Computers and Mathematics with Applications 140 (2023) 33–43

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

Discovery of PDEs driven by data with sharp gradient or discontinuity

Kang Wang [a], Lei Zhang [b,c,*], Shaoqiang Tang [a]

[a] HEDPS and LTCS, College of Engineering, Peking University, Beijing 100871, China
[b] The State Key Laboratory of Nonlinear Mechanics, Institute of Mechanics, Chinese Academy of Sciences, Beijing 100190, China
[c] School of Engineering Sciences, University of Chinese Academy of Sciences, Beijing 100049, China

ARTICLE INFO

Keywords:
System identification
Partial differential equations
Gaussian process
Discontinuous data
Machine learning

ABSTRACT

In the data-driven discovery of partial differential equations, previous researchers have successfully employed various methods to derive estimation of parameters from smooth data, but not from data with sharp gradient or discontinuity. To capture the sharp gradient/discontinuous part in data, we introduce a non-zero mean function in terms of the Sigmoid function in the Gaussian process prior. We test the method using noise-free and noisy data on the regression problem and the inverse problem of Burgers' equation, inviscid Burgers' equation, and the nonlinear wave system (NLWS), and verify its effectiveness and robustness.

1. Introduction

Based on knowledge of mathematical formulas, physical laws, or other principles, we can generate data through experiments or simulations. In turn, data can also be employed to discover the principles underneath, or in many cases, the governing partial differential equations (PDEs) [1]. The discovery of PDEs aims at determining the unknown parameters in the equations, which regulate the operation of the system or reveal certain properties of an object. A relevant term "system identification" is also used [2–7], which mostly focuses on the reconstruction of the equations based on observation data. The identification of parameters in PDEs can boost our knowledge in a group of systems with similar physical backgrounds, and provide guidance for potential applications.

Data-driven algorithms have been widely used in the discovery of PDEs. The most direct approach is sparse regression, which involves discretization of the spatial derivative [8,9], local interpolation [10–13] or time integration scheme [14–16]. Other statistical models include Deep Neural Networks (DNN) and Gaussian Processes (GP). DNN constructs a neural network for the target function, and has been applied to both the solution [17] and the discovery of PDEs [18–21]. GP sets a prior distribution to the data at a certain timepoint, and has been proved effective in linear [22,23] and non-linear PDEs [24–27], and also fractional differential equations [22,28]. Compared to DNN, the GP model introduces fewer hyperparameters, which decreases the risk of over-fitting and meanwhile speeds up the training process. GP can be thought as a Bayesian alternative to the class of kernel methods and gives well-calibrated probabilistic outputs [29], which is suitable for regression problems [30,31]. GP model has also been applied to regression problems with discontinuous data [32–34].

Currently, the application of the GP model to data-driven discovery has the following limitation. Previous researchers [35] set the mean function in the GP prior as zero for simplicity. Although the GP model with zero mean function does work for smooth cases, it fails for data with sharp gradient or discontinuity (such as shock waves in supersonic flow). A GP prior with zero mean function and squared exponential kernel function is too smooth to capture jumps in data, and results in bad performance as shown in Section 4.

In our work, the GP model is enhanced by introducing the Sigmoid function as the mean function in the GP prior. The key idea is to use a non-zero mean function to capture the sharp gradient/discontinuous part, and the conventional GP model to deal with the smooth part. A natural idea is to take functions containing discontinuity like the Heaviside function as the mean function. Our previous work introduced the Heaviside function into the GP model as the mean function to deal with inverse problems with discontinuity, while there are some problems doing this. Firstly, the Heaviside mean function is discontinuous and thus leads to a complicated optimization, for we cannot use gradient-based algorithms directly. Secondly, the loss is piecewise constant with respect to the location of the shock (discontinuity) because the Heaviside mean function is a piecewise constant function, which means that the optimal solution is not unique. To address these issues, we choose a smooth function, the Sigmoid function, as the mean function in this paper, because it is easy to obtain the analytical form of its derivatives, which

* Corresponding author at: The State Key Laboratory of Nonlinear Mechanics, Institute of Mechanics, Chinese Academy of Sciences, Beijing 100190, China.
E-mail address: zhanglei@imech.ac.cn (L. Zhang).

https://doi.org/10.1016/j.camwa.2023.03.021
Received 21 July 2022; Received in revised form 25 January 2023; Accepted 19 March 2023
Available online 29 March 2023

is convenient for the training stage, and avoid the trouble of piecewise optimization and nonuniqueness of solution.

To interpret the modification and demonstrate the effectiveness, we apply the GP model with a non-zero mean function to a regression problem with discontinuous data first. Then we illustrate the effectiveness of our method for data-driven discovery of PDEs by Burgers' equation, inviscid Burgers' equation, and the nonlinear wave system (NLWS) with sharp gradient/discontinuous data, and further consider noisy cases to test the robustness.

The rest of this paper is organized as follows. Section 2 introduces the problem setup for data-driven discovery of equations and the backward Euler scheme used in the proposed method. Section 3 describes the modified GP model with non-zero mean function for the discovery of PDEs. Section 4 shows the numerical results for the regression problem and the inverse problem of Burgers' equation, inviscid Burgers' equation, and NLWS.

2. Problem setup

Consider parameterized nonlinear partial differential equations of the general form

$$u_t + \mathcal{N}_x^\lambda u = 0, \tag{1}$$

where $u(t,x)$ denotes a function of time t and position x, $\mathcal{N}_x^\lambda$ is a nonlinear operator parameterized by λ that is unknown.

We have two snapshots of a wave profile $u(t,x)$ at t^0 and t^1, denoted by $u^1 \sim u(t^1,x)$ and $u^0 \sim u(t^0,x)$, respectively. Our objective is to discover λ numerically from the given data. Assuming the time step $\Delta t = t^1 - t^0$ is small enough, we can apply the backward Euler scheme to equation (1) and obtain the discretized equation

$$u^0 = \mathcal{L}_x^\lambda u^1. \tag{2}$$

The reason for adopting the backward Euler method is that it has a minimal stencil and is most convenient to present our theoretical work. Other schemes are also available for presenting our idea. We remark that if a multi-step method is adopted, our theory needs to be further extended, but our idea still works.

We take Burgers' equation for illustration, which is a minimal model of fluid mechanics. In one space dimension, the equation reads

$$u_t + \lambda u u_x = \nu u_{xx}, \tag{3}$$

where ν is the viscosity and λ is the unknown parameter. The corresponding linear operator is

$$\mathcal{L}_x^{\lambda,\mu} u^1(x) = \left(I + \lambda \Delta t \mu(x) \frac{d}{dx} - \nu \Delta t \frac{d^2}{dx^2} \right) u^1(x). \tag{4}$$

There exist different ways to define the parameter μ in different schemes [24]. Here, we use u^0 to approximate it in the backward Euler scheme. Note that in the numerical implementations, we only consider the mean and covariance of data points, so $\mu(x)$ approximated by $u^0(x)$ is known at the sampled spatial points $\mathbf{x}^0$.

Assume that $u^0(x)$ is provided at M sample points $\mathbf{x}^0$ denoted by $\mathbf{u}^0$ and $u^1(x)$ at N sample points $\mathbf{x}^1$ as $\mathbf{u}^1$, with

$$\mathbf{x}^0 = \{x_1^0, ..., x_M^0\},\ \mathbf{x}^1 = \{x_1^1, ..., x_N^1\},$$
$$\mathbf{u}^0 = \{u_1^0, ..., u_M^0\},\ \mathbf{u}^1 = \{u_1^1, ..., u_N^1\}.$$

They may be subjected to a certain level of noise. For the sake of simplicity, we consider the wave profile containing only one sharp gradient, though the position is unknown. Note that the sample points $\mathbf{x}^0$ and $\mathbf{x}^1$ can be different. The viscosity ν can also be inferred in the same manner.

3. Methodology

GP model with zero mean function can be used to successfully identify parameters from smooth data [22,25], but it might fail for data with sharp gradient/discontinuity, as illustrated in Section 4. To address such a problem, we develop a GP model with a non-zero mean function as below, called the GPNOM method. Fig. 1 shows a schematic of the developed framework summarizing the methodology.

3.1. Prior

The key idea of the GP model is to assume $u^1(x) = u(t^1,x)$ to be a Gaussian process with a mean function $m^1(x;\theta_m)$ and a kernel function $k^{1,1}(x,x';\theta_k)$

$$u^1(x) \sim \mathcal{GP}(m^1(x;\theta_m), k^{1,1}(x,x';\theta_k)), \tag{5}$$

with hyperparameters θ_m and $\theta_k = (\sigma_0, \sigma_1)$. The kernel function takes the form

$$k^{1,1}(x,x';\theta_k) = k_{SE}(x,x';\theta_k) = \sigma_0^2 \exp[-\sigma_1^2 (x-x')^2]. \tag{6}$$

It is the one-dimensional form of the squared exponential kernel function k_{SE}, which is common in GP models [36].

Different from the zero mean function used by Raissi et al. [22], we take a non-zero mean function

$$m^1(x;\theta_m) = a\sigma(w(x+b)) \tag{7}$$

with $\theta_m = (a,b,w)$ and the Sigmoid function $\sigma(z) = 1/(1+\exp(-z))$. The jump takes the magnitude a at position $-b$, and with the width $1/w$. See Fig. 2. The mean function of this form is designed to capture the sharp gradient/discontinuous part in the data.

Since $\mathcal{L}_x^\lambda$ is a linear operator in terms of $u^1(x)$, and the linear transformation of a Gaussian process remains Gaussian [22], $u^0(x)$ and $u^1(x)$ form a multi-output Gaussian process

$$\begin{bmatrix} u^1 \\ u^0 \end{bmatrix} \sim \mathcal{GP}\left(\begin{bmatrix} m^1 \\ m^0 \end{bmatrix}, \begin{bmatrix} k^{1,1} & k^{1,0} \\ k^{0,1} & k^{0,0} \end{bmatrix} \right) \tag{8}$$

with

$$m^0(x;\theta_m,\lambda) = \mathcal{L}_x^\lambda m^1(x;\theta_m), \quad k^{1,0}(x,x';\theta_k,\lambda) = \mathcal{L}_{x'}^\lambda k^{1,1}(x,x';\theta_k),$$
$$k^{0,1}(x,x';\theta_k,\lambda) = \mathcal{L}_x^\lambda k^{1,1}(x,x';\theta_k), \quad k^{0,0}(x,x';\theta_k,\lambda) = \mathcal{L}_x^\lambda \mathcal{L}_{x'}^\lambda k^{1,1}(x,x';\theta_k),$$

where $k^{i,j}(x,x') = \mathrm{cov}(u^i(x), u^j(x'))$ represents the covariance between $u^i(x)$ and $u^j(x'), i,j \in \{0,1\}$. The subscript x' of $\mathcal{L}_{x'}^\lambda$ means that the operator acts on the input variable x' with another variable x unchanged.

In practice, we mostly face noisy measurements rather than noise-free data. Assuming noise follows an independent identically Gaussian distribution with variance σ_u^2, we substitute the noisy data $\mathbf{x}^0, \mathbf{x}^1, \mathbf{u}^0, \mathbf{u}^1$ introduced in Section 2 into equation (8). The prior on noisy measurements is

$$\mathbf{u} \sim \mathcal{N}(\mathbf{m}, \mathbf{K}), \tag{9}$$

where column vectors $\mathbf{u}, \mathbf{m}$, and symmetric matrix $\mathbf{K}$ all of order $(M+N)$ are given by

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}^1 \\ \mathbf{u}^0 \end{bmatrix}, \quad \mathbf{m} = \begin{bmatrix} \mathbf{m}^1 \\ \mathbf{m}^0 \end{bmatrix} = \begin{bmatrix} m^1(\mathbf{x}^1) \\ m^0(\mathbf{x}^0) \end{bmatrix},$$
$$\mathbf{K} = \begin{bmatrix} k^{1,1}(\mathbf{x}^1,\mathbf{x}^1) & k^{1,0}(\mathbf{x}^1,\mathbf{x}^0) \\ k^{0,1}(\mathbf{x}^0,\mathbf{x}^1) & k^{0,0}(\mathbf{x}^0,\mathbf{x}^0) \end{bmatrix} + \sigma_u^2 \mathbf{I},$$

respectively. For noise-free data, the variance σ_u^2, which measures the level of noise, equals zero.

3.2. Training

Under the hyperparameters $\theta_m = (a,b,w), \theta_k = (\sigma_0,\sigma_1)$ and model parameter λ, the likelihood function reads

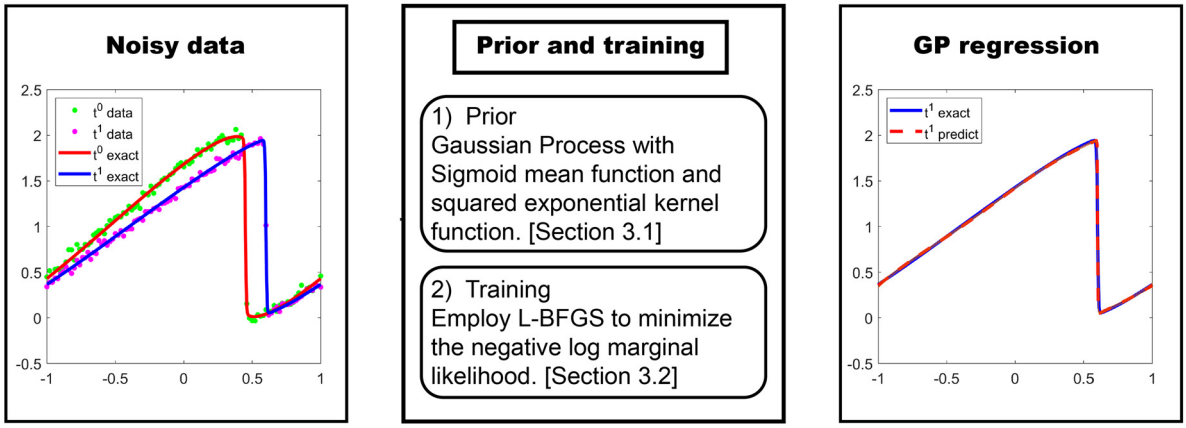

Fig. 1. Flowchart summarizing the GPNOM method developed in this work.

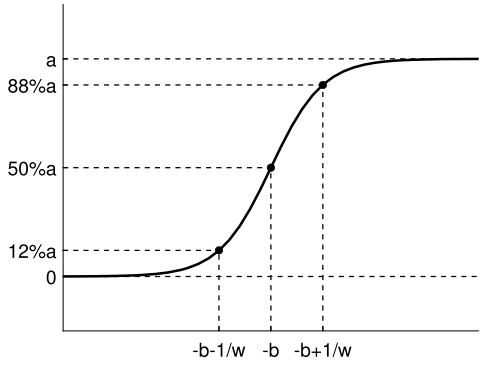

Fig. 2. Non-zero mean function $m^1(x;\theta_m)$.

$$p(\mathbf{u}|\lambda,\theta_m,\theta_k,\sigma_u^2)=\frac{1}{(2\pi)^{(M+N)/2}|\mathbf{K}|^{1/2}}\exp[-\frac{1}{2}(\mathbf{u}-\mathbf{m})^T\mathbf{K}^{-1}(\mathbf{u}-\mathbf{m})]. \quad (10)$$

The model parameter λ and hyperparameters can be trained by employing a quasi-Newton optimizer L-BFGS [37] to minimize the negative log marginal likelihood [30]

$$\begin{aligned} L &= -\log p(\mathbf{u}|\lambda,\theta_m,\theta_k,\sigma_u^2) \\ &= \frac{1}{2}\log|\mathbf{K}|+\frac{1}{2}(\mathbf{u}-\mathbf{m})^T\mathbf{K}^{-1}(\mathbf{u}-\mathbf{m})+\frac{M+N}{2}\log(2\pi). \end{aligned} \quad (11)$$

The three terms of the negative marginal likelihood in equation (11) have the following interpretation. The only term involving the observed targets is the data-fit $(\mathbf{u}-\mathbf{m})^T\mathbf{K}^{-1}(\mathbf{u}-\mathbf{m})/2$, and especially when $m(x)$ gives the exact solution, this term equals zero. $\log|\mathbf{K}|/2$ is the complexity penalty depending only on the covariance function and the inputs, and $(M+N)\log(2\pi)/2$ is a normalization constant.

3.3. Some discussions

We choose the Sigmoid function as the mean function for the following reasons.

Firstly, the Sigmoid function can approximate the sharp gradient part of the data. With appropriate parameters, the rest part $u^1(x)-m^1(x;\theta_m)$ is of better regularity, which the conventional GP model can deal with. From the perspective of stochastic process, GP with zero mean function is a stationary process, which is unsuitable for depicting the data with sharp gradient. Raissi et al. [24] choose GP with a neural network kernel function [30]

$$k_{NN}(x,x';\sigma_0,\sigma_1)=\frac{2}{\pi}\sin^{-1}\left(\frac{2(\sigma_0^2+\sigma_1^2xx')}{\sqrt{(1+2(\sigma_0^2+\sigma_1^2x^2))(1+2(\sigma_0^2+\sigma_1^2x'^2))}}\right) \quad (12)$$

instead of k_{SE} in equation (6) as prior, and get a nonstationary process. In contrast, GP with a non-zero mean function used in our GPNOM is also a nonstationary process.

Secondly, it is easy to obtain the analytical form of the derivatives for the Sigmoid function, which is convenient for the training stage. See Appendix C.

Thirdly, the traveling wave solution of Burgers' equation is in the form of the Sigmoid function [38], which reads

$$u(t,x)=\frac{2w}{1+C\exp[w(x-wt)/\nu]}, \quad (13)$$

with w as the propagation velocity, ν the viscosity, and C the integration constant. It can be rewritten in the form of the Sigmoid function

$$u(t,x)=2w\sigma\left(-w(x-wt)/\nu-\tilde{C}\right), \quad (14)$$

where $\tilde{C}=\ln C$. Especially, for a standing shock, the traveling wave solution is

$$u(t,x)=w\tanh(wx/2\nu). \quad (15)$$

The tangent hyperbolic function is an alternative form for the mean function. We remark that the Sigmoid function also works for cases without a Sigmoid-form traveling wave, such as the following regression problem and the example of NLWS.

Similar to Burgers' equation, in inviscid Burgers' equation

$$u_t+\lambda uu_x=0, \quad (16)$$

a shock wave appears when the velocity profile has a decreasing segment as a function of x, and the observation data would contain discontinuity as time evolves. GPNOM can discover the unknown parameter λ of the inviscid Burgers' equation if we set $\nu=0$ in equation (4), as shown in Subsection 4.3.

4. Numerical results

In the following, we shall present numerical results to explore the effectiveness of GPNOM and the influence of noise. We first illustrate the effectiveness of GPNOM on regression problems with discontinuity by an example of the Forrester function [39]. Then we turn to the inverse problem of Burgers' equation and inviscid Burgers' equation, where we use the data at two time steps t^0 and t^1, with a step size Δt to discover the unknown parameter λ. Finally, we study NLWS to show that GPNOM also applies to systems.

4.1. Forrester function with jump

The discontinuous data are generated by the Forrester function with jump [33]

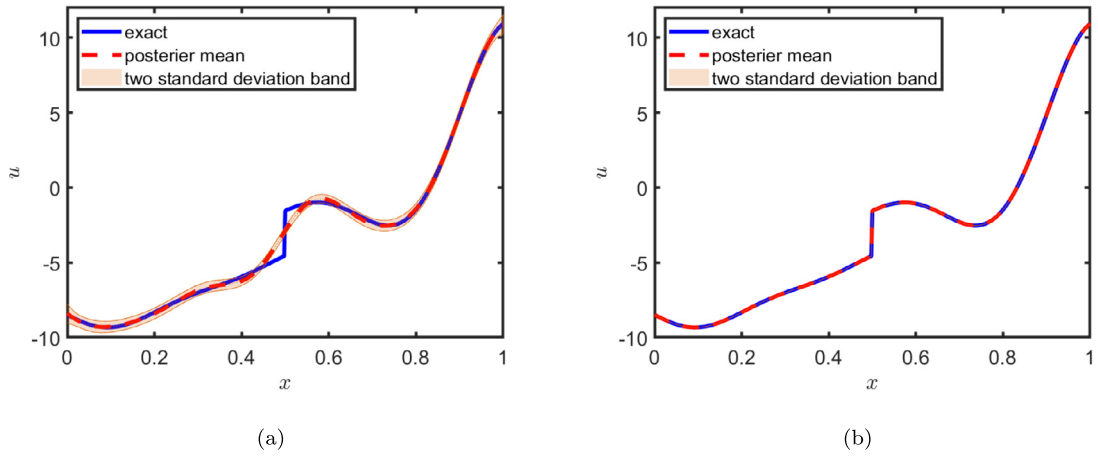

Fig. 3. Regression problem for Forrester function with a jump. (a) GP with zero mean function. (b) GPNOM.

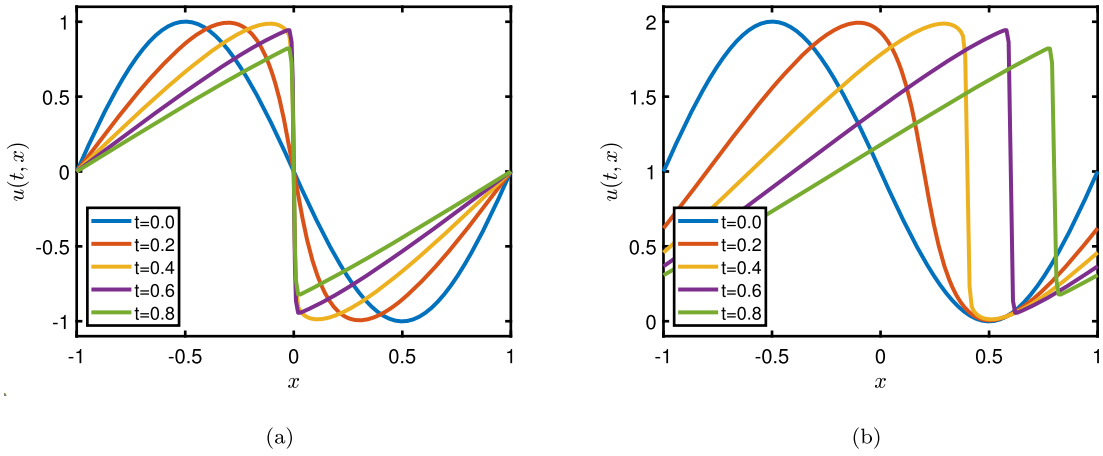

Fig. 4. Wave profiles of the solution of Burgers' equation under periodic boundary condition, with $\nu = 0.01/\pi$. A sharp gradient occurs between $t = 0.20$ and $t = 0.40$. (a) Standing wave with the initial condition $u(0, x) = -\sin(\pi x)$. (b) Moving wave with the initial condition $u(0, x) = 1 - \sin(\pi x)$.

$$f(x) = \begin{cases} 0.5(6x-2)^2 \sin(12x-4) + 10(x-0.5) - 5, & 0 \le x < 0.5, \\ 3 + 0.5(6x-2)^2 \sin(12x-4) + 10(x-0.5) - 5, & 0.5 \le x \le 1. \end{cases} \tag{17}$$

The jump develops at $x = 0.5$ with a magnitude of 3. In order to generate the training data, we pick $20 + 21 + 20$ uniformly distributed points from the interval $[0, 1] = [0, 0.4] \cup [0.4, 0.6] \cup [0.6, 1]$.

The results are shown in Fig. 3. The hyperparameters θ_m of GPNOM after training are $(a, b, w) = (2.991, -0.5000, 2579)$, which are close to true values as we recall that a is the jump magnitude and $-b$ is the jump position. In contrast to GP with zero mean function, which gives a wrongly smoothed profile, GPNOM correctly captures the discontinuity.

4.2. Burgers' equation

For Burgers' equation (3), the exact parameter λ is set as $\lambda^* = 1$, and the viscosity $\nu = 0.01/\pi$. The reference solutions are obtained by Cole's transformation and Hermite integration [40]. As shown in Fig. 4, the wave profile is smooth near $t = 0$, and a sharp gradient occurs between $t = 0.20$ and $t = 0.40$.

First, we separately discuss the standing wave and the moving wave and compare the results of GPNOM with that of GP with zero mean function. Then we add noise to both cases to verify the effectiveness and robustness of GPNOM. We uniformly choose 101 data points $\{(x_i^k, u_i^k) | i = 1, \ldots 101\}, -1 \le x_i^k \le 1, k = 0, 1$ for each time as observation data.

Table 1
Burgers' equation - standing wave: predicted λ with noise-free data.

Δt		0.005	0.0067	0.008	0.010
$t^1 = 0.2$	GP (zero mean)	1.015588	1.020491	1.024301	1.029811
	GPNOM	1.010416	1.013749	1.016358	1.020193
$t^1 = 0.6$	GP (zero mean)	0.600958	0.602040	0.603063	0.604856
	GPNOM	0.998636	0.998026	0.997534	0.996787

4.2.1. Standing wave

We get a standing wave with the following initial condition and periodic boundary condition

$$u(0, x) = -\sin(\pi x), \quad u(t, 1) = u(t, -1).$$

We take different sample data t^1 and time step sizes Δt to test GPNOM, and list the results compared with GP (zero mean) in Table 1.

The predicted λ's well reproduce the exact value when the wave profile is smooth ($t^1 = 0.2$), with a relative error of less than 3%. While sharp gradient occurs ($t^1 = 0.6$), the GP model with zero mean function loses accuracy, and the relative errors are about 40%, whereas GPNOM still captures the correct value, even better than the case $t^1 = 0.2$. See Fig. 5(a).

The exact data at $t^1 = 0.6$ and Sigmoid mean function under trained hyperparameters are plotted in Fig. 6(a). The rest part $u^1(x) - m^1(x; \theta_m)$

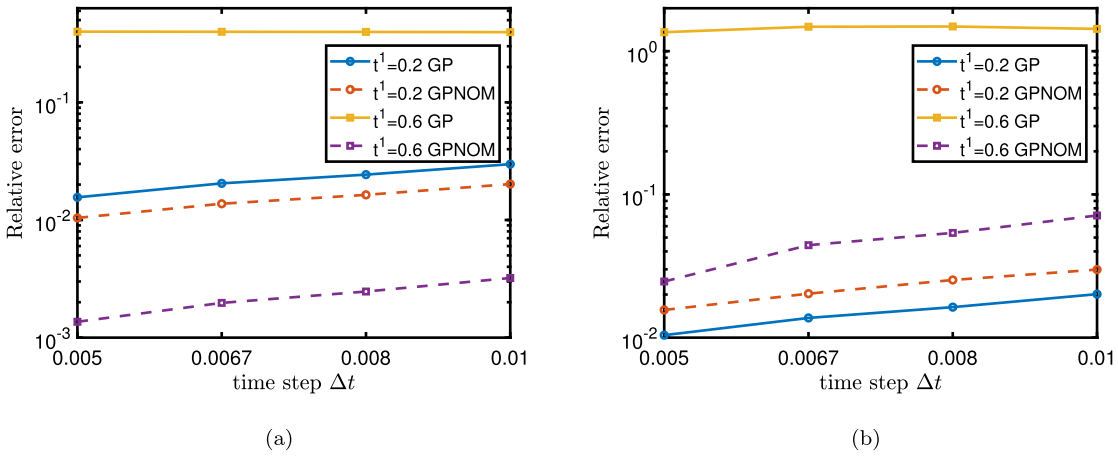

Fig. 5. Burgers' equation: relative error of predicted λ with noise-free data. (a) Standing wave. (b) Moving wave.

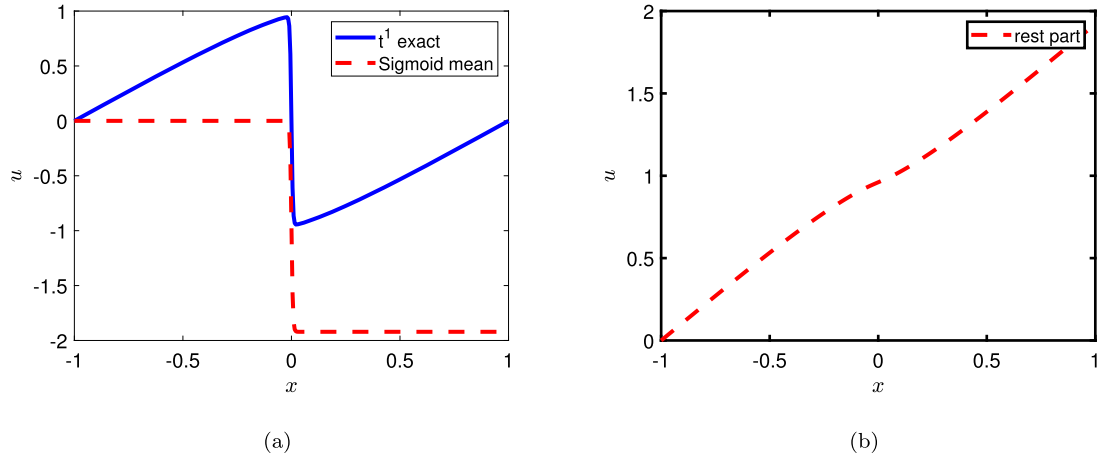

Fig. 6. Standing wave. (a) Exact data at $t^1 = 0.6$ and Sigmoid mean function. (b) The rest part $u^1(x) - m^1(x;\theta_m)$.

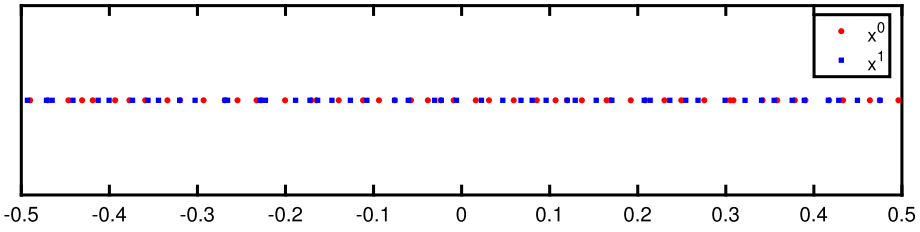

Fig. 7. Sample spatial points $\mathbf{x}^0$ and $\mathbf{x}^1$ using LHS within the interval $[-0.5, 0.5]$.

Table 2
Burgers' equation - Standing wave: predicted λ of LHS and uniform data points.

Δt		0.005	0.0067	0.008	0.010
$t^1 = 0.2$	LHS data points	1.010425	1.013797	1.016421	1.020300
	Uniform data points	1.010416	1.013749	1.016358	1.020193
$t^1 = 0.6$	LHS data points	0.994977	0.996179	0.992545	0.991682
	Uniform data points	0.998636	0.998026	0.997534	0.996787

Table 3
Burgers' equation - moving wave: predicted λ with noise-free data.

Δt		0.005	0.0067	0.008	0.010
$t^1 = 0.2$	GP (zero mean)	1.010406	1.013727	1.016326	1.020123
	GPNOM	1.015605	1.020279	1.025271	1.029859
$t^1 = 0.6$	GP (zero mean)	-0.357612	-0.481515	-0.487411	-0.431710
	GPNOM	0.975342	0.955764	0.946142	0.928716

calculated through subtracting the Sigmoid function from exact data is almost a smooth function as we claimed. See Fig. 6(b).

To show that the sample points $\mathbf{x}^0$ and $\mathbf{x}^1$ can be different, we use the Latin Hypercube Sampling (LHS) [41] to separately choose 101 data points $\{(x_i^k, u_i^k) | i = 1, ...101\}, -1 \le x_i^k \le 1, k = 0, 1$ for each time as observation data. See Fig. 7. The results have no significant difference from that of uniform data points. See Table 2.

4.2.2. *Moving wave*

We get a moving wave with the following initial condition and periodic boundary condition

$$u(0, x) = 1 - \sin(\pi x), \quad u(t, 1) = u(t, -1).$$

The results of GP (zero mean) and GPNOM with different t^1 and Δt are listed in Table 3.

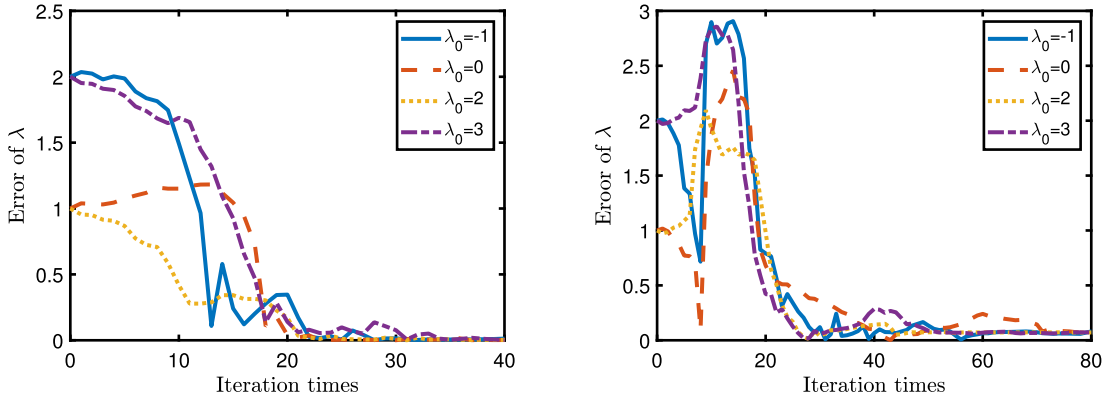

Fig. 8. Error of λ during the training process with $t^1 = 0.6$ and $\Delta t = 0.01$. (a) Standing wave. (b) Moving wave.

Table 4
Burgers' equation: hyperparameter w with noise-free data.

Δt		0.005	0.0067	0.008	0.010
$t^1 = 0.6$	Standing wave	297.7395	297.8395	297.5260	297.1894
	Moving wave	675.3984	917.0746	1156.663	1800.883

For the shock wave profile ($t^1 = 0.6$), errors of GP with zero mean function become even bigger for the case of moving wave than the standing wave, see Fig. 5(b). The standard GP does not account for sharp gradient, hence losing much information in optimizing the loss function. GPNOM has relative errors of less than 8%, which is satisfactory. We see that GPNOM addresses the problem with sharp gradient that GP can not.

For both the standing wave and the moving wave, relative error increases as time step Δt increases, which results from the numerical scheme (4). Different initial guess of parameters λ_0 are taken for the training process and λ always converges to the correct value with the iteration step increasing as shown in Fig. 8.

For the standing wave with noise-free data, the hyperparameter w of the Sigmoid mean function after training is about 297, close to $1/\nu = 100\pi$. While for the moving wave, the hyperparameter w after training has a larger value range. See Table 4.

4.2.3. Noise

In practice, measurements might be corrupted by noises. We want to test the robustness of GPNOM method by considering noises. In this subsection, Gaussian noise is added to the observation data. The magnitude of the noise is at a given percentage p of the standard deviation of noise-free data, namely,

$$\begin{aligned} \mathbf{u}^0_{\text{noisy}} &= \mathbf{u}^0 + p \cdot \text{Std}\left(\mathbf{u}^0\right) \cdot \left[(\mathcal{N}(0,1))\right]_{M\times 1}, \\ \mathbf{u}^1_{\text{noisy}} &= \mathbf{u}^1 + p \cdot \text{Std}\left(\mathbf{u}^1\right) \cdot \left[(\mathcal{N}(0,1))\right]_{N\times 1}, \end{aligned} \tag{18}$$

where Std calculates standard deviation of the data, and $\mathcal{N}(0,1)$ represents the standard Gaussian distribution. The results of GPNOM for noisy observation data from standing wave and moving wave are separately listed in Table 5 and Table 6.

We observe that, different from the case of noise-free data, where relative error increases as time step Δt increases, the relative error of predicted λ decreases as Δt increases here. See Fig. 9 and 10. That is to say, the robustness improves as the time step size increases. The reason is that the motion of the wavefront becomes easier to identify out of the noisy data when the time step size increases.

Moreover, it is natural to see that the relative error increases as the level of noise increases. See Fig. 11. We notice that most observation data falls into the shaded area restricted by $u(t^1, x) \pm p \cdot \text{Std}\left(\mathbf{u}^1\right)$, where

Table 5
Burgers' equation - standing wave: predicted λ with noisy data.

Δt		0.005	0.0067	0.008	0.010
$t^1 = 0.2$	1% noise	0.906491	0.935235	0.950560	0.967112
	2% noise	0.812468	0.864338	0.891258	0.919443
	5% noise	0.631440	0.724157	0.771405	0.819791
$t^1 = 0.6$	1% noise	1.135722	1.088918	1.065298	1.041347
	2% noise	1.377765	1.244259	1.177065	1.109342
	5% noise	1.553015	1.295692	1.168052	1.041783

Table 6
Burgers' equation - moving wave: predicted λ with noisy data.

Δt		0.005	0.0067	0.008	0.010
$t^1 = 0.2$	1% noise	1.052944	1.049708	1.044240	1.047008
	2% noise	1.094017	1.081061	1.075992	1.072695
	5% noise	1.256004	1.205807	1.182146	1.160291
$t^1 = 0.6$	1% noise	1.206469	1.113251	1.074017	1.028380
	2% noise	2.010566	1.308696	1.233805	1.153015
	5% noise	2.110147	2.417519	2.589268	1.631862

$p \cdot \text{Std}\left(\mathbf{u}^1\right)$ is the standard deviation of the added Gaussian noise as in equation (18), which means that it is difficult to distinguish the noisy observation data at t^0 and t^1.

4.3. Inviscid Burgers' equation

For inviscid Burgers' equation (16), the exact parameter λ is set as $\lambda^* = 1$. The reference solutions are obtained by the Godunov method. The wave profile is very similar to that of Burgers' equation: smooth near $t = 0$, and discontinuity occurs between $t = 0.20$ and $t = 0.40$. The difference is that the wavefront developed in the inviscid Burgers' equation is discontinuous, while the nonsingular thin transition layer developed in Burgers' equation has a width at the order of $1/\nu$. We uniformly choose 101 data points $\{(x_i^k, u_i^k) | i = 1, \ldots 101\}, -1 \le x_i^k \le 1, k = 0, n$ for each time as observation data.

4.3.1. Standing wave

We get a standing wave with the following initial condition and periodic boundary condition

$$u(0, x) = -\sin(\pi x), \quad u(t, 1) = u(t, -1).$$

We take different sample data t^1 and time step sizes Δt to test GPNOM, and list the results compared with GP in Table 7.

As shown in Fig. 12(a), when the wave profile is smooth ($t^1 = 0.2$), both the GP model with zero mean function and GPNOM have correct

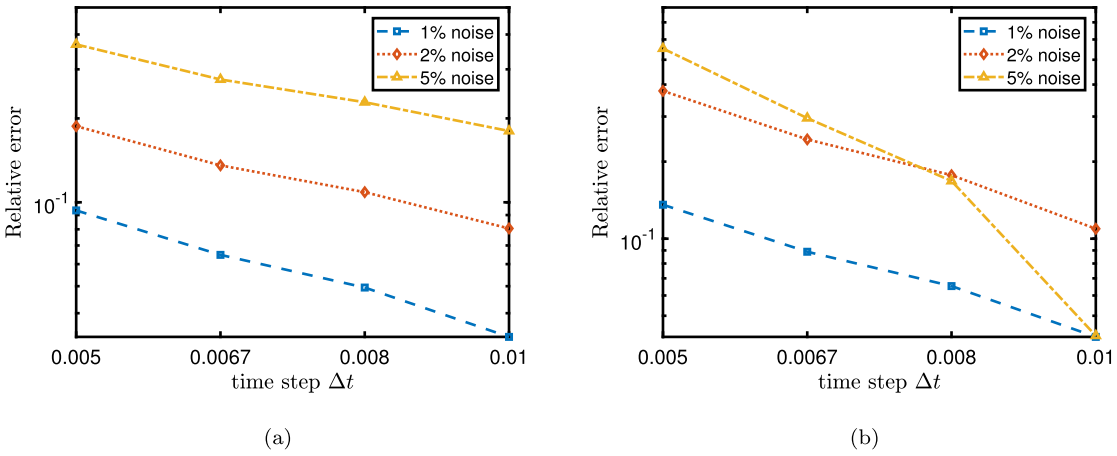

Fig. 9. Burgers' equation - standing wave: relative error of predicted λ with noisy data. (a) $t^1 = 0.2$. (b) $t^1 = 0.6$.

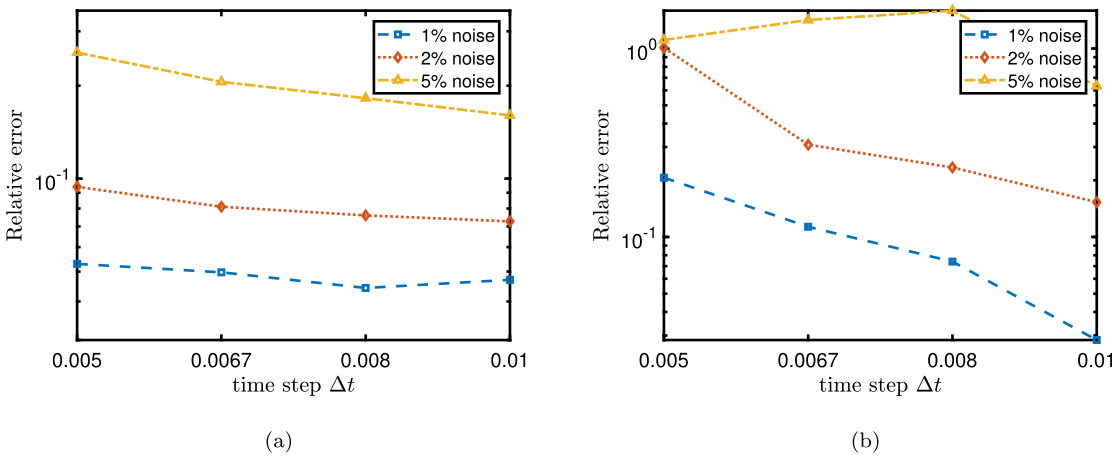

Fig. 10. Burgers' equation - moving wave: relative error of predicted λ with noisy data. (a) $t^1 = 0.2$. (b) $t^1 = 0.6$.

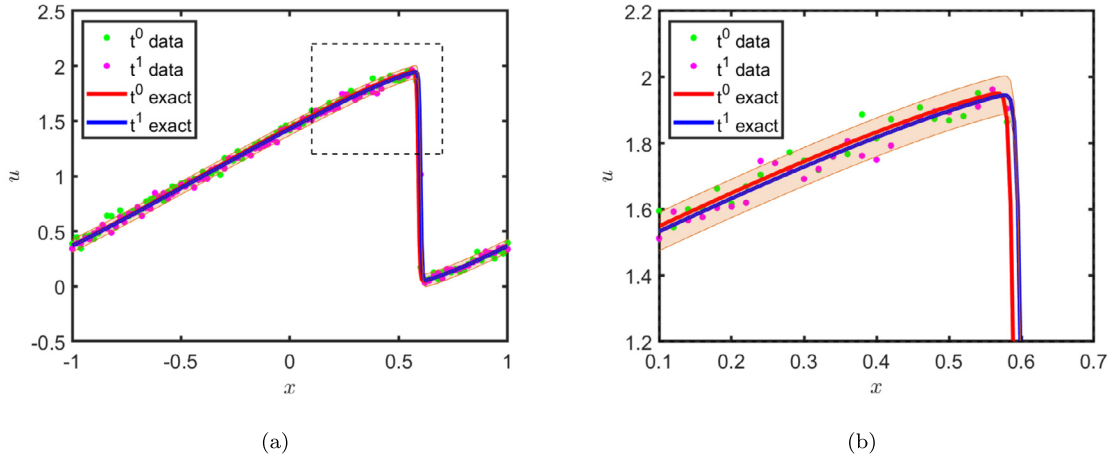

Fig. 11. Burgers' equation. (a) Noisy observation data and exact solutions at t^0 and t^1. The shaded area is restricted by $u(t^1, x) \pm p \cdot \mathrm{Std}\left(\mathbf{u}^1\right)$, and here we set the level of noise $p = 5\%$. We notice that most observation data falls into the shaded area, which means that it is difficult to distinguish the noisy observation data at t^0 and t^1. (b) Zoom in.

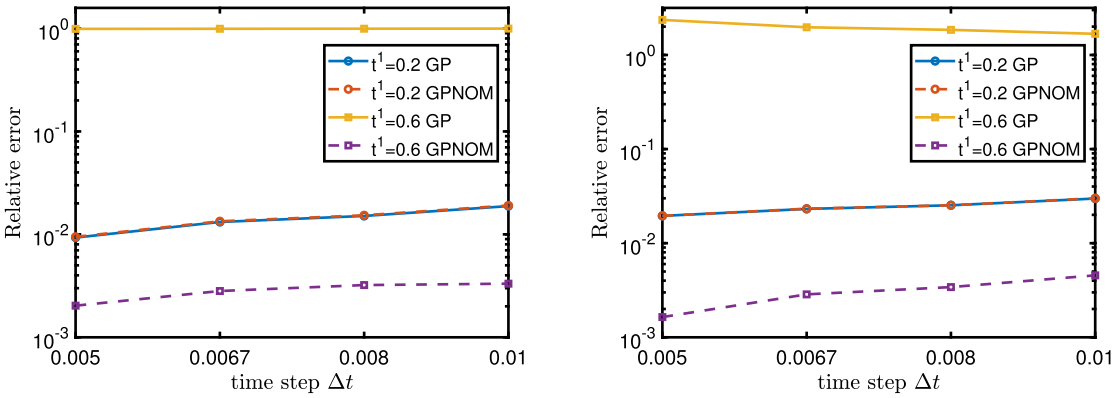

Fig. 12. Inviscid Burgers' equation: relative error of predicted λ with noise-free data. (a) Standing wave. (b) Moving wave.

Table 7
Inviscid Burgers' equation - standing wave: predicted λ with noise-free data.

Δt		0.005	0.0067	0.008	0.010
$t^1 = 0.2$	GP (zero mean)	1.009306	1.013190	1.015097	1.018813
	GPNOM	1.009466	1.013398	1.015320	1.019081
$t^1 = 0.6$	GP (zero mean)	0.008515	0.006155	0.004964	0.002565
	GPNOM	0.997971	0.997184	0.996783	0.996681

Table 8
Inviscid Burgers' equation - moving wave: predicted λ with noise-free data.

Δt		0.005	0.0067	0.008	0.010
$t^1 = 0.2$	GP (zero mean)	1.019506	1.023128	1.025278	1.029816
	GPNOM	1.019534	1.023269	1.025309	1.030034
$t^1 = 0.6$	GP (zero mean)	-1.358070	-0.969123	-0.847670	-0.677783
	GPNOM	0.998359	0.997141	0.996584	0.995447

Table 9
Inviscid Burgers' equation: hyperparameter w with noise-free data.

Δt		0.005	0.0067	0.008	0.010
$t^1 = 0.6$	Standing wave	3777.369	7299.524	11531.50	516.7663
	Moving wave	12412.30	8868.000	7760.038	6208.068

predictions and very close errors. While discontinuity occurs ($t^1 = 0.6$), the GP model with zero mean function gets totally wrong predicted values, indicating that it loses the information contained in data, whereas GPNOM still captures the correct value, even better than the case $t^1 = 0.2$.

4.3.2. Moving wave

We get a moving wave with the following initial condition and periodic boundary condition

$$u(0, x) = 1 - \sin(\pi x), \quad u(t, 1) = u(t, -1).$$

The results of GP (zero mean) and GPNOM with different t^1 and Δt are listed in Table 8.

For the shock wave profile ($t^1 = 0.6$), errors of GP with zero mean function become worse for the case of moving wave than the standing wave, while GPNOM still gets correct predicted values. See Fig. 12(b). Again we see that GPNOM addresses the problem with discontinuity that GP can not.

We further mention that compared to Burgers' equation, the hyperparameter w of the Sigmoid mean function after training is bigger because the sharp gradient is replaced by discontinuity. See Table 9.

Table 10
Inviscid Burgers' equation - standing wave: predicted λ with noisy data.

Δt		0.005	0.0067	0.008	0.010
$t^1 = 0.2$	1% noise	0.892087	0.928463	0.940523	0.958469
	2% noise	0.797950	0.860798	0.881146	0.910719
	5% noise	0.562508	0.686788	0.726262	0.782491
$t^1 = 0.6$	1% noise	1.277512	1.194742	1.167134	1.126443
	2% noise	1.563738	1.366663	1.304470	1.216688
	5% noise	1.994925	1.587796	1.460490	1.282128

Table 11
Inviscid Burgers' equation - moving wave: predicted λ with noisy data.

Δt		0.005	0.0067	0.008	0.010
$t^1 = 0.2$	1% noise	1.054458	1.049594	1.049124	1.050049
	2% noise	1.091727	1.076788	1.073186	1.069746
	5% noise	1.247380	1.191689	1.175379	1.154182
$t^1 = 0.6$	1% noise	1.212984	1.148273	1.125822	1.099155
	2% noise	1.465690	1.328597	1.285620	1.225242
	5% noise	2.414444	2.005919	1.878116	1.698967

4.3.3. Noise

From the above cases, we have seen that GPNOM also works for data with discontinuity. We further study the robustness of GPNOM with discontinuous data by considering noises. The results of GPNOM for noisy observation data from standing wave and moving wave are separately listed in Table 10 and Table 11.

We observe similar phenomena as in Subsection 4.2.3: the relative error of predicted λ decreases as Δt increases, and increases as the level of noise increases. See Fig. 13 and Fig. 14. The reason for the first one is that the motion of the wavefront becomes easier to identify out of the noisy data when the time step size increases, while the latter one is because it is difficult to distinguish the noisy observation data at t^0 and t^1.

4.4. The nonlinear wave system

GPNOM not only applies to the scalar equation but systems. We consider the nonlinear wave system, obtained either by starting with the isentropic gas dynamics equations and neglecting terms which are quadratic in the velocity or by writing the nonlinear wave equation as a first-order system [42],

$$\begin{cases} \partial_t v + \partial_x u = 0, \\ \partial_t u + \lambda \partial_x p(v) = 0, \end{cases} \tag{19}$$

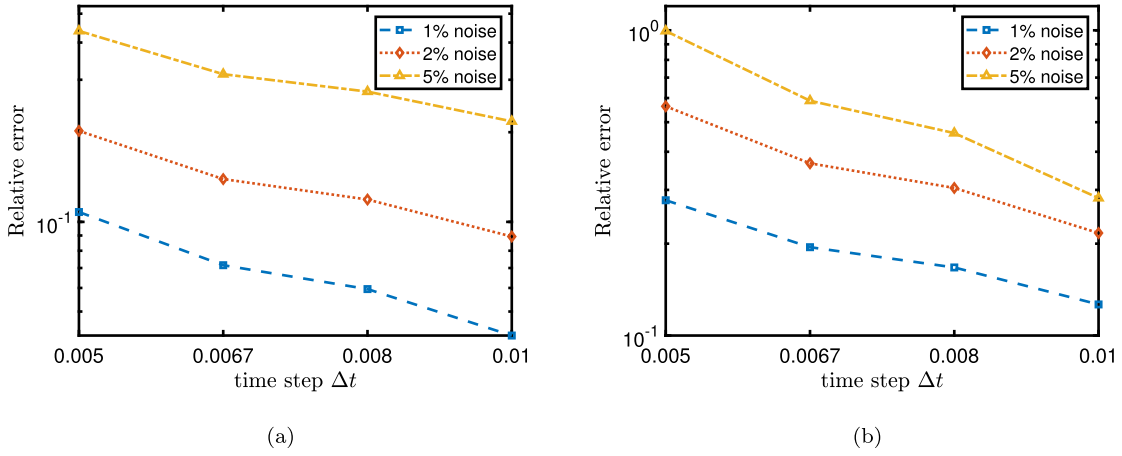

Fig. 13. Inviscid Burgers' equation - standing wave: relative error of predicted λ with noisy data. (a) $t^1 = 0.2$. (b) $t^1 = 0.6$.

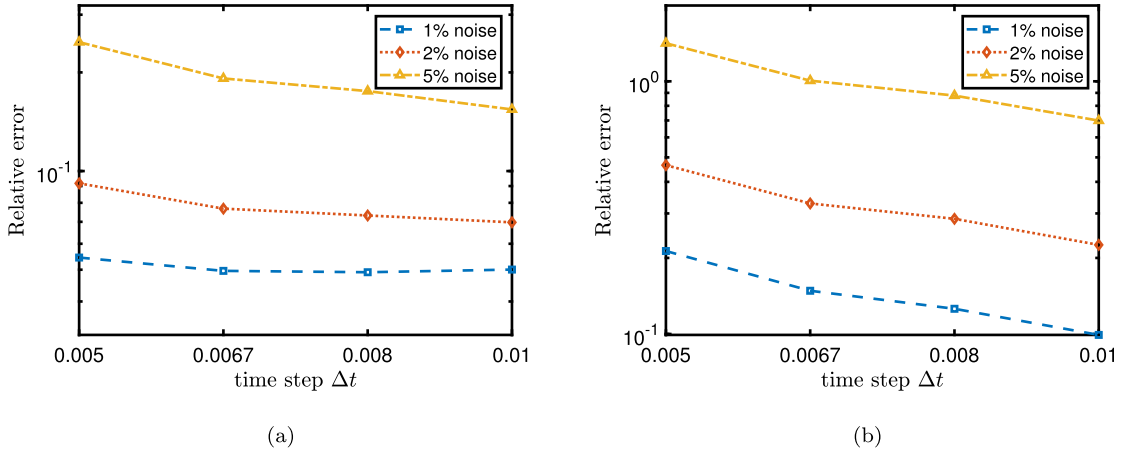

Fig. 14. Inviscid Burgers' equation - moving wave: relative error of predicted λ with noisy data. (a) $t^1 = 0.2$. (b) $t^1 = 0.6$.

Table 12
The NLWS: predicted λ with noise-free and noisy data at time $t^1 = 1$.

Δt	0.005	0.0067	0.008	0.010
0% noise	0.991567	1.019796	1.004850	1.012048
1% noise	0.978202	0.988752	0.992608	0.996296
2% noise	0.961198	1.031933	0.983908	0.996073
5% noise	0.639204	0.694239	0.704000	0.731569

where $p(v) = v^\gamma$. In all our computations we take $\gamma = 1.4$, which is the ratio of specific heats of perfect gas in aerodynamics [43]. We compute the 1-shock, 2-rarefaction solution of the Riemann problem with initial condition [44]:

$$\begin{cases} u_0(x) = \begin{cases} 0.5, & x < 0, \\ 0.60466, & x > 0, \end{cases} \\ v_0(x) = \begin{cases} 0.1, & x < 0, \\ 0.4, & x > 0. \end{cases} \end{cases} \tag{20}$$

The exact parameter λ is set as $\lambda^* = 1$. The reference solutions and data are obtained by Suliciu's method [45], and the computation has been performed on a space interval $[-2, 2]$ with a time range $[0, 1]$, see Fig. 15.

The mathematical formula derivation of GPNOM for NLWS is put in Appendix B. We uniformly choose 101 data points $\{(x_i^k, u_i^k) | i = 1, ...101\}, -2 \le x_i^k \le 0, k = 0, 1$ for each time as observation data. We take $t^1 = 1$, different time step sizes Δt, and different levels of noise to test GPNOM. See Table 12 for the results.

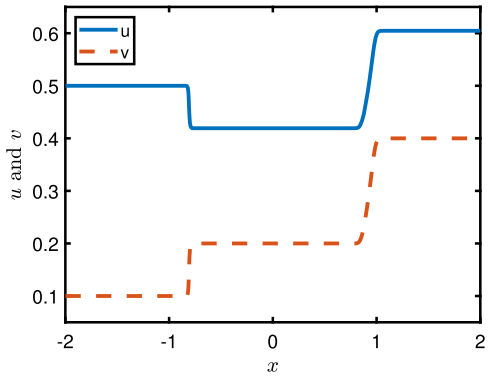

Fig. 15. Wave profiles of the solution of NLWS at time $T = 1$.

For the noise-free data, the relative error of predicted λ is less than 2%. For the noisy data, we observe that the relative error of predicted λ decreases as Δt increases here. That is to say, the robustness improves as the time step size increases, and the reason is the same as explained in Subsection 4.2.3, i.e., the motion of the wavefront becomes easier to identify out of the noisy data when the time step size increases. See Fig. 16.

5. Conclusion

In this paper, we propose a novel method to identify parameters of PDEs driven by data with sharp gradient/discontinuity, called GP-

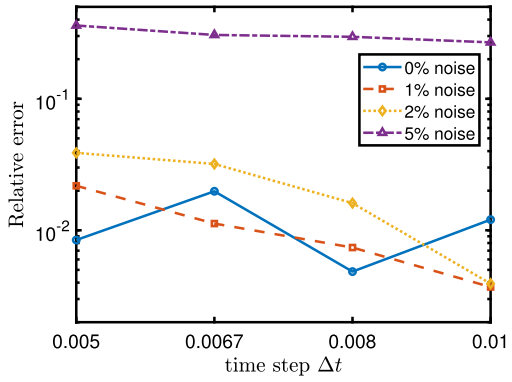

Fig. 16. NLWS: relative error of predicted λ with noise-free and noisy data at time $t^1 = 1$.

NOM. We introduce a Sigmoid function as the mean function in GP prior to capture the sharp gradient/discontinuous part. After subtracting the Sigmoid function from observation data, the rest part is of better regularity, which conventional GP can deal with.

In the numerical tests, we apply GPNOM to the regression problem, the inverse problem of Burgers' equation, and NLWS. After testing the modified GP model using noise-free data and data with different levels of noise, we verify the effectiveness and robustness of this method.

1. Effectiveness: Our modified GP model GPNOM addresses the problem when data contains sharp gradient/discontinuity, which GP with zero mean function can not.
2. Robustness: The robustness of GPNOM against noise improves as the time step increases because the motion of the wavefront becomes easier to identify out of the noisy data correspondingly.

There are some issues to be further studied. Firstly, the effectiveness of GPNOM for noisy data needs to be improved. Secondly, applied to the inverse problem of Burgers' equation, GPNOM has better performance if observation data comes from the standing wave rather than the moving wave, and the reason needs to be figured out. Besides, the sharp gradient or discontinuous cases in 2 or more dimensions are more complex. For discontinuity surface of regular shape, we may use a Sigmoid function like $m(x) = a\sigma(\alpha x + \beta y + b)$ to capture it. The challenge comes from an appropriate mathematical description/approximation of the complex shape of the discontinuity surface in real cases. This may be solved by the use of a composition of Sigmoid functions as the mean function, which will be explored in the future work.

Data availability

Data will be made available on request.

Acknowledgements

The authors would like to thank Kun Huang and Yujun Teng for their work on inverse problems with discontinuity which inspires us. This research is partially supported by the National Natural Science Foundation of China (NSFC) under Grant Nos. 11832001, 11890681 and 11988102. Lei Zhang is also supported by NSFC under Grant No. 12202451.

Appendix A. Mean functions and covariance functions of Burgers' equation

For simplicity and clarity, we use k to represent $k^{1,1}$ and m to represent m^1. The mean function m^0 in Section 3 can be derived as

$$m^0 = \mathcal{L}_x^{\lambda,\mu} m = m + \lambda\Delta t\mu(x)\frac{d}{dx}m - \nu\Delta t\frac{d^2}{dx^2}m.$$

The covariance functions $k^{1,0}, k^{0,0}$ in Section 3 can be derived as

$$k^{1,0} = \mathcal{L}_{x'}^{\lambda,\mu} k = k + \lambda\Delta t\mu(x')\frac{\partial}{\partial x'}k - \nu\Delta t\frac{\partial^2}{\partial x'^2}k$$

and

$$\begin{aligned} k^{0,0} =& \mathcal{L}_x^{\lambda,\mu}\mathcal{L}_{x'}^{\lambda,\mu} k = k + \lambda\Delta t\mu(x')\frac{\partial}{\partial x'}k - \nu\Delta t\frac{\partial^2}{\partial x'^2}k \\ &+ \lambda\Delta t\mu(x)\frac{\partial}{\partial x}k + \lambda^2\Delta t^2\mu(x)\mu(x')\frac{\partial}{\partial x}\frac{\partial}{\partial x'}k \\ &- \nu\lambda\Delta t^2\mu(x)\frac{\partial}{\partial x}\frac{\partial^2}{\partial x'^2}k - \nu\Delta t\frac{\partial^2}{\partial x^2}k \\ &- \nu\lambda\Delta t^2\mu(x')\frac{\partial^2}{\partial x^2}\frac{\partial}{\partial x'}k + \nu^2\Delta t^2\frac{\partial^2}{\partial x^2}\frac{\partial^2}{\partial x'^2}k. \end{aligned}$$

The computations can be performed using any mathematical symbolic computation program, like Wolfram Mathematica.

Appendix B. Mean functions and covariance functions of NLWS

For NLWS, employing the backward Euler time-stepping scheme, we obtain from equation (19) that

$$\begin{cases} v^1 + \Delta t u_x^1 = v^0, \\ u^1 + \lambda\Delta t\gamma(v^1)^{\gamma-1}v_x^1 = u^0. \end{cases} \tag{21}$$

The above equations can be approximated by

$$\begin{bmatrix} v^0 \\ u^0 \end{bmatrix} = \mathcal{L}_x^{\lambda,\nu}\begin{bmatrix} v^1 \\ u^1 \end{bmatrix} = \begin{bmatrix} v^1 + \Delta t u_x^1 \\ u^1 + \lambda\Delta t\gamma(\nu)^{\gamma-1}v_x^1 \end{bmatrix}, \tag{22}$$

where ν is approximated by v^0, and only uses the known values of v^0 on sampled data points in numerical calculation.

We assume $u^1(x) = u(t^1, x), v^1(x) = v(t^1, x)$ to be independent Gaussian processes

$$\begin{aligned} u^1(x) &\sim \mathcal{GP}(m_u^1(x;\theta_m^u), k_{u,u}^{1,1}(x,x';\theta_k^u)), \\ v^1(x) &\sim \mathcal{GP}(m_u^1(x;\theta_m^v), k_{v,v}^{1,1}(x,x';\theta_k^v)), \end{aligned} \tag{23}$$

with hyperparameters $\theta_m^u, \theta_k^u, \theta_m^v, \theta_k^v$. The kernel functions and the non-zero mean functions have the same form as equation (6) and equation (7).

Combine the prior assumptions (23) with equations (22), we obtain a multi-output Gaussian process

$$\begin{bmatrix} u^1 \\ v^1 \\ u^0 \\ v^0 \end{bmatrix} \sim \mathcal{GP}\left(\begin{bmatrix} m_u^1 \\ m_v^1 \\ m_u^0 \\ m_v^0 \end{bmatrix}, \begin{bmatrix} k_{u,u}^{1,1} & k_{u,v}^{1,1} & k_{u,u}^{1,0} & k_{u,v}^{1,0} \\ & k_{v,v}^{1,1} & k_{v,u}^{1,0} & k_{v,v}^{1,0} \\ & & k_{u,u}^{0,0} & k_{u,v}^{0,0} \\ sym. & & & k_{v,v}^{0,0} \end{bmatrix}\right) \tag{24}$$

with mean functions

$$m_u^0 = m_u^1 + \lambda\Delta t\gamma(\nu)^{\gamma-1}\frac{d}{dx}m_u^1,$$

$$m_v^0 = m_v^1 + \Delta t\frac{d}{dx}m_v^1,$$

and covariance functions

$$k_{u,v}^{1,1} = 0; \quad k_{u,u}^{1,0} = k_{u,u}^{1,1}, \quad k_{v,v}^{1,0} = k_{v,v}^{1,1},$$

$$k_{u,v}^{1,0} = \Delta t\frac{\partial}{\partial x'}k_{u,u}^{1,1}, \quad k_{v,u}^{1,0} = \lambda\Delta t\gamma(\nu')^{\gamma-1}\frac{\partial}{\partial x'}k_{v,v}^{1,1};$$

$$k_{u,u}^{0,0} = k_{u,u}^{1,1} + \lambda^2\Delta t^2\gamma(\nu)^{\gamma-1}\gamma(\nu')^{\gamma-1}\frac{\partial}{\partial x}\frac{\partial}{\partial x'}k_{v,v}^{1,1},$$

$$k_{u,v}^{0,0} = \Delta t\frac{\partial}{\partial x'}k_{u,u}^{1,1} + \lambda\Delta t\gamma(\nu')^{\gamma-1}\frac{\partial}{\partial x}k_{v,v}^{1,1},$$

$$k_{v,v}^{0,0} = k_{v,v}^{1,1} + \Delta t^2\frac{\partial}{\partial x}\frac{\partial}{\partial x'}k_{u,u}^{1,1}.$$

The lower triangular portion of the matrix of covariance functions is not shown due to symmetry.

Appendix C. Derivatives of the Sigmoid function

The Sigmoid function reads

$$\sigma(x) = \frac{1}{1+\exp(-x)}, \tag{25}$$

and its derivatives are as follows:

$$\sigma'(x) = \sigma(x)(1-\sigma(x)), \tag{26}$$

$$\sigma''(x) = \sigma'(x)(1-2\sigma(x)), \tag{27}$$

$$\sigma'''(x) = \sigma''(x)(1-2\sigma(x)) - 2\sigma'(x)^2. \tag{28}$$

We noticed that high-order derivatives can be computed algebraically by low-order derivatives.

References

[1] P. Perdikaris, S. Tang, Mechanistic machine learning: theory, methods, and applications, Theor. Appl. Mech. Lett. 10 (2020) 141–142.
[2] L. Ljung, Perspectives on system identification, Annu. Rev. Control 34 (1) (2010) 1–12.
[3] Y. Wang, A new concept using LSTM neural networks for dynamic system identification, in: 2017 American Control Conference (ACC), IEEE, 2017, pp. 5324–5329.
[4] P. Ni, J. Li, H. Hao, Y. Xia, X. Wang, J.-M. Lee, K.-H. Jung, Time-varying system identification using variational mode decomposition, Struct. Control Health Monit. 25 (6) (2018) e2175.
[5] L. Sersour, T. Djamah, M. Bettayeb, Nonlinear system identification of fractional Wiener models, Nonlinear Dyn. 92 (4) (2018) 1493–1505.
[6] A. Chiuso, G. Pillonetto, System identification: a machine learning perspective, Annu. Rev. Control Robot. Autonom. Syst. 2 (2019) 281–304.
[7] Z. Lai, S. Nagarajaiah, Sparse structural system identification method for nonlinear dynamic systems with hysteresis/inelastic behavior, Mech. Syst. Signal Process. 117 (2019) 813–842.
[8] S.H. Rudy, S.L. Brunton, J.L. Proctor, J.N. Kutz, Data-driven discovery of partial differential equations, Sci. Adv. 3 (4) (2017) e1602614.
[9] J.N. Kutz, S.H. Rudy, A. Alla, S.L. Brunton, Data-driven discovery of governing physical laws and their parametric dependencies in engineering, physics and biology, in: 2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), IEEE, 2017, pp. 1–5.
[10] P.A. Reinbold, R.O. Grigoriev, Data-driven discovery of partial differential equation models with latent variables, Phys. Rev. E 100 (2) (2019) 022219.
[11] Z. Wang, X. Huan, K. Garikipati, Variational system identification of the partial differential equations governing the physics of pattern-formation: inference under varying fidelity and noise, Comput. Methods Appl. Mech. Eng. 356 (2019) 44–74.
[12] M. Maslyaev, A. Hvatov, Discovery of the data-driven differential equation-based models of continuous metocean process, Proc. Comput. Sci. 156 (2019) 367–376.
[13] S. Rudy, A. Alla, S.L. Brunton, J.N. Kutz, Data-driven identification of parametric partial differential equations, SIAM J. Appl. Dyn. Syst. 18 (2) (2019) 643–660.
[14] L. Zhang, S. Tang, G. He, Learning chaotic systems from noisy data via multi-step optimization and adaptive training, Chaos, Interdiscip. J. Nonlinear Sci. 32 (12) (2022) 123134.
[15] M. Raissi, P. Perdikaris, G.E. Karniadakis, Multistep neural networks for data-driven discovery of nonlinear dynamical systems, preprint, arXiv:1801.01236, 2018.
[16] P. Goyal, P. Benner, Discovery of nonlinear dynamical systems using a Runge-Kutta inspired dictionary-based sparse regression approach, Proc. R. Soc. A 478 (2262) (2022) 20210883.
[17] L. Zhang, L. Cheng, H. Li, J. Gao, C. Yu, R. Domel, Y. Yang, S. Tang, W.K. Liu, Hierarchical deep-learning neural networks: finite elements and beyond, Comput. Mech. 67 (1) (2021) 207–230.
[18] M. Raissi, Deep hidden physics models: deep learning of nonlinear partial differential equations, J. Mach. Learn. Res. 19 (1) (2018) 932–955.
[19] Z. Long, Y. Lu, X. Ma, B. Dong, PDE-net: learning PDEs from data, in: International Conference on Machine Learning, PMLR, 2018, pp. 3208–3216.
[20] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.
[21] A.D. Jagtap, E. Kharazmi, G.E. Karniadakis, Conservative physics-informed neural networks on discrete domains for conservation laws: applications to forward and inverse problems, Comput. Methods Appl. Mech. Eng. 365 (2020) 113028.
[22] M. Raissi, P. Perdikaris, G.E. Karniadakis, Machine learning of linear differential equations using Gaussian processes, J. Comput. Phys. 348 (2017) 683–693.
[23] Y. Zhang, X. Zhu, J. Gao, Parameter estimation of acoustic wave equations using hidden physics models, IEEE Trans. Geosci. Remote Sens. 58 (7) (2020) 4629–4639.
[24] M. Raissi, P. Perdikaris, G.E. Karniadakis, Numerical Gaussian processes for time-dependent and nonlinear partial differential equations, SIAM J. Sci. Comput. 40 (1) (2018) A172–A198.
[25] M. Raissi, G.E. Karniadakis, Hidden physics models: machine learning of nonlinear partial differential equations, J. Comput. Phys. 357 (2018) 125–141.
[26] J. Chen, L. Kang, G. Lin, Gaussian process assisted active learning of physical laws, Technometrics 63 (3) (2021) 329–342.
[27] S. Lee, M. Kooshkbaghi, K. Spiliotis, C.I. Siettos, I.G. Kevrekidis, Coarse-scale pdes from fine-scale observations via machine learning, Chaos, Interdiscip. J. Nonlinear Sci. 30 (1) (2020) 013141.
[28] M. Gulian, M. Raissi, P. Perdikaris, G. Karniadakis, Machine learning of space-fractional differential equations, SIAM J. Sci. Comput. 41 (4) (2019) A2485–A2509.
[29] K.P. Murphy, Machine Learning: a Probabilistic Perspective, MIT Press, 2012.
[30] C.K.I. Williams, C.E. Rasmussen, Gaussian Processes for Machine Learning, The MIT Press, 2006.
[31] A. Pensoneault, X. Yang, X. Zhu, Nonnegativity-enforced gaussian process regression, Theor. Appl. Mech. Lett. 10 (3) (2020) 182–187.
[32] R. Calandra, J. Peters, C.E. Rasmussen, M.P. Deisenroth, Manifold Gaussian processes for regression, in: 2016 International Joint Conference on Neural Networks (IJCNN), IEEE, 2016, pp. 3338–3345.
[33] M. Raissi, G. Karniadakis, Deep multi-fidelity Gaussian processes, preprint, arXiv: 1604.07484, 2016.
[34] C. Paciorek, M.J. Schervish, Nonstationary covariance functions for Gaussian process regression, Adv. Neural Inf. Process. Syst. 16 (2003).
[35] E. Schulz, M. Speekenbrink, A. Krause, A tutorial on Gaussian process regression: modelling, exploring, and exploiting functions, J. Math. Psychol. 85 (2018) 1–16.
[36] A. Berlinet, C. Thomas-Agnan, Reproducing Kernel Hilbert Spaces in Probability and Statistics, Springer Science & Business Media, 2011.
[37] D.C. Liu, J. Nocedal, On the limited memory BFGS method for large scale optimization, Math. Program. 45 (1) (1989) 503–528.
[38] G. Falkovich, Fluid Mechanics: A Short Course for Physicists, Cambridge University Press, 2011.
[39] A.I. Forrester, A. Sóbester, A.J. Keane, Multi-fidelity optimization via surrogate modelling, Proc. R. Soc. A, Math. Phys. Eng. Sci. 463 (2088) (2007) 3251–3269.
[40] C. Basdevant, M. Deville, P. Haldenwang, J. Lacroix, J. Ouazzani, R. Peyret, P. Orlandi, A. Patera, Spectral and finite difference solutions of the Burgers equation, Comput. Fluids 14 (1) (1986) 23–41.
[41] M.D. McKay, R.J. Beckman, W.J. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 42 (1) (2000) 55–61.
[42] S. Čanić, B.L. Keyfitz, E.H. Kim, Mixed hyperbolic-elliptic systems in self-similar flows, Bol. Soc. Bras. Mat./Bull. Braz. Math. Soc. 32 (3) (2001) 377–399.
[43] J. Anderson, Fundamentals of Aerodynamics, 6th edition, McGraw-Hill Education, 2017.
[44] D. Aregba-Driollet, R. Natalini, Discrete kinetic schemes for multidimensional systems of conservation laws, SIAM J. Numer. Anal. 37 (6) (2000) 1973–2004.
[45] I. Suliciu, Some stability-instability problems in phase transitions modelled by piecewise linear elastic or viscoelastic constitutive equations, Int. J. Eng. Sci. 30 (4) (1992) 483–494.