Journal of Scientific Computing (2021) 87:1
https://doi.org/10.1007/s10915-020-01404-9

IDENT: Identifying Differential Equations with Numerical Time Evolution

Sung Ha Kang[1] · **Wenjing Liao**[1] · **Yingjie Liu**[1]

Received: 9 April 2019 / Revised: 7 March 2020 / Accepted: 28 December 2020 /
Published online: 17 February 2021

Abstract
Identifying unknown differential equations from a given set of discrete time dependent data is a challenging problem. A small amount of noise can make the recovery unstable. Nonlinearity and varying coefficients add complexity to the problem. We assume that the governing partial differential equation (PDE) is a linear combination of few differential terms in a prescribed dictionary, and the objective of this paper is to find the correct coefficients. We propose a new direction based on the fundamental convergence principle of numerical PDE schemes. We utilize Lasso for efficiency, and a performance guarantee is established based on an incoherence property. The main contribution is to validate and correct the results by time evolution error (TEE). A new algorithm, called identifying differential equations with numerical time evolution (IDENT), is explored for data with non-periodic boundary conditions, noisy data and PDEs with varying coefficients. Based on the recovery theory of Lasso, we propose a new definition of Noise-to-Signal ratio, which better represents the level of noise in the case of PDE identification. The effects of data generations and downsampling are systematically analyzed and tested. For noisy data, we propose an order preserving denoising method called least-squares moving average (LSMA), to preprocess the given data. For the identification of PDEs with varying coefficients, we propose to add Base Element Expansion (BEE) to aid the computation. Various numerical experiments from basic tests to noisy data, downsampling effects and varying coefficients are presented.

Keywords Identifying unknown differential equations · Time evolution error (TEE) · Varying coefficients · Base element expansion (BEE) · Denoising · Downsampling

S. H. Kang: Research is supported in part by Simons Foundation Grants 282311 and 584960.
W. Liao: Research is supported in part by the NSF Grants DMS 1818751 and DMS 2012652.
Y. Liu: Research is supported in part by NSF Grants DMS-1522585 and DMS-CDS&E-MSS-1622453.

✉ Wenjing Liao
wliao60@gatech.edu

Sung Ha Kang
kang@math.gatech.edu

Yingjie Liu
yingjie@math.gatech.edu

[1] School of Mathematics, Georgia Institute of Technology, Atlanta, USA

1 Introduction

Physical laws are often presented by the means of differential equations. The original discoveries of differential equations associated with real-world physical processes typically require a good understanding of the physical laws, and supportive evidence from empirical observations. We consider an inverse problem of this—from the experimental data, how to directly recognize the underlying PDE. We combine tools from machine learning and numerical PDEs to explore the given data and automatically identify the underlying dynamics.

Let $\{u_i^n | i = 1, \ldots, N_1 \text{ and } n = 1, \ldots, N_2\}$ be the given discrete time dependent data, where the index i and n represent the spatial and time discrete domain, respectively. Our objective is to find the differential equation, i.e., an operator $\mathcal{F}$:

$$u_t = \mathcal{F}(x, u, u_x, u_{xx}) \text{ such that } u(x_i, t_n) \approx u_i^n.$$

Recently there have been a number of important works on learning dynamical systems or differential equations. Two pioneering works can be found in [3,29], where symbolic regression was used to recover the underlying physical systems from experimental data. In [6], Brunton et al. considered the discovery of nonlinear dynamical systems with sparsity-promoting techniques. The underlying dynamical systems are assumed to be governed by a small number of active terms in a prescribed dictionary, and sparse regression is used to identify these active terms. Various extensions of this sparse regression approach can be found in [13,16,20,25]. In [26], Schaeffer considered the problem of learning PDEs using the spectral method, and focused on the benefit of using L^1 minimization for sparse coefficient recovery. Highly corrupted and undersampled data are considered in [28,31] for the recovery of dynamical systems. In [28], Schaeffer et al. developed a random sampling theory for the selection dynamical systems from undersampled data. These nice series of works focused on the benefit and power of using L^1 minimization to resolve dynamical systems or PDEs with certain sparse pattern [27]. A Bayesian approach was considered in [35] where Zhang et al. used dimensional analysis and sparse Bayesian regression to recover the underlying dynamical systems. Another related problem is to infer the interaction function in a system of agents from the trajectory data. In [4,18], nonparametric regression was used to predict the interaction function and a theoretical guarantee was established.

There are approaches using deep learning techniques. In [17], Long et al. proposed a PDE-Net to learn differential operators by learning convolution kernels. In [24], Raissi et al. used neural networks to learn and predict the solution of the equation without finding its explicit form. In [23], neural networks were further used to learn certain parameters in the PDEs from the given data. In [21], Residual Neural Networks (ResNet) are used as building blocks for equation approximation. In [14], neural networks are used to solve the wave equation based inverse scattering problems by providing maps between the scatterers and the scattered field (and vice versa). Related works showing the advantages of deep learning include [14,19,21,22].

In this paper, we propose a new algorithm based on the convergence principle of numerical PDE schemes. We assume that the governing PDE is a linear combination of few differential terms in a prescribed dictionary, and the objective is to find the correct set of coefficients. We use finite difference methods, such as the 5-point ENO scheme, to approximate the spatial derivatives in the dictionary. While we utilize L^1 minimization to aid the efficiency of the approach, the main idea is to validate and correct the results by Time Evolution Error (TEE). This approach, we call Identifying Differential Equations with Numerical Time evolution (IDENT) is explored for data with non-periodic boundary conditions, noisy data and PDEs with varying coefficients for nonlinear PDE identification. For noisy data, we propose an order

preserving denoising method called Least Square Moving Average (LSMA) to effectively denoise the given data. To tackle varying coefficients, we expand the number of coefficients in terms of finite element bases. This procedure called Base Element Expansion (BEE), again uses the fundamental idea of convergence in finite element approximation. From a theoretical perspective, we establish a performance guarantee based on an incoherence property, and define a new noise-to-signal ratio for the PDE identification problem. Contributions of this paper include:

1. establishing a new direction of using numerical PDE techniques for PDE identification,
2. proposing a flexible approach which can handle different boundary conditions, are robust against noise, and can identify nonlinear PDEs with varying coefficients,
3. establishing a recovery theory of Lasso, which leads to a new definition of the noise-to-signal ratio for PDE identification,
4. systematically analyzing the effects of noise and downsampling, and proposing a new denoising method called Least Square Moving Average (LSMA).

This paper is organized as follows: The main algorithm is presented in Sect. 2. Specifically, the set-up of the problem is presented in Sect. 2.2; details of the IDENT algorithm are in Sect. 2.3; a recovery theory for Lasso and the new noise-to-signal ratio are given in Sect. 2.4; and the first set of numerical experiments are in Sect. 2.5. The aspects of denoising and downsampling effects are presented in Sect. 3. In Sect. 3, the LSMA denoising method is introduced in Sect. 3.1; numerical experiments for noisy data are given in Sect. 3.2, and downsampling effects are considered in Sect. 3.3. The identification of PDEs with varying coefficients are presented in Sect. 4. In Sect. 4, we consider nonlinear PDEs with varying coefficients and introduce BEE motivated by finite element approximation. A concluding remark is given in Sect. 5 and some details are in the "Appendix".

2 Identifying Differential Equations with Numerical Time Evolution (IDENT)

2.1 Notations

We use bold letter to denote vectors, such as $\mathbf{a}$, $\mathbf{b}$. The support of a vector $\mathbf{x}$ is the set of indices at which the corresponding entry is nonzero: $\text{supp}(\mathbf{x}) := \{j : x_j \neq 0\}$. We use A^T and A^* to denote the transpose and the conjugate transpose of the matrix A. We use $x \to \varepsilon^+$ to denote $x > \varepsilon$ and $x \to \varepsilon$. Let $\mathbf{f} = \{f(x_i, t_n) | i = 1, \ldots, N_1, n = 1, \ldots, N_2\} \in \mathbb{R}^{N_1 N_2}$ be samples of a function $f : \mathcal{D} \times [0, \infty) \to \mathbb{R}$ with spatial spacing Δx and time spacing Δt. The integers N_1 and N_2 are the total number of spatial and time discretization respectively. We assume PDEs are simulated on the grid with time spacing δt and spatial spacing δx, while data are sampled on the grid with time spacing Δt and spatial spacing Δx. The vector L^p norm of $\mathbf{f}$ is $\|\mathbf{f}\|_p = (\sum_{i=1}^{N_1} \sum_{n=1}^{N_2} |f(x_i, t_n)|^p)^{1/p}$. Denote $\|\mathbf{f}\| = \|\mathbf{f}\|_2$. The function L^p norm of $\mathbf{f}$ is $\|\mathbf{f}\|_{L^p} = (\sum_{i=1}^{N_1} \sum_{n=1}^{N_2} |f(x_i, t_n)|^p \Delta x \Delta t)^{1/p}$. Notice that $\|\mathbf{f}\|_{L^p} = \|\mathbf{f} \Delta x^{1/p} \Delta t^{1/p}\|_p$.

2.2 The Set-Up of the Problem

We consider parametric PDEs where $\mathcal{F}(x, u, u_x, u_{xx})$ is a linear combination of monomials such as $1, u, u^2, u_x, u_x^2, uu_x, u_{xx}, u_{xx}^2, uu_{xx}, u_x u_{xx}$ with coefficients $\mathbf{a} = \{a_j\}_{j=1}^{10}$:

$$u_t = a_1 + a_2 u + a_3 u^2 + a_4 u_x + a_5 u_x^2 + a_6 u u_x + a_7 u_{xx} + a_8 u_{xx}^2 + a_9 u u_{xx} + a_{10} u_x u_{xx}.$$

(1)

We refer to each monomial as a feature, and let N_3 be the number of features, i.e., $N_3 = 10$ in (1). The right hand side can be viewed as a second-order Taylor expansion of $\mathcal{F}(u, u_x, u_{xx})$. It can easily be generalized to higher-order Taylor expansions, and operators $\mathcal{F}(u, u_x, u_{xx}, u_{xxx}, \partial_x^4 u, \ldots)$ depending on higher order derivatives. This model contains a rich class of differential equations, e.g., the heat equation, transport equation, Burgers' equation, KdV equation, Fisher's equation that models gene propagation.

Evaluating (1) at discrete time and space $(x_i, t_n), i = 1, \ldots, N_1, n = 1, \ldots, N_2$ yields the discrete linear system

$$F\mathbf{a} = \mathbf{b},$$

where

$$\mathbf{b} = \{u_t(x_i, t_n) | i = 1, \ldots, N_1, n = 1, \ldots, N_2\} \in \mathbb{R}^{N_1 N_2},$$

and F is a $N_1 N_2 \times N_3$ feature matrix in the form of

$$F = \begin{pmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & u(x_i, t_n) & u^2(x_i, t_n) & u_x(x_i, t_n) & u_x^2(x_i, t_n) & uu_x(x_i, t_n) & u_{xx}(x_i, t_n) & u_{xx}^2(x_i, t_n) & uu_{xx}(x_i, t_n) & u_x u_{xx}(x_i, t_n) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}. \tag{2}$$

We use $F[j]$ to denote the jth column associated with the jth feature evaluated at (x_i, t_n), $i = 1, \ldots, N_1, n = 1, \ldots, N_2$

The **objective** of PDE identification is to recover the unknown coefficient vector $\mathbf{a} \in \mathbb{R}^{N_3}$ from the given data. Real world physical processes are often presented with a few number of features in the right hand side of (1), so it is reasonable to assume that the coefficients are sparse.

For differential equations with varying coefficients, we consider PDEs of the form

$$\begin{aligned} u_t = {} & a_1(x) + a_2(x)u + a_3(x)u^2 + a_4(x)u_x + a_5(x)u_x^2 + a_6(x)uu_x \\ & + a_7(x)u_{xx} + a_8(x)u_{xx}^2 + a_9(x)uu_{xx} + a_{10}(x)u_x u_{xx} \end{aligned} \tag{3}$$

where each $a_j(x)$ is a function on the spatial domain of the PDE. We expand the coefficients in terms of finite element bases $\{\phi_l\}_{l=1}^L$ such that

$$a_j(x) \approx \sum_{l=1}^{L} a_{j,l}\phi_l(x) \text{ for } j = 1, \ldots, N_3, \tag{4}$$

where L is the number of finite element bases used to approximate $a_j(x)$. Let $y_1 < y_2 < \cdots < y_L$ be a partition of the spatial domain. We use a typical finite element basis function, e.g., $\phi_l(x)$ is continuous, and linear within each subinterval (y_i, y_{i+1}), and $\phi_l(y_i) = \delta_{li} = 1$ if $i = l$; 0 otherwise. If the $a_j(x)$'s are Lipchitz functions, and finite element bases are defined on a grid with spacing $O(1/L)$. The approximation error of the $a_j(x)$'s satisfies

$$\|a_j - \sum_{l=1}^{L} a_{j,l}\phi_l\|_{L^p} \leq O(1/L), \ p \in (0, \infty). \tag{5}$$

In the case of varying coefficients, the feature matrix F is of size $N_1N_2 \times N_3L$,

$$F = \left(\begin{array}{ccc|ccc|c|ccc} \vdots & & \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ \phi_1(x_i) & \dots & \phi_L(x_i) & u(x_i,t_n)\phi_1(x_i) & \dots & u(x_i,t_n)\phi_L(x_i) & \dots & u_xu_{xx}(x_i,t_n)\phi_1(x_i) & \dots & u_xu_{xx}(x_i,t_n)\phi_L(x_i) \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots & & \vdots \end{array} \right), \tag{6}$$

and the vector to be identified is

$$\mathbf{a} = \left(a_{1,1}, \dots, a_{1,L} | a_{2,1}, \dots, a_{2,L} | \dots | a_{N_3,1}, \dots, a_{N_3,L}\right)^T \in \mathrm{R}^{N_3L}.$$

The feature matrix F has a block structure. We use $F[j,l]$ to denote the column of F associated with the jth feature and the lth basis. To be clear, $F[j]$ is the jth column of (2), and $F[j,l]$ is the $(j-1)L+l$th column of (6). Evaluating (3) at (x_i,t_n), $i=1,\dots,N_1$, $n=1,\dots,N_2$ yields the discrete linear system

$$F\mathbf{a} = \mathbf{b} + \boldsymbol{\eta},$$

where $\boldsymbol{\eta} = \{\eta(x_i,t_n) | i=1,\dots,N_1, n=1,\dots,N_2\} \in \mathbb{R}^{N_1N_2}$ represents the approximation error of the $a_j(x)$'s by finite element bases such that

$$\eta(x_i,t_n) = \left(\sum_{l=1}^{L} a_{1,l}\phi_l(x_i) - a_1(x_i)\right) + \dots + \left(\sum_{l=1}^{L} a_{10,l}\phi_l(x_i) - a_{10}(x_i)\right) u_xu_{xx}(x_i,t_n).$$

In the case that u, u_x, u_{xx} are uniformly bounded,

$$\|\boldsymbol{\eta}\|_{L^p} \le O(1/L), \ \ p \in (0,\infty),$$

and $\boldsymbol{\eta} = 0$ when all coefficients are constants.

2.3 The Proposed Algorithm: IDENT

In this paper, we assume that only the discrete data $\{u_i^n | i=1,\dots,N_1 \text{ and } n=1,\dots,N_2\}$ and the boundary conditions are given. If data are perfectly generated and there is no measurement noise, $u_i^n = u(x_i,t_n)$ for every i and n, and we outline the proposed IDENT algorithm in this section assuming the given data do not have noise.

The first step of IDENT is to construct the empirical version of the feature matrix F and the vector $\mathbf{b}$ containing time derivatives from the given data. The derivatives are approximated by finite difference methods which gives flexibility in dealing with different types of PDEs and boundary conditions (e.g. non-periodic). We approximate the time derivative u_t by a first-order backward difference scheme:

$$u_t(x_j,t_k) \approx \widehat{u_t}(x_j,t_n) := \frac{u(x_j,t_n) - u(x_j,t_{n-1})}{\Delta t},$$

which yields the error

$$\widehat{u_t}(x_j,t_n) = u_t(x_j,t_n) + O(\Delta t).$$

Let $\widehat{\mathbf{b}}$ be the empirical version of $\mathbf{b}$ constructed from data:

$$\widehat{\mathbf{b}} = \{\widehat{u_t}(x_i,t_n) : i=1,\dots,N_1, n=1,\dots,N_2\} \in \mathbb{R}^{N_1N_2}.$$

We approximate the spatial derivative u_x through the five-point ENO method proposed by Harten et al. [11]. Let $\widehat{u_x}(x_j, t_n)$ and $\widehat{u_{xx}}(x_j, t_n)$ be approximations of $u_x(x_j, t_n)$ and $u_{xx}(x_j, t_n)$ by the five-point ENO method which yields the error:

$$\widehat{u_x}(x_j, t_n) = u_x(x_j, t_n) + O(\Delta x^4), \quad \widehat{u_{xx}}(x_j, t_n) = u_{xx}(x_j, t_n) + O(\Delta x^3).$$

Putting the $\widehat{u_x}(x_j, t_n)$'s and $\widehat{u_{xx}}(x_j, t_n)$'s to the feature matrix F in (6) gives rise to the empirical feature matrix, denoted by $\widehat{F}$. For example, the second column of $\widehat{F}$ is given by $\{u_i^n | i = 1, \ldots, N_1 \text{ and } n = 1, \ldots, N_2\}$ as an approximation of $\{u(x_i, t_n) | i = 1, \ldots, N_1 \text{ and } n = 1, \ldots, N_2\}$ as follows

$$(u_1^1, u_2^1, \ldots, u_{N_1}^1, u_1^2, \ldots, u_{N_1}^2, \ldots, u_1^{N_2}, \ldots, u_{N_1}^{N_2})^T \in \mathrm{R}^{N_1 N_2}.$$

These empirical quantities give rise to the linear system

$$\widehat{F}\mathbf{a} = \widehat{\mathbf{b}} + \mathbf{e}, \quad \mathbf{e} = \mathbf{b} - \widehat{\mathbf{b}} + (\widehat{F} - F)\mathbf{a} + \boldsymbol{\eta}, \tag{7}$$

where the terms $\mathbf{b} - \widehat{\mathbf{b}}$, $(\widehat{F} - F)\mathbf{a}$ and $\boldsymbol{\eta}$ arise from errors in approximating time and spatial derivatives, and the finite element expansion of varying coefficients, respectively. The total error $\mathbf{e}$ satisfies

$$\|\mathbf{e}\|_{L^2} \leq \varepsilon \text{ such that } \varepsilon = O(\Delta t + \Delta x^3 + 1/L). \tag{8}$$

The second step is to find possible candidates for the non-zero coefficients of $\mathbf{a}$. We utilize L^1-regularized minimization, also known as Lasso [30] or group Lasso [34], solved by Alternating Direction Method of Multipliers [2,5] to get a sparse or block-sparse vector. We minimize the following energy:

$$\widehat{\mathbf{a}}_{\text{G-Lasso}}(\lambda) = \arg\min_{\mathbf{z}} \left\{ \frac{1}{2}\|\widehat{\mathbf{b}} - \widehat{F}_\infty \mathbf{z}\|_2^2 + \lambda \sum_{j=1}^{N_3} \left(\sum_{l=1}^{L} |z_{j,l}|^2 \right)^{\frac{1}{2}} \right\}, \tag{9}$$

where λ is a balancing parameter between the first fitting term and the second regularization term. The matrix $\widehat{F}_\infty$ is obtained from $\widehat{F}$ with each column divided by the maximum magnitude of the column, namely, $\widehat{F}_\infty[j, l] = \widehat{F}[j, l]/\|\widehat{F}[j, l]\|_\infty$. We use Lasso for the constant coefficient case where $L = 1$, and group Lasso for the varying coefficient case $L > 1$. A set of possible active features is selected by thresholding the normalized coefficient magnitudes:

$$\widehat{\Lambda}_\tau := \left\{ j : \|\widehat{F}[j]\|_{L^1} \left\| \sum_{l=1}^{L} \frac{\widehat{\mathbf{a}}_{\text{G-Lasso}}(\lambda)_{j,l}}{\|\widehat{F}[j, l]\|_\infty} \phi_l \right\|_{L^1} \geq \tau \right\}. \tag{10}$$

with a fixed thresholding parameter $\tau \geq 0$.

The final step is to identify the correct support using the Time Evolution Error (TEE). (i) From the candidate coefficient index set $\widehat{\Lambda}_\tau$, consider every subset $\Omega \subseteq \widehat{\Lambda}_\tau$. For each $\Omega = \{j_1, j_2, \ldots, j_k\}$, find the coefficients $\widehat{\mathbf{a}} = (0, 0, \widehat{a}_{j_1}, 0, \ldots, \widehat{a}_{j_k}, \ldots)$ by a least-square fit such that $\widehat{\mathbf{a}}_\Omega = \widehat{F}_\Omega^\dagger \widehat{\mathbf{b}}$ and $\widehat{\mathbf{a}}_{\Omega^\complement} = \mathbf{0}$. (ii) Using these coefficients, we construct the differential equation and numerically time evolve

$$u_t = \mathcal{F}\widehat{\mathbf{a}},$$

starting from the given initial data, for each Ω. It is crucial to use a smaller time step $\widetilde{\Delta t} \ll \Delta t$, where Δt is the time spacing of the given data. TEE follows the principle of numerical convergence that the numerical solution will converge to the true PDE as the time step goes to zero. We use the first-order Euler scheme for time evolution with time step

Algorithm 1 Identifying Differential Equations with Numerical Time evolution (IDENT)

Input: The discrete data $\{u_i^n | i = 1, \ldots, N_1 \text{ and } n = 1, \ldots, N_2\}$.
[Step 1] Construct the empirical feature matrix $\widehat{F}$ and the empirical vector $\widehat{\mathbf{b}}$ using ENO schemes.
[Step 2] Find a set of possible active features by the L^1 minimization (9) followed by thresholding.
[Step 3] Pick the coefficient vector $\widehat{\mathbf{a}}$, with the minimum Time Evolution Error (TEE).
Output: The identified coefficients $\widehat{\mathbf{a}}$ where $\widehat{\mathbf{a}}_{\widehat{\Lambda}} = \widehat{F}_{\widehat{\Lambda}}^{\dagger}\widehat{\mathbf{b}}$.

$\widetilde{\Delta t} = O(\Delta x^r)$ where r is the highest order of the spatial derivatives in the PDE associated with $\widehat{\mathbf{a}}$. When the candidate PDE contains high-order spatial derivatives, this is necessary to stabilize the explicit scheme used for the TEE evolution. (iii) Finally, calculate the time evolution error for each $\widehat{\mathbf{a}}$:

$$\text{TEE}(\widehat{\mathbf{a}}) := \sum_{i=1}^{N_1}\sum_{n=1}^{N_2} |\bar{u}_i^n - u_i^n| \Delta x \Delta t,$$

where $\bar{u}_i^n$ is the numerically time evolved solution at (x_i, t_n) of the PDE with the coefficient $\widehat{\mathbf{a}}$. We pick the subset Ω and the corresponding coefficients $\widehat{\mathbf{a}}$, which give the smallest TEE, and denote the recovered support as $\widehat{\Lambda}$. This is the output of the algorithm, which is the identified PDE. Algorithm 1 summarizes this procedure.

We note that it is possible to skip the L^1 minimization step, and use TEE to recover the support of coefficients by considering all possible combinations from the beginning, however, the computational cost is very high. The L^1 minimization helps to reduce the number of combinatorial trials, and make IDENT more computationally efficient. On the other hand, while L^1 minimization is effective in finding a sparse vector, L^1 alone is often not enough: (i) Zero coefficients in the true PDE may become non-zero in the minimizer of L^1. (ii) If active terms are chosen by a thresholding, results are sensitive to the choice of thresholding parameter, e.g., τ in (10). (iii) The balancing parameter λ can affect the results. (iv) If some columns of the empirical feature matrix $\widehat{F}$ are highly correlated, Lasso is known to have a larger support than the ground truth [9]. TEE refines the results from Lasso, and relaxes the dependence on the parameters.

There are two fundamental ideas behind TEE:

1. For nonlinear PDEs, it is impossible to isolate each term separately to identify each coefficient. Any realization of PDE must be understood as a set of terms.
2. If the underlying dynamics are identified by the true PDE, any refinement in the discretization of the time domain should not deviate from the given data. This is the fundamental principle of consistency, convergence and stability of a numerical scheme, and the reason for choosing a smaller time step $\widetilde{\Delta t} \ll \Delta t$.

Therefore, the main effect of TEE is to evolve the numerical error from the wrongly identified differential terms. This method can be applied to linear or nonlinear PDEs. The effectiveness of TEE can be demonstrated with an example. Assume that the solution u is smooth and decays sufficiently fast at infinity, and consider the following linear equation with constant coefficients:

$$\frac{\partial u}{\partial t} = a_0 u + a_1 \frac{\partial u}{\partial x} + \cdots + a_m \frac{\partial^m u}{\partial x^m}.$$

After taking the Fourier transform for the equation and solving the ODE, one can obtain the transformed solution:

$$\hat{u}(\xi, t) = \hat{u}(\xi, 0) e^{a_0 t} e^{a_1 \mathbf{i}\xi t} e^{-a_2 \xi^2 t} \cdots e^{a_m (\mathbf{i}\xi)^m t},$$

where $\mathbf{i} = \sqrt{-1}$ and ξ is the variable in the Fourier domain. If a term with an even-order derivative, such as $a_2 \frac{\partial^2 u}{\partial x^2}$, is mistakenly included in the PDE, it will make every frequency mode grow or decrease exponentially in time; if a term with an odd-order derivative, such as $a_1 \frac{\partial u}{\partial x}$, is mistakenly included in the solution, it will introduce a wrong-speed oscillation of the solution. In either case, the deviation from the correct solution grows fast in time, providing an efficient way to distinguish the wrong terms. Our numerical experiments show that TEE is an effective tool to correctly identify the coefficients. Our first set of experiments are presented in Sect. 2.5.

2.4 Recovery Theory of Lasso, and New Noise-to-Signal Ratio (NSR)

In this subsection, we establish a performance guarantee of Lasso for the identification of PDEs with constant coefficients. In the Step 2 of IDENT, Lasso is applied as L^1 regularization in (9). We consider the incoherence property proposed in [8], and follow the ideas in [10,32,33] to establish a recovery theory. While the details of the proof is presented in Appendix 1, here we state the result which leads to a new definition of noise-to-signal ratio.

For PDEs with constant coefficients, we set $L = 1$ in (4), and consider the standard Lasso:

$$\widehat{\mathbf{a}}_{\text{Lasso}}(\lambda) = \arg\min_{\mathbf{z}} \left\{ \frac{1}{2} \|\widehat{\mathbf{b}} - \widehat{F}_\infty \mathbf{z}\|_2^2 + \lambda \|\mathbf{z}\|_1 \right\}. \tag{Lasso}$$

If all columns of $\widehat{F}$ are uncorrelated, $\mathbf{a}$ can be robustly recovered by Lasso. Let $\widehat{F} = [\widehat{F}[1]\ \widehat{F}[2]\ \ldots\ \widehat{F}[N_3]]$ where $\widehat{F}[j]$ stands for the jth column of $\widehat{F}$ in (2). To measure the correlation between the jth and the lth column of $\widehat{F}$, we use the pairwise coherence

$$\mu_{j,l}(\widehat{F}) = \frac{|\langle \widehat{F}[j], \widehat{F}[l] \rangle|}{\|\widehat{F}[j]\|_2 \|\widehat{F}[l]\|_2}$$

and the mutual coherence of $\widehat{F}$ as in [8]:

$$\mu(\widehat{F}) = \max_{j \neq l} \mu_{j,l}(\widehat{F}) = \max_{j \neq l} \frac{|\langle \widehat{F}[j], \widehat{F}[l] \rangle|}{\|\widehat{F}[j]\|_2 \|\widehat{F}[l]\|_2}.$$

Since normalization does not affect the coherence, we have $\mu_{j,l}(\widehat{F}_\infty) = \mu_{j,l}(\widehat{F})$ and $\mu(\widehat{F}_\infty) = \mu(\widehat{F})$. The smaller $\mu(\widehat{F})$, the less correlated are the columns of $\widehat{F}$, and $\mu(\widehat{F}) = 0$ if and only if the columns are orthogonal. Lasso will recover the correct coefficients if $\mu(\widehat{F})$ is sufficiently small.

Theorem 1 *Let* $\mu = \mu(\widehat{F})$, $w_{\max} = \max_j \|\widehat{F}[j]\|_\infty \|\widehat{F}[j]\|_{L^2}^{-1}$ *and* $w_{\min} = \min_j \|\widehat{F}[j]\|_\infty \|\widehat{F}[j]\|_{L^2}^{-1}$. *Suppose the support of* $\mathbf{a}$ *contains no more than* s *indices,* $\mu(s-1) < 1$ *and*

$$\frac{\mu s}{1 - \mu(s-1)} < \frac{w_{\min}}{w_{\max}}.$$

Let

$$\lambda = \frac{[1 - (s-1)\mu]}{w_{\min}[1 - \mu(s-1)] - w_{\max}\mu s} \cdot \frac{\varepsilon^+}{\Delta x \Delta t}. \tag{11}$$

Then

1) the support of $\widehat{\mathbf{a}}_{Lasso}(\lambda)$ *is contained in the support of* $\mathbf{a}$*;*

2) the distance between $\widehat{\mathbf{a}}_{Lasso}(\lambda)$ *and* $\mathbf{a}$ *satisfies*

$$\max_j \|\widehat{F}[j]\|_{L^2} \left| \|\widehat{F}[j]\|_\infty^{-1} \widehat{\mathbf{a}}_{Lasso}(\lambda)_j - a_j \right| \leq \frac{w_{\max} + \varepsilon/\sqrt{\Delta t \Delta x}}{w_{\min}[1-\mu(s-1)] - w_{\max}\mu s}\varepsilon; \quad (12)$$

3) if

$$\min_{j:\, a_j \neq 0} \|\widehat{F}[j]\|_{L^2} |a_j| > \frac{w_{\max} + \varepsilon/\sqrt{\Delta t \Delta x}}{w_{\min}[1-\mu(s-1)] - w_{\max}\mu s}\varepsilon, \quad (13)$$

then the support of $\widehat{\mathbf{a}}_{Lasso}(\lambda)$ *is exactly the same as the support of* $\mathbf{a}$.

Theorem 1 shows that Lasso will give rise to the correct support when the empirical feature matrix $\widehat{F}$ is incoherent, i.e. $\mu(\widehat{F}) \ll 1$, and all underlying coefficients are sufficiently large compared to noise. When the empirical feature matrix is coherent, i.e., some columns of $\widehat{F}$ are correlated, it has been observed that $\widehat{\mathbf{a}}_{\text{Lasso}}(\lambda)$ are usually supported on supp($\mathbf{a}$) and the indices that are highly correlated with supp($\mathbf{a}$) [9]. We select possible features by thresholding in (10) which is equivalent to $\widehat{\Lambda}_\tau := \left\{ j : \|\widehat{F}[j]\|_{L^1} \|\widehat{F}[j]\|_\infty^{-1} |\widehat{\mathbf{a}}_{\text{Lasso}}(\lambda)_j| \geq \tau \right\}$ in the case of constant coefficients. After this, TEE is an effective tool in complement of Lasso to distinguish the correct features from the wrong ones. The details of Theorem 1 can be found in Appendix 1.

This analysis also gives rise to a **new noise-to-signal ratio**:

$$\text{Noise-to-Signal Ratio (NSR)} := \frac{\|\widehat{F}\mathbf{a} - \widehat{\mathbf{b}}\|_{L^2}}{\min_{j:\, a_j \neq 0} \|\widehat{F}[j]\|_{L^2} |a_j|}. \quad (14)$$

The definition is derived from (13), showing that the signal level is contributed by the minimum of the product of the coefficient and the column norm in the feature matrix - $\min_{j:\, a_j \neq 0} \|\widehat{F}[j]\|_{L^2} |a_j|$. This term represents the dynamics resulted from the feature. It is important to consider the multiplication rather than the magnitude of the coefficient only. We also use this new definition of NSR to measure the level of noise in the following sections, which gives a more consistent representation.

2.5 First Set of IDENT Experiments

We present the first set of numerical experiments to illustrate the effects of IDENT. Here data are sampled from exact or simulated solutions of PDEs with constant coefficients. For boundary conditions, we use zero Dirichlet boundary conditions throughout the paper. Modification to periodic or other boundary conditions is trivial, and numerical schemes with periodic boundary conditions can achieve a higher accuracy, for the cases without noise. We observe that the Lasso results are not very sensitive to the choice of λ using TEE, and we set $\lambda = 500$ in all experiments.

The first experiment is on the Burgers' equation with Dirichlet boundary conditions:

$$\begin{aligned} &u_t + \left(\frac{u^2}{2}\right)_x = 0, \ x \in [0,1] \\ &u(x,0) = \sin 4\pi x \text{ and } u(0,t) = u(1,t) = 0. \end{aligned} \quad (15)$$

The given data are sampled from the true analytic solution, shown in Fig. 1a, with $\Delta x = 1/56$ and $\Delta t = 0.004$, for $t \in [0, 0.05]$. Figure 1b displays the coherence pattern of the empirical feature matrix: the absolute values of $\widehat{F}^*_{\text{unit}} \widehat{F}_{\text{unit}}$ where $\widehat{F}_{\text{unit}}$ is obtained

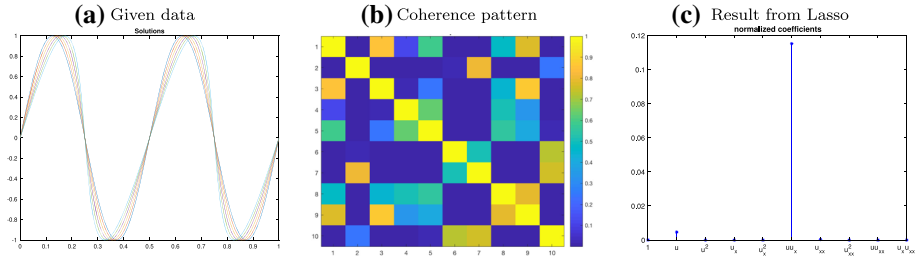


Fig. 1 Experiment with the Burgers' equation (15). **a** The given data are sampled from the true analytic solution. **b** The coherence pattern of $\widehat{F}$. **c** Normalized coefficient magnitudes from Lasso. Two possible features are identified, which are u and uu_x

from $\widehat{F}$ with column normalized to unit L^2 norm. This pattern shows the correlation between any pair of the columns in $\widehat{F}$. (c) shows the normalized coefficient magnitudes $\{\|\widehat{F}[j]\|_{L^1}\|\widehat{F}[j]\|_\infty^{-1}|\widehat{\mathbf{a}}_{\text{Lasso}}(\lambda)_j|\}$ after L^1 minimization. The magnitudes of u and uu_x are not negligible, so they are picked as a possible set of active features in $\widehat{\Lambda}_\tau$. Then, TEE is computed for all subsets $\Omega \subseteq \widehat{\Lambda}_\tau$, i.e., $u_t = au$, $u_t = buu_x$ and $u_t = cu + duu_x$ where the coefficients a, b, c, d are calculated by least-squares:

Active terms	Coefficients of active terms by least-squares	TEE
u	0.27	78.76
uu_x	-0.99	0.48
$[u\ uu_x]$	$[0.10\ \ -0.99]$	1.40

The red line with only uu_x term has the smallest TEE, and therefore is identified as the result of IDENT. Since the true PDE is $u_t = -uu_x$, the computed result shows a small coefficient error.

The second experiment is on the Burgers' equation with a diffusion term:

$$\begin{aligned} &u_t + \left(\frac{u^2}{2}\right)_x = 0.1u_{xx},\ x \in [0, 1] \\ &u(x, 0) = \sin 4\pi x \text{ and } u(0, t) = u(1, t) = 0. \end{aligned} \tag{16}$$

The given data are simulated with a first-order explicit method where $\delta x = 1/256$ and $\delta t = (\delta x)^2$ for $t \in [0, 0.1]$. Data are downsampled from the numerical simulation by a factor of 4 such that $\Delta x = 4\delta x$ and $\Delta t = 4\delta t$. (We explore the effects of downsampling in more detail in Sect. 3.)

Figure 2a shows the given data, (b) displays the coherence pattern of $\widehat{F}$, and (c) shows the normalized coefficient magnitudes $\{\|\widehat{F}[j]\|_{L^1}\|\widehat{F}[j]\|_\infty^{-1}|\widehat{\mathbf{a}}_{\text{Lasso}}(\lambda)_j|\}$. In this case, the coherence pattern in (b) shows that u and u_xu_{xx} are highly correlated with u_{xx} and uu_x, respectively, and therefore all four terms $u, uu_x, u_{xx}, u_xu_{xx}$ are identified as meaningful ones by Lasso in (c). Considering TEE for each subset refines these results:

The red line is the result of IDENT, while the blue line is the ground truth. The TEE of $[uu_x\ u_{xx}\ u_xu_{xx}]$ is the smallest, which is comparable with the TEE of the true equation with $[uu_x\ u_{xx}]$. One wrongly identified term in red, u_xu_{xx}, has a coefficient magnitude of

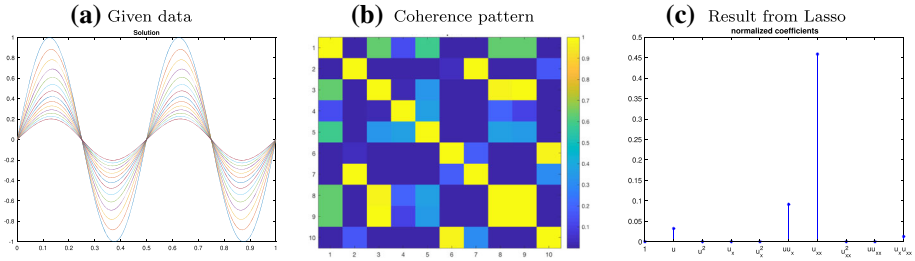


Fig. 2 Experiment with Burgers' equation with a diffusion term (16). **a** The given data are numerically simulated and downsampled. **b** shows that u and $u_x u_{xx}$ are highly correlated with u_{xx} and uu_x, respectively. From (**c**), four terms u, uu_x, u_{xx} and $u_x u_{xx}$ are selected for TEE

Active terms	Coefficients of active terms by least-squares	TEE
u	-16.08	3709.77
uu_x	-0.34	67092.21
u_{xx}	0.10	4345.98
$u_x u_{xx}$	-0.0008	∞
$[u\ uu_x]$	$[-16.17\ \ -0.50]$	2120.14
$[u\ u_{xx}]$	$[-21.42\ \ -0.03]$	1.49×10^{26}
$[u\ u_x u_{xx}]$	$[-16.44\ \ -0.003]$	∞
$[uu_x\ u_{xx}]$	$[-1.00\ 0.10]$	82.33
$[uu_x\ u_x u_{xx}]$	$[-12.03\ \ -0.07]$	∞
$[u_{xx}\ u_x u_{xx}]$	$[-0.10\ \ -0.006]$	371.08
$[u\ uu_x\ u_{xx}]$	$[-0.10\ \ -1.00\ 0.10]$	83.73
$[u\ uu_x\ u_x u_{xx}]$	$[-15.86\ \ -1.03\ \ -0.003]$	∞
$[u\ u_{xx}\ u_x u_{xx}]$	$[-0.58\ 0.10\ 0.006]$	367.68
$[uu_x\ u_{xx}\ u_x u_{xx}]$	$[-1.00\ 0.10\ \ -1.35 \times 10^{-5}]$	82.29
$[u\ uu_x\ u_{xx}\ u_x u_{xx}]$	$[-0.11\ \ -1.00\ 0.10\ \ -2.8 \times 10^{-5}]$	83.85

-1.35×10^{-5} which is negligible. The level of error in the identification is also related to the total error to be explored in (17). Without TEE, if all four terms are used from L^1 minimization, an additional wrong term u is identified with the coefficient -0.11. This is comparable to other terms with coefficients like -1 or 0.1, and cannot be ignored.

Theorem 1 proves that the identified coefficients from Lasso will converge to the ground truth as $\Delta t \to 0$ and $\Delta x \to 0$ (see Eq. (8) and (12)), when there is no noise and the empirical feature matrix has a small coherence. Figure 3 shows the recovered coefficients from Lasso versus Δt and Δx for the Burgers' equation (15) and Burgers' equation with diffusion (16). In Fig. 3a, data are sampled from the analytic solution of the Burgers' equation (15) with spacing $\Delta x = 2^k$ for $k = -12, -11, \ldots, -5$ respectively and $\Delta t = \Delta x$ for $t \in [0, 0.05]$. Figure 3a shows the result from Lasso, namely, $\{\|\widehat{F}[j]\|_\infty^{-1} \widehat{\mathbf{a}}_{\text{Lasso}}(\lambda)_j\}$, versus $\log_2 \Delta x$. Notice that the coefficient of uu_x converges to -1 and all other coefficients converge to 0 as Δt and Δx decrease. For Fig. 3b, data are sampled from the numerical simulation of the Burgers' equation with diffusion in (16), where the PDE is solved by a first-order method with $\delta x = 2^{-10}$ and $\delta t = (\delta x)^2$ for $t \in [0, 0.1]$. Data are sampled with $\Delta x = 2^{-10}, 2^{-9}, 2^{-8}, 2^{-7}, 2^{-6}$ respectively, and $\Delta t = (\Delta x)^2$ for $t \in [0, 0.1]$. Figure 3b shows the recovered coefficients from Lasso versus $\log_2 \Delta x$. Here the coefficients of uu_x and u_{xx} converge to -1 and 0.1

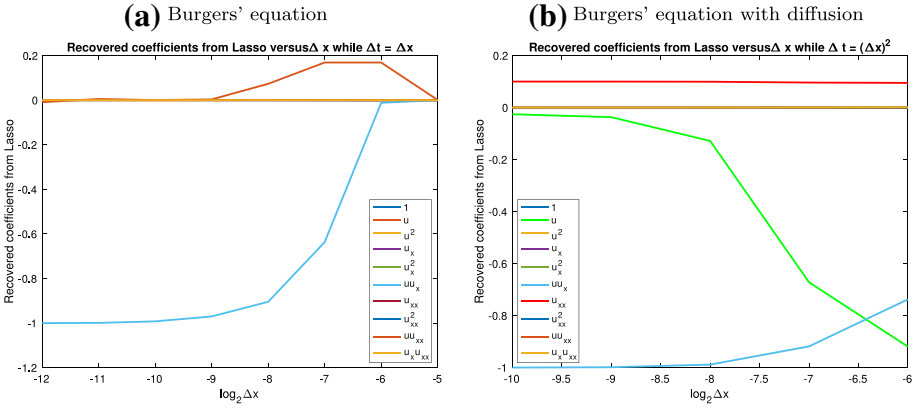


Fig. 3 Identified coefficients from Lasso (Step 2 only) versus $\log_2 \Delta x$. In (**a**), as Δt and Δx decrease (from right to left), the coefficient of uu_x correctly converges to 1, and all other terms correctly converge to 0. In (**b**), as Δt and Δx decrease (from right to left), while the coefficients of u_{xx} and uu_x correctly converge to 0.1 and -1 respectively, one wrong term u does not converge to 0, due to the error from data generations

respectively, and all other coefficients except the one of u, converge to 0, as Δt and Δx decrease. The coefficient of u does not converge to 0 because data are generated by a first order numerical scheme with the error $O[\delta t + (\delta x)^2]$, i.e., the error for Lasso $\|\mathbf{e}\|_{L^2}$ does not decay to 0 as Δt and Δx decrease. We further discuss this aspect of data generations in Sect. 3.

3 Noisy Data, Downsampling and IDENT

As noticed above, identification results depend on the accuracy of the given data. In this section, we explore the effects of inaccuracy in data generations, noise and downsampling. We derive an error formula to incorporate the errors arising from these three aspects, which provides a theoretical guidance of the difficulty of identification.

The given data $\{\widetilde{u}_i^n\}$ may contain noise, such that

$$\widetilde{u}_i^n = u_i^n + \xi_i^n,$$

where the noise ξ_i^n arises from inaccuracy in data generations and/or the measurement error. Consider a rth order PDE with the highest-order spatial derivative $\partial_x^r u$. Suppose data are numerically simulated by a qth-order method with time step δt and spatial spacing δx, and the measurement error is independently drawn from the normal distribution with mean 0 and variance σ^2. Then

$$\xi_i^n = O(\delta t + \delta x^q + \sigma).$$

We use the five-point ENO method to approximate the spatial derivatives in the empirical feature matrix $\widehat{F}$ in Sect. 2. In general, one could interpolate the data with a pth-order polynomial and use the derivatives of the polynomial to approximate u_x and u_{xx}, etc. In this case, the error for the kth-order spatial derivative $\partial_x^k u$ is $O(\Delta x^{p+1-k})$.

The error for Lasso given by (7) is $\mathbf{e} = \mathbf{b} - \widehat{\mathbf{b}} + (\widehat{F} - F)\mathbf{a} + \boldsymbol{\eta}$, where $\mathbf{b} - \widehat{\mathbf{b}}$ is from the approximation of u_t, $(\widehat{F} - F)\mathbf{a}$ is from the approximation of the spatial derivatives of u, and $\boldsymbol{\eta}$ arrises from the finite element basis expansion for the varying coefficients. If $u, u_x, u_{xx}, \ldots$

are bounded, these terms satisfy

$$\|(\widehat{F} - F)\mathbf{a}\|_\infty \le O\left(\Delta x^{p+1-r} + \frac{\delta t + \delta x^q + \sigma}{\Delta x^r}\right) \text{ and}$$
$$\|\mathbf{b} - \widehat{\mathbf{b}}\|_\infty \le O\left(\Delta t + \frac{\delta t + \delta x^q + \sigma}{\Delta t}\right),$$

and $\|\boldsymbol{\eta}\|_\infty = O(1/L)$ so that

$$\|\mathbf{e}\|_{L^2} \le \varepsilon, \text{ with } \varepsilon = O\left(\Delta t + \Delta x^{p+1-r} + \underbrace{\frac{\delta t + \delta x^q}{\Delta t} + \frac{\delta t + \delta x^q}{\Delta x^r}}_{\text{errors from data generations}} + \underbrace{\frac{\sigma}{\Delta t} + \frac{\sigma}{\Delta x^r}}_{\text{measurement noise}} + \frac{1}{L}\right). \quad (17)$$

This error formula suggests the followings:

Sensitivity to measurement noise: Finite differences are sensitive to measurement noise since Gaussian noise with mean 0 and variance σ^2 results in $O(\sigma/\Delta t + \sigma/\Delta x^r)$ in the error formula. Higher-order PDEs are more sensitive to measurement noise than lower-order PDEs. Denoising the given data is helpful to Lasso in general.

Downsampling of data: In applications, the given data are downsampled such that $\Delta t = C_t \delta t$ and $\Delta x = C_x \delta x$ where C_t and C_x are the downsampling factors in time and space. Downsampling could help to reduce the error depending on the balances among the terms in (17).

We further explore these effects below. We propose an order preserving denoising method in Sect. 3.1, experiment IDENT with noisy data in Sect. 3.2, and discuss the downsampling of data in Sect. 3.3.

3.1 An Order Preserving Denoising Method: Least-Squares Moving Average

Our error formula in (17) shows that a small amount of noise can quickly increase the complexity of the recovery, especially for higher-order PDEs. Denoising is helpful in general. We propose an order preserving method which keeps the order of the approximation to the underlying function, while smooths out possible noise.

Let the data $\{d_i\}$ be given on a one-dimensional uniform grid $\{x_i\}$ and define its five-point moving average as $\tilde{d}_i = \frac{1}{5}\sum_{l=0,\pm1,\pm2} d_{i+l}$ for all i. At each grid point x_i, we determine a quadratic polynomial $p(x) = a_0 + a_1(x - x_i) + a_2(x - x_i)^2$ fitting the local data, which preserves the order of smoothness, up to the degree of polynomial. There are a few possible choices for denoising, such as (i) Least-Squares Fitting (LS): find a_0, a_1 and a_2 to minimize the functional $F(a_0, a_1, a_2) = \sum_{\text{some } j \text{ near } i}(p(x_j) - d_j)^2$; (ii) Moving-Average Fitting (MA): find a_0, a_1 and a_2, such that the local average of the fitted polynomial matches with the local average of the data, $1/5\sum_{l=0,\pm1,\pm2} p(x_{j+l}) = \tilde{d}_j$, for $j = i, i \pm 1$ (or another set of 3 grid points near $\{x_i\}$). The polynomial generated by LS may not represent the underlying true dynamics. Moving average fitting is better in keeping the underlying dynamics, however, the matrix may be ill-conditioned when a linear system is solved to determine a_0, a_1, a_2.

We propose to use (iii) Least-Squares Moving Average (LSMA): find a_0, a_1 and a_2 to minimize the functional

$$G(a_0, a_1, a_2) = \sum_{j=i, i\pm1, i\pm2} \left\{ \left[\frac{1}{5} \sum_{l=0,\pm1,\pm2} p(x_{j+l}) \right] - \tilde{d}_j \right\}^2 .$$

The condition number of this linear system tends to be better in comparison with MA, because j is chosen from a larger set of indices. This LSMA denoising method preserves the approximation order of data and can easily be incorporated into numerical PDE techniques. MA fitting and LSMA are similar to the non-oscillatory polynomial reconstruction from cell averages which is a key step in high-resolution shock capturing schemes, see e.g. [1, 11,12]. The quadratic polynomials computed by the methods above are locally third-order approximation to the underlying function. We prove that, if the given data are sampled from a third-order approximation to a smooth function, then LSMA will keep the same order of the approximation. This theorem can be easily generalized to any higher order, we kept to 3rd order to be consistent with our experiments in this paper.

Theorem 2 *If data are given as a 3rd order approximation to a smooth function, with or without additive noise, then denosing the data (to obtain a piecewise quadratic function) with the Least-Squares Moving Average (LSMA) method will keep the same order of accuracy to the function.*

Proof Let $f(x)$ be the smooth function. The proof can be done by comparing the quadratic function to that of the Taylor expansion of $f(x)$ at a grid point x_{i_0}, see e.g., [15]. Let $p(x) = a_0 + a_1(x - x_{i_0}) + a_2(x - x_{i_0})^2$ be the quadratic function to be determined near x_{i_0}. The least-squares method solves the linear system $A^T(Ac - b) = 0$ for the coefficient vector $c = [a_0, a_1, a_2]^T$, where A is a 5×3 matrix whose rows can be written as $[1, \frac{1}{5}\sum_{j=0,\pm1,\pm2}(x_{i+j} - x_{i_0}), \frac{1}{5}\sum_{j=0,\pm1,\pm2}(x_{i+j} - x_{i_0})^2]$, for $i = i_0 - 2, \ldots, i_0 + 2$, and $b = [\tilde{d}_{i_0-2}, \ldots, \tilde{d}_{i_0+2}]^T$.

According to the assumption we have

$$\tilde{d}_i = f(x_{i_0}) + f'(x_{i_0}) \frac{1}{5} \sum_{j=0,\pm1,\pm2} (x_{i+j} - x_{i_0}) + \frac{1}{2} f''(x_{i_0}) \frac{1}{5} \sum_{j=0,\pm1,\pm2} (x_{i+j} - x_{i_0})^2 + O(\Delta x^3),$$

for any grid point x_i near x_{i_0}, i.e., $|x_i - x_{i_0}| = O(\Delta x)$. Let $s = [f(x_{i_0}), f'(x_{i_0}), \frac{1}{2}f''(x_{i_0})]^T$. We have

$$A^T(Ac - b) = HB^T\{BH(c - s) + O(\Delta x^3)\},$$

where H is the 3×3 diagonal matrix diag$\{1, \Delta x, \Delta x^2\}$, and

$$B = AH^{-1} = \begin{bmatrix} \vdots & & \\ 1 & \frac{1}{5}\sum_{j=0,\pm1,\pm2} \frac{x_{i_0+j} - x_{i_0}}{\Delta x} & \frac{1}{5}\sum_{j=0,\pm1,\pm2} \frac{(x_{i_0+j} - x_{i_0})^2}{\Delta x^2} \\ \vdots & & \end{bmatrix}.$$

Therefore $H(c - s) = (B^T B)^{-1} B^T \cdot O(\Delta x^3)$. Note that B is independent of Δx, we have $|p(x) - f(x)| = O(\Delta x^3)$ for all x such that $|x - x_{i_0}| = O(\Delta x)$. □

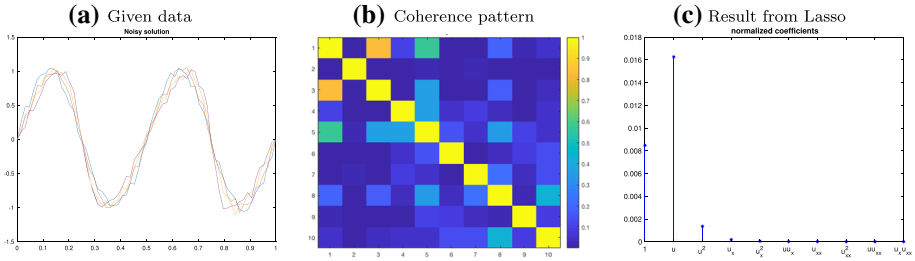


Fig. 4 Burgers' equation in (15) with 8% Gaussian noise. (a) Given noisy data, (b) Coherence pattern of the feature matrix. (c) The normalized coefficient magnitudes from Lasso. This fails to identify the correct term uu_x

3.2 IDENT Experiments for Noisy Data

We next present numerical experiments with noisy data. We say P percent Gaussian noise is added to the noise-free data $\{u_i^n : i = 1, \ldots, N_1 \text{ and } n = 1, \ldots, N_2\}$, if the observed data are $\{\widetilde{u}_i^n\}$ where $\widetilde{u}_i^n = u_i^n + \xi_i^n$ and $\xi_i^n \sim \mathcal{N}(0, \sigma^2)$ with $\sigma = \frac{P}{100}\sqrt{\sum_{i=1}^{N_1}\sum_{n=1}^{N_2}|u_i^n|^2}/\sqrt{N_1 N_2}$.

Our first experiment is on the Burgers' equation in (15) with 8% Gaussian noise. Data are sampled from the analytic solution with $\Delta x = 1/56$ and $\Delta t = 0.004$ for $t \in [0, 0.05]$, and then 8% Gaussian noise is added. For comparison, we do not denoise the given data, but directly applied IDENT. Figure 4a shows the noisy given data, (b) shows the coherence pattern, and (c) shows the normalized coefficient magnitudes from Lasso. The NSR for Lasso defined in (14) is 3.04. Lasso fails to include the correct set of terms, thus TEE identifies the wrong equation $u_t = -0.59u^2$ as a solution.

Active terms	Coefficients	TEE	Active terms	Coefficients	TEE
1	-0.27	94.20	$[1\ u]$	$[-0.27\ 1.17]$	99.15
u	1.17	99.36	$[1\ u^2]$	$[0.18\ -0.83]$	94.04
u^2	-0.59	94.03	$[u\ u^2]$	$[1.17\ -0.59]$	98.85
$[1\ u\ u^2]$	$[0.19\ 1.17\ -0.84]$	98.81			

For the same data, Fig. 5 and the table below show the results when LSMA denoising is applied. After denoising, and the given data is noticeably smoother when Fig. 5a is compared with Fig. 4a. The Lasso result shows great improvement in Fig. 5c. With the correct terms included in the Step 2 of Lasso, TEE determines the PDE with correct feature: $u_t = -0.92uu_x$.

Active terms	Coefficients	TEE	Active terms	Coefficients	TEE
1	-0.25	87.10	$[1\ u]$	$[-0.25\ 0.27]$	87.94
u	0.27	87.89	$[1\ uu_x]$	$[-0.22\ -0.92]$	29.5412
uu_x	-0.92	29.5409	$[u\ uu_x]$	$[0.07\ -0.92]$	29.81
$[1\ u\ uu_x]$	$[-0.22\ 0.07\ -0.92]$	29.80			

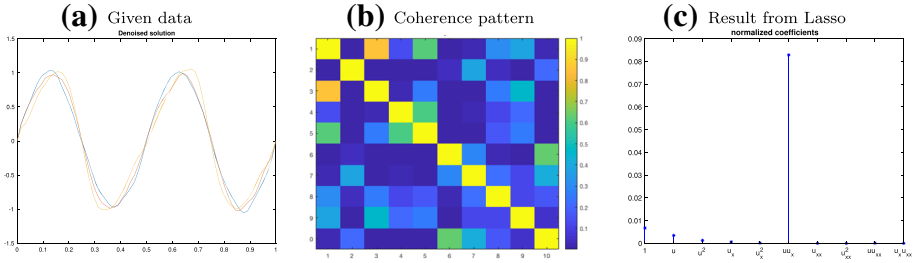
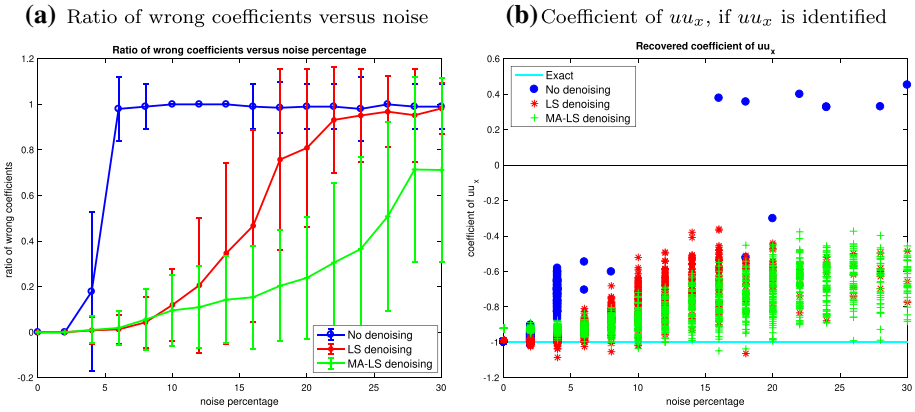


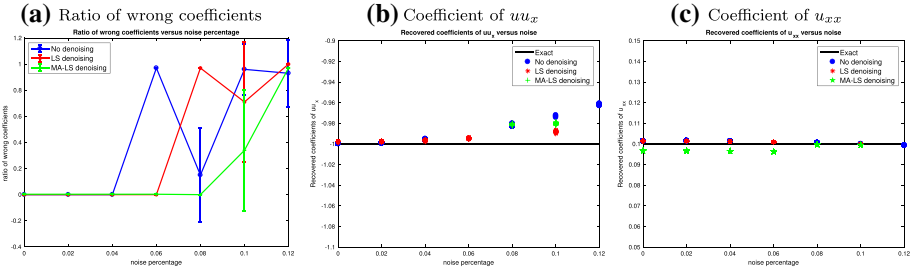
Fig. 5 Burgers' equation in (15) with 8% Gaussian noise as in Fig. 4. **a** The data after the LSMA denoising. **b** Coherence pattern of $\widehat{F}$. **c** The normalized coefficient magnitudes from Lasso identifies 1, u and uu_x which include the correct term uu_x



Added noise in %	0	4	8	12	16	20	24	28
Added noise in new NSR (14)	0.04	2.18	3.09	3.33	3.40	3.31	3.31	3.23

Fig. 6 Burgers' equation (15) with increasing noise levels. **a** The average ratio between the identified wrong coefficients and all identified coefficients over 100 trails. **b** The recovered coefficient of uu_x by IDENT. Denoising the given data with LSMA significantly improves the result. The table shows the new NSR (14) corresponding to the noise level given in percentage

In the next set of experiments, we explore different levels of noise for denosing+IDENT. In Fig. 6, we experiment on the Burgers' equation (15) with its analytic solution sampled in the same way as above, while the noise level increases from 0 to 30%. For each noise level, we (i) first generate data with 100 sets of random noises, (ii) denoise by LS and LSMA for a comparison respectively, then (iii) run IDENT. The parameter τ is chosen as 10% of the largest coefficient magnitude. Figure 6a represents how likely wrong results are found. It is computed by the average ratio between the wrong coefficients and all computed coefficients: $\sum_{j\in\widehat{\Lambda}\setminus\Lambda}|\widehat{\mathbf{a}}_j|/\|\widehat{\mathbf{a}}\|_1$, where Λ and $\widehat{\Lambda}$ are the exact support and the identified support, respectively. Each bar plot represents the standard deviation of the results among 100 trials. The green curves denoised by LSMA show the most stable results even as the noise level increases. Figure 6b shows the recovered coefficient of uu_x, where the true value is -1. Notice that the LSMA+IDENT (green points) results are closer to -1, while others find wrong coefficients more often. In general, denoising the given data with LSMA improves the result significantly.



Added noise in %	0	0.02	0.04	0.06	0.08	0.10	0.12
Added noise in new NSR (14)	0.05	2.02	4.04	6.06	8.08	10.10	12.13

Fig. 7 Burgers' equation with diffusion in (16) with varying noise levels. **a** The average ratio between the identified wrong coefficients and all identified coefficients over 100 trails. **b**, **c** the computed coefficients of uu_x and u_{xx} respectively by IDENT. While the noise level in percentage seems small, the new NSR represents the severeness of noise for PDE with high order derivatives

Figure 7 shows the Burgers' equation with diffusion in (16) with varying noise levels. The given data are sampled in the same way as in Fig. 2, the noise level increases from 0 to 0.12%. (a) shows the average ratio between the wrong coefficients and the total coefficients. (b) and (c) show the recovered coefficients of uu_x and u_{xx}, respectively. Again using LSMA shows better performance.

For both Figs. 6 and 7, we present the new NSR defined in (14). This clearly presents that noise affects different PDEs in different ways. The Burgers' equation (15) only have first order derivatives, while the Burgers' equation with diffusion in (16) has a second order derivative. This seemingly small difference makes a big impact on the NSR and identification. While in Fig. 6, the noise level is experimented up to 30%, its corresponding new NSR varies only from 0 to less than 3.5. In Fig. 7, the noise level only varies from 0 to 0.12 in percentage, however, this corresponds to new NSR varying form 0 to above 12. The level of the new NSR characterizes the difficulty of identification using IDENT (Step 2, Lasso), since having a higher-order term affects the Lasso negatively, especially in the presents of noise.

3.3 Downsampling Effects and IDENT

In applications, data are often collected on a coarse grid to save the expenses of sensors. We explore the effect of downsampling in data collections in this section. Consider a rth order PDE. Simulating its solution with a qth order method on a fine grid with time step δt and spatial spacing δx gives rise to the error $O(\delta t + \delta x^q)$. Suppose data are downsampled by a factor of C_t in time and C_x in space, such that data are sampled with spacing $\Delta t = C_t \delta t$ and $\Delta x = C_x \delta x$. Our error formula in (17) is crucially dependent on the downsampling factors C_t and C_x. Each term is affected by downsampling differently.

- The term $\Delta t + \Delta x^{p+1-r}$ arises from the approximation of time and spatial derivatives. It increases as the downsampling factors C_t and C_x increase.
- The term $\frac{\delta t + \delta x^q}{\Delta t} + \frac{\delta t + \delta x^q}{\Delta x^r}$ arises from the error in data generations. It decreases as the downsampling factors C_t and C_x increase.
- The term $\frac{\sigma}{\Delta t} + \frac{\sigma}{\Delta x^r}$ arises from the measurement noise. It decreases as the downsampling factors C_t and C_x increase.

Therefore, downsampling may positively affect the identification depending on the balance among these three terms.

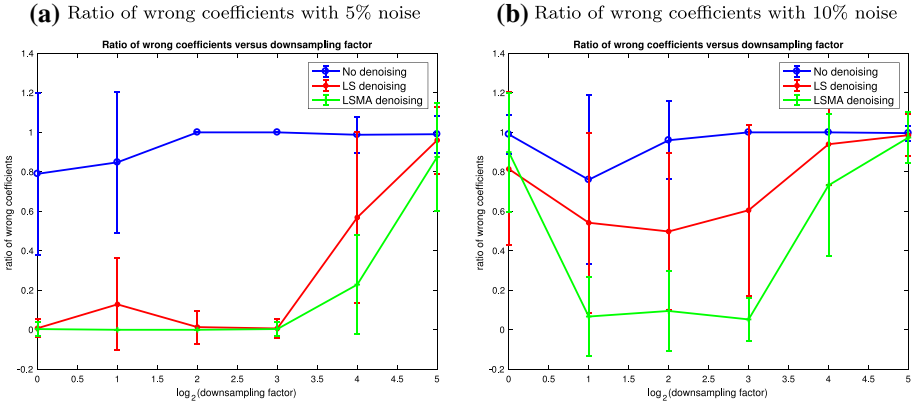


Fig. 8 Burgers' equation in (15) with various downsampling factors. **a**, **b** show the average ratio between the identified wrong coefficients and all identified coefficients in 100 trails versus $\log_2$(downsampling factor) in the presence of 5% (left) and 10% (right) noise respectively. Increasing the downsampling factors can positively affects the result until the downsampling factors become too large

As a numerical example, we consider the Burgers' equation in (15) with different downsampling factors. The analytic solution is evaluated on the grid with spacing $\delta x = 1/1024$ and $\delta t = 0.001$ for $t \in [0, 0.05]$. After evaluating the analytic solution, we generate 100 sets of random noises then downsample the noisy data with spacing $\Delta x = C_x \delta x$ and $\Delta t = C_t \delta t$ where $C_x = C_t = 1, 2, 2^2, 2^3, 2^4$, and 2^5 respectively. We run IDENT on the given downsampled noisy data, denoised by LS and LSMA respectively. Figure 8 displays the ratio of wrong coefficients by IDENT versus $\log_2 C_x$ in the presence of 5% or 10% Gaussian noise. We observe that increasing downsampling rates can positively affect the result until the downsampling rates become too large. LSMA also gives the best performance.

4 Varying Coefficients and Base Element Expansion

In this section, we consider PDEs with varying coefficients, e.g., $a_j(x)$ varying in space. As illustrated in (4), we can easily generalize the IDENT set-up to PDEs with varying coefficients, by expanding the coefficients in terms of finite element bases and solving group Lasso for $L > 1$. Due to the increasing number of coefficients, the complexity of the problem increases as L increases. In order to design a stable algorithm, we propose to let L grow before TEE is applied.

We refer to this extra procedure as Base Element Expansion (BEE). From the given discrete data $\{u_i^n | i = 1, \ldots, N_1 \text{ and } n = 1, \ldots, N_2\}$, we first compute numerical approximations of u_t, u_x, u_{xx}, etc, then apply BEE to gradually increase L until the recovered coefficients become stable. For each fixed L, we form the feature matrix $\widehat{F}$ according to (6), and solve group Lasso with the balancing parameter λ to obtain $\widehat{\mathbf{a}}_{\text{G-Lasso}}(\lambda)$. We record the normalized block magnitudes from group Lasso, as L increases:

$$\text{BEE procedure} := \left\{ \|\widehat{F}[j]\|_{L^1} \left\| \sum_{l=1}^{L} \frac{\widehat{\mathbf{a}}_{\text{G-Lasso}}(\lambda)_{j,l}}{\|\widehat{F}[j,l]\|_\infty} \phi_l \right\|_{L^1} \right\}_{j=1,\ldots,N_3} \text{versus } L.$$

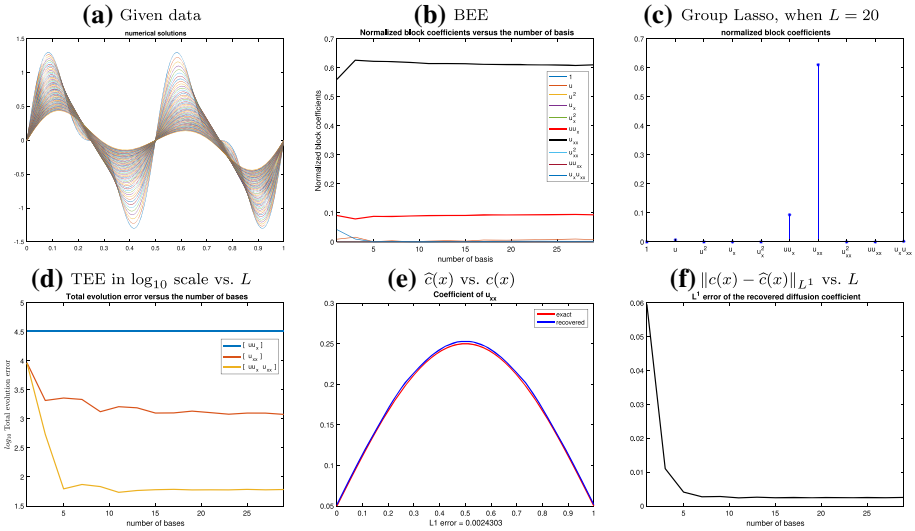


Fig. 9 Burgers' equation with a varying diffusion coefficient (18) where data are downsampled by a factor 4. **a** The given data. **b** BEE as L increases from 1 to 30. **c** An example of the magnitudes of coefficients from Group Lasso when $L = 20$. **d** TEE versus L, for all subsets of coefficients of $\{uu_x\ u_{xx}\}$. **e** Recovered diffusion coefficient $\widehat{c}(x) = \sum_{l=1}^{L} \widehat{a}_{7,l}\phi_l(x)$ when $L = 20$ (blue), compared with the true diffusion coefficient $c(x)$ (red). **f** The error $\|c(x) - \widehat{c}(x)\|_{L^1}$ as L increases from 1 to 30

The main idea of BEE is based on the convergence of the finite element approximation (5) - as more basis functions are used, the more accurate the approximation is. In the BEE procedure, the normalized block magnitudes reach a plateau as L increases, i.e., candidate features can be selected by a thresholding according to (10) when L is sufficiently large. With this added BEE procedure, IDENT continues to the Step 3 of TEE to refine the selection.

In the following, we present various numerical experiments for PDEs with varying coefficients using IDENT with BEE. For the first set of experiments, in Figs. 9, 10 and 11, we assume only one coefficient is known a priori to vary in x. For the second set of experiments, in Fig. 12, we assume two coefficients are known a priori to vary in x, and the final experiment, in Fig. 13 assumes all coefficients are free to vary without any a priori information.

The first experiment is on the Burgers' equation with a varying diffusion coefficient:

$$u_t + \left(\frac{u^2}{2}\right)_x = c(x)u_{xx}, \text{ where } c(x) = 0.05 + 0.2\sin\pi x$$
$$x \in [0,1],\ u(x,0) = \sin(4\pi x) + 0.5\sin(8\pi x) \text{ and } u(0,t) = u(1,t) = 0. \tag{18}$$

The given data, shown in Fig. 9a, is numerically simulated by a first-order method with spacing $\delta x = 1/256$ and $\delta t = (\delta x)^2/2$ for $t \in [0, 0.05]$. Data are downsampled by a factor of 4 in time and space. There is no measurement noise. Our objective is to identify the correct features uu_x and u_{xx} and recover the coefficients -1 and the varying coefficient $c(x)$. After we expand the diffusion coefficient with L finite element bases, the vectors to be identified can be written as $\mathbf{a} = [a_1 \dots a_6\ a_{7,1}\ \dots\ a_{7,L}\ a_8 \dots a_{10}]^T$ where $c(x) \approx \sum_{\ell=1}^{L} a_{7,l}\phi_l(x)$.

Figure 9b presents BEE as L increases from 1 to 30. This graph clearly shows that BEE stabilizes when $L \geq 5$. (c) is an example of the group Lasso result, the normalized block magnitudes, when $L = 20$. The magnitudes of uu_x and u_{xx} are significantly larger than the

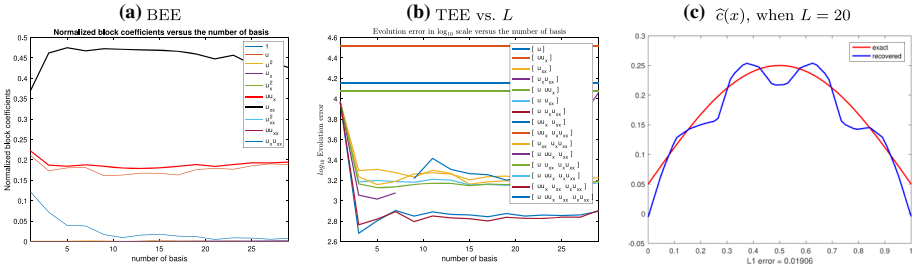


Fig. 10 Equation(18), where data are downsampled by a factor of 4 and 0.2% measurement noise is added. No denoising is applied. **a** BEE as L increases from 1 to 30. **b** TEE versus L, for all subsets of four terms selected in (**a**). The correct support $[uu_x\ u_{xx}]$ is identified with the lowest TEE when $L \geq 7$. **c** Recovered diffusion coefficient $\widehat{c}(x)$ when $L = 20$ (blue), compared with the true diffusion coefficient $c(x)$ (red)

others, so they are picked for TEE in Step 3. Figure 9d presents TEE for all different L: TEE of uu_x, u_{xx} and $[uu_x\ u_{xx}]$ in $\log_{10}$ scale, respectively. The correct set, $[uu_x\ u_{xx}]$, is identified as the recovered feature set with the smallest TEE. The coefficient $[\widehat{a}_6\ \widehat{a}_{7,1} \ldots \widehat{a}_{7,L}]$ is computed by least squares, and (e) displays the recovered diffusion coefficient $\widehat{c}(x) = \sum_{l=1}^{L} \widehat{a}_{7,l}\phi_l(x)$ when $L = 20$, compared with the true equation $c(x) = 0.05 + 0.2\sin\pi x$ given in (18). Figure 9f shows the error $\|c(x) - \widehat{c}(x)\|_{L^1}$ as L increases from 1 to 30. The error decreases as L increases, yet does not converge to 0 due to the errors arising from data generations and finite-difference approximations of u_t, u_x and u_{xx}.

For the same equation, 0.2% noise is added to the next experiments presented in Fig. 10 and 11. Figure 10 presents the result without any denoising. (a) shows BEE as L increases from 1 to 30, where the magnitudes of $u, uu_x, u_{xx}, u_xu_{xx}$ are not negligible for $L \geq 20$. These terms are picked for TEE, and Fig. 10b shows TEE versus L. The correct support $[uu_x\ u_{xx}]$ is identified with the lowest TEE when $L \geq 7$. The computed diffusion coefficient $\widehat{c}(x)$ is compared to the true one in (c), which has the error $\|c(x) - \widehat{c}(x)\|_{L^1} \approx 0.019$. Even for the data with noise, IDENT+BEE without any denoising gives a good identification of the general form of the PDE. However, the varying coefficient approximation can be improved if LSMA denoising applied to the data as discussed in Sect. 3.

Figure 11 presents the same experiment with LSMA denoising. In (a), BEE picks u, uu_x, u_{xx} for TEE. Notice that the coefficient of u_xu_{xx} almost vanishes after denoising compared to Fig. 10. (b) shows TEE versus L, where the correct support $[uu_x\ u_{xx}]$ gives the lowest TEE, when $L \geq 19$. The recovered diffusion coefficient when $L = 20$ is shown in (c), which yields the error $\|c(x) - \widehat{c}(x)\|_{L^1} \approx 0.008$. In comparison with the results in Fig. 10 without denoising, LSMA reduces the error of the recovered diffusion coefficient from 0.019 to 0.008.

In Fig. 12, we experiment on the following PDF with two varying coefficients:

$$\begin{aligned} &u_t = b(x)u_x + c(x)u_{xx}, \text{ where } b(x) = -2x \text{ and } c(x) = 0.05 + 0.2\sin\pi x, \\ &x \in [0,1],\ u(x,0) = \sin(4\pi x) + 0.5\sin(8\pi x) \text{ and } u(0,t) = u(1,t) = 0. \end{aligned} \tag{19}$$

The given data are simulated by a first-order method with spacing $\delta x = 1/256$ and $\delta t = (\delta x)^2/2$ for $t \in [0, 0.05]$. The vectors to be identified are $\mathbf{a} = [a_1 \ldots a_3\ a_{4,1} \ldots a_{4,L}\ a_5\ a_6\ a_{7,1} \ldots a_{7,L}\ a_8 \ldots a_{10}]^T$ where $b(x) \approx \sum_{\ell=1}^{L} a_{4,l}\phi_l(x)$ and $c(x) \approx \sum_{\ell=1}^{L} a_{7,l}\phi_l(x)$. Figure 12a shows the numerical solution of (19). In (b) the given data are downsampled by a factor of 4 in time and space, and in (c) data not downsampled. BEE and TEE successfully identifies the correct features. Figure 12b plots both the recovered

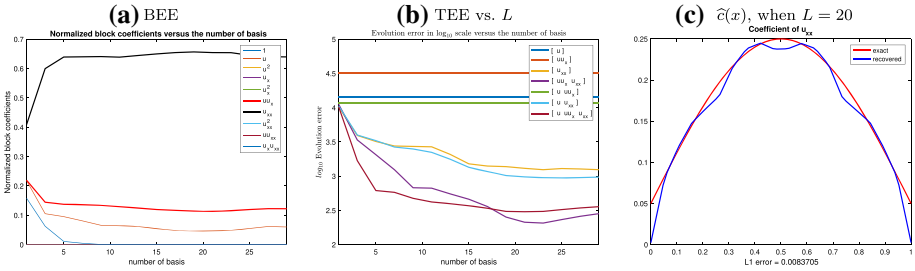


Fig. 11 The same experiment as Fig. 10, but IDENT+BEE is applied with LSMA denoising. **a** BEE as L increases from 1 to 30. **b** TEE versus L, for all subset of coefficients identified in (**a**). The correct support $[uu_x\ u_{xx}]$ is identified since it gives rise to the lowest TEE when $L \geq 19$. **c** The recovered diffusion coefficient when $L = 20$, compared with the true diffusion coefficient (red), which shows a clear improvement compared to Fig. 10c

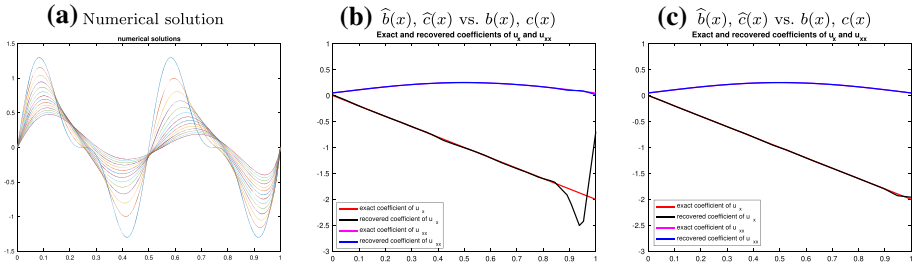


Fig. 12 Equation (19) with varying convection and diffusion coefficients. **a** The numerical solution of (19). **b** With data downsampled by a factor of 4 in time and space, the recovered coefficient $\widehat{b}(x)$ of u_x is not accurate near $x = 1$. The downsampling rate is too high near $x = 1$ so that details of the solution are lost. **c** The same experiment without any downsampling, and the recovered coefficients $\widehat{b}(x)$ and $\widehat{c}(x)$ are more accurate than (b)

coefficients $\widehat{b}(x)$, $\widehat{c}(x)$ and the true coefficients $b(x)$ and $c(x)$ when data are downsampled. Notice that the coefficient $\widehat{b}(x)$ of u_x is not accurate when x is close to 1. The result is improved in (c) where data are not downsampled. No downsampling helps to keep details of the solution around $x = 1$ and reduces the finite-difference approximation errors.

Our final experiment is on Eq. (18), but all coefficients are allowed to vary in x. The numerical solution is simulated in the same way as Fig. 9 and the given data are downsampled by a factor of 4 in time and space. After all coefficients are expanded in terms of L finite element bases, the vectors to be identified is $\mathbf{a} = \{a_{k,l}\}_{k=1,\ldots,10,\ l=1,\ldots,L}$ where $-1 = b(x) \approx \sum_{l=1}^{L} a_{4,l}\phi(x)$ and $c(x) \approx \sum_{\ell=1}^{L} a_{7,l}\phi_l(x)$. Figure 13a shows BEE, (b) shows group Lasso result, and (c) shows TEE. TEE identifies the correct support $[u_x\ u_{xx}]$ since it yields the smallest error. The coefficients $[\widehat{a}_{6,1}\ \ldots\ \widehat{a}_{6,L}\ \widehat{a}_{7,1}\ldots\widehat{a}_{7,L}]$ is computed by least squares. Figure 13d displays the computed coefficients $\widehat{b}(x) = \sum_{l=1}^{L} \widehat{a}_{6,l}\phi_l(x)$ and $\widehat{c}(x) = \sum_{l=1}^{L} \widehat{a}_{7,l}\phi_l(x)$ when $L = 20$, and (e) shows the coefficient recovery errors $\| -1 - \widehat{b}(x)\|_{L^1}$ and $\|c(x) - \widehat{c}(x)\|_{L^1}$ as L increases from 1 to 30. IDENT with BEE successfully identifies the correct terms even when all coefficients are free to vary in space. The accuracy of the recovered coefficients has room for improvement if data are simulated and sampled on a finer grid.

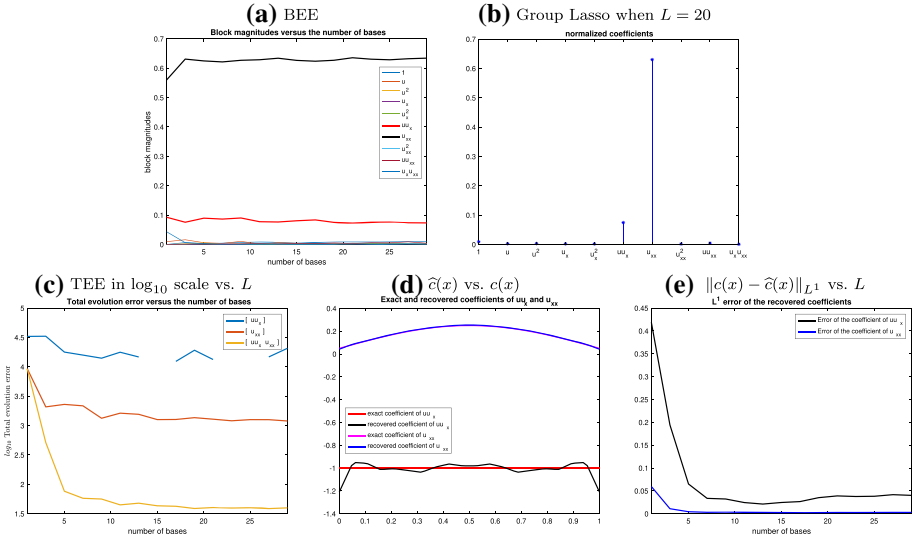


Fig. 13 Equation (18) where we are a priori given that all coefficients are free to vary with respect to x. Data are downsampled by a factor of 4 in time and space

5 Concluding Remarks

We proposed a new method to identify PDEs from a given set of time dependent data, with numerical PDE techniques and the fundamental convergence principle of numerical PDE schemes. Assuming that the PDE contains few active terms in a prescribed dictionary, we use finite differences, such as the ENO scheme, to form an empirical approximation of the dictionary, and utilize L^1 minimization to efficiently select a candidate set. Time Evolution Error (TEE) was proposed as an effective tool to pick the correct set of terms.

Starting from the first set of basic experiments, we considered noisy data, downsampling effects and PDEs with varying coefficients. A recovery theory of Lasso for PDE identification was established, where a new noise-to-signal ratio (14) was proposed to measure the noise level more accurately in the setting of PDE identification. We derived an error formula in (17) and analyzed the effects of noise and downsampling. A new order preserving denoising method called LSMA was proposed in Sect. 3.1 to aid the identification with noisy data. IDENT can be applied to PDEs with varying coefficients and a BEE procedure helps to stabilize the results and reduce the computational cost.

In this set-up of the problem, we consider a specific set of discretization in (1), but our proposed method is general. TEE uses the principle of numerical convergence, and it will work for any consistent and stable set of discretization as effectively as the current one. Typically denoising of data introduces biases to the learning process; yet, it also greatly reduces the variances and helps to identify the correct set of coefficients. The final result is a balance between the biases and the variances. Also, TEE overcomes the biases, since TEE finds the PDE whose evolution best matches the given data. The biases introduced from denoising do not have time dependency, and TEE helps to reduce the biases from denoising.

A recovery theory of Lasso for PDE recovery was established in Theorem 1 in terms of an incoherence condition. In the compressive sensing community, the matrices that are proved to satisfy this incoherence condition are mostly random matrices. In the PDE identification

setting, the feature matrix results from a PDE, and is deterministic. The mutual coherence of the feature matrix is usually close to 1, which violates the incoherence condition. However, the coherence pattern does provide a lot of empirical guidance. When some columns of the feature matrix are highly correlated, the recovered support by Lasso includes the true support and the indices that are highly correlated with the true support. Therefore, our proposed Time Evolution Error (TEE) plays a crucial role in ruling out the wrong terms and picking the correct support. Theorem 1 also gives insight to a correct definition of the Noise-to-Signal Ratio for PDE identification.

Since our problem set-up only utilizes data from one realization of the PDE with one initial condition, it is possible that IDENT identifies a different PDE instead of the true one, when the given data are not sufficient. IDENT will find the PDE, which gives the smallest TEE. When a small amount of data is given, IDENT can find a simpler PDE which may not be the true one, but gives the smallest TEE well representing the given data set. IDENT will not identify any ill-posed PDE, since an ill-posed PDE will blow up the TEE and will not be chosen by IDENT.

The idea of selecting a candidate set by Lasso and then refining the results was also considered in [20], where the authors used a well-known statistical AIC test to find the best equation, which minimizes certain loss function. IDENT utilizes numerical time evolution, which focuses on the evolution of PDE and better captures the time dependent dynamics. In practice, we observed that, for noisy data and/or nonlinear high order PDEs, TEE performs well.

Our proposed method can be compared to many existing methods in the following way: First, this method does not require large training data sets and only uses one realization of the PDE, while a large training set is required in the deep learning approach [14,17,19,21,21–24]. Second, by using various numerical PDE schemes, (i) our method is flexible with the boundary conditions, instead of only considering periodic boundary conditions [26]; (ii) it can handle noisy data—which is considered to be very challenging in this area. Third, this is the first work to address varying coefficients, which opens up many new PDEs to be identified.

Appendix A: Recovery Theory of Lasso with a Weighted L^1 Norm

In the field of compressive sensing, performance guarantees for the recovery of sparse vectors from a small number of noisy linear measurements by Lasso have been established when the sensing matrix satisfies an incoherence property [8] or a restricted isometry property [7]. We establish the incoherence property of Lasso for the case of identifying PDE, where a weighted L^1 norm is used.

Given a sensing matrix $\Phi \in \mathbb{R}^{n\times m}$ and the noisy measurement

$$\mathbf{b} = \Phi \mathbf{x}^{\text{opt}} + \mathbf{e}$$

where $\mathbf{x}^{\text{opt}}$ is s-sparse ($\|\mathbf{x}^{\text{opt}}\|_0 = s$), the goal is to recover $\mathbf{x}_{\text{opt}}$ in a robust way. Denote the support of $\mathbf{x}^{\text{opt}}$ by Λ and let Φ_Λ be the submatrix of Φ whose columns are restricted on Λ. Suppose $\Phi = [\phi[1]\ \phi[2]\ \dots \phi[m]]$ where all $\phi[j]$'s have unit norm. Let the mutual coherence of Φ be

$$\mu(\Phi) = \max_{j\neq l} |\phi[j]^T \phi[l]|.$$

The principle of Lasso with a weighted L^1 norm is to solve

$$\min_{\mathbf{x}} \frac{1}{2}\|\Phi\mathbf{x}-\mathbf{b}\|_2^2+\gamma\|W\mathbf{x}\|_1 \tag{W-Lasso}$$

where $W=\mathrm{diag}(w_1,w_2,\ldots,w_m)$, $w_j\neq 0$, $j=1,\ldots,m$ and γ is a balancing parameter. Let $w_{\max}=\max_j|w_j|$ and $w_{\min}=\min_j|w_j|$. Lasso successfully recovers the support of $\mathbf{x}^{\mathrm{opt}}$ when $\mu(\Phi)$ is sufficiently small. The following proposition is a generalization of Theorem 8 in [33] from L^1 norm regularization to weighted L^1 norm regularization.

Proposition 1 *Suppose the support of* $\mathbf{x}^{\mathrm{opt}}$*, denoted by* Λ*, contains no more than* s *indices,* $\mu(s-1)<1$ *and*

$$\frac{\mu s}{1-\mu(s-1)}<\frac{w_{\min}}{w_{\max}}.$$

Let

$$\gamma=\frac{1-\mu(s-1)}{w_{\min}[1-\mu(s-1)]-w_{\max}\mu s}\|e\|_2^+, \tag{20}$$

and $\mathbf{x}(\gamma)$ *be the minimizer of* (W-Lasso)*. Then*

1) *the support of* $\mathbf{x}(\gamma)$ *is contained in* Λ*;*
2) *the distance between* $\mathbf{x}(\gamma)$ *and* $\mathbf{x}^{\mathrm{opt}}$ *satisfies*

$$\|\mathbf{x}(\gamma)-\mathbf{x}^{\mathrm{opt}}\|_\infty\leq\frac{w_{\max}}{w_{\min}[1-\mu(s-1)]-w_{\max}\mu s}\|\mathbf{e}\|_2; \tag{21}$$

3) *if*

$$\mathbf{x}^{\mathrm{opt}}_{\min}:=\min_{j\in\Lambda}|x_j^{\mathrm{opt}}|>\frac{w_{\max}}{w_{\min}[1-\mu(s-1)]-w_{\max}\mu s}\|\mathbf{e}\|_2,$$

then $\mathrm{supp}(\mathbf{x}(\gamma))=\Lambda$.

Proof Under the condition $\mu(s-1)<1$, Λ indexes a linearly independent collection of columns of Φ. Let $\mathbf{x}^\star$ be the minimizer of (W-Lasso) over all vectors supported on Λ. A necessary and sufficient condition on such a minimizer is that

$$\mathbf{x}^{\mathrm{opt}}-\mathbf{x}^\star=\gamma(\Phi_\Lambda^*\Phi_\Lambda)^{-1}\mathbf{g}-(\Phi_\Lambda^*\Phi_\Lambda)^{-1}\Phi_\Lambda^*\mathbf{e} \tag{22}$$

where $\mathbf{g}\in\partial\|W\mathbf{x}^\star\|_1$, meaning $g_j=w_j\mathrm{sign}(x^\star)$ whenever $x_j^\star\neq 0$ and $|g_j|\leq w_j$ whenever $x_j^\star=0$. It follows that $\|\mathbf{g}\|_\infty\leq w_{\max}$ and

$$\|\mathbf{x}^\star-\mathbf{x}^{\mathrm{opt}}\|_\infty\leq\gamma\|(\Phi_\Lambda^*\Phi_\Lambda)^{-1}\|_{\infty,\infty}(w_{\max}+\|\mathbf{e}\|_2). \tag{23}$$

Next we prove $x^\star$ is also the global minimizer of (W-Lasso) by demonstrating that the objective function increases when we change any other component of $\mathbf{x}^\star$. Let

$$L(\mathbf{x})=\frac{1}{2}\|\Phi\mathbf{x}-\mathbf{b}\|_2^2+\gamma\|W\mathbf{x}\|_1.$$

Choose an index $\omega\notin\Lambda$ and let δ be a nonzero scalar. We will develop a condition which ensures that

$$L(\mathbf{x}^\star+\delta\mathbf{e}_\omega)-L(\mathbf{x}^\star)>0$$

where $\mathbf{e}_\omega$ is the ωth standard basis vector. Notice that

$$L(\mathbf{x}^\star+\delta\mathbf{e}_\omega)-L(\mathbf{x}^\star)=\frac{1}{2}\left[\|\Phi(\mathbf{x}^\star+\delta\mathbf{e}_\omega)-\mathbf{b}\|_2^2-\|\Phi\mathbf{x}^\star-\mathbf{b}\|_2^2\right]$$

$$
\begin{aligned}
&+\gamma\left(\|W(\mathbf{x}^\star+\delta\mathbf{e}_\omega)\|_1-\|W\mathbf{x}\|_1\right)\\
&=\frac{1}{2}\|\delta\phi[\omega]\|^2+\mathrm{Re}\langle\Phi\mathbf{x}^\star-\mathbf{b},\delta\phi[\omega]\rangle+\gamma|w_\omega\delta|\\
&>\mathrm{Re}\langle\Phi\mathbf{x}^\star-\mathbf{b},\delta\phi[\omega]\rangle+\gamma|w_\omega\delta|\\
&\geq\gamma w_{\min}|\delta|-|\langle\Phi\mathbf{x}^\star-\Phi\mathbf{x}^{\mathrm{opt}}-\mathbf{e},\delta\phi[\omega]\rangle|\ \text{since}\ \mathbf{b}=\Phi\mathbf{x}^{\mathrm{opt}}+\mathbf{e}\\
&=\gamma w_{\min}|\delta|-|\langle\Phi_\Lambda\mathbf{x}^\star_\Lambda-\Phi_\Lambda\mathbf{x}^{\mathrm{opt}}_\Lambda-\mathbf{e},\delta\phi[\omega]\rangle|\\
&\geq\gamma w_{\min}|\delta|-|\langle\Phi_\Lambda(\mathbf{x}^\star_\Lambda-\mathbf{x}^{\mathrm{opt}}_\Lambda),\delta\phi[\omega]\rangle|-|\langle\mathbf{e},\delta\phi[\omega]\rangle|\\
&=\gamma w_{\min}|\delta|-|\langle\gamma\Phi_\Lambda(\Phi^*_\Lambda\Phi_\Lambda)^{-1}\mathbf{g},\delta\phi[\omega]\rangle|-|\langle\mathbf{e},\delta\phi[\omega]\rangle|\ \text{thanks to(22)}\\
&\geq\gamma w_{\min}|\delta|-\gamma|\delta|\cdot|\langle\Phi_\Lambda(\Phi^*_\Lambda\Phi_\Lambda)^{-1}\mathbf{g},\phi[\omega]\rangle|-|\delta|\|\mathbf{e}\|_2\\
&=\gamma w_{\min}|\delta|-\gamma|\delta|\cdot|\langle(\Phi^\dagger_\Lambda)^*\mathbf{g},\phi[\omega]\rangle|-|\delta|\|\mathbf{e}\|_2\\
&=\gamma w_{\min}|\delta|-\gamma|\delta|\cdot|\langle\mathbf{g},\Phi^\dagger_\Lambda\phi[\omega]\rangle|-|\delta|\|\mathbf{e}\|_2\\
&\geq\gamma w_{\min}|\delta|-\gamma|\delta|\|\mathbf{g}\|_\infty\|\Phi^\dagger_\Lambda\phi[\omega]\|_1-|\delta|\|\mathbf{e}\|_2\\
&\geq\gamma w_{\min}|\delta|-\gamma|\delta|w_{\max}\max_{\omega\notin\Lambda}\|\Phi^\dagger_\Lambda\phi[\omega]\|-|\delta|\|\mathbf{e}\|_2.
\end{aligned}
$$

According to [10,32], $\max_{\omega\notin\Lambda}\|\Phi^\dagger_\Lambda\phi[\omega]\|<\frac{\mu s}{1-\mu(s-1)}$. A sufficient condition to guarantee $L(\mathbf{x}^\star+\delta\mathbf{e}_\omega)-L(\mathbf{x}^\star)>0$ is

$$
\gamma\left(w_{\min}-w_{\max}\frac{\mu s}{1-\mu(s-1)}\right)>\|\mathbf{e}\|_2,
$$

which gives rise to (20). This establishes that $\mathbf{x}^\star$ is the global minimizer of (W-Lasso). (21) is resulted from (23) along with $\|(\Phi^*_\Lambda\Phi_\Lambda)^{-1}\|_{\infty,\infty}\leq[1-\mu(s-1)]^{-1}$. □

We prove Theorem 1 based on Proposition 1.

Proof of Theorem 1 Suppose $\widehat{F}_{\mathrm{unit}}$ is obtained from $\widehat{F}$ with the columns normalized to unit L^2 norm and let $W\in\mathbb{R}^{N_3\times N_3}$ be the diagonal matrix with $W_{jj}=\|\widehat{F}[j]\|_\infty\|\widehat{F}[j]\|_2^{-1}$. The Lasso we solve is equivalent to

$$
\widehat{\mathbf{y}}=\arg\min\frac{1}{2}\|\widehat{\mathbf{b}}-\widehat{F}_{\mathrm{unit}}\mathbf{y}\|+\lambda\|W\mathbf{y}\|_1
$$

where $\mathbf{z}=W\mathbf{y}$, $\mathbf{y}^{\mathrm{opt}}_j=\mathbf{a}_j\|\widehat{F}[j]\|_2$ and $\mathbf{e}=\widehat{\mathbf{b}}-\widehat{F}_{\mathrm{unit}}\mathbf{y}^{\mathrm{opt}}$. Then we apply Proposition 1. The choice of balancing parameters in (20) suggests

$$
\lambda=\frac{1-\mu(s-1)}{\min_j\frac{\|\widehat{F}[j]\|_\infty}{\|\widehat{F}[j]\|_2}[1-\mu(s-1)]-\max_j\frac{\|\widehat{F}[j]\|_\infty}{\|\widehat{F}[j]\|_2}\mu s}\|\mathbf{e}\|_2^+,
$$

which gives rise to (11). The error bound in (21) gives

$$
\|\widehat{\mathbf{y}}-\mathbf{y}^{\mathrm{opt}}\|_\infty\leq\frac{(\max_j\|\widehat{F}[j]\|_\infty\|\widehat{F}[j]\|_2^{-1}+\|\mathbf{e}\|_2)}{\min_j\|\widehat{F}[j]\|_\infty\|\widehat{F}[j]\|_2^{-1}[1-\mu(s-1)]-\max_j\|\widehat{F}[j]\|_\infty\|\widehat{F}[j]\|_2^{-1}\mu s}\|\mathbf{e}\|_2
$$

which implies

$$
\max_j\|\widehat{F}[j]\|_{L^2}\left|\|\widehat{F}[j]\|_\infty^{-1}\widehat{\mathbf{a}}_{\mathrm{Lasso}}(\lambda)_j-\mathbf{a}_j\right|\leq\frac{w_{\max}+\varepsilon/\sqrt{\Delta t\Delta x}}{w_{\min}[1-\mu(s-1)]-w_{\max}\mu s}\varepsilon,
$$

which yields (12). □

References

1. Barth, T., Frederickson, P.: Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction. In 28th Aerospace Sciences Meeting, p. 13 (1990)
2. Bertsekas, D.P.: Constrained Optimization and Lagrange Multiplier Methods. Academic Press, London (2014)
3. Bongard, J., Lipson, H.: Automated reverse engineering of nonlinear dynamical systems. Proc. Natl. Acad. Sci. **104**(24), 9943–9948 (2007)
4. Bongini, M., Fornasier, M., Hansen, M., Maggioni, M.: Inferring interaction rules from observations of evolutive systems i: the variational approach. Math. Models Methods Appl. Sci. **27**(05), 909–951 (2017)
5. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends ® Mach. Learn. **3**(1), 1–122 (2011)
6. Brunton, S.L., Proctor, J.L., Kutz, J.N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc. Natl. Acad. Sci. **113**(15), 3932–3937 (2016)
7. Candès, E.J., Romberg, J., Tao, T.: Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Trans. Inf. Theory **52**(2), 489–509 (2006)
8. Donoho, D.L., Huo, X.: Uncertainty principles and ideal atomic decomposition. IEEE Trans. Inf. Theory **47**(7), 2845–2862 (2001)
9. Fannjiang, A., Liao, W.: Coherence pattern-guided compressive sensing with unresolved grids. SIAM J. Imaging Sci. **5**(1), 179–202 (2012)
10. Fuchs, J.-J.: On sparse representations in arbitrary redundant bases. IEEE Trans. Inf. Theory **50**(6), 1341–1344 (2004)
11. Harten, A., Engquist, B., Osher, S., Chakravarthy, S.R.: Uniformly high order accurate essentially non-oscillatory schemes, iii. J. Comput. Phys. **71**(2), 231–303 (1987)
12. Changqing, H., Shu, C.-W.: Weighted essentially non-oscillatory schemes on triangular meshes. J. Comput. Phys. **150**(1), 97–127 (1999)
13. Kaiser, E., Kutz, J.N., Brunton, S.L.: Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. Proc. R. Soc. A **474**(2219), 20180335 (2018)
14. Khoo, Y., Ying, L.: Switchnet: a neural network model for forward and inverse scattering problems. arXiv preprint arXiv:1810.09675 (2018)
15. Liu, Y., Shu, C.-W., Tadmor, E., Zhang, M.: Central discontinuous Galerkin methods on overlapping cells with a non-oscillatory hierarchical reconstruction. SIAM J. Numer. Anal. **45**, 2442–2467 (2007)
16. Loiseau, J.-C., Brunton, S.L.: Constrained sparse Galerkin regression. J. Fluid Mech. **838**, 42–67 (2018)
17. Long, Z., Lu, Y., Ma, X., Dong, B.: PDE-Net: learning PDEs from data. arXiv preprint arXiv:1710.09668 (2017)
18. Lu, F., Zhong, M., Tang, S., Maggioni, M.: Nonparametric inference of interaction laws in systems of agents from trajectory data. arXiv preprint arXiv:1812.06003 (2018)
19. Lusch, B., Kutz, J.N., Brunton, S.L.: Deep learning for universal linear embeddings of nonlinear dynamics. Nat. Commun. **9**(1), 4950 (2018)
20. Mangan, N.M., Kutz, J.N., Brunton, S.L., Proctor, J.L.: Model selection for dynamical systems via sparse regression and information criteria. Proc. R. Soc. A **473**(2204), 20170009 (2017)
21. Qin, T., Wu, K., Xiu, D.: Data driven governing equations approximation using deep neural networks. arXiv preprint arXiv:1811.05537 (2018)
22. Raissi, M.: Deep hidden physics models: Deep learning of nonlinear partial differential equations. arXiv preprint arXiv:1801.06637 (2018)
23. Raissi, M., Karniadakis, G.E.: Hidden physics models: machine learning of nonlinear partial differential equations. J. Comput. Phys. **357**, 125–141 (2018)
24. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics informed deep learning (part i): data-driven solutions of nonlinear partial differential equations. arXiv preprint arXiv:1711.10561 (2017)
25. Rudy, S.H., Brunton, S.L., Proctor, J.L., Kutz, J.N.: Data-driven discovery of partial differential equations. Sci. Adv. **3**(4), e1602614 (2017)
26. Schaeffer, H.: Learning partial differential equations via data discovery and sparse optimization. Proc. R. Soc. A Math. Phys. Eng. Sci. **473**(2197), 20160446 (2017)
27. Schaeffer, H., Caflisch, R., Hauck, C.D., Osher, S.: Sparse dynamics for partial differential equations. Proc. Nat. Acad. Sci. **110**(17), 6634–6639 (2013)
28. Schaeffer, H., Tran, G., Ward, R.: Extracting sparse high-dimensional dynamics from limited data. SIAM J. Appl. Math. **78**(6), 3279–3295 (2018)
29. Schmidt, M., Lipson, H.: Distilling free-form natural laws from experimental data. Science **324**(5923), 81–85 (2009)

30. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Ser. B (Methodol.) **58**, 267–288 (1996)
31. Tran, G., Ward, R.: Exact recovery of chaotic systems from highly corrupted data. Multisc. Model. Simul. **15**(3), 1108–1129 (2017)
32. Tropp, J.A.: Just relax: convex programming methods for subset selection and sparse approximation. ICES report, 404 (2004)
33. Tropp, J.A.: Just relax: convex programming methods for identifying sparse signals in noise. IEEE Trans. Inf. Theory **52**(3), 1030–1051 (2006)
34. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **68**(1), 49–67 (2006)
35. Zhang, S., Lin, G.: Robust data-driven discovery of governing physical laws with error bars. Proc. R. Soc. A **474**(2217), 20180305 (2018)