J. Korean Soc. Ind. Appl. Math. Vol.23, No.4, 301–327, 2019 http://doi.org/10.12941/jksiam.2019.23.301

# PROPER ORTHOGONAL DECOMPOSITION OF DISCONTINUOUS SOLUTIONS WITH THE GEGENBAUER POST-PROCESSING

BYEONG-CHUN SHIN[1] AND JAE-HUN JUNG[2]

[1]DEPARTMENT OF MATHEMATICS, CHONNAM NATIONAL UNIVERSITY, GWANGJU 61186, KOREA
*Email address*: bcshin@jnu.ac.kr

[2]DEPARTMENT OF AI & DATA SCIENCE, AJOU UNIVERSITY, SUWON 16499, KOREA, & DEPARTMENT OF MATHEMATICS, UNIVERSITY AT BUFFALO SUNY, BUFFALO, NY 14260, USA
*Email address*: jaehunjung@ajou.ac.kr, jaehun@buffalo.edu

ABSTRACT. The proper orthogonal decomposition (POD) method for time-dependent problems significantly reduces the computational time as it reduces the original problem to the lower dimensional space. Even a higher degree of reduction can be reached if the solution is smooth in space and time. However, if the solution is discontinuous and the discontinuity is parameterized e.g. with time, the POD approximations are not accurate in the reduced space due to the lack of ability to represent the discontinuous solution as a finite linear combination of smooth bases. In this paper, we propose to post-process the sample solutions and re-initialize the POD approximations to deal with discontinuous solutions and provide accurate approximations while the computational time is reduced. For the post-processing, we use the Gegenbauer reconstruction method. Then we regularize the Gegenbauer reconstruction for the construction of POD bases. With the constructed POD bases, we solve the given PDE in the reduced space. For the POD approximation, we re-initialize the POD solution so that the post-processed sample solution is used as the initial condition at each sampling time. As a proof-of-concept, we solve both one-dimensional linear and nonlinear hyperbolic problems. The numerical results show that the proposed method is efficient and accurate.

## 1. INTRODUCTION

Despite recent advances in computing power it is still a challenging task to approximate real world problems in real time. Real world problems are stochastic in nature and multiple parameters with fine spatial and temporal steps increase the computational complexity exponentially. Reduced-order models (ROMs) help to reduce the computational complexity and help to calculate approximations within a reasonable timeframe. The reader is referred to [1], a special issue on ROMs for recent progresses in ROMs. The proper orthogonal decomposition (POD) method is one of the ROMs. The POD method has been intensively developed in recent

Received by the editors October 30 2019; Accepted December 3 2019; Published online December 25 2019.
2000 *Mathematics Subject Classification.* 65M70.
*Key words and phrases.* Proper orthogonal decomposition, Reduced-order models, Hyperbolic conservation laws, Gegenbauer post-procesing, Regularization.

decades and used in various applications. The reader is also referred to [2, 3, 4, 5, 6, 7, 8, 9, 10] and references therein for the details of POD methods and related subjects.

The POD method for time-dependent partial differential equations (PDEs) is composed of three major steps to calculate POD approximations. The first step is to construct the sample space whose elements are individual solutions so-called the snapshots to the given PDE, collected at certain time intervals. Each solution in the sample space is represented as a vector constituting a sample matrix. Here note that the snapshots can be obtained using every solution obtained when solving the PDE (fine-time scale) or selectively among all solutions at certain sampling time steps (coarse-time scale). The second step is to find the ordered orthonormal basis set known as the POD basis set that can represent the collected sample solutions as a linear combination of POD bases. The last step is to approximate the given PDE in the reduced space. The element in the reduced space, also a vector, is mapped to the solution defined in the original space. If the solutions to the given PDE are smooth in space and time, the sample space obtained from the first step becomes a smooth manifold and it can be approximated with a small number of the POD bases obtained in the second step. The smoother the sample space is the smaller the number of the POD bases is needed to approximate the samples to a certain degree of accuracy. With the singular value decomposition (SVD) of the sample matrix, the reduction is done by truncating the singular values if they are smaller than the given tolerance level. The error of the approximation of the samples with the POD bases is then computed by the total measure of the truncated singular values. If the sample space is smooth the number of the non-vanishing singular values becomes small by the truncation. Accordingly if the sample space is smooth the error becomes small by the truncation and the number of the POD bases required for the approximation is also reduced. Then the reduction can decrease significantly the computational complexity that the original problem has.

If the sample solutions are discontinuous and the discontinuity is parameterized e.g. with time, however, the same degree of reduction as in the smooth case may not be reachable. This is due to the lack of ability for the POD bases to represent the discontinuous samples. Furthermore, if the solution is discontinuous any high order approximation of the solution suffers from the Gibbs phenomenon. These oscillations, known as the Gibbs oscillations, contained in the samples make the POD bases oscillatory. Several methods have been developed to remedy this problem. These remedies mostly modify the PDE solver. For example, in [11] a new type of Petrov-Galerkin method was developed to deal with the discontinuous solutions to nonlinear hyperbolic equations for POD approximations. The idea of the developed Petrov-Galerkin method is to impose the diffusion term in the ROM naturally so that the Gibbs oscillations are diminished in the reduced space.

In this paper, we present an efficient and simpler way of dealing with the discontinuous solutions for the POD approximation instead of modifying the PDE solver. The main idea of the proposed method is to improve the quality of the collected sample solutions with post-processing and re-initialize the POD approximations when the PDE is solved in the reduced space. For the post-processing, we adopt the Gegenbauer reconstruction method. The reconstruction is given as a piecewise Gegenbauer polynomial. The obtained piecewise polynomial reconstruction is uniformly convergent up to the singularity [12, 13]. For more details the reader is referred to a series of papers by Gottlieb and his co-workers [14, 15, 16, 12, 13]. Our proposed method is to first reconstruct the oscillatory sample solutions with the Gegenbauer reconstruction method so that the reconstruction is close to the exact solution and well preserves the discontinuity. Then the reconstructed solution is regularized near the discontinuity. Based on the post-processed samples, the POD bases are constructed. The given PDE is then solved in the reduced space with the POD bases. Although the post-processed samples are now smooth, the solutions in the reduced space are soon to be discontinuous according to the PDE and the reduction does not guarantee the alleviation of the Gibbs oscillations. To take advantage of the post-processed samples that are free of oscillations we use the re-initialization when the PDE is solved in the reduced space in a way that the post-processed samples are used as the initial condition at each sampling time. The re-initialization is necessary. Particularly the POD solutions become inaccurate with time without the re-initialization. Since the re-initialization uses the post-processed sample solutions, no extra computational work is needed for the POD approximations. This is explained with examples in Section 5.3.1.

As a proof-of-concept, we solve the linear and nonlinear hyperbolic problems which have the discontinuous initial condition or develop discontinuities with time. For the numerical approximation we use the Chebyshev spectral collocation method with the spectral filtering. Numerical results presented in this paper show that the proposed method yields reasonably accurate POD approximations while the computational time is reduced significantly.

The paper is composed of the following sections. In Section 2, we briefly explain the POD method. In Section 3, we explain the POD collocation method for hyperbolic problems with the Chebyshev collocation method. In Section 4, the proposed method is explained. The proposed method is composed of four steps; 1) Gegenbauer reconstruction, 2) Regularization, 3) Reprojection and 4) Re-initialization. In Section 5, three numerical examples are provided. In Section 6, a brief conclusion and our future research are provided.

## 2. PROPER ORTHOGONAL DECOMPOSITION

The POD method is composed of the following three steps [8]. The first step is to collect the sample solutions based on which the POD bases are constructed. In this work, we assume that the sample solutions are real and they are collected sparsely including the initial and final solutions. Let $u_i$ be the sample vector valued solutions collected at the time of $t = t_i$. The vectors, $u_i$, are also called *snapshots*. We make $u_i$ be column vectors whose size is $N + 1$. Suppose that we have $m$ number of snapshots. Let $X \in \mathbb{R}^{(N+1)\times m}$ be the matrix whose columns are $u_i$, i.e. $X = [u_1\, u_2\, \cdots\, u_m]$. The second step is to construct a set of orthonormal bases $v_k$ with which each sample solution $u_i$ is approximated by a finite linear combination of $v_k$, $u_i \approx \tilde{u}_i = \sum_{k=1}^{p} d_{ik} v_k$ in the sense that the error $E(p) = \sum_{i=1}^{m} ||u_i - \tilde{u}_i||^2$ is minimized. For the orthonormal bases, the SVD of $X$ is carried out such that $X = V\Sigma U^T$ where $V \in \mathbb{R}^{(N+1)\times(N+1)}$ and $U \in \mathbb{R}^{m\times m}$ are orthonormal matrices and $\Sigma \in \mathbb{R}^{(N+1)\times m}$ is the diagonal matrix whose elements are the singular values of $X$, $\sigma_1 \geq \sigma_2 \geq \cdots \sigma_r > \sigma_{r+1} = \cdots = \sigma_m = 0$ where $r = \mathrm{rank}(X) \leq \min\{N + 1, m\}$. If we let $d_i$ be the column vectors of $\Sigma U^T$, then we have $u_i = V d_i, i = 1, \cdots, m$. Thus we know that we can choose $v_i$ as the orthonormal bases that we looked for and the total number of expansion for $u_i$ is $r$ and we call $v_i$ the POD basis. If we choose those $v_i$ for $p \leq m$, then the set of $v_i$ $(1 \leq i \leq p)$ is called the POD basis of order $p$. Denote by $V_p$ the matrix composed of the first $p$ column vectors of $V$. Then the error $E(p)$ is given by $E(p) = \sum_{i=p+1}^{r} \sigma_i^2$. Thus we know that with the choice of $v_i$ as the POD basis, the smaller $\sigma_{p+1}$ is, the smaller the error $E(p)$ is. If $X$ is composed of vector values for smooth solutions in space, then the singular values of $\sigma_i$ decay fast with $i$ and the sample is approximated with a small number of the POD bases with a certain degree of accuracy. In practice we decide the value of $p$ such that $\sigma_{p+1} \leq \epsilon_p$ where $\epsilon_p$ is a small tolerance which is determined by users before the POD approximation. The last step is to reduce the original space with the POD bases. Let $v$ be a solution sought. The reduction is done by the transformation of $v$ to $w = V_p^T v$. Then the dimension of the original space of $v$ is reduced to $p$.

## 3. POD COLLOCATION FOR HYPERBOLIC CONSERVATION LAWS

We consider the following one-dimensional hyperbolic conservation laws

$$u_t + \nabla \cdot F(u) = 0, \tag{3.1}$$

with the state vector $u \equiv u(x,t) : \Omega \times I \to \mathbb{R}^d$ for the time interval $I := (0, T]$ for some $T > 0$ and an open bounded domain $\Omega \subset \mathbb{R}$. $F(u) := (f_1(u), \cdots, f_d(u))$ is the flux function. Suppose that the initial condition $u(x,0) = u_0(x)$ is given. The initial condition can be either smooth or discontinuous.

We adopt the spectral collocation method for the numerical solution to Eq. (3.1). The proposed method in this paper, however, is applicable to other numerical methods dealing with discontinuous problems. For example, one can use the spectral collocation and radial basis function (RBF) methods for the interface problem [17] in order to approximate the POD solutions of discontinuous problems.

To explain the POD method with the spectral collocation method, consider the scalar hyperbolic problem of Eq. (3.1) in $x \in [-1, 1]$ with the condition of $\partial F/\partial u > 0$ without loss of generality. In general, the sign is checked at each boundary and the boundary condition is applied according to the sign of $\partial F/\partial u$. And suppose that the boundary condition is given by

$$u(x,t) = h(t), \quad x = -1.$$

We seek a polynomial solution $u_N(x,t)$ to $u(x,t)$ in a polynomial space of degree at most $N$ such that the residual vanishes on the Chebyshev-Gauss-Lobatto (CGL) points $x_j = -\cos(\pi j/N)$, $j = 0, 1 \cdots, N$, i.e.,

$$\frac{\partial u_N(x,t)}{\partial t} + \left(\frac{\partial}{\partial x} I_N F(u_N(x,t))\right) = 0$$

where $I_N$ is the interpolation operator associated with the CGL points. Note that since we are using the Chebyshev spectral method, $I_N \frac{\partial}{\partial x} I_N = \frac{\partial}{\partial x} I_N$. The solution $u_N(x,t)$ can be represented by the Chebyshev polynomial expansion

$$u_N(x,t) = \sum_{l=0}^{N} \hat{u}_l(t) T_l(x),$$

where $T_l(x)$ are the Chebyshev polynomials and $\hat{u}_l$ are the expansion coefficients. Using the boundary condition of $u_N(x_0,t) = h(t)$ let

$$u_N(x,t) = v_N(x,t) + h(t)\,\phi_0(x) \quad \text{with} \quad v_N(x,t) = \sum_{k=1}^{N} v_k(t)\,\phi_k(x)$$

where $\phi_k(x)$ are the Lagrange polynomials of degree $N$ with respect to the CGL points. Note that $v_N(x,t)$ satisfies the homogeneous boundary condition, i.e., $v_N(x_0,t) = 0$. Let $\hat{f}_N(\hat{v}_N(t)) = \big(F(h(t)),\, F(v_1(t)), \cdots, F(v_N(t))\big)^T$ and $\hat{v}_N(t) = \big(v_1(t), \cdots, v_N(t)\big)^T$. Let $D_N \in \mathbb{R}^{N\times(N+1)}$ be the submatrix of the first-order Chebyshev differential matrix $D \in \mathbb{R}^{(N+1)\times(N+1)}$ formed by removing the first row of $D$. Then the Chebyshev collocation solution $\hat{v}_N(t)$ is obtained by the following system of differential equations

$$\frac{d\hat{v}_N(t)}{dt} \;=\; -D_N\,\hat{f}_N(\hat{v}_N(t)) \quad \text{with} \quad \hat{v}_N(0) = \big(u_0(x_1), \cdots, u_0(x_N)\big)^T. \qquad (3.2)$$

Let $\ell$, $m$ and $N_s$ be positive integers satisfying $N_s = m\ell$ and let $s_i = i\,\Delta s$ $(i = 0, 1, \cdots, N_s)$ be an uniform time sequence with a sample time spacing $\Delta s = T/N_s$. We first compute the approximate solutions $\hat{v}_N(s_i)$ $(i = 0, 1, \cdots, N_s)$ for Eq. (3.2) and then the sample solutions $\hat{v}_N(s_{j\ell})$ $(j = 0, 1, 2, \cdots, m)$ are collected for the POD basis. Here $s_{j\ell}$ denotes the ordered time sequence in $j$ $(j = 0, 1, 2, \cdots, m)$ selected from $\{s_i\}_{i=0}^{N_s}$ with the given value of $\ell$. $s_{j\ell}$ are selected from the set $\{s_i\}$ either uniformly or not. If $\ell = 1$, $s_{j\ell} = s_j$ $(j = 0, 1, 2, \cdots, m)$. If $\ell > 1$, the sampling time spacing $\Delta s$ with which the numerical solution is computed is different from the sampling interval collected for the POD basis. For this case, the sampling interval is $\ell$-times larger than the sampling time spacing.

Let $X$ be the collection of the sample solutions such that $X \in \mathbb{R}^{N\times(m+1)}$ becomes a sample matrix whose $j$th column is $\hat{v}_N(s_{j\ell})$, i.e.,

$$X = [\hat{v}_N(s_{0\ell})\;\hat{v}_N(s_{1\ell})\;\cdots\;\hat{v}_N(s_{m\ell})].$$

Applying the SVD to $X$ leads to get $X = V\Sigma U^T$ where $V \in \mathbb{R}^{N\times N}$ and $U \in \mathbb{R}^{(m+1)\times(m+1)}$ are orthonormal matrices and $\Sigma \in \mathbb{R}^{N\times(m+1)}$ is the diagonal matrix whose elements are the singular values of $X$, $\sigma_1 \geq \sigma_2 \geq \cdots \sigma_r > \sigma_{r+1} = \cdots = \sigma_{m+1} = 0$ where $r = \operatorname{rank}(X) \leq \min\{N, m+1\}$. We try to find the orthonormal bases that can represent the collected solution well. Let $V_p \in \mathbb{R}^{N\times p}$ be the POD-basis matrix composed of the first $p$ columns of $V$. The value of $p \leq (m+1)$ is obtained by the truncation with a certain tolerance $\epsilon_p$ as below

$$\sigma_p > \epsilon_p \geq \sigma_{p+1}, \quad \epsilon_p > 0.$$

If the sample solutions are smooth in space, then the singular values decay rapidly and the value of $p$ becomes small with the given tolerance $\epsilon_p$. Furthermore the sample solutions can be represented by a smaller number $p$ of bases rather than the full number of bases. The tolerance $\epsilon_p$ is usually taken small, e.g. $\epsilon \sim 10^{-6}$ depending on the problem considered. If the sample solutions are discontinuous, the singular values do not decay or decay slowly in general depending on the behavior of the discontinuity except for some special cases. The special cases include the case that the discontinuity is fixed and does not change with time for which only the first few singular values are non-vanishing and the POD can be done efficiently. If the sample solutions are discontinuous, the value of $p$ may approach to $r$ with the given value of $\epsilon_p$ in general. Examples are illustrated in Section 5.1.

We reduce the number of solution basis by the transform of $\hat{v}_N$ with $V_p$ such that

$$V_p \hat{w}_p(t) = \hat{v}_N(t). \tag{3.3}$$

Then the reduced model is described by $\hat{w}_p(t)$. From Eq. (3.2), we have the following equation for $\hat{w}_p(t)$

$$\frac{d\hat{w}_p(t)}{dt} \;=\; -V_p^T D_N \hat{f}_N\big(V_p \hat{w}_N(t)\big) \tag{3.4}$$

with the initial condition $\hat{w}_p(0) = V_p^T \, v_N(0)$. If the flux function $F$ is linear, e.g. $F(u_N(x,t)) = u_N(x,t)$, then the right hand side (RHS) of Eq. (3.4) becomes simple as

$$\text{RHS} = -V_p^T D_N V_p \, \hat{w}_p - h(t) \, V_p^T d_0$$

where $-V_p^T D_N V_p \in \mathbb{R}^{p \times p}$ and $V_p^T d_0 \in \mathbb{R}^p$ with the first column $d_0$ of the Chebyshev differential matrix $D$. In this case, the matrix $D_N$ is the submatrix formed by removing the first row and column of D. Also the reduction order of this case is optimal. If the flux function is nonlinear, the optimal reduction order is not obtainable and the complexity will increase in the computation of the flux function directly. For this case, we directly compute the flux function first before applying the operator $V_p^T D_N \in \mathbb{R}^{p \times N}$.

Let $t_n = n\,\Delta t$ $(n = 0, 1, \cdots, M)$ be an uniform time sequence with a time spacing $\Delta t = T/M$. To solve the reduced equation Eq. (3.4) in time, we use the TVD 3rd-order Runge-Kutta method [18] that for $n = 0, 1, 2, \cdots, M-1$, $\hat{w}_p^{n+1} = \hat{w}_p((n+1)\Delta t)$ is updated as follows

$$\begin{aligned}
\bar{w}_0 &= \hat{w}_p^n + \Delta t\big(-V_p^T D_N \hat{f}_N(V_p \hat{w}_p^n)\big),\\
\bar{w}_1 &= \frac{3}{4}\hat{w}_p^n + \frac{1}{4}\big(\bar{w}_0 + \Delta t\big(-V_p^T D_N \hat{f}_N(V_p \bar{w}_0)\big)\big),\\
\hat{w}_p^{n+1} &= \frac{1}{3}\hat{w}_p^n + \frac{2}{3}\big(\bar{w}_1 + \Delta t\big(-V_p^T D_N \hat{f}_N(V_p \bar{w}_1)\big)\big).
\end{aligned} \tag{3.5}$$

From now on we write the vector valued function without the ˆ symbol for the simplicity of notation, e.g. $v_N(t)$ instead of $\hat{v}_N(t)$ for the solution vector in Eq. (3.3).

If the solution to Eq. (3.2) is discontinuous, the spectral filtering method is applied to the solution $v_N(t)$. The filtering is also applied in the reduced space to $w_N(t)$. For the given solution $v_N(t)$ the filtered solution is given by the transformation

$$v_N(t) = S_q v_N(t)$$

where $S_q$ is the filtering operator of order $q$. For the filtering, we use the exponential filter of order $q$ [19]. From $w_p(t)$, we first compute $V_p w_p$ for $v_N(t)$ and then apply the filtering operator, $S_q V_p w_p$. The obtained filtered solution is projected back to the reduced space by $V_p^T S_q V_p w_p$. Thus the filtered solution in the reduced space is simply given by the following transformation

$$w_p(t) = V_p^T S_q V_p w_p := \mathfrak{F}_p w_p.$$

Since the filtering operator in the reduced space $\mathfrak{F}_p = V_p^T S_q V_p$ has the size of $p \times p$, the filtering can be done easily with a small number of $p$. For the detailed explanation of the filtering method that we use in this paper is provided in Section 4.2.2.

## 4. Post-processing of sample solutions

The Chebyshev spectral approximation of Eq. (3.2) suffers from the Gibbs phenomenon for discontinuous solutions. The sample solution is then affected by the Gibbs oscillations and degrades the overall quality of the sample. These result in two drawbacks of the POD approximation: 1) the value of $p$ is large as the singular values decay slowly, hence the reduction order is large and 2) the solution $w_p(t)$ in the reduced model is poorly conditioned making the POD approximation inaccurate and causing a potential instability. These problems can be cured by improving the quality of the sample solutions. In this section, we propose a post-processing method of the sample solutions to construct well-conditioned POD bases. With the post-processing, we do not need to modify the PDE solver for $v_N(t)$ or $w_p(t)$, but simply generate the sample solutions with the same PDE solver, e.g. the Chebyshev spectral method for our case, Eq. (3.2), and the same method for the reduced solution $w_p(t)$, Eq. (3.4). The proposed method is composed of the following steps.

Step 1 [Reconstruction] The Gegenbauer reconstruction of oscillatory samples.
Step 2 [Regularization] The regularization of the reconstructed samples using various methods such as the smoothing or filtering methods.
Step 3 [Re-projection] The projection of the regularized samples to the original solution space, e.g. the Chebyshev polynomial space of degree $N$.
Step 4 [Re-initialization] The solution $w_p(t_n)$ is set to be $V_p^T v_N(s_{j\ell})$ with the sample solution $v_N(s_{j\ell})$ if $t_n = s_{j\ell}$.

4.1. **Step 1: Reconstruction via Gegenbauer projection.** Suppose that we are given a numerical approximation of $f(x)$, which suffers from the Gibbs phenomenon. The Gegenbauer reconstruction seeks the reconstruction $f_g(x)$ of $f(x)$ in the Gegenbauer polynomial space $G_{m_G}$

$$f_g(x) \in G_{m_G} = span\{G_l^\lambda(x),\ l = 0, 1, 2, \cdots, m_G,\ x \in [-1, 1]\},$$

where the Gegenbauer polynomials $G_l^\lambda(x)$ are defined by the following orthogonal relation in the Gegenabauer inner product $(\cdot, \cdot)_\lambda$ below

$$(G_l^\lambda, G_{l'}^\lambda)_\lambda := \frac{1}{h_l^\lambda} \int_{-1}^{1} (1 - x^2)^{\lambda - \frac{1}{2}} G_l^\lambda G_{l'}^\lambda dx = \delta_{ll'}.$$

Here $\delta_{ll'}$ is the Kronecker delta and $h_l^\lambda$ are the normalization factors given by

$$h_l^\lambda = \frac{\pi 2^{1-2\lambda}\Gamma(l+2\lambda)}{n!(n+\lambda)\Gamma(\lambda)^2},$$

where $\Gamma(\cdot)$ is the Gamma function. The three-term recurrence relation is given by

$$\begin{aligned} G_0^\lambda(x) &= 1, \quad G_1^\lambda(x) = 2\lambda x, \\ G_l^\lambda(x) &= \frac{2(l+\lambda-1)}{l} x G_{l-1}^\lambda(x) - \frac{(l+2\lambda-2)}{l} G_{l-2}^\lambda(x), \; l \geq 2. \end{aligned}$$

We will use the above recurrence relation to compute the Gegenbauer polynomials. If $l$ is large, the Gegenbauer projection, with finite precision becomes ill-posed by round-off errors. The Gibbs complimentary condition, however, suggests that the dimension $dim(G_{m_G})$ of $G_{m_G}$, is smaller than the dimension of the numerical solution space, i.e. $m_G \ll N$ [12], which helps to compute the Gegenbauer expansion in a stable manner. More properties of the Gegenbauer polynomials are found in [20].

Let $f(x)$ be piecewise analytic on the interval $[-1,1]$. Without loss of generality, suppose that the function $f(x)$ is analytic in a subinterval $[a,b] \subset [-1,1]$ and has jumps at $x = a$ and $x = b$. Let $\xi$ be the linear map $\xi : x \in [a,b] \to [-1,1]$ such that

$$\xi = \frac{2}{b-a}(x-b) + 1.$$

Further we assume that we are given a set of the Fourier coefficients of $f(x)$, $\{\hat{f}_k | \hat{f}_k = \frac{1}{2}\int_{-1}^{1} f(x)\exp(-ik\pi x)dx, k = -N, \cdots, N\}$. Then the Gegenbauer reconstruction $f_g(x)$ in $x \in [a,b]$ is given by the projection of the Fourier approximation of $f(x)$ onto the Gegenbauer space as below

$$f_g(x) = \sum_{l=0}^{m_G} g_l G_l^\lambda\big(\xi(x)\big) := \sum_{l=0}^{m_G} \big(f_N(x), G_l^\lambda(\xi(x))\big)_\lambda G_l^\lambda\big(\xi(x)\big),$$

where $f_N(x) = \sum_{k=-N}^{N} \hat{f}_k \exp(ik\pi x)$ is the Fourier partial sum of $f(x)$ based on $\{\hat{f}_k\}$. The maximum error between $f(x)$ and $f_g(x)$ then decays uniformly in the interval of $(a,b)$, i.e. $||f(x) - f_g(x)||_\infty \to 0$ if $0 < \alpha < 1$ and $0 < \beta < 1$ for $\lambda = \alpha N$ and $m_G = \beta N$ [12]. There is no precise theory known on how to find the optimal values of $\alpha$ and $\beta$. However, the values of $\lambda$ and $m_G$ should be less than $N$. Otherwise $f_g(x)$ recovers $f_N(x)$ instead of $f(x)$ as $N$ increases.

For the Chebyshev spectral solution, we replace the Fourier partial sum with the Chebyshev partial sum, also denoted by $f_N(x)$, as

$$f_N(x) = \sum_{k=0}^{N} \hat{f}_k T_k(x),$$

where $T_k(x)$ is the Chebyshev polynomial of degree $k$ and the Chebyshev coefficients, $\hat{f}_k$, are given by

$$\hat{f}_k = \frac{2}{\pi c_n} \int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} f(x)\, T_k(x)\, dx.$$

Here $c_n = 2$ if $n = 0$ and $c_n = 1$ otherwise. The Gegenbauer reconstruction $f_g(x)$ on $[a, b]$ based on $f_N(x)$ is given by the projection of the Chebyshev approximation onto the Gegenbauer space

$$f_g(x) = \sum_{l=0}^{m_G} g_l G_l^\lambda\big(\xi(x)\big) = \sum_{l=0}^{m_G} \big(f_N(x), G_l^\lambda(\xi(x))\big)_\lambda G_l^\lambda\big(\xi(x)\big).$$

Let $Q \in \mathbb{R}^{(m_G+1)\times(N+1)}$ be the transform matrix whose $(l, k)$ elements are given by

$$Q_{lk} = \big(T_k(x(\xi)), G_l^\lambda(\xi(x))\big)_\lambda = \frac{1}{h_l^\lambda} \int_{-1}^{1} (1-\xi^2)^{\lambda-\frac{1}{2}}\, T_k(x(\xi))\, G_l^\lambda(\xi)\, d\xi.$$

Since the subintervals are arbitrary, the above integral has no closed form and is computed using the quadrature rule. Let $\vec{f}_N = (\hat{f}_0, \hat{f}_1, \cdots, \hat{f}_N)^T$ and $\vec{f}_g = (g_0, g_1, \cdots, g_{m_G})^T$. Then the Gegenbauer reconstruction coefficients are found by the projection of the Chebyshev partial sum to the Gegenbauer space by the following linear equation

$$\vec{f}_g = Q \cdot \vec{f}_N.$$

For the Chebyshev Gegenbauer reconstruction the analytic conditions for convergence are not found in literature. We use similar conditions as the Fourier case, $\alpha \ll 1$ and $\beta \ll 1$ for $\alpha$ and $\beta$ of $\lambda = \alpha N$ and $m_G = \beta N$.

In this paper, we use the first derivative of the Chebyshev partial sum $f_N$ to find the edge. The derivative of $f_N(x)$ becomes $f_N'(x) = \sum_{k=0}^{N} \hat{f}_k\, T_k'(x)$. Since $f_N'(x)$ is a Chebyshev polynomial of degree at most $N$, we can rewrite $f_N'(x)$ as $f_N'(x) = \sum_{k=0}^{N} \tilde{f}_k\, T_k(x)$, where using the properties of the Chebyshev polynomials $\tilde{f}_k$ are given [19]

$$\tilde{f}_k = \frac{2}{c_n} \sum_{\substack{j=k+1 \\ j+k=odd}}^{N} j\, \hat{f}_j.$$

Thus the derivative of $f_N(x)$ in terms of $\hat{f}_k$ is given by

$$f_N'(x) = \sum_{k=0}^{N} \tilde{f}_k T_k(x) = \sum_{k=0}^{N} \left[ \sum_{\substack{j=k+1 \\ j+k=odd}}^{N} j \hat{f}_j \right] T_k(x).$$

Let the concentration be $S_N[f_N(x)] = f_N'(x)$. The edge location $x_c$ is determined such that the local maximum of $|f_N'(x)|$ exists at $x_c$ and the following condition is satisfied

$$|f_N'(x_c)]|/\max_x |f_N'(x)| \geq \gamma$$

where $\gamma$ is a positive constant, $\gamma \leq 1$. In this paper we choose $\gamma = 0.3$.

4.2. **Step 2: Regularization.** The reconstructed solution with the Gegenbauer post-procesing is a piecewise analytic function with several jump discontinuities. Thus the direct projection of the reconstruction back to the Chebyshev space will suffer from the Gibbs phenomenon. For this reason, the reconstruction should be regularized near the jump discontinuities prior to the re-projection back to the original space. In this work, we use 1) Gaussian smoothing and 2) spectral filtering. For the smoothing, we first replace the boundary values $f_g(b_i^-)$ and $f_g(a_{i+1}^+)$ of each subdomain with the average value of those boundary values of the two adjacent subdomains $[a_i, b_i]$ and $[a_{i+1}, b_{i+1}]$ as

$$f_g(b_i) = f_g(a_{i+1}) = \big(f_g(b_i^-) + f_g(a_{i+1}^+)\big)/2$$

where the superscript $+$ and $-$ denote the right and left limits.

4.2.1. *Gaussian smoothing.* The smoothed solution denoted by $f_g^S(x)$ from the Gegenbauer reconstructed solution $f_g(x)$ with the Gaussian smoothing is given by the convolution of the reconstructed function with the Gaussian kernel

$$f_g^S(x) = \int \mathbb{G}(y-x) f_g(y) dy,$$

where $\mathbb{G}$ is the Gaussian kernel given by $\mathbb{G}(x) = \frac{1}{\sqrt{2\pi s^2}} \exp(-x^2/2s^2)$, for some $s > 0$. Using $\kappa$ neighboring points both in the right and left to each smoothing point $x_j$, we take the smoothed solution $f_g^S(x)$ such that

$$f_g^S(x_j) = \sum_{i=-\kappa}^{\kappa} c_i\, f(x_{j+i}) \quad \text{with} \quad c_i = \frac{\mathbb{G}(x_{j+i})}{\sum_{l=-\kappa}^{\kappa} \mathbb{G}(x_{j+l})},$$

where $x_{j+i}$ can be in the neighboring subdomain.

4.2.2. *Spectral filtering.* For the spectral filtering, we first project the Gaussian smoothed solution $f_g^S(x)$ of $f_g$ onto the Chebyshev polynomial space using the procedure described in the following section. Let $P_N f_g^S(x)$ be the resulting projected polynomial and $\hat{f}_k$ be the expansion coefficients corresponding to the Chebyshev polynomials. Here note that $\hat{f}_k$ are obtained by the direct projection of the Gegenbauer reconstruction $f_g^S(x)$. Since $f_g^S(x)$ may be still sharp near the jump discontinuities it is possible for the projection, $P_N f_g^S(x)$, is oscillatory near the jump discontinuities. To further reduce the oscillations while keeping the sharpness as much as possible, the filtered re-projection is obtained by the following

$$F_N^\rho[P_N f_g^S(x)] = \sum_{k=0}^{N} \rho(k/N; q) \hat{f}_k T_k(x), \tag{4.1}$$

where the filter function $\rho(k/N; q)$ is adopted to be the exponential filter as below $\rho(k/N; p) = \exp\big(-\eta(k/N)^{q(d(x))}\big)$. Here $q$ is the filtering order and $\eta$ is a positive constant such that $\exp(-\eta) = \epsilon_M$. And $\epsilon_M$ is machine zero. In general, the filtering order, $q$, is a function of $d(x)$ with $d(x)$ defined as the distance from the nearest edge. If $q$ is a constant for every $x$, the filtering, Eq. (4.1), is a global operation and if $q$ depends on $x$ it is a local operation requiring the adaptation condition of $q$ as a function of $d(x)$. It was shown that this adaptive approach improves the quality of highly complex flow solutions under the presence of multiple shocks [21]. The adaptive filtering can also be used for the high order mollification keeping the convergence near the edges enhanced [22]. For the adaptive filter, at a given collocation point $x_j$, let $x_e$ be the location of the nearest edge from $x_j$. Then $d(x) = |x_j - x_e|$. The adaptive filter chooses the filtering order $q$ so that the filtering order gradually decreases toward the edge in the following way: $q \to \infty$ for $d \to \infty$, and $p \to 0$ for $d \to 0$. We use the global filter for the numerical examples in Section 5 and the adaptive filtering for variable filtering order $q$ will be considered in our future work.

4.3. **Step 3: Re-projection.** Once the reconstruction is regularized, the regularized solution is re-projected onto the Chebyshev polynomial space so that it can be used as the sample solution to create a POD basis set. Let $f_g^R(x)$ be the regularized solution via the Gaussian smoothing or spectral filtering or both The projection of $f_g^S(x)$ is given by

$$P_N f_g^R(x) = \sum_{k=0}^{N} \hat{f}_k \, T_k(x),$$

where

$$\hat{f}_k = \frac{2}{\pi c_n} \int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} \, f_g^R(x) \, T_k(x) dx.$$

Note that $f_g^R(x)$ can be piecewise continuous, hence the above integral has to be computed by a composite quadrature rule considering the discontinuous points.

4.4. **Step 4: Re-initialization.** The regularized Gegenbauer samples are used for the construction of a new set of POD bases. The new samples are smooth, yet close to the exact solution and the singular values decay desirably. However, according to the given PDE the solution easily develops discontinuity with time even though the samples or the POD bases are smooth. To take advantage of the already post-processed samples we use the re-initialization of the solution in the reduced space as follows. For each time step $t_n = n\Delta t$ when solving the reduced model for the POD approximation, $w_p^n$, using Eq. (3.5), first we check whether $t_n$ coincides with one of the sampling time $s_{j\ell} = j\ell\Delta s$. Here we assume that we collect the samples uniformly without loss of generality. If $t_n = s_{j\ell}$, then we replace the reduced solution $w_p^n = V_p^T v_N(s_{j\ell})$ using the post-processed sample solution $v_N(s_{j\ell})$ as the initial condition for the time integration at $t_n$ towards $w_p^{n+1}$. We can choose the time stepping $\Delta t$ for the reduced solution $w_p(t)$ such that $s_{j\ell}$ are the integer multiples of $\Delta t$ or we can also choose the time stepping arbitrarily and adjust the time stepping accordingly to make $t_n$ coincide with $s_{j\ell}$ when $t$ approaching $s_{j\ell}$.

## 5. Numerical examples

5.1. **POD basis for discontinuous solutions.** To show the behavior of the singular values discussed in Section 3 we consider the following one-dimensional advection equation

$$u_t + u_x = g(x), \quad t > 0, \quad x \in [-1, 1], \tag{5.1}$$

with the initial condition $u(x, 0) = u_0(x)$ and the boundary condition $u(-1, t) = h(t)$. Equation (5.1) is found in various applications such as traffic flow models [23]. We will show how the singular values behave when some sample solutions are discontinuous and the discontinuity is either fixed or moving with time using the following cases:

1) $u_0(x) = \sin(\pi x)$, $h(t) = \sin(\pi(-1-t))$ and $g(x) = 0$,
2) $u_0(x) = \sin(\pi x) + H(x)$, $h(t) = \sin(\pi(-1-t))$ and $g(x) = \delta(x)$,
3) $u_0(x) = \sin(\pi x) + H(x)$, $h(t) = \sin(\pi(-1-t))$ and $g(x) = 0$.

Here $H(x)$ is the Heaviside function and $\delta(x)$ is the Dirac delta function. These cases have the exact solutions $u(x, t) = \sin(\pi(x - t))$, $u(x, t) = \sin(\pi(x - t)) + H(x)$ and $u(x, t) = \sin(\pi(x-t)) + H(x-t)$, respectively. The solution for Case 1) is smooth both in space and time. The solution for Case 2) has a jump discontinuity at $x = 0$ for all time $t$. The solution for Case 3) has a jump discontinuity, but the discontinuity propagates with time. For the Chebyshev spectral approximation, we choose the CGL collocation points $x_j$ with $N = 384$. We take $\ell = N/4$ and $\Delta s = T/N_s$ with $N_s = m\ell$ for $(m + 1)$ samples to construct POD basis. Figure 1 shows the singular values of the sample matrix $X$ with various $m$ and $T$. The blue line with circle symbols represents the singular values for Case 1), the black line with cross symbols for Case 2) and the red line with bullet symbols for Case 3). The top figures show the case that the final time is $T = 0.4$ and the bottom figures show the case that the final time is $T = 0.8$. The left figures are for $m = 40$ and the right figures for $m = 80$. As shown in the figures, the singular values for Cases 1) and 2) decay fast; the singular values $\sigma_k$ become as small as machine zero if $k \geq 3$ for Case 1) and $k \geq 4$ for Case 2). Here we note that the solution for Case 2) is discontinuous at $x = 0$ for every $t$, but the first 3 singular values are significant and the rest is ignorable. The jump discontinuity for Case 3), however, propagates with time in space and makes the singular values decay slowly. If the time stepping for samples is large (figures in top left and bottom figures) the singular values decay slowly. For the case of $m = 80$ and $T = 0.4$, the singular values drop to machine zero around $k = 54$. Note that the singular values in these figures are based on the exact solutions. The numerical solutions with the spectral collocation method will be oscillatory near the jump discontinuity as shown in the following examples.

5.2. **Advection equation.** To apply our proposed method, we first consider the linear advection equation with the discontinuous initial condition

$$\begin{aligned} u_t + u_x &= 0, \quad t > 0, \quad x \in [-1, 1], \\ u(x, 0) &= u_0(x) = \begin{cases} 1, & -0.25 \leq x \leq 0.2 \\ 0, & \text{otherwise} \end{cases}. \end{aligned}$$

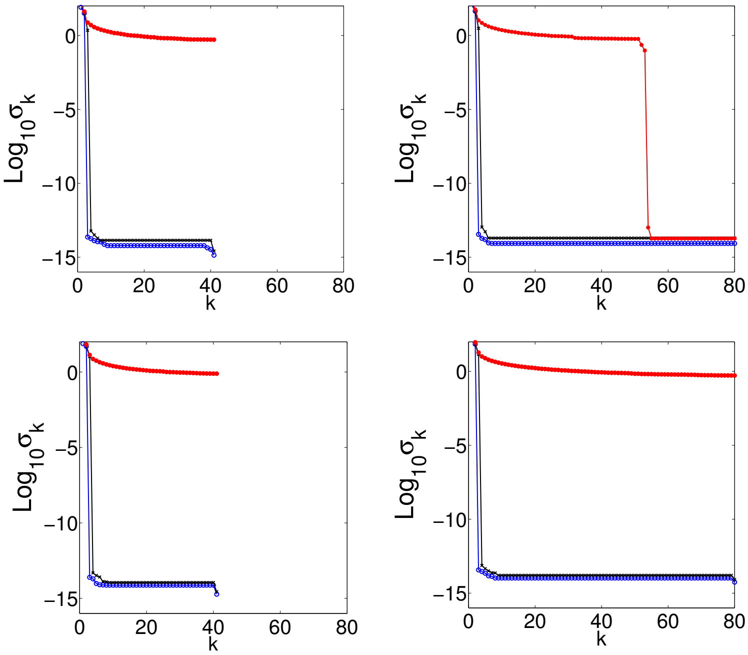

FIGURE 1. Singular values with $N = 384$. Case 1): blue line with circle (-O), Case 2): black line with cross (-×) and Case 3: red line with bullet (-●). Left columns: $m = 40$. Right columns: $m = 80$. Top: $T = 0.4$. Bottom: $T = 0.8$.

For this simple problem, the reduced equation is simply given by from Eq. (3.4)

$$\frac{dw_p(t)}{dt} = -V_p^T D_N V_p w_p(t).$$

The initial condition for $w_p(0)$ is given by $w_p(0) = V_p^T u_0$.

The numerical solution $u_N(x,t)$ is sought in the Chebyshev space of degree $N$, $C_N = span\{T_k(x)|k = 0, 1, \cdots, N\}$. The initial condition is discontinuous and not in $C_N$. The interpolation operator is applied to the initial condition so that the Chebyshev initial condition $u_N(x,0)$ is in the polynomial space. We also apply the spectral filtering to smooth $u_N(x,0)$. Since the equation is linear, the filtering is not applied to the subsequent approximations. Thus all the numerical solutions are oscillatory near the jumps including $u_N(x,0)$. This equation is a archetypical problem for more general nonlinear problems considered in the following sections. The exact solution, $u_e(x,t)$ is given by

$$u_e(x,t) = \begin{cases} 1, & -0.25 \le x - t \le 0.2 \\ 0, & \text{otherwise} \end{cases}.$$

We set $\ell = N/4$ and let $N_s = m\ell$ with the number $m = 80$. Then the sample time spacing is $\Delta s = T/N_s$. The time integration is done with the 3rd-order TVD Runge-Kutta method [18].

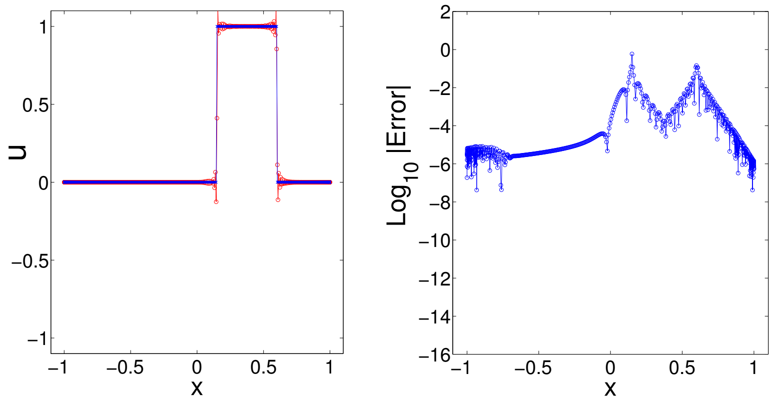

FIGURE 2. Left: the numerical solution (red) and the exact solution (blue). Right: the pointwise errors for $T = 0.4$ with $N = 512$.

Since the given problem is linear, the given discontinuous initial condition simply translates with time and we do not need to impose the re-initialization.

The left figure of Fig. 2 shows a sample numerical solution (red) and the exact solution (blue) at the final time $T = 0.4$. The right figure shows the pointwise errors between the approximation and exact solution with the polynomial of degree $N = 512$. Here the order of filtering is $q = 16$. As shown in the right figure, the errors at the jumps, $x = T - 0.25$ and $x = T + 0.2$, do not decay.

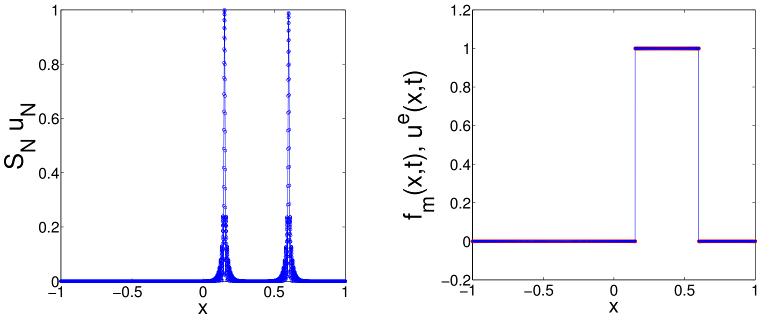

FIGURE 3. Left: the concentration $S[u_N]$ at $T = 0.4$ with $N = 512$. Right: the exact solution (blue) and the post-processed solution (red).

Now we apply the Gegenbauer reconstruction method to the obtained $(m + 1)$ sample solutions for post-processing. For the Gegenbauer reconstruction, we use the conditions for $\lambda$ and $m_G$ as $\lambda = N/64$ and $m_G = N/64$. Figure 3 shows the normalized concentration $S_N[u_N]$, the exact solution and the post-processed solution at $T = 0.4$ with $N = 512$. As shown in

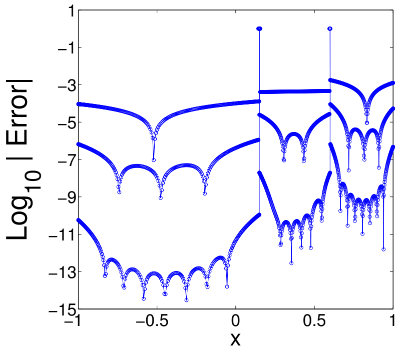

FIGURE 4. Pointwise errors of the post-processed solution at $T = 0.4$ with $N = 128, 256, 512$ and $\lambda = N/64$, $m_G = N/64$.

the left figure, the jump discontinuities are well detected with the concentration $S[u_N]$. The right figure shows that the Gegenbauer reconstruction well matches the exact solution. Figure 4 shows the pointwise errors of the Gegenbauer post-processed solution of $u_N(x,t)$ in logarithmic scale for $N = 128, 256, 512$. As shown in the figure, the Gegenbauer post-processed solution converges uniformly in each interval up to the discontinuity as $N$ increases. Because of the error in the location of the detected discontinuity, the error does not decay near the jumps. The Gegenbauer post-processed solution is more accurate than the numerical solution $u_N(x,t)$. The Gegenbauer reconstruction is regularized using the Gaussian smoothing. Figure 5 shows

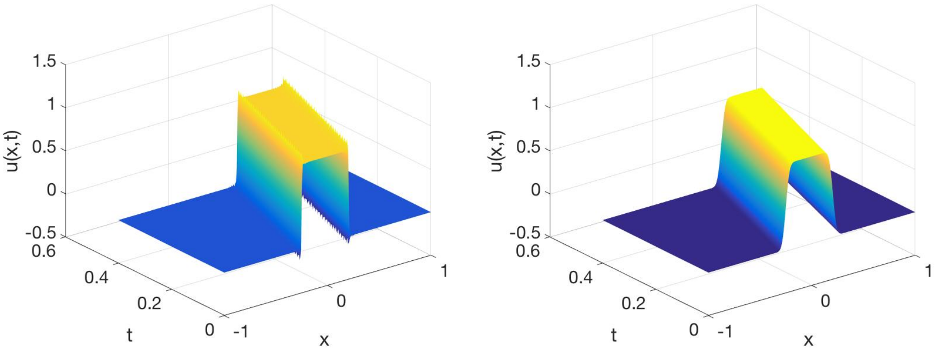

FIGURE 5. Left: Sample solutions. Right: Gegenbauer post-processed sample solutions for $T = 0.4$ with $N = 384$ and $\lambda = N/64$, $m_G = N/64$.

81 sample solutions and their Gegenbauer post-processed solutions for $m = 80$. As shown in the figure, the post-processed sample solutions are free of Gibbs oscillations. Once the post-processing is completed we compute the POD bases, for which we use the truncation criteria as $\sigma_{p+1} \leq \epsilon_p = 10^{-3}$. With those POD bases constructed, we solve the reduced model for the

POD approximations. For the POD approximations, we use a finer time stepping $\Delta t = T/M$ and $M = C_t N_s$ with $C_t = 80$. The final solutions through the POD approximation at $T = 0.4$ are given in Fig. 6. The left figure shows the POD solution (red) without the post-processing and the right figure the POD solution (red) with the post-processing. As shown in the figure, the POD approximation with the post-processing diminishes the Gibbs oscillations.

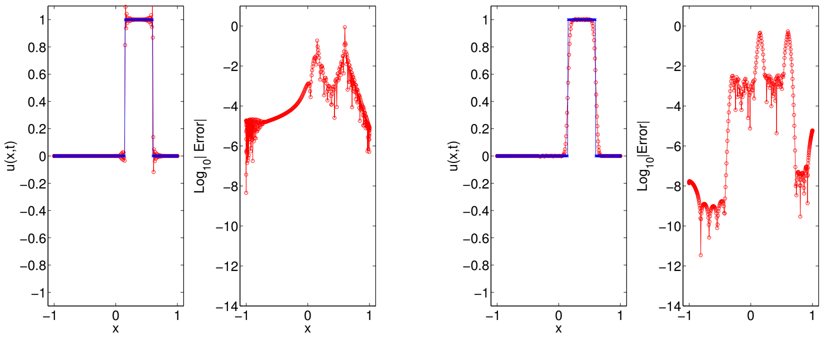

FIGURE 6. POD solutions. Left: Without the post-processing. Right: With the post-processing. $T = 0.4$ with $N = 384$ and $\lambda = N/64$, $m_G = N/64$.

TABLE 1. CPU time (sec) for the sample, Gegenbauer post-processing, full and POD solutions. $N = 384$.

| $C_t$ | Sample solutions | Post-processing | Full solution | POD solution |
|---|---|---|---|---|
| $C_t = 40$ | 3.103510 | 8.165839 | 116.294333 | 7.717133 |

Table 1 shows the CPU time in second for the sample, Gegenbauer post-processsing, full and POD solutions with $N = 384$. The full and POD solutions are obtained with the time stepping $\Delta t = \Delta s/C_t$. The table shows each CPU time for $C_t = 40$. That is, the time stepping $\Delta t$ is 40 times smaller than the time stepping used for computing the sample solutions. The CPU time for the SVD is not included because it is insignificant compared to the CPU times for other procedures. The CPU time for the SVD is about $0.005839$ $sec$. The total CPU time toward the completion of the POD solutions is about $18.992321$ $sec$, which is much smaller than the CPU time for the full solutions.

5.3. **Burgers' equation.** In [24], we considered viscous Burgers' equation with the Chebyshev collocation method based on the POD method and demonstrated that the POD approximation for the Burgers' equation is efficient and accurate. In this paper we extend the work to the inviscous Burgers' equation with the initial condition that is smooth and makes the solution become discontinuous with time $t > 0$ and $x \in [-1, 1]$ as below

$$u_t + \frac{1}{2}\left(u^2\right)_x = 0, \quad u(x,0) = u_0(x) = \frac{1}{4} + \sin(\pi(x+1)),$$

with the periodic boundary condition. The given initial condition is smooth and a shock forms at a later time $t = t_s$. Once the shock forms, the shock propagates according to the Rankine-Hugoniot relation. The above equation is solved until the final time $T = 1$. For the sample solutions, we use the filtering order $q = 32$. Here note that the filtering is not necessary if $t \ll t_s$ and one can adaptively change the filtering order or apply the local filtering adaptively near the shock area once the shock forms. For the numerical experiment, we fix the filtering order as $q = 32$ for all $t$. We choose $N = 384$, $\ell = N/2$ and $m = 80$. We set $N_s = m\ell$ and $\Delta s = T/N_s$. For the POD approximation in the reduced space, we use a finer time spacing given by $\Delta t = T/M$ with $M = C_t\, N_s$ with $C_t = 40$, i.e. $\Delta t = \Delta s/C_t$. In this case, the time spacing for the POD approximation is $\Delta t = 1.6276 \times 10^{-6}$. For the Gegenbauer projection, we use the same conditions as before, i.e. $\lambda = N/64$ and $m_G = N/64$. For the Gaussian projection we use $s = 0.05$ and $\kappa = 10$. Since the given PDE is nonlinear and a discontinuity develops with time, we apply the re-initialization during the POD approximation in the reduced space. For the truncation condition, we use $\epsilon_p = 10^{-5}$.

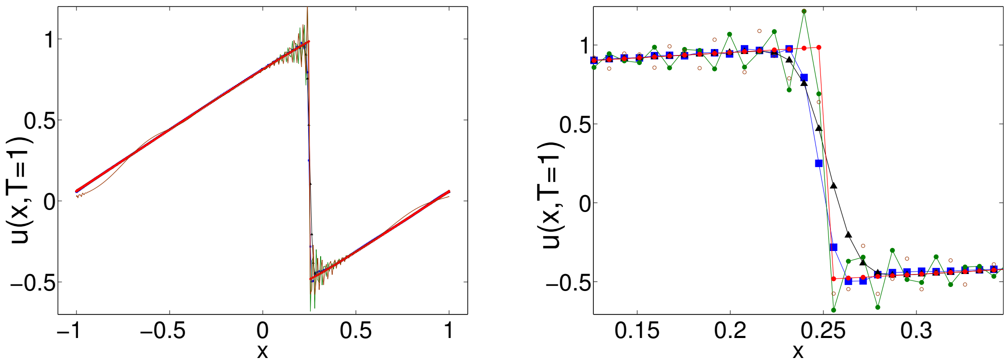

FIGURE 7. (Color online) Left: Various solutions at $T = 1$. Right: Oscillatory sample solution (-●), Gegenbauer projected solution (-●), Smoothed Gegenbauer projected solution (-▲), POD solution *without* the post-processing and re-initialization (o), POD solution *with* the post-processing and re-initialization (-■).

Figure 7 shows various solutions at the final time $T = 1$. The left figure shows the solutions in the whole domain and the right figure shows the solutions near the discontinuity. In the figure, the green solid line with the green dots (-●) represents the sample solution. The red line with the red dots (-●) is the Gegenbauer projected solution, which is a direct projection of the oscillatory sample solution (the green) to the Gegenbauer polynomial space collected on the Chebyshev collocation points $x_i$, $i = 0, \cdots, N$. The black solid line with the black triangle symbol (-▲) represents the smoothed Gegenbauer projection (-●). The blue line with the blue square symbol (-■) represents the POD solution. The brown solid line (—) in the left figure and the brown open circle (o) in the right figure represent the POD solution from the sample solutions without the Gegenbauer post-processing and re-initialization. The figure shows how oscillatory these solutions are. First of all, the figure shows that the sample solution and the POD solution without the Gegenbauer post-processing and re-initialization are the most oscillatory solutions. The left figure shows that the POD solution without the post-processing and re-initialization is not only oscillatory near the discontinuity but also oscillatory near the left domain boundary. This is because the POD bases from the sample solutions with $\epsilon_p = 10^{-5}$ do not well resolve the discontinuous solution over the whole domain. As the right figure shows the POD solution (brown circle, o) without the Gegenbauer post-processing and reoscillatory-initialization is oscillatory near the discontinuity. As the discontinuity develops the filtering does not completely suppress the oscillations. Based on the oscillatory sample solutions the Gegenbauer reconstructions are found and projected back to the Chebyshev space. As shown in the figure, the projected solution is free of the Gibbs oscillations and represents the developed shock clearly (-●). For the POD approximation we smoothed the projected solution (black solid line with black triangle symbols). The smoothed projection is smooth near the shock without the Gibbs oscillations. Finally the figure shows that the POD solution (-■) with the post-processing and reinitialization is close to the Gegenbauer reconstructed solution while the oscillations are much dimished. Overall the POD solution with the Gegenbauer post-processing and re-initialization yields a solution that represents the discontinuous solution reasonably.

TABLE 2. CPU time (sec) for the sample, Gegenbauer post-processing, full and POD solutions.

| $C_t$ | Sample solutions | Post-processing | Full solution | POD solution |
|---|---|---|---|---|
| $C_t = 10$ | 22.884629 | 6.739353 | 224.888082 | 21.159279 |
| $C_t = 20$ | 22.884629 | 6.739353 | 444.481493 | 42.354732 |
| $C_t = 40$ | 22.884629 | 6.739353 | 900.438388 | 83.621026 |

Table 2 shows the CPU time in second used for the sample solutions, the Gegenbauer post-processing, full and POD solutions. For the sample solutions, we used $m = 80$ and $\ell = N/2$ so that $\Delta s = T/N_s$ with $N_s = m\ell$. The full and POD based solutions are obtained with the time stepping $\Delta t = \Delta s/C_t$. The table shows the CPU time for different $C_t$ for the full and POD solutions, $C_t = 10, 20$ and 40. For each case, the total amounts of time for the sample solutions and the Gegenbauer post-processing are same. As shown in the table, it took less than 7 seconds to complete the Gegenbauer post-processing, which includes the edge detection, Gegenbauer reconstruction, projection to the Chebyshev points and Gaussian smoothing. Compared to the actual calculation of the POD solutions, the total amount of time for the Gegenbauer post-processing is insignificant. The table also shows that the total amount of time for the full solutions is about 10 times larger than the total amount of time for the POD solutions. That is, the POD approximation with the Gegenbauer post-processing is much faster and more efficient than the full approximations while it is accurate and less oscillatory. Here note that in order to obtain the POD solutions, we need the sample solutions and the Gegenbauer post-processing. Thus the actual amount of time to complete the POD approximations is the sum of the total amounts of time for the sample solutions, Gegenbauer post-processing and the POD solutions

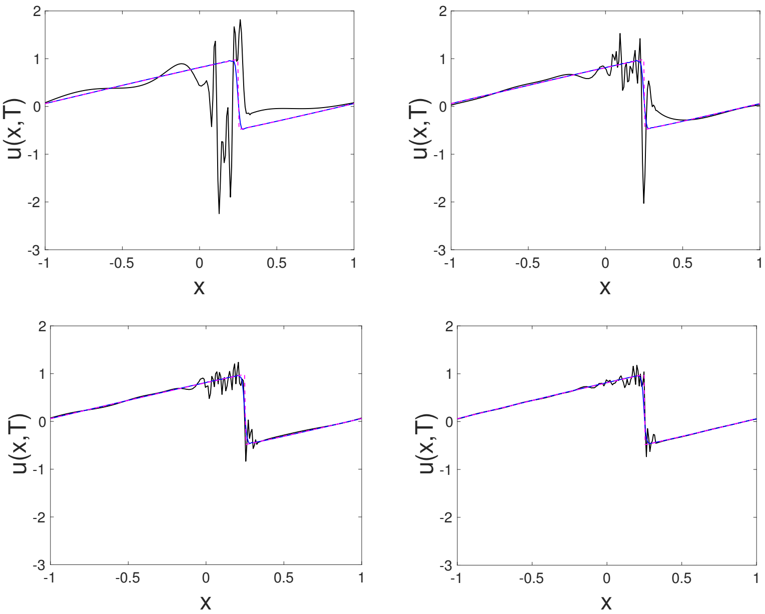

FIGURE 8. Solutions to Burgers' equation at the final time $T = 1$: the POD solutions without the re-initialization (black solid line), the POD solutions with the re-initialization (blue solid line) and the exact solution (dashed line in magenta). $N_s = 20, 30, 40, 60$ from top to bottom and left to right.

in the table. Even adding up all these, the actual time for the POD completion is still smaller than the time for the full solutions. Here we note that the CPU times consumed for the SVD is not included in the table because they are insignificant, e.g. they are $0.010067$ $sec$ for $N = 384$ and $0.013045$ $sec$ for $N = 512$.

5.3.1. *Re-initialization.* To justify the re-initialization we compare the POD solutions with and without the re-initialization after the Gegenbauer post-processing. We consider the same Burgers' equation in the previous section. Figure 8 shows the exact solution (dashed line in magenta), the POD solution with the re-initialization (blue solid line) and the POD solution without the re-initialization (black solid line) at the final time $T = 1$ and with $N = 384$. The numbers of the POD basis used for the figures are $20, 30, 40$ and $60$, from top to bottom and left to right, out of total $81$ samples. As shown in those figures, the POD solutions without the re-initialization are highly oscillatory near the shock while the POD solutions with the re-initialization are close to the exact solution with the oscillations much reduced. We also observe that the POD solutions without the re-initialization become less oscillatory and close to the exact solution as the number of the POD basis increases. However, the POD solutions with the re-initialization are similar for every case in Fig. 8.

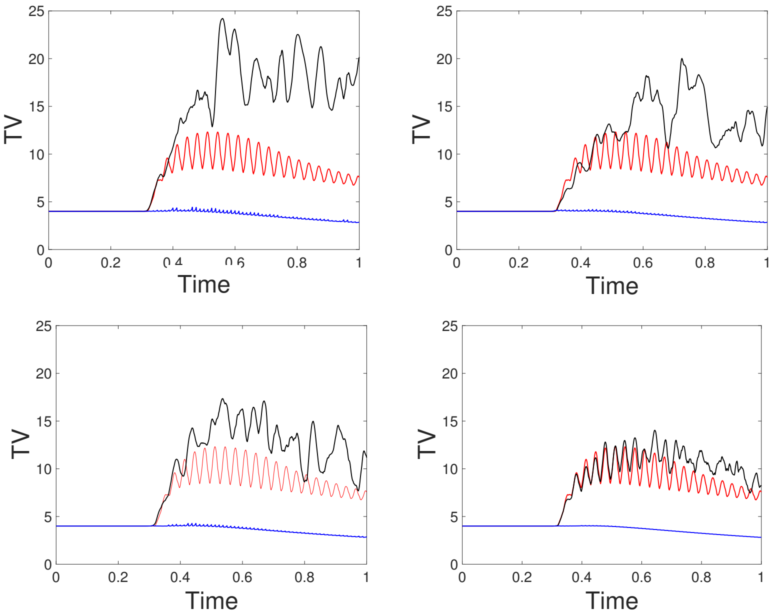

FIGURE 9. Total variation norms versus time for the POD solutions without the re-initialization (black solid line), the POD solutions with the re-initialization (blue solid line) and the full solutions (red solid line).

Figure 9 shows the total variation (TV) norm of the solution with time. The TV norm of the discrete solution, $v(x,t)$, at $t$ is defined by $TV(t) = \sum_j |v(x_{j+1},t) - v(x_j,t)|$. Figure 9 shows the TV norms versus time of the POD solutions without the re-initialization (black solid line), the POD solutions with the re-initialization (blue solid line) and the full solution (red solid line). The number of the POD basis used in each figure is same as in Fig. 8. Those figures show that the TV norms increase drastically after the shock forms for the POD solutions without the re-initialization and the full solutions. However, the variations of the TV norms for the POD solutions with the re-initialization are small and the solutions are smooth.

Figure 10 shows the pointwise errors of the POD solutions without the re-initialization (black solid line), the POD solutions with the re-initialization (blue solid line) and the full solution (red solid line). The number of the POD basis used for each figure is same as in Figs. 8 and 9. Those figures show that the pointwise error profiles for the POD solutions with the re-initialization are sharper near the shock than those for the POD solutions without the re-initialization and full solutions. As we expect, the full solutions are more accurate in the smooth region than the POD solutions with the re-initialization for the given $N$. Even though the POD solutions with the re-initialization are less accurate than the full solutions, they are obtained about 10 times faster than the full solutions (see Table 2), which is the main reason we use the POD method.

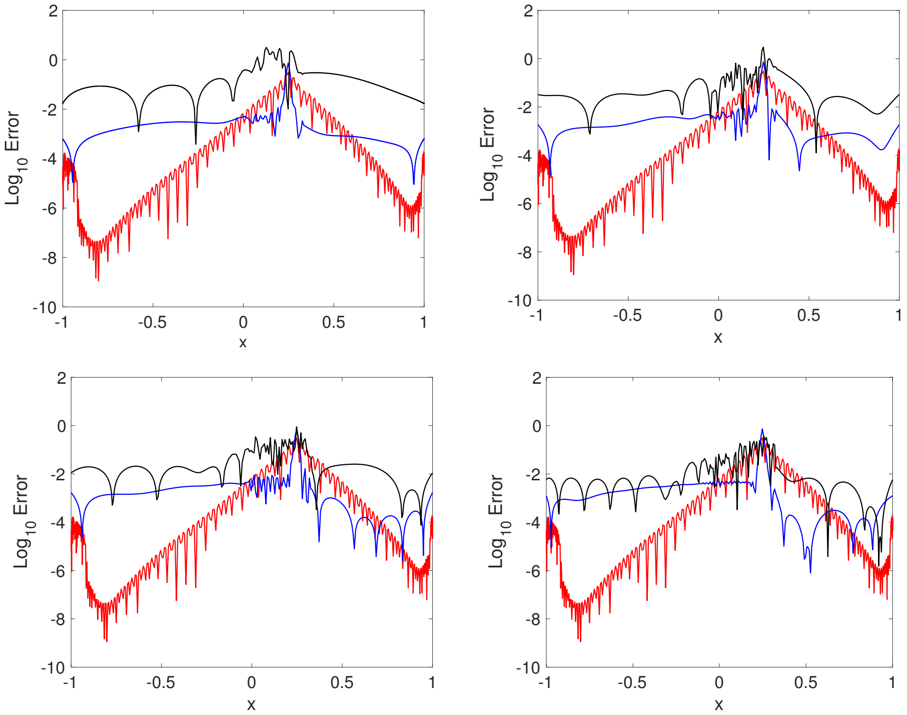

FIGURE 10. Pointwise errors, in logarithmic scale, of the POD solutions without the re-initialization (black solid line), the POD solutions with the re-initialization (blue solid line) and the full solutions (red solid line) at $T = 1$.

5.3.2. *Errors.* Table 3 shows the $L_1$, $L_2$ and $L_\infty$ errors of the full solutions and the POD solutions at $T = 0.2$. At $T = 0.2$, the solution to the Burgers' equation is smooth. Table 5 shows the errors for the POD solutions with $N_s = 11$ for $N < 256$ and $N_s = 13$ for $N = 256$. As expected for smooth problems the errors of the full and POD solutions are similar for each $N$. For this reason, the POD method is beneficial as it finds the solution faster with reasonable accuracy.

Table 4 shows the $L_1$, $L_2$ and $L_\infty$ errors of the full solutions and the POD solutions at $T = 1$. At $T = 1$, the exact solution is discontinuous and the full solutions are highly oscillatory near the shock. As there exists a shock, the errors are large compared to those for the smooth solutions in Table 3. The errors of the POD solutions with the re-initialization are larger than the errors of the full solutions. The $L_1$ and $L_2$ errors decay slowly for both the POD with the re-initialization and full solutions while the $L_\infty$ errors do not decay for both cases as expected for discontinuous problems.

5.4. **Lax problem.** Consider the 1D Euler equations for gas dynamics for the state vector $u$,

$$u = (\rho, \rho v, E)^T,$$

TABLE 3. $L_1, L_2$ and $L_\infty$ errors of the full solution and POD solution to Burgers' equation at $T = 0.2$. $N_s = 11(tol = 10^{-5})$ for $N < 256$, $N_s = 13(tol = 10^{-6})$ for $N = 256$.

| Method | N | $L_1$ error | $L_1$ order | $L_2$ error | $L_2$ order | $L_\infty$ error | $L_\infty$ order |
|---|---|---|---|---|---|---|---|
| | 32 | 1.5612 (-1) | – | 6.4609 (-3) | – | 1.5602 (-2) | – |
| Full | 64 | 1.6600 (-2) | 3.2 | 4.9705 (-4) | 3.7 | 1.9936 (-3) | 3.0 |
| Solution | 128 | 2.0705 (-4) | 6.3 | 3.9884 (-6) | 7.0 | 2.0003 (-5) | 6.6 |
| | 256 | 6.2572 (-8) | 11.7 | 7.8702 (-10) | 12.3 | 5.1004 (-9) | 11.9 |
| | 32 | 1.6350 (-1) | – | 6.8766 (-3) | – | 1.9359 (-2) | – |
| POD | 64 | 2.0855 (-2) | 3.0 | 5.5489 (-4) | 3.6 | 2.4801 (-3) | 3.0 |
| Solution | 128 | 1.1342 (-4) | 7.5 | 2.3402 (-6) | 7.9 | 1.5206 (-5) | 7.3 |
| | 256 | 1.0754 (-7) | 10.0 | 7.0528 (-10) | 11.7 | 4.0332 (-9) | 11.9 |

$(n) = 10^n$

TABLE 4. $L_1, L_2$ and $L_\infty$ errors of the full solution and POD solution to Burgers' equation at $T = 1$. $N_s = 11(tol = 10^{-5})$ for $N < 256$, $N_s = 13(tol = 10^{-6})$ for $N \geq 256$.

| Method | N | $L_1$ error | $L_2$ error | $L_\infty$ error |
|---|---|---|---|---|
| Full Solution | 128 | 3.2274 | 5.2303 (-2) | 3.0347 (-1) |
| | 256 | 3.4235 | 5.5447 (-2) | 6.9606 (-1) |
| | 384 | 3.0947 | 3.2406 (-2) | 3.0659 (-1) |
| | 512 | 2.5210 | 2.0858 (-2) | 1.8972 (-1) |
| POD Solution | 128 | 2.8602 | 9.0243 (-2) | 7.1116 (-1) |
| | 256 | 2.6220 | 6.2400 (-2) | 7.6757 (-1) |
| | 384 | 2.3659 | 4.6478 (-2) | 7.3753 (-1) |
| | 512 | 2.2641 | 3.9874 (-2) | 7.8855 (-1) |

$(n) = 10^n$

and the flux function $F(u)$, $F(u) = (\rho v, \rho v^2 + P, (E + P)v)^T$, where $\rho$, $v$, $P$ and $E$ are density, velocity, pressure and total energy, respectively. The pressure $P$ is determined by the equation of state

$$P = (\gamma - 1)\left(E - \frac{1}{2}\rho v^2\right),$$

where $\gamma = 1.4$ for the ideal gas. We consider the Lax problem specified with the following initial conditions

$$(\rho, v, P) = \begin{cases} (0.445, 0.698, 3.528), & x \leq 0 \\ (0.5, 0.0, 0.571), & x > 0 \end{cases}, \tag{5.2}$$

with the Dirichlet boundary conditions at $x = \pm 1$. The solution is sought until the final time $T = 0.13$. Sample solutions are obtained with $N = 1024$, $m = 60$ and $\ell = N$, i.e. the sample time stepping is $\Delta s = T/(m\ell)$. For the POD based solutions we take the finer time spacing $\Delta t = \Delta s / C_t$ with $C_t = 20$. The exponential filter order is $q = 12$.

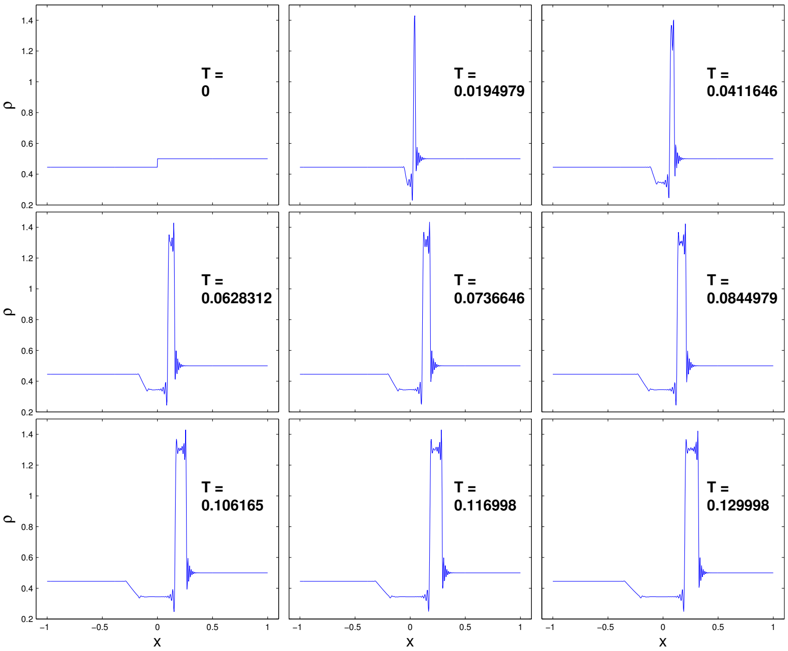

FIGURE 11. Sample density solutions with various times for the Lax problem, Eq. (5.2).

Figure 11 shows the sample density solutions at various times, $t = 0, 0.0194979, 0.0411646, 0.0628312, 0.0736646, 0.0844979, 0.106165, 0.116998, 0.129998$ from top to bottom and left to right in the figure. As the figures show, the shocks develop with time and the spectral solutions are oscillatory near the jump discontinuities. For the reconstruction, as in the previous examples, we first detect the discontinuities and apply the Gegenbauer reconstruction. However, we note that the edge detection with the derivative explained in the previous section is not as efficient as the Burgers' equation to identify the rarefaction region because the magnitude of the derivatives near the region is much smaller than the shock area where the solution profile is sharp. This implies that we need more sophisticated edge detection method with which all the edges are found successively from strong to weak edges. The concentration method with the derivative explain in Section 4.1 detects the strong shock areas clearly ($x \geq -0.05$) and we use the Gegenbauer reconstruction method for those areas only. For the rest, we use the sample solutions.

Figure 12 shows the reconstructed sample density solutions at various times as in Fig. 11. For the Gegenbauer reconstruction, we use the conditions $\lambda = N/128$ and $m_G = N/128$. As shown in the figure, the Gegenbauer reconstructions with the regularization are free of the Gibbs oscillations near the strong shocks and the overall reconstructed solutions are smooth.

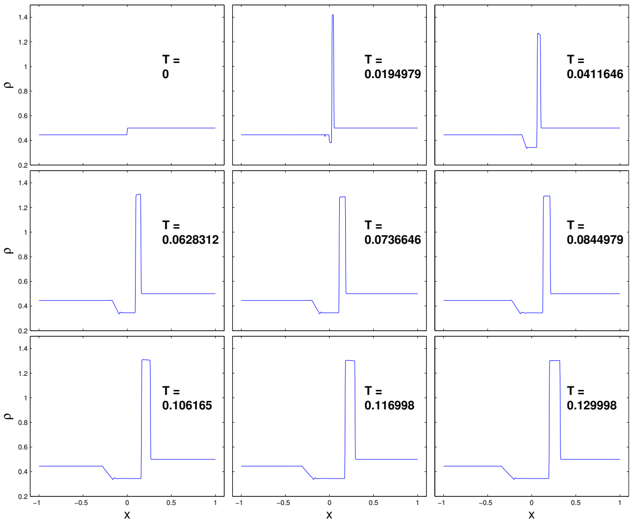

FIGURE 12. Reconstructed density solutions with various times for the Lax problem, Eq. (5.2).

Figure 13 shows the POD density solutions based on the Gegenbauer reconstructed solutions. We choose the order of the POD basis such that $p = m/2 = 30$. Here note that one can choose $p$ based on the truncation with the tolerance value. In general the optimal value of the tolerance level depends on the problem and parameter values used in the post-processing. Here we manually choose the value of $p$ to simply show how the proposed method behaves for the fully nonlinear problem. For the construction of the POD bases, we further smooth the post-processed sample solutions with the spectral filtering of order $q = 2$ in a global manner. Figure 13 shows the POD solutions. The POD solutions are oscillatory but the degree of oscillations is reduced with the filtering before the POD calculation.

Figure 14 shows the detailed density solutions with various methods – the oscillatory sample solution (-●), the Gegenbauer reconstruction (-●), the regularized Gegenbauer reconstruction with filtering (-▲) and the POD solution *with* the post-processing and re-initialization (-■). With the filtering of the processed samples, the POD solution at $t = 0.13$ becomes smoother and less oscillatory as the figure shows.

TABLE 5. CPU time (sec) for the sample, full and POD solutions.

| $C_t$ | Sample solutions | Post-processing $(\rho, \rho v, E)$ | Full solution | POD solution |
|---|---|---|---|---|
| $C_t = 20$ | 1964.815520 | 340.077863 | 40829.156065 | 681.097892 |

Table 5 shows the CPU times for the sample, full and POD solutions. The CPU time for the reconstruction includes each reconstruction time of $\rho$, $\rho v$ and $E$ and the regularization. As

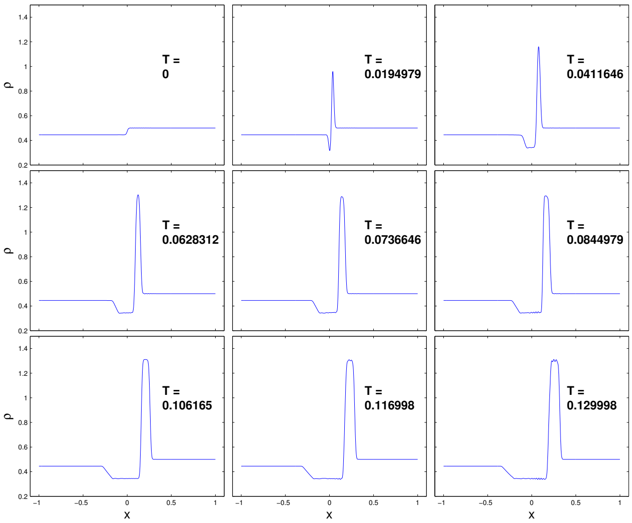

FIGURE 13. POD density solutions with the Gegenbauer reconstruction and the reinitialization for the Lax problem, Eq. (5.2).

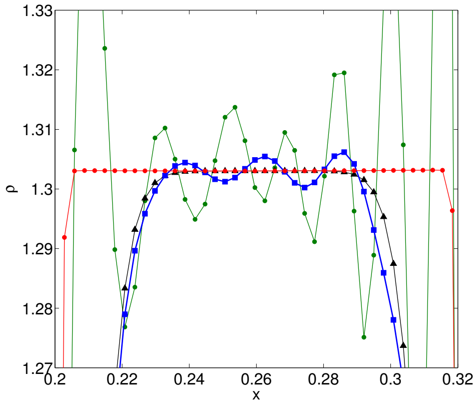

FIGURE 14. (Color online) Various solutions at $t = 0.13$. Sample solution (-●), Gegenbauer reconstruction (-●), smoothed Gegenbauer reconstruction with filtering (-▲) and POD solution *with* the post-processing and re-initialization (-■).

shown in the table, the calculation of the POD solutions is completed much more quickly than the full solutions while the POD solutions are less oscillatory.

## 6. Conclusion

The POD method efficiently provides a fast algorithm to calculate unknown solutions based on the given sample solutions. The reduction of the computational complexity is mainly based on the smoothness of the samples. However, if the solutions sought are discontinuous and the discontinuity is parameterized e.g. with time, we do not expect the same degree of reduction. To take advantage of the POD method for discontinuous problems, we proposed to reconstruct and regularize the sample solutions based on the Gegenbauer reconstruction method and re-initialize the reconstructions during the POD calculations. With the Gegenbauer reconstruction, we were able to find accurate reconstructions to the exact solutions. These solutions are used sporadically during the POD calculations through re-initialization. Our numerical results show that the proposed methods yield reasonable POD solutions with the CPU time and Gibbs oscillations both reduced.

Here we remark that our paper more concerns the process in how to find accurate POD approximations in the reduced space with a small number of POD bases for the given samples rather than in how to find accurate sample solutions. Thus the proposed method will be useful for the case that the samples are given to the users and the users are supposed to find POD solutions based on the given samples.

We also remark that the quality of the POD solutions presented in this paper is affected by the performance of the individual methods used in the reconstruction and regularization steps. Thus one can achieve better POD performances if more advanced methods are used in such steps. In this paper, however, we did not attempt to further develop these reconstruction and regularization methods. Instead we attempted to show how the POD solutions can be improved for the discontinuous solutions with the proposed method as a proof-of-concept, which was demonstrated by the numerical experiments. We will consider using more sophisticated methods in the reconstruction and regularization steps in our future research.

For the 2D problem, one possible way is to apply the Gegenbauer post-processing using the slice-by-slide approach in the Cartesian grid such as those found in [25, 26]. Then the POD approximation can be applied as a tensor product. We will extend the proposed method to the 2D problem in our future research.

## Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT and Future Planning(2017R1E1A1A03070059).

## References

[1] P. Benner, M. Ohlberger, A. T. Patera, G. Rozza, D. C. Sorensen, K. Urban (Eds.), Special Issue: Model Order Reduction of Parameterized Systems, Adv. Comput. Math. 41, 2015.

[2] G. Berkooz, P. Holmes, J. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, Ann. Rev. Fluid Mech. 25 (1993) 539–575.

[3] J. Burkardt, M. Gunzburger, H.-C. Lee, Centroidal Voronoi tessellation-based reduced-order modeling of complex systems, SIAM J. Sci. Comput. 28(2) (2006) 459–484.

[4] J. Burkardt, M. Gunzburger, H.-C. Lee, POD and CVT-based reduced-order modeling of Navier-Stokes flows, Comput. Meth. Appl. Mech. Eng. 196 (2006) 337–355.

[5] Q. Du, V. Faber, M. Gunzburger, Centroidal Voronoi tessellations: applications and algorithms, SIAM Rev. 41(4) (1999) 637–676.

[6] M. Gunzburger, J. Peterson, J. Shadid, Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data, Comput. Meth. Appl. Mech. Eng. 196 (2007) 1030–1047.

[7] K. Kunisch, S. Volkwein, Galerkin proper orthogonal decomposition methods for parabolic equations, Numer. Math. 90 (2001) 117–148.

[8] Y. C. Liang, H. P. Lee, S. P. Lim, W. Z. Lin, K. H. Lee, C. G. Wu, Proper orthogonal decomposition and its application-part1: Theory, J. Sound and Vibration 252(3) (2002) 527–544.

[9] H.-C. Lee, S. Lee, G. Piao, Reduced-order modeling of Burgers equations based on centroidal Voronoi tessellation, Int. J. Numer. Anal. Model 4(3-4) (2007) 559–583.

[10] S. Ravindran, Reduced-order adaptive controllers for fluid flows using POD, SIAM J. Sci. Comput. 15 (2000) 457–478.

[11] F. Fang, C. C. Pain, I. M. Navon, A. H. Elsheikh, J. Du, D. Xiao, Non-linear Petrov-Galerkin methods for reduced order hyperbolic equations and discontinuous finite element methods, J. Comput. Phy. 234 (2013) 540–559.

[12] D. Gottlieb, C-W.Shu, On the Gibbs phenomenon and its resolution, SIAM Rev. 39 (1997) 644–668.

[13] D. Gottlieb, C-W. Shu, A.Solomonoff, H. Vandeven, On the Gibbs phenomenon 1: recovering exponential accuracy from the Fourier partial sum of a nonperiodic analytic function, J. Comput. Appl. Math. 43 (1992) 81–92.

[14] D. Gottlieb, C.-W. Shu, On the Gibbs phenomenon V: recovering exponential accuracy from collocation point values of a piecewise analytic function, Numerische Mathematik 71 (1995) 511–526.

[15] D. Gottlieb, C.-W. Shu, On the Gibbs phenomenon IV: recovering exponential accuracy in a sub-interval from a Gegenbauer partial sum of a piecewise analytic function, Math. Comput. 64 (1995) 1081–1095.

[16] D. Gottlieb, C.-W. Shu, On the Gibbs phenomenon III: recovering exponential accuracy in a subinterval from a spectral partial sum of a piecewise analytic function, SIAM J. Numer. Anal. 33 (1996) 280–290.

[17] B.-C. Shin, J.-H. Jung, Spectral collocation and radial basis function methods for one-dimensional interface problems, Appl. Numer. Math. 61(8) (2011) 911–928.

[18] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (1988) 439–471.

[19] J. Hesthaven, S. Gottlieb, D. Gottlieb, Spectral Methods for Time-Dependent Problems, Cambridge University Press, Cambridge, 2007.

[20] M. Abramowitz, I. Stegun (Eds.), Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, tenth ed., New York, 1964.

[21] W.-S. Don, D. Gottlieb, J.-H. Jung, Multi-domain Spectral Method for Supersonic Reactive Flows, J. Comput. Phys. 192(1) (2003) 325–354.

[22] E. Tadmor, J. Tanner, Adaptive filters for piecewise smooth spectral data, IMA J. Num. Anal. 25 (2005) 635–647.

[23] R. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge UP, Cambridge, 2002.

[24] J.-K. Seo, B.-C. Shin, Numerical solutions of Burgers equation by reduced-order modeling based on pseudo-spectral collocation method, J. KSIAM 19(2) (2015) 123–135.

[25] R. Archibald, K.W. Chen, A. Gelb, R. Renaut, Improving tissue segmentation of human brain MRI through preprocessing by the Gegenbauer reconstruction method, Neuroimage, 20 (2003) 489–502.

[26] A. Gelb,, D. Gottlieb, The resolution of the Gibbs phenomenon for “spliced” functions in one and two dimensions, Comput. Math. Appl. 33 (1997) 35–58.