INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
*Int. J. Numer. Meth. Fluids* 2016; **00**:1–32
Published online in Wiley Online Library (www.onlinelibrary.wiley.com). DOI: 10.1002/fld.4249

# 2D Burgers Equation with Large Reynolds Number Using POD/DEIM and Calibration

Yuepeng Wang[a],I.Michael Navon [b*], Xinyue Wang [c], Yue Cheng [a]

[a] *School of Mathematics and Statistics Nanjing University of Information Science and Technology (NUIST);Nanjing;China; 210044 eduwyp@163:com*
[b] *Department of Scientific Computing Florida State University; Tallahassee; FL 32306; U:S:A *inavon@fsu:edu*
[c] *School of Atmospheric Science Nanjing University of Information Science and Technology (NUIST);Nanjing;China; 210044*

SUMMARY

Model order reduction (MOR) of the 2D Burgers equation is investigated. The mathematical formulation of POD/DEIM reduced order model (ROM) is derived based on the Galerkin projection and discrete empirical interpolation method (DEIM) from the existing high fidelity implicit finite difference full model. For validation we numerically compared the POD ROM, POD/DEIM and the full model in two cases of $Re = 100$ and $Re = 1000$, respectively. We found that the POD/DEIM ROM leads to a speed-up of CPU time by a factor of $O(10)$. The computational stability of POD/DEIM ROM is maintained by means of a careful selection of POD modes and the DEIM interpolation points. The solution of POD/DEIM in the case of $Re = 1000$ has an accuracy with error $O(10^{-3})$ versus $O(10^{-4})$ in the case of $Re = 100$ when compared to the high fidelity model. For this turbulent flow, a closure model consisting of a Tikhonov regularization is carried out in order to recover the missing information and is developed to account for the small scale dissipation effect of the truncated POD modes. It is shown that the computational results of this calibrated reduced order model (ROM) exhibit considerable agreement with the high-fidelity model, which implies the efficiency of the closure model used. Copyright © 2016 John Wiley & Sons, Ltd.

Received . . .

KEY WORDS: 2D Burgers equation; POD/DEIM reduced order Model; Tikhonov regularization; Calibration

*Correspondence to: I.Michael Navon, Department of Scientific Computing Florida State University; Tallahassee; FL 32306; U:S:A *inavon@fsu:edu

## 1. INTRODUCTION

The two-dimensional Burgers equation is a fundamental mathematical model from fluid mechanics which has the same convective and diffusion terms as the Navier-Stokes equation, and is widely used in various areas as a simple model for understanding of various physical flows and problems, such as modeling of dynamics, the phenomena of turbulence, and flow through a shock wave traveling in a viscous fluid and traffic flow [1,2,3]. Numerical solution of Burgers equation is a logical first step towards developing methods for the computation of complex flows. Burgers equation is also a useful tool for examining the robustness of numerical discretization schemes [4]. It has become customary to test new numerical approaches in computational fluid dynamics by implementing novel numerical approaches to the Burgers equation. So far, many numerical solution approaches to 2D Burgers equation have been developed by scientists and engineers, such as [3,5,6]. In addition, its analytical solution was also explored [7] using the Cole-Hopf transformation.

However, in the conventional numerical approach the computational cost of these calculations becomes higher as the Reynolds number ($Re$) increases, particularly when a smooth wave becomes a single-shock wave. Also the computational cost increases as we refine the resolution of the spatial discretization mesh. This will become more accentuated when attempting to solve a control or PDE optimization problem in a wide class of engineering applications by DNS (direct numerical simulation) of the full 2D Burgers equation model due to the requirement of quick and repeated numerical simulations [8,9,10,11], which often pose important mathematical and computational challenges in both CPU and memory requirements.

To overcome the difficulty encountered during simulating, controlling, and optimizing such systems, reduced order modelling offers a remedy as a powerful and feasible approach enabling a representation of the dynamics of high-dimensional systems with a smaller number of degrees of freedom. Developing low-dimensional models for partial differential equations (PDEs) is one of the active research topics today [12,13].

An effective technique of low-order modelling, Proper Orthogonal Decomposition (POD) is attractive to apply [14,15], and has become one of the most important reduced order model (ROM) methods combined with Galerkin projection, due to its ability to rebuild numerical signals, with a high order of precision gained with only few POD-modes. Additionally, POD is also known under other names such as Karhunen-Loève expansions [16,17], principal component analysis [18], empirical orthogonal functions [19], and the Hotelling transform [20]. It is one of the most prevalent model reduction methods for nonlinear problems [12]. Data analysis using POD is conducted in order to extract basis functions from experimental data or detailed simulations of high-dimensional systems for subsequent use in Galerkin projections that yield low-dimensional dynamical models. Nevertheless, beside its efficiency, the POD Galerkin approach still exhibits some disadvantages.

In both her Master and PhD thesis Chaturantabut [21,22] was the first to propose a new method for handling the quadratic nonlinear terms -referred to as the precomputing-POD technique. It consists of employing the simple structure of the quadratic nonlinearities to remove the dependence on the dimension of the original discretized system by manipulating the order of computing and separating the spatial variables from the time variable.

For particular model problems such as 1-D Burgers equation which have quadratic nonlinear terms, she found that the precomputing technique works better than empirical interpolation method (EIM) in terms of accuracy and complexity, as evidenced by the numerical results.

The method efficiency is restricted only to quadratic nonlinear terms. For higher order nonlinearities POD/DEIM is the method of choice.

Stefanescu et al. (2014) [12] used a similar idea for the quadratic nonlinearity in the 2D shallow water equations model referring to it as the tensorial POD. They have shown that precomputing (tensorial POD) is slower in the general quadratic case than POD/DEIM. See Table I in Stefanescu et al. (2014). There is no contradiction, since results of POD/DEIM vs. precomputed POD will depend on numbers of spatial points $n$, POD modes $k$ and DEIM points $m$ and on the simplicity of FEM (finite element method) discretization. Hence the interest in POD/DEIM.

We intend to consider the precomputed-POD approach in an upcoming follow-up reserch paper.

For models with nonlinear terms, the nonlinear reduced terms have to be evaluated on the original state space making the simulation of the reduced order system computationally expensive. An interpolation technique known as discrete empirical interpolation method (DEIM) [23,24,25] avoids this problem by using an interpolation algorithm [26,27,28]. The DEIM approach employs a small selected set of spatial grid points to avoid evaluation of the expensive inner products at every time step required to evaluate the nonlinear term. It focuses on approximating each nonlinear function, so that a certain coefficient matrix can be precomputed. As a result, the complexity in evaluating the nonlinear term is proportional only to a small number of selected spatial indices, thus achieving a considerable reduction in computational complexity.

For turbulent flows, the POD-ROM-Galerkin yields inaccurate results. As carefully explained in the thesis of Z. Wang. [29] for realistic turbulent flows, the high index POD modes that are not included in the POD-ROM-Galerkin do have a significant effect on the dynamics of the POD-ROM-Galerkin.

Several numerical stabilization strategies have been used to address this issue building on the analogy with large eddy simulation (LES).

To improve the accuracy and stability of such POD-based reduced order models, various calibration methods were proposed [30,31,32], such as $H^1$ Sobolev norm [13,33,34], eddy viscosities [35], least squares or adjoint method [31,36,37,38,39]. Calibration of ROM can enhance its performance at low computational cost, but still remains a challenging task.

When the one-dimensional Burgers equation is considered, the corresponding POD/DEIM reduced order model has been developed, and the control problem involved was also discussed. It has been shown in [40,41,42,43] that good results are achieved. Yet to the best of our knowledge, there are very few results reporting the POD/DEIM reduced order modeling issues for 2D Burgers equation, particularly in the case where the Reynolds number becomes large, and it is an open question if the use of a proper number of DEIM points is really beneficial for POD/DEIM CPU cost. The present work can be viewed as a new step towards the goal of modeling and control of more complicated PDE systems.

The main focus of the paper is the size of the reduced order system and the quality of approximation compared to the full-order system, as well as computational efficiency gained by using POD/DEIM in nonlinear model reduction applications to 2D Burgers equation. Even though these are fundamental questions related to POD, we believe that they have not been addressed in the literature related to 2D Burgers equation with large Reynolds number.

The main contributions of the research presented in this paper consist first in development of the mathematical formulation of POD/DEIM ROM based on existing fully implicit finite difference time discretization scheme. Second, derivation of a calibrated low order ODEs system from the precomputed POD coefficients using the Tikhonov regularization method as a closure model when dealing with large Reynolds number [44]. Due to the demanding memory requirements, we had to restrict study of the turbulent case only to a Reynolds number equal to 1000. It is expected that a calibrated reduced order ODEs model of the two-dimensional Burgers equation will result in a decrease in both memory storage and CPU time requirements.

The rest of this paper is organized as follows. Section 2 provides a description of two-dimensional Burgers equation and its discretization by fully implicit finite difference scheme. Section 3 is devoted to the construction process of the POD/DEIM reduced order model of 2D Burgers equation, and the discrete empirical interpolation (DEIM) approach is applied to deal with the nonlinear terms arising from the nonlinear advection terms in the 2-D Burgers model. In section 4, the validation of POD/DEIM ROM is carried out , and numerical experiments are performed to verify the performance of the POD/DEIM ROM in the case of $Re = 100$ and $Re = 1000$, respectively. Special care needs to be taken to find the reduced basis and to choose DEIM interpolation points. In section 5, a calibrated ROM based on Tikhonov regularization method serving as a closure model is developed for turbulent flow ($Re = 1000$). Concluding remarks are provided at the end of the paper, and an Appendix providing a simple introduction to the discrete Picard condition is added for a detailed presentation of the Tikhonov regularization closure model. In addition, both the Newton iterative method necessary to integrate in time the discrete 2D Burgers equation and derivation of the POD reduced order model (ROM) of 2D Burgers equation are also included in the Appendix.

## 2. FULL-ORDER MODEL OF 2D BURGERS EQUATION

Newton iteration solution method is described in section 2.1 and 2.2. We are now considering a two-dimensional nonlinear viscous Burgers equation assuming the form (called **Full Model**):

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=\frac{1}{Re}(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}), \tag{2.0.1$a$}$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=\frac{1}{Re}(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}), \tag{2.0.1$b$}$$

$$(x,y)\in\Omega=(a,b)\times(c,d),\ t\in(0,T)$$

subject to the boundary conditions

$$u(a,y,t)=g_{u1}(y,t);\quad u(b,y,t)=g_{u2}(y,t);$$

$$u(x,c,t)=g_{u3}(x,t);\ \ u(x,d,t)=g_{u4}(x,t); \tag{2.0.2$a$}$$

$$v(a,y,t)=g_{v1}(y,t);\quad v(b,y,t)=g_{v2}(y,t);$$

$$v(x,c,t)=g_{v3}(x,t);\ \ v(x,d,t)=g_{v4}(x,t); \tag{2.0.2$b$}$$

and the initial conditions

$$u(x,y,t)|_{t=0}=\varphi(x,y), \tag{2.0.2$c$}$$

$$v(x,y,t)|_{t=0}=\psi(x,y),\quad (x,y)\in\Omega \tag{2.0.2$d$}$$

where the $u(x,y,t)$ and $v(x,y,t)$ represent the velocity components. When $2D$ spatial computational domain $\Omega$ is divided uniformly into $n_x-1$ and $n_y-1$ subintervals in $x$ and $y$ direction, respectively. It is assumed that the discrete functions are defined on an $n_x\times n_y$ -grids in space domain $\Omega=[a,b]\times[c,d]$. The following notation will be used: $x_i=i\triangle x,\ y_j=j\triangle y$, $u_{ij}=u(x_i,y_j,t), v_{ij}=v(x_i,y_j,t)$. Here $\triangle x=\frac{b-a}{n_x-1}, \triangle y=\frac{d-c}{n_y-1}$. Then the centered-difference scheme corresponding to the first- or second-order derivatives in space will result in the following form of $2D$ Burgers equation [3]:

$$\begin{aligned}&\frac{du_{i,j}}{dt}+\frac{u_{i+1,j}-u_{i-1,j}}{2\triangle x}u_{i,j}+\frac{u_{i,j+1}-u_{i,j-1}}{2\triangle y}v_{i,j}=\\&\quad\frac{1}{Re\triangle x^2}(u_{i+1,j}-2u_{i,j}+u_{i-1,j})+\frac{1}{Re\triangle y^2}(u_{i,j+1}-2u_{i,j}+u_{i,j-1})\end{aligned} \tag{2.0.3a}$$

$$\begin{aligned}&\frac{dv_{i,j}}{dt}+\frac{v_{i+1,j}-v_{i-1,j}}{2\triangle x}u_{i,j}+\frac{u_{i,j+1}-v_{i,j-1}}{2\triangle y}v_{i,j}=\\&\quad\frac{1}{Re\triangle x^2}(v_{i+1,j}-2v_{i,j}+v_{i-1,j})+\frac{1}{Re\triangle y^2}(v_{i,j+1}-2v_{i,j}+v_{i,j-1})\end{aligned} \tag{2.0.3b}$$

which can be cast in matrix form as follows:

$$\begin{aligned}\frac{dU}{dt} + f_1(U,V) - \frac{1}{2\triangle x}B_{ul}U - \frac{1}{2\triangle y}B_{ub}V = \\ \frac{1}{Re\triangle x^2}(D_1U + b_{1u}) + \frac{1}{Re\triangle y^2}(D_2U + b_{2u})\end{aligned} \tag{2.0.4a}$$

$$\begin{aligned}\frac{dV}{dt} + f_2(U,V) - \frac{1}{2\triangle x}B_{vl}U - \frac{1}{2\triangle y}B_{vb}V = \\ \frac{1}{Re\triangle x^2}(D_1V + b_{1v}) + \frac{1}{Re\triangle y^2}(D_2V + b_{2v})\end{aligned} \tag{2.0.4b}$$

where

$$U = (u_{1,1}, u_{2,1}, \cdots, u_{n,1}, u_{1,2}, \cdots, u_{n,2}, \cdots, u_{1,n}, \cdots, u_{n,n})^T$$
$$V = (v_{1,1}, v_{2,1}, \cdots, v_{n,1}, v_{1,2}, \cdots, v_{n,2}, \cdots, v_{1,n}, \cdots, v_{n,n})^T$$

hereafter, the superscript '$T$' stands for transpose. Let $n_{xy} = (n_x - 2) \times (n_y - 2)$, two maps $f_1, f_2 : \mathbb{R}^{n_{xy}} \times \mathbb{R}^{n_{xy}} \to \mathbb{R}^{n_{xy}}$ are then defined as follows:

$$f_1(U,V) = \frac{1}{2\triangle x}MU.*U + \frac{1}{2\triangle y}NU.*V, \tag{2.0.5$a$}$$

$$f_2(U,V) = \frac{1}{2\triangle x}MV.*U + \frac{1}{2\triangle y}NV.*V, \tag{2.0.5$b$}$$

and $M = \begin{pmatrix} M_1 & & \\ & \ddots & \\ & & M_1 \end{pmatrix}_{(n_y-2)\times(n_y-2)}$, $M_1 = \begin{pmatrix} 0 & 1 & \\ -1 & \ddots & 1 \\ & -1 & 0 \end{pmatrix}_{(n_x-2)\times(n_x-2)}$

with '$.*$' denoting componentwise multiplication as used in Matlab. The $B_{ul}$, $B_{ub}$, $b_{1u}$ and $b_{2u}$ are related to the boundary conditions, which are denoted by

$$B_{ul} = \text{diag}(\text{kron}(u(1, 2 : n_y - 1), [1, 0, 0, \cdots, 0]_{1\times(n_x-2)})),$$
$$B_{ub} = \text{diag}(\text{kron}([1, 0, 0, \cdots, 0]_{1\times(n_y-2)}, u(2 : n_x - 1, 1)),$$
$$b_{1u} = (u_{1,2}, 0, \cdots, 0, u_{5,2}, u_{1,3}, 0, \cdots, u_{5,4})^T,$$
$$b_{2u} = [u(2 : n_x - 1, 1)^T, 0, \cdots, 0, u(2 : n_x - 1, 5)^T]^T,$$

$$D_1 = \begin{pmatrix} D_{11} & & \\ & \ddots & \\ & & D_{11} \end{pmatrix}_{(n_y-2)\times(n_y-2)}, \quad D_2 = \begin{pmatrix} -2E_{(n_x-2)\times(nx-2)} & E & \\ E & \ddots & E \\ & E & -2E \end{pmatrix}_{(n_y-2)\times(n_y-2)}$$

$$D_{11} = \begin{pmatrix} -2 & 1 & \\ 1 & \ddots & 1 \\ & 1 & \end{pmatrix}_{(n_x-2)\times(n_x-2)}, \quad N = \begin{pmatrix} E & & \\ -E & \ddots & \\ & -E & E \end{pmatrix}$$

where $E$ is $(n_y - 2)$-by-$(n_y - 2)$ the identity matrix, and 'kron' is a Matlab function which means that $K = \text{kron}(A, B)$ returns the Kronecker tensor product of $A$ and $B$. Note that $B_{vl}$, $B_{vb}$, $b_{1v}$ and $b_{2v}$ are not mentioned here due to their similarity to $B_{ul}$, $B_{ub}$, $b_{1u}$ and $b_{2u}$. The numerical solution procedure, can be accomplished by the Newton iteration solution method[45]. For model reduction of dynamical systems, POD may be used on data points obtained from system trajectories obtained via experiments, numerical simulations, or analytical derivations. For more details, please see [46].

The POD method has been widely discussed in literature during the last decades, and is still a very active field of research [12,13,14,15,23,24]. The main advantage of POD lies in the fact that it requires only standard matrix computation. In combination with Galerkin projection, it provides a powerful tool to obtain low-dimensional models of high-dimensional systems. It is well known that in a finite-dimensional space or in Euclidean space, the POD model order reduction method can be accomplished by SVD or EVD technique.

## 2.1. *Snapshot Collection and POD Basis*

As far as the calculation of the POD modes is concerned, let us first give the model variables solution $\Psi = [\Psi_k(\mathbf{x}, t_k)] \in \mathbb{R}^{n_{xy} \times n_k}$ (e.g. either one of the velocity components $u, v$ ) to form a set of snapshots sampled at the defined checkpoints during the numerical simulation at equally distributed time instances $t_{n_1}, t_{n_2}, \cdots, t_{n_k}$, where $n_k$ is the number of snapshots. Due to possible linear dependence, the snapshots themselves are not suitable for use as a basis. Singular value decomposition (SVD) for $\Psi \in \mathbb{R}^{n_{xy} \times n_k}$, eigenvalue decomposition for $\Psi\Psi^T \in \mathbb{R}^{n_{xy} \times n_{xy}}$ or eigenvalue decomposition for $\Psi^T\Psi \in \mathbb{R}^{n_k \times n_k}$ are used to derive the so-called POD basis. In this work, based on the consideration that $u$ and $v$ are governed by distinct physics, i.e., different partial differentiation equations (2.0.1a,b), respectively, the corresponding POD basis vectors for them are built from the snapshots of the solution for each variable separately. This technique has already been used in existing literatures [47,48]. Here we present only the construction of the POD basis corresponding to $u$ as a similar procedure is used to determine the POD basis for $v$. When taking into account that $n_k \ll n_{xy}$, we choose to construct the POD basis matrix $\Phi_u = [\Phi_u^i, i = 1, 2, \cdots, m_1] \in \mathbb{R}^{n_{xy} \times m_1}$ by solving the eigenvalue problem

$$\Psi^T\Psi\hat{u}_i = \lambda_i\hat{u}_i, i = 1, 2, \cdots, n_k \tag{2.1.1}$$

and we can choose an orthogonal basis of eigenvectors$\{\hat{u}_1, \hat{u}_2, \cdots, \hat{u}_{m_1}\}$ corresponding to the $m_1$ largest eigenvalue, then POD modes of velocity $u$ are given by $\Phi_u^i = \frac{1}{\sqrt{\lambda_i}}\Psi\hat{u}_i$. Similarly, let $\Phi_v \in \mathbb{R}^{n_{xy} \times m_2}$ be the POD basis matrix of velocity component $v$. Although these POD modes provide an optimal representation of the snapshot matrix, some information is inevitably lost. This loss of information can be qualified by the following ratio,

$$I(m) = \frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{n_k} \lambda_i} \tag{2.1.2}$$

through which one defines a relative information content to choose a low-dimensional basis of size $\tilde{M} \ll n_k$ by neglecting modes corresponding to the small eigenvalues. We can choose $\tilde{M}$ such that $\tilde{M} = \arg\min\{I(m) : I(m) \geq \gamma^0\}$, where $0 \leq \gamma^0 \leq 1$ is the percentage of total information captured by the reduced space $span\{\Phi_u\}$. The tolerance $\gamma^0$ must be chosen to be near the unity in order to capture most of the energy of the snapshot basis.

## 2.2. *Acceleration Based on the Application of the DEIM Approach*

To get a reduced order POD model, we use the POD bases obtained above to approximate $U$,$V$ as following way:

$$U(t_n) \approx \Phi_u\alpha(t_n), V(t_n) \approx \Phi_v\beta(t_n), \tag{2.2.1}$$

where $\alpha(t_n) \in \mathbb{R}^{m_1}, \beta(t_n) \in \mathbb{R}^{m_2}$. Plugging the equations (2.2.1) in the equations (2.0.4) and multiplying the equation (2.0.4a) and (2.0.4b) by $\Phi_u^T$ and $\Phi_v^T$, respectively, we can obtain the POD reduced order model (**POD ROM**)with a significantly smaller number of unknowns, $m_1 \ll n_{xy}$, $m_2 \ll n_{xy}$. For additional details of POD ROM and the gradients of nonlinear functions with respect to $\alpha, \beta$ used for the Newton iteration method, please see appendix B.

However, it is necessary to note that computing matrix-vector product to project on POD basis in the nonlinear terms below

$$f_1^{POD}(\alpha, \beta) = \underbrace{\Phi_u^T}_{m_1 \times n_{xy}} \underbrace{f_1(\Phi_u\alpha, \Phi_v\beta)}_{n_{xy} \times 1}, \tag{2.2.2a}$$

$$f_2^{POD}(\alpha,\beta) = \underbrace{\Phi_v^T}_{m_2\times n_{xy}} \underbrace{f_2(\Phi_u\alpha,\Phi_v\beta)}_{n_{xy}\times 1}, \tag{2.2.2$b$}$$

still has a computational complexity depending on the number of unknowns of the **Full Model** which may cause the POD ROM to be inefficient.

The subsequent development focuses on the application of Discrete Empirical Interpolation Method (**DEIM**) to the nonlinear functions (2.2.2a,b).

DEIM is a discrete variation of the Empirical Interpolation method (EIM) proposed by Barrault et al. (2004) [26] , the effect of which is to decrease the computational cost and cause the nonlinear terms to be independent of the high fidelity model dimension.

Let us now describe the process of application of **DEIM**. Similar to the building of POD basis $\Phi_u$ and $\Phi_v$, the snapshots of $f_1(U,V)$ and $f_2(U,V)$ are first collected at time instances $t_k \in \{t_1,\cdots,t_l\} \subset [0,T]$, then the DEIM approximates the projected functions (2.2.2a,b)such that

$$f_1^{POD} \simeq \hat{f}_1^{DEIM} = \underbrace{\Phi_u^T\Xi_{f_1}(P_1^T\Xi_{f_1})^{-1}}_{\Xi^{f_1}\in\mathbb{R}^{m_1\times\tau_1}} \underbrace{P_1^T f_1(\Phi_u\alpha,\Phi_v\beta)}_{\tilde{f}_1(\alpha,\beta)\in\mathbb{R}^{\tau_1\times 1}}, \tag{2.2.3$a$}$$

$$f_2^{POD} \simeq \hat{f}_2^{DEIM} = \underbrace{\Phi_v^T\Xi_{f_2}(P_2^T\Xi_{f_2})^{-1}}_{\Xi^{f_2}\in\mathbb{R}^{m_2\times\tau_2}} \underbrace{P_2^T f_2(\Phi_u\alpha,\Phi_v\beta)}_{\tilde{f}_2(\alpha,\beta)\in\mathbb{R}^{\tau_2\times 1}}, \tag{2.2.3$b$}$$

where $\Xi_{f_i} \in \mathbb{R}^{n_{xy}\times\tau_i}, i = 1,2$ contains the first $\tau_i$ POD basis of the space spanned by the snapshots $\{f_i(U(t_k),V(t_k)), i = 1,2; k = 1,2,\cdots,n_k\}$ associated with the largest eigenvalues (or singular values). And the selection matrix $P_i = [e_{\rho_1},\cdots,e_{\rho_{\tau_i}}] \in \mathbb{R}^{n_{xy}\times\tau_i}, i = 1,2$, selects the rows of $f_i$ corresponding to the DEIM indices $\rho_1,\cdots,\rho_{\tau_i}$ which are obtained by the greedy algorithm, see [25,26] for details. By using the fact that $\tilde{f}_1, \tilde{f}_1$ are componentwise, we can calculate them as

$$\tilde{f}_1(\alpha,\beta) := \frac{1}{2\triangle x}\underbrace{P_1^T M\Phi_u}_{\hat{\Xi}_1}\alpha.*(\underbrace{P_1^T\Phi_u}_{\hat{\Xi}_2}\alpha) + \frac{1}{2\triangle y}\underbrace{P_1^T N\Phi_u}_{\hat{\Xi}_3}\alpha.*(\underbrace{P_1^T\Phi_v}_{\hat{\Xi}_4}\beta), \tag{2.2.4$a$}$$

$$\tilde{f}_2(\alpha,\beta) := \frac{1}{2\triangle x}\underbrace{P_2^T M\Phi_v}_{\hat{\Xi}_5}\beta.*(\underbrace{P_2^T\Phi_u}_{\hat{\Xi}_6}\alpha) + \frac{1}{2\triangle y}\underbrace{P_2^T N\Phi_v}_{\hat{\Xi}_7}\beta.*(\underbrace{P_2^T\Phi_v}_{\hat{\Xi}_8}\beta), \tag{2.2.4$b$}$$

Finally, the **POD/DEIM ROM** is of the form:

$$\begin{aligned}
&\frac{d\alpha}{dt} + \hat{f}_1^{DEIM} - \frac{1}{2\triangle x}B_{ul}^{POD}\alpha - \frac{1}{2\triangle y}B_{ub}^{POD}\beta = \\
&\qquad\qquad \frac{1}{Re\triangle x^2}(D_{11}^{POD}\alpha + b_{1u}^{POD}) + \frac{1}{Re\triangle y^2}(D_{12}^{POD}\alpha + b_{2u}^{POD}) \qquad (2.2.5\text{a})\\
&\frac{d\beta}{dt} + \hat{f}_2^{DEIM} - \frac{1}{2\triangle x}B_{vl}^{POD}\alpha - \frac{1}{2\triangle y}B_{vb}^{POD}\beta = \\
&\qquad\qquad \frac{1}{Re\triangle x^2}(D_{21}^{POD}\beta + b_{1v}^{POD}) + \frac{1}{Re\triangle y^2}(D_{22}^{POD}\beta + b_{2v}^{POD}) \qquad (2.2.5\text{b})
\end{aligned}$$

For details of the relevant coefficient matrix, please refer to appendix C. It is observed from the relations (2.2.3a,b) and (2.2.4a,b) that the coefficient matrix $\Xi^{f_1}$, $\Xi^{f_2}$, $\hat{\Xi}_i$, $i = 1,2,\cdots,8$, does not depend on time and can thus be precomputed, saved and reused in all time steps. As for the resulting approximation of $\hat{f}_1^{DEIM}(\alpha,\beta)$ and $\hat{f}_2^{DEIM}(\alpha,\beta)$, no dependence on the dimension $n_{xy}$ of the full order system exists anymore. Thus, a substantial reduction in computational cost can be expected due to their dependence only on dimensions $m_i$ of POD and $\tau_i$ of DEIM, $i = 1,2$. This efficiency will reflect the speed-up of simulation time.

## 3. THE VALIDATION OF POD/DEIM ROM

In this section we will provide numerical experiments aiming at illustrating the accuracy and efficiency of the POD/DEIM. For the **Full Model, POD ROM** and **POD/DEIM ROM**, as ODEs, the implicit Euler scheme was used. The resulting nonlinear algebraic system of equations is solved by Newton-iteration method. The computational domain is taken as $\Omega = [0, 1] \times [0, 1]$, $T$ is set as $T = 1$. The number of the spatial grid points is taken to be $n_x \times n_y = 60 \times 60$, with $n_t = 250$ in the case of Reynolds number $= 100$, and $n_x \times n_y = 200 \times 200$, with $n_t = 1000$ in the case of Reynolds number $= 1000$. The initial conditions (**ICs**) and boundary conditions (**BCs**) related to $u(x, y, t)$ and $v(x, y, t)$ are derived directly from the exact traveling wave solution of 2D Burgers equation [7].

$$u(x,y,t) = \frac{3}{4} - \frac{1}{4[1 + exp((-4x + 4y - t)Re/32)]} \tag{3.1$a$}$$

$$v(x,y,t) = \frac{3}{4} + \frac{1}{4[1 + exp((-4x + 4y - t)Re/32)]} \tag{3.1$b$}$$

To assess how well our reduced order model (**ROM**)approximates the full model, we use the root mean square error (**RMSE**) and the correlation coefficients **Corr** to measure the difference between POD or POD/DEIM ROM velocity solution and the Full Model solutions at the time level $n$

$$\mathbf{RMSE^n} = \sqrt{\frac{\sum_{i=1}^{N}(U_i^n - U_{0,i}^n)^2}{N}} \tag{3.2}$$

$$\mathbf{Corr(U, U_0)^n} = \frac{E(U^n - \mu_U)(U_0^n - \mu_{U_0^n})}{\sigma_{U^n}\sigma_{U_0^n}} \tag{3.3}$$

where $\mu_U$, and $\mu_{U_0}$ are the given expected value and the standard deviations $\sigma_U$, and $\sigma_{U_0}$. In addition, the average relative error (**E**)[12] are also provided in the analysis. Its computational formula is defined as

$$\mathbf{E_u} = \frac{1}{nt}\sum_{i=1}^{nt}\frac{\|U^{Full}(:,i) - U^{POD/DEIM}(:,i)\|_2}{\|U^{Full}(:,i)\|_2} \tag{3.4}$$

### 3.1. *Case of* $Re = 100$

The POD basis are constructed using 125 snapshots obtained from the numerical solution of the full-order fully implicit finite difference scheme of 2D Burgers equation at equally spaced time steps for time interval $[0, T]$.

Figure 1 shows the decay around the eigenvalues of the snapshot solutions for $u$,$v$ and nonlinear snapshots of $f_1, f_2$. The dimension of the POD basis for $u$,$v$ was taken to be 5, respectively, capturing more than 99.8% ($I_u(m) = I_v(m) \geqslant 0.998$) of the system energy. The DEIM approach is used to improve the efficiency of the POD ROM, and achieves a complexity reduction of the nonlinear terms due to the first 50 spacial interpolation points selected from the DEIM approach using the POD bases of $f_1$ and $f_2$ as inputs. For the distribution, see Figure 2. The selection of the 50 DEIM interpolation points, is only based on results shown in Table 1.

From Table 2, it is observed that the POD ROM fails to decrease computational complexity since the nonlinear terms depend on the size $n_{xy}$ of the original high fidelity model. While the POD/DEIM 50-ROM is shown to be very effective in overcoming the deficiencies of POD, being faster than the POD ROM and high fidelity model by a factor of 10 due to the DEIM computation of nonlinear functions $f_1, f_2$ using the selected 50 DEIM interpolation points ( see figure 2).

We see from figure 3 that the solutions $u$ of POD/DEIM50 ROM at time steps $nt = 10, 100, 200$ are very close to those of Full Model. And a small RMSE at different times is also observed in Figure 4. In addition, the correlation of $u$ between POD/DEIM50 ROM and the Full Model as displayed in Figure 5, is very close to 1.

Next we carry out a second experiment to test the performance of POD/DEIM ROM in the case of $Re = 1000$,(turbulent flow).

### 3.2. *Case of $Re = 1000$ (turbulent case)*

To further demonstrate the capability of the POD/DEIM, the Reynolds number is then increased to $Re = 1000$, which leads to appearance of a shock wave, and a sharp front can be observed in Figure 7. This is of interest, as it frequently occurs in practical engineering applications. In this case, the addition of extra spatial grid points and a shrinking of the time step size is required to maintain stability of the computation. In the current case, let the spatial grid points mesh be $200 \times 200$, and the time step size be chosen as $0.001$. Though $I = 99.8\%$, we find that additional number of POD bases are chosen, and the number of DEIM interpolation points needs also to be increased according to the requirement of our computation. Considering a compromise between accuracy and computational time in Tables 3,4, we take the number of POD modes and DEIM points as 15 and 250, respectively. From Table 3 , we see that the POD ROM hardly reduces the computational time (CPU time) in comparison with the Full Model whose CPU time is 814.6910, whereas the POD15/DEIM250 still maintains its advantage with the computational time reduced by a factor of $O(10)$, see Table 4. This implies that the POD scheme is not able to reduce the computational complexity for nonlinear systems but on the contrary, it increases sometimes the computational cost. In addition, we also provide a comparison of $u(x, y, t)$ between the full model and POD15/DEIM250 at different model times $nt = 100, 600, 1000$, respectively, which illustrate that the POD/DEIM is applicable to the case of large Reynolds number, see Figure 7.

Through the comparison between the cases of $Re = 100$ and $Re = 1000$, we find:

(1). the POD/DEIM ROM can retain the same reduction of computational time by a factor of $O(10)$. The computational stability of POD/DEIM ROM is kept by means of the careful selection of the finite POD modes and DEIM interpolation points. The solution of POD/DEIM in the case of $Re = 1000$ has accuracy with error $O(10^{-3})$ versus $O(10^{-4})$ in the case of $Re = 100$ due to the additional energy contained the neglected POD state modes and POD nonlinear modes used for DEIM;

(2). the numerical experiment suggests that additional points are beneficial when determining the DEIM interpolation points. As for the optimal selection of POD modes, it is not sufficient to rely only on equation (2.1.2) since a slower decay of the eigenvalue spectrum can be observed for large Reynolds number by comparing Figure 1 with Figure 6. This suggests that a modest number of POD modes is required for attaining a high accuracy computational result. Certainly, this will inevitably increase the computational burden. Providing an optimal truncation of POD modes remains an open problem [29].

(3). When the number of POD modes is increased up to 15, some spurious oscillation can still be found near the steepened front, see Figure 7, due to the truncation of POD modes and DEIM interpolation of nonlinear term.

This leads us to consider recovery of the lost information by calibrating the evolutionary coefficients which amounts to a turbulence closure.

## 4. CALIBRATION USING TIKHONOV REGULARIZATION

From above, we see that low-order modeling indeed provides a way to simplify the 2D Burgers equation into a minimal set of ordinary differential equations (ODEs). However, as pointed out in [31,38], there still exists a major barrier for POD Galerkin approach at high Reynolds numbers due to lack of a turbulence closure. As carefully explained in [29], for realistic turbulent flows, the high index POD modes that are not included in the POD-ROM Galerkin do have a significant effect on the dynamics of the POD-ROM. We will deal with this issue by using a closure model consisting of Tikhonov regularization to account for the small scale dissipation effect of the truncated POD modes. (As recently used in [38]). So it is first assumed that the calibrated ROM can be expressed as follows:

$$\dot{a}(t) = f(y, a(t)) \tag{4.1}$$

where $y$ represents polynomial coefficients, $N_y$ is denoted as the number of components of $y$ that is equal to [$N_0$(constant terms)+$N_1$ (linear terms)+$N_2$ (quadratic terms)], while $a(t) =$

$(a_1(t), a_2(t), \cdots, a_m(t))$, $f(y, a(t)) = C + La(t) + Ha(t).*(Qa(t)))$ where $C, L, H$, and $Q$ are the relevant coefficient matrices to be determined. $f$ is a polynomial of degree 2 in $a(t)$, which can be written componentwise as follows:

$$f^{(k)}(y, a(t)) = c_k + \sum_{i=1}^{m} l_{ik} a_i + \sum_{i=1}^{m}\sum_{j=1}^{m} a_i D_{ij} a_j \tag{4.2}$$

In our present case, $m = 15$ is set as the number of POD modes for the case of POD15/DEIM250. The calibration process consists in correcting whole or part of polynomial coefficients originating from POD Galerkin by using the exact temporal coefficients known in advance. For that purpose, we introduce an error function,

$$e(y, t) = \dot{a}(t) - f(y, a(t)) \tag{4.3}$$

the calibration of the coefficients can be then done by minimizing the function with Tikhonov regularization term in space $R^{N_y}$

$$\begin{aligned} J(y) &= \left\langle \|e(y,t)\|_2^2 \right\rangle_{T_0} + \gamma^2 \|Ly\|_2^2 \\ &= \frac{1}{N}\sum_{k=1}^{N}\sum_{i=1}^{m} \left(\dot{a}_i(t_k) - f^{(i)}(y, a(t_k))\right)^2 + \gamma^2 \|Ly\|_2^2, \end{aligned} \tag{4.4}$$

where the $\gamma$ is regularization parameter, $L = I$ and $\|\cdot\|_2$ stands for 2-norm (or Euclidean length). The minimization of the function (4.4) amounts to solving the linear system

$$(A^T A + \gamma^2 I)y = A^T b \tag{4.5}$$

where $I$ is the identity matrix of order $N_y$, for details of $A$ and $b$, please refer to [38]. Details of the Tikhonov regularization procedure are based on the following considerations:

(1). the minimization problem (4.4) is not well conditioned if $\gamma = 0$. That is clearly seen when the solution to the linear system (4.5) is written in the following form using the singular value decomposition (SVD)of $A$:

$$y = \Sigma_{i=1}^{m} \frac{\sigma_i^2}{\sigma_i^2 + \gamma^2} \frac{u_i^T b}{\sigma_i} v_i \tag{4.6}$$

here $u$ and $v$ are left (right) singular vectors, $\sigma$ is the singular value, and $u_i^T b$ is called the Fourier expansion coefficient. We can find from Figure 10 that some small Fourier coefficients do not decrease sufficiently fast compared with the small singular value, while the quotient $\frac{|u_i^T b|}{\sigma_i}$ increases to a higher level after a certain index $i$. This implies that the Picard criterion [49,50,51] is only partially satisfied, which will lead to an amplification of noise included in $b$. The quality of the solution is thus greatly affected. However, if let $\gamma \neq 0$, we see that $\frac{\sigma_i^2}{\sigma_i^2+\gamma^2}$ will act as a filter function and a suitable $\gamma$ is necessary.

(2). The key to the success of Tikhonov regularization technique is related to the accurate determination of the regularization parameter $\gamma$. To do so, the L-curve method implemented in the package REGULARIZATION TOOLS [52] is used throughout the paper. The L-curve method is based on the analysis of the curve representing the semi-norm of the regularized solution $\|Ly\|_2$ versus the corresponding residual norm $\|Ay - b\|_2$. In most of the cases, this curve exhibits a typical L shape (see Figure 10). The corner of the L-curve represents a fair compromise between the minimization of the norm of the residual (horizontal branch) and the semi-norm of the solution (vertical part). In present case of $Re = 1000$, $\gamma$ is set as 0.0053172. This means that the contribution of Fourier coefficient corresponding to the small singular value is dampened, a stable solution $y$ is therefore derived.

Substitution of solution result of equation (4.5) into the equation (4.1) yields a determined calibrated model (called calibrated ROM). Subject to the initial condition

$$a_\alpha|_{t=0} = \Phi_u^T U\big|_{t=0},\ a_\beta|_{t=0} = \Phi_v^T V\big|_{t=0} \tag{4.7}$$

the computational results are obtained using classical fourth order Runge-Kutta algorithm. The partial evolutionary results of the calibrated coefficient are shown in Figure 11. We see that the calibrated temporal POD coefficients and the original ones are in good agreement over the entire time interval $[0, T]$. In addition, the calibration of the coefficients leads to improvement of the solution $u(x, y, t)$, which can be observed from Figures 8,9, respectively.

It is worth noting that the flow calibration with Tikhonov regularization is essentially a least-squares estimation (data fitting), which is different from other well-known closure modeling approaches based on physical insight stemming from the turbulent simulation [29]. The calibration of evolutionary coefficient in the present study is accomplished through the correction of the polynomial coefficient of the constructed dynamical system. Apart from providing good retrieval results for the coefficients of high index POD modes, another advantage of the current method is that the calibration process is automatically performed i.e. the interest of the Tikhonov regularization is that the value of the parameter of regularization is determined automatically, via the L-curve method.

## 5. CONCLUSIONS

The model order reduction with POD/DEIM approach is applied to the two dimensional Burgers equation in the case of $Re = 100$ and $Re = 1000$ (representing the turbulent case), respectively. The main goal was to assess the effect of large Reynolds number when using a combination of POD/DEIM and a Tikhonov regularization as a closure model. This combination appears to be novel in as far as we can assess. For the sake of simplicity, the computational results shown in the present study are for $u$ only, and the ones for $v$ are not provided here. Five POD modes in the case of $Re = 100$ are used to derive the POD ROM which captures more than $99.8\%$ of the system energy, and fifty DEIM interpolation points are selected for the interpolation of nonlinear functions, which leads to a large improvement in the computational speed. In the case of $Re = 1000$ (turbulent case), several facts can be observed. First, a smooth wave becomes a shock wave, and some spurious oscillation can be found near the steepened front due to the truncted modes which play an important role in dissipating energy. Second, the eigenvalue spectrum has a slower decay than that in the case of $Re = 100$. This means that main energy concentrates on more POD modes. Additional POD modes are required to maintain sufficient computational accuracy. However, this in turn will require additional computational cost. Thirdly, how to provide an optimal truncation of POD modes remains an open problem. It is not sufficient to rely only on the criterion (2.1.2). As for the DEIM, additional interpolation points prove to be beneficial in terms of CPU speed-up. Fourth, larger Reynolds number incurs a larger error due to the truncation of POD modes and DEIM interpolation of nonlinear term as well as numerical computation, which cannot be neglected for a longer time integration interval. For this purpose we apply calibration with Tikhonov regularization serving as the closure model. A noticeable improvement is obtained resulting in large computational gain (CPU time 1.7597 using four-order Runge-Kutta method). Our present high fidelity model is discretized using a second-order central finite difference discretization in space and a first order backward Euler scheme in time, which allow us to perform the simulation only up to $Re = 1000$. For a larger Reynolds number, it is possible to construct a POD/DEIM ROM combined with a higher-order numerical scheme so as to maintain the numerical stability. This will be investigated in future research work.

## ACKNOWLEDGEMENT

The authors are very grateful to the anonymous reviewers for carefully reading this manuscript and for their pertinent and detailed remarks that contributed to improve presentation clarity and readability of the manuscript. This work reported here is supported by the National Natural Science Foundation of China (Grant No.41375115; 61572015). The first author would like to acknowledge

Department of Scientific Computing, Florida State University that hosted him during his visit to the US.

## APPENDIX

### A. A simple introduction to the discrete Picard condition

It is well known that the Picard theorem states in a continuous setting that in order for the equation

$$Kx = y \qquad (A.1.1)$$

to have a solution $x^\dagger \in X$, it is necessary and sufficient that $y \in \overline{R(K)}$ and that

$$\Sigma_{i=1}^{\infty} \frac{< u_i, y >^2}{{\sigma_i}^2} < \infty \qquad (A.1.2)$$

where $K$ is a compact operator between the real Hilbert spaces $X$ and $Y$, and $(\sigma_i, u_i, v_i)$ is a singular system of $K$. The infinite sum in (A.1.2) must converge, which means that the terms in the sum must decay to zero, or equivalently, that the generalized Fourier coefficients $| < u_i, y > |$ must decay faster to zero than the singular values $\sigma_i$ for $i = 1, 2, \cdots$ [49].

For the discrete ill-posed problems there is, strictly speaking, no Picard condition because the solution always exists and can never become unbounded. Nevertheless it makes sense to introduce the discrete Picard condition. Suppose that the operator equation in question has been reduced by discretization to a set of linear equations

$$Kx = y \qquad (A.1.3)$$

Here let $K$ be a $n \times p$ matrix of rank $p$, $(n \geq p)$, and $K = U\Sigma V^T = \Sigma_{i=1}^{p} \sigma_i u_i v_i$, then the least squares solution of the linear equation (A.1.3)[49] is given by

$$x_{ols} = K^\dagger y = V\Sigma^\dagger U^T y = \Sigma_{i=1}^{p} \frac{u_i^T y}{\sigma_i} v_i \qquad (A.1.4)$$

From the relation (A.1.4), we observe that due to the presence of a very small $\sigma_p$ in the denominator, the solution will be very sensitive to the errors (the inaccurate measurements, discrete error as well as finite precision numerical computation) included in $y$. To be more concrete, we choose a large singular-value index $i^*$ and consider a perturbation of the exact vector $y$ in the direction of the singular vector $u_i^*$, i.e.

$$y^\delta = y + \beta u_i^* \qquad (A.1.5)$$

with $\beta = \|y^\delta - y\|_2$ being the noise level. The least squares solution is then provided by

$$x_{ols}^\delta = x + \frac{\beta}{\sigma_i^*} v_i^* \qquad (A.1.6)$$

so the following relation results in [49,50]

$$\frac{\|x_{ols}^\delta - x\|_2}{\|y^\delta - y\|_2} = \frac{1}{\sigma_i^*} \qquad (A.1.7)$$

if $\sigma_i^*$ is very small, the computed solution by (A.1.4) can be completely dominated by the SVD coefficients corresponding to the smallest singular values. In other words, the $x_{ols}$ will be very far from the exact solution, and instability of solution will occur.

Based on the analysis of the SVD coefficients, together with an understanding of their relation to the SVE (singular vector expansion) coefficients in (A.1.4), the discrete version of the Picard condition is therefore introduced. This was pointed out by Per Christian Hansen in [51]. Let $\tau$ denote the level at which the computed singular values $\sigma_i$ level off due to rounding errors, The discrete Picard condition is satisfied if, for all singular values larger than $\tau$, the corresponding Fourier coefficients $\mid u_i^T y \mid$, on average, decay faster than the $\sigma_i$. The discrete Picard condition will play an important role in dealing with the discrete ill-posed problems to ensure solution stability.

The quantities of importance are the ratios of Fourier coefficients and the singular values rather than their individual values. When the Picard condition is satisfied, the ratios will decrease, and if the ratios begin to drop and then grow again , then the Picard condition is not satisfied, as in the case shown in this paper, illustrated by figure 10. The violation of the Picard condition may serve as an explanation of the instability of the linear inverse problem under consideration.

The insight we gain from a study of the quantities associated with the SVE provides a hint on how to deal with noisy problems that violate the Picard condition, that is, we need regularization methods that can result in less sensitive approximations to the exact solution. We aim to filter out the unwanted part of (A.1.4). In view of the similarity of two representative methods: Truncated SVD (TSVD) and Tikhonov regularization[51], we only focus on the Tikhonov regularization method. The Tikhonov solution $x_{ols}^{\lambda}$ is defined as the solution to the problem

$$min\{\|Kx-y\|_2^2+\lambda^2\|x\|_2^2\} \qquad (A.1.8)$$

The first term measures the goodness-of-fit, and the second term measures the regularity of the solution. The regularization parameter $\lambda$ controls the weighting between the two terms. If we can control the norm of $x$, then we can suppress (most of) the large noise components, which can be seen clearly from the solution of (A.1.8)

$$x_{ols}^{\lambda}=\Sigma_{i=1}^{p} f_i^{\lambda}\frac{u_i^T y}{\sigma_i}v_i \qquad (A.1.9)$$

Comparing equations (A.1.4) and (A.1.9), we see that the filter factors are introduced here for $i=1,2,\cdots,p$, which satisfy

$$f_i^{\lambda}=\frac{\sigma_i^2}{\sigma_i^2+\lambda^2}=\begin{cases}1 & \sigma_i \gg \lambda\\ \frac{\sigma_i^2}{\lambda^2} & \sigma_i \ll \lambda\end{cases} \qquad \text{(A.1.10)}$$

We stress here that the selection of the parameter $\lambda$ is crucial for the regularization process. The method to determine it in the current paper is based on the L-curve method. For additional details, please see [49,50,51].

## B. Newton method of Full-order Model

Equations (2.0.4a,b) are just a semi-discretized system (ODEs) of equations (2.0.1a,b). The numerical solution of the system can be carried out by using one of the known procedures for the temporal discretization of the approximate solutions of ordinary differential equations such as Runge-Kutta method. However, when backward Euler scheme in time is performed, there is another way to obtain the solution, namely the Newton iterative method is employed [45]. This method converges fast with each subsequent convergence error proportional to the square of its predecessor provided we start from a good initial guess for the Newton iteration. In a time-stepping context, a good guess is always available, namely taking the value obtained at the end of the previous time step. The time interval $[0,T]$ is discretized into $n_t=\frac{T}{\triangle t}$ equal segments, where $\triangle t$ denotes the time step size. According to the expression of model (2.0.4a,b), we introduce two maps $F_1, F_2:\mathbb{R}^{n_{xy}}\times\mathbb{R}^{n_{xy}}\to\mathbb{R}^{n_{xy}}$ to define the relation as follows for the purpose of carrying out the Newton iteration for $U^{n+1}=U(t_n), V^{n+1}=V(t_n)$ :

$$F_1(U^{n+1},V^{n+1})=0 \qquad (A.2.1a)$$

$$F_2(U^{n+1},V^{n+1})=0 \qquad (A.2.1b)$$

where $t_n=n\triangle t$, and the $F_1$ and $F_2$ are represented by

$$F_1(U^{n+1},V^{n+1})=U^{n+1}-U^n+\triangle t f_1(U^{n+1},V^{n+1})-\frac{\triangle t}{2\triangle x}B_{ul}U^{n+1}-$$
$$\frac{\triangle t}{2\triangle y}B_{ub}V^{n+1}-\frac{\triangle t}{Re\triangle x^2}(D_1U^{n+1}+b_{1u})-\frac{\triangle t}{Re\triangle y^2}(D_2U^{n+1}+b_{2u}); \qquad \text{(A.2.2a)}$$

$$F_2(U^{n+1},V^{n+1}) = V^{n+1} - V^n + \triangle t f_2(U^{n+1},V^{n+1}) - \frac{\triangle t}{2\triangle x}B_{vl}U^{n+1} -$$
$$\frac{\triangle t}{2\triangle y}B_{vb}V^{n+1} - \frac{\triangle t}{Re\triangle x^2}(D_1V^{n+1}+b_{1v}) - \frac{\triangle t}{Re\triangle y^2}(D_2V^{n+1}+b_{2v}) \tag{A.2.2b}$$

Thus a linear system of algebraic equations can be derived due to the Taylor series expansion at the $k$ iteration,

$$J^k \begin{pmatrix}\delta U_k^{n+1}\\ \delta V_k^{n+1}\end{pmatrix} = -\begin{pmatrix}F_1\\ F_2\end{pmatrix}^k \tag{A.2.3}$$

The coefficient matrix is the Jacobian, which is expressed as

$$J = \begin{pmatrix}F_{1u} & F_{1v}\\ F_{2u} & F_{2v}\end{pmatrix} \tag{A.2.4}$$

where

$$F_{1u} = \frac{\partial F_1}{\partial U^{n+1}} = E + \triangle t(\frac{\partial f_1}{\partial U^{n+1}} - \frac{1}{2\triangle x}B_{ul}\ ) - \frac{\triangle t}{Re\triangle x^2}D_1 - \frac{\triangle t}{Re\triangle y^2}D_2; \tag{A.2.5}$$

$$F_{1v} = \frac{\partial F_1}{\partial V^{n+1}} = \triangle t(\frac{\partial f_1}{\partial V^{n+1}} - \frac{1}{2\triangle y}B_{ub}); \tag{A.2.6}$$

$$F_{2u} = \frac{\partial F_2}{\partial U^{n+1}} = \triangle t(\frac{\partial f_2}{\partial U^{n+1}} - \frac{1}{2\triangle x}B_{vb}); \tag{A.2.7}$$

$$F_{2v} = \frac{\partial F_1}{\partial V^{n+1}} = E + \triangle t(\frac{\partial f_2}{\partial V^{n+1}} - \frac{1}{2\triangle y}B_{vb}) - \frac{\triangle t}{Re\triangle x^2}D_1 - \frac{\triangle t}{Re\triangle y^2}D_2; \tag{A.2.8}$$

while $\frac{\partial f_1}{\partial U^{n+1}}$, $\frac{\partial f_1}{\partial V^{n+1}}$, $\frac{\partial f_2}{\partial U^{n+1}}$ and $\frac{\partial f_2}{\partial V^{n+1}}$ are such that

$$\frac{\partial f_1}{\partial U^{n+1}} = \frac{1}{2\triangle x}(\text{diag}(MU^{n+1}) + \text{diag}(U^{n+1})M) + \frac{1}{2\triangle y}\text{diag}(V^{n+1})N; \tag{A.2.9}$$

$$\frac{\partial f_1}{\partial V^{n+1}} = \frac{1}{2\triangle y}\text{diag}(NU^{n+1}); \tag{A.2.10}$$

$$\frac{\partial f_2}{\partial U^{n+1}} = \frac{1}{2\triangle x}\text{diag}(MV^{n+1}); \tag{A.2.11}$$

$$\frac{\partial f_2}{\partial V^{n+1}} = \frac{1}{2\triangle x}\text{diag}(U^{n+1})M + \frac{1}{2\triangle y}(\text{diag}(NV^{n+1}) + \text{diag}(V^{n+1})N). \tag{A.2.12}$$

By solving equations (A.2.3), the $(U_k^{n+1}, V_k^{n+1})^T$ can be updated like this

$$\begin{pmatrix}U_{k+1}^{n+1}\\ V_{k+1}^{n+1}\end{pmatrix} = \begin{pmatrix}U_k^{n+1}\\ V_k^{n+1}\end{pmatrix} + \begin{pmatrix}\delta U_k^{n+1}\\ \delta V_k^{n+1}\end{pmatrix} \tag{A.2.13}$$

Check if the conditions given by equation (A.2.14) are satisfied, in which case stop the iteration and adopt as solution the last iterate. Otherwise continue the iterative process.

$$\left\|\begin{pmatrix}\delta U_k^{n+1}\\ \delta V_k^{n+1}\end{pmatrix}\right\| < \text{tol}, or\ \left\|F(U_{k+1}^{n+1})\right\| < \text{tol}, \tag{A.2.14}$$

The tol can be used as a stopping criterion. In the present work, we let tol $= 10^{-6}$ serve as stopping criterion. And the solutions $U^n$ and $V^n$ of model (2.2.1a,b) are denoted as $\mathbf{U_{full}}$ and $\mathbf{V_{full}}$, respectively.

**C. POD Reduced order model of 2D Burgers equation**

Through applying the Galerkin projection method to the eqs.(2.0.4a,b), the $U$,$V$ therein can be replaced with $\Phi_u\alpha(t_n)$ and $\Phi_v\beta(t_n)$, respectively. And then multiplying the equations (2.0.4a) and (2.0.4b) by $\Phi_u^T$ and $\Phi_v^T$, we obtain the POD reduced order model (**POD ROM**)

$$\frac{d\alpha}{dt}+f_1^{POD}(\alpha,\beta)-\frac{1}{2\triangle x}B_{ul}^{POD}\alpha-\frac{1}{2\triangle y}B_{ub}^{POD}\beta= \frac{1}{Re\triangle x^2}(D_{11}^{POD}\alpha+b_{1u}^{POD})+\frac{1}{Re\triangle y^2}(D_{12}^{POD}\alpha+b_{2u}^{POD}) \quad \text{(A.3.1a)}$$

$$\frac{d\beta}{dt}+f_2^{POD}(\alpha,\beta)-\frac{1}{2\triangle x}B_{vl}^{POD}\alpha-\frac{1}{2\triangle y}B_{vb}^{POD}\beta= \frac{1}{Re\triangle x^2}(D_{21}^{POD}\beta+b_{1v}^{POD})+\frac{1}{Re\triangle y^2}(D_{22}^{POD}\beta+b_{2v}^{POD}) \quad \text{(A.3.1b)}$$

where

$$\alpha(t_n)\in\mathbb{R}^{m_1},\beta(t_n)\in\mathbb{R}^{m_2}$$

$$f_1^{POD}(\alpha,\beta)=\Phi_u^T f_1(\Phi_u\alpha,\Phi_v\beta), \quad (A.3.2a)$$

$$f_2^{POD}(\alpha,\beta)=\Phi_v^T f_2(\Phi_u\alpha,\Phi_v\beta), \quad (A.3.2b)$$

$$D_{11}^{POD}=\Phi_u^T D_1\Phi_u, D_{12}^{POD}=\Phi_u^T D_2\Phi_u, D_{21}^{POD}=\Phi_v^T D_1\Phi_v, D_{22}^{POD}=\Phi_v^T D_2\Phi_v,$$

$$B_{ul}^{POD}=\Phi_u^T B_{ul}\Phi_u, B_{ub}^{POD}=\Phi_u^T B_{ub}\Phi_v, b_{1u}^{POD}=\Phi_u^T b_{1u}, b_{2u}^{POD}=\Phi_u^T b_{2u},$$

$$B_{vl}^{POD}=\Phi_v^T B_{vl}\Phi_u, B_{vb}^{POD}=\Phi_v^T B_{ub}\Phi_v, b_{1v}^{POD}=\Phi_v^T b_{1v}, b_{2v}^{POD}=\Phi_v^T b_{2v},$$

It can be seen that the equations (A.3.1a,b) have the same form as the equations (2.0.4a,b). For comparison purpose, the Newton iterative method is still adopted. But the gradients of nonlinear functions with respect to $\alpha,\beta$ will assume following form:

$$\frac{\partial f_1^{POD}}{\partial\alpha}=\Phi_u^T\frac{1}{2\triangle x}[\text{diag}(\Phi_u\alpha)M\Phi_u+ \text{diag}(M\Phi_u\alpha)\Phi_u]+\Phi_u^T\frac{1}{2\triangle y}\text{diag}(\Phi_v\beta)N\Phi_u, \quad \text{(A.3.3)}$$

$$\frac{\partial f_1^{POD}}{\partial\beta}=\Phi_u^T\frac{1}{2\triangle y}\text{diag}(N\Phi_u\alpha)\Phi_v, \quad (A.3.4)$$

$$\frac{\partial f_2^{POD}}{\partial\beta}=\Phi_v^T\frac{1}{2\triangle x}\text{diag}(\Phi_u\alpha)M\Phi_v+ \Phi_v^T\frac{1}{2\triangle y}[\text{diag}(N\Phi_v\beta)\Phi_v+\text{diag}(\Phi_v\beta)N\Phi_v], \quad \text{(A.3.5)}$$

$$\frac{\partial f_2^{POD}}{\partial\alpha}=\Phi_v^T\frac{1}{2\triangle x}\text{diag}(M\Phi_v\beta)\Phi_u. \quad (A.3.6)$$

## REFERENCES

1. P. Arminjon, C. Beauchamp, Numerical solution of Burgers' equation in two space dimensions,*Computer Methods in Applied Mechanics and Engineering* ,1979; 19(3): 351-365.
2. M. Basto, V. Semiao, F. Calheiros, Dynamics and synchronization of numerical solutions of the 3 equation, *J. Comput. Appl. Math*, 2009; 231:793-803.
3. Bahadir, A.R., A fully implicit finite difference scheme for two-dimensional Burgers equations, *Appl. Math. Comput.*, 2003; 137:131-137.
4. H.M.Park, Y.D.Jang, Control of Burgers equation by means of mode reduction, *International Journal of Engineering Science*, 2007; 38:785-805.

5. B. V. Rathish Kumar and M. Mehra, A three-step wavelet Galerkin method for parabolic and hyperbolic partial differential equations, *International Journal of Computer Mathematics*, 2006; 83(1):143-157.
6. H.Zhu, H. Shu and M. Ding, Numerical solutions of two-dimensional Burgers' equation by discrete Adomian decomposition method, *Computers and Mathematics with Applications*, 2010; (60):840-848.
7. C.A.J. Fletcher, Generating exact solutions of the two-dimensional Burgers equation, *Int. J. Numer. Meth. Fluids*, 1983; (3):213C216.
8. N. Smaoui, Boundary and distributed control of the viscous Burgers equation, *Journal of Computational and Applied Mathematics*, 2005; 182:91-104.
9. H.M.Park, Y.D.Jang, Control of Burgers equation by means of mode reduction,*International Journal of Engineering Science*, 2000; 38:785-805.
10. K. Kunisch, S. Volkwein, Control of the Burgers Equation by a Reduced- Order Approach Using Proper Orthogonal Decomposition,*Journal of Optimization Theory and Applications*, 1999; 102(2):345-371.
11. Z. Wang, I. Akhtar, J. Borggaard, and T.Iliescu, Two-Level Discretizations of Nonlinear Closure Models for Proper Orthogonal Decomposition,*J. Comput. Phys.*, 2011, 230: 126-146.
12. R. Stefanescu, A. Sandu and I.M.Navon, Comparison of POD reduced order strategies for the nonlinear 2D shallow water equations, *Int. J. Numer. Meth. Fluids*, 2014; 76:497-521.
13. F. Fang , C.C. Pain, I.M. Navon, M.D. Piggott, G.J. Gorman, P. Allison, A.J.H. Goddard, A POD reduced order unstructured mesh ocean modelling method for moderate Reynolds number flows, *Ocean Modelling*, 2009; 28:127-136.
14. S.Ravindran, A reduced-order approach for optimal control of fluids using proper orthogonal decomposition,*Int. J. Numer. Meth. Fluids*, 2000; 34:425-448.
15. S.Ravindran, Reduced-order adaptive controllers for fluid flows using POD,*J. Sci, Comput.*, 2000; 15(4):457-478.
16. Loѐve M., Probability Theory. Van Nostrand: Princeton, NJ, 1955.
17. D.H.Chambers, R.J.Adrian, P.Moin, D.S.Stewart, and H.J.Sung, Karhunen-Loѐve expansion of Burgers model of turbulence,*Phys. Fluid*, 1988; 31(9):2573-2582.
18. Hotelling H. Analysis of a complex of statistical variables with principal components. *Journal of Educational Psychology*, 1933; 24:417-441.
19. Lorenz EN. Empirical orthogonal functions and statistical weather prediction. Technical Report, Massachusetts Institute of Technology, Dept. of Meteorology: Cambridge, MA, 1956.
20. R.C.Gonzalez, P.A. Wintz, Digital Image Processing, Addison-Wesley, Reading, MA, 1987.
21. S.Chaturantabut, Dimension Reduction for Unsteady Nonlinear Partial Differential Equations via Empirical Interpolation Methods. Master of science, Rice University, 2008.
22. S.Chaturantabut, Nonlinear Model Reduction via Discrete Empirical Interpolation. Ph.D. Thesis, Rice University, 2011.
23. S.Chaturantabut, Dimension Reduction for Unsteady Nonlinear Partial Differential Equations via Empirical Interpolation Methods. Technical Report TR09-38, CAAM, Rice University: Houston, TX, 2008.
24. S.Chaturantabut, D.C.Sorensen, A state space error estimate for POD-DEIM nonlinear model reduction.*SIAM Journal on Numerical Analysis*, 2012; 50(1):46-63.
25. S.Chaturantabut, D.C.Sorensen, Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing* , 2010; 32(5):2737-2764.
26. Barrault M, Maday Y, Nguyen NC, Patera AT. An empirical interpolation method: application to efficient reduced basis discretization of partial differential equations.*Comptes Rendus Mathematique*, 2004; 339(9):667-672.
27. Grepl MA, Patera AT. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 2005; 39(01):157-181.
28. Rozza G, Huynh DBP, Patera AT. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations.*Archives of Computational Methods in Engineering*, 2008;15(3):229-275.
29. Z. Wang, Reduced-order modeling of complex engineering and geophysical flows: Analysis and computations, Ph.D. dissertation, Dept. App. Mathematics, Virginia Polytechnic Institute and State University, 2012.
30. N. Aubry, W. Y. Lian, and E.S. Titi, Preserving symmetries in the proper orthogonal decomposition, *SIAM Journal on Scientific Computing*, 1993; 14:483-505.
31. M. Couplet, C. Basdevant, P. Sagaut, Calibrated reduced-order POD-Galerkin system for fluid flow modelling, *Journal of Computational Physics*, 2005; 207:192-220.
32. M. Bergmann, C. H. Bruneau, A. Iollo, Enablers for robust POD models,*Journal of Computational Physics*, 2009; 228: 516-538.
33. A. Iollo, S. Lanteri, J. Desideri, Stability Properties of POD-Galerkin Approximations for the Compressible Navier-Stokes Equations, Theoret. Comput. Fluid Dynamics, 2000; 13:377-396.
34. J. Du, I. M. Navon, J. L. Steward, A. K. Alekseev and Z. Luo, Reduced-order modeling based on POD of a parabolized Navier-Stokes equation model I: forward model, *Int. J. Numer. Meth. Fluids*, 2012; 69:710-730.
35. O. San and T. Iliescu, Proper Orthogonal Decomposition Closure Models for Fluid Flows:Burgers Equation, *International Journal of Numerical Analysis and Modeling Series B* , 2013; 1(1):1-18.
36. L. Perret, E. Collin and J. Delville, Polynomial identification of POD based low-order dynamical system, *Journal of Turbulence*, 2006; 7(17):1-15.
37. V.L. Kalb, A.E. Deane, An intrinsic stabilization scheme for proper orthogonal decomposition based low-dimensional models, *Phys. Fluids*, 2007; 19: 054106.
38. L.Cordier, B. Abou El Majd and J. Favier, Calibration of POD reduced-order models using Tikhonov Regularization, *Int. J. Numer. Meth. Fluids*, 2010; 63:269-296.
39. B. Galletti, A. Bottaro, C.-H. Bruneau, A. Iollo, Accurate model reduction of transient and forced wakes,*European Journal of Mechanics B/Fluids*, 2007; (26):354-366.
40. H.M.Park, Y.D.Jang, Control of Burgers equation by means of mode reduction, *International Journal of Engineering Science*, 2007; 38:785-805.

41. M. Onder Efe, H. Ozbay, Low dimensional modelling and Dirichlet boundary controller design for Burgers equation, *Int. J. Control*, 2004; 77(10):895-906.
42. S.Sahyoun, S. M. Djouadi, Nonlinear Model Reduction Using Space Vectors Clustering POD with Application to the Burgers Equation, *2014 American Control Conference (ACC)* June 4-6, 2014. Portland, Oregon, USA.
43. H. Imtiaz and I. Akhtar, Closure in Reduced-Order Model of Burgers Equation, *Proceedings of 2015 12th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, Islamabad, Pakistan, 13-17th January, 2015.
44. M. Sugihara, S. Fujino, Numerical Solutions of Burgers equation with a large Reynolds number, *Reliable Computing*, 1996; 2(2):173-179.
45. C.T.Kelley, Iterative Methods for Linear and Nonlinear Equations, SIAM, Philadelphia, 1995.
46. P.Holmes, J. L. Lumley, G. Berkooz,C.W. Rowley, Turbulence, Coherent Structures,Dynamical Systems and Symmetry (2nd edition), Cambridge University Press, 2012.
47. S.Chaturantabut, D.C.Sorensen, Application of POD and DEIM on dimension reduction of non-linear miscible viscous fingering in porous media,*Mathematical and Computer Modelling of Dynamical Systems* , 2011; 17(4):337-353.
48. R.Stefanescu, I.M.Navon, POD/DEIM nonlinear model order reduction of an ADI implicit shallow water equations model. *Journal of Computational Physics*, 2013; 237:95-114.
49. A. Doicu, T. Trautmann, and F. Schreier, Numerical Regularization for Atmospheric Inverse Problems, Springer-Verlag Berlin Heidelberg, 2010.
50. P.C.Hansen, Discrete inverse problems: insight and algorithms, SIAM, Philadelphia, 2010.
51. P.C.Hansen, The discrete Picard condition for discrete ill-posed problems, BIT, 1990; 30(4):658-672.
52. P.C.Hansen, Regularization tools: a Matlab package for analysis and solution of discrete ill-posed problems. *Numerical Algorithms*, 1994; 6:1-35.

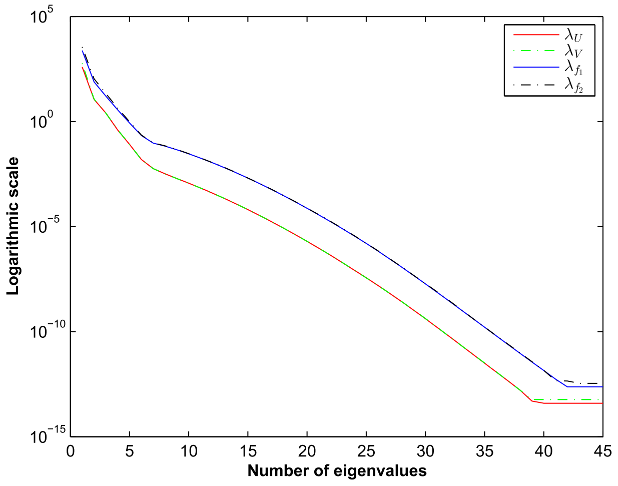


Figure 1. Eigenvalues of solution snapshots and nonlinear function snapshots($Re = 100$)

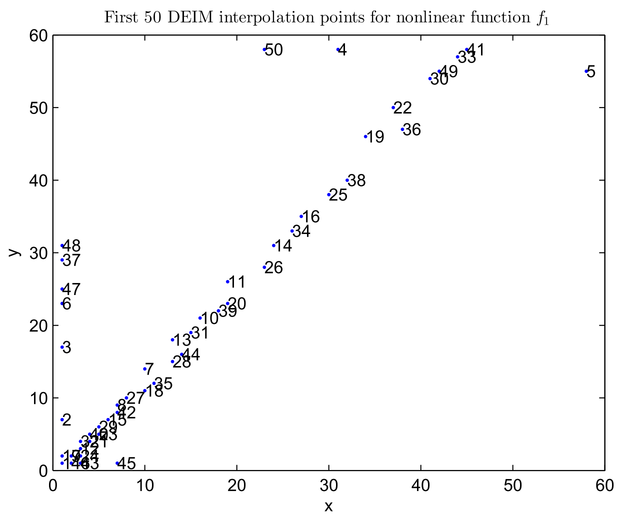


Figure 2. First 50 DEIM interpolation points for nonlinear function $f_1$ ($Re = 100$)

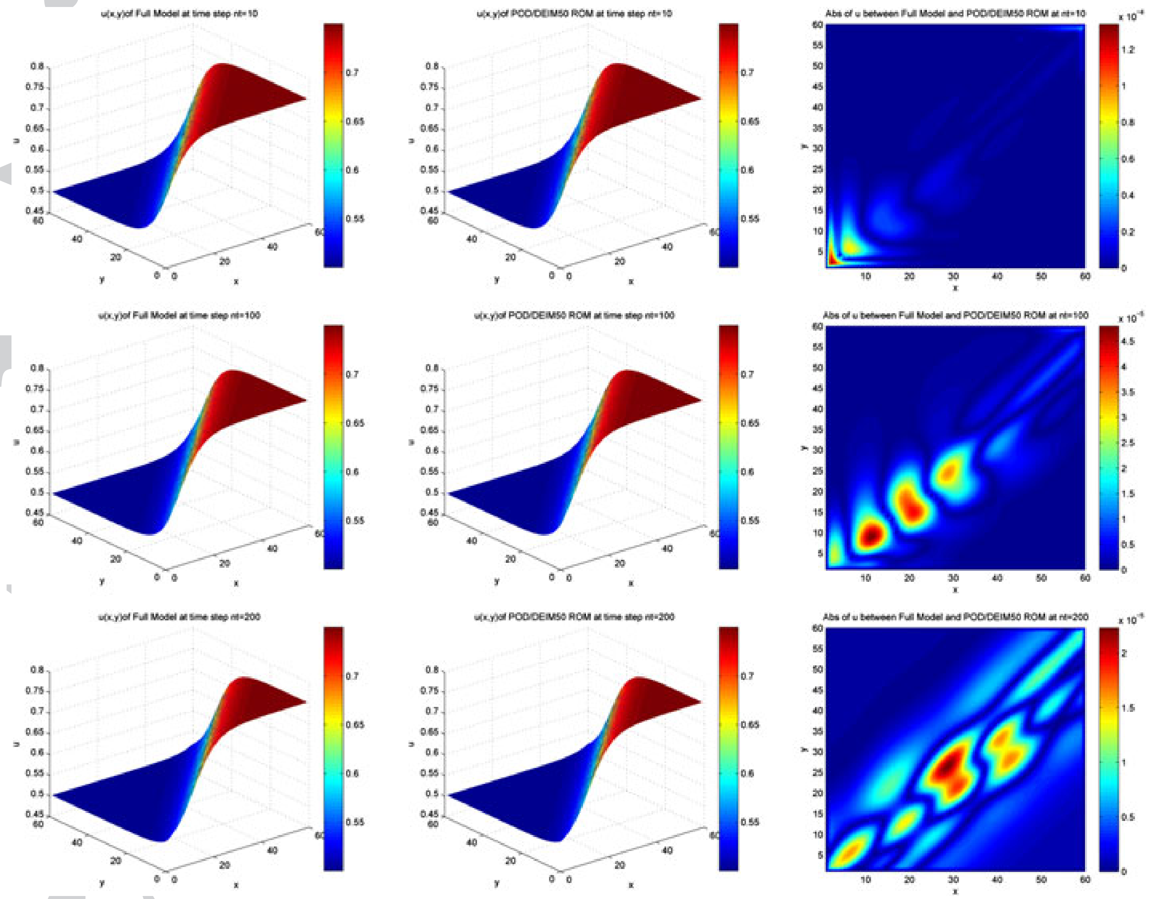


Figure 3. Comparison of $u$ between Full Model and POD5/DEIM50 ROM at $nt$=10,100 and 200 in the case of $Re = 100$.

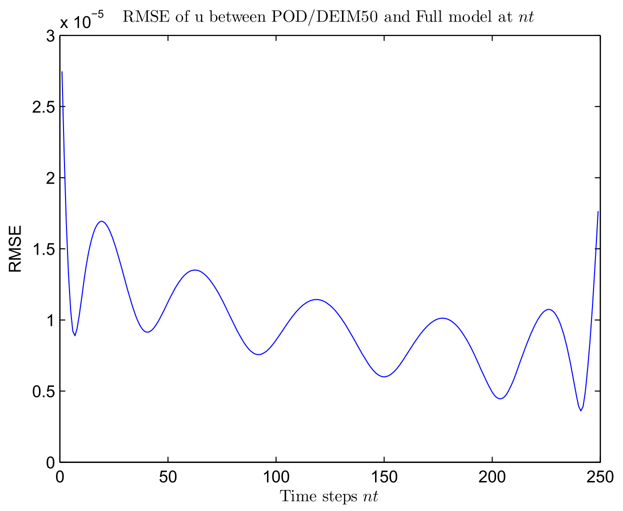


Figure 4. RMSE of $u$ between Full Model and POD15/DEIM250 ROM at each time $nt$, ($Re = 100$)

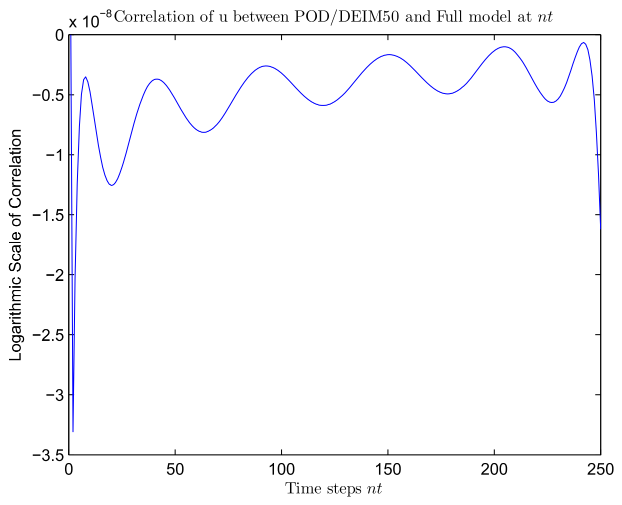


Figure 5. Correlation of $u$ between Full Model and POD15/DEIM250 ROM at each time $nt$, $(Re = 100)$

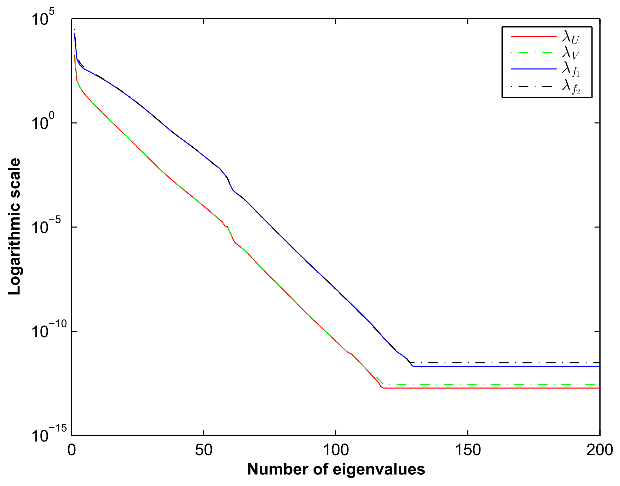


Figure 6. Eigenvalues of solution snapshots and nonlinear function snapshots ($Re = 1000$)

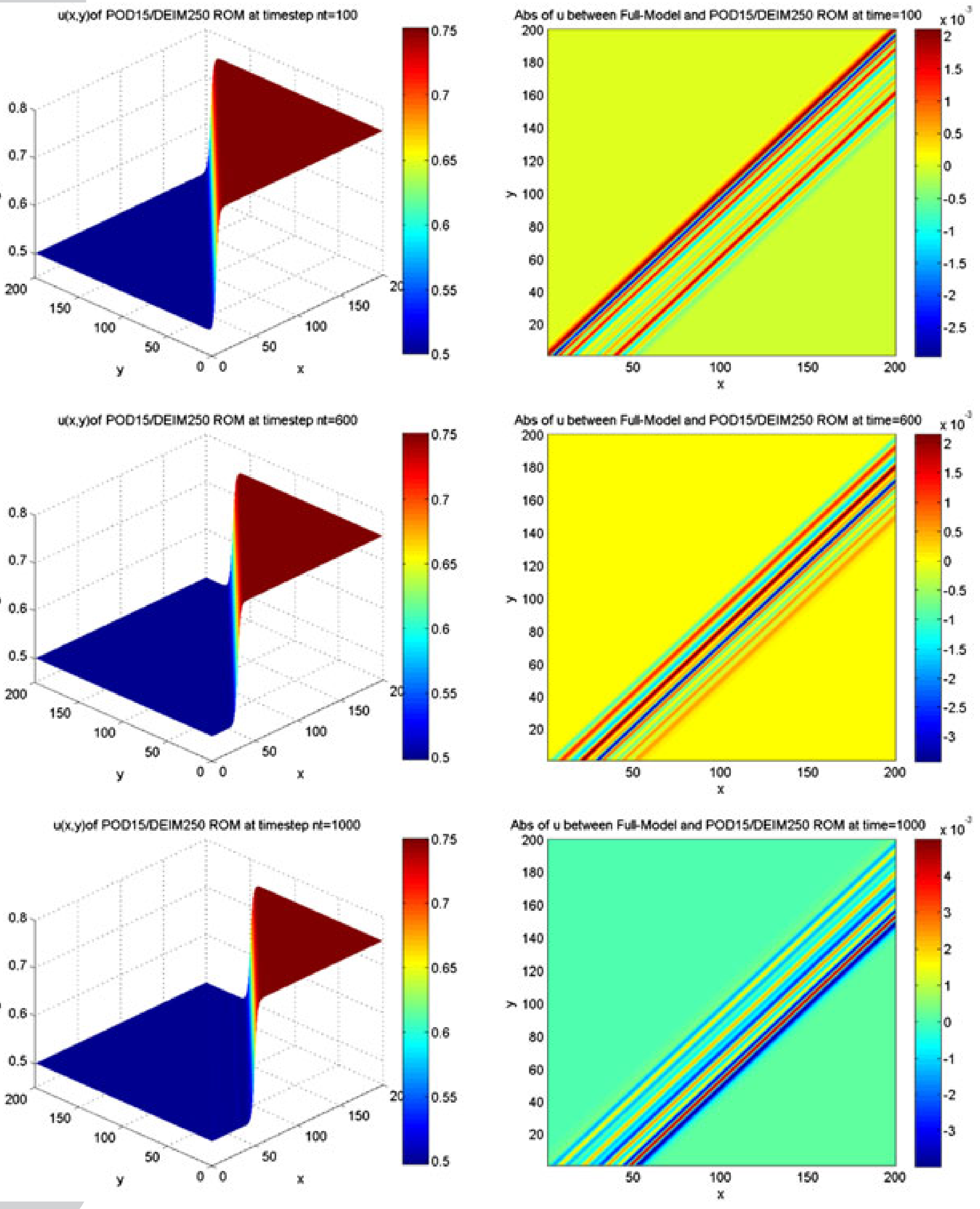


Figure 7. Comparison of $u$ between Full Model and POD15/DEIM250 ROM at nt=100,600 and 1000 in the case of $Re = 1000$.

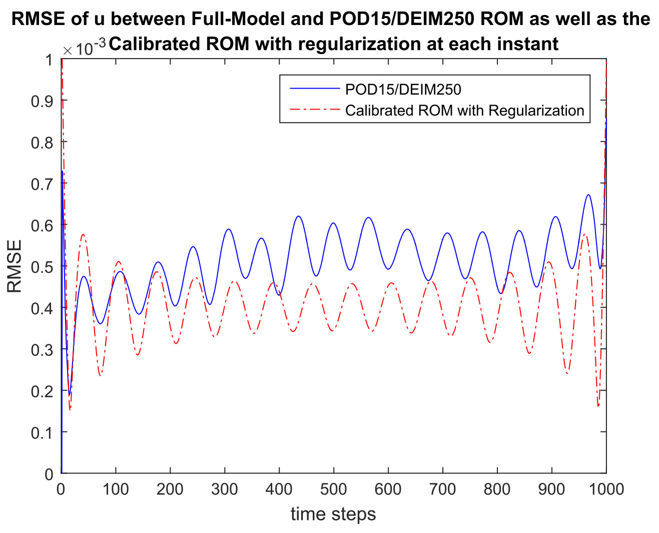


Figure 8. RMSE of $u$ between Full Model and POD15/DEIM250 ROM as well as the Calibrated ROM with Regularization at each time instant $nt$, ($Re = 1000$)

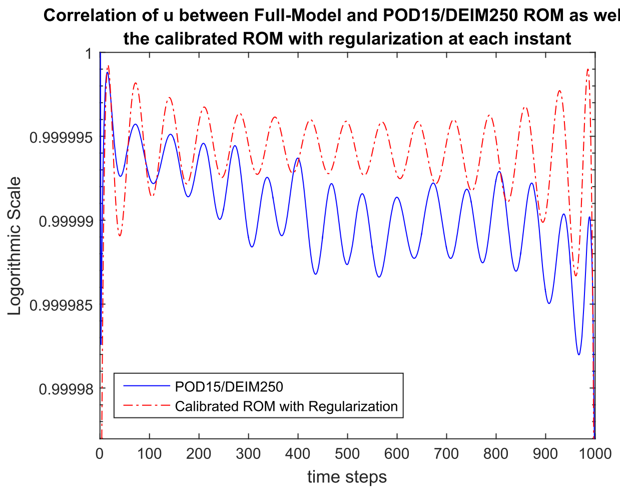
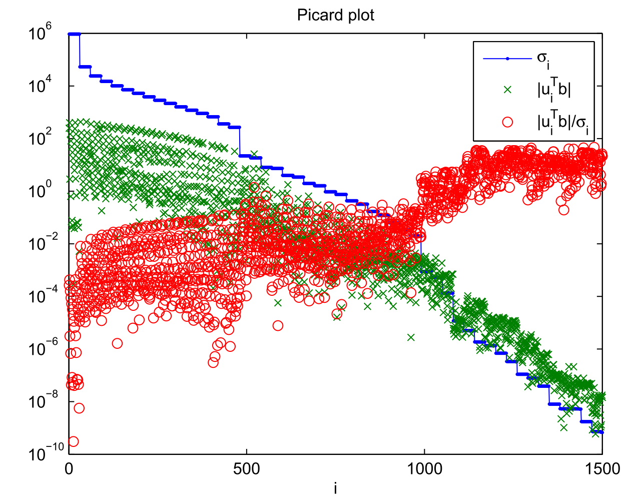


Figure 9. Correlation of $u$ between Full Model and POD15/DEIM250 ROM as well as the Calibrated ROM with Regularization at each time instant $nt$, $(Re = 1000)$



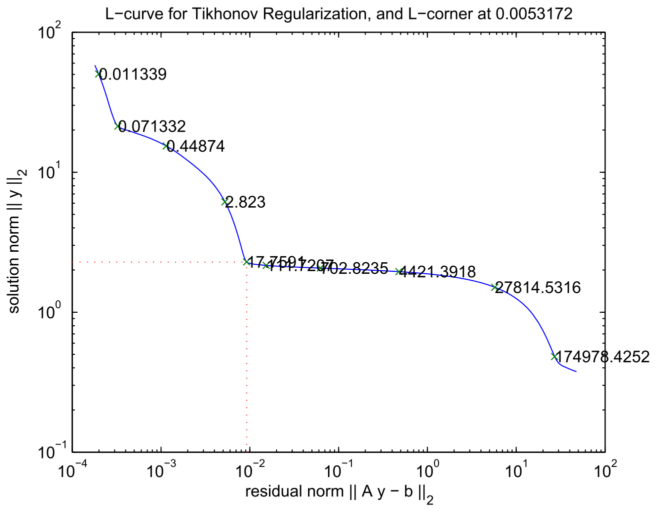


Figure 10. Check of the Picard condition and L-curve

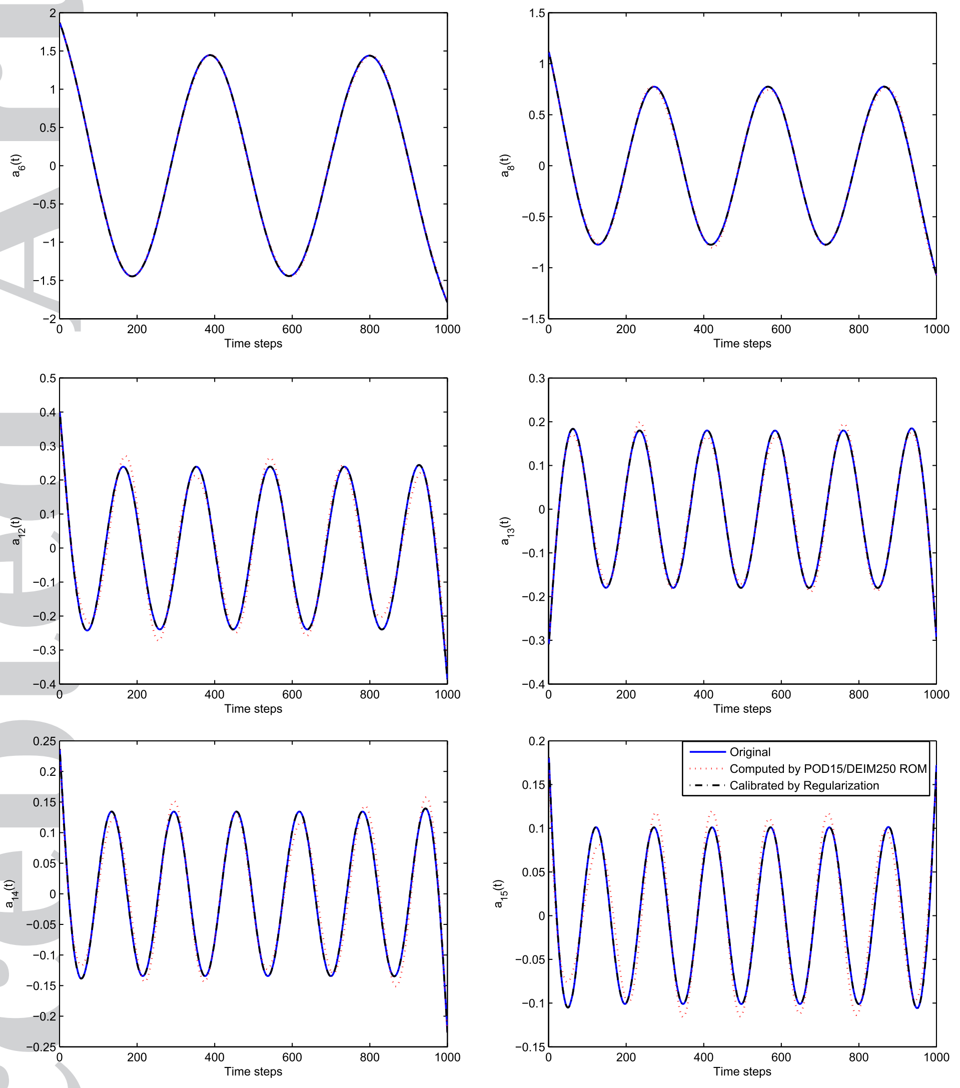


Figure 11. Comparison of the temporal coefficients before and after calibration.

**Table 1.** Comparisons of the CPU time and the average relative errors of POD5/DEIM ROM at different number of DEIM interpolation points for a $60 \times 60$ spatial mesh discretization.

| DEIM points | 10 | 30 | 50 | 60 | 70 | 80 |
|---|---|---|---|---|---|---|
| Cpu time | 1.4088 | 1.4518 | 1.4812 | 1.5355 | 1.5850 | 1.6057 |
| $E_u(\times 10^{-5})$ | 1.6141 | 1.5883 | 1.6219 | 1.6214 | 1.6279 | 1.6472 |

**Table 2.** Comparisons of the CPU time between the Full-Model, the POD-ROM and the POD5/DEIM 50-ROM for a 250 time step integration window (with time step size 0.004) in spatial domain $\Omega = [0, 1] \times [0, 1]$.

| Meshgrids | Full-Model | POD-ROM | POD/DEIM50-ROM |
|---|---|---|---|
| $30 \times 30$ | 3.3465 | 2.3481 | 1.0890 |
| $60 \times 60$ | 12.5944 | 6.7459 | 1.4812 |
| $90 \times 90$ | 32.5888 | 13.3619 | 2.2287 |
| $120 \times 120$ | 58.0337 | 31.6470 | 3.4297 |
| $150 \times 150$ | 96.9749 | 66.0908 | 6.9015 |
| $200 \times 200$ | 172.7232 | 149.9842 | 12.0493 |

**Table 3.** The CPU time and the average relative errors of POD ROM at different modes for a $200 \times 200$ spatial mesh discretization ($Re = 1000$).

| POD MODEs | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|
| Cpu time | 570.4287 | 630.7205 | 694.3305 | 767.7129 | 852.4420 | 927.2546 |
| $E_u(\times 10^{-4})$ | 28.000 | 21.000 | 15.000 | 11.000 | 8.2051 | 6.0803 |

**Table 4.** The CPU time and the average relative errors of POD15/DEIM ROM at different number of DEIM interpolation points for a $200 \times 200$ spatial mesh discretization ($Re = 1000$).

| DEIM points | 200 | 230 | 250 |
|---|---|---|---|
| Cpu time | 75.5601 | 73.8339 | 73.5964 |
| $E_u(\times 10^{-4})$ | 10.000 | 7.2978 | 8.4108 |