A Reduced High-order Compact Finite Difference Scheme Based on Proper Orthogonal Decomposition for the Generalized Kuramoto-Sivashinsky Equation

Xiaohua Zhang, Ping Zhang, and Yu Ding

***Abstract*—In this paper, a reduced high-order compact finite difference scheme is proposed for numerical solution of the generalized Kuramoto-Sivashinsky equation. This approach uses implicit high-order compact finite difference scheme to attain high accuracy for generalized Kuramoto-Sivashinsky equation and combines proper orthogonal decomposition technique to improve the computational efficiency of the high-order compact finite difference scheme. The validation of the proposed method is demonstrated by four test problems. The numerical solutions are compared with the exact solutions and the solutions obtained by the corresponding high-order compact finite difference scheme. The numerical results indicate that the proposed method can largely improve the computational efficiency without a significant loss in accuracy for solving generalized Kuramoto-Sivashinsky equation compared with the corresponding high-order compact finite difference scheme.**

***Index Terms*—high-order compact finite difference scheme, proper orthogonal decomposition, generalized Kuramoto-Sivashinsky equation, computational efficiency.**

I. Introduction

AS is said in [1] the generalized Kuramoto-Sivashinsky (GKS) equation is originally derived in the context of plasma instabilities, flame front propagation, and phase turbulence in reaction-diffusion system. Thus, GKS equation can as a model for a variety of physical contexts, such as long waves on the interface between two viscous fluids, thin hydrodynamics films, thin-water-film flow on a vertical wall [2]. Accordingly, GKS is an important nonlinear evolution partial differential equation. Meanwhile, GKS is a simple partial differential equation which exhibits chaotic behavior [2]. In this paper, we shall concentrate on the numerical solution of GKS equation [3]

$$u_t + 0.5(u^2)_x + \alpha u_{xx} + \beta u_{xxx} + \gamma u_{xxxx} = 0 \qquad (1)$$

where α, β, γ are real constants. For $\beta = 0$, The equation (1) is usually called the Kuramoto-Sivashinsky (KS) equation. In generally, it is difficult to get an exact solution for the GKS equation because of its complex nonlinearity, thus, it is solved by numerical methods. Meanwhile,GKS equation is also an important model for testing various numerical algorithm. In recently, several types of numerical methods have been developed for numerical simulation of the KS equation and GKS equation. For example, Akrivis [4] presented a Crank-Nicolson-type finite difference scheme for KS equation with periodic boundary conditions. Khater et al. [5] extended the Chebyshev spectral collocation method to solve GKS equation. Mittal et al. [1] implemented quantic B-spline collocation method to find numerical solution of KS equation. Uddin et al. [2] applied radial basis function based meshfree method for the solution of KS equation. Later, Dabboura et al. [6] used moving least squares meshfree method to solve the GKS equation. Lakestani et al. [3] proposed B-spline function to solve this equation. Lai et al. [7] and Otomo et al. [8] investigated KS equation by lattice Boltzmann method. Singh et al. [9] presented the high-order compact finite difference scheme to simulate the KS equation.

In the last decade, the high-order compact finite difference scheme (CFDS) has widely been paid attention and implemented for numerical simulation of various types of partial differential equations. Such as parabolic equations [10], Burgers' equation [11], Korteweg-de Vries (KdV) equation [10], Navier-Stokes equations [12], [13], Schrödinger equation [14], Sine-Gordon equation [15], Rosenau-RLW equation [16], [17], time fractional sub-diffusion equation [18]. Although the high-order CFDS usually can get high accurate solution, it need very small time step for numerical stability consideration. Thus, high-order CFDS may be need long computational time for very long period of time of evolution. In general, the computational accuracy and computational efficiency are often the two important factors to assess a numerical algorithm, and that once the computational time is in conflict with computational accuracy, the algorithm with less time is usually given priority. Therefore, it is necessary to develop a high-order CFDS with high computational efficiency to simulate the GKS equation. In recent years, the model reduction technique such as proper orthogonal decomposition (POD) has received increasing attention in the field of computational mechanics [19]. POD, also known as Karhunen-Loève decomposition (KLD), principal component analysis (PCA) or singular value decomposition (SVD), provides a powerful technique to reduce a large number of interdependent variables to a much smaller number of

Manuscript received August 29, 2018; revised January 29, 2019. This work is financially supported by the Academic Mainstay Foundation of Hubei Province of China (No. D20171202), the National Key Research and Development Program of China (Grant No. 2017YFC0504902-05) and the Key Laboratory of Geological Hazards on Three Gorges Reservoir Area (China Three Gorges University), Ministry of Education (Grant No. 2017KDZ08).

X. Zhang is with the College of Science, China Three Gorges University, Yichang, 443002, China, e-mail: (zhangxiaohua07@163.com)

P. Zhang is with the College of Science, China Three Gorges University, Yichang, 443002, China, e-mail: (zhangping9978@126.com)

Y. Ding is with the Key Laboratory of Geological Hazards on Three Gorges Reservoir Area (China Three Gorges University) Ministry of Education, Yichang, 443002, China, e-mail: (thirdding@163.com)

uncorrelated variables while retaining as much as possible of the variation in the original variables [20]-[22]. Thus, using POD technique, the computational cost can be greatly reduced.

In the past few decades, the POD technique has been attracted wide attention and used in the numerical solution to construct some reduced models. For example, Lou et al. coupled POD technique with finite difference method, finite element method and finite volume method to solve the parabolic equations [23]-[25], Burgers' equation [26], Navier-Stokes equations [27], [28], and hyperbolic equations [29]. Bialecki et al. [30] used the finite element method and POD to solve transient thermal problems. Bill et al. [31] coupled finite difference method with POD to solve transient mass transport problems. Zhang et al. had proposed a fast and efficient meshless method based on POD for solving transient heat conduction problems [32] and convection-diffusion problems [19]. Recently, Dehghan et al. used POD and radial basis function meshfree methods to solve groundwater equation[33] and shallow water equations [34]. However, to our best knowledge, there are no published results when POD is used to reduce the implicit high-order compact finite difference scheme (CFDS) for GKS equation. The main goal of this paper is to construct a numerical algorithm which has high computational accuracy and efficiency for solving GKS equation. Thus, the focus of the present paper is on combining the CFDS and the POD method, namely the CFDS&POD, to solve GKS equation.

The organization of this paper is as follows. In Section II, a brief background is given on the theoretical foundations of high-order CFDS and POD technique. Then, the CFDS&POD is explained for the GKS equation. In Section III, four test numerical examples are presented to demonstrate the capabilities and potential of the proposed method. Also the comparisons of the global relative error at different time and the computational time of CFDS and CFDS&POD method are discussed. A summary is given at the end of the paper in Section IV.

II. Numerical algorithms

In this section, we first briefly review an implicit high-order CFDS and POD technique, then based on CFDS and POD, the CFDS&POD for solving GKS equation is constructed.

A. An implicit high-order CFDS for GKS equation

To gain the solution of the GKS equation, discretizations are needed in both space and time. In the high-order CFDS, one can obtain all the numerical derivatives along a grid line using small stencils and solving a linear system of equations [10], that is, the derivatives of u are obtained by solving a tridiagonal or pentadiagonal system for any scalar value u . More details on how to derive such formulae can be found in [10], [13].

For simplicity, we consider a uniform one-dimensional mesh which consisting of N nodes:$x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_N$. The mesh size is denoted by $h = x_{i+1} - x_i$. Because the GKS equation (1) contains the first-order to fourth-order spatial derivatives, in the following, we will list final formulas of an implicit high-order compact finite difference scheme for first-order derivative to fourth-order derivative.

For the first-order derivative at interior nodes, we have the formula [10]

$$\alpha u'_{i-1} + u'_i + \alpha u'_{i+1} = b\frac{u_{i+2} - u_{i-2}}{4h} + a\frac{u_{i+1} - u_{i-1}}{2h} \quad (2)$$

a simple sixth-order tridiagonal scheme is given by the coefficients [10]

$$\alpha = \frac{1}{3}, a = \frac{14}{9}, b = \frac{1}{9} \quad (3)$$

For those near boundary nodes, approximation formulas for the derivatives of non-periodic problems can be derived by one-sided schemes. More details about the derivations for the first derivative can be referenced in [10], [13].

At boundary point 1, the sixth-order formula for the first-order derivative is [10], [13]

$$\begin{gathered} u'_1 + 5u'_2 = \tfrac{1}{h}(-\tfrac{197}{60}u_1 - \tfrac{5}{12}u_2 + \\ 5u_3 - \tfrac{5}{3}u_4 + \tfrac{5}{12}u_5 - \tfrac{1}{20}u_6) \end{gathered} \quad (4)$$

At boundary point 2, the sixth-order formula for the first-order derivative is [10], [13]

$$\begin{gathered} \tfrac{2}{11}u'_1 + u'_2 + \tfrac{2}{11}u'_3 = \tfrac{1}{h}(-\tfrac{20}{33}u_1 - \\ \tfrac{35}{132}u_2 + \tfrac{34}{33}u_3 - \tfrac{7}{33}u_4 + \tfrac{2}{33}u_5 - \tfrac{1}{132}u_6) \end{gathered} \quad (5)$$

At boundary point $N-1$, the sixth-order formula for the first-order derivative is [10], [13]

$$\begin{gathered} \tfrac{2}{11}u'_{N-2} + u'_{N-1} + \tfrac{2}{11}u'_N = \tfrac{1}{h}(\tfrac{20}{33}u_N + \tfrac{35}{132}u_{N-1} \\ -\tfrac{34}{33}u_{N-2} + \tfrac{7}{33}u_{N-3} - \tfrac{2}{33}u_{N-4} + \tfrac{1}{132}u_{N-5}) \end{gathered} \quad (6)$$

At boundary point N, the sixth-order formula for the first-order derivative is [10], [13]

$$\begin{gathered} 5u'_{N-1} + u'_N = \tfrac{1}{h}(\tfrac{197}{60}u_N + \tfrac{5}{12}u_{N-1} - \\ 5u_{N-2} + \tfrac{5}{3}u_{N-3} - \tfrac{5}{12}u_{N-4} + \tfrac{1}{20}u_{N-5}) \end{gathered} \quad (7)$$

For the second-order derivative at interior nodes, one can derive the formula [9], [10]

$$\begin{gathered} \alpha u''_{i-1} + u''_i + \alpha u''_{i+1} = \\ b\tfrac{u_{i+2}-2u_i+u_{i-2}}{4h^2} + a\tfrac{u_{i+1}-2u_i+u_{i-1}}{h^2} \end{gathered} \quad (8)$$

which provides a $\alpha-$family of fourth-order triadiagonal schemes with $a = \frac{4}{3}(1-\alpha)$, $b = \frac{1}{3}(-1+10\alpha)$. When $\alpha = \frac{2}{11}$, the scheme becomes sixth-order accurate [10]. But in the paper, we choose $\alpha = \frac{1}{10}$ because that sixth-order scheme is not convergence in our numerical experiments. For $\alpha = \frac{1}{10}$, Eq. (8) reduced to [9]

$$\frac{1}{10}u''_{i-1} + u''_i + \frac{1}{10}u''_{i+1} = \frac{12}{10h^2}(u_{i+1} - 2u_i + u_{i-1}) \quad (9)$$

Similar to the scheme for first-order derivative at boundary point 1, the fourth-order formula for second-order derivative is [9]

$$\begin{gathered} u''_1 + \tfrac{1}{10}u''_2 = \tfrac{12}{10h^2}(\tfrac{115}{36}u_1 - \tfrac{1555}{144}u_2 \\ +\tfrac{89}{6}u_3 - \tfrac{773}{72}u_4 + \tfrac{151}{36}u_5 - \tfrac{11}{16}u_6) \end{gathered} \quad (10)$$

At boundary point N, the fourth-order formula for second-order derivative is [9]

$$\begin{gathered} \tfrac{1}{10}u''_{N-1} + u''_N = \tfrac{12}{10h^2}(\tfrac{115}{36}u_N - \tfrac{1555}{144}u_{N-1} \\ +\tfrac{89}{6}u_{N-2} - \tfrac{773}{72}u_{N-3} + \tfrac{151}{36}u_{N-4} - \tfrac{11}{16}u_{N-5}) \end{gathered} \quad (11)$$

The matrix representation of the scheme for first-order and second-order derivatives are given as follows

$$\mathbf{B}_x\mathbf{u}' = \mathbf{A}_x\mathbf{u} \tag{12}$$

$$\mathbf{B}_{xx}\mathbf{u}'' = \mathbf{A}_{xx}\mathbf{u} \tag{13}$$

where

$$\mathbf{u} = (u_1, u_2, \ldots, u_N)^T \tag{14}$$

$$\mathbf{B}_x = \begin{pmatrix} 1 & 5 & & & & & \\ \frac{2}{11} & 1 & \frac{2}{11} & & & & \\ & \frac{1}{3} & 1 & \frac{1}{3} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \frac{1}{3} & 1 & \frac{1}{3} & \\ & & & & \frac{2}{11} & 1 & \frac{2}{11} \\ & & & & & 5 & 1 \end{pmatrix} \tag{15}$$

$$\mathbf{A}_x = \frac{1}{h}\begin{pmatrix} \frac{-197}{60} & \frac{-5}{12} & 5 & \frac{-5}{3} & \frac{5}{12} & \frac{-1}{20} & \\ \frac{-20}{33} & \frac{-35}{132} & \frac{34}{33} & \frac{-7}{33} & \frac{2}{33} & \frac{-1}{132} & \\ \frac{-1}{36} & \frac{-7}{9} & 0 & \frac{7}{9} & \frac{1}{36} & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & \frac{-1}{36} & \frac{-7}{9} & 0 & \frac{7}{9} & \frac{1}{36} \\ & \frac{1}{132} & \frac{-2}{33} & \frac{7}{33} & \frac{-34}{33} & \frac{35}{132} & \frac{20}{33} \\ & \frac{1}{20} & \frac{-5}{12} & \frac{5}{3} & -5 & \frac{5}{12} & \frac{197}{60} \end{pmatrix} \tag{16}$$

$$\mathbf{B}_{xx} = \begin{pmatrix} 1 & \frac{1}{10} & & & & \\ \frac{1}{10} & 1 & \frac{1}{10} & & & \\ & \ddots & \ddots & \ddots & & \\ & & \frac{1}{10} & 1 & \frac{1}{10} & \\ & & & \frac{1}{10} & 1 & \end{pmatrix} \tag{17}$$

$$\mathbf{A}_{xx} = \frac{12}{10h^2}\begin{pmatrix} \frac{115}{36} & \frac{-1555}{144} & \frac{89}{6} & \frac{-773}{72} & \frac{151}{36} & \frac{-11}{16} & \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -2 & 1 \\ & \frac{-11}{16} & \frac{151}{36} & \frac{-773}{72} & \frac{89}{6} & \frac{-1555}{144} & \frac{115}{36} \end{pmatrix} \tag{18}$$

As far as third-order and fourth-order derivatives in GKS equation are concerned, we also can use Taylor expansion to deduce high-order implicit CFDS, and this may be tedious and labour-consuming. Here, we approximate the third-order and fourth-order derivatives by formula (12) and (13) directly, i.e.

$$\mathbf{B}_x\mathbf{u}''' = \mathbf{A}_x\mathbf{u}'' \tag{19}$$

$$\mathbf{B}_{xx}\mathbf{u}^{(4)} = \mathbf{A}_{xx}\mathbf{u}'' \tag{20}$$

After the spatial derivative is discretized by the compact scheme (12), (13), (19) and (20), we obtain a system of initial value problem of ordinary differential equations (ODEs),

$$\frac{d\mathbf{u}}{dt} = L(\mathbf{u}) \tag{21}$$

where the operator $L(\mathbf{u})$ denotes the residual. This set of ODEs can be discretized by a third-order TVD Runge-Kutta (TVD-RK3) method, which is given as follows [11]:

$$\begin{aligned} \mathbf{u}^{(1)} &= \mathbf{u}^n + \Delta t L(\mathbf{u}^n), \\ \mathbf{u}^{(2)} &= \tfrac{3}{4}\mathbf{u}^n + \tfrac{1}{4}\mathbf{u}^{(1)} + \tfrac{1}{4}\Delta t L(\mathbf{u}^{(1)}), \\ \mathbf{u}^{n+1} &= \tfrac{1}{3}\mathbf{u}^n + \tfrac{2}{3}\mathbf{u}^{(2)} + \tfrac{2}{3}\Delta t L(\mathbf{u}^{(2)}). \end{aligned} \tag{22}$$

and thus, the solutions of $u(x,t)$ at the required time level are obtained.

Other higher order versions of time discretization method such as the fourth-order four-stage Runge-Kutta (RK4) method can be also applied.

From the above, we can give solving process of the CFDS for GKS equation in one time step, that is, the first-order spatial derivative in Eq. (1) is obtained by solving Eq. (12), the second-order spatial derivative is obtained by solving Eq. (13), third-order derivative is obtained by solving Eq. (19) and fourth-order derivative uses Eq. (20). Then, the TVD-RK3 method is applied to approximate the corresponding semi-discrete equation.

B. The review of POD technique

We briefly describe the POD method, following [19]. For a detailed presentation, the reader can refer to [20]-[29] . Meanwhile, a detailed discussion about the equivalence of the POD, KLD, PCA and SVD can be referred to [20]-[22].

This section has mainly been taken from [32]. The main idea of the POD is to find a set of ordered orthonormal basis vectors in a subspace where a random vector takes its values, such that the samples in the sample space can be expressed optimally using the selected first k basis vectors [20]. In the paper, we use SVD to construct the optimal basis. According to the POD theory in [23], the high dimension data generally rely on the use of a sequence of snapshots to build a low-dimensional discretized system. Thus, the fundamental notion of POD is the snapshots which can be obtained from either the numerical simulation or experiments. The set of snapshots can be expressed as a $N \times d$ matrix $\mathbf{T}_s$ as follows

$$\mathbf{T}_s = \left(\mathbf{T^1}, \mathbf{T^2}, \ldots, \mathbf{T}^d\right) \tag{23}$$

where the columns of $\mathbf{T}_s$ represent snapshots, d is the number of the snapshots.

In the following, we first generate a group of optimal basis from the set of snapshots $\mathbf{T}_s$. In order to construct the optimal POD basis, we use SVD method, which can be viewed as the extension of the eigenvalue decomposition for the case of non-square matrices. Using the SVD on matrix $\mathbf{T_s}$, we have

$$\mathbf{T}_s = \mathbf{U}\begin{pmatrix} \mathbf{D}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}\mathbf{V}^T \tag{24}$$

where $\mathbf{U} = \mathbf{U}_{N\times N}$ and $\mathbf{V} = \mathbf{V}_{d\times d}$ are orthogonal matrices,$\mathbf{D}_r = diag(\lambda_1, \lambda_2, \ldots, \lambda_r)$. The matrix $\mathbf{U} = (\mathbf{\Psi}_1, \mathbf{\Psi}_2, \ldots, \mathbf{\Psi}_N)$contains the orthogonal eigenvectors of $\mathbf{T}_s\mathbf{T}_s^T$, while the singular values $\lambda_i (i = 1, 2, \ldots, r)$ satisfy $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r > 0$.

If we denote d columns of $\mathbf{T}_s$ by $\mathbf{a}^l = (T_1^l, T_2^l, \ldots, T_N^l)^T (l = 1, 2, \ldots, d)$, and define a projection

P_k as follows

$$P_k(\mathbf{a}^l) = \sum_{i=1}^{k} (\mathbf{\Psi}_i, \mathbf{a}^l)\mathbf{\Psi}_i \tag{25}$$

where $0 < k \leq d$ and $(.,.)$ denotes the inner product of vectors, then according to [23], one has the following result:

$$\|\mathbf{a}^l - P_k(\mathbf{a}^l)\|_2 \leq \lambda_{k+1} \tag{26}$$

where $\|.\|_2$ is standard norm of vector. Therefore, $\mathbf{\Psi}_1, \mathbf{\Psi}_2, \ldots, \mathbf{\Psi}_k$ are a group of the optimal POD basis and stored in basis matrix $\mathbf{\Phi} = (\mathbf{\Psi}_1, \mathbf{\Psi}_2, \ldots, \mathbf{\Psi}_k)$. Obviously, the basis matrix fulfills the orthogonality condition, i.e., $\mathbf{\Phi}^T\mathbf{\Phi} = \mathbf{I}$ ($\mathbf{I}$ is unit matrix of dimension k).

C. The CFDS&POD method for GKS equation

In the following, we combine CFDS with POD method to derive CFDS&POD for GKS equation.

If $\mathbf{u}$ of Eqs. (12), (13), (19) and (20) are substituted for

$$\mathbf{u}^* = \mathbf{\Phi}\mathbf{W} = \mathbf{\Phi}_{N\times k}(\mathbf{W})_{k\times 1}, \tag{27}$$

we have

$$\mathbf{B}_x\mathbf{\Phi}\mathbf{W}' = \mathbf{A}_x\mathbf{\Phi}\mathbf{W} \tag{28}$$

$$\mathbf{B}_{xx}\mathbf{\Phi}\mathbf{W}'' = \mathbf{A}_{xx}\mathbf{\Phi}\mathbf{W} \tag{29}$$

$$\mathbf{B}_x\mathbf{\Phi}\mathbf{W}''' = \mathbf{A}_x\mathbf{\Phi}\mathbf{W}'' \tag{30}$$

and

$$\mathbf{B}_{xx}\mathbf{\Phi}\mathbf{W}^{(4)} = \mathbf{A}_{xx}\mathbf{\Phi}\mathbf{W}'' \tag{31}$$

Multiplying Eqs. (28)-(31) by $\mathbf{\Phi}^T$ from left, we get

$$(\mathbf{\Phi}^T\mathbf{B}_x\mathbf{\Phi})_{k\times k}(\mathbf{W}')_{k\times 1} = (\mathbf{\Phi}^T\mathbf{A}_x\mathbf{\Phi})_{k\times k}(\mathbf{W})_{k\times 1} \tag{32}$$

$$(\mathbf{\Phi}^T\mathbf{B}_{xx}\mathbf{\Phi})_{k\times k}(\mathbf{W}'')_{k\times 1} = (\mathbf{\Phi}^T\mathbf{A}_{xx}\mathbf{\Phi})_{k\times k}(\mathbf{W})_{k\times 1} \tag{33}$$

$$(\mathbf{\Phi}^T\mathbf{B}_x\mathbf{\Phi})_{k\times k}(\mathbf{W}''')_{k\times 1} = (\mathbf{\Phi}^T\mathbf{A}_x\mathbf{\Phi})_{k\times k}(\mathbf{W}'')_{k\times 1} \tag{34}$$

and

$$(\mathbf{\Phi}^T\mathbf{B}_{xx}\mathbf{\Phi})_{k\times k}(\mathbf{W}^{(4)})_{k\times 1} = (\mathbf{\Phi}^T\mathbf{A}_{xx}\mathbf{\Phi})_{k\times k}(\mathbf{W}'')_{k\times 1} \tag{35}$$

Similarly, if $\mathbf{u}$ of (22) is substituted for

$$\mathbf{u}^{*n} = \mathbf{\Phi}\mathbf{W}^n = \mathbf{\Phi}_{N\times k}(\mathbf{W}^n)_{k\times 1}, \quad n = 0, 1, 2, \ldots \tag{36}$$

and noting that $\mathbf{\Phi}^T\mathbf{\Phi} = \mathbf{I}$, we obtain TVD-RK3 for the reduced solution as follows

$$\begin{aligned} \mathbf{W}^{(1)} &= \mathbf{W}^n + \Delta t L(\mathbf{W}^n), \\ \mathbf{W}^{(2)} &= \tfrac{3}{4}\mathbf{W}^n + \tfrac{1}{4}\mathbf{W}^{(1)} + \tfrac{1}{4}\Delta t L(\mathbf{W}^{(1)}), \\ \mathbf{W}^{n+1} &= \tfrac{1}{3}\mathbf{W}^n + \tfrac{2}{3}\mathbf{W}^{(2)} + \tfrac{2}{3}\Delta t L(\mathbf{W}^{(2)}). \end{aligned} \tag{37}$$

where $\mathbf{W}^0 = \mathbf{\Phi}^T\mathbf{u}^0 = \mathbf{\Phi}^T(u_1^0, u_2^0, \ldots, u_N^0)$, Once reduced solution $\mathbf{W}^{n+1}$ is obtained from Eq. (37), one can obtain the global solution $\mathbf{u}^{n+1} = \mathbf{\Phi}\mathbf{W}^{n+1}$.

Here, we summarize the CFDS&POD algorithm for GKS equation as follows:

(1) Generate the snapshots (samples) ensemble $\mathbf{T}_s$;

(2) Use SVD method to obtain the optimal POD basis matrix $\mathbf{\Phi}$;

(3) Solve reduced Eqs. (32) and (33) to get the reduced first-order derivative $\mathbf{W}'$ and second-order derivative $\mathbf{W}''$;

(4) Solve reduced Eqs. (34) and (35) to get the reduced third-order derivative $\mathbf{W}'''$ and fourth-order derivative $\mathbf{W}^{(4)}$;

(5) Solve the Eq. (37) and get the reduced solution $\mathbf{W}^{n+1}$;

(6) Expand the reduced solution to the global solution: $\mathbf{u}^{n+1} = \mathbf{\Phi}\mathbf{W}^{n+1}$.

Form the above algorithm, it can be clearly found that the CFDS&POD only solves $k \times k$ system equations (Eqs. (32) to (35)) at each time loop, while CFDS scheme needs to solve $N \times N$ system equations (Eqs. (12), (13), (19) and (20)) at each time loop. In general, N is much larger than k , which implies that CFDS&POD requires less computational time than that of CFDS, although the SVD processes needs some extra expense.

III. Numerical examples and discussion

In this section, to test the CFDS&POD proposed in the above section, numerical simulations of GKS and GS equations are performed. In comparison with the analytical solutions and CFDS solutions, the accuracy and efficiency of CFDS&POD are validated. To illustrate accuracy of the method, we compute the global relative error (GRE) which is defined as follows [8], [9]

$$GER = \frac{\sum_i |u(x_i, t) - u^*(x_i, t)|}{\sum_i |u^*(x_i, t)|} \tag{38}$$

where $u(x_i, t), u^*(x_i, t)$ are numerical solution and exact solution, respectively.

Example 1 In this example, we first consider the GS equation, represented by $\alpha = \gamma = 1$ and $\beta = 0$. The exact solution is [1]-[3],[5]-[9]

$$\begin{aligned} u^*(x,t) = b + \tfrac{15}{19}\sqrt{\tfrac{15}{19}}[\tanh(\kappa(x - bt - x_0)) + \\ 11\tanh^3(\kappa(x - bt - x_0))] \end{aligned} \tag{39}$$

where $b = 5, \kappa = \frac{1}{2}\sqrt{\frac{11}{19}}$ and $x_0 = -12$. We will use this solution, evaluated at $t = 0$, as the initial condition, and the boundary conditions correspond to the data from the exact solution, too. The computational domain is fixed on the interval $[-30, 30]$. The obtained solutions and point-wise absolute errors of CFDS with 121 uniformly distributed points at time $t = 1, 2, 3$ and 4 are shown in Fig. 1. Meanwhile, we also plot the corresponding numerical results of CFDS&POD with 30 POD bases for comparison purposes (see Fig. 2). In our computations, the time step $\Delta t = 0.001$. It not difficult to see that the results of CFDS&POD are in very good agreement with those of CFDS and exact solutions. In order to make further improvement on the computational accuracy of CFDS&POD, the comparison of the global relative errors obtained by CFDS and CFDS&POD at different time levels $t \leq 4$ taking different nodes and time steps are listed in Tables I and II. It can be seen that the accuracy of CFDS&POD is almost identical with that of CFDS under the same nodes and time steps. Table III shows the computational time of CFDS and CFDS&POD at $t = 4$ for different nodes. It can be seen that under the same number of nodes, the computational times of CFDS&POD are less than those of CFDS, especially the more nodes, the higher computational efficiency is obtained.

Example 2 Consider Eq. (1) with $\alpha = -1, \beta = 0$ and $\gamma = 1$. The exact solution of the problem is given by [1]-

TABLE I
THE GLOBAL RELATIVE ERRORS OF CFDS AT DIFFERENT NODES AND TIMES FOR EXAMPLE 1

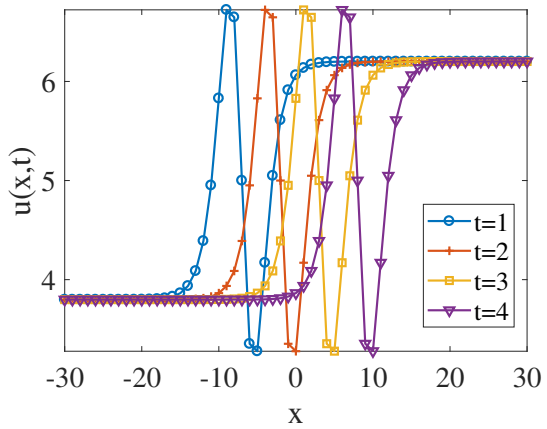
node numbers	time step	CFDS			
		$t=1$	$t=2$	$t=3$	$t=4$
61	1E-2	4.04E-4	5.33E-4	7.77E-4	1.10E-3
121	1E-3	2.39E-5	3.42E-5	4.64E-5	6.26E-5
241	1E-4	1.61E-6	2.23E-6	3.07E-6	4.12E-6

TABLE II
THE GLOBAL RELATIVE ERRORS OF CFDS&POD AT DIFFERENT NODES AND TIMES FOR EXAMPLE 1

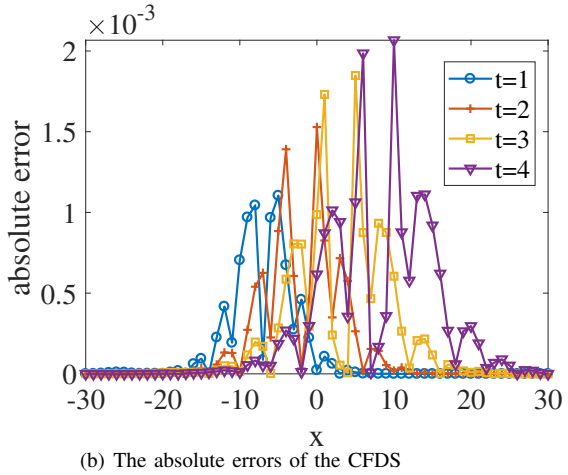
node numbers	time step	CFDS&POD			
		$t=1$	$t=2$	$t=3$	$t=4$
61	1E-2	4.15E-4	5.44E-4	7.85E-4	1.10E-4
121	1E-3	1.23E-4	1.24E-4	1.33E-4	1.76E-4
241	1E-4	8.17E-5	8.26E-5	8.06E-5	9.28E-5

TABLE III
COMPARE THE COMPUTATIONAL TIME AT $t=4$ WITH DIFFERENT NUMBER OF NODES FOR EXAMPLE 1

node numbers	time step	computational time(second)	
		CFDS	CFDS&POD
61	1E-2	0.335781	0.185879
121	1E-3	5.78383	1.296004
241	1E-4	192.504437	14.911494

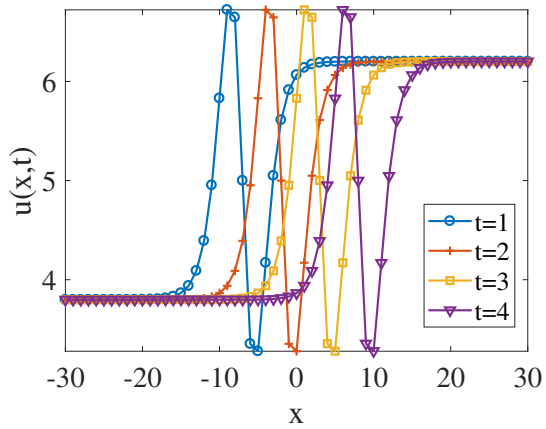


(a) The solutions of the CFDS

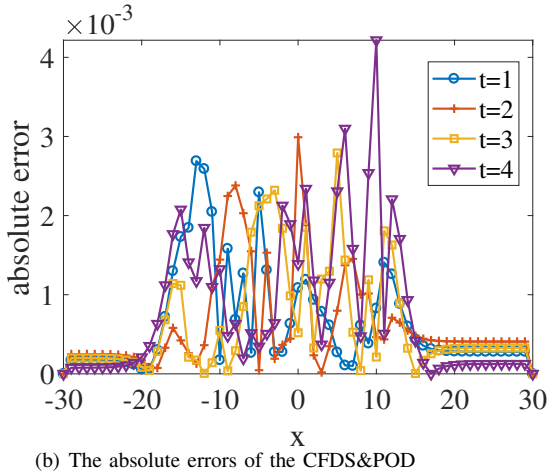


(b) The absolute errors of the CFDS

Fig. 1. The solutions and absolute errors of CFDS at $t=1,2,3$ and 4 with 121 nodes for Example 1.



(a) The solutions of the CFDS&POD



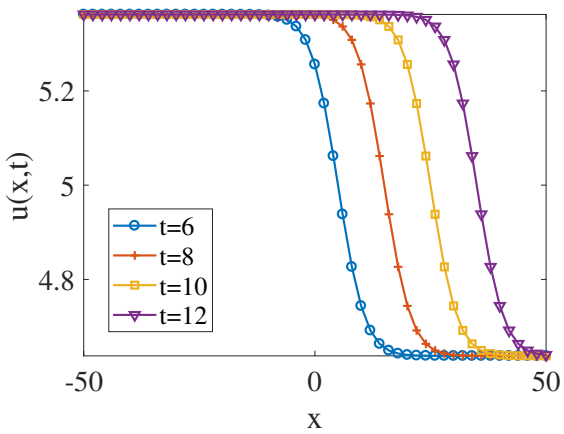
(b) The absolute errors of the CFDS&POD

Fig. 2. The solutions and absolute errors of CFDS&POD at $t=1,2,3$ and 4 with 121 nodes for Example 1.

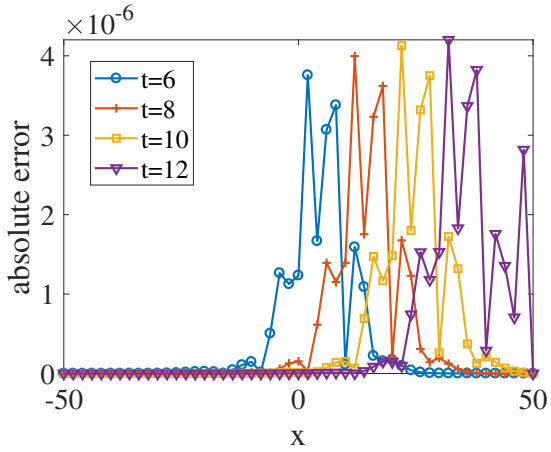
[3],[5]-[7]

$$u^*(x,t) = b + \frac{15}{19\sqrt{19}}[-3\tanh(\kappa(x-bt-x_0)) + \tanh^3(\kappa(x-bt-x_0))] \quad (40)$$

The following parameters have been used:$b=5, \kappa=\frac{1}{2\sqrt{19}}$ and $x_0=-25$.

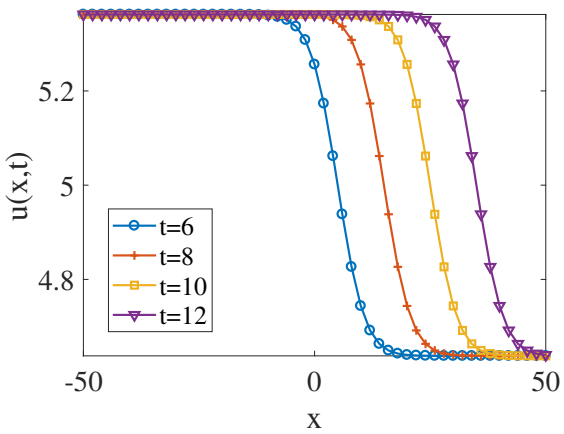


(a) The solutions of the CFDS

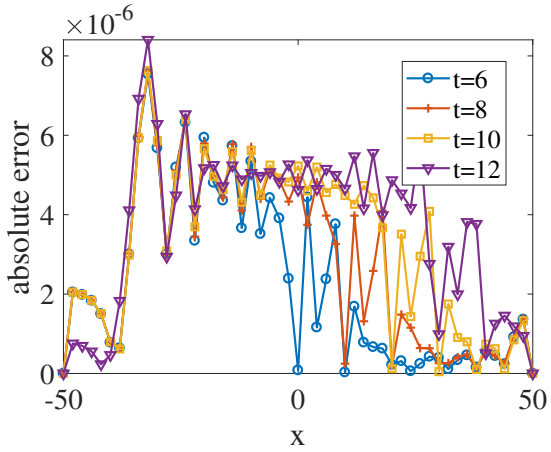


(b) The absolute errors of the CFDS

Fig. 3. The solutions and absolute errors of CFDS at $t = 6, 8, 10$ and 12 with 101 nodes for Example 2.



(a) The solutions of the CFDS&POD



(b) The absolute errors of the CFDS&POD

Fig. 4. The solutions and absolute errors of CFDS&POD at $t = 6, 8, 10$ and 12 with 101 nodes for Example 2.

In this example, the computational domain is fixed on the interval $[-50, 50]$. Similar to the Example 1, we also consider three different node numbers, that is, 101, 201 and 401 nodes uniformly distributed in the computational domain. Fig. 3 and Fig. 4 plot the numerical solutions and point-wise absolute errors of CFDS and CFDS&POD with 30 POD bases at time $t = 6, 8, 10$ and 12, respectively. It can be found that the simulating results obtained by both CFDS and CFDS&POD are in excellent agreement with the exact solutions after a relatively long period of time of evolution. Meanwhile, in Tables IV and V the global relative errors of CFDS and CFDS&POD are recorded at time $t = 6, 8, 10$ and 12 with different nodes and time steps. It can be seen that the CFDS&POD is less accurate than the CFDS in this example. Table VI shows the computational time of CFDS and CFDS&POD with 30 POD bases at $t = 12$ and different nodes. It can be clearly found that the computational time of CFDS&POD is much less than that of CFDS under the same number of nodes. Moreover, it also can be seen that the computational efficiency of CFDS&POD increases as the number of node increases.

Example 3 In this example, we consider the GKS equation with $\alpha = \gamma = 1$ and $\beta = 4$. The exact solution of the problem is given by [6], [7]

$$u^*(x,t) = b + 9 - 15[\tanh(\kappa(x - bt - x_0)) + \tanh^2(\kappa(x - bt - x_0)) + \tanh^3(\kappa(x - bt - x_0))] \quad (41)$$

where $b = 6, \kappa = 0.5$ and $x_0 = -10$. Similar to the previous examples, the initial condition and boundary condition are taken from the exact solution.

For comparison purposes, we also plot the numerical solutions and point-wise absolute errors of CFDS and CFDS&POD with 40 POD bases at time $t = 1, 2, 3$ and 4 in Figs. 5 and 6, respectively. It can be seen that the CFDS&POD scheme is slightly less accurate than that of CFDS scheme. Meanwhile, the global relative error for the solutions of GKS equation at different times and nodes can be found in Tables VII and VIII. It can be seen that the order of accuracy of the CFDS&POD is the same as that of the CFDS under the same number of nodes and time step. Table IX reports the computational time of CFDS and CFDS&POD with 40 POD bases at $t = 4$ with different number of nodes and time steps. It can be obviously seen that compared with CFDS, the CFDS&POD greatly saved the time-consuming and vastly improved the computational efficiency as the number of nodes increases.

Example 4 In this example, we consider the following equation with $\alpha = 1, \gamma = 0.5$ and $\beta = 0$. The exact solution of the problem is taken from [3]

$$u^*(x,t) = -\tfrac{1}{\kappa} + \tfrac{60}{19}\kappa(-38\gamma\kappa^2 + \alpha)tanh\theta + 120\gamma\kappa^3 tanh^3\theta \quad (42)$$

where $\theta = \kappa x + t$ and $\kappa = (1/2)\sqrt{11\alpha/19\gamma}$. Again,

TABLE IV
THE GLOBAL RELATIVE ERRORS OF CFDS AT DIFFERENT NODES AND TIMES FOR EXAMPLE 2

node numbers	time step	CFDS			
		$t=6$	$t=8$	$t=10$	$t=12$
101	1E-2	7.79E-8	8.28E-8	8.55E-8	1.04E-7
201	1E-3	4.89E-8	4.85E-8	6.27E-8	1.37E-8
401	1E-4	3.34E-8	3.30E-8	4.33E-8	9.88E-8

TABLE V
THE GLOBAL RELATIVE ERRORS OF CFDS&POD AT DIFFERENT NODES AND TIMES FOR EXAMPLE 2

node numbers	time step	CFDS&POD			
		$t=6$	$t=8$	$t=10$	$t=12$
101	1E-2	4.59E-7	5.52E-7	6.38E-7	7.12E-7
201	1E-3	4.40E-7	5.48E-7	6.52E-7	7.58E-7
401	1E-4	4.56E-7	5.76E-7	6.92E-7	8.07E-7

TABLE VI
COMPARE THE COMPUTATIONAL TIME AT $t=12$ WITH DIFFERENT NUMBER OF NODES FOR EXAMPLE 2

node numbers	time step	computational time(second)	
		CFDS	CFDS&POD
101	1E-2	1.700317	0.465633
201	1E-3	43.676295	4.160707
401	1E-4	2638.773898	76.513533

TABLE VII
THE GLOBAL RELATIVE ERRORS OF CFDS AT DIFFERENT NODES AND TIMES FOR EXAMPLE 3

node numbers	time step	CFDS			
		$t=1$	$t=2$	$t=3$	$t=4$
61	1E-2	2.63E-2	3.39E-2	3.71E-2	3.64E-2
121	1E-3	1.60E-3	1.70E-3	2.60E-3	5.10E-3
241	1E-4	1.04E-4	1.07E-4	1.67E-4	3.22E-4

TABLE VIII
THE GLOBAL RELATIVE ERRORS OF CFDS&POD AT DIFFERENT NODES AND TIMES FOR EXAMPLE 3

node numbers	time step	CFDS&POD			
		$t=1$	$t=2$	$t=3$	$t=4$
61	1E-2	2.72E-2	1.30E-2	1.26E-2	1.50E-2
121	1E-3	1.30E-3	1.20E-3	2.10E-3	4.50E-3
241	1E-4	2.49E-4	2.30E-4	3.44E-4	5.72E-4

TABLE IX
COMPARE THE COMPUTATIONAL TIME AT $t=4$ WITH DIFFERENT NUMBER OF NODES FOR EXAMPLE 3

node numbers	time step	computational time(second)	
		CFDS	CFDS&POD
61	1E-2	0.364877	0.21566
121	1E-3	6.741534	2.047361
241	1E-4	227.451492	23.850858

the boundary and initial conditions are given by the exact solution on the interval $[-30, 20]$.

Similar to the examples above, the numerical solutions and point-wise absolute errors of CFDS and CFDS&POD with 40 POD bases at time $t = 1, 2, 3$ and 4 are depicted in Figs. 7 and 8 , respectively. It can be seen that the CFDS&POD scheme is slightly less accurate than that of CFDS scheme. Meanwhile, Tables X and XI show the global relative errors of CFDS and CFDS&POD at different times and nodes, respectively. Table XII presents the computational time of CFDS and CFDS&POD with 40 POD bases at $t = 4$ with different number of nodes and time steps. From Tables X to XII, it can be also seen that the CFDS&POD greatly saves compuational time and gets the same compuational accuracy compared with the CFDS.

IV. CONCLUSION

In this paper, the CFDS&POD method is presented and applied to solve GKS equation. In this algorithm, the numerical simulation results or experiment data are firstly collected

TABLE X
THE GLOBAL RELATIVE ERRORS OF CFDS AT DIFFERENT NODES AND TIMES FOR EXAMPLE 4

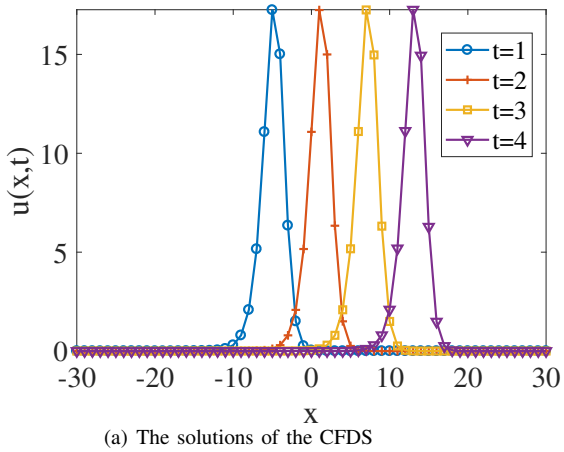
node numbers	time step	CFDS			
		$t=1$	$t=2$	$t=3$	$t=4$
76	1E-2	1.20E-3	2.20E-3	3.60E-3	5.40E-3
151	1E-3	7.81E-5	1.41E-4	2.27E-4	3.40E-4
301	1E-4	4.919E-6	8.89E-6	1.43E-5	2.13E-5

TABLE XI
THE GLOBAL RELATIVE ERRORS OF CFDS&POD AT DIFFERENT NODES AND TIMES FOR EXAMPLE 4

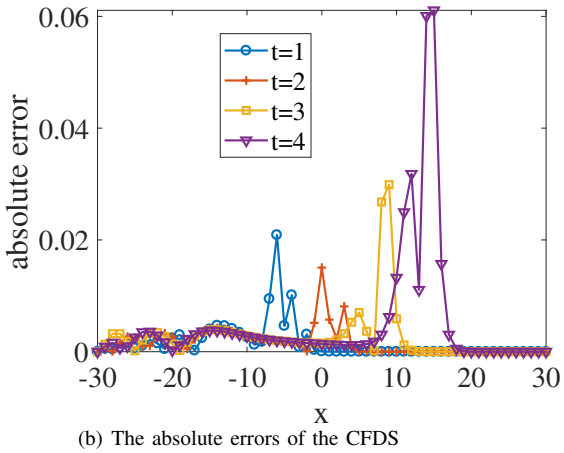
node numbers	time step	CFDS&POD			
		$t=1$	$t=2$	$t=3$	$t=4$
76	1E-2	1.20E-3	2.20E-3	3.90E-3	5.60E-3
151	1E-3	7.86E-5	1.58E-4	2.39E-4	3.68E-4
301	1E-4	4.64E-6	1.07E-5	3.53E-5	3.04E-4

TABLE XII
COMPARE THE COMPUTATIONAL TIME AT $t=4$ WITH DIFFERENT NUMBER OF NODES FOR EXAMPLE 4

node numbers	time step	computational time(second)	
		CFDS	CFDS&POD
76	1E-2	0.498469	0.278459
151	1E-3	10.66866	2.278924
301	1E-4	367.119599	26.274226

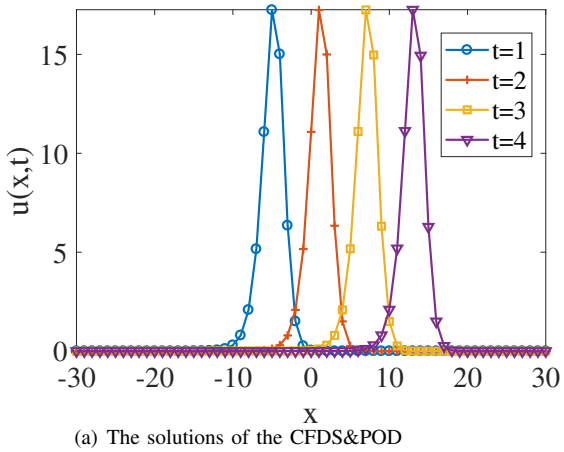


(a) The solutions of the CFDS

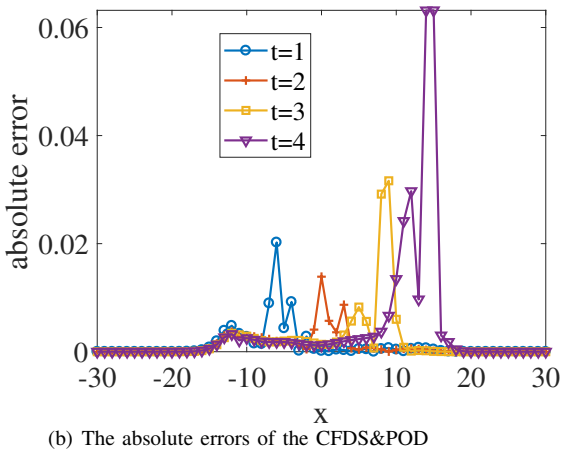


(b) The absolute errors of the CFDS

Fig. 5. The solutions and absolute errors of CFDS at $t=1,2,3$ and 4 with 121 nodes for Example 3.

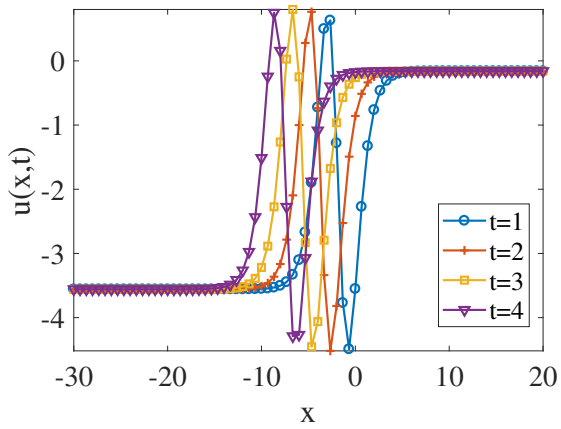


(a) The solutions of the CFDS&POD

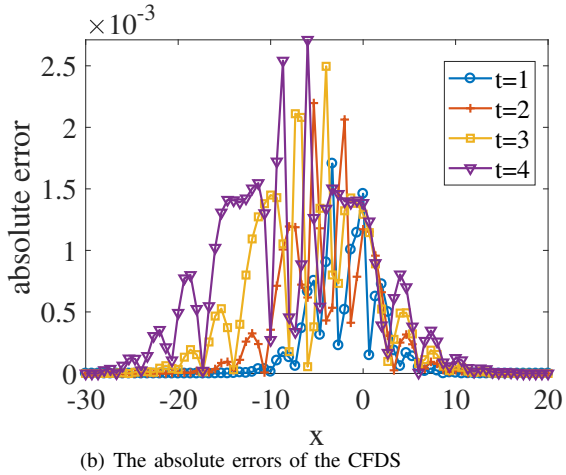


(b) The absolute errors of the CFDS&POD

Fig. 6. The solutions and absolute errors of CFDS&POD at $t=1,2,3$ and 4 with 121 nodes for Example 3.

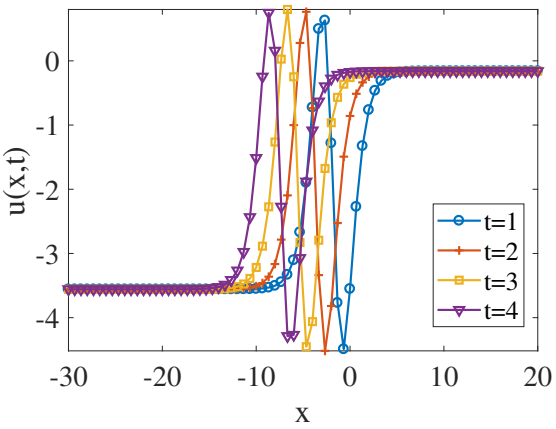


(a) The solutions of the CFDS

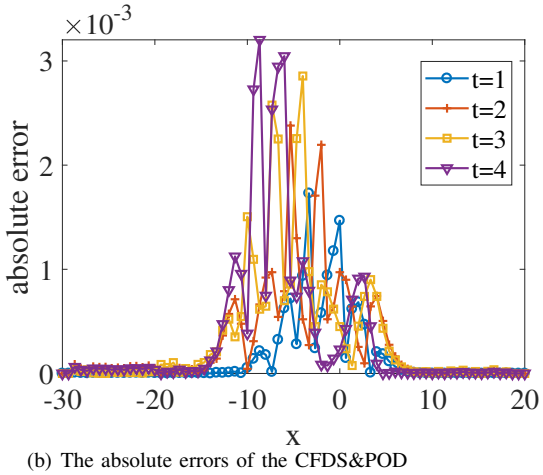


(b) The absolute errors of the CFDS

Fig. 7. The solutions and absolute errors of CFDS at $t = 1, 2, 3$ and 4 with 151 nodes for Example 4.



(a) The solutions of the CFDS&POD



(b) The absolute errors of the CFDS&POD

Fig. 8. The solutions and absolute errors of CFDS&POD at $t = 1, 2, 3$ and 4 with 151 nodes for Example 4.

as snapshots, then the optimal POD basis is obtained by SVD, finally POD in conjunction with the implicit high-order CFDS is applied to generate the reduced model.

To the best of our knowledge, this is the first time for CFDS&POD method to be used for solving GKS equation. The efficient and accuracy of the proposed algorithm were examined by four test examples, the findings can be summarized as follows:

(1) the numerical results obtained by CFDS&POD are found to be in very good agreement with the exact solutions and the corresponding CFDS solutions, but compared with the CFDS, the accuracy of CFDS&POD is more or less reduced in some examples.

(2) as far as the computational time is concerned, it can be found that compared with the corresponding CFDS method, the CFDS&POD method can bring significant computational time saving for solving GKS equation, especially for larger number of nodes and smaller time step cases.

(3) as the same as CFDS, the approximate process of CFDS&POD needs not any transformation or linearization, thus it is easy to implement to a nonlinear equation and easy to program.

ACKNOWLEDGMENTS

The authors express their sincere thanks to the anonymous reviews for their valuable suggestions and corrections for improving the quality of this paper.

REFERENCES

[1] R. C. Mittal, G. Arora, "Quintic B-spline collocation method for numerical solution of the KuramotoSivashinsky equation," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 10, pp.2798-2808, 2010.

[2] U. Marjan, S. Haq, "A mesh-free numerical method for solution of the family of KuramotoSivashinsky equations," *Applied Mathematics and Computation*, vol. 212, no. 2, pp. 458-469, 2009.

[3] L. Mehrdad, M. Dehghan, "Numerical solutions of the generalized KuramotoSivashinsky equation using B-spline functions," *Applied Mathematical Modelling*, vol. 36, no. 2, pp. 605-617, 2012.

[4] G. D. Akrivis, "Finite difference discretization of the Kuramoto-Sivashinsky equation," *Numerische Mathematik*, vol. 63, no. 1, pp. 1-11, 1992.

[5] A. H. Khater, R. S. Temsah, "Numerical solutions of the generalized KuramotoSivashinsky equation by Chebyshev spectral collocation methods," *Computers & Mathematics with Applications*, vol. 56, no. 6, pp. 1465-1472, 2008.

[6] E. Dabboura, H. Sadat, C. Prax, "A moving least squares meshless method for solving the generalized Kuramoto-Sivashinsky equation," *Alexandria Engineering Journal*, vol. 55, no. 3, pp. 2783-2787, 2016.

[7] H. L. Lai , C. F. Ma, "Lattice Boltzmann method for the generalized KuramotoSivashinsky equation," *Physica A: Statistical Mechanics and its Applications*, vol. 388, no. 8 pp. 1405-1412, 2009.

[8] H. Otomo, B. M. Boghosian, F. Dubois, "Efficient lattice Boltzmann models for the Kuranmoto-Sivashinsky equation," *Computers & Fluids*, 2018. https://doi.org/10.1016/j.compfluid.2018.01.036

[9] B. K. Singh, G. Arora, P. Kumar, "A note on solving the fourth-order Kuramoto-Sivashinsky equation by the compact finite difference scheme," *Ain Shams Engineering Journal*, 2016. http://dx.dox.org/10.1016/j.asej.2016.11.008.

[10] J. Li, Y. Chen, *Computational partial differential equations using MATLAB*, New York: Crc Press, 2009.

[11] M. Sari, G. Gürarslan, "A sixth-order compact finite difference scheme to the numerical solutions of Burgers' equation," *Applied Mathematics and Computation*, vol. 208, no. 2, pp. 475-483, 2009.
[12] D. V. Gaitonde, M. R. Visbal, "High-order schemes for Navier-Stokes equations: algorithm and implementation into FDL3DI. No. AFRL-VA-WP-TR-1998-3060," *Air Force Research Lab Wright-Patterson AFB OH Air Vehicles Directorate 1998*, pp. 1-42.
[13] S. K. Lele, "Compact finite difference schemes with spectral-like resolution," *Journal of computational physics*, vol. 103, pp. 16-42, 1992.
[14] M. Dehghan, A. Taleei, "A compact split-step finite difference method for solving the nonlinear Schrödinger equations with constant and variable coefficients," *Computer Physics Communications*, vol. 181, pp. 43-51, 2010.
[15] M. Sari, G. Gürarslan, "A sixth-order compact finite difference method for the one-dimensional sine-Gordon equation," *International Journal for Numerical Methods in Biomedical Engineering*, vol. 27, pp. 1126-1138, 2011.
[16] S. Li, "Numerical analysis for fourth-order compact conservative difference scheme to solve the 3D Rosenau-RLW equation," *Computers and Mathematics with Applications*, vol. 72, pp. 2388-2407, 2016.
[17] B. Wongsaijai, K. Poochinapan, T. Disyadej, "A compact finite difference method for solving the general Rosenau-RLW equation, " *IAENG International Journal of Applied Mathematics*, vol. 44, no. 4, pp. 192-199, 2014.
[18] S. Zhou, F. Meng, Q. Feng, L. Dong, "A spatial sixth order finite difference scheme for time fractional sub-diffusion equation with variable coefficient, " *IAENG International Journal of Applied Mathematics*, vol. 47, no. 2, pp. 175-181, 2017.
[19] P. Zhang, X. Zhang, L. Song, "A fast and stabilized meshless method for the convection-dominated convection-diffusion problems," *Numerical Heat Transfer, Part A: Applications*, vol. 70, pp. 420-431, 2016.
[20] Y. C. Liang, H. P. Lee, S. P. Lim, W. Z. Lin, K. H. Lee, C. G. Wu, "Proper orthogonal decomposition and its applications-Part I: Theory," *Journal of Sound and vibration*, vol. 252, pp. 527-544, 2002.
[21] M. Rathinam, L. R. Petzold, "A new look at proper orthogonal decomposition," *SIAM Journal on Numerical Analysis*, vol. 41, pp. 1893-1925, 2003.
[22] G. Kerschen, J. C. Golinval, A. F. Vakakis, L. A. Bergman, "The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview," *Nonlinear dynamics*, vol. 41, no. 1-3, pp. 147-169, 2005.
[23] P. Sun, Z. Luo, Y. Zhou, "Some reduced finite difference schemes based on a proper orthogonal decomposition technique for parabolic equations," *Applied Numerical Mathematics*, vol. 60, no. 1-2, pp. 154-164, 2010.
[24] Z. Luo, H. Li, P. Sun, "A reduced-order CrankNicolson finite volume element formulation based on POD method for parabolic equations," *Applied Mathematics and Computation*, vol. 219, no. 11, pp. 5887-5900, 2013.
[25] J. An, Z. Luo, H. Li, P. Sun, "Reduced-order extrapolation spectral-finite difference scheme based on POD method and error estimation for three-dimensional parabolic equation," *Frontiers of Mathematics in China*, vol. 10, no. 5, pp. 1025-1040, 2015.
[26] Z. Luo, X. Yang, Y. Zhou, "A reduced finite difference scheme based on singular value decomposition and proper orthogonal decomposition for Burgers equation," *Journal of Computational and Applied Mathematics*, vol. 229, no. 1, pp. 97-107, 2009.
[27] Z. Luo, H. Li, P. Sun, J. Gao, "A reduced-order finite difference extrapolation algorithm based on POD technique for the non-stationary NavierStokes equations," *Applied Mathematical Modelling*, vol. 37, no. 7, pp. 5464-5473, 2013.
[28] Z. Luo, J. Chen, I. M. Navon, X. Yang, "Mixed finite element formulation and error estimates based on proper orthogonal decomposition for the nonstationary NavierStokes equations," *SIAM Journal on Numerical Analysis*, vol. 47, no. 1, pp. 1-19, 2009.
[29] Z. Luo, S. Jin, J. Chen, "A reduced-order extrapolation central difference scheme based on POD for two-dimensional fourth-order hyperbolic equations," *Applied Mathematics and Computation*, vol. 289, pp. 396-408, 2016.
[30] R. A. Bialecki, A. J. Kassab, A. Fic, "Proper orthogonal decomposition and modal analysis for acceleration of transient FEM thermal analysis," *International journal for numerical methods in engineering*, vol. 62, no. 6, pp. 774-797, 2005.
[31] X. Li, B. X. Hu, "Proper orthogonal decomposition reduced model for mass transport in heterogenous media," *Stochastic Environmental Research and Risk Assessment*, vol. 27, no. 5, pp. 1181-1191, 2013.
[32] X. Zhang, H. Xiang, "A fast meshless method based on proper orthogonal decomposition for the transient heat conduction problems," *International Journal of Heat and Mass Transfer*, vol. 84, pp. 729-739, 2015.
[33] M. Dehghan, M. Abbaszadeh, "A combination of proper orthogonal decompositiondiscrete empirical interpolation method (PODDEIM) and meshless local RBF-DQ approach for prevention of groundwater contamination," *Computers & Mathematics with Applications*, vol. 75, no. 4, pp. 1390-1412, 2018.
[34] M. Dehghan, M. Abbaszadeh, "The use of proper orthogonal decomposition (POD) meshless RBF-FD technique to simulate the shallow water equations," *Journal of Computational Physics*, vol. 351, pp. 478-510, 2017.