Identification of Underlying Dynamic System from Noisy Data with Spline

Yujie Zhao, Yajun Mei, Xiaoming Huo

October 15, 2020

Abstract

In many applications, it is important to identify the underlying dynamic models from noisy data. This is not a trivial work, since there are mainly two challenges to be overcome. The first challenge is that, the only data available is the noisy observations, where all the derivatives – which potentially govern the dynamic model – are unknown. However, naively estimating the derivatives by differentiation has very poor performance. So it is important for us to figure out a proper way to estimate derivatives. The second challenge is that, there could be lots of data-driven models that suit the noisy data very well. Among all these models, a simple model would be desired. However, it is not clear how can we identify the simple model.

In this paper, we propose a method called *Spline Assisted Partial Differential Equation involved Model Identification (SAPDEMI)* to efficiently identify the underlying dynamic models from noisy data. To solve the first challenge, we employ cubic spline to fit the noisy data. Then, under goodness of fit, it is reasonable for us to claim that, the derivative of the true dynamic models can be well approximated by the derivatives of the cubic splines. To solve the second challenge, we apply Least Absolute Shrinkage and Selection Operator (LASSO) to identify the significant derivatives (or their combinations). To ensure the correctness of the identification, we develop the sufficient conditions for correct identification, where the main tool we use is the primal-dual witness (PDW) condition. Finally, we validate our theory through various numerical examples.

Compared with the existing literature, we have several contributions. First, we realize the identification of underlying dynamic models, which can also be called "model selection" in the statistics terminology, while lots of existing literature focus on "parameter estimation". Second, we develop the asymptotic prosperity of identified model, which is rarely done in the literature.

1 Introduction

In the analysis of data from dynamic system, differential equations plays an important role, since they are widely used to model complex dynamic systems in various scientific

fields, including physics, engineering, economic and biomedical sciences. Among the research on differential equations, there are mainly two popular areas. The first one is so-called *forward problem*, i.e., given the known dynamic system, doing simulation and analysis of the behavior of state variables. This area has been extensively studied by mathematicians. The second area is the so-called *inverse problem*, i.e., given the noisy data, identifying the underlying models. The research on inverse problem is very useful in practice, which can be applied in lots of areas, including industrial engineer [see Wang et al., 2019], disease dynamic [see Liang and Wu, 2008; Wu et al., 2012], etc.. In our paper, we will focus on the inverse problem.

Although the inverse problem is firstly proposed in applied mathematics, we can also solve it by using statistics methods. The correlation between the inverse problem and statistics is described as follows. For the response vector of the inverse problem, we use the derivative with respective to (w.r.t) time, which is a measure of how fast the dynamic systems change. For the covariates of the inverse problem, we use a dictionary of other derivatives (or their combinations), for instance, the derivative w.r.t the spatial variables. (For the mathmatics details of response vector and covariates, please refer to Section 1.1 for more details.) Now it is clear that, under the statistics framework, the essence of the inverse problem is to predict the response vector by a dictionary of covariates, where both the response vector and covariates are derivatives.

After figuring out the correlation between inverse problem and statistics, we discuss how to solve the inverse problem in the statistics framework, which is not a easy job since we need to overcome two challenges. The first challenge is that, the only data available is the noisy observations, and all the derivatives – which construct the response vector and covariates – are unknown. This is different from the general framework of statistics in the sense that the covariates/response in statistics are always known, however, the covariates/response in the inverse problem remain unknown. The second challenge is that, there could be lots of data-driven models that fit the noisy data very well. Among all these models, a simple model is desire. However, it remains as a rarely-explored problem in the existing literature.

In this paper, we develop an efficient statistical methods, called *Spline Assisted Partial Differential Equation involved Model Identification (SAPDEMI)* to overcome the above two challenges in the inverse problem. To solve the first challenge, we employ cubic splines to fit the noisy data. Then, under goodness of fit, it is reasonable for us to claim that, the derivative of the true dynamic models can be well approximated by the derivatives of the cubic splines. To solve the second challenge, we apply Least Absolute Shrinkage and Selection Operator (LASSO) to identify the derivatives (or their combinations) that significantly govern the underlying models. To ensure the correctness of the identification, we develop the sufficient conditions for correct identification, where the main tool we use is the primal-dual witness (PDW) condition.

Compared with existing methods in the inverse problem, our proposed SAPDEMI method has several contributions. First, we focus on "model identification" in the inverse problem, which can also be called "model selection" in the statistics terminology.

However, most of the existing methods focus on "parameter estimation". Specifically, they assume that the format of the dynamic models are known, and their job is merely to estimate the coefficients in the given model [see Xun et al., 2013; Wang et al., 2019; Miao et al., 2009; Wu et al., 2012]. It is true that sometimes the format of the differential equations could be known. For example, in industrial, the format of the differential equation is usually developed by experts with a wealth of industrial experience. However, in many cases, experts have difficulties to define the formulation of the differential equation, due to a lack of thorough understanding on complex thermal systems or limited budgets for conducting extensive experiments. Under this scenario, we need to do "model identification", instead of "parameter estimation". Second, the correctness of our identified models can be guaranteed. We develop the sufficient conditions for correct model identification. Meanwhile, we analyze the asymptotic properties of the identified models, including the estimation erro bound of the identified models. This type of asymptotic properties is relative sparsely analyzed in the existing methods. For instance, Kang et al. [2019] demonstrated some empirical success of the LASSO in the inverse problem, but the rigorous theoretical justification of the usage of LASSO still remains vague.

The structure of this section is described as follows. In Section 1.1, we show the problem formulation we want to analyze in the remaining of the paper. In Section 1.2, we do a detailed literature review on solving inverse problem in the statistics framework.

1.1 Problem Formulation

In this section, we discuss the problem formulation we want to analyze in the remaining of the paper. In the content of Partial Differential Equation (PDE) identification problem, the data we have is a noisy dataset, which is generated from an unknown PDE model:

$$\mathcal{D} = \{(x_i, t_n, u_i^n) \mid i = 0, 1, \dots M-1, j = 0, 1, \dots N-1\} \subseteq (0, X_{\max}) \times (0, T_{\max}) \times \mathbb{R}.$$

Here $x_i \in \mathbb{R}$ is the spatial variable with $0 < x_i < X_{\max}$ for $i = 0, 1, \dots, M-1$. To highlight our main ideas, we only use x as an univariate variable as an illustration example on how cubic spline and LASSO can be applied to the inverse problem. But our proposed method can also be extended to multivariate, i.e., $x \in \mathbb{R}^d$ with some modification. Moreover, in this dataset, $t_n \in \mathbb{R}$ is the temporal variable with $0 < t_n < T_{\max}$ for $j = 0, 1, \dots, N-1$, and u_i^n is a representation of $u(x_i, t_n)$ contaminated by additive Gaussian noise:

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

Here, the function $u(x, t)$ is the desired function that governs the underlying model:

$$\frac{\partial}{\partial t}u(x,t) \;=\; \beta_{00} + \sum_{k=0}^{q}\sum_{i=0}^{p} \beta_{k^i} \left[\frac{\partial^k}{\partial^k x}u(x,t)\right]^i + \sum_{i,j\geq 0}\sum_{k,l\leq q} \beta_{k^i,l^j} \left[\frac{\partial^k}{\partial^k x}u(x,t)\right]^i \left[\frac{\partial^l}{\partial^l x}u(x,t)\right]^j, \tag{1}$$

where the left hand side of the above equation is the partial derivative w.r.t the temporal variable t, while the right side hand is a linear combination of polynomial of the derivatives w.r.t the spatial variable x.

In the above equation, there are lots of potential *feature variables* that could govern the PDE model, for example, $u(x,t)$, $\frac{\partial}{\partial x}u(x,t)$, $\frac{\partial^2}{\partial x^2}u(x,t)$ and so on. However, among all the feature variables candidates, only few of them can significantly govern the PDE model. Thus, it is reasonable to assume that, among all the feature variable candidates, only few of coefficient $\boldsymbol{\beta} = (\beta_{00}, \beta_0, \beta_1, \ldots)$ are significantly non-zero. In our paper, our objective is to identify the PDE models in (1), i.e., identify the non-zeros coefficient in $\boldsymbol{\beta}$.

As mentioned before, there are two challenges to solve the above problem. The first challenge is that, the only data available is the noisy observations $\mathcal{D}$, and all the derivatives, including $\frac{\partial}{\partial t}u(x,t)$, $\frac{\partial}{\partial x}u(x,t)$, $\frac{\partial^2}{\partial x^2}u(x,t), \ldots$, are unknown. The second challenge is that, how can we identify the significant none-zero coefficient in $\boldsymbol{\beta}$. In Section 2, we will discuss how to address this two challenges in details.

1.2 Literature Review

Recently, statisticians have started to develop various statistical methods to investigate the inverse problem. Most of them are focusing on Ordinal Differential Equation (ODE), and only few on Partial Differential Equation (PDE). In this section, we will discuss how the existing methods handle the two challenges mentioned in Section 1.

To solve the first challenge – given the noisy data $\mathcal{D}$, how to solve derivatives which builds the response/covariate – there are generally two types of approaches. The first approach is naive differentiation [see Brunton et al., 2016; Tran and Ward, 2017; Wu et al., 2012]. Take the derivative $\frac{\partial}{\partial t}u(x,t)$ as an example, differentiation naively approximate it as

$$\frac{\partial}{\partial t}u(x,t) \approx \frac{u(x+\Delta x,t) - u(x,t)}{\Delta x},$$

where $(x+\Delta x, t)$ is the closest point of (x,t) in the x-domain. The disadvantage of differentiation is that, it is not robust to noise. Although there are some de-noise technique can be applied after differentiation, such as total variation regularized derivative [see Chartrand, 2011], the estimation result of differentiation is still not satisfying. The second approach is to approximate the unknown dynamic curves, and use the derivatives of the approximated curves as the derivative of the underlying dynamic models. The approximation methods mainly fails in the framework of basis expansion, and the choice of basis can be various. One popular basis is the polynomial basis. For example, Bär et al. [1999] approximate unknown PDE models by multivariate polynomials of sufficiently high order. More examples related to the polynomial basis can be found in Schaeffer [2017]; Liang and Wu [2008]; Rudy et al. [2017]; Parlitz and Merkwirth [2000]; Voss et al. [1999]. Another popular choice of basis is spline basis [see Ramsay et al., 2007; Ramsay, 1996; Wu et al., 2012; Xun et al., 2013; Wang et al., 2019]. No matter which type of basis to chose, the approximation is always satisfying. However, the computational complexity of different basis varies lots. In our paper, we choose cubic spline to save computational complexity.

To solve the second challenge – how to identify a simple model – there are typical two approaches. The first type of methodology widely used are Bayesian approaches. For example, Putter et al. [2002]; Huang and Wu [2006]; Huang et al. [2006] developed hierarchical

Bayesian approaches to estimate dynamic ODE model parameters in HIV dynamic models for longitudinal data. This method requires repeatedly solving ODE models numerically, which could be computationally expensive. These researchers focus on the inverse problem in the ODE models. These papers deal with identifying ODE by Bayesian approaches, but the Bayesian approaches can also be used in PDE models [see Xun et al., 2013]. The second type of methodology widely used is to (penalize) least square. In the ODE framework, for example, Liang and Wu [2008] develop a two-stage method for a general first-order ODE model, where the authors use local polynomial regression in the first stage to derive the derivatives, and then use least square model to identify the ODE system in the second stage. Similar methodology is also be used by Miao et al. [2009], where the authors use least square method to estimate the parameters in unknown ODE system. In the PDE framework, Bär et al. [1999] and Wu et al. [2012] use least square to estimate the parameters. And Xun et al. [2013] and Wang et al. [2019] penalized the smoothes of the unknown PDE models, which shares similar ideas with reproducing kernel Hilbert space (RKHS).

The above papers make lots of contribution in the inverse problem, however, there are still some limitations. First, these papers we mentioned above mostly focus on ODE, instead of PDE, because generally speaking PDE is much complicated than ODE. Second, they focus more on "parameter estimation", instead of "model identification"/"model selection". Specifically, because the existing methods mainly use ℓ_2 penalty (ridged regression), so their models fail to identify models in a simple format. For example, the coefficients found by Xun et al. [2013] and Wang et al. [2019] are not be sparse enough, since the authors use ℓ_2 penalty instead of ℓ_1 penalty. If the model is not simple enough, then it would be difficult for to discriminate the type of the dynamic system, which leads the characteristic of the dynamic systems very vague. However, in our paper, we use ℓ_1 (LASSO) penalty, which is capable to identify models in simple formats. Third, the asymptotic property of the identified models is seldom analyzed in the literature. For example, Schaeffer [2017] approximate the unknown dynamic system via local polynomial regression, and then used proximal mapping to efficient solve the least square with ℓ_1 penalty. Yet, the statistical property is not theoretical developed by Schaeffer [2017]. Kang et al. [2019] utilize LASSO to select candidate monomials in PDE models, and then proposed the time evolution error to select the underlying true model. Although Kang et al. [2019] demonstrated some empirical success of the LASSO in PDE identification, the rigorous theoretical justification of the usage of LASSO still remains vague. In this paper, we develop the nice asymptotic property the models identified by our proposed SAPDEMI method.

It should be acknowledged that, in our paper, we assume the dynamic model is not time-varying, i.e., both the governing derivatives and their coefficients are fixed as time goes by. Researchers also explore the case when the coefficients are time-varying, see Chen and Wu [2008a,b]; Li et al. [2002] for more details. In this paper, we will not focus on this scenario because our objective is to show the usage of LASSO an cubic spline in PDE inverse problem. But the time-varying PDE inverse problem is a promising research topic, and we will leave it for future research.

The remaining of the paper is organized as follows. In Section 2, we propose our

SAPDEMI method. In Section 3, we represent our main theory. In Section 4, we conduct numerical experiments to validate the theory in Section 3.

2 Proposed Method: SAPDEMI

In this section, we develop an efficient statistical method called SAPDEMI to identify the PDE model from noisy data $\mathcal{D}$. Recall in Section 1, to achieve this goal, there are mainly two challenges. The first challenge is that, the only data available is the noisy observations $\mathcal{D}$, and all the derivatives, including $\frac{\partial}{\partial t}u(x,t)$, $\frac{\partial}{\partial x}u(x,t)$, $\frac{\partial}{\partial x^2}u(x,t)$, etc., are unknown. The second challenge is that, how can we identity PDE models in a simple format. In this section 2, we will discuss how to address this two challenges one by one. In Section 2.1, we address the first challenge. In Section 2.2, we address the second challenge. In Section 2.3, we design an initial point to facilitate the calculation of identifying process in Section 2.2.

2.1 Solve Derivatives in Terms of the Sample Data

This section discuss how to address the first challenge, i.e., how to estimate the derivatives from the noisy data $\mathcal{D}$. These derivatives include the derivatives w.r.t spatial variable x and the derivatives w.r.t temporal variable t.

The main tool we use is the cubic spline, and the main idea to implement cubic spline is described as follows. First, to approximate the true dynamic curve, we fit the raw data $\mathcal{D}$ by cubic spline. Then, under goodness of fit, it is reasonable for us to claim that the derivative of the true dynamic system can be well approximated by the derivatives of the cubic spline.

It should be acknowledged that, since we use cubic spline, the highest order derivative is the second derivative. In other words, in our case, $q = 2$ in (1). We use cubic spline as an illustration example due to its simplification and computational efficiency. However, our proposed SAPDEMI method can also be extended to $q > 2$ if cubic spline is replaced with higher order spline. Yet, in real practice, most of the PDE models are governed by derivatives up to second derivative, for instance, heat equation, wave equation, Laplace's equation, Helmholtz equation, Poisson's equation and so on. Therefore, our proposed method can still works well in real practice.

In the remaining of this section, we will take the derivatives w.r.t spatial variable x as an illustration example, i.e., the estimation of $u(x,t)$, $\frac{\partial}{\partial x}u(x,t)$, $\frac{\partial^2}{\partial x^2}u(x,t)$, and the derivative w.r.t temporal variable t can be derived similarly.

To obtain the derivatives w.r.t spatial variable x, we will first fix the temporal variable t. Without loss of generality, we suppose we fix t as t_n for any $n = 0, \ldots, N-1$. Furthermore, we also assume that the spatial variable x is sorted in nondecreasing order, i.e., $x_0 < x_1 \ldots < x_{M-1}$. Then we would like to fit the data $\{(x_i, u_i^n)\}_{i=0,\ldots,M-1}$ into the cubic spline $s(x_i)$ for $x \in [x_0, x_{M-1}]$, where the fitting is done by minimizing the following weighted sum of

squares:

$$\min_s \sum_{i=0}^{M-1} w_i[u_i^n - s(x_i)]^2,$$

where $w_0, w_1, \ldots, w_{M-1} > 0$ are the weights accordingly. At the same time, we would like to control the smoothness of the cubic spline $s(x)$. And we define the smoothness of $s(x)$ as

$$\int_{x_0}^{x_{M-1}} s''(x)^2 dx,$$

where $s''(x)$ is the second derivative of $s(x)$. Considering the trade off between smoothes of s and goodness of fit, the cubic spline is developed as follows:

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

where α trades off the goodness of fit and the smoothness of the cubic spline. By minimizing $J_\alpha(s)$ with respect to s, we can get the estimated smoothing cubic spline $s(x)$. If the cubic spline approximate the underlying PDE curves very well, then we could declare that the derivatives of the underlying dynamic system, i.e., $u(x,t_n)$, $\frac{\partial}{\partial x}u(x,t_n)$, $\frac{\partial^2}{\partial x^2}u(x,t_n)$, can be well approximate by $s(x), s'(x), s''(x)$ respectively for any $x \in \{x_0, x_1, \ldots, x_{M-1}\}$.

A nice property of cubic spline is that, its derivatives can be solved explicitly, and we summarize the result in the following theorem.

Theorem 2.1. Assume that

1. the spatial variable x in the dataset $\mathcal{D} = \{(x_i, t_n, u_i^n) \mid i = 0, 1, \ldots M-1, j = 0, 1, \ldots N-1, \} \subseteq (0, X_{\max}) \times (0, T_{\max}) \times \mathbb{R}$, is sorted in nondecreasing order, i.e., $x_0 < x_1 < \ldots < x_{M-1}$;
2. the smoothing cubic polynomial spline on each node-to node interval $[x_i, x_{i+1}]$ is $s(x)$;
3. the second derivative of $s(x)$ at the two end points are zero, i.e., $s''(x_0) = s''(x_{M-1}) = 0$.

For the fixed $t_n (n = 0, 1, \ldots N-1)$, the zero-order derivative $\widehat{\mathbf{f}} = (u(x_0,t_n), u(x_1,t_n), \ldots, u(x_{M-1},t_n))^\top$ of the underlying PDE model can be estimated by cubic spline in (2) as

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

and the first-order derivative $\widehat{\boldsymbol{\theta}} = \left(\frac{\partial}{\partial x}u(x_0,t_n), \frac{\partial}{\partial x}u(x_1,t_n), \ldots, \frac{\partial}{\partial x}u(x_{M-1},t_n)\right)^\top$ can be estimated as

$$\widehat{\boldsymbol{\theta}} = \mathbf{Q}^{-1}\mathbf{B}\widehat{\mathbf{f}} = [\alpha\mathbf{Q} + (1-\alpha)\mathbf{B}\mathbf{W}^{-1}\mathbf{B}^\top]^{-1}\alpha\mathbf{B}\mathbf{u}_:^n,$$

and the second-order derivative $\widehat{\boldsymbol{\sigma}} = \left(\frac{\partial^2}{\partial x^2}u(x_0,t_n), \frac{\partial^2}{\partial x^2}u(x_1,t_n), \ldots, \frac{\partial^2}{\partial x^2}u(x_{M-1},t_n)\right)^\top$ can be estimated as

$$\widehat{\boldsymbol{\sigma}} = \mathbf{M}^{-1}\mathbf{A}\widehat{\mathbf{f}} = [\alpha\mathbf{M} + (1-\alpha)\mathbf{A}^\top\mathbf{W}\mathbf{A}]^{-1}\alpha\mathbf{A}\mathbf{u}_:^n,$$

where matrix $\mathbf{W} = \mathrm{diag}(w_0, w_1, \ldots, w_{M-1})$, vector $\mathbf{u}_{:}^{n} = \left(u_0^n, \ldots, u_{M-1}^n\right)^\top$, and matrix $\mathbf{A}, \mathbf{B}, \mathbf{Q}, \mathbf{M}$ are defined as

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

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

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

$$\mathbf{M} = \begin{pmatrix} \frac{h_0+h_1}{3} & \frac{h_1}{6} & 0 & \ldots & 0 & 0 \\ \frac{h_1}{6} & \frac{h_1+h_2}{3} & \frac{h_2}{6} & \ldots & 0 & 0 \\ 0 & \frac{h_2}{6} & \frac{h_2+h_3}{3} & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & \frac{h_{M-4}+h_{M-3}}{3} & \frac{h_{M-3}}{6} \\ 0 & 0 & 0 & \ldots & \frac{h_{M-3}}{6} & \frac{h_{M-3}+h_{M-2}}{3} \end{pmatrix}.$$

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

Proof. See Appendix B. □

A special case of Theorem 2.1 is when $\alpha = 1$, which means that there is no penalty on smoothness. When there is no penalty for the smoothness, the cubic spline are always called *interpolating cubic spline*, because $s(x_i) = u_i^n$ for $i = 0, 1, \ldots, M-1$ with a fixed n. Under this scenario, the derivatives w.r.t the spatial variable x can be estimated by the following corollary.

Corollary 2.1. When $\alpha = 1$, Theorem 2.1 becomes:
the zero-order derivative $\widehat{\mathbf{f}} = (u(x_0, t_n), u(x_1, t_n), \ldots, u(x_{M-1}, t_n))^\top$ is estimated as

$$\widehat{\mathbf{f}} = \left(u_0^n, u_1^n, \ldots, u_{M-1}^n)\right)^\top,$$

and the first-order derivative $\widehat{\boldsymbol{\theta}} = \left(\frac{\partial}{\partial x}u(x_0,t_n), \frac{\partial}{\partial x}u(x_1,t_n), \ldots, \frac{\partial}{\partial x}u(x_{M-1},t_n)\right)^\top$ is estimated as

$$\widehat{\boldsymbol{\theta}} = \mathbf{Q}^{-1}\mathbf{B}\widehat{\mathbf{f}},$$

and the second-order derivative $\widehat{\boldsymbol{\sigma}} = \left(\frac{\partial^2}{\partial x^2}u(x_0,t_n), \frac{\partial^2}{\partial x^2}u(x_1,t_n), \ldots, \frac{\partial^2}{\partial x^2}u(x_{M-1},t_n)\right)^\top$ is estimated as

$$\widehat{\boldsymbol{\sigma}} = \mathbf{M}^{-1}\mathbf{A}\widehat{\mathbf{f}},$$

and matrix $\mathbf{A}, \mathbf{B}, \mathbf{M}, \mathbf{Q}$ is the same as in Theorem 2.1.

Following the similar logic with solving derivative w.r.t x, the derivative w.r.t temporal variable x can be estimated similarly. The only difference between the estimation of the derivative w.r.t x and the estimation of the derivative w.r.t t is that, in the estimation of the derivative w.r.t x, we fix the t, however, in the estimation of the derivative w.r.t t, we will fix the x. The estimation results can be found in the following corollary.

Corollary 2.2. Assume that

1. the temporal variable t in the dataset $\mathcal{D} = \{(x_i, t_n, u_i^n) \mid i = 0, 1, \ldots M-1, j = 0, 1, \ldots N-1, \} \subseteq (0, X_{\max}) \times (0, T_{\max}) \times \mathbb{R}$, is sorted in nondecreasing order, i.e., $t_0 < t_1 < \ldots < t_{N-1}$;
2. the smoothing cubic polynomial spline on each node-to node interval $[t_n, t_{n+1}]$ is $\underline{s}(t)$;
3. the second derivative of $\underline{s}(t)$ at the two end points are zero, i.e., $\underline{s}(t_0)'' = \underline{s}(t_{N-1})'' = 0$.

For the fixed $x_i (i = 0, 1, \ldots, M-1)$, the first-order derivative w.r.t the temporal variable t, i.e., $\underline{\widehat{\boldsymbol{\theta}}} = \left(\frac{\partial}{\partial t}u(x_i,t_0), \frac{\partial}{\partial t}u(x_i,t_1), \ldots, \frac{\partial}{\partial t}u(x_i,t_{N-1})\right)^\top$ is estimated by cubic spline in (12) as

$$\underline{\widehat{\boldsymbol{\theta}}} = \underline{\mathbf{Q}}^{-1}\underline{\mathbf{B}}\,\underline{\widehat{\mathbf{f}}} = [\underline{\alpha}\,\underline{\mathbf{Q}} + (1-\underline{\alpha})\underline{\mathbf{B}}\,\underline{\mathbf{W}}^{-1}\underline{\mathbf{B}}^\top]^{-1}\alpha\underline{\mathbf{B}}\mathbf{u}_i^{\cdot},,$$

where $\mathbf{u}_i^{\cdot} = \left(u_i^0, u_i^1, \ldots, u_i^{N-1}\right)^\top$ and matrix $\underline{\mathbf{A}}, \underline{\mathbf{B}}, \underline{\mathbf{M}}, \underline{\mathbf{Q}}$ is the defined in a similar way as matrix$\mathbf{A}, \mathbf{B}, \mathbf{M}, \mathbf{Q}$ as in Theorem 2.1 with $h_i = x_{i+1} - x_i$ replaced by $h_i = t_{i+1} - t_i$ for $i = 0, 1, \ldots, N-2$.

By far, we solve the first challenge in the PDE inverse problem, i.e., estimate the derivatives from noisy data. In the next section, we will discuss how to solve the second challenge in the PDE inverse problem, i.e., identify the underlying PDE model.

2.2 Identify the Unknown PDE Model

In this section, we discuss how to solve the second challenge, i.e., identify a simple PDE model where the noisy data $\mathcal{D}$, which is equivalent to identify the non-zero coefficient in $\boldsymbol{\beta}$ in (1). To achieve this objective, we introduce LASSO and estimate $\boldsymbol{\beta}$ as

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \frac{1}{2MN} \sum_{i=0}^{M-1}\sum_{n=0}^{N-1}\left(\widehat{\frac{\partial u(x_i,t_n)}{\partial t}} - \widehat{\mathbf{x}_i^n}^\top\boldsymbol{\beta}\right)^2 + \lambda\|\boldsymbol{\beta}\|_1, \tag{3}$$

where $\lambda > 0$ is a turning parameter which trade off the sparsity of $\boldsymbol{\beta}$ and goodness of fit, $\widehat{\frac{\partial u(x_i,t_n)}{\partial t}}$ is the first derivative w.r.t t estimated from Section 2.1, and

$$\widehat{\mathbf{x}_i^n} = \left(1, \widehat{u(x_i,t_n)}, \left[\widehat{u(x_i,t_n)}\right]^2, \ldots, \left[\widehat{u(x_i,t_n)}\right]^p, \ldots, \left[\widehat{\frac{\partial u(x_i,t_n)}{\partial x}}\right]^{p-1} \left[\widehat{\frac{\partial^2 u(x_i,t_n)}{\partial x^2}}\right]\right)^\top,$$

Given the ℓ_1 penalty in (1), $\widehat{\boldsymbol{\beta}}$ will be sparse, i.e., only few of its entries will be non-zero. Accordingly, we can identify the underlying PDE model in a simple format as

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

where

$$\mathbf{x} = \left(1, \widehat{u(x,t)}, \left[\widehat{u(x,t)}\right]^2, \ldots, \left[\widehat{u(x,t)}\right]^p, \ldots, \left[\widehat{\frac{\partial u(x,t)}{\partial x}}\right]^{p-1} \left[\widehat{\frac{\partial^2 u(x,t)}{\partial x^2}}\right]\right)^\top.$$

Now we will discuss how to solve the optimization problem in (3). By introducing matrix algebra, (3) can be equivalently written as

$$\arg\min_{\boldsymbol{\beta}} \frac{1}{2MN}\|\widehat{\nabla_t \mathbf{u}} - \widehat{\mathbf{X}}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1 \tag{4}$$

where vector $\widehat{\nabla_t \mathbf{u}}$ and matrix $\widehat{\mathbf{X}}$ are defined as

$$\begin{array}{rcccccccc} \widehat{\nabla_t \mathbf{u}} & = & (\widehat{\frac{\partial u(x_0,t_0)}{\partial t}} & \widehat{\frac{\partial u(x_1,t_0)}{\partial t}} & \ldots & \widehat{\frac{\partial u(x_{M-1},t_0)}{\partial t}} & \widehat{\frac{\partial u(x_0,t_1)}{\partial t}} & \ldots & \widehat{\frac{\partial u(x_{M-1},t_{N-1})}{\partial t}})^\top, \\ \widehat{\mathbf{X}} & = & (\widehat{\mathbf{x}_0^0} & \widehat{\mathbf{x}_1^0} & \ldots & \widehat{\mathbf{x}_{M-1}^0} & \widehat{\mathbf{x}_1^0} & \ldots & \widehat{\mathbf{x}_{M-1}^{N-1}})^\top. \end{array}$$

Essentially, the optimization problem in (4) is a LASSO-type optimization, and there are various algorithms can be applied. One of the widely used algorithm to solve LASSO is coordinate descent, because it is well established and an corresponding R package named *glmnet* [see Friedman et al., 2010]. And this method is also implemented in the function *lasso(·)* in Matlab.

In the remaining of this section, for completeness, we briefly review the implement of coordinate descent algorithm in Friedman et al. [2010] to solve (4). The main idea of coordinate descent is to update the estimator in a coordinate-wise fashion, which is the main difference between coordinate descent and regular gradient descent. In the k-th iteration, coordinate descent update the iterative estimator $\boldsymbol{\beta}^{(k)}$ by using partial of the gradient information, instead of the whole gradient information. Mathematically speaking, coordinate descent optimize $F(\boldsymbol{\beta}) = \frac{1}{2MN}\|\widehat{\nabla_t \mathbf{u}} - \widehat{\mathbf{X}}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1$ by

$$\beta_j^{(k+1)} = \arg\min_{\beta_j} F((\beta_{00}^{(k)}, \ldots, \beta_{j-1}^{(k)}, \beta_j, \beta_{j+1}^{(k)}, \ldots, \beta_{1,2}^{(k)}))$$

where $\boldsymbol{\beta}_j$ is the j-th index of $\boldsymbol{\beta}$. To minimize the above optimization problem, it is equivalent to set the first derivative as 0:

$$\frac{\partial}{\partial \beta_j} F(\boldsymbol{\beta}^{(k)}) = \frac{1}{MN}\left(\mathbf{e}_j^\top \widehat{\mathbf{X}}^\top \widehat{\mathbf{X}} \boldsymbol{\beta}^{(k)} - \widehat{\nabla_t \mathbf{u}}^\top \widehat{\mathbf{X}} \mathbf{e}_j\right) + \lambda \text{sign}(\boldsymbol{\beta}_j) = 0,$$

where $\mathbf{e}_j$ is a vector of length K whose entries are all zero expect the j-th entry is 1. And K is the number of feature variable candidates. For example, if we take all the interaction into account, then $K = 10$, and if we ignore the interaction, then $K = 7$. By solving the above equation, we can solve $\boldsymbol{\beta}_j^{(k+1)}$ by

$$\boldsymbol{\beta}_j^{(k+1)} = S\left(\widehat{\nabla_t \mathbf{u}}^\top \widehat{\mathbf{X}} \mathbf{e}_j - \sum_{l \neq j} (\mathbf{X}^\top \mathbf{X})_{jl} \boldsymbol{\beta}_l^{(k)}, MN\lambda\right) \Big/ (\mathbf{X}^\top \mathbf{X})_{jj},$$

where $S(\cdot)$ is the soft-thresholding function defined as

$$S(x, \alpha) = \begin{cases} x - \alpha & \text{if } x \geq \alpha \\ x + \alpha & \text{if } x \leq -\alpha \\ 0 & \text{otherwise} \end{cases}$$

The algorithm for coordination descent to solve optimization problem in (4) is summarized in Algorithm 1.

Algorithm 1: Algorithm for coordinate descent to minimize $F(\boldsymbol{\beta})$

Input: dataset $\mathcal{D}$, number of iterations K
Output: coefficient estimation $\widehat{\boldsymbol{\beta}}$
1 **Initialize** $\boldsymbol{\beta}^{(0)}$
2 **for** $i = 1, \ldots, K$ **do**
3 **for** $p = 1, \ldots, 10$ **do**
4 $\boldsymbol{\beta}_j^{(k)} = S\left(\widehat{\nabla_t \mathbf{u}}^\top \widehat{\mathbf{X}} \mathbf{e}_j - \sum_{l \neq j} (\mathbf{X}^\top \mathbf{X})_{jl} \boldsymbol{\beta}_l^{(k-1)}, MN\lambda\right) \Big/ (\mathbf{X}^\top \mathbf{X})_{jj}$
5 $\widehat{\boldsymbol{\beta}} = \boldsymbol{\beta}^{(K)}$

It is proved by Beck and Tetruashvili [2013]; Tseng [2001] that, Algorithm 1 converges with a convergence rate $O(1/k)$, where k is the number of iteration executed. One more detail we would like to discuss is how to select the initial point $\boldsymbol{\beta}^{(0)}$ in Line 1 of Algorithm 1. We will discuss how to design the initial point $\boldsymbol{\beta}^{(0)}$ in Section 2.3.

2.3 Initial Point to Begin the Algorithm in PDE Identification

In this section, we discuss the initial point to begin Algorithm 1, i.e., the selection of $\boldsymbol{\beta}^{(0)}$ in Line 1 of Algorithm 1. It is important to choose a good initial point, because a good initial point will help us reduce the number of iterations to achieve the desired accuracy, which would save lots of time when the sample size is overwhelmingly large.

Motivated by the fact that ℓ_1 penalty is closely related with ℓ_2 penalty, we design our initial point as:

$$\boldsymbol{\beta}^{(0)} = \arg\min_{\boldsymbol{\beta}} \frac{1}{2MN} \|\widehat{\nabla_t \mathbf{u}} - \widehat{\mathbf{X}}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2 \tag{5}$$

The only difference between (5) and (4) is that, (5) uses ℓ_2 penalty while (4) uses ℓ_1 penalty. Given the quadratic property of the ℓ_2 penalty and the sum of square residuals, there is a closed-form solution to $\boldsymbol{\beta}^{(0)}$:

$$\boldsymbol{\beta}^{(0)} = (\widehat{\mathbf{X}}^\top \widehat{\mathbf{X}} + \lambda MN \mathbf{I})^{-1} \widehat{\mathbf{X}}^\top \widehat{\nabla_t \mathbf{u}},$$

where $\mathbf{I}$ is a identity matrix of dimension $K \times K$. The advantage of designing the initial point in the above way is that, its computational complexity is small:

Theorem 2.2. Given data $\mathcal{D} = \{(x_i, t_n, u_i^n) \mid i = 0, 1, \ldots M-1, j = 0, 1, \ldots N-1, \} \subseteq (0, X_{\max}) \times (0, T_{\max}) \times \mathbb{R}$, if $\alpha = 1$, then the computation complexity to derive $\boldsymbol{\beta}^{(0)}$ defined as in equation (5) is of order $O(MN)$.

Proof. See Appendix D. □

As suggested by Theorem 2.2, it only require $O(MN)$ operations to derive $\boldsymbol{\beta}^{(0)}$. Since M is the number of spatial variables (spatial resolution), and N is the number of temporal variables (temporal resolution), the product of MN is exactly the number of the sample size. It can be concluded that, it would be very efficient for us to calculate $\boldsymbol{\beta}^{(0)}$, because the computational complexity to derive $\boldsymbol{\beta}^{(0)}$ is a linear polynomial of the sample size.

For comparision, we show the computational complexity of the local polynomial regression. Local polynomial regression is widely used to approximate the unknown PDE/ODE models [see Liang and Wu, 2008]. For example, in order to estimate the derivative $\frac{\partial^k}{\partial x^k} u(x_i, t_n)$, local polynomial regression approximates the unknown PDE model as a locally fitting a degree $\check{p}$ polynomial over the data:

$$u(x_i, t_n) = u(x, t_n) + \frac{\partial}{\partial x} u(x, t_n)(x_i - x) + \ldots + \frac{\partial^{\check{p}}}{\partial x^{\check{p}}} u(x, t_n)(x_i - x)^{\check{p}}.$$

For the selection of $\check{p}$, people always choose it as $\check{p} = k + 3$ [Fan et al., 1997, see], where k is the order of the derivative we want to estimate.

By approximating $u(x_i, t_n)$ by polynomial basis of order $\check{p}$, then $\frac{\partial^k}{\partial x^k} u(x, t_n)$ can be estimated as

$$\widehat{\frac{\partial^k u(x, t_n)}{\partial x^k}} = \widehat{\mathbf{b}_{k+1}(x)},$$

where $\widehat{\mathbf{b}_{k+1}(x)}$ represents the $(k+1)$th entry of $\widehat{\mathbf{b}(x)}$ and $\widehat{\mathbf{b}(x)}$ is obtained by the following optimization problem:

$$\widehat{\mathbf{b}(x)} = \arg\min_{\mathbf{b}(x)} \sum_{i=0}^{M-1} \left[u_i^n - \sum_{j=0}^{\check{p}} \frac{\partial^j}{\partial x^{\check{p}}} u(x, t_n)(x_i - x)^j \right]^2 \mathcal{K}\left(\frac{x_i - x}{h} \right), \tag{6}$$

where $\mathcal{K}(\cdot)$ is the Epanechnikov kernel function defined by $\mathcal{K}(z) = \frac{3}{4}\max\{1-z^2, 0\}$.

By implementing the local polynomial in this way, the computational complexity is much higher than our method, and we summarize its computational complexity in the following theorem.

Theorem 2.3. Given data $\mathcal{D} = \{(x_i, t_n, u_i^n) \mid i = 0, 1, \dots M-1, j = 0, 1, \dots N-1, \} \subseteq (0, X_{\max}) \times (0, T_{\max}) \times \mathbb{R}$, the computation complexity to derive $\boldsymbol{\beta}^{(0)}$ defined in equation (5) is of order $\max\{O(NM^2), O(MN^2), O(5^3NM)\}$, where $\widehat{\nabla_t \mathbf{u}}, \widehat{\mathbf{X}}$ is derived by local polynomial regression in (6).

Proof. See Appendix F. □

As the above theorem suggests, compared with cubic spline, the computational complexity of local polynomial regression is much higher than that in cubic spline. However, we do admit the merit of local polynomial regression, that is, it can derive any order of derivatives. Yet, for cubic spline, the highest order of derivatives is the second-order derivatives.

2.4 Overview of Our Proposed SAPDEMI method

In this section, we summarized our propose SAPDEMI method into pseudo code showing in Algorithm 2.

Algorithm 2: Pseudo code of our proposed SAPDEMI method

Input:

1. data from unknown PDE system:
   $\mathcal{D} = \{(x_i, t_n, u_i^n) \mid i = 0, 1, \ldots M-1, j = 0, 1, \ldots N-1, \} \subseteq (0, X_{\max}) \times (0, T_{\max}) \times \mathbb{R}$;
2. penalty parameter used in the LASSO identify model: λ;
3. smoothing parameter used in the smoothing cubic spline: α.

Output: The identified/recovered PDE model.

Initialize:

$\boldsymbol{\beta}^{(0)} = \arg\min_{\boldsymbol{\beta}} \frac{1}{2MN} \|\widehat{\nabla_t \underline{\mathbf{u}}} - \widehat{\underline{\mathbf{X}}}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2$, where vector $\widehat{\underline{\mathbf{X}}}$ and $\widehat{\nabla_t \underline{\mathbf{u}}}$ are derived by Theorem 2.1 and Theorem 2.2, respectively with $\alpha = 1$.

PDE model identification:

The unknown PDE system is recovered as:

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

where $\mathbf{x} = \left(1, \widehat{u(x,t)}, \left[\widehat{u(x,t)}\right]^2, \ldots, \left[\widehat{u(x,t)}\right]^p, \ldots, \left[\widehat{\frac{\partial u(x,t)}{\partial x}}\right]^{p-1} \left[\widehat{\frac{\partial^2 u(x,t)}{\partial x^2}}\right]\right)^\top$,

and

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

where vector $\widehat{\mathbf{X}}$ and $\widehat{\nabla_t \mathbf{u}}$ are derived by Theorem 2.1 and Theorem2.2, respectively with $0 < \alpha < 1$.

3 Recovery Theory

In this section, we represent our main theorems. These two main theorems serves to evaluate the asymptotic prosperity of our identified PDE model. The evaluation is done from two aspects. First, we are interested to check out if our identified PDE model contains correct feature variables from the underlying system. This is the so-called *support recovery*, i.e., $\text{supp}(\widehat{\boldsymbol{\beta}}) \subseteq \text{supp}(\boldsymbol{\beta}^*)$. However, the support recovery depends on the choice of the penalty parameter λ. The proper way to select λ is discuss in Theorem 3.1. Second, we are interested in the estimation error bound of our estimator, i.e., $\|\widehat{\boldsymbol{\beta}}_{\mathcal{S}} - \boldsymbol{\beta}^*_{\mathcal{S}}\|_\infty$, and this error bound is analyzed in Theorem 3.2.

Before developing the main theorems, we first list the model assumptions in Section 3.1. And then, we present our main theorem in Section 3.2.

3.1 Model Assumptions

In this section, we introduce some key conditions used in our paper. We begin with three frequently used conditions in ℓ_1-regularized regression models, i.e., *invertibility condition, mutual incoherence condition*, and *minimal eigenvalue condition*. They were typically used to prove sufficient conditions for exact sparse recovery [see Friedman et al., 2001, Chapter 11]. In our context, they carry special meanings and impose general properties on the observed solution u and the underlying PDEs for a successful PDE identification from the given data.

Invertibility condition — Possibility of identification of PDE from a single solution

$$\widehat{\mathbf{X}}_{\mathcal{S}}^{\top}\widehat{\mathbf{X}}_{\mathcal{S}} \text{ is invertible, almost surely.} \tag{A1}$$

Mutual incoherence condition — Exhibition of signature variation For some *incoherence parameter* $\mu \in (0, 1]$ and $P_\mu \in [0, 1]$:

$$\mathbb{P}\left[\left\|\widehat{\mathbf{X}}_{\mathcal{S}^c}^{\top}\widehat{\mathbf{X}}_{\mathcal{S}}(\widehat{\mathbf{X}}_{\mathcal{S}}^{\top}\widehat{\mathbf{X}}_{\mathcal{S}})^{-1}\right\|_{\infty} \leq 1-\mu\right] \geq P_\mu \ . \tag{A2}$$

Minimal eigenvalue condition —Threshold on noticeable magnitude There exists some constant $C_{\min} > 0$ such that:

$$\Lambda_{\min}\left(\frac{1}{NM}\widehat{\mathbf{X}}_{\mathcal{S}}^{\top}\widehat{\mathbf{X}}_{\mathcal{S}}\right) \geq C_{\min} \text{ , almost surely.} \tag{A3}$$

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 strengthen (A1). Similarly to (A2), (A3) concerns the quantitative properties of the feature variables associated with the solution u. However, (A3) generally does not involve the characteristic variation of u that is unique to the combination of the underlying feature variables.

Next, we introduce several conditions that is frequently assumed in smoothing spline [see Silverman, 1984, (2.5)-(2.8)].

Convergence c.d.f Suppose for all fixed n, we have F_M^n, which is sequence of probability functions with $F_M^n(u_{(0)}^n) = 0, F_M^n(u_{(M-1)}^n) = 1$ for all M ($u_{(i)}^n$ is the i-th order statistics of $\{u_i^n\}_{i=0,\dots,M-1}$. In the application to the spline smoothing problem, the function F_n will be the empirical distribution function of the design points $\{u_i^n\}_{i=0,\dots,M-1}$. There exists an absolutely continuous distribution function F^n on $[u_{(0)}^n, u_{(M-1)}^n]$ such that

$$F_M^n \to F^n \tag{A4}$$

uniformly as $M \to +\infty$.

Bounded p.d.f. Setting $f^n = F'^n$, we have

$$0 < \inf_{[u_{(0)}^n, u_{(M-1)}^n]} f^n \leq \sup_{[u_{(0)}^n, u_{(M-1)}^n]} f^n < +\infty \tag{A5}$$

and the density f^n has bounded first derivatives on $[u^n_{(0)}, u^n_{(M-1)}]$.
Decreasing penalty parameter Setting $\alpha(M,n) = \sup_{[u^n_{(0)}, u^n_{(M-1)}]} |F^n_M - F^n|$, the smoothing parameter λ depends on M in such a way that

$$\lambda \to 0 \ \text{ and } \ \lambda^{-1/4}\alpha(M,n) \to 0 \ \text{ as } \ M \to +\infty \tag{A6}$$

for any fixed $n = 0, \ldots, N-1$.
This condition ensures that the smoothing parameter does not tend to zero too rapidly.
Bounded Equivalent Kernel $K(\cdot)$ is uniformly continuous with modulus of continuity ω_K and of bounded variation $V(K)$. And

$$\int \sqrt{|x\log(|x|)|}|dK(x)| < +\infty \tag{A7}$$

3.2 Main Theory

In this section, we present our main theory, where Theorem 3.1 develops the lower bound of λ to realize the correct support set, and Theorem 3.2 develops the upper bound of the estimation error.

Theorem 3.1. Provided with the data $\mathcal{D} = \{(x_i, t_n, u^n_i) : i = 0, \ldots, M-1,\ n = 0, \ldots, N-1\} \in \Omega$ and supposed the assumptions (A1)-(A6) hold. And $u^*(x, t_n) \in C^5, u^*(x_i, t) \in C^4$. When $M = N$ and

$$\lambda > \frac{\sqrt{K}}{\mu}\frac{32\log N}{LN^{57/140}}, \tag{7}$$

then with probability greater than

$$P_\mu - \underbrace{4Ne^{-\frac{(N^{16/105} - \|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}}}_{P'} \to P_\mu, \ \text{ as } \ N \to \infty,$$

we have

$$\mathcal{S}(\widehat{\boldsymbol{\beta}}) \subseteq \mathcal{S}(\boldsymbol{\beta}^*)$$

and $\widehat{\boldsymbol{\beta}}$ is unique.

Proof. See Appendix J. To ease the understanding of the proofs of main theorem, readers can refer to some preliminaries in Appendix A. □

The above theorem states the lower bound to realize the proper support set recovery. And this lower bound in (7) is affected by several factors. First, it is affected by the temporal resolution N: as N increase, there is more flexibility in tuning this penalty parameter λ. Also the lower bound in (7) is affected by the incoherence parameter μ: if μ is small, then the lower bound increases. This is because that, small μ means that the group of feature variable candidates are similar to each other.

We also point out that, $P_\mu - P'$ is indeed a probability for sufficiently large N, and P' reduces to 0 exponentially fast after certain amount of data is collected. It can be

seen from Theorem 3.1 that P' depends on the temporal resolution N, the magnitude of the noise σ, as well as the magnitude of the solution bound $\|u\|_{L^\infty(\Omega)}$. To investigate the dependence of $N, \sigma, \|u\|_{L^\infty(\Omega)}$, we plot Figure 1. As seen in Figure 1(a), we find that when the observed data is contaminated by heavy noise, it requires large temporal resolution N to guarantee the conclusion in Theorem 3.1 with high probability. As seen in Figure 1(b), we find the conclusion in Theorem 3.1 is still guaranteed with high probability, because even if the functional magnitude is high, P' can still be reduced to less than 1 if we have large temporal resolution N.

It is worth noting that the limiting probability P_μ is determined by the data $\mathcal{D}$ (see Condition (A2)). Thus, different data and different underlying PDE vary the conclusion in Theorem 3.1. But this is what we can't control, because P_μ is the original attribution of the data and underlying PDE.

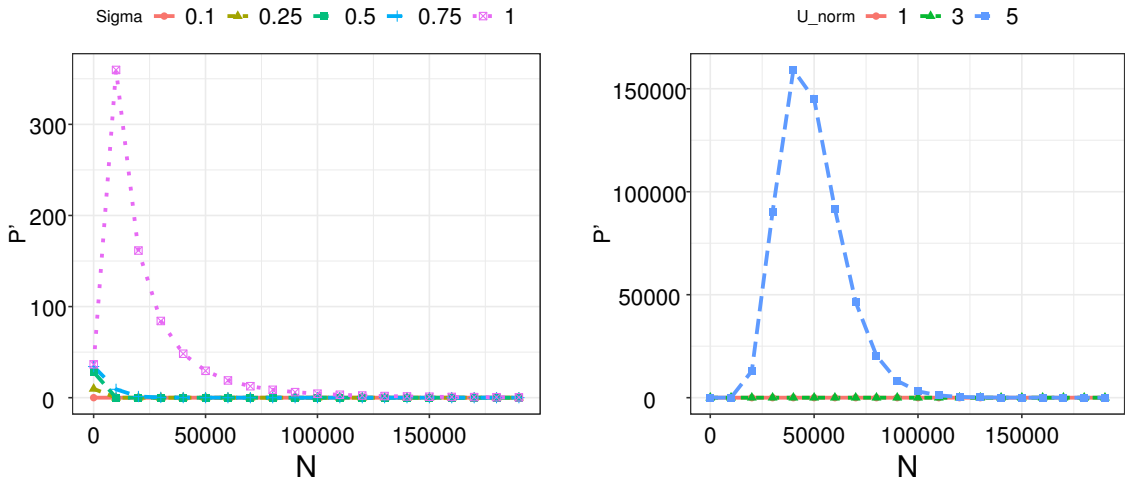


Figure 1: Factors that influences P' in Theorem 3.1

Now, we develop the theorem to obtain the estimation error bound in the following theorem.

Theorem 3.2. Suppose the conditions for Theorem 3.1 hold, then with probability greater than

$$P_\mu - 4Ne^{-\frac{(N^{5/9}-\|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}} \to P_\mu, \quad \text{as} \quad N \to \infty,$$

we have

$$\left\|\widehat{\boldsymbol{\beta}}_{\mathcal{S}} - \boldsymbol{\beta}^*_{\mathcal{S}}\right\|_\infty \leq \sqrt{K} C_{\min} \left[\left(2\sqrt{K} N e^{-\frac{(N^{16/105}-\|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}} + \frac{\left\|\widehat{\mathbf{X}}^*_{\mathcal{S}}\right\|_1}{N^2} \right) 4Ne^{-\frac{(N^{16/105}-\|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}} + \lambda \right]. \tag{8}$$

Proof. See Appendix K. □

From the above theorem, we can see that, the estimation error bound for the ℓ_∞-norm of the coefficient error in (8) consists of two components.

The first component is affected by the temporal resolution N, the number of feature variable candidates K, the magnitude of the solution bound $\|u\|_{L^\infty(\Omega)}$, and the magnitude of the noise σ. As N increase to $+\infty$, this first component convergence to 0 without explicit dependence on the choice of feature variable selected from (4).

The second component is $\sqrt{K}C_{\min}\lambda$. When N increase to $+\infty$, this second component does not vary. However, noted from Theorem 3.1, we find that N increase to $+\infty$, the lower bound of λ to realize correct support recovery converges to 0. As an overall effect, the accuracy of the coefficient estimation will improve if we increase the temporal resolution N.

An important implication of Theorem 3.2 is the correct signed support. Many PDEs are sensitive to the sign of the coefficients. For example, changing the sign of the advection term in transport equation reverses the moving direction, and changing the sign of the Laplacian term of the heat equation leads to instability. Theorem 3.2 guarantees the correct signs provided that the magnitudes of the coefficients of the correct feature variables are larger than a threshold same as (8). Asymptotically, the recovered coefficients are close to the true ones, and the signs are correct even for those with small absolute values.

4 Simulations

This section realizes the numerical examples to validate the main theory we develops ahead. Section 4.1 validates the computational complexity of the initial point. Section 4.2 validates the theory on the support set recovary.

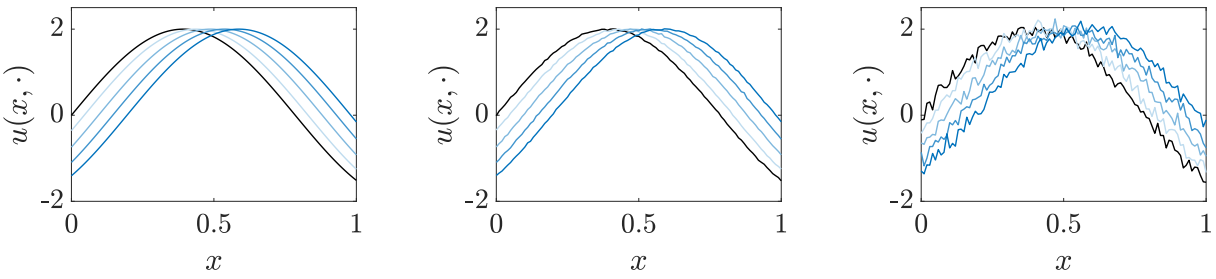
4.1 Simulation 1: Verify the Computational Complexity

This simulation is to validate Theorem 2.2 and Theorem 2.3. These two theorem 2.2 shows that, the computational complexity of our method is much lower than that of local polynomial regression.

The PDE problem we used in this section is transport equation:

$$\begin{cases} \frac{\partial}{\partial t}u(x,t) &= a\frac{\partial}{\partial x}u(x,t) & \forall 0 \le x \le 1, 0 \le t \le T_{\max}; \\ u(x,0) &= f(x) & \forall 0 \le x \le 1; \end{cases} \tag{9}$$

We set $f(x) = 2\sin(4x), a = -2, X_{\max} = 1, T_{\max} = 1$, and accordingly, the solution is $u(x,t) = 2\sin(4x - 8t)$. Figure 2 shows the shape of the above transport equation, where (a) of Figure 2 is the truegroundth, while (b) and (c) of Figure 2 is the noised observations. (b) of Figure 2 is noised by Gaussian noise with zero mean and stand dividation 0.01, andn (c) of Figure 2 is noised by Gaussian noise with zero mean and stand dividation 0.1. Figure 2 shows that, when noise is very large, the shape of the PDE system would be more unsmoothed.



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

Figure 2: The curve of the transport equation ($M = 100, N = 40$)

Theoretically, as stated in Theorem 2.2, the complexity for cubic spline to derive a initial point is $O(MN)$. Thus, when M is fixed, the complexity is a order-1 polynomial of N. Similarly, when N is fixed, the complexity is a order-1 polynomial of M. Besides, as stated in Theorem 2.3, the complexity for local polynomial to derive a initial point is $\max\{O(MN^2), O(M^2N), O(5^3MN)\}$. Thus, we know that when M is fixed and $N > 125$, the complexity is dominated by $O(N^2)$. Similarly, when N is fixed and $M > 125$, the complexity is is dominated by $O(M^2)$.

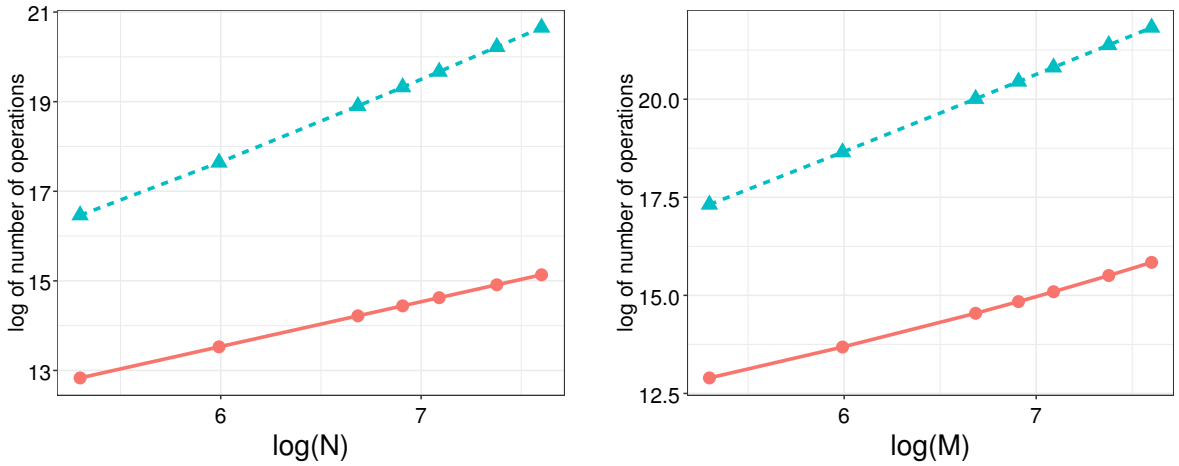
The above results in Theorem 2.2, 2.3 are numerically validated in Table 1. The first four lines in Table 1 deals with the scenario when M(spatial resolution) is fixed and N(temporal resolution) ranges from a small number to a large number. The last four lines in Table 1 deals with the scenario when N(temporal resolution) is fixed and M(spatial resolution) ranges from a small number to a large number. Besides, we also visualize Table 1 in Figure 3.

From this figure, we know that, for cubic spline (Theorem 2.2), when M is fixed (left hand side of Figure 3), the slope of the cubic spline (red solid line) is 0.9998, which validate that the computational complexity of cubic spline when M is fixed is of order $O(N)$. And as N goes to infinity, the slope will get closer to 1. Besides, when N is fixed (right hand side of Figure 3), the slope of the cubic spline (red solid line) is 1.274, which validate that the computational complexity of cubic spline when N is fixed is of order $O(M)$. And as M goes to infinity, the slope will get closer to 1.

Similarly, for local polynomial (Theorem 2.3), we know that, when M is fixed (left hand side of Figure 3), the slope of the local polynomial regression (blue dashed line) is 1.822, which validate that the computational complexity of cubic spline when M is fixed is of order $O(N^2)$. And as N goes to infinity, the slope will get closer to 2. Besides, when N is fixed (right hand side of Figure 3), the slope of the local polynomial regression (blue dashed line) is 1.960, which validate that the computational complexity of local polynomial regression when N is fixed is of order $O(M^2)$. And as M goes to infinity, the slope will get closer to 2.

	M = 20						
	N=200	N=400	N=800	N=1000	N=1200	N=1600	2000
cubic spline	**374389**	**748589**	**1496989**	**1871189**	**2245389**	**2993789**	**3742189**
local poly	14136936	45854336	162089136	246606536	348723936	605758736	933193536
	N = 20						
	M=200	M=400	M=800	M=1000	M=1200	M=1600	M=2000
cubic spline	**398573**	**875773**	**2070173**	**2787373**	**3584573**	**5418973**	**7573373**
local poly	33046336	125596136	489255736	760365536	1090995336	1930814936	3008714536

Table 1: Computational complexity of cubic spline



(a) (b)

Figure 3: (a) Computational complexity of cubic spline (red solid line) & local polynomial regression (blue dashed line) with fixed M=20, (b) computational complexity of cubic spline (red solid line) & local polynomial regression (blue dashed line) with fixed N=20

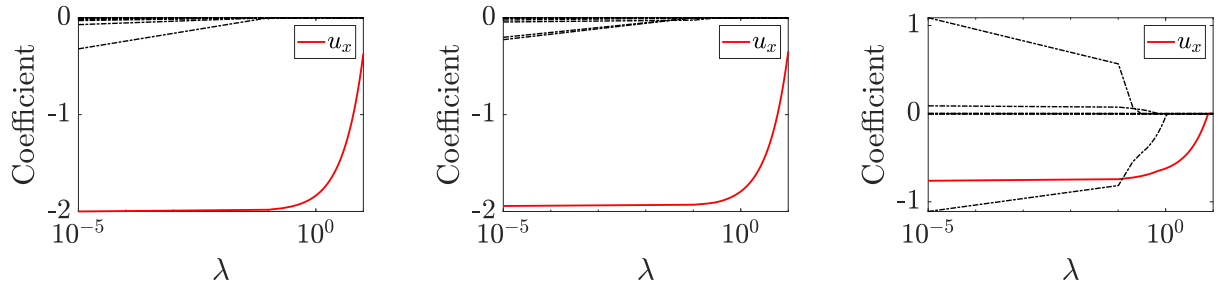
4.2 Simulation 2: Varify the Support Set Recovery Theory

In this section, we apply our proposed SAPDEMI method to identify classical PDE models under noised observations. Transport equation and Burger's equation are considered, since both of them plays a fundamental role in modeling physical phenomenon and demonstrate characteristic behaviors shared by more complex system, such as dissipation and shock-formation. Readers can refer to Haberman [1983] for more derivations. For both transport equation and Burger's equation, we aim to analyze the effect of noise on PDE identification.

4.2.1 Example 1: Transport Equation

Transport equation is considered in this section, whose formulation is the same as that in Section 4.1. The objective in this example is to correctly identify the unknown PDE dynamic system from the noisy dataset $\mathcal{D}$. From equation (9), we know that the correct feature variable is $\frac{\partial}{\partial x}u(x,t)$. While other feature variable, for example, $u(x,t)$, $\frac{\partial^2}{\partial x^2}u(x,t)$ etc., should not be selected. After correctly selecting the $\frac{\partial}{\partial x}u(x,t)$, we can do linear regression between $\frac{\partial}{\partial t}u(x,t)$ and $\frac{\partial}{\partial x}u(x,t)$ and get a nice estimation of a.

The model identifying result is shown in Figure 4 . The x-axis of Figure 4 is λ, which increase from a very small number to a large number. The y-axis of Figure 4 is the coefficients prior to all candidates of feature variables. Plot (a) in Figure 4 shows the support set recovery when the noise is zero mean and has standard deviation 0.01. Plot (b) in Figure 4 shows the support set recovery when the noise is zero mean and has standard deviation 0.1. Plot (c) in Figure 4 shows the support set recovery when the noise is zero mean and has standard deviation 1. The red line in (a), (b), (c) represents the coefficient prior to $\frac{\partial}{\partial x}u(x,t)$, which is the correct feature variable. While the black dashed lines represent the coefficient prior to other incorrect feature variables, which shouldn't be selected. It can be seen from Figure 4 that, when the noise is very small ($\sigma = 0.01$), the support set recovery is very easy. However, when the noise became large ($\sigma = 0.1$), the support set recovery is not as easy as the small noise scenario. That is, to realize correct support set recovery, we need larger lambda.



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

Figure 4: The effect of noise on support set recovery of transport equation ($M = 100, N = 40, a = -2.X_{\max} = 1, T_{\max} = 1, f(x) = 2\sin(4x - 8t)$.)

4.2.2 Example 2: Burger's Equation

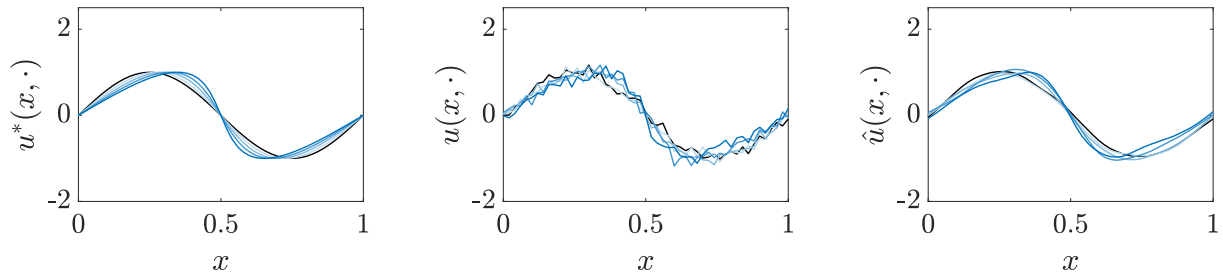
Burgers' equation is considered in this section, whose definition is shown as follows:

$$\begin{cases} \frac{\partial}{\partial t}u(x,t) &= -\frac{1}{2}u(x,t)\frac{\partial}{\partial x}u(x,t) + \nu\frac{\partial^2}{\partial x^2}u(x,t); \\ u(x,0) &= f(x), \quad 0 \le x \le 1; \\ u(0,t) &= u(1,t) = 0, \quad 0 \le t \le T_{\max} \end{cases} \tag{10}$$

When $\nu = 0$, it is called *inviscid Burgers' equation*, and when $\nu > 0$, it is called *viscous Burgers' equation*. We will consider both these two type of Burgers' equation in this section. And for inviscid Burgers' equation, we set $f(x) = \sin(2\pi x), T_{\max} = 0.1$, while for viscous Burgers' equation, we set $f(x) = \sin^2(4\pi x) + \sin^3(2\pi x), T_{\max} = 0.1$.

First, let's discuss inviscid Burgers' equation. Figure 5 shows the shape of the inviscid Burgers' equation, where (a) is the truegrounth, and (b) is the noised observation, and (c) is the de-noised version by applying SAPDEMI with the cubic spline.

The identification of inviscid Burgers' equation is shown in Figure 6. The correct feature variable in inviscid Burgers' equation is $u(x,t)\frac{\partial}{\partial x}u(x,t)$, whose coefficient is marked with red in Figure 6. The other coefficient with incorrect feature variable is marked with dashed black lines. Plot (a) in Figure 6 show the identification results when the noise is zero mean with standard deviation 0.01. Plot (b) in Figure 6 show the identification results when the noise is zero mean with standard deviation 0.1. Plot (c) in Figure 6 show the identification results when the noise is zero mean with standard deviation 1. We can see from Figure 6 that, when noise is very small ($\sigma = 0.01$), small λ can realize correct support set recovery. However, when noise is large ($\sigma = 0.1$), larger λ is required for correct support set recovery. But when the noise is too large ($\sigma = 1$), it may become very hard for us to identification the PDE models. This is because that, when noise is very large, the shape of the PDE models become too twisted to capture its dynamic changes.

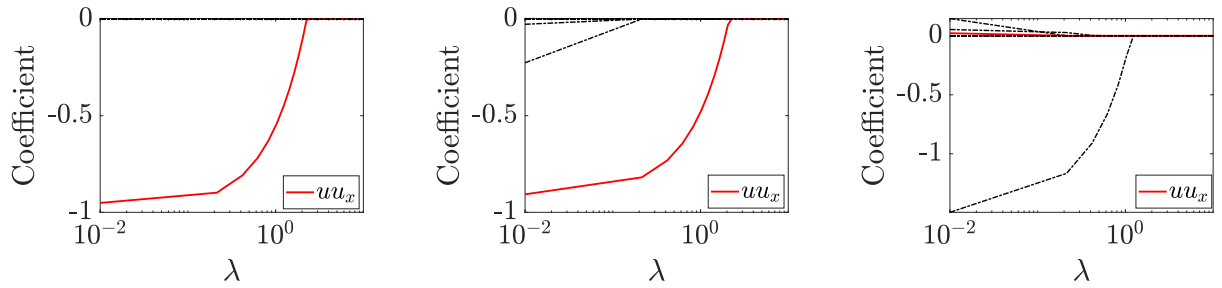


(a)true (b) noised(observed) (c)de-noised

Figure 5: The curve of the inviscid Burgers' equation ($M = 50, N = 50, \alpha = 0.8, \sigma = 0.1$)

Second, let's discuss viscous Burgers' equation. Figure 7 shows the shape of the viscous Burgers' equation, where (a) is the truegrounth, and (b) is the noised observation, and (c) is the de-noised version by applying SAPDEMI with the smoothing cubic spline.

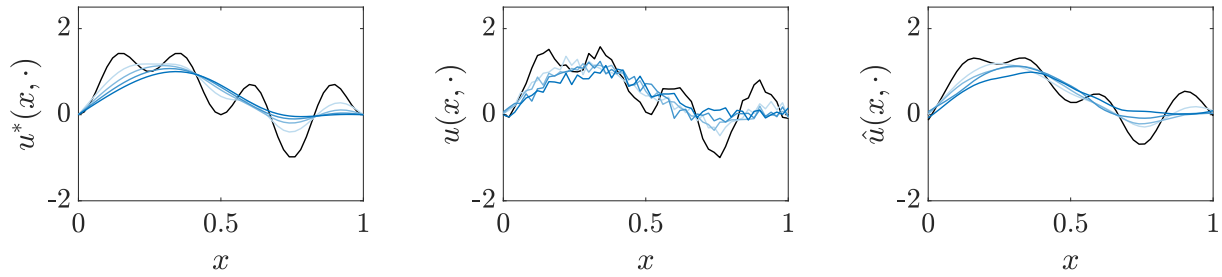
The identification of viscous Burgers' equation is shown in Figure 6. The correct feature variable in viscous Burgers' equation is $u(x,t)\frac{\partial}{\partial x}u(x,t)$ and $\frac{\partial^2}{\partial x^2}u(x,t)$, whose coefficients



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

Figure 6: The effect of noise on support set recovery ($M = 50, N = 50, \alpha = 0.8$) for the inviscid Burger's equation

are marked with blue and red in Figure 8, respectively. The other coefficient with incorrect feature variable is marked with dashed black lines. Plot (a) in Figure 8 show the identification results when the noise is zero mean with standard deviation 0.01. Plot (b) in Figure 8 show the identification results when the noise is zero mean with standard deviation 0.1. Plot (c) in Figure 8 show the identification results when the noise is zero mean with standard deviation 1. We can see from Figure 8 that, when noise is very small ($\sigma = 0.01$), small λ can realize correct support set recovery. However, when noise is large ($\sigma = 0.1$), larger λ is required for correct support set recovery. But when the noise is too large ($\sigma = 1$), it may become very hard for us to identification the PDE models. This is because that, when noise is very large, the shape of the PDE models become too twisted to capture its dynamic changes.

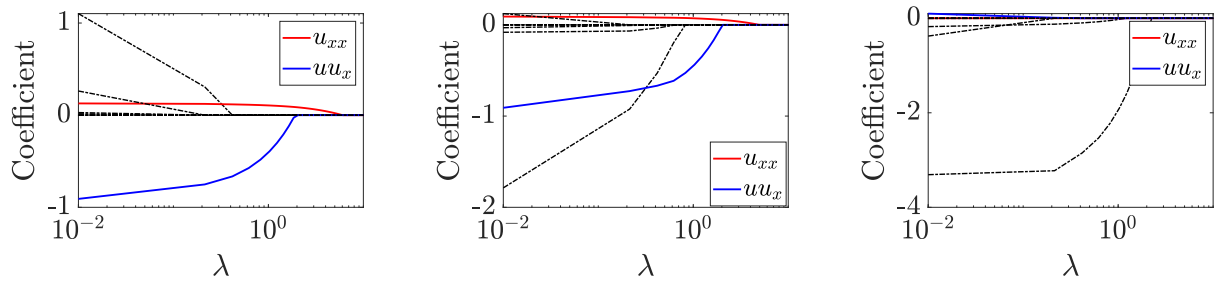


(a) true (b) noised(observed) (c) de-noised

Figure 7: The curve of the viscous Burgers' equation ($M = 50, N = 50, \alpha = 0.8, \sigma = 0.1, \nu = 0.1$)

4.3 Simulation 3: Validation of Theorem 3.1: Lower Bound of λ

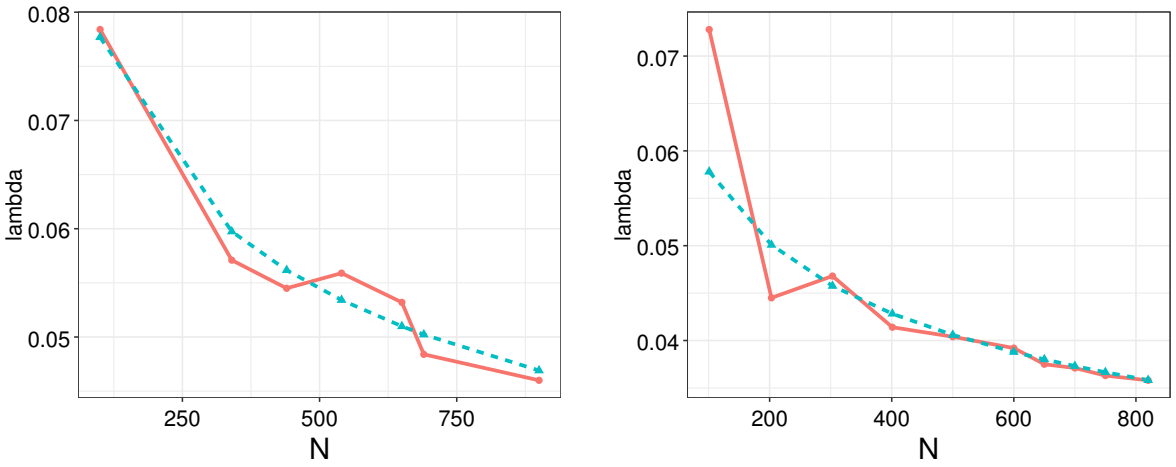
In this section, an numerical example is done to validate Theorem 3.1. The main idea convey by Theorem 3.1 is that, the minimal λ to correctly recover $\mathcal{S}(\boldsymbol{\beta}^*)$ decreases in the same order as $\frac{\log(N)}{N^{57/140}}$ as $N \to +\infty$. We validate this in two different types of PDE



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

Figure 8: The effect of noise on support set recovery ($M = 50, N = 50, \alpha = 0.8$) for the viscous Burger's equation

problems. The first type of PDE is transport equation (see equation (9)), and the noise level is $\sigma = 0.1$. We set the initial condition as $f(x) = 2\sin(4x)$ and $a = -2$, with the temporal/spatial variable ranges from 0 to 1. This setting is the same as that in Section 4.2.1. And the second type of PDE is the inviscid Burgers' equation (see equation (10)), and the noise level is $\sigma = 0.1$. Because it is a inviscid Burgers' equation, according to it definition, we set $\nu = 0$. Besides, we choose the initial condition is $f(x) = \sin(2\pi x)$ with temporal variable range from 0 to $T_{\max} = 0.1$. This setting is the same as that in Section 4.2.2. In both (a) and (b) of Figure 9, the blue dashed line represents the numerical results and the red solid line represents the product of a constant and $\frac{\log(N)}{N^{57/140}}$. These constants in front of $\frac{\log(N)}{N^{57/140}}$ are found by simple linear regression. It can be seen from Figure 9 that, the lower bound of λ that can properly recovers the support set is the red curve, which is some multiple of $\frac{\log(N)}{N^{57/140}}$. Thus, in both cases, the variation of the lower bound for λ is well captured by the ratio $\frac{\log(N)}{N^{57/140}}$ as $N \to +\infty$.



(a) transport equation (b) inviscid Burgers' equation

Figure 9: Numerical and theoretical lower bound of λ. Left side hand (a) shows the result when the underlying PDE is transport equation (see equation (9)), and the noise level is $\sigma = 0.1$. Right side hand (b) shows the result when the underlying PDE is the inviscid Burgers' equation (see equation (10)), and the noise level is $\sigma = 0.1$ and $\nu = 0$. In both (a) and (b), the blue dashed line represents the numerical results and the red solid line represents the product of a constant and $\frac{\log(N)}{N^{57/140}}$. These constants in front of $\frac{\log(N)}{N^{57/140}}$ are found by simple linear regression. As predicted in Theorem 3.1, the minimal λ to correctly recover $\mathcal{S}(\boldsymbol{\beta}^*)$ decreases in the same order as $\frac{\log(N)}{N^{57/140}}$ as $N \to +\infty$.

A Some Preliminaries

In this section, we present some important preliminaries, which are important blocks for the main theories. To begin with, we first give the upper bound of $\widehat{u(x,t_n)} - u^*(x,t_n)$, which is the error bound of the zero-order derivatives.

Lemma A.1. For any fixed $n = 0, 1, \ldots, N-1$, denote $\eta^2 = \max_{i=0,\ldots,M-1} E(U_i^n)^2$, $K_{\max} = \|K\|_\infty$. Suppose $u^*(x,t_n) \in C^4$ and $\partial_x^2 u^*(0,t_n) = \partial_x^2 u^*(1,t_n) = 0, \partial_x^3 u^*(0,t_n) \neq 0, \partial_x^3 u^*(1,t_n) = 0$ and the assumption (A4) $\sim$ (A6) hold as well. There exist finite positive constant L, which do not depend on the temporal resolution M, such that with sufficiently large M, for any ϵ satisfying

$$\epsilon > \frac{32 \log M}{L M^{57/140}}$$

we have

$$P\left[\sup_{x\in[0,X_{\max}]} \left|\widehat{u(x,t_n)} - u^*(x,t_n)\right| > \epsilon\right] < 2Me^{-\frac{(M^{16/105} - \|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}}$$

Proof. See Appendix G. □

Then we give the upper bound of $\partial_x \widehat{u(x,t_n)} - \partial_x u^*(x,t_n)$, which is the error bound of the first-order derivatives.

Lemma A.2. For any fixed $n = 0, 1, \ldots, N-1$, suppose $u^*(x,t_n) \in C^4$ and we denote $\eta^2 = \max_{i=0,\ldots,M-1} E(U_i^n)^2$. There exist finite positive constant L, which do not depend on the temporal resolution M, such that with sufficiently large M, for any ϵ satisfying

$$\epsilon > \frac{32 \log M}{L M^{57/140}}$$

we have

$$P\left[\sup_{x\in[0,X_{\max}]} \left|\partial_x \widehat{u(x,t_n)} - \partial_x u^*(x,t_n)\right| > \epsilon\right] < 2Me^{-\frac{(M^{16/105} - \|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}}$$

Proof. See Appendix H. □

Next, we give the upper bound of $\partial_t \widehat{u(x,t_n)} - \partial_t u^*(x,t_n)$, which is the error bound of the first-order derivatives.

Lemma A.3. For any fixed $i = 0, 1, \ldots, M-1$, suppose $u^*(x_i,t) \in C^4$ and we denote $\eta^2 = \max_{n=0,\ldots,N-1} E(U_i^n)^2$. There exist finite positive constant L, which do not depend on the temporal resolution N, such that with sufficiently large N, for any ϵ satisfying

$$\epsilon > \frac{32 \log M}{L M^{57/140}}$$

we have

$$P\left[\sup_{t\in[0,T_{\max}]} \left|\partial_x \widehat{u(x_i,t)} - \partial_x u^*(x_i,t)\right| > \epsilon\right] < 2Me^{-\frac{(M^{16/105} - \|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}}$$

Proof. See Appendix H. □

Finally, we give the upper bound of $\widehat{\partial_x^2 u(x,t_n)} - \partial_x^2 u^*(x,t_n)$, which is the error bound of the second-order derivatives.

Lemma A.4. For any fixed $n = 0, 1, \ldots, N-1$, suppose $u^*(x,t_n) \in C^5$ and we denote $\eta^2 = \max_{i=0,\ldots,M-1} E(U_i^n)^2$. There exist finite positive constant L, which do not depend on the temporal resolution M, such that with sufficiently large M, for any ϵ satisfying

$$\epsilon > \frac{32 \log M}{L M^{57/140}}$$

we have

$$P\left[\sup_{x\in[0,X_{\max}]} \left|\widehat{\partial_x u(x,t_n)} - \partial_x u^*(x,t_n)\right| > \epsilon\right] < 2Me^{-\frac{(M^{16/105} - \|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}}$$

After bounding the error of all the derivatives, we then aim to bound $\|\nabla_t \mathbf{u} - \mathbf{X}\boldsymbol{\beta}^*\|_\infty$. The reason why we want to bound $\|\nabla_t \mathbf{u} - \mathbf{X}\boldsymbol{\beta}^*\|_\infty$ is described as follows:

By KKT-condition, any minimizer $\boldsymbol{\beta}$ of the following optimization

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

satisfies:

$$-\frac{1}{MN}\mathbf{X}^\top(\nabla_t \mathbf{u} - \mathbf{X}\boldsymbol{\beta}) + \lambda \mathbf{z} = 0 \ \text{ for } \ \mathbf{z} \in \partial\|\boldsymbol{\beta}\|_1,$$

where $\partial\|\boldsymbol{\beta}\|_1$ is the sub-differential of $\|\boldsymbol{\beta}\|_1$. The above equation can be equivalently transformed into

$$\mathbf{X}^\top\mathbf{X}(\boldsymbol{\beta} - \boldsymbol{\beta}^*) + \mathbf{X}^\top\left[(\mathbf{X} - \mathbf{X}^*)\boldsymbol{\beta}^* - (\nabla_t \mathbf{u} - \nabla_t \mathbf{u}^*)\right] + \lambda MN\mathbf{z} = 0$$

We decomposed the above equation as follows:

$$\begin{pmatrix} \mathbf{X}_\mathbf{S}^\top\mathbf{X}_\mathbf{S} & \mathbf{X}_\mathbf{S}^\top\mathbf{X}_{\mathbf{S}^c} \\ \mathbf{X}_{\mathbf{S}^c}^\top\mathbf{X}_\mathbf{S} & \mathbf{X}_{\mathbf{S}^c}^\top\mathbf{X}_{\mathbf{S}^c} \end{pmatrix}\begin{pmatrix} \boldsymbol{\beta} - \boldsymbol{\beta}^* \\ \boldsymbol{\beta}_{\mathcal{S}^c} \end{pmatrix} + \begin{pmatrix} \mathbf{X}_\mathcal{S}^\top \\ \mathbf{X}_{\mathcal{S}^c}^\top \end{pmatrix}\left[(\mathbf{X} - \mathbf{X}^*)_\mathcal{S}\boldsymbol{\beta}_\mathcal{S}^* - (\nabla_t \mathbf{u} - \nabla_t \mathbf{u}^*)\right] + \lambda MN\begin{pmatrix} \mathbf{z}_\mathcal{S} \\ \mathbf{z}_{\mathcal{S}^c} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}$$

Suppose the primal-dual witness (PDW) construction gives us an solution $(\check{\boldsymbol{\beta}}, \check{\mathbf{z}}) \in \mathbb{R}^K \times \mathbb{R}^K$, where $\check{\boldsymbol{\beta}}_{\mathcal{S}^c} = 0$ and $\check{\mathbf{z}} \in \partial\|\check{\boldsymbol{\beta}}\|_1$. By plugging $(\check{\boldsymbol{\beta}}, \check{\mathbf{z}})$ into the above equation, we have

$$\check{\mathbf{z}}_{\mathcal{S}^c} = \mathbf{X}_{\mathcal{S}^c}^\top\mathbf{X}_\mathcal{S}(\mathbf{X}_\mathcal{S}^\top\mathbf{X}_\mathcal{S})^{-1}\mathbf{z}_\mathcal{S} - \mathbf{X}_{\mathcal{S}^c}^\top\underbrace{(\mathbf{I} - \mathbf{X}_\mathcal{S}(\mathbf{X}_\mathcal{S}^\top\mathbf{X}_\mathcal{S})^{-1}\mathbf{X}_\mathcal{S}^\top)}_{\mathbf{H}}\frac{\left[(\mathbf{X} - \mathbf{X}^*)_\mathcal{S}\boldsymbol{\beta}_\mathcal{S}^* - (\nabla_t \mathbf{u} - \nabla_t \mathbf{u}^*)\right]}{\lambda MN}$$

By Lemma A.1 in Namjoon's PDE paper, we have

$$P\left[\max_{j\in\mathcal{S}^c}|\widetilde{Z_j}| > \epsilon\right] \le P\left[\|\nabla_t \mathbf{u} - \mathbf{X}\boldsymbol{\beta}^*\|_\infty > \epsilon\right],$$

where $\widetilde{Z_j} = (\mathbf{X}_{\mathcal{S}^c})_j^\top(\mathbf{I} - \mathbf{X}_\mathcal{S}(\mathbf{X}_\mathcal{S}^\top\mathbf{X}_\mathcal{S})^{-1}\mathbf{X}_\mathcal{S}^\top)\frac{\left[(\mathbf{X} - \mathbf{X}^*)_\mathcal{S}\boldsymbol{\beta}_\mathcal{S}^* - (\nabla_t \mathbf{u} - \nabla_t \mathbf{u}^*)\right]}{\lambda MN} = (\mathbf{X}_{\mathcal{S}^c})_j^\top\mathbf{H}\frac{\left[(\mathbf{X} - \mathbf{X}^*)_\mathcal{S}\boldsymbol{\beta}_\mathcal{S}^* - (\nabla_t \mathbf{u} - \nabla_t \mathbf{u}^*)\right]}{\lambda MN}$$
So we only need to bound $\|\nabla_t \mathbf{u} - \mathbf{X}\boldsymbol{\beta}^*\|_\infty$, whose bound can be found in Lemma A.5.

Lemma A.5. Suppose $u^*(x,t_n) \in C^5$ and we denote $\eta^2 = \max_{i=0,\ldots,M-1} E(U_i^n)^2$.

$$\epsilon > \max\left\{\frac{32\log M}{LM^{57/140}}, \frac{32\log M}{LM^{57/140}}\right\}$$

we have

$$P\left(\|\nabla_t \mathbf{u} - \mathbf{X}\boldsymbol{\beta}^*\|_\infty > \epsilon\right) < 2Me^{-\frac{(M^{16/105} - \|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}} + 2Ne^{-\frac{(N^{16/105} - \|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}}$$

Proof. See Appendix I. □

B Proof of the Derivative of the Cubic Spline

Proof. We focus on solving the zero-order / first-order / second-order derivatives of $u(x,t_n)$ with respective to x for fixed t_n. First, let's solve the second-order derivatives of $u(x,t_n)$ with respective to x, i.e., $\widehat{\partial_x^2 u(x,t_n)}$ for $x \in \{x_0, x_1, \ldots, x_{M-1}\}$.

Suppose the cubic polynomial spline on each node-to node interval $[x_i, x_{i+1}]$ is $s(x)$. We further denote the cubic polynomial spline on interval $[x_i, x_{i+1}]$ is $s_i(x)$.

First, we solve the zero-order derivatives of $s(x)$. By introducing matrix algebra, the objective function in equation (2) can be rewritten as

$$J_\alpha(s) = \alpha(\mathbf{u}_\cdot^n - \mathbf{f})^\top \mathbf{W}(\mathbf{u}_\cdot^n - \mathbf{f}) + (1-\alpha)\mathbf{f}^\top \mathbf{A}^\top \mathbf{M}^{-1}\mathbf{A}\mathbf{f} \tag{11}$$

where $\mathbf{W} = \mathrm{diag}(w_0, w_1, \ldots, w_{M-1})$,

$$\mathbf{f} = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{M-1} \end{pmatrix} = \begin{pmatrix} s(x_0) \\ s(x_1) \\ \vdots \\ s(x_{M-1}) \end{pmatrix}, \mathbf{u}_\cdot^n = \begin{pmatrix} u_0^n \\ u_1^n \\ \vdots \\ u_{M-1}^n \end{pmatrix}$$

and

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

By taking the derivative of 11 with respective to $\mathbf{f}$ and set it as zero, we have

$$\widehat{\mathbf{f}} = [\alpha\mathbf{W} + (1-\alpha)\mathbf{A}^\top\mathbf{M}\mathbf{A}]^{-1}\alpha\mathbf{W}\mathbf{u}_\cdot^n$$

Second, we solve the second-order derivative with respective to x. Because $s_i''(x)$ is linear within $[x_i, x_{i+1}]$, we can assume that $s_i''(x_i) = \sigma_i, s''(x_{i+1}) = \sigma_{i+1}$ we have ($0 \le i \le M-2$)

$$\begin{array}{rrcl} & s_i''(x) & = & \sigma_i \frac{x_{i+1}-x}{h_i} + \sigma_{i+1}\frac{x-x_i}{h_i} \\ \Rightarrow & s_i(x) & = & \frac{\sigma_i}{6h_i}(x_{i+1}-x)^3 + \frac{\sigma_{i+1}}{6h_i}(x-x_i)^3 + c_1(x-x_1) + c_2(x_{i+1}-x) \end{array} \tag{12}$$

Because $s_i(x)$ interpolates two endpoints (x_i, f_i) and (x_{i+1}, f_{i+1}) , if we plug x_i, x_{i+1} into the above $s_i(x)$, we have

$$\begin{array}{rcl} f_i & = & s_i(x_i) = \frac{\sigma_i}{6}h_i^2 + c_2 h_i \\ f_{i+1} & = & s_i(x_{i+1}) = \frac{\sigma_{i+1}}{6}h_i^2 + c_1 h_i, \end{array}$$

where we can solve c_1, c_2 as

$$\left\{ \begin{array}{rcl} c_1 & = & (f_{i+1} - \frac{\sigma_{i+1}}{6}h_i^2)/h_i \\ c_2 & = & (f_i - \frac{\sigma_i}{6}h_i^2)/h_i \end{array} \right.$$

By plugging in the value of c_1, c_2 into equation (12), we have ($0 \le i \le M-2$)

$$s_i(x) = \frac{\sigma_i}{6h_i}(x_{i+1}-x)^3 + \frac{\sigma_{i+1}}{6h_i}(x-x_i)^3 + \left(\frac{f_{i+1}}{h_i} - \frac{\sigma_{i+1}h_i}{6}\right)(x-x_i) + \left(\frac{f_i}{h_i} - \frac{\sigma_i h_i}{6}\right)(x_{i+1}-x)$$

with its first derivative as

$$s_i'(x) = -\frac{\sigma_i}{2h_i}(x_{i+1}-x)^2 + \frac{\sigma_{i+1}}{2h_i}(x-x_i)^2 + \frac{f_{i+1}-f_i}{h_i} - \frac{h_i}{6}(\sigma_{i+1}-\sigma_i). \tag{13}$$

Because $s_{i-1}'(x_i) = s_i'(x_i)$, we have ($1 \le i \le M-2$)

$$\frac{1}{6}h_{i-1}\sigma_{i-1} + \frac{1}{3}(h_{i-1}+h_i)\sigma_i + \frac{1}{6}h_i\sigma_{i+1} = \frac{f_{i+1}-f_i}{h_i} - \frac{f_i - f_{i-1}}{h_{i-1}}. \tag{14}$$

Equation (14) gives $M-2$ equations. Recall $\sigma_0 = \sigma_{M-1} = 0$, so totally we get M equations, which is enough to solve M parameters, i.e., $\sigma_0, \sigma_1, \ldots, \sigma_{M-1}$, therefore, we can determine the spline function $s(x)$.

We write out the above system of linear equations, where we hope to identify a fast numerical approach to solve it. The system of linear equations is:

$$\left\{ \begin{array}{cccccc} & \frac{1}{3}(h_0+h_1)\sigma_1 & +\frac{1}{6}h_1\sigma_2 & = & \frac{u_2^n - u_1^n}{h_1} & -\frac{f_1-u_0}{h_0} \\ \frac{1}{6}h_1\sigma_1 & \frac{1}{3}(h_1+h_2)\sigma_2 & +\frac{1}{6}h_2\sigma_3 & = & \frac{f_3-f_2}{h_1} & -\frac{f_2-f_1}{h_0} \\ & & & \vdots & & \\ \frac{1}{6}h_{M-4}\sigma_{M-4} & \frac{1}{3}(h_{M-4}+h_{M-3})\sigma_{M-3} & +\frac{1}{6}h_{M-3}\sigma_{M-2} & = & \frac{f_{M-2}-f_{M-3}}{h_{M-3}} & -\frac{f_{M-3}-f_{M-4}}{h_{M-4}} \\ \frac{1}{6}h_{M-3}\sigma_{M-3} & \frac{1}{3}(h_{M-3}+h_{M-2})\sigma_{M-2} & & = & \frac{f_{M-1}-f_{M-2}}{h_{M-2}} & -\frac{f_{M-2}-f_{M-3}}{h_{M-3}} \end{array} \right.$$

So we can see that the second derivative of cubic spline can be solved by the above system of linear equation. The summarize the discussion above, we have

$$\widehat{\boldsymbol{\sigma}} = \mathbf{M}^{-1}\mathbf{A}\widehat{\mathbf{f}}$$

where vector $\mathbf{f}$ is defined as $\widehat{\mathbf{f}} = (f_0, f_1, \ldots, f_{M-1})^\top$ and matrix $\mathbf{A}, \mathbf{M}$ is defined as

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

$$\mathbf{M} = \begin{pmatrix} \frac{h_0+h_1}{3} & \frac{h_1}{6} & 0 & \dots & 0 & 0 \\ \frac{h_1}{6} & \frac{h_1+h_2}{3} & \frac{h_2}{6} & \dots & 0 & 0 \\ 0 & \frac{h_2}{6} & \frac{h_2+h_3}{3} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \frac{h_{M-4}+h_{M-3}}{3} & \frac{h_{M-3}}{6} \\ 0 & 0 & 0 & \dots & \frac{h_{M-3}}{6} & \frac{h_{M-3}+h_{M-2}}{3} \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} \frac{-3}{h_0^2} & \frac{3}{h_0^2} & 0 & 0 & \dots & 0 & 0 & 0 \\ \frac{-3}{h_0^2} & \frac{3}{h_0^2} - \frac{3}{h_1^2} & \frac{3}{h_1^2} & 0 & \dots & 0 & 0 & 0 \\ 0 & \frac{-3}{h_1^2} & \frac{3}{h_1^2} - \frac{3}{h_2^2} & \frac{3}{h_2^2} & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ & 0 & 0 & 0 & 0 & \frac{-3}{h_{M-3}^2} & \frac{3}{h_{M-3}^2} - \frac{3}{h_{M-2}^2} & \frac{3}{h_{M-2}^2} \\ & 0 & 0 & 0 & 0 & 0 & \frac{-3}{h_{M-2}^2} & \frac{3}{h_{M-2}^2} \end{pmatrix}$$

and matrix $\mathbf{W} = \mathrm{diag}(w_0, w_1, \dots, w_{M-1})$.

$$\mathbf{Q} = \begin{pmatrix} \frac{2}{h_0} & \frac{1}{h_0} & 0 & 0 & \dots & 0 & 0 & 0 \\ \frac{1}{h_0} & \frac{2}{h_0} + \frac{2}{h_1} & \frac{1}{h_1} & 0 & \dots & 0 & 0 & 0 \\ 0 & \frac{1}{h_1} & \frac{2}{h_1} + \frac{2}{h_2} & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \frac{1}{h_{M-3}} & \frac{2}{h_{M-3}} + \frac{2}{h_{M-2}} & \frac{1}{h_{M-2}} \\ 0 & 0 & 0 & 0 & \dots & 0 & \frac{1}{h_{M-2}} & \frac{2}{h_{M-2}} \end{pmatrix},$$

Now, we focus on solving the first derivative of cubic spline $s(x)$. Let $\theta_i = s'(x_i)$ for $i = 0, 1, \dots, M-1$, then we have

$$\begin{aligned} s_i(x) &= \theta_i \tfrac{(x_{i+1}-x)^2(x-x_i)}{h_i^2} - \theta_{i+1} \tfrac{(x-x_i)^2(x_{i+1}-x)}{h_i^2} + f_i \tfrac{(x_{i+1}-x)^2[2(x-x_i)+h_i]}{h_i^3} + f_{i+1} \tfrac{(x-x_i)^2[2(x_{i+1}-x)+h_i]}{h_i^3} \\ s_i'(x) &= \theta_i \tfrac{(x_{i+1}-x)(2x_i+x_{i+1}-3x)}{h_i^2} - \theta_{i+1} \tfrac{(x-x_i)(2x_{i+1}+x_i-3x)}{h_i^2} + 6\tfrac{u_{i+1}^n-u_i^n}{h_i^3}(x_{i+1}-x)(x-x_i) \\ s_i''(x) &= -2\theta_i \tfrac{2x_{i+1}+x_i-3x}{h_i^2} - 2\theta_{i+1} \tfrac{2x_i+x_{i+1}-3x}{h_i^2} + 6\tfrac{u_{i+1}^n-u_i^n}{h_i^3}(x_{i+1}+x_i-2x) \end{aligned}$$

By plugging x_i into $s_i''(x)$ and $s_{i-1}''(x)$, we have

$$\begin{aligned} & s_i''(x) = -2\theta_i \tfrac{2x_{i+1}+x_i-3x}{h_i^2} - 2\theta_{i+1} \tfrac{2x_i+x_{i+1}-3x}{h_i^2} + 6\tfrac{f_{i+1}-f_i}{h_i^3}(x_{i+1}+x_i-2x) \\ \Rightarrow\ & s_i''(x_i) = -2\theta_i \tfrac{2x_{i+1}+x_i-3x_i}{h_i^2} - 2\theta_{i+1} \tfrac{2x_i+x_{i+1}-3x_i}{h_i^2} + 6\tfrac{f_{i+1}-f_i}{h_i^3}(x_{i+1}+x_i-2x_i) \\ & = \tfrac{-4}{h_i}\theta_i + \tfrac{-2}{h_i}\theta_{i+1} + 6\tfrac{f_{i+1}-f_i}{h_i^2} \end{aligned}$$

$$\begin{aligned} & s_{i-1}''(x) = -2\theta_{i-1} \tfrac{2x_i+x_{i-1}-3x}{h_{i-1}^2} - 2\theta_i \tfrac{2x_{i-1}+x_i-3x}{h_{i-1}^2} + 6\tfrac{f_i-f_{i-1}}{h_{i-1}^3}(x_i+x_{i-1}-2x) \\ \Rightarrow\ & s_{i-1}''(x_i) = -2\theta_{i-1} \tfrac{2x_i+x_{i-1}-3x_i}{h_{i-1}^2} - 2\theta_i \tfrac{2x_{i-1}+x_i-3x_i}{h_{i-1}^2} + 6\tfrac{f_i-f_{i-1}}{h_{i-1}^3}(x_i+x_{i-1}-2x_i) \\ & = \tfrac{2}{h_{i-1}}\theta_{i-1} + \tfrac{4}{h_{i-1}}\theta_i - 6\tfrac{f_i-f_{i-1}}{h_{i-1}^2} \end{aligned}$$

Because $s_i''(x_i) = s_{i-1}''(x_i)$, we have $(\forall i = 1, 2, \ldots, M-2)$

$$\begin{array}{rl} & \frac{-4}{h_i}\theta_i + \frac{-2}{h_i}\theta_{i+1} + 6\frac{f_{i+1}-f_i}{h_i^2} = \frac{2}{h_{i-1}}\theta_{i-1} + \frac{4}{h_{i-1}}\theta_i - 6\frac{f_i-f_{i-1}}{h_{i-1}^2} \\ \Leftrightarrow & \frac{2}{h_{i-1}}\theta_{i-1} + (\frac{4}{h_{i-1}} + \frac{4}{h_i})\theta_i + \frac{2}{h_i}\theta_{i+1} = 6\frac{f_{i+1}-f_i}{h_i^2} + 6\frac{f_i-f_{i-1}}{h_{i-1}^2} \\ \Leftrightarrow & \frac{1}{h_{i-1}}\theta_{i-1} + (\frac{2}{h_{i-1}} + \frac{2}{h_i})\theta_i + \frac{1}{h_i}\theta_{i+1} = 3\frac{f_{i+1}-f_i}{h_i^2} + 3\frac{f_i-f_{i-1}}{h_{i-1}^2} \end{array}$$

By organizing the above system of equation into matrix algebra, we have

$$\begin{pmatrix} \frac{1}{h_0} & \frac{2}{h_0}+\frac{2}{h_1} & \frac{1}{h_1} & 0 & \ldots & 0 & 0 & 0 \\ 0 & \frac{1}{h_1} & \frac{2}{h_1}+\frac{2}{h_2} & 0 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & \frac{1}{h_{M-3}} & \frac{2}{h_{M-3}}+\frac{2}{h_{M-2}} & \frac{1}{h_{M-2}} \end{pmatrix} \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_{M-1} \end{pmatrix} = \begin{pmatrix} 3\frac{f_2-f_1}{h_1^2} + 3\frac{f_1-f_0}{h_0^2} \\ 3\frac{f_3-f_2^n}{h_2^2} + 3\frac{f_2-f_1}{h_1^2} \\ \vdots \\ 3\frac{f_{M-1}-f_{M-2}}{h_{M-2}^2} + 3\frac{f_{M-2}^n-f_{M-3}}{h_{M-3}^2} \end{pmatrix}$$

For the two endpoint θ_0, θ_{M-1}, according to $s_0''(x_0) = s_{M-2}''(x_{M-1}) = 0$, we have

$$\begin{array}{rrcl} & s_0''(x) &=& -2\theta_0\frac{2x_1+x_0-3x}{h_0^2} - 2\theta_1\frac{2x_0+x_1-3x}{h_0^2} + 6\frac{f_1-f_0}{h_0^3}(x_1+x_0-2x) \\ \Rightarrow & s_0''(x_0) &=& -2\theta_0\frac{2x_1+x_0-3x_0}{h_0^2} - 2\theta_1\frac{2x_0+x_1-3x_0}{h_0^2} + 6\frac{f_1-f_0}{h_0^3}(x_1+x_0-2x_0) \\ & &=& \frac{-4}{h_0}\theta_0 + \frac{-2}{h_0}\theta_1 + 6\frac{f_1-f_0}{h_0^2} \\ & &=& 0 \end{array}$$

$$\begin{array}{rrcl} & s_{M-2}''(x) &=& -2\theta_{M-2}\frac{2x_{M-1}+x_{M-2}-3x}{h_{M-2}^2} - 2\theta_{M-1}\frac{2x_{M-2}+x_{M-1}-3x}{h_{M-2}^2} + 6\frac{f_{M-1}-f_{M-2}}{h_{M-2}^3}(x_{M-1}+x_{M-2}-2x) \\ \Rightarrow & s_{M-2}''(x_{M-1}) &=& -2\theta_{M-2}\frac{2x_{M-1}+x_{M-2}-3x_{M-1}}{h_{M-2}^2} - 2\theta_{M-1}\frac{2x_{M-2}+x_{M-1}-3x_{M-1}}{h_{M-2}^2} + 6\frac{f_{M-1}-f_{M-2}}{h_{M-2}^3}(x_{M-1}+x_{M-2}-2x_{M-1}) \\ & &=& \frac{2}{h_{M-2}}\theta_{M-2} + \frac{4}{h_{M-2}}\theta_{M-1} - 6\frac{f_{M-1}-f_{M-2}}{h_{M-2}^2} \\ & &=& 0 \end{array}$$

So the first order derivative $\boldsymbol{\theta} = (\theta_0, \theta_1, \ldots, \theta_{M-1})^\top$ can be solved by

$$\underbrace{\begin{pmatrix} \frac{2}{h_0} & \frac{1}{h_0} & 0 & 0 & \ldots & 0 & 0 & 0 \\ \frac{1}{h_0} & \frac{2}{h_0}+\frac{2}{h_1} & \frac{1}{h_1} & 0 & \ldots & 0 & 0 & 0 \\ 0 & \frac{1}{h_1} & \frac{2}{h_1}+\frac{2}{h_2} & 0 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & \frac{1}{h_{M-3}} & \frac{2}{h_{M-3}}+\frac{2}{h_{M-2}} & \frac{1}{h_{M-2}} \\ 0 & 0 & 0 & 0 & \ldots & 0 & \frac{1}{h_{M-2}} & \frac{2}{h_{M-2}} \end{pmatrix}}_{\mathbf{Q}} \underbrace{\begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \vdots \\ \theta_{M-1} \end{pmatrix}}_{\boldsymbol{\theta}} = \underbrace{\begin{pmatrix} 3\frac{f_1-f_0}{h_0^2} \\ 3\frac{f_2-f_1}{h_1^2} + 3\frac{f_1-f_0}{h_0^2} \\ 3\frac{f_3-f_2}{h_2^2} + 3\frac{f_2-f_1}{h_1^2} \\ \vdots \\ 3\frac{f_{M-1}-f_{M-2}}{h_{M-2}^2} + 3\frac{f_{M-2}-f_{M-3}}{h_{M-3}^2} \\ 3\frac{f_{M-1}-f_{M-2}}{h_{M-2}^2} \end{pmatrix}}_{\mathbf{q}}$$

In matrix algebra, the first order derivative $\boldsymbol{\theta} = (\theta_0, \theta_1, \ldots, \theta_{M-1})^\top$ can be solved by

$$\widehat{\boldsymbol{\theta}} = \mathbf{Q}^{-1}\mathbf{B}\widehat{\mathbf{f}}.$$

□

C Review of Local Polynomial Regression

We apply the local polynomial regression method to obtain all the derivatives in terms of sample data $\mathcal{D}$. Recall that the derivatives needed to estimate $\partial_t^1 u(x_i,t_n), \partial_x^1 u(x_i,t_n), \partial_x^2 u(x_i,t_n), \ldots, \partial_x^p u(x_i,t_n)$, and here we first take the derivation $\widehat{\partial_t^1 u(x_i,t)}$ as an example. To derive it, we fix the spatial variable x_i and locally fit a degree $\check{p}$ polynomial over the data $\{(x_i,t_n,u_i^n)\}_{n=0,\ldots,N-1}$ to obtain $\widehat{\partial_t^1 u(x_i,t)}$, that is,

$$\begin{cases} u(x_i,t_0) &= u(x_i,t)+\partial_t^1 u(x_i,t)(t_0-t)+\ldots+\partial_t^{\check{p}} u(x_i,t)(t_0-t)^{\check{p}} \\ u(x_i,t_1) &= u(x_i,t)+\partial_t^1 u(x_i,t)(t_1-t)+\ldots+\partial_t^{\check{p}} u(x_i,t)(t_1-t)^{\check{p}} \\ \vdots & \vdots \\ u(x_i,t_{N-1}) &= u(x_i,t)+\partial_t^1 u(x_i,t)(t_{N-1}-t)+\ldots+\partial_t^{\check{p}} u(x_i,t)(t_{N-1}-t)^{\check{p}} \end{cases}$$

And we denote $\mathbf{b}(t)=(u(x_i,t),\partial_t^1 u(x_i,t),\ldots,\partial_t^{\check{p}} u(x_i,t))'$. Then $\partial_t^1 u(x_i,t)$ can be obtained as the second entry of the vector $\mathbf{b}(t)$, and $\mathbf{b}(t)$ is obtained by the following optimization problem:

$$\widehat{\mathbf{b}(t)} = \arg\min_{\mathbf{b}(t)} \sum_{n=0}^{N-1} \left[u_i^n - \sum_{j=0}^{\check{p}} \partial_t^j u(x_i,t)(t_n-t)^j \right]^2 \mathcal{K}\left(\frac{t_n-t}{h}\right) \tag{15}$$

Essentially, the optimization problem in equation (15) is a weighted least square model, where $\mathbf{b}(t)$ can be solved in a close form:

$$\mathbf{b}(t) = \left(\mathbf{X}_{\text{tep}}'\mathbf{W}_{\text{tep}}\mathbf{X}_{\text{tep}}\right)^{-1}\mathbf{X}_{\text{tep}}'\mathbf{W}_{\text{tep}}\mathbf{u}(x_i,:), \tag{16}$$

where

$$\mathbf{X}_{\text{tep}} = \begin{bmatrix} 1 & (t_0-t) & \ldots & (t_0-t)^{\check{p}} \\ 1 & (t_1-t) & \ldots & (t_1-t)^{\check{p}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & (t_{N-1}-t) & \ldots & (t_0-t)^{\check{p}} \end{bmatrix}, \mathbf{u}(x_i,:) = \begin{bmatrix} u_i^0 \\ u_i^1 \\ \vdots \\ u_i^{N-1} \end{bmatrix}$$

and $\mathbf{W}_{\text{tep}} = \text{diag}\left(\mathcal{K}\left(\frac{t_0-t}{h}\right),\ldots,\mathcal{K}\left(\frac{t_{N-1}-t}{h}\right)\right)$.

D Proof of the computation complexity in cubic spline (without interaction)

Proof. It can easily seen that the computational complexity of $\widehat{\boldsymbol{\beta}}$ with interaction terms and without interaction terms are the same, so we take the scenario without interaction terms as an example in this proof.

We know that there is a close-form solution of the ridge regression problem:

$$\begin{aligned} & \widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} (\nabla_t \mathbf{u} - \mathbf{X}\boldsymbol{\beta})^2 + \lambda\boldsymbol{\beta}^2 \\ \Rightarrow \quad & \widehat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}'\nabla_t\mathbf{u} \end{aligned}$$

$$\mathbf{X} = \begin{pmatrix} 1 & u_0^0 & \partial_x u_0^0 & \partial_x^2 u_0^0 \\ 1 & u_1^0 & \partial_x u_1^0 & \partial_x^2 u_1^0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & u_{M-1}^0 & \partial_x u_{M-1}^0 & \partial_x^2 u_{M-1}^0 \\ 1 & u_0^1 & \partial_x u_0^1 & \partial_x^2 u_0^1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & u_{M-1}^{N-1} & \partial_x u_{M-1}^{N-1} & \partial_x^2 u_{M-1}^{N-1} \end{pmatrix} = \begin{pmatrix} \mathbf{1} & \mathbf{u}_\cdot^0 & \mathbf{Q}^{-1}\mathbf{q}_0 & \mathbf{M}^{-1}\mathbf{y}_0 \\ \mathbf{1} & \mathbf{u}_\cdot^1 & \mathbf{Q}^{-1}\mathbf{q}_1 & \mathbf{M}^{-1}\mathbf{y}_1 \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{1} & \mathbf{u}_\cdot^{N-1} & \mathbf{Q}^{-1}\mathbf{q}_{N-1} & \mathbf{M}^{-1}\mathbf{y}_{N-1} \end{pmatrix}$$

where

$$\mathbf{M} = \begin{pmatrix} \frac{h_0+h_1}{3} & \frac{h_1}{6} & 0 & \dots & 0 & 0 \\ \frac{h_1}{6} & \frac{h_1+h_2}{3} & \frac{h_2}{6} & \dots & 0 & 0 \\ 0 & \frac{h_2}{6} & \frac{h_2+h_3}{3} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \frac{h_{M-4}+h_{M-3}}{3} & \frac{h_{M-3}}{6} \\ 0 & 0 & 0 & \dots & \frac{h_{M-3}}{6} & \frac{h_{M-3}+h_{M-2}}{3} \end{pmatrix}.$$

Following Cholesky decomposition, matrix $\mathbf{M}$ can be decomposed as

$$\mathbf{M} = \mathbf{L}\mathbf{D}\mathbf{L}^\top,$$

where $\mathbf{L}, \mathbf{D}$ has the form of

$$\mathbf{L} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ l_1 & 1 & 0 & \dots & 0 \\ 0 & l_2 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & l_{M-3} & 1 \end{pmatrix}, \mathbf{D} = \begin{pmatrix} d_1 & 0 & \dots & 0 \\ 0 & d_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & d_{M-2} \end{pmatrix},$$

where matrix $\mathbf{L}, \mathbf{D}$ can be solved by Algorithm 3.

Algorithm 3: Pseudo code to solve $\mathbf{L}, \mathbf{D}$

Input: matrix $\mathbf{M}$
Output: matrix $\mathbf{L}, \mathbf{D}$
Initialize $d_1 = \mathbf{M}_{1,1}$
for $i = 1, 2, \dots, M-3$ **do**
    $l_i = \mathbf{M}_{i,i+1}/d_i$
    $d_{i+1} = \mathbf{M}_{i+1,i+1} - d_i a_i^2$

It can be seen from Algorithm 3 that, the computational complexity of solve $\mathbf{L}, \mathbf{D}$ is $O(M)$.

Now, we prove that, with known $\mathbf{L}, \mathbf{D}, \mathbf{M}$, the computational complexity of solving $\boldsymbol{\sigma} = \mathbf{M}^{-1}\mathbf{y} = (\mathbf{L}^\top)^{-1}\mathbf{D}^{-1}\mathbf{L}^{-1}\mathbf{y}$ is still $O(M)$. The logic flow to prove it is that, we begin to prove the computational complexity of $\bar{\mathbf{y}} = \mathbf{L}^{-1}\mathbf{y}$ is $O(M)$. Then we prove that the

computational complexity of $\bar{\bar{\mathbf{y}}} = \mathbf{D}\bar{\mathbf{y}}$ is $O(M)$. Finally, we prove that the computational complexity of $\bar{\bar{\bar{\mathbf{y}}}} = (\mathbf{L}^\top)^{-1}\bar{\bar{\mathbf{y}}}$ is $O(M)$.

To begin with, we prove the computational complexity of $\bar{\mathbf{y}} = \mathbf{L}^{-1}\mathbf{y}$ is $O(M)$. Since $\mathbf{L}\bar{\mathbf{y}} = \mathbf{y}$, we have

$$\begin{cases} \mathbf{y}_1 = \bar{\mathbf{y}}_1 \\ \mathbf{y}_2 = \bar{\mathbf{y}}_2 + l_1\bar{\mathbf{y}}_1 \\ \vdots \\ \mathbf{y}_{M-2} = \bar{\mathbf{y}}_{M-2} + l_{M-3}\bar{\mathbf{y}}_{M-3} \end{cases}$$

where $\mathbf{y}_i, \bar{\mathbf{y}}_i$ is the i-th entry in $\mathbf{y}, \bar{\mathbf{y}}$. Thus, we can solve $\bar{\mathbf{y}}$ through Algorithm 4. From Algorithm 4, we know that the computational complexity of solving $\mathbf{L}^{-1}\mathbf{y}$ is $O(M)$.

Algorithm 4: Pseudo code to solve $\mathbf{L}^{-1}\mathbf{y}$

Input: matrix $\mathbf{L}, \mathbf{y}$
Output: matrix $\bar{\mathbf{y}}$
Initialize $\bar{\mathbf{y}}_1 = \mathbf{y}_1$
for $i = 2, \ldots, M-2$ **do**
  $\bar{\mathbf{y}}_i = \mathbf{y}_i - l_{i-1}\bar{\mathbf{y}}_{i-1}$

Next, we prove that the computational complexity of $\bar{\bar{\mathbf{y}}} = \mathbf{D}\bar{\mathbf{y}}$ is $O(M)$, which is very obvious.

Finally, we prove that the computational complexity of $\bar{\bar{\bar{\mathbf{y}}}} = (\mathbf{L}^\top)^{-1}\bar{\bar{\mathbf{y}}}$ is $O(M)$. Following the same logical flow in Algorithm 4, it can also be easily proved.

□

E Proof of the computation complexity in cubic spline (with interaction)

Theorem E.1. The computation complexity of cubic spline to solve $\widehat{\boldsymbol{\beta}}$ in

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} (\nabla_t \mathbf{u} - \mathbf{X}\boldsymbol{\beta})^2 + \lambda\boldsymbol{\beta}^2$$

is $O(MN)$, where

$$\mathbf{X} = \begin{pmatrix} 1 & x_{12} & x_{13} & x_{14} & \ldots & x_{1,10} \\ 1 & x_{22} & x_{23} & x_{24} & \ldots & x_{2,10} \\ \vdots & \vdots & \vdots & \vdots & \ldots & \vdots \\ 1 & x_{M-1,2} & x_{M-1,3} & x_{M-1,4} & \ldots & x_{M-1,10} \\ 1 & x_{M,2} & x_{M,3} & x_{M,4} & \ldots & x_{M,10} \\ \vdots & \vdots & \vdots & \vdots & \ldots & \vdots \\ 1 & x_{MN,2} & x_{MN,3} & x_{MN,4} & \ldots & x_{MN,10} \end{pmatrix}, \nabla_t \mathbf{u} = \begin{pmatrix} \partial_t u_0^0 \\ \partial_t u_1^0 \\ \vdots \\ \partial_t u_{M-1}^0 \\ \partial_t u_0^1 \\ \vdots \\ \partial_t u_{M-1}^{N-1} \end{pmatrix}.$$

with $x_{ij}{}_{i,j=1,\ldots,10}$ defined by

Proof.

$$\begin{aligned} & \widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} (\nabla_t \mathbf{u} - \mathbf{X}\boldsymbol{\beta})^2 + \lambda \boldsymbol{\beta}^2 \\ \Rightarrow \quad & \widehat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}' \nabla_t \mathbf{u} \end{aligned}$$

$$\begin{aligned} \mathbf{X} &= \begin{pmatrix} 1 & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} & x_{17} & x_{18} & x_{19} & x_{1,10} \\ 1 & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} & x_{27} & x_{28} & x_{29} & x_{2,10} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{M-1,2} & x_{M-1,3} & x_{M-1,4} & x_{M-1,5} & x_{M-1,6} & x_{M-1,7} & x_{M-1,8} & x_{M-1,9} & x_{M-1,10} \\ 1 & x_{M,2} & x_{M,3} & x_{M,4} & x_{M,5} & x_{M,6} & x_{M,7} & x_{M,8} & x_{M,9} & x_{M,10} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{MN,2} & x_{MN,3} & x_{MN,4} & x_{MN,5} & x_{MN,6} & x_{MN,7} & x_{MN,8} & x_{MN,9} & x_{MN,10} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 & \mathbf{x}_6 & \mathbf{x}_7 & \mathbf{x}_8 & \mathbf{x}_9 & \mathbf{x}_{10} \end{pmatrix} \end{aligned}$$

$$x_{11} = x_{21} = \ldots = x_{MN,1} = 1$$

$$\begin{cases} x_{12} = u_0^0 \\ x_{13} = \partial_x u_0^0 \\ x_{14} = \partial_x^2 u_0^0 \\ x_{15} = (u_0^0)^2 \\ x_{16} = (\partial_x u_0^0)^2 \\ x_{17} = (\partial_x^2 u_0^0)^2 \\ x_{18} = u_0^0 \partial_x u_0^0 \\ x_{19} = u_0^0 \partial_x^2 u_0^0 \\ x_{1,10} = \partial_x u_0^0 \partial_x^2 u_0^0 \end{cases} \quad \begin{cases} x_{22} = u_1^0 \\ x_{23} = \partial_x u_1^0 \\ x_{24} = \partial_x^2 u_1^0 \\ x_{25} = (u_1^0)^2 \\ x_{26} = (\partial_x u_1^0)^2 \\ x_{27} = (\partial_x^2 u_1^0)^2 \\ x_{28} = u_1^0 \partial_x u_1^0 \\ x_{29} = u_1^0 \partial_x^2 u_1^0 \\ x_{2,10} = \partial_x u_1^0 \partial_x^2 u_1^0 \end{cases}$$

$$\begin{cases} x_{M-1,2} = u_{M-1}^0 \\ x_{M-1,3} = \partial_x u_{M-1}^0 \\ x_{M-1,4} = \partial_x^2 u_{M-1}^0 \\ x_{M-1,5} = (u_{M-1}^0)^2 \\ x_{M-1,6} = (\partial_x u_{M-1}^0)^2 \\ x_{M-1,7} = (\partial_x^2 u_{M-1}^0)^2 \\ x_{M-1,8} = u_{M-1}^0 \partial_x u_{M-1}^0 \\ x_{M-1,9} = u_{M-1}^0 \partial_x^2 u_{M-1}^0 \\ x_{M-1,10} = \partial_x u_{M-1}^0 \partial_x^2 u_{M-1}^0 \end{cases} \quad \begin{cases} x_{M2} = u_0^1 \\ x_{M3} = \partial_x u_0^1 \\ x_{M4} = \partial_x^2 u_0^1 \\ x_{M5} = (u_0^1)^2 \\ x_{M6} = (\partial_x u_0^1)^2 \\ x_{M7} = (\partial_x^2 u_0^1)^2 \\ x_{M8} = u_0^1 \partial_x u_0^1 \\ x_{M9} = u_0^1 \partial_x^2 u_0^1 \\ x_{M,10} = \partial_x u_0^1 \partial_x^2 u_0^1 \end{cases} \quad \begin{cases} x_{MN,2} = u_{M-1}^{N-1} \\ x_{MN,3} = \partial_x u_{M-1}^{N-1} \\ x_{MN,4} = \partial_x^2 u_{M-1}^{N-1} \\ x_{MN,5} = (u_{M-1}^{N-1})^2 \\ x_{MN,6} = (\partial_x u_{M-1}^{N-1})^2 \\ x_{MN,7} = (\partial_x^2 u_{M-1}^{N-1})^2 \\ x_{MN,8} = u_{M-1}^{N-1} \partial_x u_{M-1}^{N-1} \\ x_{MN,9} = u_{M-1}^{N-1} \partial_x^2 u_{M-1}^{N-1} \\ x_{MN,10} = \partial_x u_{M-1}^{N-1} \partial_x^2 u_{M-1}^{N-1} \end{cases}$$

$$\mathbf{x}_1 = \mathbf{1}, \mathbf{x}_2 = \begin{pmatrix} \mathbf{u}_{\cdot}^0 \\ \mathbf{u}_{\cdot}^1 \\ \vdots \\ \mathbf{u}_{\cdot}^{N-1} \end{pmatrix}, \mathbf{x}_3 = \begin{pmatrix} \mathbf{Q}^{-1}\mathbf{q}_0 \\ \mathbf{Q}^{-1}\mathbf{q}_1 \\ \vdots \\ \mathbf{Q}^{-1}\mathbf{q}_{N-1} \end{pmatrix}, \mathbf{x}_4 = \begin{pmatrix} \mathbf{M}^{-1}\mathbf{y}_0 \\ \mathbf{M}^{-1}\mathbf{y}_1 \\ \vdots \\ \mathbf{M}^{-1}\mathbf{y}_{N-1} \end{pmatrix}, \mathbf{x}_5 = \begin{pmatrix} \mathbf{u}_{\cdot}^0 \odot \mathbf{u}_{\cdot}^0 \\ \mathbf{u}_{\cdot}^1 \odot \mathbf{u}_{\cdot}^1 \\ \vdots \\ \mathbf{u}_{\cdot}^{N-1} \odot \mathbf{u}_{\cdot}^{N-1} \end{pmatrix}$$

$$\mathbf{x}_6 = \begin{pmatrix} \mathbf{Q}^{-1}\mathbf{q}_0 \odot \mathbf{Q}^{-1}\mathbf{q}_0 \\ \mathbf{Q}^{-1}\mathbf{q}_1 \odot \mathbf{Q}^{-1}\mathbf{q}_1 \\ \vdots \\ \mathbf{Q}^{-1}\mathbf{q}_{N-1} \odot \mathbf{Q}^{-1}\mathbf{q}_{N-1} \end{pmatrix}, \mathbf{x}_7 = \begin{pmatrix} \mathbf{M}^{-1}\mathbf{y}_0 \odot \mathbf{M}^{-1}\mathbf{y}_0 \\ \mathbf{M}^{-1}\mathbf{y}_1 \odot \mathbf{M}^{-1}\mathbf{y}_1 \\ \vdots \\ \mathbf{M}^{-1}\mathbf{y}_{N-1} \odot \mathbf{M}^{-1}\mathbf{y}_{N-1} \end{pmatrix}, \mathbf{x}_8 = \begin{pmatrix} \mathbf{u}_{\cdot}^0 \odot \mathbf{Q}^{-1}\mathbf{q}_0 \\ \mathbf{u}_{\cdot}^1 \odot \mathbf{Q}^{-1}\mathbf{q}_1 \\ \vdots \\ \mathbf{u}_{\cdot}^{N-1} \odot \mathbf{Q}^{-1}\mathbf{q}_{N-1} \end{pmatrix},$$

$$\mathbf{x}_9 = \begin{pmatrix} \mathbf{u}_{:}^{0} \odot \mathbf{M}^{-1}\mathbf{y}_0 \\ \mathbf{u}_{:}^{1} \odot \mathbf{M}^{-1}\mathbf{y}_1 \\ \vdots \\ \mathbf{u}_{:}^{N-1} \odot \mathbf{M}^{-1}\mathbf{y}_{N-1} \end{pmatrix}, \mathbf{x}_{10} = \begin{pmatrix} \mathbf{Q}^{-1}\mathbf{q}_0 \odot \mathbf{M}^{-1}\mathbf{y}_0 \\ \mathbf{Q}^{-1}\mathbf{q}_1 \odot \mathbf{M}^{-1}\mathbf{y}_1 \\ \vdots \\ \mathbf{Q}^{-1}\mathbf{q}_{N-1} \odot \mathbf{M}^{-1}\mathbf{y}_{N-1} \end{pmatrix}$$

$$\begin{aligned} \mathbf{X}^\top\mathbf{X} &= \begin{pmatrix} \widetilde{x}_{11} & \widetilde{x}_{12} & \widetilde{x}_{13} & \widetilde{x}_{14} & \widetilde{x}_{15} & \widetilde{x}_{16} & \widetilde{x}_{17} & \widetilde{x}_{18} & \widetilde{x}_{19} & \widetilde{x}_{1,10} \\ \widetilde{x}_{21} & \widetilde{x}_{22} & \widetilde{x}_{23} & \widetilde{x}_{24} & \widetilde{x}_{25} & \widetilde{x}_{26} & \widetilde{x}_{27} & \widetilde{x}_{28} & \widetilde{x}_{29} & \widetilde{x}_{2,10} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \widetilde{x}_{10,1} & \widetilde{x}_{10,2} & \widetilde{x}_{10,3} & \widetilde{x}_{10,4} & \widetilde{x}_{10,5} & \widetilde{x}_{10,6} & \widetilde{x}_{10,7} & \widetilde{x}_{10,8} & \widetilde{x}_{10,9} & \widetilde{x}_{10,10} \end{pmatrix} \\ &= \begin{pmatrix} \widetilde{\mathbf{x}}_1 & \widetilde{\mathbf{x}}_2 & \widetilde{\mathbf{x}}_3 & \widetilde{\mathbf{x}}_4 & \widetilde{\mathbf{x}}_5 & \widetilde{\mathbf{x}}_6 & \widetilde{\mathbf{x}}_7 & \widetilde{\mathbf{x}}_8 & \widetilde{\mathbf{x}}_9 & \widetilde{\mathbf{x}}_{10} \end{pmatrix} \end{aligned}$$

$$\widetilde{\mathbf{x}}_1 = \begin{pmatrix} MN \\ \sum_i \sum_n u_i^n \\ \mathbf{1}^\top\mathbf{Q}^{-1} \sum_{n=0}^{N-1} \mathbf{q}_n \\ \mathbf{1}^\top\mathbf{M}^{-1} \sum_{n=0}^{N-1} \mathbf{y}_n \\ \sum_{n=0}^{N-1} (\mathbf{u}_{:}^n \odot \mathbf{u}_{:}^n)^\top\mathbf{1} \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{Q}^{-1}\mathbf{q}_n)^\top\mathbf{1} \\ \sum_{n=0}^{N-1} (\mathbf{M}^{-1}\mathbf{y}_n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top\mathbf{1} \\ \sum_{n=0}^{N-1} (\mathbf{u}_{:}^n \odot \mathbf{Q}^{-1}\mathbf{q}_n)^\top\mathbf{1} \\ \sum_{n=0}^{N-1} (\mathbf{u}_{:}^n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top\mathbf{1} \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top\mathbf{1} \end{pmatrix}, \widetilde{\mathbf{x}}_2 = \begin{pmatrix} \sum_i \sum_n u_i^n \\ \sum_i \sum_n (u_i^n)^2 \\ \sum_{n=0}^{N-1} \mathbf{u}_{:}^{n\top}\mathbf{Q}^{-1}\mathbf{q}_n \\ \sum_{n=0}^{N-1} (\mathbf{M}^{-1}\mathbf{y}_n)^\top\mathbf{u}_{:}^n \\ \sum_{n=0}^{N-1} (\mathbf{u}_{:}^n \odot \mathbf{u}_{:}^n)^\top\mathbf{u}_{:}^n \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{Q}^{-1}\mathbf{q}_n)^\top\mathbf{u}_{:}^n \\ \sum_{n=0}^{N-1} (\mathbf{M}^{-1}\mathbf{y}_n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top\mathbf{u}_{:}^n \\ \sum_{n=0}^{N-1} (\mathbf{u}_{:}^n \odot \mathbf{Q}^{-1}\mathbf{q}_n)^\top\mathbf{u}_{:}^n \\ \sum_{n=0}^{N-1} (\mathbf{u}_{:}^n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top\mathbf{u}_{:}^n \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top\mathbf{u}_{:}^n \end{pmatrix}$$

$$\widetilde{\mathbf{x}}_3 = \begin{pmatrix} \mathbf{1}^\top \mathbf{Q}^{-1} \sum_{n=0}^{N-1} \mathbf{q}_n \\ \sum_{n=0}^{N-1} \mathbf{u}_:^{n\top} \mathbf{Q}^{-1} \mathbf{q}_n \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n)^\top \mathbf{Q}^{-1} \mathbf{q}_n \\ \sum_{n=0}^{N-1} (\mathbf{M}^{-1}\mathbf{y}_n)^\top \mathbf{Q}^{-1} \mathbf{q}_n \\ \sum_{n=0}^{N-1} (\mathbf{u}_:^n \odot \mathbf{u}_:^n)^\top \mathbf{Q}^{-1} \mathbf{q}_n \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{Q}^{-1}\mathbf{q}_n)^\top \mathbf{Q}^{-1} \mathbf{q}_n \\ \sum_{n=0}^{N-1} (\mathbf{M}^{-1}\mathbf{y}_n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top \mathbf{Q}^{-1} \mathbf{q}_n \\ \sum_{n=0}^{N-1} (\mathbf{u}_:^n \odot \mathbf{Q}^{-1}\mathbf{q}_n)^\top \mathbf{Q}^{-1} \mathbf{q}_n \\ \sum_{n=0}^{N-1} (\mathbf{u}_:^n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top \mathbf{Q}^{-1} \mathbf{q}_n \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top \mathbf{M}^{-1} \mathbf{y}_n \end{pmatrix}, \widetilde{\mathbf{x}}_4 = \begin{pmatrix} \mathbf{1}^\top \mathbf{M}^{-1} \sum_{n=0}^{N-1} \mathbf{y}_n \\ \sum_{n=0}^{N-1} \mathbf{u}_:^{n\top} \mathbf{M}^{-1} \mathbf{y}_n \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n)^\top \mathbf{M}^{-1} \mathbf{y}_n \\ \sum_{n=0}^{N-1} (\mathbf{M}^{-1}\mathbf{y}_n)^\top \mathbf{M}^{-1} \mathbf{y}_n \\ \sum_{n=0}^{N-1} (\mathbf{u}_:^n \odot \mathbf{u}_:^n)^\top \mathbf{M}^{-1} \mathbf{y}_n \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{Q}^{-1}\mathbf{q}_n)^\top \mathbf{M}^{-1} \mathbf{y}_n \\ \sum_{n=0}^{N-1} (\mathbf{M}^{-1}\mathbf{y}_n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top \mathbf{M}^{-1} \mathbf{y}_n \\ \sum_{n=0}^{N-1} (\mathbf{u}_:^n \odot \mathbf{Q}^{-1}\mathbf{q}_n)^\top \mathbf{M}^{-1} \mathbf{y}_n \\ \sum_{n=0}^{N-1} (\mathbf{u}_:^n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top \mathbf{M}^{-1} \mathbf{y}_n \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top \mathbf{Q}^{-1} \mathbf{q}_n \end{pmatrix}$$

$$\widetilde{\mathbf{x}}_5 = \begin{pmatrix} \sum_i \sum_n (u_i^n)^2 \\ \sum_{n=0}^{N-1} \mathbf{u}_:^{n\top} (\mathbf{u}_:^n \odot \mathbf{u}_:^n) \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n)^\top (\mathbf{u}_:^n \odot \mathbf{u}_:^n) \\ \sum_{n=0}^{N-1} (\mathbf{M}^{-1}\mathbf{y}_n)^\top (\mathbf{u}_:^n \odot \mathbf{u}_:^n) \\ \sum_{n=0}^{N-1} (\mathbf{u}_:^n \odot \mathbf{u}_:^n)^\top (\mathbf{u}_:^n \odot \mathbf{u}_:^n) \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{Q}^{-1}\mathbf{q}_n)^\top (\mathbf{u}_:^n \odot \mathbf{u}_:^n) \\ \sum_{n=0}^{N-1} (\mathbf{M}^{-1}\mathbf{y}_n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top (\mathbf{u}_:^n \odot \mathbf{u}_:^n) \\ \sum_{n=0}^{N-1} (\mathbf{u}_:^n \odot \mathbf{Q}^{-1}\mathbf{q}_n)^\top (\mathbf{u}_:^n \odot \mathbf{u}_:^n) \\ \sum_{n=0}^{N-1} (\mathbf{u}_:^n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top (\mathbf{u}_:^n \odot \mathbf{u}_:^n) \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top (\mathbf{u}_:^n \odot \mathbf{u}_:^n) \end{pmatrix}, \widetilde{\mathbf{x}}_6 = \begin{pmatrix} \sum_{n=0}^{N-1} \mathbf{1}^\top (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{Q}^{-1}\mathbf{q}_n) \\ \sum_{n=0}^{N-1} \mathbf{u}_:^{n\top} (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{Q}^{-1}\mathbf{q}_n) \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n)^\top (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{Q}^{-1}\mathbf{q}_n) \\ \sum_{n=0}^{N-1} (\mathbf{M}^{-1}\mathbf{y}_n)^\top (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{Q}^{-1}\mathbf{q}_n) \\ \sum_{n=0}^{N-1} (\mathbf{u}_:^n \odot \mathbf{u}_:^n)^\top (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{Q}^{-1}\mathbf{q}_n) \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{Q}^{-1}\mathbf{q}_n)^\top (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{Q}^{-1}\mathbf{q}_n) \\ \sum_{n=0}^{N-1} (\mathbf{M}^{-1}\mathbf{y}_n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{Q}^{-1}\mathbf{q}_n) \\ \sum_{n=0}^{N-1} (\mathbf{u}_:^n \odot \mathbf{Q}^{-1}\mathbf{q}_n)^\top (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{Q}^{-1}\mathbf{q}_n) \\ \sum_{n=0}^{N-1} (\mathbf{u}_:^n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{Q}^{-1}\mathbf{q}_n) \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{Q}^{-1}\mathbf{q}_n) \end{pmatrix}$$

$$\widetilde{\mathbf{x}}_7 = \begin{pmatrix} \sum_{n=0}^{N-1} \mathbf{1}^\top (\mathbf{M}^{-1}\mathbf{y}_n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} \mathbf{u}_:^{n\top} (\mathbf{M}^{-1}\mathbf{y}_n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n)^\top (\mathbf{M}^{-1}\mathbf{y}_n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} (\mathbf{M}^{-1}\mathbf{y}_n)^\top (\mathbf{M}^{-1}\mathbf{y}_n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} (\mathbf{u}_:^n \odot \mathbf{u}_:^n)^\top (\mathbf{M}^{-1}\mathbf{y}_n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{Q}^{-1}\mathbf{q}_n)^\top (\mathbf{M}^{-1}\mathbf{y}_n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} (\mathbf{M}^{-1}\mathbf{y}_n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top (\mathbf{M}^{-1}\mathbf{y}_n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} (\mathbf{u}_:^n \odot \mathbf{Q}^{-1}\mathbf{q}_n)^\top (\mathbf{M}^{-1}\mathbf{y}_n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} (\mathbf{u}_:^n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top (\mathbf{M}^{-1}\mathbf{y}_n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top (\mathbf{M}^{-1}\mathbf{y}_n \odot \mathbf{M}^{-1}\mathbf{y}_n) \end{pmatrix}, \widetilde{\mathbf{x}}_8 \begin{pmatrix} \sum_{n=0}^{N-1} \mathbf{1}^\top (\mathbf{u}_:^n \odot \mathbf{Q}^{-1}\mathbf{q}_n) \\ \sum_{n=0}^{N-1} \mathbf{u}_:^{n\top} (\mathbf{u}_:^n \odot \mathbf{Q}^{-1}\mathbf{q}_n) \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n)^\top (\mathbf{u}_:^n \odot \mathbf{Q}^{-1}\mathbf{q}_n) \\ \sum_{n=0}^{N-1} (\mathbf{M}^{-1}\mathbf{y}_n)^\top (\mathbf{u}_:^n \odot \mathbf{Q}^{-1}\mathbf{q}_n) \\ \sum_{n=0}^{N-1} (\mathbf{u}_:^n \odot \mathbf{u}_:^n)^\top (\mathbf{u}_:^n \odot \mathbf{Q}^{-1}\mathbf{q}_n) \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{Q}^{-1}\mathbf{q}_n)^\top (\mathbf{u}_:^n \odot \mathbf{Q}^{-1}\mathbf{q}_n) \\ \sum_{n=0}^{N-1} (\mathbf{M}^{-1}\mathbf{y}_n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top (\mathbf{u}_:^n \odot \mathbf{Q}^{-1}\mathbf{q}_n) \\ \sum_{n=0}^{N-1} (\mathbf{u}_:^n \odot \mathbf{Q}^{-1}\mathbf{q}_n)^\top (\mathbf{u}_:^n \odot \mathbf{Q}^{-1}\mathbf{q}_n) \\ \sum_{n=0}^{N-1} (\mathbf{u}_:^n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top (\mathbf{u}_:^n \odot \mathbf{Q}^{-1}\mathbf{q}_n) \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top (\mathbf{u}_:^n \odot \mathbf{Q}^{-1}\mathbf{q}_n) \end{pmatrix}$$

$$\widetilde{\mathbf{x}}_9 \begin{pmatrix} \sum_{n=0}^{N-1} \mathbf{1}^\top (\mathbf{u}_:^n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} \mathbf{u}_:^{n\top} (\mathbf{u}_:^n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n)^\top (\mathbf{u}_:^n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} (\mathbf{M}^{-1}\mathbf{y}_n)^\top (\mathbf{u}_:^n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} (\mathbf{u}_:^n \odot \mathbf{u}_:^n)^\top (\mathbf{u}_:^n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{Q}^{-1}\mathbf{q}_n)^\top (\mathbf{u}_:^n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} (\mathbf{M}^{-1}\mathbf{y}_n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top (\mathbf{u}_:^n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} (\mathbf{u}_:^n \odot \mathbf{Q}^{-1}\mathbf{q}_n)^\top (\mathbf{u}_:^n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} (\mathbf{u}_:^n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top (\mathbf{u}_:^n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} (\mathbf{u}_:^n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{M}^{-1}\mathbf{y}_n) \end{pmatrix}, \widetilde{\mathbf{x}}_{10} = \begin{pmatrix} \sum_{n=0}^{N-1} \mathbf{1}^\top (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} \mathbf{u}_:^{n\top} (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n)^\top (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} (\mathbf{M}^{-1}\mathbf{y}_n)^\top (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} (\mathbf{u}_:^n \odot \mathbf{u}_:^n)^\top (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{Q}^{-1}\mathbf{q}_n)^\top (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} (\mathbf{M}^{-1}\mathbf{y}_n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} (\mathbf{u}_:^n \odot \mathbf{Q}^{-1}\mathbf{q}_n)^\top (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top (\mathbf{u}_:^n \odot \mathbf{M}^{-1}\mathbf{y}_n) \\ \sum_{n=0}^{N-1} (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{M}^{-1}\mathbf{y}_n)^\top (\mathbf{Q}^{-1}\mathbf{q}_n \odot \mathbf{M}^{-1}\mathbf{y}_n) \end{pmatrix}$$

- Computation complexity of $\sum_{n=1}^{N-1}\mathbf{1}^\top(\mathbf{Q}^{-1}\mathbf{q}_n\odot\mathbf{Q}^{-1}\mathbf{q}_n)$:
  - Computation complexity of $\mathbf{Q}^{-1}$: $O(M)$;
  - Computation complexity of $\mathbf{Q}^{-1}\mathbf{q}_n$: $O(M)$;
  - Computation complexity of $\mathbf{Q}^{-1}\mathbf{q}_n\odot\mathbf{Q}^{-1}\mathbf{q}_n$: $O(M)$
  - Computation complexity of $\mathbf{1}^\top(\mathbf{Q}^{-1}\mathbf{q}_n\odot\mathbf{Q}^{-1}\mathbf{q}_n)$: $O(M)$
  - Computation complexity of $\sum_{n=1}^{N-1}\mathbf{1}^\top(\mathbf{Q}^{-1}\mathbf{q}_n\odot\mathbf{Q}^{-1}\mathbf{q}_n)$: $O(MN)$
- Computation complexity of $\sum_{n=1}^{N-1}(\mathbf{Q}^{-1}\mathbf{q}_n\odot\mathbf{M}^{-1}\mathbf{y}_n)^\top(\mathbf{Q}^{-1}\mathbf{q}_n\odot\mathbf{M}^{-1}\mathbf{y}_n)$:
  - Computation complexity of $\mathbf{Q}^{-1}$: $O(M)$;
  - Computation complexity of $\mathbf{Q}^{-1}\mathbf{q}_n$: $O(M)$;
  - Computation complexity of $\mathbf{Q}^{-1}\mathbf{q}_n\odot\mathbf{M}^{-1}\mathbf{y}_n$: $O(M)$
  - Computation complexity of $(\mathbf{Q}^{-1}\mathbf{q}_n\odot\mathbf{M}^{-1}\mathbf{y}_n)^\top(\mathbf{Q}^{-1}\mathbf{q}_n\odot\mathbf{M}^{-1}\mathbf{y}_n)$: $O(M)$;
  - Computation complexity of $\sum_{n=1}^{N-1}(\mathbf{Q}^{-1}\mathbf{q}_n\odot\mathbf{M}^{-1}\mathbf{y}_n)^\top(\mathbf{Q}^{-1}\mathbf{q}_n\odot\mathbf{M}^{-1}\mathbf{y}_n)$: $O(MN)$.
- Computational complexity for $(\mathbf{X}'\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}'\nabla_t\mathbf{u}$: $O(MN)$.

□

F Proof of the computation complexity in local polynomial regression

Proof. To solve $\partial_x u(x,t_n)$ for $x\in\{x_0,x_1,\ldots,x_{M-1}\}$, we do Taylor expansion as

$$\begin{cases} u(x_0,t_n) &= u(x_0,t)+\partial_t^1 u(x_0,t_n)(x_0-x)+\ldots+\partial_t^{\check{p}}u(x_0,t_n)(x_0-x)^{\check{p}} \\ u(x_1,t_1) &= u(x_1,t_n)+\partial_t^1 u(x_1,t)(x_1-x)+\ldots+\partial_t^{\check{p}}u(x_1,t_n)(x_1-x)^{\check{p}} \\ \vdots & \vdots \\ u(x_{M-1},t_n) &= u(x,t_n)+\partial_t^1 u(x_i,t)(x_{M-1}-x)+\ldots+\partial_t^{\check{p}}u(x_{M-1},t_n)(x_{M-1}-x)^{\check{p}} \end{cases}$$

And we denote $\mathbf{c}(t)=(u(x_0,t_n),\partial_x^1 u(x,t_n),\ldots,\partial_x^{\check{p}}u(x,t_n))'$, which can be obtained as the second entry of the vector $\mathbf{c}(t)$, and $\mathbf{c}(t)$ is obtained by the following optimization problem:

$$\widehat{\mathbf{c}(x)}=\arg\min_{\mathbf{c}(t)}\sum_{i=0}^{M-1}\left[u_i^n-\sum_{j=0}^{\check{p}}\partial_x^j u(x,t_n)(x_i-x)^j\right]^2\mathcal{K}\left(\frac{x_i-x}{h}\right) \tag{17}$$

Essentially, the optimization problem in equation (15) is a weighted least square model, where $\mathbf{c}(x)$ can be solved in a close form:

$$\mathbf{c}(x)=\left(\mathbf{X}^\top\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{W}\mathbf{u}_{\cdot}^n, \tag{18}$$

where

$$\mathbf{X}=\begin{pmatrix}1 & (x_0-x) & \dots & (x_0-x)^4\\ 1 & (x_1-x) & \dots & (x_1-x)^4\\ \vdots & \vdots & \ddots & \vdots\\ 1 & (x_{M-1}-x) & \dots & (x_{M-1}-x)^4\end{pmatrix}_{M\times 5}, \mathbf{u}_{:}^{n}=\begin{pmatrix}u_0^n\\ u_1^n\\ \vdots\\ u_{M-1}^n\end{pmatrix}_{M\times 1}$$

and $\mathbf{W}=\operatorname{diag}\left(\mathcal{K}\left(\frac{x_0-x}{h}\right),\dots,\mathcal{K}\left(\frac{x_{M-1}-x}{h}\right)\right)\triangleq\operatorname{diag}\left(w_0,\dots,w_{M-1}\right)$.

$$\mathbf{X}^\top\mathbf{W}\mathbf{X}=\begin{pmatrix}\sum_{i=0}^{M-1}w_i & \sum_{i=0}^{M-1}w_i(x_i-x) & \sum_{i=0}^{M-1}w_i(x_i-x)^2 & \sum_{i=0}^{M-1}w_i(x_i-x)^3 & \sum_{i=0}^{M-1}w_i(x_i-x)^4\\ \sum_{i=0}^{M-1}w_i(x_i-x) & \sum_{i=0}^{M-1}w_i(x_i-x)^2 & \sum_{i=0}^{M-1}w_i(x_i-x)^3 & \sum_{i=0}^{M-1}w_i(x_i-x)^4 & \sum_{i=0}^{M-1}w_i(x_i-x)^5\\ \sum_{i=0}^{M-1}w_i(x_i-x)^2 & \sum_{i=0}^{M-1}w_i(x_i-x)^3 & \sum_{i=0}^{M-1}w_i(x_i-x)^4 & \sum_{i=0}^{M-1}w_i(x_i-x)^5 & \sum_{i=0}^{M-1}w_i(x_i-x)^6\\ \sum_{i=0}^{M-1}w_i(x_i-x)^3 & \sum_{i=0}^{M-1}w_i(x_i-x)^4 & \sum_{i=0}^{M-1}w_i(x_i-x)^5 & \sum_{i=0}^{M-1}w_i(x_i-x)^6 & \sum_{i=0}^{M-1}w_i(x_i-x)^7\\ \sum_{i=0}^{M-1}w_i(x_i-x)^4 & \sum_{i=0}^{M-1}w_i(x_i-x)^5 & \sum_{i=0}^{M-1}w_i(x_i-x)^6 & \sum_{i=0}^{M-1}w_i(x_i-x)^7 & \sum_{i=0}^{M-1}w_i(x_i-x)^8\end{pmatrix}_{5\times 5}$$

$$\mathbf{X}^\top\mathbf{W}\mathbf{u}_{:}^{n}=\begin{pmatrix}\sum_{i=0}^{M-1}w_iu_i^n\\ \sum_{i=0}^{M-1}w_i(x_i-x)u_i^n\\ \sum_{i=0}^{M-1}w_i(x_i-x)^2u_i^n\\ \sum_{i=0}^{M-1}w_i(x_i-x)^3u_i^n\\ \sum_{i=0}^{M-1}w_i(x_i-x)^4u_i^n\end{pmatrix}_{5\times 1}$$

Because

- For a fixed t,x, the computational complex of computing $\mathbf{X}^\top\mathbf{W}\mathbf{X}$ is $O(M)$.
- For a fixed t,x, with known $\mathbf{X}^\top\mathbf{W}\mathbf{X}$ is $O(M^2)$, the computational complex of computing $(\mathbf{X}^\top\mathbf{W}\mathbf{X})^{-1}$ is at most $O(5^3)$.
- For a fixed t,x, the computational complex of computing $\mathbf{X}^\top\mathbf{W}\mathbf{u}_{:}^{n}$ is $O(M)$.
- For a fixed t,x, the computational complex of computing $(\mathbf{X}^\top\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{W}\mathbf{u}_{:}^{n}$ is $\max\{O(M),O(5^3)\}$.
- For a fixed t and all $\{x_i\}_{i=0,\dots,M-1}$, the computational complex of computing all the derivatives w.r.t x is $\max\{O(M^2),O(5^3M)\}$.
- For $\{t_n\}_{n=0,\dots,N-1}$ and all $\{x_i\}_{i=0,\dots,M-1}$, the computational complex of computing all the derivatives w.r.t x is $\max\{O(NM^2),O(5^3NM)\}$.

- Similarly, for all $\{x_i\}_{i=0,\ldots,M-1}$ and all $\{t_n\}_{n=0,\ldots,N-1}$, the computational complex of computing all the derivatives w.r.t t is $\max\{O(MN^2), O(5^3NM)\}$.

The above computation complexity only involves the computation of the derivatives. Now, we discuss the computation complexity on solve the ridge regression estimator $\widehat{\boldsymbol{\beta}}$.

$$\begin{aligned} & \widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} (\nabla_t \mathbf{u} - \mathbf{X}\boldsymbol{\beta})^2 + \lambda \boldsymbol{\beta}^2 \\ \Rightarrow \quad & \widehat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \nabla_t \mathbf{u} \end{aligned}$$

$$\mathbf{X} = \begin{pmatrix} 1 & u_0^0 & \partial_x u_0^0 & \partial_x^2 u_0^0 \\ 1 & u_1^0 & \partial_x u_1^0 & \partial_x^2 u_1^0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & u_{M-1}^0 & \partial_x u_{M-1}^0 & \partial_x^2 u_{M-1}^0 \\ 1 & u_0^1 & \partial_x u_0^1 & \partial_x^2 u_0^1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & u_{M-1}^{N-1} & \partial_x u_{M-1}^{N-1} & \partial_x^2 u_{M-1}^{N-1} \end{pmatrix}$$

$$\mathbf{X}^\top \nabla_t \mathbf{u} = \begin{pmatrix} \sum_{i=0}^{M-1} \sum_{n=0}^{N-1} \partial_t u_i^n \\ \sum_{i=0}^{M-1} \sum_{n=0}^{N-1} u_i^n \partial_t u_i^n \\ \sum_{i=0}^{M-1} \sum_{n=0}^{N-1} \partial_x u_i^n \partial_t u_i^n \\ \sum_{i=0}^{M-1} \sum_{n=0}^{N-1} \partial_x^2 u_i^n \partial_t u_i^n \end{pmatrix}$$

- Computational complexity of the derivation of all entries in $\mathbf{X}$: $\max\{O(NM^2), O(5^3NM)\}$.
- Computational complexity of the derivation of all entries in $\nabla_t \mathbf{u}$: $\max\{O(MN^2), O(5^3NM)\}$.
- Computational complexity of computing $\mathbf{X}^\top \mathbf{X}$ is $O(MN)$
- Computational complexity of $\mathbf{X}^\top \nabla_t \mathbf{u}$: $O(MN)$.
- With known $\mathbf{X}^\top \mathbf{X}$ and $\mathbf{X}^\top \nabla_t \mathbf{u}$, the computational complexity of $(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \nabla_t \mathbf{u}$ is at most $O(4^3)$.
- Computational complexity of deriving $(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \nabla_t \mathbf{u}$ is $\max\{O(NM^2), O(MN^2), O(5^3NM)\}$.

□

G Proof of the Bound of $|\widehat{u(x,t_n)} - u^*(x,t_n)|$

Proof. The fitted value of the smoothing cubic spline is the minimizer of the following optimization problem:

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

It is well know and easily shown from the quadratic nature of the above optimization problem, that the spline smoother $\widehat{s}$ is linear in the observation $\{u_i^n\}_{i=0,\ldots,M-1}$. Thus, there exists a weight function $K(\cdot)$ such that

$$\widehat{\mathbf{f}_i} = \frac{1}{M}\sum_{j=0}^{M-1} K(x_i, x_j)\, u_j^n,$$

where $\widehat{\mathbf{f}_i} = \widehat{u(x_i,t_n)}$.

By Theorem A in Silverman [1984] (also mentioned by Messer et al. [1991] in the Section 1) that when assumption of (A4) $\sim$ (A6) hold and for large M and small λ, we have

$$\widehat{\mathbf{f}_i} = \frac{1}{M\lambda^{1/4}}\sum_{j=0}^{M-1} K\left(\frac{x_i - x_j}{\lambda^{1/4}}\right) u_j^n,$$

For a general spatial variable x, we denote $f^* = u^*(x,t_n)$, $\widehat{f} = \widehat{u(x,t_n)} = \frac{1}{M\lambda^{1/4}}\sum_{j=0}^{M-1} K\left(\frac{x-x_j}{\lambda^{1/4}}\right) u_j^n$. In order to bound $P\left(\sup|\widehat{f} - f^*| > \epsilon\right)$, we decompose it as follows:

$$\begin{aligned}
P\left(\sup|\widehat{f} - f^*| > \epsilon\right) &\overset{(19a)}{=} P\left(\sup|\widehat{f} - \widehat{f^B} + \widehat{f^B} - \widehat{f^*}| > \epsilon\right) \\
&= P\left(\sup|\widehat{f} - \widehat{f^B} - E(\widehat{f} - \widehat{f^B}) + E(\widehat{f} - \widehat{f^B}) + f^B - \widehat{f^*}| > \epsilon\right) \\
&= P\left(\sup|\underbrace{\widehat{f} - \widehat{f^B}}_{\mathcal{A}} - \underbrace{E(\widehat{f} - \widehat{f^B})}_{\mathcal{B}} + \underbrace{E(\widehat{f}) - f^*}_{\mathcal{C}} + \underbrace{\widehat{f^B} - E(\widehat{f^B})}_{\mathcal{D}}| > \epsilon\right) \\
&\leq P\left(\sup|\mathcal{A}| > \tfrac{\epsilon}{4}\right) + P\left(\sup|\mathcal{B}| > \tfrac{\epsilon}{4}\right) + P\left(\sup|\mathcal{C}| > \tfrac{\epsilon}{4}\right) + P\left(\sup|\mathcal{D}| > \tfrac{\epsilon}{4}\right)
\end{aligned} \tag{19}$$

where the $\widehat{f^B}$ in (19a) the truncated estimator, which is defined as

$$\widehat{f^B} = \frac{1}{M\lambda^{1/4}}\sum_{j=0}^{M-1} K\left(\frac{x - x_j}{\lambda^{1/4}}\right) y_j \mathbb{1}\{u_j^n < B_M\},$$

where $\{B_M\}$ is an increasing sequence and $B_M \to +\infty$ as $M \to +\infty$.

In the remaining of the proof, we work on the bound of the four decomposed terms, i,e, $P\left(\sup|\mathcal{A}| > \frac{\epsilon}{4}\right), P\left(\sup|\mathcal{B}| > \frac{\epsilon}{4}\right), P\left(\sup|\mathcal{C}| > \frac{\epsilon}{4}\right), P\left(\sup|\mathcal{D}| > \frac{\epsilon}{4}\right)$.

For $\mathcal{A}$:

$$\begin{aligned} P\left(\sup\left|\widehat{f}-\widehat{f}^B\right|>\frac{\epsilon}{4}\right) &= P\left(\sup\left|\frac{1}{M\lambda^{1/4}}\sum_{j=0}^{M-1}K\left(\frac{x-x_j}{\lambda^{1/4}}\right)u_j^n\mathbb{1}\{u_j^n\geq B_M\}\right|>\frac{\epsilon}{4}\right) \\ &\leq P\left(\sup\left|\frac{K_{\max}}{M\lambda^{1/4}}\sum_{j=0}^{M-1}u_j^n\mathbb{1}\{u_j^n\geq B_M\}\right|>\frac{\epsilon}{4}\right) \end{aligned}$$

If we let $\frac{\epsilon}{4}>\frac{K_{\max}}{M\lambda^{1/4}}B_M$, then we have

$$\begin{aligned} P\left(\sup\left|\widehat{f}-\widehat{f}^B\right|>\frac{\epsilon}{4}\right) &\leq P\left(\exists i=0,\ldots,M-1, s.t.|u_i^n|\geq B_M\right) \\ &= P\left(\max_{i=0,\ldots,M-1}|u_i^n|\geq B_M\right) \end{aligned}$$

Let $C_M=B_M-\|U\|_{L^\infty(\Omega)}$, we have

$$\begin{aligned} P\left(\sup\left|\widehat{f}-\widehat{f}^B\right|>\frac{\epsilon}{4}\right) &\leq P\left(\max_{i=0,\ldots,M-1},|U_i^n-u_i^n|\geq B_M\right) \\ &\leq 2Me^{-C_M^2/(2\sigma^2)} \end{aligned}$$

For $\mathcal{B}$:

By the Proposition 1 of Mack and Silverman [1982], for some $s>0$ with $E(|U_i^n|^s)<+\infty$, we have

$$\begin{aligned} \mathcal{B} &= E\left(|\widehat{f}-\widehat{f}^B|\right) \\ &= E\left(\left|\frac{1}{M\lambda^{1/4}}\sum_{j=0}^{M-1}K\left(\frac{x-x_j}{\lambda^{1/4}}\right)u_j^n\mathbb{1}\{u_j^n\geq B_M\}\right|\right) \\ &\leq E\left(\frac{1}{M\lambda^{1/4}}\sum_{j=0}^{M-1}\left|K\left(\frac{x-x_j}{\lambda^{1/4}}\right)\right||u_j^n|\mathbb{1}\{u_j^n\geq B_M\}\right) \\ &= \frac{1}{\lambda^{1/4}}\int\int_{|y|\geq B_M}\left|K\left(\frac{x-x_j}{\lambda^{1/4}}\right)\right||y|f(x_j,y)dy\,dx_j \\ &\leq \int|K(\xi)|d\xi\times\sup_\tau\int_{|y|\geq B_M}|y|f(\tau,y)dy. \\ &= O(B_M^{1-s}) \end{aligned}$$

So we have

$$E\left(|\widehat{f}-\widehat{f}^B|\right)\leq AB_M^{1-s},$$

where $A = \int |K(\xi)|d\xi \times \sup_\tau \int_{|y|\geq B_M} |y|^s f(\tau, y|t_n)dy$ with $f(\cdot,\cdot|t_n)$ as the distribution of $(x, u(x, t_n))$.

So when $\frac{\epsilon}{4} > AB_M^{1-s}$, we have

$$P\left(E\left(|\widehat{f} - \widehat{f}^B|\right) \geq \frac{\epsilon}{4}\right) = 0$$

For $\mathcal{C}$:

According to the Lemma 5 in Rice and Rosenblatt [1983], we have that, when $\frac{\epsilon}{4} > \partial_x^3 u^*(0, t_n)\lambda^{3/4}\sqrt{2}e^{-x2^{-1/2}\lambda^{-1/4}}\cos(2^{-1/2}x\lambda^{-1/4})$ and $\lambda^3 M^8 \to \infty, \lambda \to 0$ as $M \to \infty$ hold, we have

$$P\left(\sup|\mathcal{C}| > \frac{\epsilon}{4}\right) = 0$$

For $\mathcal{D}$:

In order to bound $P\left(\sup|D| > \frac{\epsilon}{4}\right)$, we decompose $\mathcal{D}$ into two components, i.e., $\mathcal{D} = e_M(x, t_n) + \rho_M(x, t_n)$. The decomposition procedure and the definition of $e_M(x, t_n), \rho_M(x, t_n)$ are described in the following system of equations.

$$\begin{aligned}
\mathcal{D} &= \widehat{f}^B - E(\widehat{f}^B) \\
&= \frac{1}{\sqrt{M}\lambda^{1/4}} \int_{x\in\mathbb{R}} \int_{|y|<B_M} K\left(\frac{x_i - x}{\lambda^{1/4}}\right) yd\left(\sqrt{M}(F_M(x, y) - F(x, y))\right) \\
&= \frac{1}{\sqrt{M}\lambda^{1/4}} \int_{x\in\mathbb{R}} K\left(\frac{x_i - x}{\lambda^{1/4}}\right) \int_{|y|<B_M} yd \underbrace{\left(\sqrt{M}(F_M(x, y) - F(x, y))\right)}_{Z_M(x,y)} \\
&= \frac{1}{\sqrt{M}\lambda^{1/4}} \int_{x\in\mathbb{R}} K\left(\frac{x_i - x}{\lambda^{1/4}}\right) \left[\int_{|y|<B_M} yd(Z_M(x, y) - B_0(T(x, y))) + \int_{|y|<B_M} ydB_0(T(x, y))\right] \\
&= \frac{1}{\sqrt{M}\lambda^{1/4}} \int_{x\in\mathbb{R}} K\left(\frac{x_i - x}{\lambda^{1/4}}\right) \int_{|y|<B_M} yd(Z_M(x, y) - B_0(T(x, y))) \\
&\quad + \frac{1}{\sqrt{M}\lambda^{1/4}} \int_{x\in\mathbb{R}} K\left(\frac{x_i - x}{\lambda^{1/4}}\right) \int_{|y|<B_M} ydB_0(T(x, y)) \\
&= \underbrace{\frac{1}{\sqrt{M}\lambda^{1/4}} \int_{x\in\mathbb{R}} \int_{|y|<B_M} K\left(\frac{x_i - x}{\lambda^{1/4}}\right) yd(Z_M(x, y) - B_0(T(x, y)))}_{e_M(x,t_n)} \\
&\quad + \frac{1}{\sqrt{M}} \underbrace{\frac{1}{\lambda^{1/4}} \int_{x\in\mathbb{R}} \int_{|y|<B_M} K\left(\frac{x_i - x}{\lambda^{1/4}}\right) ydB_0(T(x, y))}_{\rho_M(x,t_n)}
\end{aligned}$$

where $F_M(\cdot,\cdot) := F_M(\cdot,\cdot|t_n)$ is the empirical c.d.f of $\{u_i^n\}_{i=0,\ldots,M-1}$, and $Z_M(x, y) = \sqrt{M}(F_M(x, y) - F(x, y))$ is a 2D-empirical process. Besides, B_0 is a sample path of 2D Brownian bride.

By decomposition $\mathcal{D}$, then $P\left(\sup|D| > \frac{\epsilon}{4}\right)$ can be bounded by

$$P\left(\sup|D| > \frac{\epsilon}{4}\right) \leq P\left(\sup|e_M(x,t_n)| > \frac{\epsilon}{8}\right) + P\left(\sup|\rho_M(x,t_n)| > \frac{\epsilon}{8}\right)$$

For $e_M(x,t_n)$, we have

$$\begin{aligned} & P\left(\sup|e_M(x,t_n)| > \frac{\epsilon}{8}\right) \\ = \; & P\left(\sup\left|\frac{1}{\sqrt{M}\lambda^{1/4}}\int_{x\in\mathbb{R}}\int_{|y|<B_M} K\left(\frac{x_i-x}{\lambda^{1/4}}\right) y d(Z_M(x,y) - B_0(T(x,y)))\right| > \frac{\epsilon}{8}\right) \\ \leq \; & P\left(\sup_x \frac{2B_M}{\sqrt{M}\lambda^{1/4}} \sup_{x,y}|Z_M(x,y) - B_0(T(x,y))| > \frac{\epsilon}{8}\right). \end{aligned}$$

Proved by Theorem 1 in Tusnády [1977], we know that, for any γ, we have

$$P\left(\sup_{x,y}|Z_M(x,y) - B_0(T(x,y))| > \frac{(C\log M + \gamma)\log M}{\sqrt{M}}\right) \leq Qe^{-L\gamma},$$

where C, Q, L are absolute positive constants. Thus, when $\frac{\epsilon}{8} \geq \frac{2B_M(C\log M+\gamma)\log M}{\lambda^{1/4}M}$, we have

$$P\left(\sup|e_M(x,t_n)| > \frac{\epsilon}{8}\right) < Qe^{-L\gamma}.$$

For $\rho_M(x,t_n)$, by equation (7) in Mack and Silverman [1982], we have

$$\begin{aligned} \frac{\lambda^{1/8}\sup|\rho_M(x,t_n)|}{\sqrt{\log(1/\lambda^{1/4})}} \leq \; & 16(\log V)^{1/2}S^{1/2}\left(\log(\tfrac{1}{\lambda^{1/4}})\right)^{-1/2}\int|\xi|^{1/2}|dK(\xi)| \\ & +16\sqrt{2}\lambda^{-1/8}\left(\log(\tfrac{1}{\lambda^{1/4}})\right)^{-1/2}\int q(S\lambda^{1/4}|\tau|)|d(K(\tau))|, \end{aligned} \tag{20}$$

where V is a random variable satisfying $E(V) \leq 4\sqrt{2}\eta^4$ for $\eta^2 = \max_{i=0,\ldots,M-1} E(U_i^n)^2$, $q(z) = \int_0^z \frac{1}{2}\sqrt{\frac{1}{y}\log\left(\frac{1}{y}\right)}dy$ and $S = \sup_x \int y^2 f(y,x)dy$.

Using arguments similar to Silverman [1978] (page. 180-181), the second term of (20) tends to

$$16\sqrt{2}\sqrt{S}\int|\xi|^{1/2}|dK(\xi)|$$

by assumption (A7). Besides, the first term of (20) is $O_p(1)$ if $\lambda \to 0$.

Let $\bar{C} = 16\sqrt{S}\int|\xi|^{1/2}|dK(\xi)|$, when $\frac{\epsilon}{8} > 2\bar{C}\sqrt{\frac{2\log(1/\lambda^{1/4})}{M\lambda^{1/4}}}$, we have

$$P\left(\sup\left|\frac{1}{\sqrt{M}}\rho_M(x,t_n)\right| > \frac{\epsilon}{8}\right) < 4\sqrt{2}\eta^4 e^{-\frac{M(\epsilon/8)^2\lambda^{1/4}}{4\bar{C}\log(1/\lambda^{1/4})}}.$$

So when

- $\frac{\epsilon}{4} > \frac{K_{\max}}{M\lambda^{1/4}} B_M$
- $\frac{\epsilon}{4} > AB_M^{1-s}$
- $\frac{\epsilon}{4} > \partial_x^3 u^*(0,t_n)\lambda^{3/4}\sqrt{2}e^{-x2^{-1/2}\lambda^{-1/4}}\cos(2^{-1/2}x\lambda^{-1/4})$
- $\frac{\epsilon}{8} = \frac{2B_M(C\log M+\gamma)\log M}{\lambda^{1/4}M}$
- $\frac{\epsilon}{8} > 2\bar{C}\sqrt{\frac{2\log(1/\lambda^{1/4})}{M\lambda^{1/4}}}$

we have

$$P\left(\sup|\mathcal{A}+\mathcal{B}+\mathcal{C}+\mathcal{D}|>\epsilon\right) < 2Me^{-\frac{C_M^2}{2\sigma^2}} + Qe^{-L\gamma} + 4\sqrt{2}\eta^4 e^{-72M}.$$

Let

$$\begin{array}{lll} E_1 = \frac{4K_{\max}}{M\lambda^{1/4}}B_M & E_2 = 4AB_M^{1-s} & E_3 = 4\sqrt{2}\partial_x^3 u^*(0,t_n)\lambda^{3/4} \\ E_4 = \frac{16B_M(C\log M+\gamma)\log M}{\lambda^{1/4}M} & E_5 = 16\bar{C}\sqrt{\frac{2\log(1/\lambda^{1/4})}{M\lambda^{1/4}}} & \end{array}$$

Let $\lambda = M^{-a}, B_M = M^b, \gamma = M^c/L$, then we have

$$\left\{\begin{array}{rcl} E_1 &=& \frac{4K_{\max}}{M\lambda^{1/4}}B_M = \frac{4K_{\max}}{M^{1-a/4-b}} \\ E_2 &=& 4AB_M^{1-s} = 4A\frac{1}{M^{b(s-1)}} \\ E_3 &=& 4\sqrt{2}\partial_x^3 u^*(0,t_n)\lambda^{3/4} = 4\sqrt{2}\partial_x^3 u^*(0,t_n)\frac{1}{M^{3a/4}} \\ E_4 &=& \frac{16B_M(C\log M+\gamma)\log M}{\lambda^{1/4}M} = \frac{16C(\log M)^2}{M^{1-a/4-b}} + \frac{16\log M}{LM^{1-a/4-b-c}} \\ E_5 &=& 16\bar{C}\sqrt{\frac{2\log(1/\lambda^{1/4})}{M\lambda^{1/4}}} = 16\bar{C}\sqrt{\frac{\frac{1}{2a}\log(M)}{M^{1-a/4}}} \end{array}\right.$$

To guarantee that $E_1, E_2, E_3, E_4, E_5 \to 0$ as $M \to +\infty$, we need

$$\left\{\begin{array}{l} 1-a/4-b>0 \\ b(s-1)>0 \\ a>0 \\ 1-a/4-b-c>0 \\ 1-a/4>0 \end{array}\right.$$

When we set a, b, c as follows

$$\left\{\begin{array}{l} a = \frac{19}{35} \\ b = \frac{16}{105} \\ c = \frac{32}{105} \end{array}\right. ,$$

we get that as long as $\epsilon > \frac{32\log M}{LM^{57/140}}$, we have

$$P\left(\sup|\mathcal{A}+\mathcal{B}+\mathcal{C}+\mathcal{D}|>\epsilon\right) < 2Me^{-\frac{(M^{16/105}-\|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}}$$

□

H Proof of the Bound of $|\widehat{\partial_x u(x,t_n)} - \partial_x u^*(x,t_n)|$

Proof. In order to bound $P\left(\sup|\widehat{\boldsymbol{\theta}}_k - \widehat{\boldsymbol{\theta}}_k^*| > \epsilon\right)$, we decompose it as follows:

$$\begin{aligned}
P\left(\sup|\widehat{\boldsymbol{\theta}}_k - \widehat{\boldsymbol{\theta}}_k^*| > \epsilon\right) &= P\left(\sup|\widehat{\boldsymbol{\theta}}_k - \widehat{\boldsymbol{\theta}}_k^{\,B} + \widehat{\boldsymbol{\theta}}_k^{\,B} - \widehat{\boldsymbol{\theta}}_k^*| > \epsilon\right) \\
&= P\left(\sup|\widehat{\boldsymbol{\theta}}_k - \widehat{\boldsymbol{\theta}}_k^{\,B} - E(\widehat{\boldsymbol{\theta}}_k - \widehat{\boldsymbol{\theta}}_k^{\,B}) + E(\widehat{\boldsymbol{\theta}}_k - \widehat{\boldsymbol{\theta}}_k^{\,B}) + \widehat{\boldsymbol{\theta}}_k^{\,B} - \widehat{\boldsymbol{\theta}}_k^*| > \epsilon\right) \\
&= P\left(\sup|\underbrace{\widehat{\boldsymbol{\theta}}_k - \widehat{\boldsymbol{\theta}}_k^{\,B}}_{\mathcal{A}} - \underbrace{E(\widehat{\boldsymbol{\theta}}_k - \widehat{\boldsymbol{\theta}}_k^{\,B})}_{\mathcal{B}} + \underbrace{E(\widehat{\boldsymbol{\theta}}_k) - \widehat{\boldsymbol{\theta}}_k^*}_{\mathcal{C}} + \underbrace{\widehat{\boldsymbol{\theta}}_k^{\,B} - E(\widehat{\boldsymbol{\theta}}_k^{\,B})}_{\mathcal{D}}| > \epsilon\right) \\
&\leq P\left(\sup|\mathcal{A}| > \frac{\epsilon}{4}\right) + P\left(\sup|\mathcal{B}| > \frac{\epsilon}{4}\right) + P\left(\sup|\mathcal{C}| > \frac{\epsilon}{4}\right) + P\left(\sup|\mathcal{D}| > \frac{\epsilon}{4}\right)
\end{aligned}$$

The first derivative of the smoothing cubic spline can be derived as

$$\widehat{\boldsymbol{\theta}} = \mathbf{Q}^{-1}\mathbf{B}\widehat{\mathbf{f}}.$$

By Theorem 1 of Rice and Rosenblatt [1983] (also mentioned by Messer et al. [1991] in the Section 1) that when assumption of (2.5)-(2.8) in Rice and Rosenblatt [1983] hold, we have

$$\widehat{\mathbf{f}}_i = \frac{1}{M\lambda^{1/4}} \sum_{j=0}^{M-1} K\left(\frac{x_i - x_j}{\lambda^{1/4}}\right) y_j.$$

By $\widehat{\boldsymbol{\theta}} = \mathbf{Q}^{-1}\mathbf{B}\widehat{\mathbf{f}}$, we know that $\widehat{\boldsymbol{\theta}}$ is a linear transformation of $\widehat{\mathbf{f}}$, so we can assume that

$$\widehat{\boldsymbol{\theta}}_i = \frac{1}{M\lambda^{1/4}} \sum_{j=0}^{M-1} w_{ij} K\left(\frac{x_i - x_j}{\lambda^{1/4}}\right) y_j$$

Accordingly, we build a truncated estimator of $\widehat{\boldsymbol{\theta}}_k$ as

$$\widehat{\boldsymbol{\theta}}_k^{\,B} = \frac{1}{M\lambda^{1/4}} \sum_{j=0}^{M-1} w_{ij} K\left(\frac{x_i - x_j}{\lambda^{1/4}}\right) y_j \mathbb{1}\{u_j^n < B_M\}$$

For $\mathcal{A}$:

$$\begin{aligned}
P\left(\sup\left|\widehat{\boldsymbol{\theta}}_k - \widehat{\boldsymbol{\theta}}_k^{\,B}\right| > \frac{\epsilon}{4}\right) &= P\left(\sup\left|\frac{1}{M\lambda^{1/4}} w_{ij} \sum_{j=0}^{M-1} K\left(\frac{x_i - x_j}{\lambda^{1/4}}\right) y_j \mathbb{1}\{u_j^n \geq B_M\}\right| > \frac{\epsilon}{4}\right) \\
&\leq P\left(\sup\left|\frac{w_{\max}}{M\lambda^{1/4}} \sum_{j=0}^{M-1} K\left(\frac{x_i - x_j}{\lambda^{1/4}}\right) y_j \mathbb{1}\{u_j^n \geq B_M\}\right| > \frac{\epsilon}{4}\right)
\end{aligned}$$

When we let $\frac{\epsilon}{4} > \frac{w_{\max}}{M\lambda^{1/4}} B_M$, we have

$$\begin{aligned} P\left(\sup\left|\widehat{\boldsymbol{\theta}}_k - \widehat{\boldsymbol{\theta}}_k^{\,B}\right| > \frac{\epsilon}{4}\right) &\leq P\left(\exists i = 0, \ldots, M-1, s.t. |u_i^n| > B_M\right) \\ &= P\left(\max_{i=0,\ldots,M-1}, |u_i^n| > B_M\right) \end{aligned}$$

Let $C_M = B_M - \|U\|_{L^\infty(\Omega)}$, we have

$$\begin{aligned} P\left(\sup\left|\widehat{\boldsymbol{\theta}}_k - \widehat{\boldsymbol{\theta}}_k^{\,B}\right| > \frac{\epsilon}{4}\right) &\leq P\left(\max_{i=0,\ldots,M-1}, |U_i^n - u_i^n| > B_M\right) \\ &\leq 2Me^{-C_M^2/(2\sigma^2)} \end{aligned}$$

For $\mathcal{B}$:

$$\begin{aligned} \mathcal{B} &= E(\widehat{\boldsymbol{\theta}}_k - \widehat{\boldsymbol{\theta}}_k^{\,B}) \\ &= E\left(\frac{1}{M\lambda^{1/4}} w_{ij} \sum_{j=0}^{M-1} K\left(\frac{x_i - x_j}{\lambda^{1/4}}\right) y_j \mathbb{1}\{u_j^n \geq B_M\}\right) \\ &\leq E\left(\frac{w_{\max}}{M\lambda^{1/4}} \sum_{j=0}^{M-1} K\left(\frac{x_i - x_j}{\lambda^{1/4}}\right) y_j \mathbb{1}\{u_j^n \geq B_M\}\right) \\ &= \int\int_{|y|\geq B_M} \frac{1}{\lambda^{1/4}} K\left(\frac{x_i - x_j}{\lambda^{1/4}}\right) |y| f(x_j, y) dy \, dx_j \\ &\leq w_{\max} A B_M^{1-s} \end{aligned}$$

For $\mathcal{A}, \mathcal{B}$:
When $\frac{\epsilon}{4} > \max\{\frac{w_{\max}}{M\lambda^{1/4}} B_M, w_{\max} A B_M^{1-s}\}$, we have

$$P\left(\sup|\mathcal{A} + \mathcal{B}| > \frac{\epsilon}{2}\right) \leq 2Me^{-C_M^2/(2\sigma^2)}.$$

For $\mathcal{C}$:
When $\frac{\epsilon}{4} > -\partial_x^3 u^*(0, t_n)\lambda^{1/2} e^{-2^{-1/2}\lambda^{-1/4}}[\sin(2^{-1/2}\lambda^{-1/4}) + \cos(2^{-1/2}\lambda^{-1/4})]$ and $\lambda^3 M^8 \to \infty, \lambda \to 0$ as $M \to \infty$ hold, we have

$$P\left(\sup|\mathcal{C}| > \frac{\epsilon}{4}\right) = 0$$

For $\mathcal{D}$:

$$
\begin{aligned}
\mathcal{D} &= \widehat{\boldsymbol{\theta}}_k^{\,B} - E(\widehat{\boldsymbol{\theta}}_k^{\,B}) \\
&= \frac{1}{\sqrt{M}\lambda^{1/4}} \int_{x\in\mathbb{R}} \int_{|y|<B_M} w_{ij} K\left(\frac{x_i - x}{\lambda^{1/4}}\right) yd\left(\sqrt{M}(F_M(x,y) - F(x,y))\right) \\
&= \frac{1}{\sqrt{M}\lambda^{1/4}} \int_{x\in\mathbb{R}} w_{ij} K\left(\frac{x_i - x}{\lambda^{1/4}}\right) \int_{|y|<B_M} yd \underbrace{\left(\sqrt{M}(F_M(x,y) - F(x,y))\right)}_{Z_M(x,y)} \\
&= \frac{1}{\sqrt{M}\lambda^{1/4}} \int_{x\in\mathbb{R}} w_{ij} K\left(\frac{x_i - x}{\lambda^{1/4}}\right) \left[\int_{|y|<B_M} yd(Z_M(x,y) - B_0(T(x,y))) + \int_{|y|<B_M} ydB_0(T(x,y))\right] \\
&= \frac{1}{\sqrt{M}\lambda^{1/4}} \int_{x\in\mathbb{R}} w_{ij} K\left(\frac{x_i - x}{\lambda^{1/4}}\right) \int_{|y|<B_M} yd(Z_M(x,y) - B_0(T(x,y))) \\
&\quad + \frac{1}{\sqrt{M}\lambda^{1/4}} \int_{x\in\mathbb{R}} w_{ij} K\left(\frac{x_i - x}{\lambda^{1/4}}\right) \int_{|y|<B_M} ydB_0(T(x,y)) \\
&= \underbrace{\frac{1}{\sqrt{M}\lambda^{1/4}} \int_{x\in\mathbb{R}} \int_{|y|<B_M} w_{ij} K\left(\frac{x_i - x}{\lambda^{1/4}}\right) yd(Z_M(x,y) - B_0(T(x,y)))}_{e_M(x,t_n)} \\
&\quad + \frac{1}{\sqrt{M}} \underbrace{\frac{1}{\lambda^{1/4}} \int_{x\in\mathbb{R}} \int_{|y|<B_M} w_{ij} K\left(\frac{x_i - x}{\lambda^{1/4}}\right) ydB_0(T(x,y))}_{\rho_M(x,t_n)}
\end{aligned}
$$

For $e_M(x,t_n)$, we have

$$
\begin{aligned}
& P\left(\sup|e_M(x,t_n)| > \frac{\epsilon}{8}\right) \\
= \; & P\left(\sup\left|\frac{1}{\sqrt{M}\lambda^{1/4}} \int_{x\in\mathbb{R}} \int_{|y|<B_M} w_{ij} K\left(\frac{x_i - x}{\lambda^{1/4}}\right) yd(Z_M(x,y) - B_0(T(x,y)))\right| > \frac{\epsilon}{8}\right) \\
\leq \; & P\left(w_{\max}\sup\left|\frac{1}{\sqrt{M}\lambda^{1/4}} \int_{x\in\mathbb{R}} \int_{|y|<B_M} K\left(\frac{x_i - x}{\lambda^{1/4}}\right) yd(Z_M(x,y) - B_0(T(x,y)))\right| > \frac{\epsilon}{8}\right).
\end{aligned}
$$

When $\frac{\epsilon}{8} = \frac{2w_{\max}B_M(C\log M+\gamma)\log M}{\lambda^{1/4}M}$, we have

$$
P\left(\sup|e_M(x,t_n)| > \frac{\epsilon}{8}\right) < Qe^{-L\gamma}.
$$

For $\rho_M(x,t_n)$, we have

$$\begin{aligned}\frac{\lambda^{1/8}\sup|\rho_M(x,t_n)|}{\log(1/\lambda^{1/4})} &\leq \frac{\lambda^{1/8}w_{\max}\int_{x\in\mathbb{R}}K\left(\frac{x_i-x}{\lambda^{1/4}}\right)\int_{|y|<B_M}yd(B_0(T(x,y)))}{\log(1/\lambda^{1/4})}\\ &\leq 16w_{\max}(\log V)^{1/2}S^{1/2}\left(\log(\frac{1}{\lambda^{1/4}})\right)^{-1/2}\int|\xi|^{1/2}|dK(\xi)|\\ &\quad +16\sqrt{2}w_{\max}\lambda^{-1/8}\left(\log(\frac{1}{\lambda^{1/4}})\right)^{-1/2}\int q(S\lambda^{1/4}|\tau|)|d(K(\tau))|\end{aligned}$$

When $\frac{\epsilon}{8}>2w_{\max}\bar{C}\sqrt{\frac{2\log(1/\lambda^{1/4})}{M\lambda^{3/4}}}$, we have

$$P\left(\sup\left|\frac{1}{\sqrt{M}}\rho_M(x,t_n)\right|>\frac{\epsilon}{8}\right)<2\sqrt{2}\eta^4e^{-\frac{M(\epsilon/8)^2\lambda^{3/4}}{4\bar{C}\log(1/\lambda^{1/4})}}.$$

So when

- $\frac{\epsilon}{4}>\frac{w_{\max}}{M\lambda^{1/4}}$ (Let $E_1=\frac{4w_{\max}}{M\lambda^{1/4}}$)
- $\frac{\epsilon}{4}>w_{\max}AB_M^{1-s}$ (Let $E_2=4w_{\max}AB_M^{1-s}$)
- $\frac{\epsilon}{4}>-\partial_x^3u^*(0,t_n)\lambda^{1/2}e^{-2^{-1/2}\lambda^{-1/4}}[\sin(2^{-1/2}\lambda^{-1/4})+\cos(2^{-1/2}\lambda^{-1/4})]$
- $\frac{\epsilon}{8}=\frac{2w_{\max}B_M(C\log M+\gamma)\log M}{\lambda^{1/4}M}$ (Let $E_3=\frac{16w_{\max}B_M(C\log M+\gamma)\log M}{\lambda^{1/4}M}$
- $\frac{\epsilon}{8}>2w_{\max}\bar{C}\sqrt{\frac{2\log(1/\lambda^{1/4})}{M\lambda^{3/4}}}$ (Let $E_4=16w_{\max}\bar{C}\sqrt{\frac{2\log(1/\lambda^{1/4})}{M\lambda^{3/4}}}$

we have

$$P\left(\sup|\mathcal{A}+\mathcal{B}+\mathcal{C}+\mathcal{D}|>\epsilon\right)<2Me^{-\frac{C_M^2}{2\sigma^2}}+Qe^{-L\gamma}+4\sqrt{2}\eta^4e^{-72M}.$$

Let $\lambda=M^{-a},B_M=M^b,\gamma=M^c/L$, then we have

$$\begin{cases}E_1 &= \frac{4w_{\max}}{M\lambda^{1/4}}B_M=\frac{4w_{\max}}{M^{1-a/4-b}}\\ E_2 &= 4w_{\max}AB_M^{1-s}=4w_{\max}A\frac{1}{M^{b(s-1)}}\\ E_3 &= -\partial_x^3u_0^{n*}\sqrt{\lambda}e^{-\frac{\lambda^{-1/4}}{\sqrt{2}}}[\sin(\frac{\lambda^{-1/4}}{\sqrt{2}})+\cos(\frac{\lambda^{-1/4}}{\sqrt{2}})]=-\partial_x^3u_0^{n*}M^{-a/2}e^{-\frac{M^{a/4}}{\sqrt{2}}}[\sin(\frac{M^{a/4}}{\sqrt{2}})+\cos(\frac{M^{a/4}}{\sqrt{2}})]\\ E_4 &= \frac{16w_{\max}B_M(C\log M+\gamma)\log M}{\lambda^{1/4}M}=\frac{16w_{\max}C(\log M)^2}{M^{1-a/4-b}}+\frac{16w_{\max}B_M\log M}{LM^{1-a/4-b-c}}\\ E_5 &= 16w_{\max}\bar{C}\sqrt{\frac{2\log(1/\lambda^{1/4})}{M\lambda^{3/4}}}=16w_{\max}\bar{C}\sqrt{\frac{\frac{1}{2a}\log(M)}{M^{1-3a/4}}}\end{cases}$$

Since $E_1\preceq E_4$, $E_2\preceq E_4$, $E_3\preceq E_4$, and we know that when $\frac{1}{2}(1-3a/4)=1-a/4-b$, $E_5\preceq E_4$. Besides, to garentee that $E_1,E_2,E_3,E_4,E_5\to 0$ as $M\to+\infty$, we need

$$\begin{cases}1-a/4-b>0\\ b(s-1)>0\\ a>0\\ 1-a/4-b-c>0\\ 1-3a/4>0\end{cases}$$

So when $\frac{1}{2}(1-3a/4) = 1-a/4-b$, $E_4 \preceq E_3$, we have $a < \frac{4}{3}$. Since $b = \frac{1}{2} + \frac{1}{8}a$, we have $b < \frac{2}{3}$. If we take $a = 1$, then $b = \frac{5}{8}$. So we have

$$P\left(\sup|\mathcal{A}+\mathcal{B}+\mathcal{C}+\mathcal{D}| > \epsilon\right) < 2Me^{-\frac{(M^{5/8}-\|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}}$$

□

I Proof of Bound of $\|\nabla_t \mathbf{u} - \mathbf{X}\boldsymbol{\beta}^*\|_\infty$

Proof.

$$\begin{aligned}
\|\nabla_t \mathbf{u} - \mathbf{X}\boldsymbol{\beta}^*\|_\infty &= \|\nabla_t \mathbf{u} - \nabla_t \mathbf{u}^* + \nabla_t \mathbf{u}^* - \mathbf{X}\boldsymbol{\beta}^*\|_\infty \\
&= \|\nabla_t \mathbf{u} - \nabla_t \mathbf{u}^* + \mathbf{X}^*\boldsymbol{\beta}^* - \mathbf{X}\boldsymbol{\beta}^*\|_\infty \\
&\leq \|\nabla_t \mathbf{u} - \nabla_t \mathbf{u}^*\|_\infty + \|(\mathbf{X}^* - \mathbf{X})\boldsymbol{\beta}^*\|_\infty
\end{aligned}$$

For $\|\nabla_t \mathbf{u} - \nabla_t \mathbf{u}^*\|_\infty$:

$$\begin{aligned}
P\left(\|\nabla_t \mathbf{u} - \nabla_t \mathbf{u}^*\|_\infty > \tfrac{\epsilon}{2}\right) &\leq P\left(\max_{i=0,\dots,M-1} \sup_{t\in[0,T_{\max}]} \left|\widehat{\partial u(x_i,t)} - \partial u^*(x_i,t)\right| > \tfrac{\epsilon}{2}\right) \\
&\leq \sum_{i=0}^{M-1} P\left(\sup_{t\in[0,T_{\max}]} \left|\widehat{\partial u(x_i,t)} - \partial u^*(x_i,t)\right| > \tfrac{\epsilon}{2}\right) \\
&\overset{(21a)}{\leq} 2Ne^{-\frac{(N^{5/9}-\|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}},
\end{aligned} \tag{21}$$

where inequity (21a) exists according to Lemma A.3, when $\frac{\epsilon}{2} > \frac{32C(\log N)^2}{N^{1/3}}$

For $\|(\mathbf{X}^* - \mathbf{X})\boldsymbol{\beta}^*\|_\infty$:

$$\begin{aligned}
P\left(\|(\mathbf{X}^* - \mathbf{X})\boldsymbol{\beta}^*\|_\infty > \tfrac{\epsilon}{2}\right) &\overset{(22a)}{\leq} P\left(\|\boldsymbol{\beta}^*\|_\infty \sup_{x\in[0,X_{\max}]} \sum_{k=1}^K \|(\mathbf{X}_k^*(x,t_n) - \mathbf{X}_k(x,t_n))\|_\infty > \tfrac{\epsilon}{2}\right) \\
&= P\left(\sup_{x\in[0,X_{\max}]} \sum_{k=1}^K \|(\mathbf{X}_k^*(x,t_n) - \mathbf{X}_k(x,t_n))\|_\infty > \tfrac{\epsilon}{2\|\boldsymbol{\beta}^*\|_\infty}\right) \\
&\leq \sum_{n=0}^{N-1}\sum_{k=1}^K P\left(\sup_{x\in[0,X_{\max}]} \|(\mathbf{X}_k^*(x,t_n) - \mathbf{X}_k(x,t_n))\|_\infty > \tfrac{\epsilon}{2K\|\boldsymbol{\beta}^*\|_\infty}\right) \\
&\overset{(22b)}{\leq} 2Me^{-\frac{(M^{5/9}-\|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}}
\end{aligned} \tag{22}$$

In inequality (22a), $\mathbf{X}_k(x,t_n) = \widehat{\partial_x^{k-1} u(x,t_n)}$ for $k = 1,2,3$ and $\mathbf{X}_4(x,t_n) = (\widehat{u(x,t_n)})^2$, $\mathbf{X}_5(x,t_n) = \widehat{u(x,t_n)}\widehat{\partial_x u(x,t_n)}$, $\mathbf{X}_6(x,t_n) = \widehat{u(x,t_n)}\widehat{\partial_x^2 u(x,t_n)}$, $\mathbf{X}_7(x,t_n) = (\widehat{\partial_x u(x,t_n)})^2$, $\mathbf{X}_8(x,t_n) = \widehat{\partial_x u(x,t_n)}\widehat{\partial_x^2 u(x,t_n)}$, $(\widehat{\partial_x^2 u(x,t_n)})^2$. Similarly, $\mathbf{X}_k^*(x,t_n)$ is defined on the true values. Inequality (22b) exists when $\frac{\epsilon}{2K\|\boldsymbol{\beta}^*\|_\infty} > \frac{32C(\log M)^2}{M^{1/3}}$.

Thus, we finish the proof of the theorem. □

J Proof of Support Set Recovery

Proof. By Lemma A.1 in Namjoon's paper, we have

$$P\left[\max_{j\in\mathcal{S}^c}|\widetilde{Z_j}|>\epsilon\right]\leq P\left[\|\nabla_t\mathbf{u}-\mathbf{X}\boldsymbol{\beta}^*\|_\infty>\frac{\lambda\epsilon}{\sqrt{K}}\right].$$

By Theorem A.5, we have

$$P\left(\|\nabla_t\mathbf{u}-\mathbf{X}\boldsymbol{\beta}^*\|_\infty>\epsilon\right)<2Ne^{-\frac{(N^{16/105}-\|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}}+2Me^{-\frac{(M^{16/105}-\|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}}.$$

Since $M=N$, by combining these two results, when

$$\frac{\lambda\epsilon}{\sqrt{K}}>\frac{32\log N}{N^{57/140}},,$$

we have

$$P\left[\max_{j\in\mathcal{S}^c}|\widetilde{Z_j}|>\epsilon\right]\leq 2Ne^{-\frac{(N^{16/105}-\|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}}+2Me^{-\frac{(M^{16/105}-\|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}}.$$

Recall $\check{\mathbf{z}}_{\mathcal{S}^c}=\mathbf{X}_{\mathcal{S}^c}^\top\mathbf{X}_{\mathcal{S}}(\mathbf{X}_{\mathcal{S}}^\top\mathbf{X}_{\mathcal{S}})^{-1}\mathbf{z}_{\mathcal{S}}-\mathbf{X}_{\mathcal{S}^c}^\top\mathbf{H}\frac{\left[(\mathbf{X}-\mathbf{X}^*)_{\mathcal{S}}\boldsymbol{\beta}^*_{\mathcal{S}}-(\nabla_t\mathbf{u}-\nabla_t\mathbf{u}^*)\right]}{\lambda MN}$, $\widetilde{Z_j}=(\mathbf{X}_{\mathcal{S}^c})_j^\top\mathbf{H}\frac{\left[(\mathbf{X}-\mathbf{X}^*)_{\mathcal{S}}\boldsymbol{\beta}^*_{\mathcal{S}}-(\nabla_t\mathbf{u}-\nabla_t\mathbf{u}^*)\right]}{\lambda MN}$, so we have

$$\begin{aligned}
P(\|\mathbf{z}_{\mathcal{S}^c}\|_\infty\geq 1) &= P\left(\left\|\mathbf{X}_{\mathcal{S}^c}^\top\mathbf{X}_{\mathcal{S}}(\mathbf{X}_{\mathcal{S}}^\top\mathbf{X}_{\mathcal{S}})^{-1}\mathbf{z}_{\mathcal{S}}-\mathbf{X}_{\mathcal{S}^c}^\top\mathbf{H}\frac{[(\mathbf{X}-\mathbf{X}^*)_{\mathcal{S}}\boldsymbol{\beta}^*_{\mathcal{S}}-(\nabla_t\mathbf{u}-\nabla_t\mathbf{u}^*)]}{\lambda MN}\right\|_\infty>1\right)\\
&\leq P\left(\left\|\mathbf{X}_{\mathcal{S}^c}^\top\mathbf{X}_{\mathcal{S}}(\mathbf{X}_{\mathcal{S}}^\top\mathbf{X}_{\mathcal{S}})^{-1}\mathbf{z}_{\mathcal{S}}\right\|_\infty>1-\mu\right)+\\
&\quad P\left(\left\|\mathbf{X}_{\mathcal{S}^c}^\top\mathbf{H}\frac{[(\mathbf{X}-\mathbf{X}^*)_{\mathcal{S}}\boldsymbol{\beta}^*_{\mathcal{S}}-(\nabla_t\mathbf{u}-\nabla_t\mathbf{u}^*)]}{\lambda MN}\right\|_\infty>\mu\right)\\
&\leq P\left(\left\|\mathbf{X}_{\mathcal{S}^c}^\top\mathbf{X}_{\mathcal{S}}(\mathbf{X}_{\mathcal{S}}^\top\mathbf{X}_{\mathcal{S}})^{-1}\right\|_\infty>1-\mu\right)+P\left(\max_{j\in\mathcal{S}^c}|\widetilde{Z_j}|>\mu\right)\\
&\leq P\left(\left\|\mathbf{X}_{\mathcal{S}^c}^\top\mathbf{X}_{\mathcal{S}}(\mathbf{X}_{\mathcal{S}}^\top\mathbf{X}_{\mathcal{S}})^{-1}\right\|_\infty>1-\mu\right)\\
&\quad +2Ne^{-\frac{(N^{16/105}-\|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}}+2Me^{-\frac{(M^{16/105}-\|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}}
\end{aligned}$$

The probability for proper support set recovery is

$$\begin{aligned}
P(\|\mathbf{z}_{\mathcal{S}^c}\|_\infty < 1) &= 1 - P(\|\mathbf{z}_{\mathcal{S}^c}\|_\infty \geq 1) \\
&\geq 1 - \Bigg[P\left(\left\| \mathbf{X}_{\mathcal{S}^c}^\top \mathbf{X}_{\mathcal{S}} (\mathbf{X}_{\mathcal{S}}^\top \mathbf{X}_{\mathcal{S}})^{-1} \right\|_\infty > 1 - \mu \right) + 2Ne^{-\frac{(N^{16/105} - \|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}} + \\
&\quad 2Me^{-\frac{(M^{16/105} - \|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}} \Bigg] \\
&\geq 1 - \Bigg[1 - P\left(\left\| \mathbf{X}_{\mathcal{S}^c}^\top \mathbf{X}_{\mathcal{S}} (\mathbf{X}_{\mathcal{S}}^\top \mathbf{X}_{\mathcal{S}})^{-1} \right\|_\infty \leq 1 - \mu \right) + 2Ne^{-\frac{(N^{16/105} - \|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}} + \\
&\quad 2Me^{-\frac{(M^{16/105} - \|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}} \Bigg] \\
&\geq 1 - \left[1 - P_\mu + 2Ne^{-\frac{(N^{16/105} - \|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}} + 2Me^{-\frac{(M^{16/105} - \|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}} \right] \\
&= P_\mu - 2Ne^{-\frac{(N^{16/105} - \|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}} - 2Me^{-\frac{(M^{16/105} - \|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}}
\end{aligned}$$

Let $M = N$, we finish the proof. □

K Proof of Estimation Error Bound

Proof. By KKT-condition, any minimizer $\boldsymbol{\beta}$ of the following optimization

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

satisfies:

$$-\frac{1}{MN} \mathbf{X}^\top (\nabla_t \mathbf{u} - \mathbf{X}\boldsymbol{\beta}) + \lambda \mathbf{z} = 0 \ \text{ for } \ \mathbf{z} \in \partial \|\boldsymbol{\beta}\|_1,$$

where $\partial \|\boldsymbol{\beta}\|_1$ is the sub-differential of $\|\boldsymbol{\beta}\|_1$. The above equation can be equivalently transformed into

$$\mathbf{X}^\top \mathbf{X} (\boldsymbol{\beta} - \boldsymbol{\beta}^*) + \mathbf{X}^\top \left[(\mathbf{X} - \mathbf{X}^*) \boldsymbol{\beta}^* - (\nabla_t \mathbf{u} - \nabla_t \mathbf{u}^*) \right] + \lambda MN \mathbf{z} = 0,$$

where we can solve $\boldsymbol{\beta} - \boldsymbol{\beta}^*$ as

$$\boldsymbol{\beta} - \boldsymbol{\beta}^* = (\mathbf{X}^\top \mathbf{X})^{-1} \left[\mathbf{X}^\top \left(\nabla_t \mathbf{u} - \nabla_t \mathbf{u}^* - (\mathbf{X} - \mathbf{X}^*) \boldsymbol{\beta}^* \right) - \lambda MN \mathbf{z} \right].$$

Thus, we have the following series of equations:

$$
\begin{aligned}
\max_{k\in\mathcal{S}} |\boldsymbol{\beta}_k - \boldsymbol{\beta}_k^*| &\leq \left\|(\mathbf{X}_{\mathcal{S}}^\top \mathbf{X}_{\mathcal{S}})^{-1}\right\|_\infty \left\|\mathbf{X}_{\mathcal{S}}^\top \left(\nabla_t \mathbf{u} - \nabla_t \mathbf{u}^* - (\mathbf{X}_{\mathcal{S}} - \mathbf{X}_{\mathcal{S}}^*)\boldsymbol{\beta}^*\right) - \lambda MN\mathbf{z}\right\|_\infty \\
&\leq \left\|(\mathbf{X}_{\mathcal{S}}^\top \mathbf{X}_{\mathcal{S}})^{-1}\right\|_\infty \left\|\mathbf{X}_{\mathcal{S}}^\top \left(\nabla_t \mathbf{u} - \nabla_t \mathbf{u}^* - (\mathbf{X}_{\mathcal{S}} - \mathbf{X}_{\mathcal{S}}^*)\boldsymbol{\beta}^*\right)\right\|_\infty + \lambda MN \left\|(\mathbf{X}_{\mathcal{S}}^\top \mathbf{X}_{\mathcal{S}})^{-1}\right\|_\infty \\
&\leq \left\|(\mathbf{X}_{\mathcal{S}}^\top \mathbf{X}_{\mathcal{S}})^{-1}\right\|_\infty \left\|\mathbf{X}_{\mathcal{S}}^\top \left(\nabla_t \mathbf{u} - \mathbf{X}_{\mathcal{S}}\boldsymbol{\beta}^*\right)\right\|_\infty + \lambda MN \left\|(\mathbf{X}_{\mathcal{S}}^\top \mathbf{X}_{\mathcal{S}})^{-1}\right\|_\infty \\
&= \left\|\left(\frac{\mathbf{X}_{\mathcal{S}}^\top \mathbf{X}_{\mathcal{S}}}{MN}\right)^{-1}\right\|_\infty \left(\frac{\left\|\mathbf{X}_{\mathcal{S}}^\top (\nabla_t \mathbf{u} - \mathbf{X}_{\mathcal{S}}\boldsymbol{\beta}^*)\right\|_\infty}{MN} + \lambda\right) \\
&\leq \sqrt{K} C_{\min} \left(\frac{\left\|\mathbf{X}_{\mathcal{S}}^\top (\nabla_t \mathbf{u} - \mathbf{X}_{\mathcal{S}}\boldsymbol{\beta}^*)\right\|_\infty}{MN} + \lambda\right) \\
&= \sqrt{K} C_{\min} \left(\frac{\left\|(\mathbf{X}_{\mathcal{S}} - \mathbf{X}_{\mathcal{S}}^* + \mathbf{X}_{\mathcal{S}}^*)^\top (\nabla_t \mathbf{u} - \mathbf{X}_{\mathcal{S}}\boldsymbol{\beta}^*)\right\|_\infty}{MN} + \lambda\right) \\
&\leq \sqrt{K} C_{\min} \left(\frac{\left(\left\|\mathbf{X}_{\mathcal{S}} - \mathbf{X}_{\mathcal{S}}^*\right\|_1 + \left\|\mathbf{X}_{\mathcal{S}}^*\right\|_1\right)^\top \|(\nabla_t \mathbf{u} - \mathbf{X}_{\mathcal{S}}\boldsymbol{\beta}^*)\|_\infty}{MN} + \lambda\right) \\
&\leq \sqrt{K} C_{\min} \left(\frac{\left(MN\sqrt{K}\left\|\mathbf{X}_{\mathcal{S}} - \mathbf{X}_{\mathcal{S}}^*\right\|_{\max} + \left\|\mathbf{X}_{\mathcal{S}}^*\right\|_1\right) \|(\nabla_t \mathbf{u} - \mathbf{X}_{\mathcal{S}}\boldsymbol{\beta}^*)\|_\infty}{MN} + \lambda\right) \\
&= \sqrt{K} C_{\min} \left(\left(MN\sqrt{K}\left\|\mathbf{X}_{\mathcal{S}} - \mathbf{X}_{\mathcal{S}}^*\right\|_{\max} + \left\|\mathbf{X}_{\mathcal{S}}^*\right\|_1\right) \frac{\|(\nabla_t \mathbf{u} - \mathbf{X}_{\mathcal{S}}\boldsymbol{\beta}^*)\|_\infty}{MN} + \lambda\right) \\
&\leq \sqrt{K} C_{\min} \left(\left(MN\sqrt{K}\left\|\mathbf{X}_{\mathcal{S}} - \mathbf{X}_{\mathcal{S}}^*\right\|_{\max} + \left\|\mathbf{X}_{\mathcal{S}}^*\right\|_1\right) \frac{4Ne^{-\frac{(N^{16/105} - \|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}}}{MN} + \lambda\right) \\
&= \sqrt{K} C_{\min} \left(\left(\sqrt{K}\left\|\mathbf{X}_{\mathcal{S}} - \mathbf{X}_{\mathcal{S}}^*\right\|_{\max} + \frac{\left\|\mathbf{X}_{\mathcal{S}}^*\right\|_1}{MN}\right) 4Ne^{-\frac{(N^{16/105} - \|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}} + \lambda\right) \\
&\leq \sqrt{K} C_{\min} \left[\left(2\sqrt{K}Ne^{-\frac{(N^{16/105} - \|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}} + \frac{\left\|\mathbf{X}_{\mathcal{S}}^*\right\|_1}{MN}\right) 4Ne^{-\frac{(N^{16/105} - \|u\|_{L^\infty(\Omega)})^2}{2\sigma^2}} + \lambda\right]
\end{aligned}
\tag{23}
$$

□

References

Bär, M., Hegger, R., and Kantz, H. (1999). Fitting partial differential equations to space-time dynamics. *Physical Review E*, 59(1):337.

Beck, A. and Tetruashvili, L. (2013). On the convergence of block coordinate descent type methods. *SIAM journal on Optimization*, 23(4):2037–2060.

Brunton, S. L., Proctor, J. L., and Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937.

Chartrand, R. (2011). Numerical differentiation of noisy, nonsmooth data. *ISRN Applied Mathematics*, 2011.

Chen, J. and Wu, H. (2008a). Efficient local estimation for time-varying coefficients in deterministic dynamic models with applications to hiv-1 dynamics. *Journal of the American Statistical Association*, 103(481):369–384.

Chen, J. and Wu, H. (2008b). Estimation of time-varying parameters in deterministic dynamic models. *Statistica Sinica*, pages 987–1006.

Fan, J., Gasser, T., Gijbels, I., Brockmann, M., and Engel, J. (1997). Local polynomial regression: optimal kernels and asymptotic minimax efficiency. *Annals of the Institute of Statistical Mathematics*, 49(1):79–99.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1.

Haberman, R. (1983). *Elementary applied partial differential equations*, volume 987. Prentice Hall Englewood Cliffs, NJ.

Huang, Y., Liu, D., and Wu, H. (2006). Hierarchical bayesian methods for estimation of parameters in a longitudinal hiv dynamic system. *Biometrics*, 62(2):413–423.

Huang, Y. and Wu, H. (2006). A bayesian approach for estimating antiviral efficacy in hiv dynamic models. *Journal of Applied Statistics*, 33(2):155–174.

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

Li, L., Brown, M. B., Lee, K.-H., and Gupta, S. (2002). Estimation and inference for a spline-enhanced population pharmacokinetic model. *Biometrics*, 58(3):601–611.

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(484):1570–1583.

Mack, Y.-p. and Silverman, B. W. (1982). Weak and strong uniform consistency of kernel regression estimates. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 61(3):405–415.

Messer, K. et al. (1991). A comparison of a spline estimate to its equivalent kernel estimate. *The Annals of Statistics*, 19(2):817–829.

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(1):292–300.

Parlitz, U. and Merkwirth, C. (2000). Prediction of spatiotemporal time series based on reconstructed local states. *Physical review letters*, 84(9):1890.

Putter, H., Heisterkamp, S., Lange, J., and De Wolf, F. (2002). A bayesian approach to parameter estimation in hiv dynamical models. *Statistics in medicine*, 21(15):2199–2214.

Ramsay, J. O. (1996). Principal differential analysis: Data reduction by differential operators. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(3):495–508.

Ramsay, J. O., Hooker, G., Campbell, D., and Cao, J. (2007). Parameter estimation for differential equations: a generalized smoothing approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(5):741–796.

Rice, J. and Rosenblatt, M. (1983). Smoothing splines: regression, derivatives and deconvolution. *The annals of Statistics*, pages 141–156.

Rudy, S. H., Brunton, S. L., Proctor, J. L., and Kutz, J. N. (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.

Silverman, B. W. (1978). Weak and strong uniform consistency of the kernel estimate of a density and its derivatives. *The Annals of Statistics*, pages 177–184.

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

Tran, G. and Ward, R. (2017). Exact recovery of chaotic systems from highly corrupted data. *Multiscale Modeling & Simulation*, 15(3):1108–1129.

Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494.

Tusnády, G. (1977). A remark on the approximation of the sample df in the multidimensional case. *Periodica Mathematica Hungarica*, 8(1):53–55.

Voss, H. U., Kolodner, P., Abel, M., and Kurths, J. (1999). Amplitude equations from spatiotemporal binary-fluid convection data. *Physical review letters*, 83(17):3422.

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*.

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(2):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(503):1009–1020.