$ , – where  $(\boldsymbol{\beta}_{\mathcal{S}}^*)_i$  refers to the  $i$ th element in vector  $\boldsymbol{\beta}_{\mathcal{S}}^*$  – we will have a correct signed-support of  $\widehat{\boldsymbol{\beta}}$ . This helps for the selection of the penalty parameters  $\lambda$ . Besides, the solution paths plot also helps with the selection of the penalty parameters  $\lambda$ , and we will discuss it in Section 4 with examples.

#### 4. Numerical Examples

We conduct numerical experiments to verify the computational efficiency and the statistical accuracy of our proposed SAPDEMI method.

Our examples are based on (1) the transport equation, (2) the inviscid Burgers' equation, and (3) the viscous Burgers' equation. We select these three PDE models as representatives because all these PDE models play fundamental roles in modeling physical phenomenon and demonstrate characteristic behaviors shared by a more complex system, such as dissipation and shock-formation (Haberman, 1983). In addition to wide applications, they also cover a wide range of categories, including first-order PDE, second-order PDE, linear PDE, and non-linear PDE. And these wide range of categories cover most of the PDEs frequently seen in practice. Furthermore, the difficulty to identify the above PDE models increases from the first example—the transport equation—to the last example—the viscous

#### 4.1 Example 1: Transport Equation

---

Burgers' equation. We set  $p_{\max} = 2, q_{\max} = 2$  in (1.3) for all the three numerical examples (see the full formula of the full model in the supplementary material), i.e., we identify the PDE model from the full model

For the computational efficiency, the results of these three examples are the same, so we only present the result for the first example. We also verify Condition 3.1 - 3.5 of the above three examples. The details of the verification can be found in the online supplementary material.

#### 4.1 Example 1: Transport Equation

The PDE problem to be studied in this section is the transport equation. It is a linear, first-order PDE model. Given its simplicity and straight physical meaning, it has been widely used to model the concentration of a substance flowing in a fluid at a constant rate. For example, it can be used to model a pollutant, in a uniform fluid flow that is moving with velocity  $a$  (Olver, 2014, Section 2.2):

$$\begin{cases} \frac{\partial}{\partial t}u(x, t) = a \frac{\partial}{\partial x}u(x, t), & \forall 0 \leq x \leq X_{\max}, 0 \leq t \leq T_{\max}; \\ u(x, 0) = f(x). \end{cases} \quad (4.13)$$

Here  $a \in \mathbb{R}$  is a fixed, non-zero constant, known as the *wave speed*. In this section, we set  $a = -2, f(x) = 2 \sin(4x), X_{\max} = 1, T_{\max} = 0.1$ . Given this settings, there is a closed-form solution, which is  $u(x, t) = 2 \sin(4x - 8t)$ .

#### 4.1 Example 1: Transport Equation

The dynamic pattern of the above transport equation is visualized in Fig. 1, where subfigures (a), (b), and (c) are the ground truth, and noisy observations under  $\sigma = 0.05$  and  $\sigma = 0.1$ , respectively. From this figure, we can see that the larger the noise, the more un-smoothed the shape of the transport equation would be, which potentially leads to more difficulties in the PDE model identification.

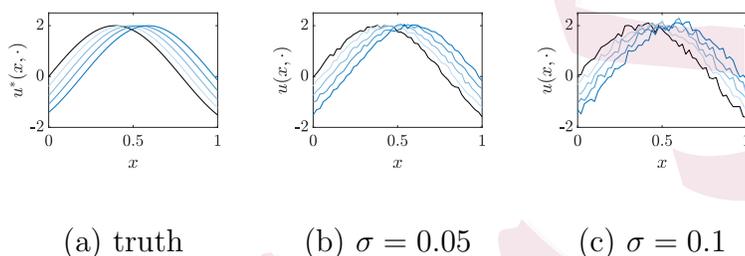


Figure 1: Noisy/True curves from (4.13) ( $M = N = 100$ )

First of all, let us take a look at the computational complexity of the functional estimation stage. We select the local polynomial regression as a benchmark and visualize the number of numerical operations of the above two methods in Fig. 2, where the x-axis is  $\log(M)$  or  $\log(N)$ , and the y-axis is the logarithm of the number of numerical operations. In Fig. 2, two scenarios are discussed: (1)  $M$  is fixed as 20 and  $N$  varies from 200 to 2000; (2)  $N$  is fixed as 20 and  $M$  varies from 200 to 2000. As we can see from Fig. 2 that, as  $M$  or  $N$  increases, the number of numerical operations in the functional estimation stage becomes larger. We find that

#### 4.1 Example 1: Transport Equation

the cubic splines method needs fewer numerical operations, compared with the local polynomial regression. Furthermore, if we conduct a simple linear regression of the four lines in Fig. 2, we find that in (a), the slope of the cubic spline is 0.9998, and as  $N$  goes to infinity, the slope will get closer to 1. This validates that the computational complexity of the cubic splines-based method is of order  $O(N)$  when  $M$  is fixed. A similar story happens to (b), so we numerically verify the computational complexity of cubic spline is of order  $O(MN)$ . Similarly, for local polynomial, we can also numerically validate its computational complexity, which is  $\max\{O(M^2N), O(MN^2)\}$ .

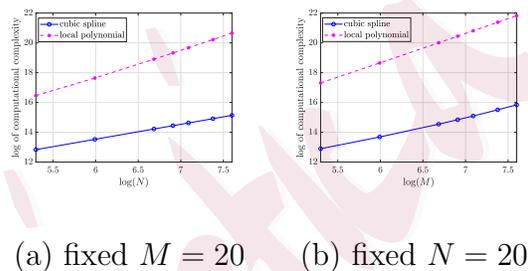


Figure 2: Computational complexity of cubic spline & local polynomial

We now numerically verify that with high probability, our SAPDEMI can correctly identify the underlying PDE models. From the formula of the transport equation in equation (4.13), we know that the correct feature variable is only  $\frac{\partial}{\partial x}u(x, t)$ . While other feature variables should not be identified. We discuss the identification accuracy under different sample sizes and magnitudes of noises. We find that the accuracy stays at 100%.

## 4.2 Example 2: Inviscid Burgers' Equation

To explain the high accuracy, we plot the solution paths in Fig. 3 under different  $\sigma$ , i.e.,  $\sigma = 0.01, 0.1, 1$ . From Fig. 3, we find one can increase  $\lambda$  to overcome this difficulty, and thus achieve correct PDE identification.

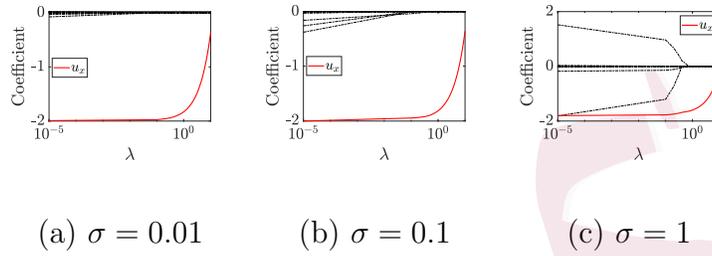


Figure 3: Solution paths in the transport equation under different  $\sigma$  and  $M = N = 100$ . Notation  $u_x$  is the simplification of  $\frac{\partial}{\partial x}u(x, t)$ .

## 4.2 Example 2: Inviscid Burgers' Equation

In this section, we investigate the inviscid Burgers' equation (see Olver, 2014, Section 8.4). It is a representative of first-order nonlinear PDE and has been frequently used in applied mathematics, such as fluid mechanics, nonlinear acoustics, gas dynamics, and traffic flow. This PDE model was first introduced by Harry Bateman in 1915 and later studied by Johannes Martinus Burgers in 1948 (Whitham, 2011). The formula of the inviscid

## 4.2 Example 2: Inviscid Burgers' Equation

Burger's equation is listed below.

$$\begin{cases} \frac{\partial}{\partial t}u(x, t) = -\frac{1}{2}u(x, t)\frac{\partial}{\partial x}u(x, t) \\ u(x, 0) = f(x) & 0 \leq x \leq X_{\max} \\ u(0, t) = u(1, t) = 0 & 0 \leq t \leq T_{\max} \end{cases}, \quad (4.14)$$

where we set  $f(x) = \sin(2\pi x)$ ,  $X_{\max} = 1$ ,  $T_{\max} = 0.1$ . Fig. 4(a), (b), and (c) show the ground truth, noisy observations under  $\sigma = 0.05, 0.1$ , respectively.

Compared with our first example (transport equation in (4.13)), the inviscid

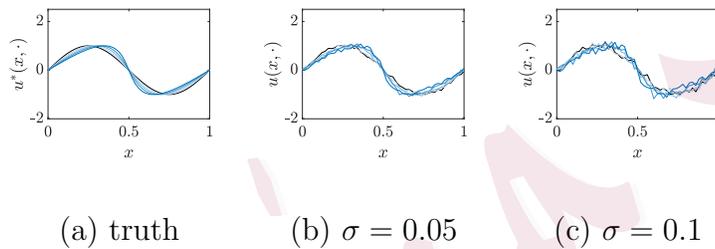


Figure 4: Noisy/True curves from (4.14) ( $M = 50$ ,  $N = 50$ )

Burgers' equation can be regarded as its extension from linear transport equation to nonlinear transport equation. Specifically, if we set  $a$  in (4.13) as  $a = -\frac{1}{2}u(x, t)$ , then (4.13) is equivalent to (4.14). In the literature, this PDE model is considerably more challenging than the linear transport PDE in (4.13): the wave speed in (4.13) only depends on the spatial variable  $x$ , while the wave speed in (4.14) depends both on the spatial variable  $x$ , but also on the size of the disturbance  $u(x, t)$ . Given the complicated wave speed in (4.14), it can model more complicated dynamic patterns. For example,

### 4.3 Example 3: Viscous Burgers' Equation

larger waves move faster, and overtake smaller, slow-moving waves.

In this example, SAPDEMI can correctly identify, with accuracy above 99% (see Fig. 8(a)). The effect of  $\sigma$  also reflects in Fig. 5, where the length of  $\lambda$ -interval for correct identification decreases as  $\sigma$  increases.

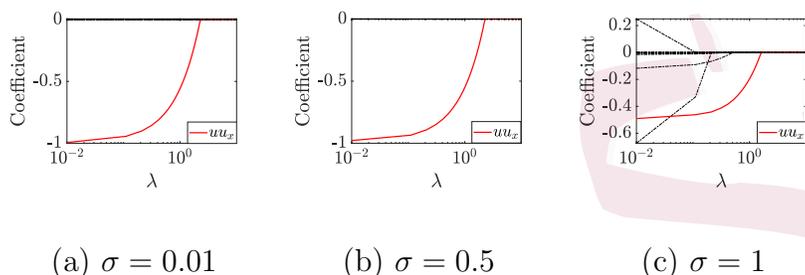


Figure 5: Solution paths in inviscid Burger's equation under different  $\sigma$ 's and  $M = N = 100$ .  $u$  and  $u_x$  are simplifications of  $u(x, t)$  and  $\frac{\partial}{\partial x}u(x, t)$ .

### 4.3 Example 3: Viscous Burgers' Equation

In this section, we investigate more challenging viscous Burgers' equation (see Olver, 2014, Section 8.4), which is a fundamental second-order semilinear PDE. It is frequently employed to model physical phenomena in fluid dynamics (Bonkile et al., 2018) and nonlinear acoustic in dissipative media (Rudenko and Soluian, 1975). For example, in the fluid and gas dynamics, one can interpret the term  $\nu \frac{\partial^2}{\partial x^2}u(x, t)$  as modeling the effect of viscosity (Olver, 2014, Section 8.4). Thus, the viscous Burgers' equation represents a version of the equations of viscous fluid flows, including the celebrated

### 4.3 Example 3: Viscous Burgers' Equation

and widely applied Navier-Stokes equations (Whitham, 2011).

$$\left\{ \begin{array}{l} \frac{\partial u(x,t)}{\partial t} = -\frac{1}{2}u(x,t)\frac{\partial}{\partial x}u(x,t) + \nu\frac{\partial^2}{\partial x^2}u(x,t) \\ u(x,0) = f(x) \\ u(0,t) = u(1,t) = 0 \end{array} \right. \quad \begin{array}{l} 0 \leq x \leq X_{\max} \\ 0 \leq t \leq T_{\max} \end{array} , \quad (4.15)$$

where we set  $f(x) = \sin^2(4\pi x) + \sin^3(2\pi x)$ ,  $X_{\max} = 1$ ,  $T_{\max} = 0.1$ ,  $\nu = 0.1$ .

Fig. 6 shows the corresponding curves, where (a), (b), and (c) are the ground truth, noisy observations under  $\sigma = 0.05$  and  $\sigma = 0.1$ , respectively.

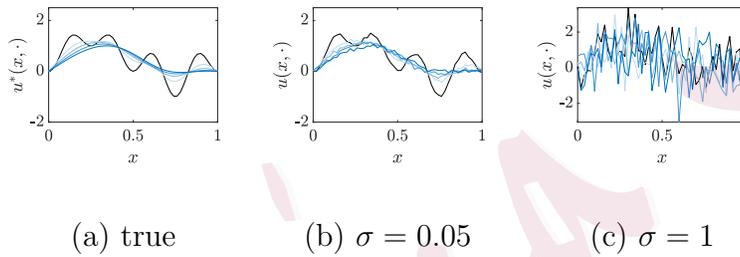


Figure 6: Noisy/True curves from (4.15) ( $M = 50$ ,  $N = 50$ ) .

Compare with the previous two PDE models (transport equation in (4.13) and inviscid Burgers' equation in (4.14)), the above PDE is more complicated and challenging. This is because the viscous Burgers' equation involves not only the first order derivative, but also the second order derivatives. And it is a sufficiently complicate example in our simulations.

Based on Fig. 8(b), we conclude that with high probability, our proposed SAPDEMI can correctly identify the underlying viscous Burgers' equation, with the reasons as follows. When  $M = N = 200$  or 150, the ac-

### 4.3 Example 3: Viscous Burgers' Equation

curacy stays above 90% for all levels of  $\sigma \in [0.01, 1]$ . When  $M = N = 100$ , the accuracy are above 70% when  $\sigma \in [0.01, 0.5]$ , and reduces to 50% when  $\sigma = 1$ , which makes sense because as reselected by Fig. 7, when  $\sigma$  increase from 0.01 to 1, the length of  $\lambda$ -interval for correct identification decreases, which make it difficult to realize correct identification. So if we encounter a heavily noised dataset  $\mathcal{D}$ , a larger sample size is preferred.

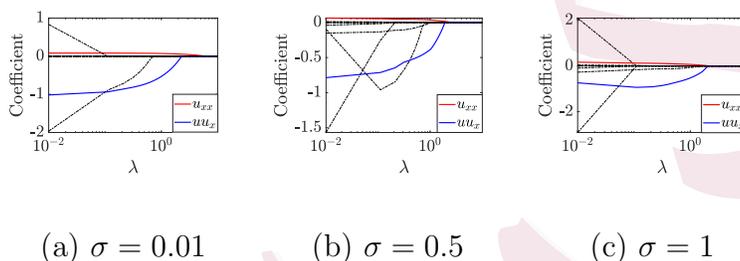


Figure 7: Solution paths in the viscous Burger's equation under different  $\sigma$  and  $M = N = 100$ .  $u_{xx}, uu_x$  stand for  $u(x, t) \frac{\partial}{\partial x} u(x, t), \frac{\partial^2}{\partial x^2} u(x, t)$ .

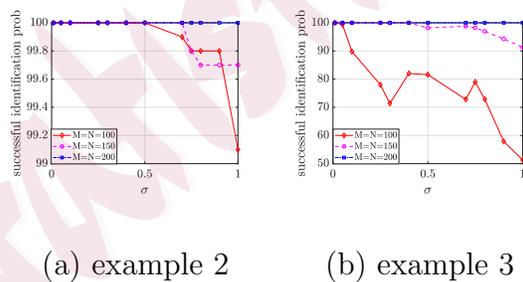
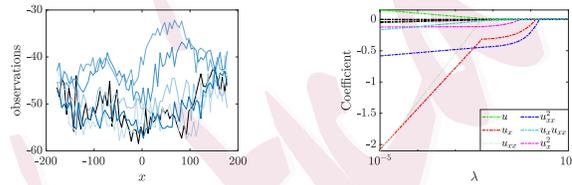


Figure 8: Curves of successful identification probability

## 5. Case Study

In this section, we apply SAPDEMI to a real-world dataset, which is a subset of the Cloud-Aerosol Lidar and Infrared Pathfinder Satellite Observations (CALIPSO) dataset downloaded from the NASA. The CALIPSO reports the monthly mean of temperature in 2017 at 34°N and 110.9418 meters above earth surface over a uniform spatial grid from 180°W to 180°E with equally spaced 5° interval. The missing data is handled either by direct imputation or the instrument methods Chen et al. (2018, 2021); Chen and Fang (2019); Chen et al. (2018).



(a) observed temperature (b) solution path

Figure 9: Visualization and identification of the CALIPSO data

The identified PDE model ( $N = 12, M = 72$ ), reasonably speaking, is

$$\frac{\partial}{\partial t}u(x, t) = a \frac{\partial}{\partial x}u(x, t) + b \left( \frac{\partial^2}{\partial x^2}u(x, t) \right)^2, \quad (5.16)$$

where the values of  $a, b$  can be estimated by a simple linear regression only on the selected derivatives, i.e.,  $\frac{\partial}{\partial x}u(x, t)$  and  $\left( \frac{\partial^2}{\partial x^2}u(x, t) \right)^2$ . The linear regression suggests reasonable values of  $a = -0.2505, b = 1.7648$ . It should

---

be noticed that our paper mainly focuses on the identification, i.e., identify  $\frac{\partial}{\partial x}u(x, t)$  and  $\left(\frac{\partial^2}{\partial x^2}u(x, t)\right)^2$  from many derivatives candidates, instead of coefficients estimation, so we use  $a = -0.2505$  and  $b = 1.7648$  as reference.

Because the CALIPSP is a real-world dataset, we do not know the ground truth of the underlying PDE model. But here we provide some justifications. First, from the solution path in Fig. 9(b), we found the coefficients of  $\frac{\partial}{\partial x}u(x, t)$  and  $\left(\frac{\partial^2}{\partial x^2}u(x, t)\right)^2$  remain non-zeros under  $\lambda = 0.05$ , while other coefficients are all zero. Second, the identified PDE model in (5.16) fits well to the training data (see Fig. 10 (a.1)-(a.3)). Third, the identified PDE model in (5.16) predicts well in the testing data (see Fig 10 (b.1)-(b.3)). Considering the above reasons, we think our proposed SAPDEMI method performs well in the CALIPSO dataset since it adequately predicts the feature values in 2018.

## 6. Conclusion

In this paper, we propose the SAPDEMI method to identify underlying PDE models from noisy data. The proposed method is computationally efficient, and we can derive a statistical guarantee on its performance. We realize there are lots of promising future research directions, including but not limited to multivariate spatial variable ( $\mathbf{x} \in \mathbb{R}^d$  with  $d \geq 2$ ) (Haber-

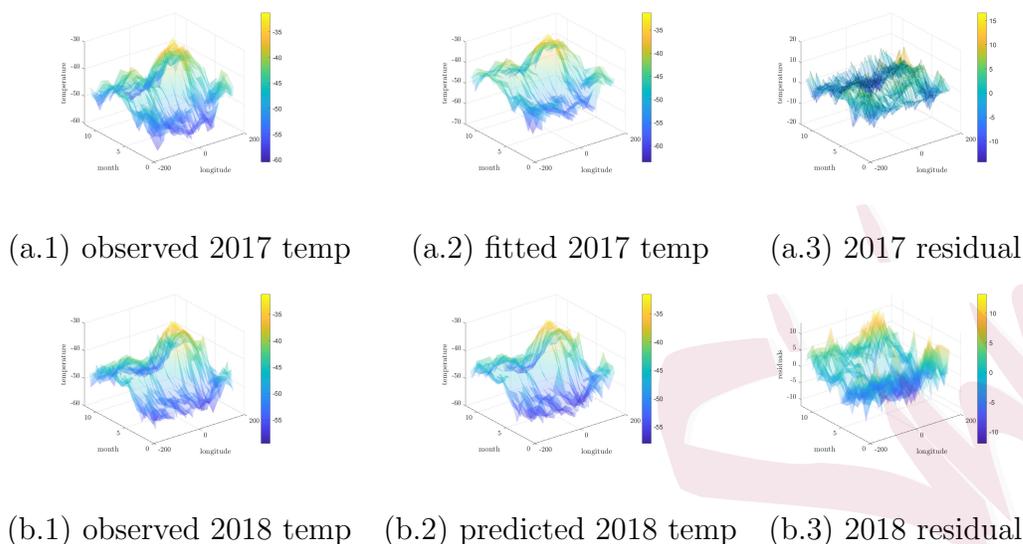


Figure 10: 3D surface plots of the temperatures in 2017/2018.

mann and Kindermann, 2007), interactions between spatial and temporal variables. In our paper, we aim at showing the methodology to solve the PDE identification, so we skip discuss the above future research and our paper should provide a good starting point for these further research.

### Acknowledgements

The authors gratefully acknowledge NSF grant DMS-2015405, DMS-2015363, and the TRIAD (a part of the TRIPODS program at NSF and locates at Georgia Tech, enabled by the NSF grant CCF-1740776).

## REFERENCES

---

### References

- Ahlberg, J., J. Walsh, R. Bellman, and E. N. Nilson (1967). *The theory of splines and their applications*. Academic press.
- Bär, M., R. Hegger, and H. Kantz (1999). Fitting partial differential equations to space-time dynamics. *Physical Review E* 59(1), 337.
- Beck, A. and L. Tetruashvili (2013). On the convergence of block coordinate descent type methods. *SIAM journal on Optimization* 23(4), 2037–2060.
- Bonkile, M. P., A. Awasthi, C. Lakshmi, V. Mukundan, and V. Aswin (2018). A systematic literature review of burgers' equation with recent advances. *Pramana* 90(6), 1–21.
- Chen, J. and F. Fang (2019). Semiparametric likelihood for estimating equations with non-ignorable non-response by non-response instrument. *Journal of Nonparametric Statistics* 31(2), 420–434.
- Chen, J., F. Fang, and Z. Xiao (2018). Semiparametric inference for estimating equations with nonignorable missing covariates. *Journal of Nonparametric Statistics* 30(3), 796–812.
- Chen, J., D. Ohlssen, and Y. Zhou (2018). Functional mixed effects model for the analysis of dose-titration studies. *Statistics in Biopharmaceutical Research* 10(3), 176–184.
- Chen, J., J. Shao, and F. Fang (2021). Instrument search in pseudo-likelihood approach for nonignorable nonresponse. *Annals of the Institute of Statistical Mathematics* 73(3), 519–533.

## REFERENCES

---

- Chen, J., B. Xie, and J. Shao (2018). Pseudo likelihood and dimension reduction for data with nonignorable nonresponse. *Statistical Theory and Related Fields* 2(2), 196–205.
- Haberman, R. (1983). *Elementary applied partial differential equations*, Volume 987. Prentice Hall Englewood Cliffs, NJ.
- Habermann, C. and F. Kindermann (2007). Multidimensional spline interpolation: Theory and applications. *Computational Economics* 30(2), 153–169.
- Hastie, T., R. Tibshirani, and M. Wainwright (2015). *Statistical learning with sparsity: the lasso and generalizations*. CRC press.
- Kang, S. H., W. Liao, and Y. Liu (2019). IDENT: Identifying differential equations with numerical time evolution. *arXiv preprint arXiv:1904.03538*.
- Lagergren, J. H., J. T. Nardini, G. Michael Lavigne, E. M. Rutter, and K. B. Flores (2020). Learning partial differential equations for biological transport models from noisy spatio-temporal data. *Proceedings of the Royal Society A* 476(2234), 20190800.
- Liang, H. and H. Wu (2008). Parameter estimation for differential equation models using a framework of measurement error in regression models. *Journal of the American Statistical Association* 103(484), 1570–1583.
- Liu, H., Y.-S. Ong, X. Shen, and J. Cai (2020). When gaussian process meets big data: A review of scalable gps. *IEEE transactions on neural networks and learning systems* 31(11).
- Loyola-Gonzalez, O. (2019). Black-box vs. white-box: Understanding their advantages and

## REFERENCES

---

- weaknesses from a practical point of view. *IEEE Access* 7, 154096–154113.
- Lu, T., H. Liang, H. Li, and H. Wu (2011). High-dimensional ODEs coupled with mixed-effects modeling techniques for dynamic gene regulatory network identification. *Journal of the American Statistical Association* 106(496), 1242–1258.
- Mangan, N. M., J. N. Kutz, S. L. Brunton, and J. L. Proctor (2017). Model selection for dynamical systems via sparse regression and information criteria. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 473(2204), 20170009.
- McKinley, S. and M. Levine (1998). Cubic spline interpolation. *College of the Redwoods* 45(1), 1049–1060.
- Miao, H., C. Dykes, L. M. Demeter, and H. Wu (2009). Differential equation modeling of HIV viral fitness experiments: model identification, model selection, and multimodel inference. *Biometrics* 65(1), 292–300.
- Olver, P. J. (2014). *Introduction to partial differential equations*. Springer.
- Parlitz, U. and C. Merkwirth (2000). Prediction of spatiotemporal time series based on reconstructed local states. *Physical review letters* 84(9), 1890.
- Perona, P., A. Porporato, and L. Ridolfi (2000). On the trajectory method for the reconstruction of differential equations from time series. *Nonlinear Dynamics* 23(1), 13–33.
- Rashidinia, J. and R. Mohammadi (2008). Non-polynomial cubic spline methods for the solution of parabolic equations. *International Journal of Computer Mathematics* 85(5), 843–850.

## REFERENCES

---

- Rubin, S. G. and R. A. Graves Jr (1975). A cubic spline approximation for problems in fluid mechanics. *NASA STI/Recon Technical Report N 75*, 33345.
- Rudenko, O. and S. Soluian (1975). The theoretical principles of nonlinear acoustics. *Moscow Izdatel Nauka*.
- Rudy, S. H., S. L. Brunton, J. L. Proctor, and J. N. Kutz (2017). Data-driven discovery of partial differential equations. *Science Advances* 3(4), e1602614.
- Schaeffer, H. (2017). Learning partial differential equations via data discovery and sparse optimization. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 473(2197), 20160446.
- Shridhar, M. and N. Balatoni (1974). A generalized cubic spline technique for identification of multivariable systems. *Journal of Mathematical Analysis and Applications* 47(1), 78–90.
- Silverman, B. W. (1984). Spline smoothing: the equivalent variable kernel method. *The Annals of Statistics*, 898–916.
- Srivastava, K., M. Ahlawat, J. Singh, and V. Kumar (2020). Learning partial differential equations from noisy data using neural networks. In *Journal of Physics: Conference Series*, Volume 1655, pp. 012075. IOP Publishing.
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* 58(1), 267–288.
- Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable

## REFERENCES

---

- minimization. *Journal of optimization theory and applications* 109(3), 475–494.
- Ueberhuber, C. W. (2012). *Numerical computation 1: methods, software, and analysis*. Springer Science & Business Media.
- Wang, D., K. Liu, and X. Zhang (2019). Spatiotemporal thermal field modeling using partial differential equations with time-varying parameters. *IEEE Transactions on Automation Science and Engineering*.
- Wang, H., D. Yang, and S. Zhu (2014). Inhomogeneous Dirichlet boundary-value problems of space-fractional diffusion equations and their finite element approximations. *SIAM Journal on Numerical Analysis* 52(3), 1292–1310.
- Wei, S.-b., W. Wang, N. Liu, J. Chen, X.-y. Guo, R.-b. Tang, R.-h. Yu, D.-y. Long, C.-h. Sang, C.-x. Jiang, et al. (2018). U-shaped association between serum free triiodothyronine and recurrence of atrial fibrillation after catheter ablation. *Journal of Interventional Cardiac Electrophysiology* 51(3), 263–270.
- Whitham, G. B. (2011). *Linear and nonlinear waves*. John Wiley & Sons.
- Wu, H., H. Xue, and A. Kumar (2012). Numerical discretization-based estimation methods for ordinary differential equation models via penalized spline smoothing with applications in biomedical research. *Biometrics* 68(2), 344–352.
- Xun, X., J. Cao, B. Mallick, A. Maity, and R. J. Carroll (2013). Parameter estimation of partial differential equation models. *Journal of the American Statistical Association* 108(503), 1009–1020.