Theoretical and Applied Mechanics Letters 13 (2023) 100480

Contents lists available at ScienceDirect

Theoretical and Applied Mechanics Letters

journal homepage: www.elsevier.com/locate/taml

Full Article

Machine learning of partial differential equations from noise data

Wenbo Cao, Weiwei Zhang*

School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China

ARTICLE INFO

Keywords:
Partial differential equation
Machine learning
Sparse regression
Noise data

ABSTRACT

Machine learning of partial differential equations (PDEs) from data is a potential breakthrough for addressing the lack of physical equations in complex dynamic systems. Recently, sparse regression has emerged as an attractive approach. However, noise presents the biggest challenge in sparse regression for identifying equations, as it relies on local derivative evaluations of noisy data. This study proposes a simple and general approach that significantly improves noise robustness by projecting the evaluated time derivative and partial differential term into a subspace with less noise. This method enables accurate reconstruction of PDEs involving high-order derivatives, even from data with considerable noise. Additionally, we discuss and compare the effects of the proposed method based on Fourier subspace and POD (proper orthogonal decomposition) subspace. Generally, the latter yields better results since it preserves the maximum amount of information.

1. Introduction

Partial differential equations (PDEs) are increasingly important in modern science. PDEs are used to describe mathematical laws behind physical systems and play a vital role in the analysis, prediction, and control of many systems. In the past, PDEs were derived via basic conservation laws, which resulted in many canonical models in physics, engineering, and other fields. However, in modern applications, the mechanisms of many complex systems remain unclear, making it difficult to derive PDEs. This is particularly true in fields such as multiphase flow, neuroscience, finance, bioscience, and others. In the past decade, with the rapid development of sensors, computing power and data storage, the cost of data collection and computing has been greatly reduced, resulting in a large amount of experimental data. Meanwhile the rapid development of machine learning [1] has also provided a reliable tool to discover the potential laws of the system from large datasets. Nowadays, the machine learning of differential equations has become a promising new technology to discover physical laws in complex systems.

Among all the methods investigated for model discovery [2–9], sparse regression [4–6] has gained the most attention in recent studies because of its ability to discover interpretable and generalizable models with a balance of accuracy and efficiency. This approach provides an important model discovery framework, relying on sparse linear regression to select the sparse terms to match the data from a predefined function library which containing many nonlinear and partial derivative terms. Sparse regression has been applied to many challenging problems to identify potential ordinary differential equations (ODEs) or PDEs, including fluid dynamics [10–12], turbulence closures [13–15], vortex-induced vibration [16], subsurface flow equations [17,18], and others. More details on the important applications and extensions of sparse regression can be found in Ref. [19].

Nowadays, the biggest challenge with sparse regression is to identify equations from noisy data. The sensitivity of sparse regression to noise seriously damages the reliability of the results because sparse regression relies on the accurate evaluation of derivatives, which is especially challenging for PDEs where noise can be strongly amplified when computing higher-order spatial derivatives. Various approaches have been proposed to improve the noise robustness of sparse regression, which can be roughly divided into three types. The first is to obtain smooth data by using smooth function to approximate noisy data globally, including the use of neural network [8,20–23] or filter program. This approach can filter out part of the noise, but usually introduces new approximation errors, which may result in the data no longer strictly satisfying the original governing equation. A typical demonstration is that performing such preprocessing on clean data usually leads to loss of accuracy. The second is to use accurate derivation approximation methods [5,21], including spline smoothing, polynomial fitting, Gaussian kernel smoothing and Tikhonov differentiation. These methods approximate the noisy data locally, and they also introduce approximation errors, but usually smaller than the first type of methods. The third is to avoid the numerical differentiation by explicitly or implicitly introducing numerical integration [20,24–26] for the ODE discovery or reduce the order of the derivative by using weak form [27–31] for the PDE discovery. These methods can greatly improve the robustness of sparse regression against noise. Because they are derived by mathematical methods, they will not lose accuracy or even improve accuracy when used for clean data. However,

* Corresponding author.
E-mail address: aeroelastic@nwpu.edu.cn (W. Zhang).

https://doi.org/10.1016/j.taml.2023.100480
Received 26 October 2023; Received in revised form 8 November 2023; Accepted 12 November 2023
Available online 14 November 2023

those methods are complex to use and often contain many hyperparameters.

The present paper significantly improves the noise robustness of sparse regression by simply projecting time derivative and library functions into a low noise subspace. Below we introduce our method and discuss its similarities and differences with other methods, and then give some representative examples.

2. Methods

Here, we first describe the general framework of sparse regression, then propose our improved method.

2.1. Sparse regression

Consider the general form of nonlinear PDE $u_t = N(u, u_x, u_{xx}, \cdots, x, \mu)$, where, the subscripts denote partial differentiation in either time or space, and $N(\cdot)$ is an unknown nonlinear function of the state $u(x,t)$. The method builds an over-completed library that contains all the terms that may appear in the PDE, such as

$$\theta(u) = \left[u, u^2, u_x, uu_x, u^2u_x, u_{xx}, uu_{xx}, u^2u_{xx}, u_{xxx}, uu_{xxx}, u^2u_{xxx}\right]. \tag{1}$$

Then, the time derivative u_t can be expressed as a linear combination of library terms

$$u_t = \xi_1 f_1(u) + \xi_2 f_2(u) + \cdots + \xi_K f_K(u) = \theta(u)\xi. \tag{2}$$

For the given experimental data of a physical field, the values of u_t and $f_k(u)$ at many spatiotemporal points are evaluated, which lead to a system of linear equations

$$\left(\boldsymbol{U}_t\right)_c = (\boldsymbol{\Theta}(u))_c\xi, \tag{3}$$

where, $(\cdot)_c$ is the operator that arranges the data into a column vector, $(\boldsymbol{\Theta})_c$ denotes $[(\boldsymbol{F}_1)_c, (\boldsymbol{F}_2)_c, \cdots, (\boldsymbol{F}_K)_c]$, $\boldsymbol{U}_t$ and $\boldsymbol{F}_k$ are matrixes whose i,j element holds the function values of u_t and $f_k(u)$, respectively, at the ith time at the jth space position, and ξ is a vector of unknown coefficients. Each element in ξ is a coefficient corresponding to a term in the PDE. Many dynamical systems have relatively few active terms in the governing equations. Thus, we may employ sparse regression to identify the sparse vector ξ, which signifies the fewest active terms from the library that result in a good model fit. This can be represented as

$$\xi = argmin_\xi \|\left(\boldsymbol{U}_t\right)_c - (\boldsymbol{\Theta}(u))_c\xi\|_2^2 + R(\xi). \tag{4}$$

The regularizer $R(\xi)$ is chosen to promote sparsity of ξ. For example, sequentially thresholded least-squares (STLS) [4] uses $R(\xi) = \lambda\|\xi\|_0$, whereas sequentially thresholded ridge regression (STRidge) [5] uses $R(\xi) = \lambda_1\|\xi\|_0 + \lambda_2\|\xi\|_2$. Solution of the sparse vector ξ reveals the hidden PDE of the given system.

2.2. Subspace projection denoising

Sparse regression is simple and effective, but it is difficult to identify PDE from noisy data because the local derivative evaluation strongly amplifies noise. In this paper, we propose subspace projection denoising (SPD) method to solve the thorny problem.

As mentioned above, as a basic assumption of sparse regression, any linear or nonlinear PDE can be expressed as a linear combination of library terms (Eq. (2)). Therefore, performing a same linear transformation on u_t and f_k, we have

$$L\left(u_t\right) = L(\theta)\xi, \tag{5}$$

where, L is a linear operator, $L(\theta)$ denotes $[L(f_1), L(f_2), \cdots, L(f_K)]$. To filter the noise as much as possible, we project both the time derivative $\boldsymbol{U}_t$ and the library function $\boldsymbol{F}_k$ into a subspace with less noise

$$\begin{aligned}\widetilde{\boldsymbol{U}}_t &= \boldsymbol{\Phi}_t^T \boldsymbol{U}_t \boldsymbol{\Phi}_x,\\ \widetilde{\boldsymbol{F}}_k &= \boldsymbol{\Phi}_t^T \boldsymbol{F}_k \boldsymbol{\Phi}_x,\end{aligned} \tag{6}$$

where, $\boldsymbol{\Phi}_t$ is a matrix whose columns represent basis functions related to time, and $\boldsymbol{\Phi}_x$ is a matrix whose columns represent basis functions related to space. Since the projection is a linear operator, the vector ξ can be obtained by solving

$$\left(\widetilde{\boldsymbol{U}}_t\right)_c = (\tilde{\boldsymbol{\Theta}})_c\xi, \tag{7}$$

where, $(\tilde{\boldsymbol{\Theta}})_c$ denotes $[(\tilde{\boldsymbol{F}}_1)_c, (\tilde{\boldsymbol{F}}_2)_c, \cdots, (\tilde{\boldsymbol{F}}_K)_c]$. The Fourier basis can be used as the basis function, in which case the projection (Eq. (6)) can be easily implemented by fast Fourier transformation (FFT) on each spatiotemporal axis, then the low frequency components are chosen to solve the vector ξ. On the other hand, the proper orthogonal decomposition (POD) is a linear method for establishing an optimal basis, or modal decomposition, of an ensemble of continuous or discrete functions. Therefore, POD basis can also be selected to better characterize the current data and PDE, in which case $\boldsymbol{\Phi}_t$ and $\boldsymbol{\Phi}_x$ are given by singular value decomposition of $\boldsymbol{U}$

$$\boldsymbol{U} \cong \boldsymbol{W}_r\boldsymbol{\Sigma}_r\boldsymbol{V}_r^* = \boldsymbol{\Phi}_t\boldsymbol{\Sigma}_r\boldsymbol{\Phi}_x^*, \tag{8}$$

where, $\boldsymbol{\Phi}_x^*$ is the transpose of $\boldsymbol{\Phi}_x$, r is a user-specified hyperparameter, which indicates that only the first r bases with larger energy are retained, and $\boldsymbol{\Sigma}_r = \mathrm{diag}(\lambda_i)$ is a diagonal matrix with the singular values of $\boldsymbol{U}$, which represents the energy of the corresponding POD basis function. Obviously, the proposed subspace projection denoising (SPD) method is independent of the evaluation method of derivatives. It only projects the evaluated time derivatives and library functions into a new space. Therefore, it can still be effectively combined with other robust derivation approximation methods. The SPD method is quite simple and requires only minimal additional processing (Eq. (6)), but it significantly enhances the robustness to noisy data, which will be verified later. Furthermore, the method is suitable for high dimensional spatiotemporal data and PDE, requiring only additional rounds of FFT or projection.

Before presenting the results, we further illustrate the proposed method by comparing it with low-pass filtering. When using Fourier basis, the proposed method is similar to low-pass filtering of the state $u(x,t)$, but their purposes and effects are different. Low-pass filtering of the state $u(x,t)$ may change its distribution so that the filtered data no longer satisfies the original governing equation. Therefore, it can only filter out little noise to preserve the real signal as much as possible. In contrast, the SPD method performs low-pass filtering on the evaluated time derivative and library functions. It can filter out part of the real signal to filter the noise as much as possible, and can use the lowest frequency component with low-noise data to identify the PDE, because Eq. (5) is always automatically satisfied for any linear operator L.

3. Parameter identification of PDEs

For simplicity, this subsection assumes that the equation structure is known, and only the parameters in the equation are calculated from the noisy data to evaluate the effectiveness of proposed method. Obviously, the accuracy of the calculated parameter errors directly affects the likelihood of identifying the equation from the library.

We adopt the definition of noise given by Rudy [5]

$$u_n = u + \sigma \times \mathrm{std}(u) \times \mathrm{randn}, \tag{9}$$

where, u is numerical solution, σ is the noise level, $\mathrm{std}(u)$ is the standard deviation of u, randn is a random variable with standard normal distribution, and u_n represents the data with noise level σ. The accuracy of the model reconstruction quantified by the relative errors

$$\mathrm{error}_i = \left|\left(\hat{\xi}_i - \xi_i\right)/\xi_i\right|, \tag{10}$$

where, ξ_i are the correct coefficients and $\hat{\xi}_i$ are the coefficient estimated from noisy data. In all cases, the hyperparameter r is set to 5. Therefore, the total number of equations in the system defined by Eq. (7) is 5^2 for 2-D PDE and 5^3 for 3-D PDE.

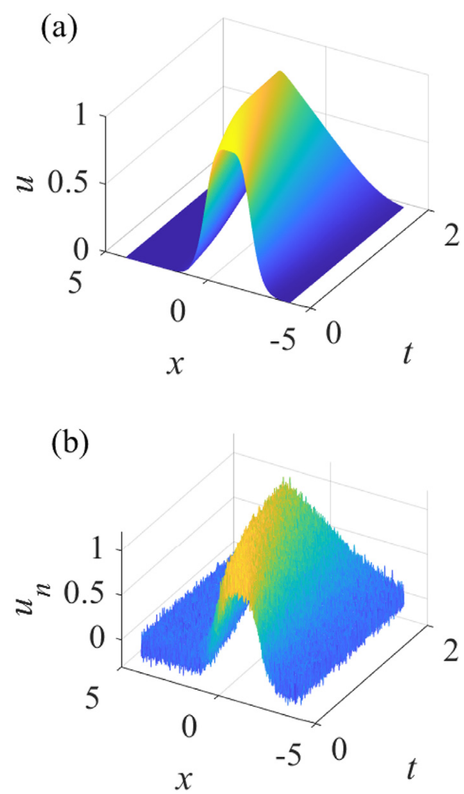

Fig. 1. Solution of Burgers' equation with (a) 0% noise and (b) 20% noise.

3.1. Burgers' equation

Burgers' equation arises in many technological contexts, including fluid mechanics, nonlinear acoustics, gas dynamics, and traffic flow. It takes the form $u_t = -uu_x + \nu u_{xx}$, where $\nu > 0$ is the diffusion coefficient. Burgers' equation is a non-linear second-order PDE. We test SPD method on Burgers' equation $u_t = \xi_1 uu_x + \xi_2 u_{xx}$with unknown coefficients $\xi_1 = -1, \xi_2 = 0.05$. The data set used to identify the coefficients is shown in Fig. 1.

Firstly, we compare the relative error distributions of $\boldsymbol{U}_{xx}$ evaluated from the data with 20% noise in physical space, Fourier space and POD space. Figure 2a shows that the error is almost evenly distributed throughout the original physical space. In Fourier space, the error decreases as the frequency decreases (Fig. 2b), while in POD space, the error decreases as the singular value increases (Fig. 2c). Therefore, projecting each term of the library into the Fourier subspace with lower frequencies or the POD subspace with higher energy can significantly reduce the error, as shown in the lower left corner of Fig. 2b and c.

After that, PDE-FIND is used as the baseline, in which the derivative is evaluated by finite difference, and SPD method is used to improve the noise robustness. The results for different noise levels are shown in Fig. 3, where "initial" refers to the identification results without using SPD method (Eq. (3)), while "SPD-Fourier" and "SPD-POD" refer to the identification results of projecting time derivative and library functions into Fourier subspace and POD subspace respectively (Eq. (7)).

As shown in Fig. 3, since the noise is strongly amplified when computing derivative by finite difference, the identified coefficients error of the initial method is greater than 10% for only 1% noise, and the coefficient error of u_{xx} is even close to 100%, which is actually the maximum error because the identified coefficient tends to zero as the noise level increases. Compared with the initial method, SPD method with either Fourier bases or POD bases has a great improvement, which can decrease the error by two to four orders of magnitude. Even for clean data, the SPD method has less error than the baseline. In addition, since the time derivative and library functions have larger components in the truncated POD subspace than Fourier subspace, the former has smaller errors.

In addition, we present a result of applying low-pass filtering on $u(x,t)$ with 0% noise, preserving the same frequencies as the SPD-Fourier method. As shown in Fig. 4, low-pass filtering introduces considerable error because it retains only very few frequencies. As a result, the coefficient identification error for clean data is as high as 20% ($\hat{\xi}_1 = -0.7973, \hat{\xi}_2 = 0.0602$). This supports the point mentioned in Section 2.2: low-pass filtering of $u(x,t)$ can cause the data to no longer satisfy the equation, whereas low-pass filtering of the evaluated time derivative and function libraries does not.

Furthermore, polynomial fitting is used to evaluate the local derivatives due to its robustness to noisy data. Figure 5 shows that polynomial fitting reduces the coefficient errors for the noisy data compared with

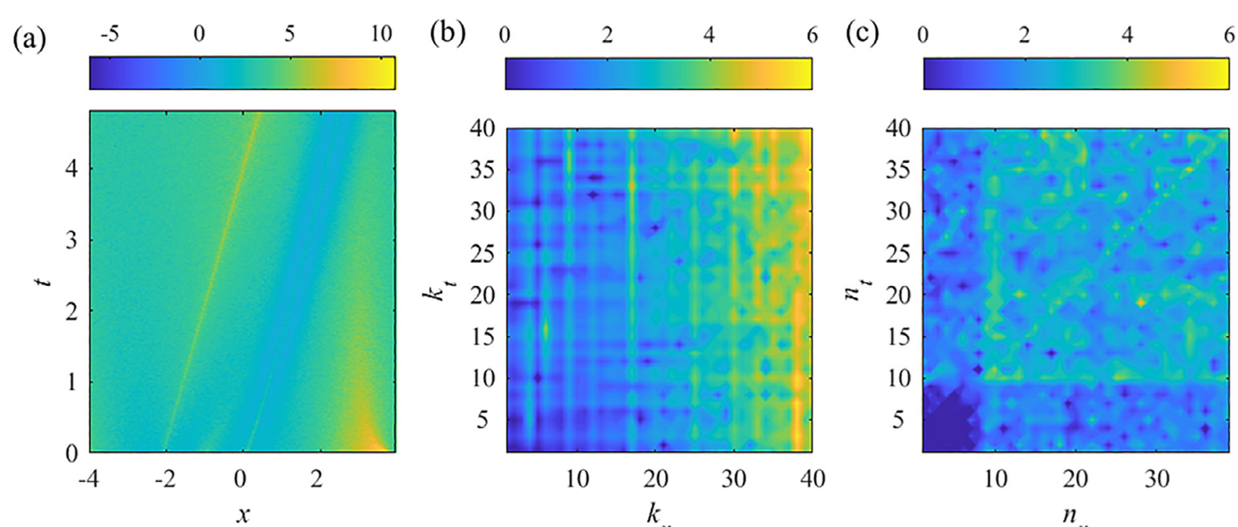

Fig. 2. Relative error $\log 10(|(\hat{U}_{xx} - U_{xx})/U_{xx}|)$ evaluated from the data with 20% noise in (a) physical space, (b) Fourier space and (c) POD space. k_x and k_t represent the corresponding coordinates in the Fourier space, which are frequency indexes. n_x and n_t represent the corresponding coordinates in the POD space, and the energy (i.e., singular value) of the corresponding basis function decreases as they increase. The number of basis functions for both time and space are limited to 40 to make the picture clearer.

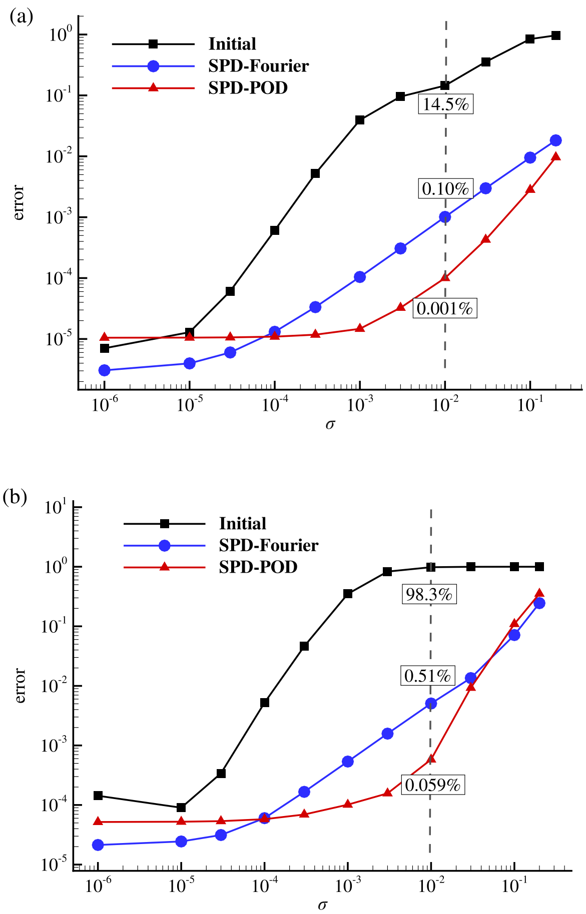

Fig. 3. The coefficient identification errors of (a) uu_x and (b) u_{xx} in Burgers' equation against different noise levels, where the local derivatives are evaluated by finite difference.

finite difference. But at the same time, the errors for clean data are also increased due to the approximation error of polynomial fitting. Therefore, as mentioned above, although polynomial fitting can improve the robustness to noise, it also inevitably introduces new approximation errors. In contrast, the SPD method filters noise without introducing any new errors because Eq. (5) is always true for any linear operator L. Although polynomial fitting reduces the derivative evaluation error for noisy data, the coefficient error of the initial method is still greater than 40% for 20% noise, while the SPD method can reduce the errors by two orders of magnitude.

3.2. Kuramoto–Sivashinsky equation

Kuramoto–Sivashinsky describes the chaotic dynamics of laminar flame fronts, reaction-diffusion systems, and coating flows. It takes the form $u_t = -uu_x - u_{xx} - u_{xxxx}$. This is a notable example of a nonlinear PDE that involves high-order partial derivatives, which has made it difficult to identify from noisy data accurately. We test SPD method on the equation $u_t = \xi_1 uu_x + \xi_2 u_{xx} + \xi_3 u_{xxxx}$ with unknown coefficients $\xi_1 = \xi_2 = \xi_3 = -1$. The data set used to identify the coefficients is shown in Fig. 6.

We use PDE-FIND as the baseline and polynomial fitting is used to evaluate the derivatives. Figure 7 shows the maximum coefficient error for different noise levels, which corresponds to one of the terms in PDE. It shows that even for high-order derivative with 20% noise, SPD method can still reduce the coefficient error to about 5%. This result is roughly equivalent to that of weak form [29,30], which is the most robust method for noisy data reported. Moreover, the SPD method is much

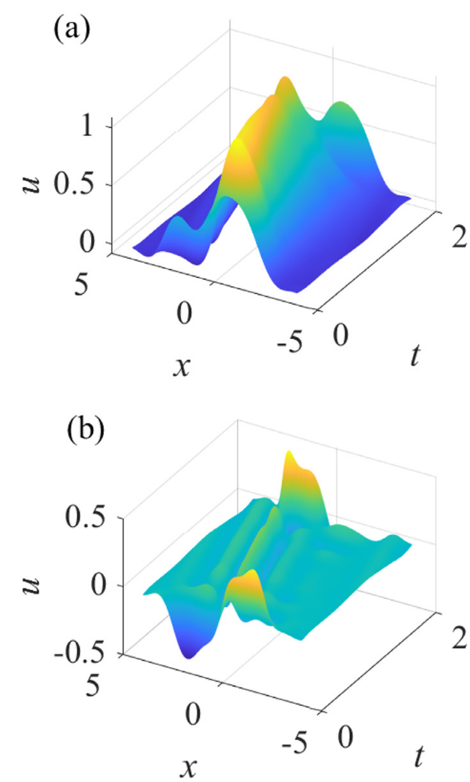

Fig. 4. (a) Low-pass filtered data. (b) The error between the filtered data and the original data (Fig. 1a).

simpler than the weak form. In addition, the identified coefficients have small error for clean data, which is caused by the approximation error of polynomial fitting.

4. Sparse regression

Finally, as an example of how the proposed method could be used in the context of sparse regression, we consider a numerical example in Ref. [29], which applies the weak form to the discover the $\lambda-\omega$ reaction-diffusion system

$$\begin{aligned} u_t &= D\nabla^2 u + \lambda u - \omega v, \\ v_t &= D\nabla^2 v + \omega u + \lambda u, \end{aligned} \tag{11}$$

where, $\omega = -(u^2+v^2)$, $\lambda = 1-u^2-v^2$, and $D = 0.1$ is constant. The weak form is one of the current state-of-the-art methods for identifying equations from noisy data. It attempts to reduce the order of the derivatives by multiplying basis functions over the terms in the library and integrating the result by parts over a spatiotemporal domain. In order to compare with the weak form, we use the same data set (as shown in Fig. 8) and library functions (as listed in Eq. (12)) as Ref. [29]. In total, the generalized model involves a total of 20 different terms (two diffusion terms and 18 polynomial terms). Correspondingly, 20 unknown coefficients need to be determined

$$\begin{aligned} \boldsymbol{\theta}_{u_t} &= \left[\nabla^2 u, u, u^2, u^3, v, v^2, v^3, uv, u^2 v, uv^2\right], \\ \boldsymbol{\theta}_{v_t} &= \left[\nabla^2 v, u, u^2, u^3, v, v^2, v^3, uv, u^2 v, uv^2\right]. \end{aligned} \tag{12}$$

Firstly, we evaluate the accuracy of identified parameter under different noise levels and compare it with the weak form [29]. The initial method and SPD method uses finite difference instead of polynomial fitting to evaluate the derivative to avoid the high computational cost. Figure 9 shows that under all noise levels, the errors of the weak form and SPD method are dramatically reduced compared with the baseline. When the noise level is greater than 3%, the weak form and SPD method have about the same accuracy; when the noise level is smaller, the SPD

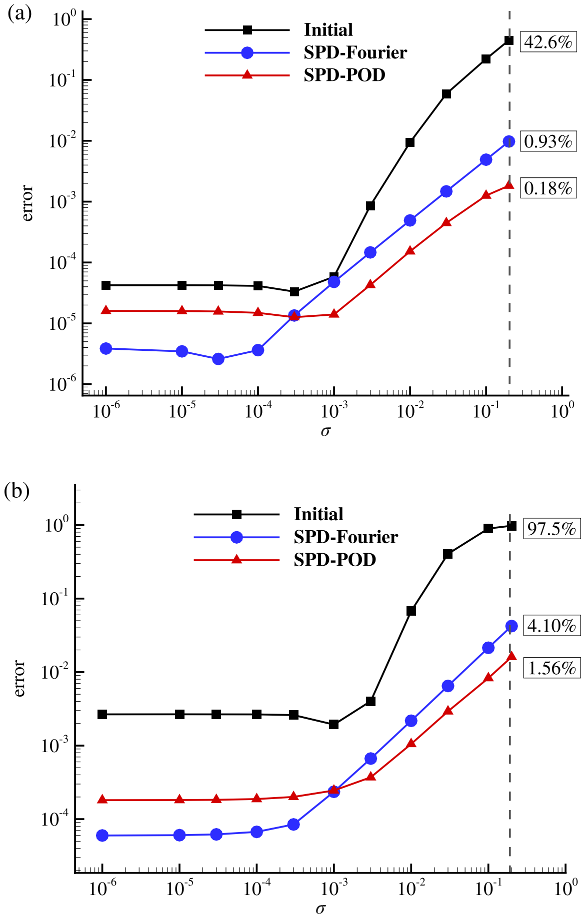

Fig. 5. The coefficient identification errors of (a) uu_x and (b) u_{xx} in Burgers' equation against different noise levels, where the local derivatives are evaluated by polynomial fitting.

method has higher accuracy. In addition, we emphasize that the implementation of SPD is much simpler than the weak form. In this case, the error of the SPD using Fourier bases is lower than that of the POD bases. This can be explained by the data's approximate periodicity along each space-time axis.

After that, we use the SINDy [4] algorithm to determine the parsimonious model. We find that, for noise levels of up to 10%, the PDE was identified correctly for 200 cases with different random noises by the SPD method with Fourier bases or POD bases. With 30% noise, the model is identified correctly in about 6% of cases, with the remaining cases featuring spurious terms that are not present in the $\lambda - \omega$ system. For reference, PDE-FIND failed to correctly identify this PDE for as little as 1% noise and the weak form correctly identify this PDE in about 95% of cases for 10% noise. Therefore, the SPD method has roughly the same effect as the weak form, with a dramatic improvement over the baseline.

5. Discussion

This work has developed a simple and effective method that greatly improves the robustness and accuracy for model discovery, reducing the data requirements and increasing noise tolerance. The proposed subspace projection denoising method projects the evaluated time derivative and library terms into the low frequency subspace with low noise or the POD subspace with high energy so as to greatly reduce the influence of noise. Moreover, the method can be used in combination with many other methods, including polynomial fitting, neural network smoothing, to further improve the robustness to noisy data. Several typical exam-

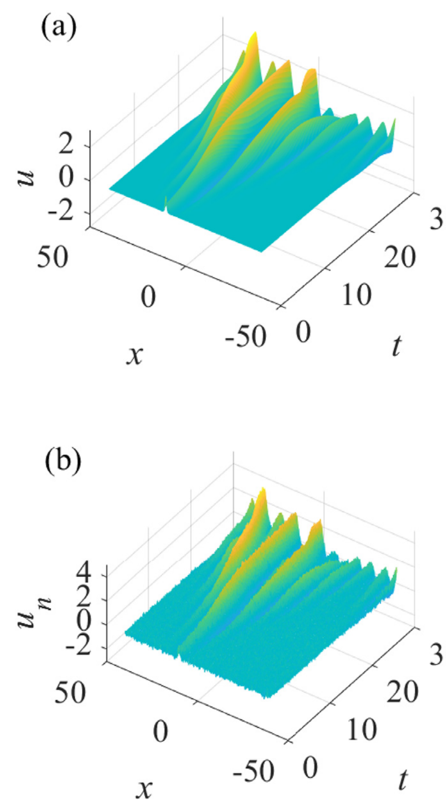

Fig. 6. Solution of Kuramoto–Sivashinsky equation with (a)0 % noise and (b) 20% noise.

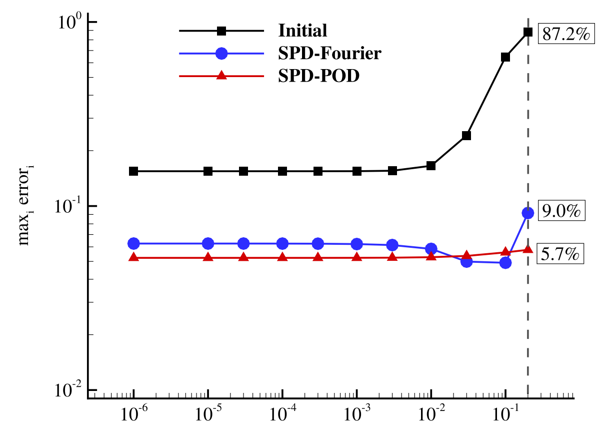

Fig. 7. Maximum coefficient error of Kuramoto–Sivashinsky equation against different noise levels.

ples show that compared with the baseline, the SPD method can reduce the error by several orders of magnitude, achieving the same effect as the weak form, while the SPD method is simpler and has only one hyperparameter. More importantly, this study points out that the original PDE automatically satisfied when any linear operator applied to evaluated time derivative and library functions, which provides a general framework for denoising, thus allowing more denoising methods with different linear operators.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

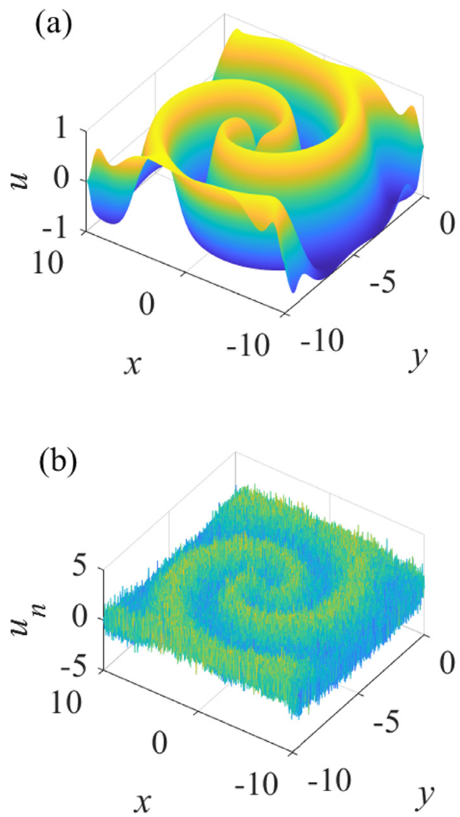

Fig. 8. A typical snapshot of the solution of $\lambda - \omega$ reaction-diffusion system with (a) 0% noise and (b) 100% noise.

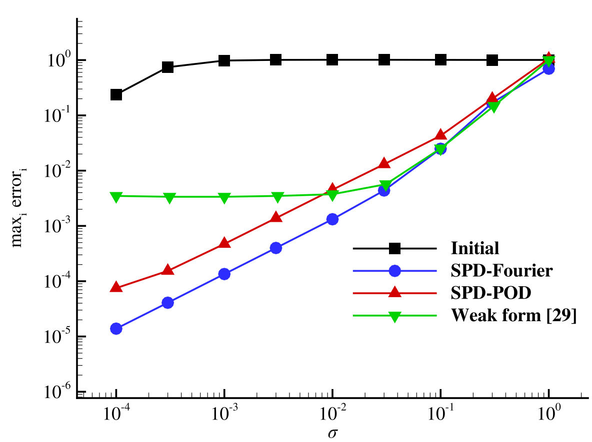

Fig. 9. Maximum coefficient error of $\lambda - \omega$ system against different noise levels.

Acknowledgments

We would like to acknowledge the support of the National Natural Science Foundation of China (Grant No. 92152301).

References

[1] M.I. Jordan, T.M. Mitchell, Machine learning: trends, perspectives, and prospects, Science 349 (6245) (2015) 255–260, doi:10.1126/science.aaa8415.
[2] J. Bongard, H. Lipson, Automated reverse engineering of nonlinear dynamical systems, Proc. Natl. Acad. Sci. U.S.A. 104 (24) (2007) 9943–9948, doi:10.1073/pnas.0609476104.
[3] M. Schmidt, H. Lipson, Distilling free-form natural laws from experimental data, Science 324 (5923) (2009) 81–85, doi:10.1126/science.1165893.
[4] S.L. Brunton, J.L. Proctor, J.N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proc. Natl. Acad. Sci. U.S.A. 113 (15) (2016) 3932–3937, doi:10.1073/pnas.1517384113.
[5] S.H. Rudy, et al., Data-driven discovery of partial differential equations, Sci. Adv. 3 (4) (2017), doi:10.1126/sciadv.1602614.
[6] H. Schaeffer, Learning partial differential equations *via* data discovery and sparse optimization, Proc. R. Soc. Math. Phys. Eng. Sci. 473 (2197) (2017), doi:10.1098/rspa.2016.0446.
[7] Z. Long, Y. Lu, B. Dong, PDE-Net 2.0: learning PDEs from data with a numeric-symbolic hybrid deep network, J. Comput. Phys. (2019) 399, doi:10.1016/j.jcp.2019.108925.
[8] H. Xu, H. Chang, D. Zhang, DLGA-PDE: discovery of PDEs with incomplete candidate library *via* combination of deep learning and genetic algorithm, J. Comput. Phys. (2020) 418, doi:10.1016/j.jcp.2020.109584.
[9] Z.L. Wang, et al., A perspective on regression and Bayesian approaches for system identification of pattern formation dynamics, Theor. Appl. Mech. Lett. 10 (3) (2020) 188–194, doi:10.1016/j.taml.2020.01.028.
[10] J.-C. Loiseau, S.L. Brunton, Constrained sparse Galerkin regression, J. Fluid Mech. 838 (2018) 42–67, doi:10.1017/jfm.2017.823.
[11] Y.F. Guan, S.L. Brunton, I. Novosselov, Sparse nonlinear models of chaotic electroconvection, R. Soc. Open Sci. 8 (8) (2021), doi:10.1098/rsos.202367.
[12] W.B. Cao, W.W. Zhang, Data-driven and physical-based identification of partial differential equations for multivariable system, Theor. Appl. Mech. Lett. 12 (2) (2022), doi:10.1016/j.taml.2022.100334.
[13] S. Beetham, J. Capecelatro, Formulating turbulence closures using sparse regression with embedded form invariance, Phys. Rev. Fluids 5 (8) (2020), doi:10.1103/PhysRevFluids.5.084611.
[14] M. Schmelzer, R.P. Dwight, P. Cinnella, Discovery of algebraic Reynolds-stress models using sparse symbolic regression, Flow Turbul. Combust. 104 (2–3) (2020) 579–603, doi:10.1007/s10494-019-00089-x.
[15] S. Beetham, R.O. Fox, J. Capecelatro, Sparse identification of multiphase turbulence closures for coupled fluid-particle flows, J. Fluid Mech. (2021) 914, doi:10.1017/jfm.2021.53.
[16] S. Li, et al., Discovering time-varying aerodynamics of a prototype bridge by sparse identification of nonlinear dynamical systems, Phys. Rev. E 100 (2) (2019), doi:10.1103/PhysRevE.100.022220.
[17] H. Chang, D. Zhang, Machine learning subsurface flow equations from data, Comput. Geosci. 23 (5) (2019) 895–910, doi:10.1007/s10596-019-09847-2.
[18] W. Song, et al., Data-driven discovery of soil moisture flow governing equation: a sparse regression framework, Water Resour. Res. 58 (8) (2022), doi:10.1029/2022wr031926.
[19] U. Fasel, et al., Ensemble-SINDy: robust sparse model discovery in the low-data, high-noise limit, with active learning and control, Proc. Math. Phys. Eng. Sci. 478 (2260) (2022) 20210904, doi:10.1098/rspa.2021.0904.
[20] S.H. Rudy, J.N. Kutz, S.L. Brunton, Deep learning of dynamics and signal-noise decomposition with time-stepping constraints, J. Comput. Phys. 396 (2019) 483–506, doi:10.1016/j.jcp.2019.06.056.
[21] Z. Zhang, Y. Liu, A robust framework for identification of PDEs from noisy data, J. Comput. Phys. (2021) 446, doi:10.1016/j.jcp.2021.110657.
[22] R. Stephany, C. Earls, PDE-READ: human-readable partial differential equation discovery using deep learning, Neural Netw. 154 (2022) 360–382, doi:10.1016/j.neunet.2022.07.008.
[23] K. Kaheman, S.L. Brunton, J.N. Kutz, Automatic differentiation to simultaneously identify nonlinear dynamics and extract noise probability distributions from data, Mach. Learn. Sci. Technol. 3 (1) (2022), doi:10.1088/2632-2153/ac567a.
[24] C. Cheng, et al., A two-stage sparse algorithm for localization and characterization of local nonlinear structures, J. Sound Vib. (2022) 526, doi:10.1016/j.jsv.2022.116823.
[25] P. Goyal, P. Benner, Discovery of nonlinear dynamical systems using a Runge-Kutta inspired dictionary-based sparse regression approach, Proc. R. Soc. Math. Phys. Eng. Sci. 478 (2262) (2022), doi:10.1098/rspa.2021.0883.
[26] F. Lejarza, M. Baldea, Data-driven discovery of the governing equations of dynamical systems *via* moving horizon optimization, Sci. Rep. 12 (1) (2022) 11836, doi:10.1038/s41598-022-13644-w.
[27] D.A. Messenger, D.M. Bortz, Weak SINDy: galerkin-based data-driven model selection, Multiscale Model. Simul. 19 (3) (2021) 1474–1497, doi:10.1137/20m1343166.
[28] H. Schaeffer, S.G. McCalla, Sparse model selection *via* integral terms, Phys. Rev. E (2) (2017) 96, doi:10.1103/PhysRevE.96.023302.
[29] P.A.K. Reinbold, D.R. Gurevich, R.O. Grigoriev, Using noisy or incomplete data to discover models of spatiotemporal dynamics, Phys. Rev. E 101 (1–1) (2020) 010203, doi:10.1103/PhysRevE.101.010203.
[30] D.A. Messenger, D.M. Bortz, Weak sindy for partial differential equations, J. Comput. Phys. (2021) 443, doi:10.1016/j.jcp.2021.110525.
[31] P.A.K. Reinbold, et al., Robust learning from noisy, incomplete, high-dimensional experimental data *via* physically constrained symbolic regression, Nat. Commun. 12 (1) (2021) 3219, doi:10.1038/s41467-021-23479-0.