Statistica Sinica **34** (2024), 1461-1482
doi:https://doi.org/10.5705/ss.202022.0061

IDENTIFICATION OF PARTIAL DIFFERENTIAL EQUATIONS-BASED MODELS FROM NOISY DATA VIA SPLINES

Yujie Zhao*[1], Xiaoming Huo[2] and Yajun Mei[2]

[1]*Merck & Co., Inc. and* [2]*Georgia Tech*

Abstract: We propose a two-stage method called *Spline-Assisted Partial Differential Equations-based Model Identification* that can be used to identify models based on partial differential equations (PDEs) from noisy data. In the first stage, we employ 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 the product of a constant and the sample size, which is the lowest possible order of computational complexity. In the second stage, we apply the least absolute shrinkage and selection operator to identify the underlying PDE-based model. Statistical properties are developed, including the model identification accuracy. We validate our theory using numerical examples and a real-data case study based on an National Aeronautics and Space Administration data set.

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

1. Introduction

Partial differential equations (PDEs) are widely used to model physical processes in fields such as engineering (Wang, Liu and Zhang (2019)), physics (Xun et al. (2013)), and biology (Lagergren et al. (2020)). 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, Yang and Zhu (2014)). Different from forward problems, inverse problems try to identify PDE-based models from the observed noisy data. Research on the inverse problem is relatively sparse, and the corresponding statistical property is notably less known. In this paper, we propose a method for solving the inverse problem, which we refer to as a *PDE identification problem*.

With the rise of big data, the PDE identification problem has become indispensable. A good PDE identification approach offers at least the following two benefits. First, we can predict future trends using the identified PDE model, conditional that such a model reflects the underlying processes. Second,

*Corresponding author.

interpretable PDE models enable scientists to validate/reexamine the underlying physical/biological laws governing the process.

We propose a new method for the PDE identification problem, called *Spline-Assisted Partial Differential Equation-based Model Identification (SAPDEMI)*. SAPDEMI can efficiently identify the underlying PDE model from noisy data $\mathcal{D}$:

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

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

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

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

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

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, and 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, $\boldsymbol{\beta}^* = (\beta_{00}^*, \beta_{0^1}^*, \beta_{1^1}^*, \ldots, \beta_{q_{\max}^{p_{\max}}}^*)$, as $\boldsymbol{\beta}^* = (\beta_1^*, \beta_2^*, \beta_3^*, \ldots, \beta_K^*)^\top$, where $K = 1 + (p_{\max}+1)q_{\max} + (1/2)q_{\max}(q_{\max}+1)(p_{\max}-1)!$ is the total number of coefficients on the right-hand side. Noted that, in practice, the majority of the entries in $\boldsymbol{\beta}^*$ are zero. For instance, in the transport equation $\partial u(x,t)/\partial t = a(\partial u(x,t)/\partial x)$, with any $a \neq 0$, we have only $\beta_3^* \neq 0$ and $\beta_i^* = 0$, for any $i \neq 3$ (see Olver (2014, Sec. 2.2)). Therefore, it is reasonable to assume that the coefficient $\boldsymbol{\beta}^*$ in (1.3) is sparse.

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

We design our proposed SAPDEMI method as a two-stage method to identify the underlying PDE models from the noisy data $\mathcal{D}$. The first stage is called the *functional estimation stage*, where we estimate all the derivatives from the noisy data $\mathcal{D}$, including $\partial u(x,t)/\partial t, \partial u(x,t)/\partial x$, and so on. In this stage, we first use cubic splines (Shridhar and Balatoni (1974)) to fit the noisy data $\mathcal{D}$, and then we approximate the derivatives of the underlying function from 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 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, Tibshirani and Wainwright (2015, Chap. 11)).

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

1.1. Literature review

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

The first class of extensions modifies the functional estimation stage of Liang and Wu (2008), and can be divided into three categories. **(F1)**. In the numerical differentiation category (Wu, Xue and Kumar (2012)), the derivative $\partial u(x,t)/\partial x$ is simply approximated as $\partial u(x,t)/\partial x \approx (u(x+\Delta x,t) - u(x-\Delta x,t))/(2\Delta x)$, where $(x+\Delta x,t)$ and $(x-\Delta x,t)$ are the two closest points to (x,t) in the 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 can be highly biased, because its accuracy depends greatly on Δx: a small value of Δx yields large rounding errors in the subtraction (Ueberhuber (2012)), and a large value of Δx leads to poor performance when estimating the tangent slope using secants. Thus, this naive numerical differentiation is not preferred owing to its bias. **(F2)**. In the basis expansion category, researchers first approximate the unknown functions using basis expansion methods, and then approximate the derivatives of the underlying function as those of the approximated functions. There are multiple options for the choice of bases. The most popular basis is the local polynomial basis (see Liang and Wu (2008); Bär, Hegger and Kantz (1999); Schaeffer (2017); Rudy et al. (2017); Parlitz and Merkwirth (2000)). Another popular choice is the spline basis

(see Wu, Xue and Kumar (2012); Xun et al. (2013); Wang, Liu and Zhang (2019)). Our proposed method belongs here. In this category, the major limitation of 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 show that our proposed SAPDEMI method requires only $O(MN)$. 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 $O(MN)$, As achieved by our proposed SAPDEMI method. **(F3)**. In the machine or deep learning category, researchers first fit unknown functions using machine/deep learning methods, and then approximate the derivatives of the underlying functions as those of the approximated functions. A popular machine/deep learning method is the neural network (NN) approach. For instance, Srivastava et al. (2020) use an artificial neural network (ANN). These methods are limited by potential overfitting and the selection of the hyper-parameters.

The second class of extensions modifies the model identification stage of Liang and Wu (2008). Here, existing methods fall within the framework of the (penalized) least squares method, and we can again divide them into three categories. **(M1)**. In the least squares category, researchers study ordinary differential equation (ODE) identification (Miao et al. (2009)) and PDE identification (Bär, Hegger and Kantz (1999); Wu, Xue and Kumar (2012)) , although they too have problems with overfitting. **(M2)**. In the ℓ_2-penalized least squares category, Xun et al. (2013) and Wang, Liu and Zhang (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 within the framework of the ℓ_2-penalized least squares method. Although this method helps to avoid overfitting by introducing the ℓ_2-penalty, it has limited power in terms of "model selection". **(M3)**. In the ℓ_1-penalized least squares method category, Schaeffer (2017) identifies unknown dynamic models (i.e., functions) using the ℓ_1-penalized least squares method. The author provides an efficient algorithm, based on the proximal mapping method, but does not discuss the statistical proprieties of the identified model. Recently, Kang, Liao and Liu (2019) use a similar method to that of Schaeffer (2017), and demonstrate empirical successes. However, the derivation of the statistical theory is still missing. Our study addresses this gap in the literature.

In addition to the ℓ_2- or ℓ_1-penalized least squares methods, other methods have been proposed for the model selection stage, but are not as widely used. Here, examples include the Akaike information criterion (AIC) 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 numerical implementation. The last one is ad-hoc, and may be difficult to analyze. Thus, we do not address these alternative approaches.

Although our proposed SAPDEMI method applies to the PDE model, other nonparametric models are possible. Here, we take PDEs as an initial research project mainly because they are 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 the PDE to machine learning (ML) models (e.g., neural network, random forest), because a PDE offers insight into the physical law. However, the ML models are usually black-box methods (Loyola-Gonzalez (2019)). We also prefer the PDE to the time series models, because it behaves like a "continuous version" of a time series model (Perona, Porporato and Ridolfi (2000); Chen, Ohlssen and Zhou (2018)) at a high level. Furthermore, we prefer the PDE to the Gaussian process (GP) model, because the GP model restricts its response variables to follow a Gaussian distribution (Liu et al. (2020); Wei et al. (2018)). Again, although we take the PDE as our initial research project, we are open to using the aforementioned nonparametric models in future work.

1.2. Our contributions

Here, 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 requires computational complexity of order $\max\{O(M^2N), O(MN^2)\}$, which is higher. **(2)** For our proposed SAPDEMI method, we establish a theoretical guarantee of the model identification accuracy, which, to the best of our knowledge, is a novel result. **(3)** We extend our method to PDE-based model identification, and compare it with ODE-based model identification. The latter has more related work, whereas the former is not yet well understood.

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

2. Proposed Method: SAPDEMI

The proposed SAPDEMI method is a two-stage method for identifying the underlying PDE model from noisy data $\mathcal{D}$. The first stage is called the *functional estimation stage*. Here, we estimate the function and its derivatives from the noisy

data $\mathcal{D}$ in (1.1), and use these as input in the second stage. The second stage is called the *model identification stage*, where we identify the underlying PDE-based model.

In our notation, scalars are denoted by lowercase letters (e.g., β). Vectors are denoted by lowercase bold face letters (e.g., $\boldsymbol{\beta}$), and its ith entry is denoted as β_i. Matrices are denoted by uppercase boldface 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 kth norm is defined as $\|\boldsymbol{\beta}\|_k := \left(\sum_{i=1}^p |\beta_i|^k\right)^{1/k}$. 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 an 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 describe 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 the spatial variable x as an example; 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 $\{(x_i, u_i^n)\}_{i=0,1,\ldots,M-1}$ 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 $i = 1, \ldots, M-1$, $s(x)$ is a polynomial of degree three in $[x_{i-1}, x_i]$; **(3)**. For the two end-points, x_0 and 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 the data $\{(x_i, u_i^n)\}_{i=0,1,\ldots,M-1}$ (with a general fixed $n \in \{0, 1, \ldots, N-1\}$) into the above cubic spline $s(x)$, one can solve $s(x)$ as the minimizer of the following optimization problem:

$$J_\alpha(s) = \alpha \sum_{i=0}^{M-1} w_i[u_i^n - s(x_i)]^2 + (1-\alpha)\int_{x_0}^{x_{M-1}} s''(x)^2 dx, \tag{2.1}$$

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 the residuals, and we take the weight $w_0 = w_1 = \cdots = w_{M-1} = 1$. In the second term, $(1-\alpha)\int_{x_0}^{x_{M-1}} s''(x)^2 dx$, the 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 obtain an 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 $s(x)$, that is, we have $\widehat{u(x,t_n)} \approx \widehat{s(x)}, \widehat{\partial u(x,t_n)/\partial x} \approx \widehat{s'(x)}, \widehat{\partial^2 u(x,t_n)/\partial x^2} \approx \widehat{s''(x)}$ (Ahlberg et al. (1967); Rubin and Graves Jr (1975); Rashidinia and Mohammadi (2008)). Following a similar procedure to obtain the derivatives with respect to the spatial variable x, we can get the derivatives with respect to the temporal variable t, that is, $\widehat{\partial u(x_i,t_n)/\partial t}$, for any $i = 0, \ldots, M-1$ and $n = 0, \ldots, N-1$.

A nice property of the cubic spline is that there is a closed-form solution for (2.1). First, the value of the cubic spline $s(x)$ at the point $\{x_0, x_1, \ldots, x_{M-1}\}$, that is, $\widehat{\mathbf{s}} = (\widehat{s(x_0)}, \widehat{s(x_1)}, \ldots, \widehat{s(x_{M-1})})^\top$, can be solved as

$$\widehat{\mathbf{s}} = [\alpha\mathbf{W} + (1-\alpha)\mathbf{A}^\top\mathbf{M}\mathbf{A}]^{-1}\alpha\mathbf{W}\mathbf{u}_{:}^{n}. \tag{2.2}$$

The above closed-form estimation can be used to approximate the function that corresponds to the underlying PDE model, that is, $\widehat{\mathbf{s}} \approx \widehat{\mathbf{f}} = (\widehat{u(x_0,t_n)}, \widehat{u(x_1,t_n)}, \ldots, \widehat{u(x_{M-1},t_n)})^\top$. Here, $\mathbf{W} = \mathrm{diag}(w_0, \ldots, w_{M-1}) \in \mathbb{R}^{M\times M}$, vector $\mathbf{u}_{:}^{n} = (u_0^n, \ldots, u_{M-1}^n)^\top \in \mathbb{R}^M$, and the matrices $\mathbf{A} \in \mathbb{R}^{(M-2)\times M}$ and $\mathbf{M} \in \mathbb{R}^{(M-2)\times(M-2)}$ are

$$\mathbf{A} = \begin{pmatrix} \frac{1}{h_0} & \frac{-1}{h_0} - \frac{1}{h_1} & \frac{1}{h_1} & \ldots & 0 & 0 & 0 \\ 0 & \frac{1}{h_1} & \frac{-1}{h_1} - \frac{1}{h_2} & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & \frac{1}{h_{M-3}} & \frac{-1}{h_{M-3}} - \frac{1}{h_{M-2}} & \frac{1}{h_{M-2}} \end{pmatrix}, \tag{2.3}$$

$$\mathbf{M} = \begin{pmatrix} \frac{h_0+h_1}{3} & \frac{h_1}{6} & 0 & \ldots & 0 & 0 \\ \frac{h_1}{6} & \frac{h_1+h_2}{3} & \frac{h_2}{6} & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & \frac{h_{M-3}}{6} & \frac{h_{M-3}+h_{M-2}}{3} \end{pmatrix}, \tag{2.4}$$

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

For the mathematical derivation of (2.2) from (2.1), and the derivation of first- and second-order derivatives, please refer to the Supplementary Material S2.

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

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

Proposition 1. *Given the data $\mathcal{D}$ in* (1.1)*, if we use the cubic spline in* (2.1) *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 S10.1.

As suggested by Proposition 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 sample size of $\mathcal{D}$, and reading the data is a task of order $O(MN)$. Therefore, it is very efficient to use a cubic spline, because its computational complexity achieves the lowest possible order of complexity.

By way of comparisons, we discuss the computational complexity of the local polynomial regression, which is widely used in the literature (Liang and Wu (2008); Bär, Hegger and Kantz (1999); Schaeffer (2017); Rudy et al. (2017); Parlitz and Merkwirth (2000)). This 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 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 that of the cubic spline method in Proposition 1. The related proposition and proof are available in Supplementary Materials S10. We summarize the pros and cons of the cubic spline and the local polynomial regression in Table 1.

2.2. Model identification stage

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

Note that the model in (1.3) can be regarded as a linear regression model with a response variable that is the first-order derivative with respect to the temporal variable t, that is, $\partial u(x,t)/\partial t$, and the covariates are the derivative(s) with

respect to the spatial variable x, including $\partial u(x_i,t_n)/\partial x, \partial^2 u(x_i,t_n)/\partial x^2, \ldots,$ $(\partial^2 u(x_i,t_n)/\partial x^2)^{p_{\max}}$. Because we have MN observations in the data set $\mathcal{D}$ in (1.1), the response vector is of length MN:

$$\nabla_t \mathbf{u} = \left(\frac{\partial \widehat{u(x_0,t_0)}}{\partial t}, \ldots, \frac{\partial \widehat{u(x_{M-1},t_0)}}{\partial t}, \ldots, \frac{\partial \widehat{u(x_{M-1},t_{N-1})}}{\partial t} \right)^\top, \tag{2.5}$$

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

$$\mathbf{X} = (\, \widehat{\mathbf{x}_0^0}, \widehat{\mathbf{x}_1^0}, \ldots, \widehat{\mathbf{x}_{M-1}^0}, \widehat{\mathbf{x}_1^0}, \ldots, \widehat{\mathbf{x}_{M-1}^{N-1}} \,)^\top \in \mathbb{R}^{MN \times K}. \tag{2.6}$$

For the above design matrix $\mathbf{X}$, its $(nN+i+1)$st row is $\widehat{\mathbf{x}_i^n} = (1, \widehat{u(x_i,t_n)}, \partial \widehat{u(x_i,t_n)}/\partial x, \partial^2 \widehat{u(x_i,t_n)}/\partial x^2, (\widehat{u(x_i,t_n)})^2, \ldots, (\partial^2 \widehat{u(x_i,t_n)}/\partial x^2)^{p_{\max}})^\top$. The K components of $\widehat{\mathbf{x}}_i^n$ are candidate terms in the PDE model. Note that all of the derivatives listed in (2.5) and (2.6) are estimated from the functional estimation stage described in Section 2.1.

Next, we use the Lasso to identify the nonzero coefficients in (1.3):

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{2MN} \|\nabla_t \mathbf{u} - \mathbf{X}\boldsymbol{\beta}\|_2^2, + \lambda \|\boldsymbol{\beta}\|_1, \tag{2.7}$$

where $\lambda > 0$ is a turning parameter that controls the trade-off between the sparsity of $\boldsymbol{\beta}$ and the goodness of fit. Given the ℓ_1 penalty in (2.7), $\widehat{\boldsymbol{\beta}}$ is sparse, that is, only a few of its entries are likely to be nonzero. Accordingly, we can identify the underlying PDE model as

$$\frac{\partial}{\partial t} u(x,t) = \mathbf{x}^\top \widehat{\boldsymbol{\beta}}, \tag{2.8}$$

where $\mathbf{x} = (1, u(x,t), \partial u(x,t)/\partial x, \partial^2 u(x,t)/\partial x^2, (u(x,t))^2, \ldots, (\partial^2 u(x,t)/\partial x^2)^{p_{\max}})^\top \in \mathbb{R}^K$. To solve equation (2.7), one can use the coordinate descent method (Beck and Tetruashvili (2013); Tseng (2001)); see the online Supplementary Material S4.

3. Theory on Statistical Properties

The theoretical evaluation is performed from two aspects. **(S1).** First, we check whether our identified PDE model contains derivatives that are included in the "true" underlying PDE model. This is called the *support set recovery* property. Mathematically, we check whether $\operatorname{supp}(\widehat{\boldsymbol{\beta}}) \subseteq \operatorname{supp}(\boldsymbol{\beta}^*)$, where $\widehat{\boldsymbol{\beta}}$ is the minimizer of (2.7), $\boldsymbol{\beta}^*$ is the ground truth, and $\operatorname{supp}(\boldsymbol{\beta}) = \{i : \beta_i \neq 0,\ \forall\, i, 1 \leq i \leq K\}$, for a general vector $\boldsymbol{\beta} \in \mathbb{R}^K$. However, the support recovery depends on the choice of the penalty parameter λ: a large value of λ leads to $\operatorname{supp}(\widehat{\boldsymbol{\beta}}) = \emptyset$ (empty set), whereas a small value of λ results in a nonsparse $\widehat{\boldsymbol{\beta}}$. A proper

selection of λ hopefully leads to the correct recovery of the support set recovery, that is, we have $\text{supp}(\widehat{\boldsymbol{\beta}}) \subseteq \text{supp}(\boldsymbol{\beta}^*)$. We discuss the selection of λ to achieve the above goal in Theorem 1. **(S2).** Second, we are interested in an upper bound of the estimation error of our estimator. Specifically, we consider $\|\widehat{\boldsymbol{\beta}}_{\mathcal{S}} - \boldsymbol{\beta}^*_{\mathcal{S}}\|_\infty$, where $\mathcal{S} = \text{supp}(\boldsymbol{\beta}^*)$, and the vectors $\widehat{\boldsymbol{\beta}}_{\mathcal{S}}$ and $\boldsymbol{\beta}^*_{\mathcal{S}}$ are subvectors of $\widehat{\boldsymbol{\beta}}$ and $\boldsymbol{\beta}^*$, respectively, and contain only elements with indices that are in $\mathcal{S}$. An upper bound of the above estimation error is discussed in Theorem 2.

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

3.1. Conditions for the theorems

In this section, we introduce the conditions we use for our theorems. We begin with three frequently used conditions in ℓ_1-regularized regression models. These conditions provide sufficient conditions for exact sparse recovery (see Hastie, Tibshirani and Wainwright (2015, Chap. 11)). Subsequently, we introduce three conditions that are widely used in cubic spline-based functional estimation (see Silverman (1984, (2.5)-(2.8))).

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

Condition 2 (Minimal Eigenvalue Condition). *There exists some constant $C_{\min} > 0$ such that $\Lambda_{\min}\left((1/NM)\mathbf{X}_{\mathcal{S}}^\top \mathbf{X}_{\mathcal{S}}\right) \geq C_{\min}$, 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 a stronger version of the invertibility condition (see Hastie, Tibshirani and Wainwright (*2015*, Chap.* 11*)).*

Condition 3 (Knots c.d.f. Convergence Condition). *Suppose the sequence of the empirical distribution function over the design points $a = x_0 < \cdots < x_{M-1} = b$, with different sample size M, is denoted as $F_M(x)$, that is, we have $F_M(x) = (1/M)\sum_{i=0}^{M-1} \mathbb{1}\{x_i \leq x\}$. Then, there exists an absolutely continuous distribution function F on $[a, b]$ such that $F_M \to F$ uniformly as $M \to +\infty$. Here, $\mathbb{1}\{A\}$ is the indicator of event A. A similar condition holds for the temporal variable: suppose the sequence of the empirical distribution function over the design points $\bar{a} = t_0 < \cdots < t_{N-1} = \bar{b}$, with different sample size N, is denoted as $G_N(x)$. Then, there exists an absolutely continuous distribution function G on $[\bar{a}, \bar{b}]$ such that $G_N \to G$ uniformly as $N \to +\infty$.*

Condition 4 (Knots p.d.f. Convergence Condition). *Suppose the first derivatives of the functions F and G (defined in Condition* 3*) are denoted as f and g, respectively. Then we have*

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

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

Condition 5 (Gentle Decrease of Smoothing Parameter Condition). *Suppose that $\zeta(M) = \sup_{[x_0, x_{M-1}]} |F_M - F|$. The smoothing parameter α in* (2.1) *depends on M in such a way that $\alpha \to 0$ and $\alpha^{-1/4}\zeta(M) \to 0$ as $M \to +\infty$. A similar condition also holds for the temporal variable.*

3.2. Main theory

In the first theorem, we develop the lower bound of λ to realize the correct recovery of the support set, that is, $\mathcal{S}(\widehat{\boldsymbol{\beta}}) \subseteq \mathcal{S}(\boldsymbol{\beta}^*)$.

Theorem 1. *Given the data in* (1.1)*, suppose the conditions in Lemma* S6.1 *and Corollary* S6.1 *(see the online Supplementary Material* S6*) hold, as do Condition* 1 - 5. *If we take $M = O(N)$, then there exists a constant $\mathscr{C}_{(\sigma, \|u\|_{L^\infty(\Omega))}} > 0$ that is independent of the spatial resolution M and the temporal resolution N. Thus, if we set the cubic spline smoothing parameter with the spatial variable x in* (2.1) *as $\alpha = O((1 + M^{-4/7})^{-1})$, set the cubic spline smoothing parameter with temporal variable t as $\bar{\alpha} = O((1 + N^{-4/7})^{-1})$, and set the turning parameter*

$$\lambda \geq \mathscr{C}(\sigma, \|u\|_{L^\infty(\Omega)}) \frac{\sqrt{K}\log(N)}{\mu N^{3/7 - r}} \tag{3.1}$$

to identify the PDE model in (2.7)*, for some $r \in (0, 3/7)$, with sufficient large N, then with probability greater than $P_\mu - \underbrace{O\left(Ne^{-N^r}\right)}_{P'}$, we have $\mathcal{S}(\widehat{\boldsymbol{\beta}}) \subseteq \mathcal{S}(\boldsymbol{\beta}^*)$. Here, K is the number of columns of the design matrix $\mathbf{X}$ in* (2.7)*, and μ and P_μ are defined in Condition* 1.

The proof of the above theorem can be found in the Supplementary Material S10, along with several lemmas, the conditions of which are standardized in cubic splines. The above theorem provides the lower bound of λ to realize the correct recovery of the support set. As indicated by (3.1), the lower bound is affected by several factors. First, it is affected by the temporal resolution N: as N increases, there is greater flexibility in tuning the penalty parameter λ. Second, the lower bound in (3.1) is affected by the incoherence parameter μ: if μ is small, then the lower bound increases. This is because a small μ means that the feature variable candidates are similar to each other. This phenomenon is called *multicollinearity*. In this case, we have a very limited choice in terms of selecting λ. However, we cannot increase the value of μ, because this is decided by the data set $\mathcal{D}$ (see Condition 1). Third, the lower bound in (3.1) is affected by the number of columns of the matrix $\mathbf{X}$. If its number of columns is very large, then it requires a larger λ to identify the significant feature variables from

among potential feature variables.

Note too that the probability $P_\mu - P'$ converges to P_μ as $N \to +\infty$. This limiting probability P_μ is determined by the data $\mathcal{D}$ (see Condition 1). Thus, when N is very large, our proposed SAPDEMI method can realize $\mathcal{S}(\widehat{\boldsymbol{\beta}}) \subseteq \mathcal{S}(\boldsymbol{\beta}^*)$ with probability close to P_μ.

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

Theorem 2. *Suppose the conditions in Theorem* 1 *hold. Then with probability greater than* $1 - O(Ne^{-N^r}) \to 1$, *there exists an* $\dot{N} > 0$, *such that when* $N > \dot{N}$, *we have*

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

where K *is the number of columns of the matrix* $\mathbf{X}$, $\mathcal{S} := \{i : \beta_i^* \neq 0,\ \forall i = 1, \ldots, K\}$ *and the vectors* $\widehat{\boldsymbol{\beta}}_{\mathcal{S}}$, *and* $\boldsymbol{\beta}^*_{\mathcal{S}}$ *are subvectors of* $\widehat{\boldsymbol{\beta}}$ *and* $\boldsymbol{\beta}^*$, *respectively, that contain only those elements with indices that are in* $\mathcal{S}$. *The theorem shows that when* $N \to +\infty$, *the error bound convergences to* 0.

The proof can be found in the Supplementary Material S10. The previous theorem shows that the estimation error bound for the ℓ_∞-norm of the coefficient error in (3.2) consists of two components. The first component is affected by the temporal resolution N and the number of feature variable candidates K. As $N \to +\infty$, this first component converges to zero without an explicit dependence on the feature variable selected from (2.7). The second component is $\sqrt{K} C_{\min} \lambda$. When N increases to $+\infty$, this second component also converges to zero. This is because, as stated in Theorem 1, when $N \to +\infty$, the lower bound of λ, which realizes the correct support recovery, converges to zero. Thus, the accuracy of the coefficient estimation improves as we increase N.

By combining Theorems 1 and 2, we find that when the minimum absolute value of the nonzero entries of $\boldsymbol{\beta}^*$ is sufficiently large, with an adequate choice of λ, we can guarantee the exact recovery. Mathematically, when $\min_{i \in \mathcal{S}} |(\boldsymbol{\beta}^*_{\mathcal{S}})_i| > \sqrt{K} C_{\min} (\sqrt{K} \mathscr{C}_{(\sigma, \|u\|_{L^\infty(\Omega)})} (\log(N)/N^{3/7-r}) + \lambda)$, where $(\boldsymbol{\beta}^*_{\mathcal{S}})_i$ refers to the ith element in the vector $\boldsymbol{\beta}^*_{\mathcal{S}}$, we have a correct signed support of $\widehat{\boldsymbol{\beta}}$. This helps when selecting the penalty parameters λ. In addition, the plot of the solution paths helps with the selection of the penalty parameters λ; see Section 4.

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 they all play fundamental roles in modeling physical phenomena and demonstrate characteristic behaviors of a more

complex system, such as dissipation and shock formation (Haberman (1983)). In addition to wide applications, they cover a wide range of categories, including the first-order PDE, second-order PDE, linear PDE, and nonlinear PDE, which cover most of the PDEs frequently seen in practice. Furthermore, the difficultly of identifying the above PDE models increases from the first example—the transport equation—to the last example—the viscous Burgers' equation. We set $p_{\max} = 2$ and $q_{\max} = 2$ in (1.3) for the three numerical examples (see the full formula of the full model in the Supplementary Material S11), that is, we identify the PDE model from the full model.

In terms of computational efficiency, the results of these three examples are the same, so we present only the result for the first example. We also verify Conditions 1 - 5 for the above three examples. The details of the verification are provided in the Supplementary Material S12.

4.1. Example 1: transport equation

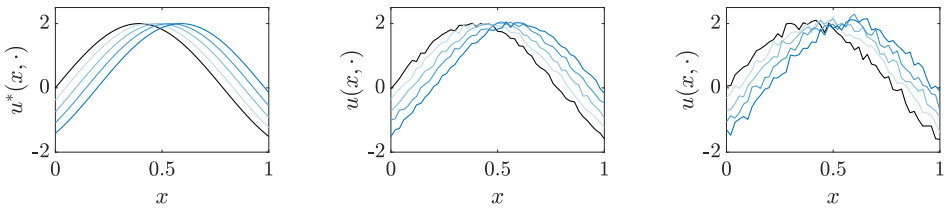
The PDE problem studied in this section is the transport equation. It is a linear first-order PDE model. Given its simplicity and straightforward physical meaning, it is widely used to model the concentration of a substance flowing in a fluid at a constant rate, For example, it can model a pollutant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

$$\begin{cases} \dfrac{\partial}{\partial t}u(x,t) = a\dfrac{\partial}{\partial x}u(x,t), \quad \forall\, 0 \le x \le X_{\max},\ 0 \le t \le T_{\max}; \\ u(x,0) = f(x). \end{cases} \tag{4.1}$$

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

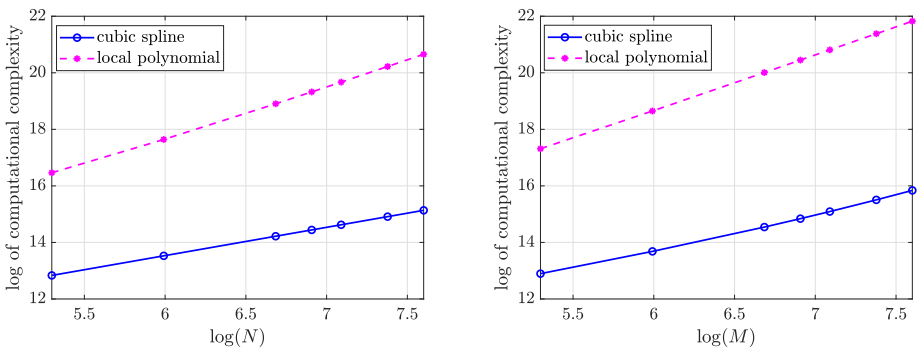
The dynamic pattern of the above transport equation is visualized in Fig. 1, where the subfigures (a), (b), and (c) show the ground truth and noisy observations under $\sigma = 0.05$ and $\sigma = 0.1$, respectively. The figure shows that a larger noise results in the shape of the transport equation being less smooth, potentially leading to additional difficulties in the PDE model identification.

First, we consider 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 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; and (2) N is fixed as 20, and M varies from 200 to 2000. Fig. 2 shows that, as M or N increases, so does the number of numerical operations in the functional estimation stage. We find that the cubic splines method needs fewer numerical operations, compared with the local



(a) truth (b) $\sigma = 0.05$ (c) $\sigma = 0.1$

Figure 1. Noisy/True curves from (4.1) ($M = N = 100$).

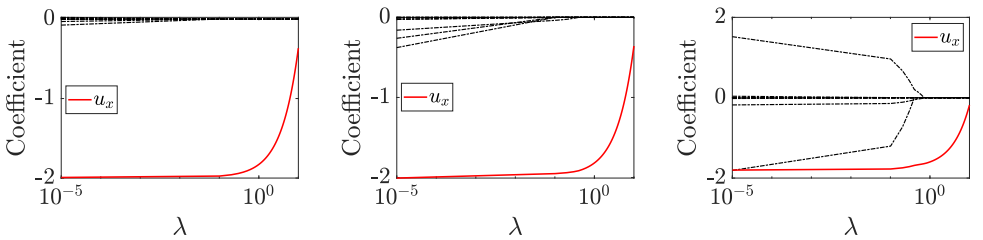


(a) fixed $M = 20$ (b) fixed $N = 20$

Figure 2. Computational complexity of the cubic spline and a local polynomial.

polynomial regression. Furthermore, a simple linear regression of the four lines in Fig. 2 shows that in (a), the slope of the cubic spline is 0.9998, and as N goes to infinity, the slope gets closer to one. This validates that the computational complexity of the cubic splines-based method is of order $O(N)$ when M is fixed. The result in (b) is similar. Thus, we numerically verify that the computational complexity of the cubic spline method is of order $O(MN)$. Similarly, for a local polynomial, we can numerically validate its computational complexity, which is $\max\{O(M^2N), O(MN^2)\}$.

We now verify numerically that with high probability, our SAPDEMI can correctly identify the underlying PDE models. From the formula of the transport equation in equation (4.1), we know that the correct feature variable is $\partial u(x,t)/\partial x$, and that other feature variables should not be identified. We discuss the identification accuracy under different sample sizes and magnitudes of noise. We find that the accuracy stays at 100%. To explain the high accuracy, we plot the solution paths in Fig. 3 under different σ, namely, $\sigma = 0.01, 0.1, 1$. From Fig. 3, we can increase λ to overcome this difficulty, and thus achieve a correct PDE identification.



(a) $\sigma = 0.01$ (b) $\sigma = 0.1$ (c) $\sigma = 1$

Figure 3. Solution paths in the transport equation under different σ and $M = N = 100$. The otation u_x is a simplification of $\partial u(x,t)/\partial x$.

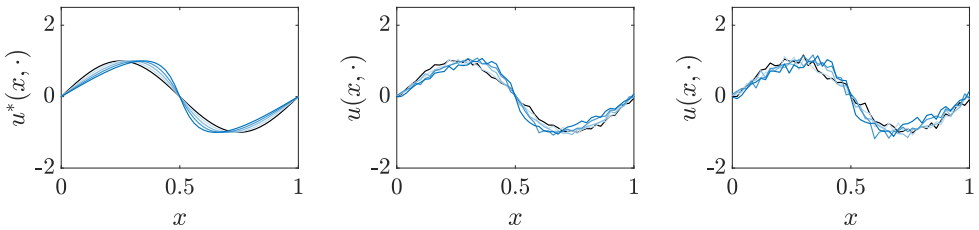
4.2. Example 2: inviscid Burgers' equation

In this section, we investigate the inviscid Burgers' equation (see Olver (2014, Sec. 8.4)), which is representative of a first-order nonlinear PDE and is used frequently 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 Burgers' equation is listed below:

$$\begin{cases} \dfrac{\partial}{\partial t}u(x,t) = -\dfrac{1}{2}u(x,t)\dfrac{\partial}{\partial x}u(x,t) & \\ u(x,0) = f(x) & 0 \le x \le X_{\max} \\ u(0,t) = u(1,t) = 0 & 0 \le t \le T_{\max} \end{cases}, \tag{4.2}$$

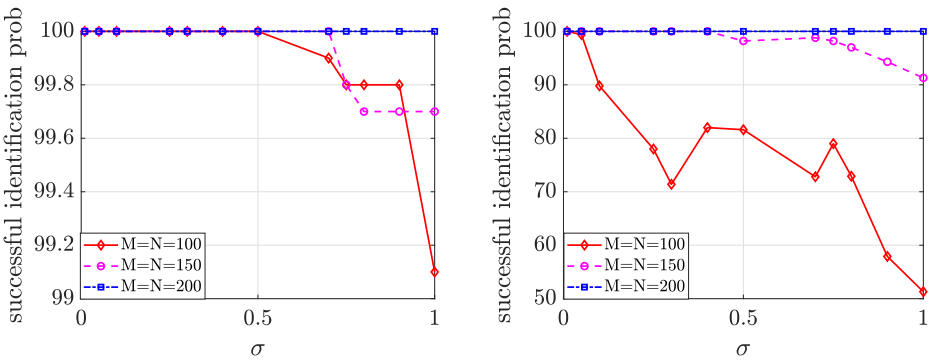
where we set $f(x) = \sin(2\pi x), X_{\max} = 1$ and $T_{\max} = 0.1$. Fig. 4(a), (b), and (c) show the ground truth and noisy observations under $\sigma = 0.05$ and 0.1, respectively. Compared with our first example (transport equation in (4.1)), the inviscid Burgers' equation can be regarded an extension from the linear transport equation to a nonlinear transport equation. Specifically, if we set a in (4.1) as $a = -(1/2)u(x,t)$, then (4.1) is equivalent to (4.2). In the literature, this PDE model is considerably more challenging than the linear transport PDE in (4.1): the wave speed in (4.1) depends only on the spatial variable x, whereas the wave speed in (4.2) depends on both the spatial variable x and the size of the disturbance $u(x,t)$. Given the complicated wave speed in (4.2), it can model more complicated dynamic patterns. For example, larger waves move faster, and overtake smaller, slow-moving waves.

In this example, SAPDEMI correctly identifies with an accuracy above 99% (see Fig. 5(a)). The effect of σ is also reflected in Fig. 6, where the length of the λ-interval for correct identification decreases as σ increases.



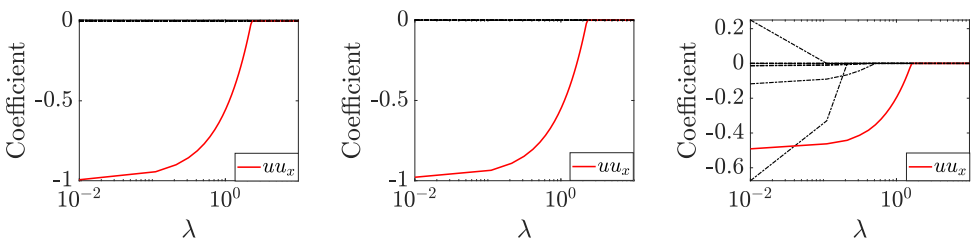
(a) truth (b) $\sigma = 0.05$ (c) $\sigma = 0.1$

Figure 4. Noisy/True curves from (4.2) ($M = 50, N = 50$).



(a) example 2 (b) example 3

Figure 5. Curves of successful identification probability.

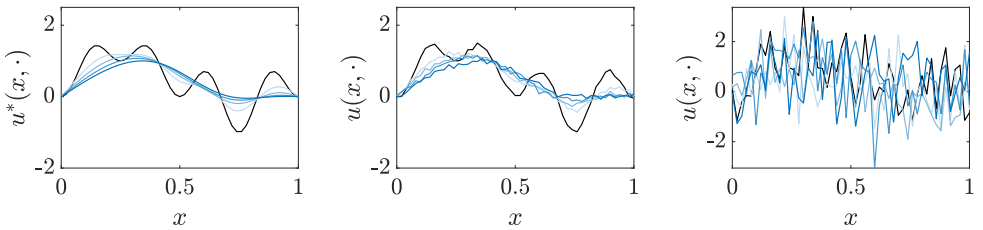


(a) $\sigma = 0.01$ (b) $\sigma = 0.5$ (c) $\sigma = 1$

Figure 6. Solution paths in the inviscid Burgers' equation under different σ and $M = N = 100$. Here u and u_x are simplifications of $u(x,t)$ and $\partial u(x,t)/\partial x$, respectively.

4.3. Example 3: viscous Burgers' equation

In this section, we investigate the more challenging viscous Burgers' equation (see Olver (2014, Sec. 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 acoustics in dissipative media (Rudenko and Soluian (1975)). For example, in fluid and gas dynamics, we can interpret the term $\nu(\partial^2 u(x,t)/\partial x^2)$ as modeling the effect of viscosity (Olver (2014, Sec. 8.4)). Thus,



(a) true (b) $\sigma = 0.05$ (c) $\sigma = 1$

Figure 7. Noisy/True curves from (4.3) ($M = 50, N = 50$).

the viscous Burgers' equation represents a version of the equations of the viscous fluid flows, including the celebrated and widely applied Navier-Stokes equations (Whitham (2011)):

$$\begin{cases} \dfrac{\partial u(x,t)}{\partial t} = -\dfrac{1}{2}u(x,t)\dfrac{\partial}{\partial x}u(x,t) + \nu\dfrac{\partial^2}{\partial x^2}u(x,t) & \\ u(x,0) = f(x) & 0 \le x \le X_{\max}\,, \\ u(0,t) = u(1,t) = 0 & 0 \le t \le T_{\max} \end{cases} \tag{4.3}$$

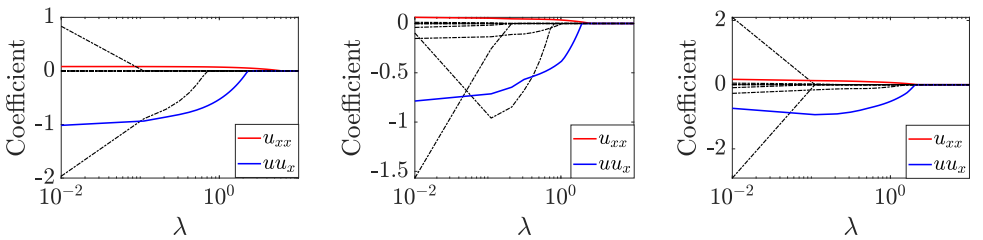
where we set $f(x) = \sin^2(4\pi x) + \sin^3(2\pi x), X_{\max} = 1, T_{\max} = 0.1$ and $\nu = 0.1$. Fig. 7 shows the corresponding curves, where (a), (b), and (c) are the ground truth and noisy observations under $\sigma = 0.05$ and $\sigma = 0.1$, respectively.

Compared with the previous two PDE models (transport equation in (4.1) and inviscid Burgers' equation in (4.2)), 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. Our simulations provide sufficiently complicated examples.

Based on Fig. 5(b), we conclude that with high probability, our proposed SAPDEMI can correctly identify the underlying viscous Burgers' equation, for the following reasons. When $M = N = 200$ or 150, the accuracy stays above 90% for all levels of $\sigma \in [0.01, 1]$. When $M = N = 100$, the accuracy is above 70% when $\sigma \in [0.01, 0.5]$, and reduces to 50% when $\sigma = 1$. This makes sense, because as shown in Fig. 8, when σ increases from 0.01 to 1, the length of the λ-interval for correct identification decreases, making it more difficult to realize a correct identification. Thus, if we encounter a very noisy data set $\mathcal{D}$, a larger sample size is preferred.

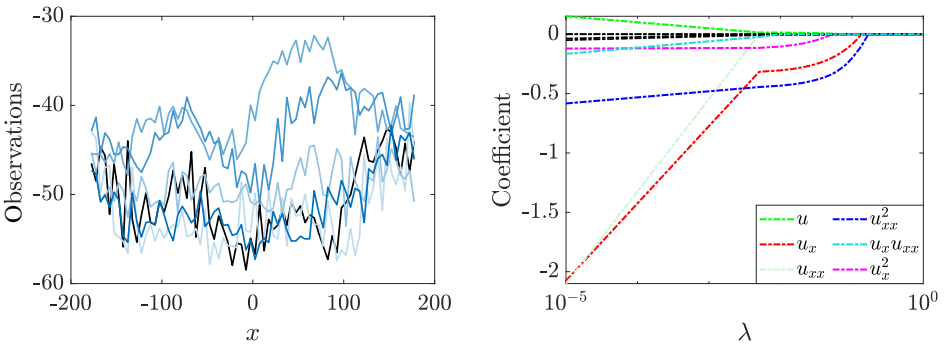
5. Case Study

In this section, we apply SAPDEMI to a real-world data set that is a subset of the Cloud-Aerosol Lidar and Infrared Pathfinder Satellite Observations (CALIPSO) data set downloaded from NASA. The CALIPSO reports the



(a) $\sigma = 0.01$ (b) $\sigma = 0.5$ (c) $\sigma = 1$

Figure 8. Solution paths in the viscous Burgers' equation under different σ and $M = N = 100$. The notation u_{xx} and uu_x stand for $u(x,t)(\partial u(x,t)/\partial x)$ and $\partial^2 u(x,t)/\partial x^2$, respectively.



(a) observed temperature (b) solution path

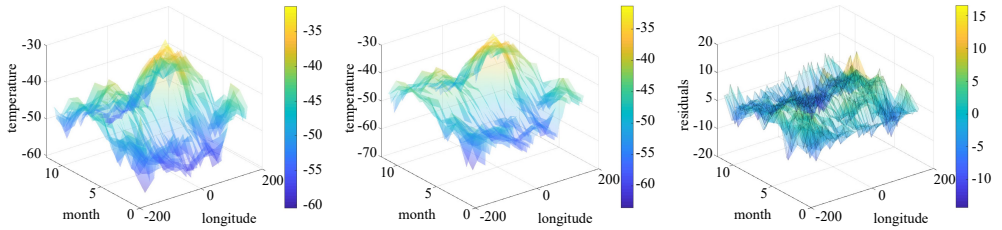
Figure 9. Visualization and identification of the CALIPSO data.

monthly mean of temperature in 2017 at 34°N and 110.9418 meters above the Earth's surface over a uniform spatial grid from 180°W to 180°E, with equally spaced 5° intervals. The missing data are handled either by direct imputation or by using the instrument methods (Chen, Xie and Shao (2018); Chen, Shao and Fang (2021); Chen and Fang (2019); Chen, Fang and Xiao (2018)).

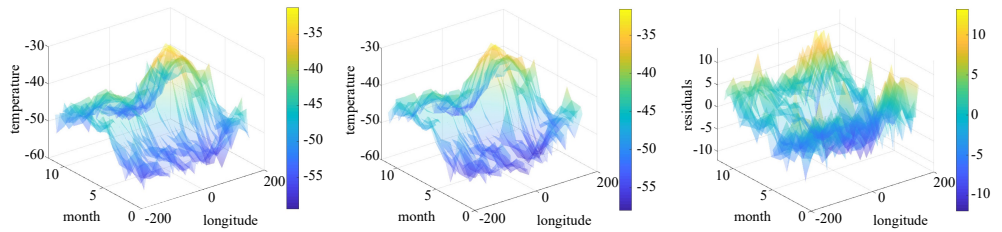
The identified PDE model ($N = 12, M = 72$), reasonably speaking, is

$$\frac{\partial}{\partial t}u(x,t) = a\frac{\partial}{\partial x}u(x,t) + b\left(\frac{\partial^2}{\partial x^2}u(x,t)\right)^2, \tag{5.1}$$

where the values of a and b can be estimated using a simple linear regression on the selected derivatives, that is, $\partial u(x,t)/\partial x$ and $\left(\partial^2 u(x,t)/\partial x^2\right)^2$. The linear regression suggests reasonable values of $a = -0.2505$ and $b = 1.7648$. Note that we focus on identification, that is, identifying $\partial u(x,t)/\partial x$ and $\left(\partial^2 u(x,t)/\partial x^2\right)^2$ from many derivative candidates, rather than estimating the coefficients. Therefore, we use $a = -0.2505$ and $b = 1.7648$ as a reference.



(a.1) observed 2017 temp (a.2) fitted 2017 temp (a.3) 2017 residual



(b.1) observed 2018 temp (b.2) predicted 2018 temp (b.3) 2018 residual

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

Because the CALIPSP is a real-world data set, we do not know the ground truth of the underlying PDE model. Here we provide some justifications. First, from the solution path in Fig. 9(b), the coefficients of $\partial u(x,t)/\partial x$ and $(\partial^2 u(x,t/\partial x^2))^2$ remain nonzeros under $\lambda = 0.05$, whereas the other coefficients are all zero. Second, the identified PDE model in (5.1) fits well to the training data (see Fig. 10 (a.1)-(a.3)). Third, the identified PDE model in (5.1) predicts well in the testing data (see Fig. 10 (b.1)-(b.3)). Thus, our proposed SAPDEMI method performs well in the CALIPSO data set, beacuse it adequately predicts the feature values in 2018.

6. Conclusion

We have proposed an SAPDEMI method for identifying underlying PDE models from noisy data. The proposed method is computationally efficient, and we derive a statistical guarantee on its performance. We realize there are many promising future research directions, including, but not limited to, incorporating a multivariate spatial variable ($\mathbf{x} \in \mathbb{R}^d$ with $d \geq 2$) (Habermann and Kindermann (2007)), and the interactions between spatial and temporal variables. In our paper, we aim at showing the methodology to solve the PDE identification, so we skip discussing the above future research and our paper should provide a good starting point for these further research.

Supplementary Material

There is an online Supplementary Material for this paper, which includes (1) lemmas to derive the main theory; (2) numerical details of the figures in the simulation; (3) proofs and other technical details which is not covered in the main body of the paper due to the page limitation.

Acknowledgments

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

References

Ahlberg, J., Walsh, J., Bellman, R. and Nilson, E. N. (1967). *The Theory of Splines and Their Applications*. Academic Press.

Bär, M., Hegger, R. and Kantz, H. (1999). Fitting partial differential equations to space-time dynamics. *Physical Review E* **59**, 337.

Beck, A. and Tetruashvili, L. (2013). On the convergence of block coordinate descent type methods. *SIAM Journal on Optimization* **23**, 2037–2060.

Bonkile, M. P., Awasthi, A., Lakshmi, C., Mukundan, V. and Aswin, V. (2018). A systematic literature review of burgers' equation with recent advances. *Pramana* **90**, 1–21.

Chen, J. and Fang, F. (2019). Semiparametric likelihood for estimating equations with non-ignorable non-response by non-response instrument. *Journal of Nonparametric Statistics* **31**, 420–434.

Chen, J., Fang, F. and Xiao, Z. (2018). Semiparametric inference for estimating equations with nonignorably missing covariates. *Journal of Nonparametric Statistics* **30**, 796–812.

Chen, J., Ohlssen, D. and Zhou, Y. (2018). Functional mixed effects model for the analysis of dose-titration studies. *Statistics in Biopharmaceutical Research* **10**, 176–184.

Chen, J., Shao, J. and Fang, F. (2021). Instrument search in pseudo-likelihood approach for nonignorable nonresponse. *Annals of the Institute of Statistical Mathematics* **73**, 519–533.

Chen, J., Xie, B. and Shao, J. (2018). Pseudo likelihood and dimension reduction for data with nonignorable nonresponse. *Statistical Theory and Related Fields* **2**, 196–205.

Haberman, R. (1983). *Elementary Applied Partial Differential Equations*. Prentice Hall, Englewood Cliffs.

Habermann, C. and Kindermann, F. (2007). Multidimensional spline interpolation: Theory and applications. *Computational Economics* **30**, 153–169.

Hastie, T., Tibshirani, R. and Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press.

Kang, S. H., Liao, W. and Liu, Y. (2019). IDENT: Identifying differential equations with numerical time evolution. *arXiv:1904.03538*.

Lagergren, J. H., Nardini, J. T., Michael Lavigne, G., Rutter, E. M. and Flores, K. B. (2020). Learning partial differential equations for biological transport models from noisy spatio-temporal data. *Proceedings of the Royal Society A* **476**, 20190800.

Liang, H. and Wu, H. (2008). Parameter estimation for differential equation models using a framework of measurement error in regression models. *Journal of the American Statistical Association* **103**, 1570–1583.

Liu, H., Ong, Y.-S., Shen, X. and Cai, J. (2020). When Gaussian process meets big data: A review of scalable GPs. *IEEE Transactions on Neural Networks and Learning Systems* **31**, 4405–4423.

Loyola-Gonzalez, O. (2019). Black-box vs. white-box: Understanding their advantages and weaknesses from a practical point of view. *IEEE Access* **7**, 154096–154113.

Lu, T., Liang, H., Li, H. and Wu, H. (2011). High-dimensional ODEs coupled with mixed-effects modeling techniques for dynamic gene regulatory network identification. *Journal of the American Statistical Association* **106**, 1242–1258.

Mangan, N. M., Kutz, J. N., Brunton, S. L. and Proctor, J. L. (2017). Model selection for dynamical systems via sparse regression and information criteria. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **473**, 20170009.

McKinley, S. and Levine, M. (1998). Cubic spline interpolation. *College of the Redwoods* **45**, 1049–1060.

Miao, H., Dykes, C., Demeter, L. M. and Wu, H. (2009). Differential equation modeling of HIV viral fitness experiments: Model identification, model selection, and multimodel inference. *Biometrics* **65**, 292–300.

Olver, P. J. (2014). *Introduction to Partial Differential Equations*. Springer.

Parlitz, U. and Merkwirth, C. (2000). Prediction of spatiotemporal time series based on reconstructed local states. *Physical Review Letters* **84**, 1890.

Perona, P., Porporato, A. and Ridolfi, L. (2000). On the trajectory method for the reconstruction of differential equations from time series. *Nonlinear Dynamics* **23**, 13–33.

Rashidinia, J. and Mohammadi, R. (2008). Non-polynomial cubic spline methods for the solution of parabolic equations. *International Journal of Computer Mathematics* **85**, 843–850.

Rubin, S. G. and Graves Jr, R. A. (1975). A cubic spline approximation for problems in fluid mechanics. *NASA STI/Recon Technical Report N* **75**, 33345.

Rudenko, O. and Soluian, S. (1975). *The Theoretical Principles of Nonlinear Acoustics*. Moscow Izdatel Nauka.

Rudy, S. H., Brunton, S. L., Proctor, J. L. and Kutz, J. N. (2017). Data-driven discovery of partial differential equations. *Science Advances* **3**, 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**, 20160446.

Shridhar, M. and Balatoni, N. (1974). A generalized cubic spline technique for identification of multivariable systems. *Journal of Mathematical Analysis and Applications* **47**, 78–90.

Silverman, B. W. (1984). Spline smoothing: The equivalent variable kernel method. *The Annals of Statistics* **12**, 898–916.

Srivastava, K., Ahlawat, M., Singh, J. and Kumar, V. (2020). Learning partial differential equations from noisy data using neural networks. In *Journal of Physics: Conference Series*, 012075. IOP Publishing.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* **58**, 267–288.

Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications* **109**, 475–494.

Ueberhuber, C. W. (2012). *Numerical Computation 1: Methods, Software, and Analysis*. Springer Science & Business Media.

Wang, D., Liu, K. and Zhang, X. (2019). Spatiotemporal thermal field modeling using partial differential equations with time-varying parameters. *IEEE Transactions on Automation Science and Engineering* **17**, 646–657.

Wang, H., Yang, D. and Zhu, S. (2014). Inhomogeneous Dirichlet boundary-value problems of space-fractional diffusion equations and their finite element approximations. *SIAM Journal on Numerical Analysis* **52**, 1292–1310.

Wei, S.-b., Wang, W., Liu, N., Chen, J., Guo, X.-Y., Tang, R.-B. et al. (2018). U-shaped association between serum free triiodothyronine and recurrence of atrial fibrillation after catheter ablation. *Journal of Interventional Cardiac Electrophysiology* **51**, 263–270.

Whitham, G. B. (2011). *Linear and Nonlinear Waves*. John Wiley & Sons.

Wu, H., Xue, H. and Kumar, A. (2012). Numerical discretization-based estimation methods for ordinary differential equation models via penalized spline smoothing with applications in biomedical research. *Biometrics* **68**, 344–352.

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

Yujie Zhao

Biostatistics and Research Decision Sciences (BARDS), Merck & Co. Inc., Rahway, NJ 07065, USA.

E-mail: littlebeannie0604@gmail.com

Xiaoming Huo

H. Milton Stewart School of Industrial and Systems Engineering, Georgia Tech, Atlanta, GA 30332, USA.

E-mail: huo@gatech.edu

Yajun Mei

H. Milton Stewart School of Industrial and Systems Engineering, Georgia Tech, Atlanta, GA 30332, USA.

E-mail: yajun.mei@isye.gatech.edu

(Received February 2022; accepted November 2022)