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

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

[a,d)] *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

Abstract: 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 discarded POD modes. It is shown that the computational results of this calibrated low-order model (LOM) exhibit considerable agreement with the real high-fidelity model, which implies the efficiency of the closure model used.

Keywords: $2D$ Burgers equation; POD/DEIM reduced order Model; Tikhonov regularization; Calibration

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, and 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 equations have been developed by scientists and engineers, such as [3,5,6,7]. In addition, its analytical solution was also explored [8]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 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 equations model due to the requirement of quick and repeated numerical simulations[9,10,11,12], 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 possible remedy as a powerful and feasible approach enabling a representation of the dynamics of high-dimensional systems on a smaller number of degrees of freedom. Developing low-dimensional models for partial differential equations (PDEs) is one of the active research topics today [13,14].

An effective technique of low-order modelling, Proper Orthogonal Decomposition (POD) is attractive to apply [15,16], and has become one of the most important reduced order model (ROM) methods combined with Galerkin projection, thanks 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-Love expansions [17,18], principal component analysis [19], empirical orthogonal functions [20], and the Hotelling transform [21]. It is one of the most prevalent model reduction methods for nonlinear problems [13]. 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.

First, for models with nonlinear terms, the nonlinear reduced terms still have to be evaluated on the original state space making the simulation of the reduced order system computationally expensive. Presently an interpolation technique known as discrete empirical interpolation method (DEIM) [22,23,24] is the best candidate method that avoids this problem by using an interpolation algorithm [25,26,27]. DEIM approach employs a small selected a set of spatial grid points to avoid evaluation of the expensive inner products at ev-

ery time step that are required to evaluate the nonlinear term and focuses on approximating each nonlinear function so that a certain coefficient matrix can be precomputed and, as a result, the complexity in evaluating the nonlinear term is proportional to only a small number of selected spatial indices, thus a considerable reduction in complexity is achieved.

Second, for the conventional POD-ROM, few modes are sufficient to perform a feasible computation and give a good representation of the kinetic energy of the flow, however, the low-order ODE systems obtained are unstable flow with high Reynolds number due to the discarded POD modes. This is due to discarded small singular values through which energy dissipation occurs in overall dynamical system [28,29]. This can bring about a lack of effectiveness and stability of the POD-Galerkin method [30,31]. Thus robustness is still an issue in turbulent POD-Galerkin ROM. To fix and improve the accuracy and stability of such POD-based reduced-order models, various calibration methods were proposed, such as H^1 Sobolev norm [14,32,and 33], eddy viscosities [34], least square or adjoint method [30,35,36,37,38]. Calibration of LOM 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 [39,40,41,42] 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 the question remains if the use of a proper number of DEIM points is really beneficial for POD/DEIM CPU cost. This paper fills the gap between very simple models and very complicated systems such as those reported in [43]. 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 some fundamental questions related to POD, we believe that they have not been addressed in literature related to 2D Burgers equations with large Reynolds number (Re).

The main contributions of the research presented in this paper are , first, 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 closure model in the case where we deal with a large Re [44] (due to the rather demanding memory requirements, we had to restrict the turbulent case only to Re number equal to 1000). It is expected that a reduced-order ODEs model of the two-

dimensional Burgers equation can yield a decrease in both the memory storage requirements and the CPU time.

The rest of this paper is organized as follows. Section II provides the construction process of the POD/DEIM reduced order model of 2D Burgers equation, including: description of two-dimensional Burgers equation and fully implicit finite difference scheme in section 2.1, and the corresponding Newton iteration method in section 2.2, the POD ROM of 2D Burgers equation and Galerkin projection are presented in section 2.3, as well as discrete empirical interpolation method (DEIM) dealing with the nonlinear functions arising from the nonlinear advection terms in the 2-D Burgers model in section 2.4, while in section 2.5, numerical experiments are performed to verify and validate 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 3, a calibrated LOM 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 better presentation of the regularization closure model.

2 POD/DEIM reduced order model of 2D Burgers equation

Considering the construction of POD or POD/DEIM ROM, some important details about the fully implicit finite difference scheme and its Newton iteration solution method are stressed here, and presented in sections 2.1 and 2.2, respectively.

2.1 Full-order Model of 2D Burgers equation

We consider a two-dimensional nonlinear viscous Burgers equations like this (called **Full Model**):

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = \frac{1}{Re}(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}), \tag{2.1.1a}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = \frac{1}{Re}(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}), \tag{2.1.1b}$$

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

subject to the boundary conditions

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

$$u(x,c,t) = g_{u3}(x,t); \quad u(x,d,t) = g_{u4}(x,t); \tag{2.1.2a}$$
$$v(a,y,t) = g_{v1}(y,t); \quad v(b,y,t) = g_{v2}(y,t);$$
$$v(x,c,t) = g_{v3}(x,t); \quad v(x,d,t) = g_{v4}(x,t); \tag{2.1.2b}$$

and the initial conditions

$$u(x,y,t)|_{t=0} = \varphi(x,y), \tag{2.1.2c}$$
$$v(x,y,t)|_{t=0} = \psi(x,y), \quad (x,y) \in \Omega \tag{2.1.2d}$$

where Re is the Reynolds number, and the $u(x,y,t)$ and $v(x,y,t)$represent the velocity components, respectively. When $2D$ spatial computational domain Ω 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: $x_j = jdx$, $y_i = idy$, $u_{ij} = u(x_i, y_j, t), v_{ij} = v(x_i, y_j, t)$. Here $dx = \frac{b-a}{n_x-1}, dy = \frac{d-c}{n_y-1}$. Then the centered-difference scheme corresponding to the first- or second-order derivatives in space will eventually result in the following form of $2D$ Burgers equation[3]:

$$\frac{du_{i,j}}{dt} + \frac{u_{i+1,j} - u_{i-1,j}}{2dx}u_{i,j} + \frac{u_{i,j+1} - u_{i,j-1}}{2dy}v_{i,j} =$$
$$\frac{1}{Redx^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) + \frac{1}{Redy^2}(u_{i,j+1} - 2u_{i,j} + u_{i,j-1}) \tag{2.1.3a}$$
$$\frac{dv_{i,j}}{dt} + \frac{v_{i+1,j} - v_{i-1,j}}{2dx}u_{i,j} + \frac{u_{i,j+1} - v_{i,j-1}}{2dy}v_{i,j} =$$
$$\frac{1}{Redx^2}(v_{i+1,j} - 2v_{i,j} + v_{i-1,j}) + \frac{1}{Redy^2}(v_{i,j+1} - 2v_{i,j} + v_{i,j-1}) \tag{2.1.3b}$$

which can be cast in matrix form as follows:

$$\frac{dU}{dt} + f_1(U,V) - \frac{1}{2dx}B_{ul}U - \frac{1}{2dy}B_{ub}V =$$
$$\frac{1}{Redx^2}(D_1U + b_{1u}) + \frac{1}{Redy^2}(D_2U + b_{2u}) \tag{2.1.4a}$$
$$\frac{dV}{dt} + f_2(U,V) - \frac{1}{2dx}B_{vl}U - \frac{1}{2dy}B_{vb}V =$$
$$\frac{1}{Redx^2}(D_1V + b_{1v}) + \frac{1}{Redy^2}(D_2V + b_{2v}) \tag{2.1.4b}$$

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 $f_1, f_2 : R^{n_{xy}} \times R^{n_{xy}} \to R^{n_{xy}}$ are then defined as follows:

$$f_1(U, V) = \frac{1}{2dx} MU. * U + \frac{1}{2dy} NU. * V, \tag{2.1.5a}$$

$$f_2(U, V) = \frac{1}{2dx} MV. * U + \frac{1}{2dy} NV. * V, \tag{2.1.5b}$$

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

$$B_{ul} = diag(kron(u(1, 2 : n_y - 1), [1, 0, 0, \cdots, 0]_{1\times(n_x-2)})),$$

$$B_{ub} = diag(kron([1, 0, 0, \cdots, 0]_{1\times(n_y-2)}, u(2 : n_x - 1, 1)),$$

$$b_{1u} = (u_{1,2}, 0, \cdots, 0, u_{5,2}, u_{1,3}, 0, \cdots, u_{5,4})^T,$$

$$b_{2u} = [u(2 : n_x - 1, 1)^T, 0, \cdots, 0, u(2 : n_x - 1, 5)^T]^T,$$

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

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

where E is $(n_y - 2)$-by-$(n_y - 2)$ the identity matrix, and '$kron$' is a Matlab function which means that $K = kron(A, B)$ 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 same expression with B_{ul},B_{ub},b_{1u} and b_{2u}.

2.2 Newton method of Full-order Model

Equations (2.1.4a,b) is just a semi-discretized system (ODEs) of equations (2.1.1a,b). The numerical solution of its can be accomplished by one of the known procedure from the differentiation equation such as Runge-Kutta method etc.. However, when backward Euler scheme in time is performed,

hopefully and practically, there is another way that Newton method is employed [45]. This method converges fast with each subsequent convergence error proportional to the square of its predecessor provided we have a good initial guess. In a time-stepping context, a good guess is always available, that is, the value at the end of the last step. Discretizing the time interval $[0, T]$ into $n_t = \frac{T}{dt}$ equal segments, according to the expression of model (2.1.4a,b), we introduce two maps $F_1, F_2 : R^{n_{xy}} \times R^{n_{xy}} \to R^{n_{xy}}$ to define the relation as follows for the purpose of carrying out the Newton iteration for $U^{n+1} = U(t_n), V^{n+1} = V(t_n)$:

$$F_1(U^{n+1}, V^{n+1}) = 0; \tag{2.2.1a}$$

$$F_2(U^{n+1}, V^{n+1}) = 0; \tag{2.2.1b}$$

where $t_n = ndt$, and the F_1 and F_2 are represented by

$$F_1(U^{n+1}, V^{n+1}) = U^{n+1} - U^n + dtf_1(U^{n+1}, V^{n+1}) - \frac{dt}{2dx}B_{ul}U^{n+1} - \frac{dt}{2dy}B_{ub}V^{n+1} - \frac{dt}{Redx^2}(D_1U^{n+1} + b_{1u}) - \frac{dt}{Redy^2}(D_2U^{n+1} + b_{2u}); \tag{2.2.2a}$$

$$F_2(U^{n+1}, V^{n+1}) = V^{n+1} - V^n + dtf_2(U^{n+1}, V^{n+1}) - \frac{dt}{2dx}B_{vl}U^{n+1} - \frac{dt}{2dy}B_{vb}V^{n+1} - \frac{dt}{Redx^2}(D_1V^{n+1} + b_{1v}) - \frac{dt}{Redy^2}(D_2V^{n+1} + b_{2v}) \tag{2.2.2b}$$

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

$$J^k \begin{pmatrix} \delta U_k^{n+1} \\ \delta V_k^{n+1} \end{pmatrix} = -\begin{pmatrix} F_1 \\ F_2 \end{pmatrix}^k \tag{2.2.3}$$

The coefficient matrix is the Jacobian, which is expressed as

$$J = \begin{pmatrix} F_{1u} & F_{1v} \\ F_{2u} & F_{2v} \end{pmatrix} \tag{2.2.4}$$

where

$$F_{1u} = \frac{\partial F_1}{\partial U^{n+1}} = E + dt(\frac{\partial f_1}{\partial U^{n+1}} - \frac{1}{2dx}B_{ul}\ ) - \frac{dt}{Redx^2}D_1 - \frac{dt}{Redy^2}D_2; \tag{2.2.5}$$

$$F_{1v} = \frac{\partial F_1}{\partial V^{n+1}} = dt(\frac{\partial f_1}{\partial V^{n+1}} - \frac{1}{2dy}B_{ub}); \tag{2.2.6}$$

$$F_{2u} = \frac{\partial F_2}{\partial U^{n+1}} = dt(\frac{\partial f_2}{\partial U^{n+1}} - \frac{1}{2dx}B_{vb}); \tag{2.2.7}$$

$$F_{2v} = \frac{\partial F_1}{\partial V^{n+1}} = E + dt(\frac{\partial f_2}{\partial V^{n+1}} - \frac{1}{2dy}B_{vb}) - \frac{dt}{Redx^2}D_1 - \frac{dt}{Redy^2}D_2; \quad (2.2.8)$$

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

$$\frac{\partial f_1}{\partial U^{n+1}} = \frac{1}{2dx}(diag(MU^{n+1}) + diag(U^{n+1})M) + \frac{1}{2dy}diag(V^{n+1})N; \quad (2.2.9)$$

$$\frac{\partial f_1}{\partial V^{n+1}} = \frac{1}{2dy}diag(NU^{n+1}); \quad (2.2.10)$$

$$\frac{\partial f_2}{\partial U^{n+1}} = \frac{1}{2dx}diag(MV^{n+1}); \quad (2.2.11)$$

$$\frac{\partial f_2}{\partial V^{n+1}} = \frac{1}{2dx}diag(U^{n+1})M + \frac{1}{2dy}(diag(NV^{n+1}) + diag(V^{n+1})N). \quad (2.2.12)$$

By solving equations (2.2.3), the $(U_k^{n+1}, V_k^{n+1})^T$ can be updated like this

$$\begin{pmatrix} U_{k+1}^{n+1} \\ V_{k+1}^{n+1} \end{pmatrix} = \begin{pmatrix} U_k^{n+1} \\ V_k^{n+1} \end{pmatrix} + \begin{pmatrix} \delta U_k^{n+1} \\ \delta V_k^{n+1} \end{pmatrix} \quad (2.2.13)$$

Check if the following conditions are satisfied, or continue the iterative process above.

$$\left\| \begin{pmatrix} \delta U_k^{n+1} \\ \delta V_k^{n+1} \end{pmatrix} \right\| < tol, or \left\| F(U_{k+1}^{n+1}) \right\| < tol, \quad (2.2.14)$$

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

2.3 POD Reduced-order Burgers Equation and Galerkin Projection

POD can be seen as one of the most popular model reduction techniques or as a method for data representation that has been used in data analysis, pattern recognition, optimal control and inverse problem. POD provides a method for finding the best approximating subspace to a given set of data. Originally POD was used as a data represention technique. 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 [13,14,15,16,22,23]. 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.

2.3.1 Snapshot collection and POD basis

As far as the calculation of the POD modes is concerned, let us first give the model variables solution $\Psi = [\Psi_k(\mathbf{x}, t_k)] \in R^{n_{xy} \times n_k}$ (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. At this time, singular value decomposition (SVD) for $\Psi \in R^{n_{xy} \times n_k}$, eigenvalue decomposition for $\Psi\Psi^T \in R^{n_{xy} \times n_{xy}}$ or eigenvalue decomposition for $\Psi^T\Psi \in R^{n_k \times n_k}$ are used to derive the so-called POD basis. In this work, the POD basis vectors for u, v are built from the snapshots of the solution for each variable separately. 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 $\Phi_u = [\Phi_u^i, i = 1, 2, \cdots, m_1] \in R^{n_{xy} \times m_1}$ by solving the eigenvalue problem

$$\Psi^T\Psi\hat{u}_i = \lambda_i\hat{u}_i, i = 1, 2, \cdots, n_k \tag{2.3.1}$$

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 $\Phi_v \in R^{n_{xy} \times m_2}$ 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,

$$I(m) = \frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{n_k} \lambda_i} \tag{2.3.2}$$

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

2.3.2 Construction of POD-ROM of 2D Burgers' Equation with Galerkin Projection

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

$$U(t_n) \approx \Phi^u\alpha(t_n), V(t_n) \approx \Phi^v\beta(t_n), \tag{2.3.3}$$

where $\alpha(t_n) \in R^{m_1}, \beta(t_n) \in R^{m_2}$. Plugging the equations (2.3.3) in the equations (2.1.4) and multiplying the equation (2.1.4a) and (2.1.4b) by Φ_u and Φ_v,

respectively, we can have the POD reduced order model (**POD ROM**)

$$\frac{d\alpha}{dt} + f_1^{POD}(\alpha,\beta) - \frac{1}{2dx}B_{ul}^{POD}\alpha - \frac{1}{2dy}B_{ub}^{POD}\beta = \frac{1}{Redx^2}(D_{11}^{POD}\alpha + b_{1u}^{POD}) + \frac{1}{Redy^2}(D_{12}^{POD}\alpha + b_{2u}^{POD}) \tag{2.3.4a}$$

$$\frac{d\beta}{dt} + f_2^{POD}(\alpha,\beta) - \frac{1}{2dx}B_{vl}^{POD}\alpha - \frac{1}{2dy}B_{vb}^{POD}\beta = \frac{1}{Redx^2}(D_{21}^{POD}\beta + b_{1v}^{POD}) + \frac{1}{Redy^2}(D_{22}^{POD}\beta + b_{2v}^{POD}) \tag{2.3.4b}$$

where

$$f_1^{POD}(\alpha,\beta) = \Phi_u^T f_1(\Phi_u\alpha, \Phi_v\beta), f_2^{POD}(\alpha,\beta) = \Phi_v^T f_2(\Phi_u\alpha, \Phi_v\beta), \tag{2.3.5}$$

$$D_{11}^{POD} = \Phi_u^T D_1\Phi_u, D_{12}^{POD} = \Phi_u^T D_2\Phi_u, D_{11}^{POD} = \Phi_v^T D_1\Phi_v, D_{12}^{POD} = \Phi_v^T D_2\Phi_v,$$
$$B_{ul}^{POD} = \Phi_u^T B_{ul}\Phi_u, B_{ub}^{POD} = \Phi_u^T B_{ub}\Phi_v, b_{1u}^{POD} = \Phi_u^T b_{1u}, b_{2u}^{POD} = \Phi_u^T b_{2u},$$
$$B_{vl}^{POD} = \Phi_v^T B_{vl}\Phi_u, B_{vb}^{POD} = \Phi_v^T B_{ub}\Phi_v, b_{1v}^{POD} = \Phi_v^T b_{1v}, b_{2u}^{POD} = \Phi_v^T b_{2v},$$

All the matrix-matrix multiplications are calculated in an offline-phase. It can be seen that the equations (2.3.4) have the same form as the equations (2.1.4). For comparison purpose, the Newton iteration method is still adopted. But the gradients of nonlinear functions with respect to α, β will be changed as follows:

$$\frac{\partial f_1^{POD}}{\partial\alpha} = \Phi_u^T\frac{1}{2dx}[diag(\Phi_u\alpha)M\Phi_u + diag(M\Phi_u\alpha)\Phi_u] + \Phi_u^T\frac{1}{2dy}diag(\Phi_v\beta)N\Phi_u, \tag{2.3.6}$$

$$\frac{\partial f_1^{POD}}{\partial\beta} = \Phi_u^T\frac{1}{2dy}diag(N\Phi_u\alpha)\Phi_v, \tag{2.3.7}$$

$$\frac{\partial f_2^{POD}}{\partial\beta} = \Phi_v^T\frac{1}{2dx}diag(\Phi_u\alpha)M\Phi_v + \Phi_v^T\frac{1}{2dy}[diag(N\Phi_v\beta)\Phi_v + diag(\Phi_v\beta)N\Phi_v], \tag{2.3.8}$$

$$\frac{\partial f_2^{POD}}{\partial\alpha} = \Phi_v^T\frac{1}{2dx}diag(M\Phi_v\beta)\Phi_u. \tag{2.3.9}$$

A POD reduced order model has been constructed above. It is necessary to note that the nonlinear functions (2.3.5) have to be evaluated online which means that the computational complexity of the reduced order model still depends on the number of unknowns of the **Full Model** which may cause the POD ROM to be inefficient. To overcome this problem, the subsequent consideration is the application of Discrete Empirical Interpolation Method (**DEIM**) to the nonlinear functions (2.3.5).

2.4 The Acceleration of POD-ROM Based on Application of DEIM Approach

Let us now describe briefly the process of application of **DEIM**. Similar to the building of POD basis Φ_u and Φ_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 (2.3.5)such that

$$f_1^{POD} \simeq \underbrace{\Phi_u^T \Xi_{f_1} (P_1^T \Xi_{f_1})^{-1}}_{\Xi^{f_1}} \underbrace{P_1^T f_1(\Phi_u \alpha, \Phi_v \beta)}_{\tilde{f}_1(\alpha,\beta)}, \tag{2.4.1}$$

$$f_2^{POD} \simeq \underbrace{\Phi_v^T \Xi_{f_2} (P_2^T \Xi_{f_2})^{-1}}_{\Xi^{f_2}} \underbrace{P_2^T f_1(\Phi_u \alpha, \Phi_v \beta)}_{\tilde{f}_2(\alpha,\beta)} \tag{2.4.2}$$

where $\Xi_{f_i} \in R^{n_{xy} \times \tau_i}, i = 1,2$ contains the first τ_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 $P_i = [e_{\rho 1}, \cdots, e_{\rho_{\tau_i}}] \in R^{n_{xy} \times \tau_i}, i = 1,2$, 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] for details. By using the fact that $\tilde{f}_1, \tilde{f}_1$ are pointwise, we can calculate them as

$$\tilde{f}_1(\alpha,\beta) := \frac{1}{2dx} P_1^T M \Phi_u \alpha .* (P_1^T \Phi_u \alpha) + \frac{1}{2dy} P_1^T N \Phi_u \alpha .* (P_1^T \Phi_v \beta), \tag{2.4.3a}$$

$$\tilde{f}_2(\alpha,\beta) := \frac{1}{2dx} P_2^T M \Phi_v \beta .* (P_2^T \Phi_u \alpha) + \frac{1}{2dy} P_2^T N \Phi_v \beta .* (P_2^T \Phi_v \beta), \tag{2.4.3b}$$

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

$$\frac{d\alpha}{dt} + \Xi^{f_1} \tilde{f}_1(\alpha,\beta) - \frac{1}{2dx} B_{ul}^{POD} \alpha - \frac{1}{2dy} B_{ub}^{POD} \beta = \frac{1}{Redx^2}(D_{11}^{POD} \alpha + b_{1u}^{POD}) + \frac{1}{Redy^2}(D_{12}^{POD} \alpha + b_{2u}^{POD}) \tag{2.4.4a}$$

$$\frac{d\beta}{dt} + \Xi^{f_2} \tilde{f}_2(\alpha,\beta) - \frac{1}{2dx} B_{vl}^{POD} \alpha - \frac{1}{2dy} B_{vb}^{POD} \beta = \frac{1}{Redx^2}(D_{21}^{POD} \beta + b_{1v}^{POD}) + \frac{1}{Redy^2}(D_{22}^{POD} \beta + b_{2v}^{POD}) \tag{2.4.4b}$$

We see from the equations (2.4.3a,b) that a substantial reduction in computational cost can be expected due to their dependence only on dimensions τ_i instead of the original dimensions n_{xy}.

2.5 The Validation of POD/DEIM ROM

In this section we will provide numerical experiments and aim 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 all 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 $Re = 100$, and $n_x \times n_y = 200 \times 200$, with $n_t = 1000$ in the case of $Re = 1000$. 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 equations [8].

$$u(x,y,t) = \frac{3}{4} - \frac{1}{4[1 + exp((-4x + 4y - t)Re/32)]} \tag{2.5.1a}$$

$$v(x,y,t) = \frac{3}{4} + \frac{1}{4[1 + exp((-4x + 4y - t)Re/32)]} \tag{2.5.1b}$$

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

$$\mathbf{RMSE^n} = \sqrt{\frac{\sum_{i=1}^{N}(U_i^n - U_{0,i}^n)^2}{N}} \tag{2.5.2}$$

$$\mathbf{Corr(U, U_0)^n} = \frac{E(U^n - \mu_U)(U_0^n - \mu_{U_0^n})}{\sigma_{U^n}\sigma_{U_0^n}} \tag{2.5.3}$$

where μ_U, and μ_{U_0} are the given expected value and the standard deviations σ_U, and σ_{U_0}. In addition, the average relative error (**E**)[13] are also provided in the analysis. Its computational formula is defined as

$$\mathbf{E_u} = \frac{1}{nt}\sum_{i=1}^{nt}\frac{\|U^{Full}(:,i) - U^{POD/DEIM}(:,i)\|_2}{\|U^{Full}(:,i)\|_2} \tag{2.5.4}$$

2.5.1 Case of $Re = 100$

The POD basis is 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. As for the selection of 50 DEIM interpolation points, it is only based on results shown in Table 1, and other particular consideration is not involved here.

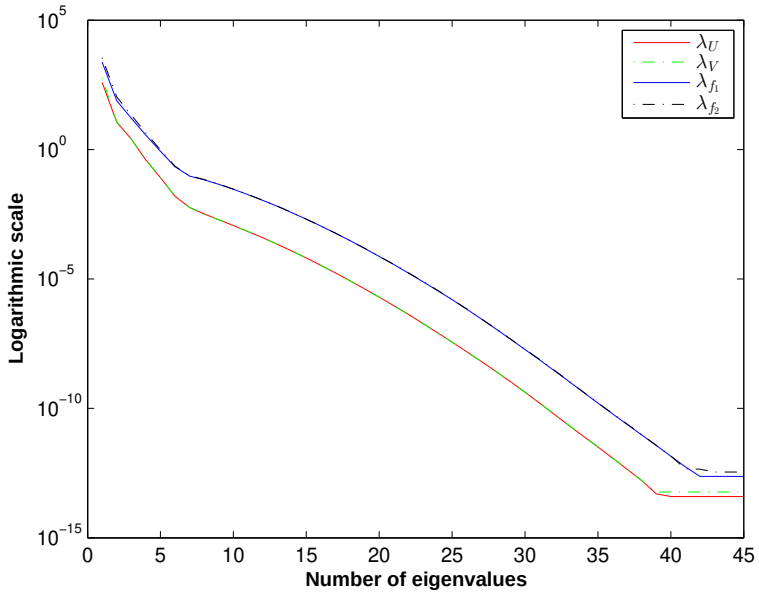

Figure 1: **Eigenvalues of solution snapshots and nonlinear function snapshots**($Re = 100$)

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 60×60 spatial mesh discretization.

DEIM points	10	30	50	60	70	80
Cpu time	1.5706	1.5559	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

From the Table 2, it is observed that the POD ROM fails to decrease complexity since the dependence of nonlinear terms on the size n_{xy} of the original full-order 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 mainly due to the DEIM computation of nonlinear functions f_1, f_2 on the selected 50 DEIM interpolation points, see figure 2.

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
30×30	3.3465	2.3481	1.0890
60×60	12.5944	6.7459	1.4812
90×90	32.5888	13.3619	2.2287
120×120	58.0337	31.6470	3.4297
150×150	96.9749	66.0908	6.9015
200×200	172.7232	149.9842	11.8085

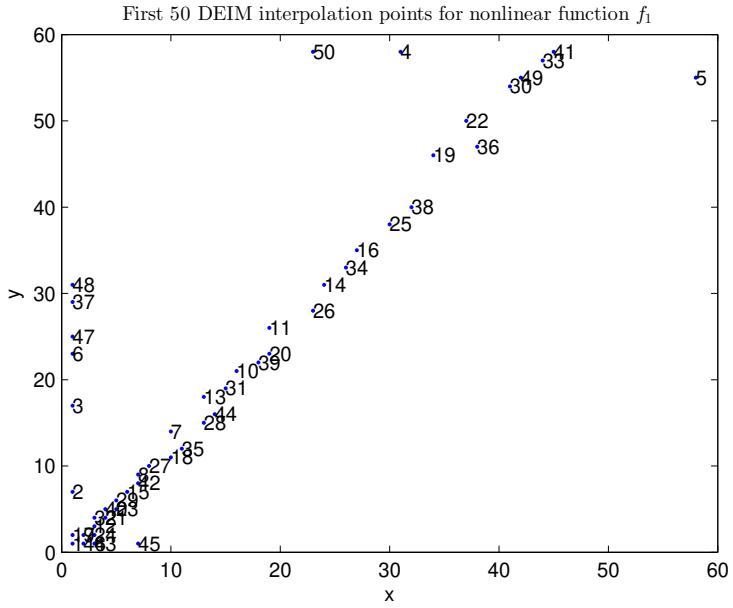

Figure 2: **First 50 DEIM interpolation points for nonlinear function** f_1 **(**$Re = 100$**)**

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 the good RMSE at different time is also observed in Figure 4. In addition, the correlation of u between POD/DEIM50 ROM and Full Model is provided in Figure 5, which 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).

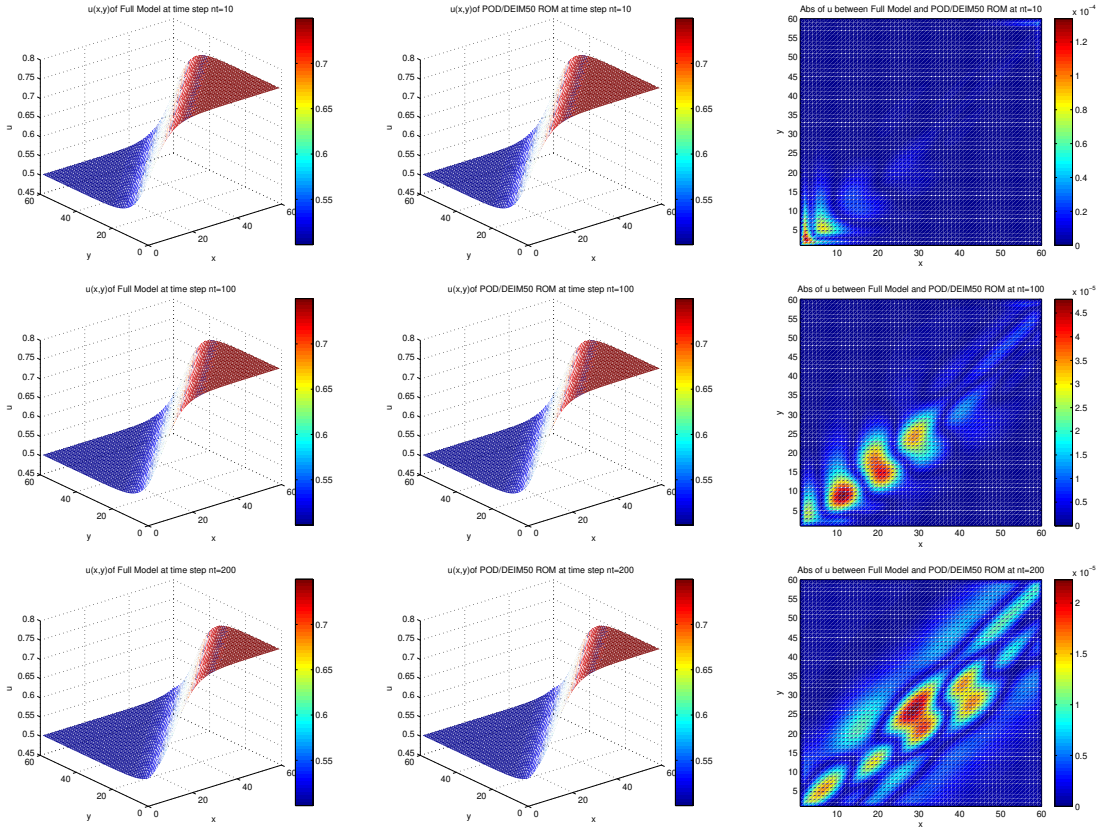

Figure 3: Comparison of u between Full Model and POD5/DEIM50 ROM at nt=10,100 and 200 in the case of $Re = 100$.

2.5.2 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 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, adding the spatial grid points and shrinking the time step size is required to maintain stability of the computation. In the current case, let the spatial grid points mesh be 200×200, and the time step size is 0.001. Though $I = 99.8\%$, we find that additional number of POD bases are chosen, and DEIM interpolation points need also to be increased according to the requirement of our computation. Considering a good 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 keeps the advantage with the computational time reduced by factor of $O(10)$, see

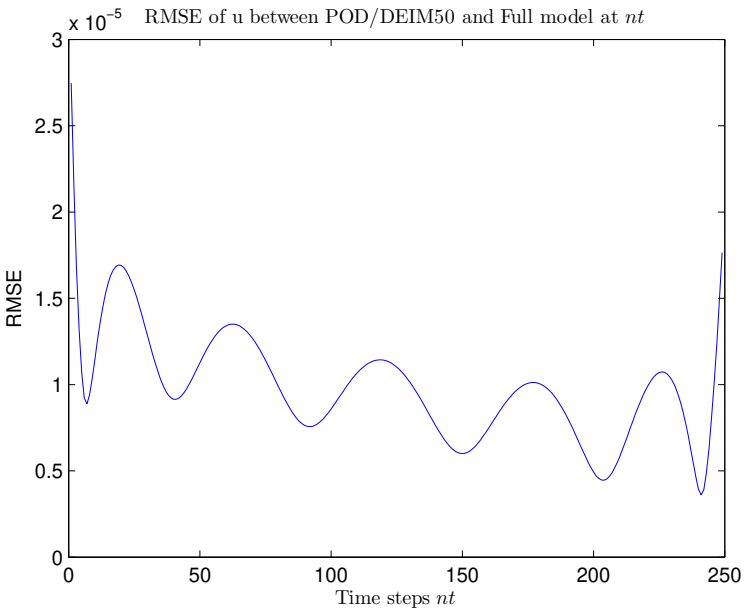

Figure 4: **RMSE of u between Full Model and POD15/DEIM250 ROM at each time nt, ($Re = 100$)**

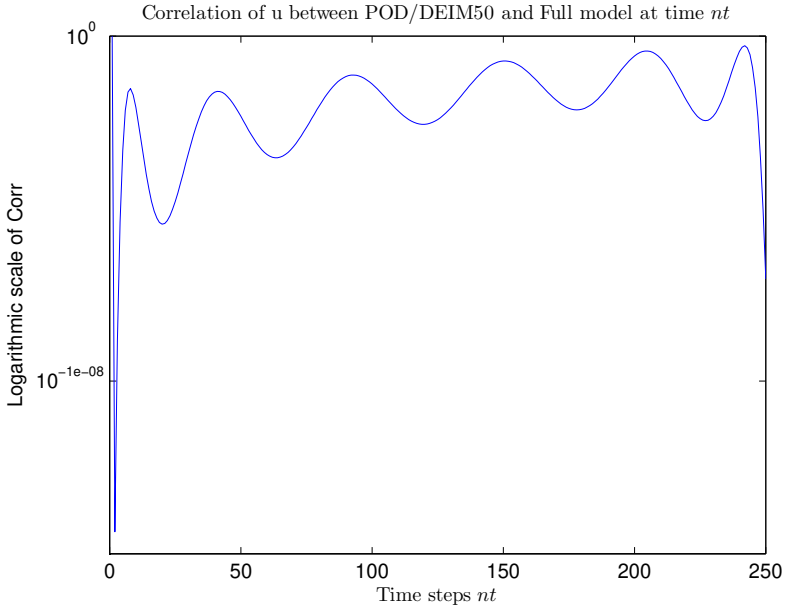

Figure 5: **Correlation of u between Full Model and POD15/DEIM250 ROM at each time nt, ($Re = 100$)**

Table 4. It implies that the POD scheme is not able to really reduce the computational complexity for nonlinear systems but on the contrary, it increases

sometimes the computational cost in some sense. 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 Re, see Figure 7.

Table 3. The CPU time and the average relative errors of POD ROM at different modes for a 200×200 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 200×200 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

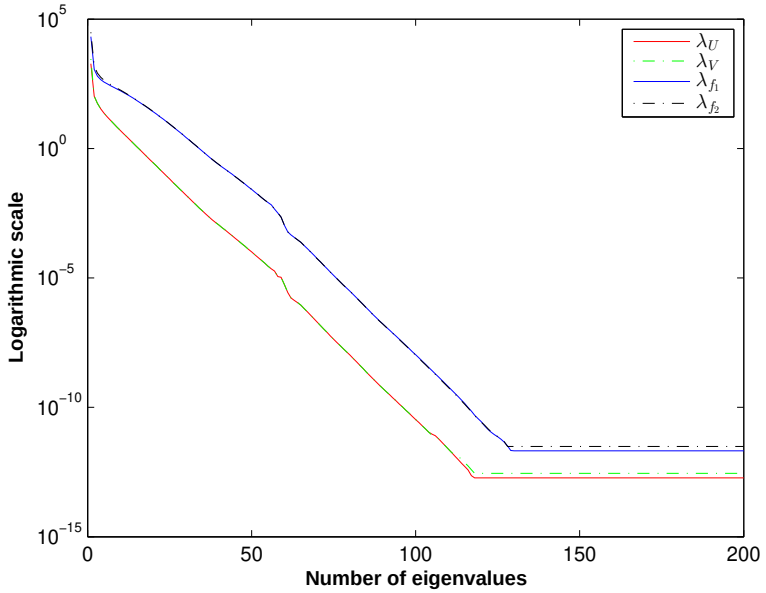

Figure 6: **Eigenvalues of solution snapshots and nonlinear function snapshots** ($Re = 1000$)

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

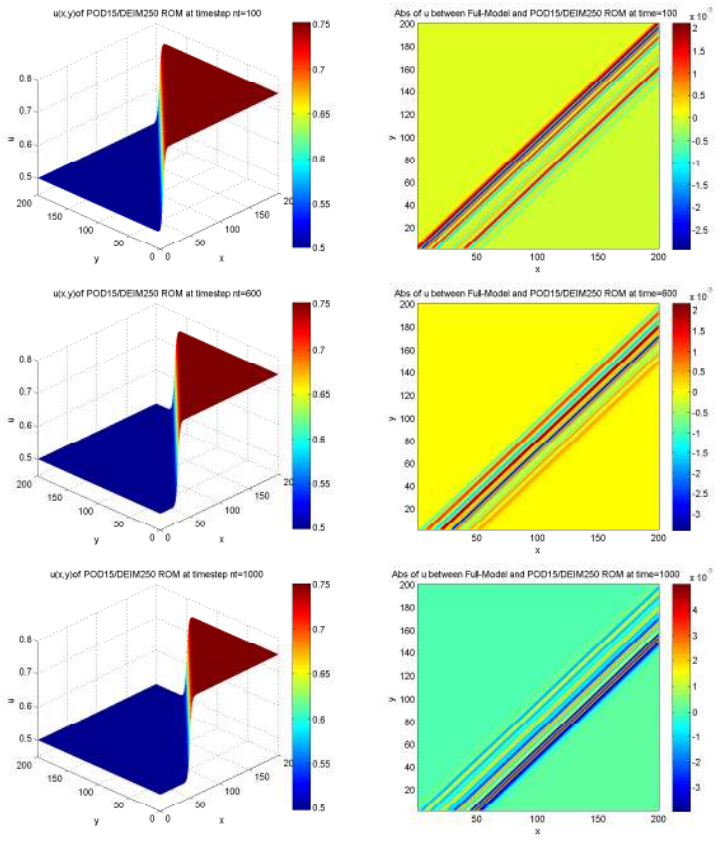

Figure 7: Comparison of u between Full Model and POD15/DEIM250 ROM at nt=100,600 and 1000 in the case of $Re = 1000$.

(1). the POD/DEIM ROM can retain the same reduction of computational time by 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 distributed on the neglected POD state modes and POD nonlinear modes used for DEIM;

(2). the numerical experiment reminds us that more points are beneficial when determining the DEIM interpolation points. As for the optimal selection of POD modes, it is not sufficient to depend only on equation (2.3.2) since slower decaying of eigenvalue spectrum for large Re can be observed by comparing Figure 1 with Figure 6. It suggests that a modest number of POD modes are required for a computational result with higher accuracy. Certainly, this will inevitably increase the computational burden. How to provide an optimal truncation of POD modes remains an open problem [28].

(3). When the number of POD modes is increased up to 15, some spurious oscillations can be found near the steepened front, see Figure 7, due to the

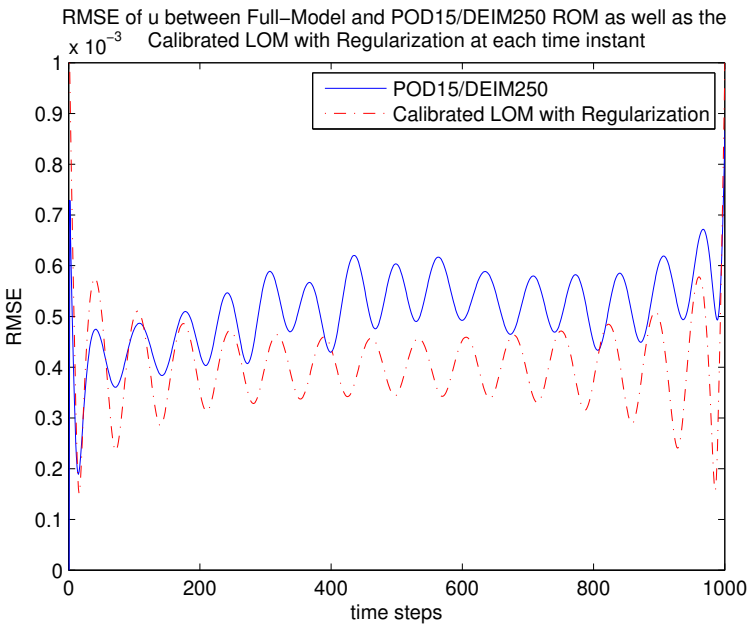

Figure 8: **RMSE of u between Full Model and POD15/DEIM250 ROM as well as the Calibrated LOM with Regularization at each time instant nt, ($Re = 1000$)**

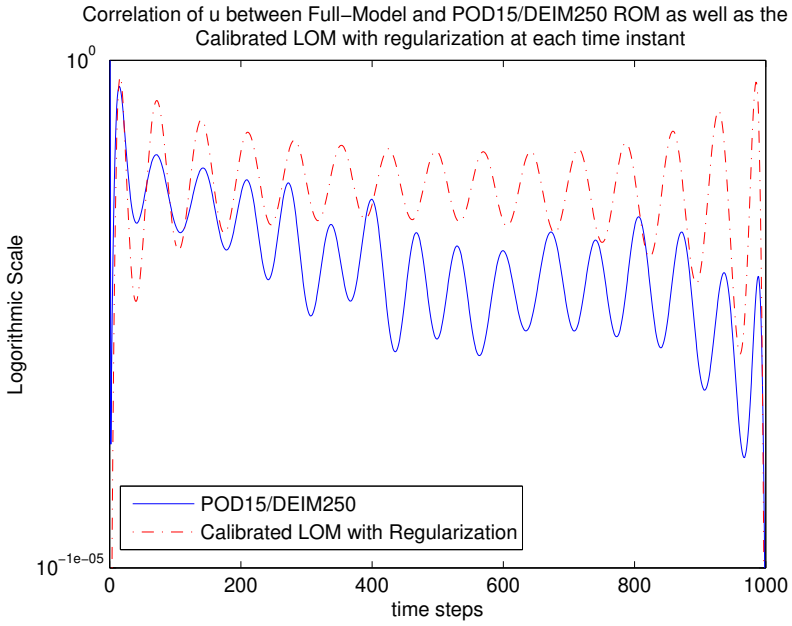

Figure 9: **Correlation of u between Full Model and POD15/DEIM250 ROM as well as the Calibrated LOM with Regularization at each time instant nt, ($Re = 1000$)**

truncation of POD modes and DEIM interpolation of nonlinear term, which cannot be neglected for a longer time integration interval. This leads us to consider recovery of the lost information by calibrating the evolutionary coefficients.

2.6 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 [30,37], there still exists a major barrier for POD Galerkin approach at high Reynolds numbers due to lack of turbulence closure. As carefully explained in [28], 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 it using a closure model consisting of Tikhonov regularization to account for the small scale dissipation effect of the discarded POD modes. (As recently used in [37]). So it is first assumed that the calibrated ROM can be expressed as follows:

$$\dot{a}(t) = f(y, a(t)) \tag{2.6.1}$$

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:

$$f^{(k)}(y, a(t)) = c_k + \sum_{i=1}^{m} l_{ik} a_i + \sum_{i=1}^{m} \sum_{j=1}^{m} a_i D_{ij} a_j \tag{2.6.2}$$

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 a error function,

$$e(y, t) = \dot{a}(t) - f(y, a(t)) \tag{2.6.3}$$

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

$$\begin{aligned} J(y) &= \left\langle \|e(y,t)\|_2^2 \right\rangle_{T_0} + \gamma^2 \|Ly\|_2^2 \\ &= \frac{1}{N} \sum_{k=1}^{N} \sum_{i=1}^{m} \left(\dot{a}_i(t_k) - f^{(i)}(y, a(t_k)) \right)^2 + \gamma^2 \|Ly\|_2^2, \end{aligned} \tag{2.6.4}$$

where the γ is regularization parameter, $L = I$ and $\|\cdot\|_2$ stands for 2-norm (or Euclidean length). The minimization of the function (2.6.4) amounts to solving the linear system

$$(A^T A + \gamma^2 I)y = A^T b \tag{2.6.5}$$

where I is the identity matrix of order N_y, for details of A and b, please refer to [37]. Details of the Tikhonov regularization procedure are based on the several considerations:

(1). the minimization problem (2.6.4) is not well conditioned if $\gamma = 0$. That is clearly seen when the solution to the linear system (2.6.5) is written in the following form using the singular value decomposition (SVD)of A:

$$y = \Sigma_{i=1}^{m} \frac{\sigma_i^2}{\sigma_i^2 + \gamma^2} \frac{u_i^T b}{\sigma_i} v_i \tag{2.6.6}$$

here u and v are left (right) singular vectors, σ 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 [47,48,49] is only partially satisfied, which will lead to an amplification of noise included in b. The quality of the solution is thus greatly effected. 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 γ is necessary.

(2). The key to the success of Tikhonov regularization technique is related to the computation of the regularization parameter γ. To do so, the L-curve method implemented in the package REGULARIZATION TOOLS [50] 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 $Re = 1000$, γ 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 (2.6.5) into the equation (2.6.1) yields a determined calibrated model (called calibrated ROM). Subject to the initial condition

$$a_\alpha|_{t=0} = \Phi_u^T U\big|_{t=0}, \; a_\beta|_{t=0} = \Phi_v^T V\big|_{t=0} \tag{2.6.7}$$

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 and 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 [28]. 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 given the snapshots and its SVD result.

3 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. Its feasibility has been examined. The main goal was to assess the effect of large Re when using 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. The five POD modes in the case of $Re = 100$ are used to derive the POD ROM which can capture more than 99.8% of the system energy, and fifty DEIM interpolation points are selected to use for the interpolation of nonlinear functions, which leads to a great 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 discarded 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 more computational cost. Thirdly, how to provide an optimal truncation of POD modes remains an open problem. It is not sufficient to depend only on the criterion (2.3.2). As for the DEIM, more interpolation points prove to be beneficial in terms of CPU speed-up. Fourth, larger Re incurs bigger 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 with large computational gain (CPU

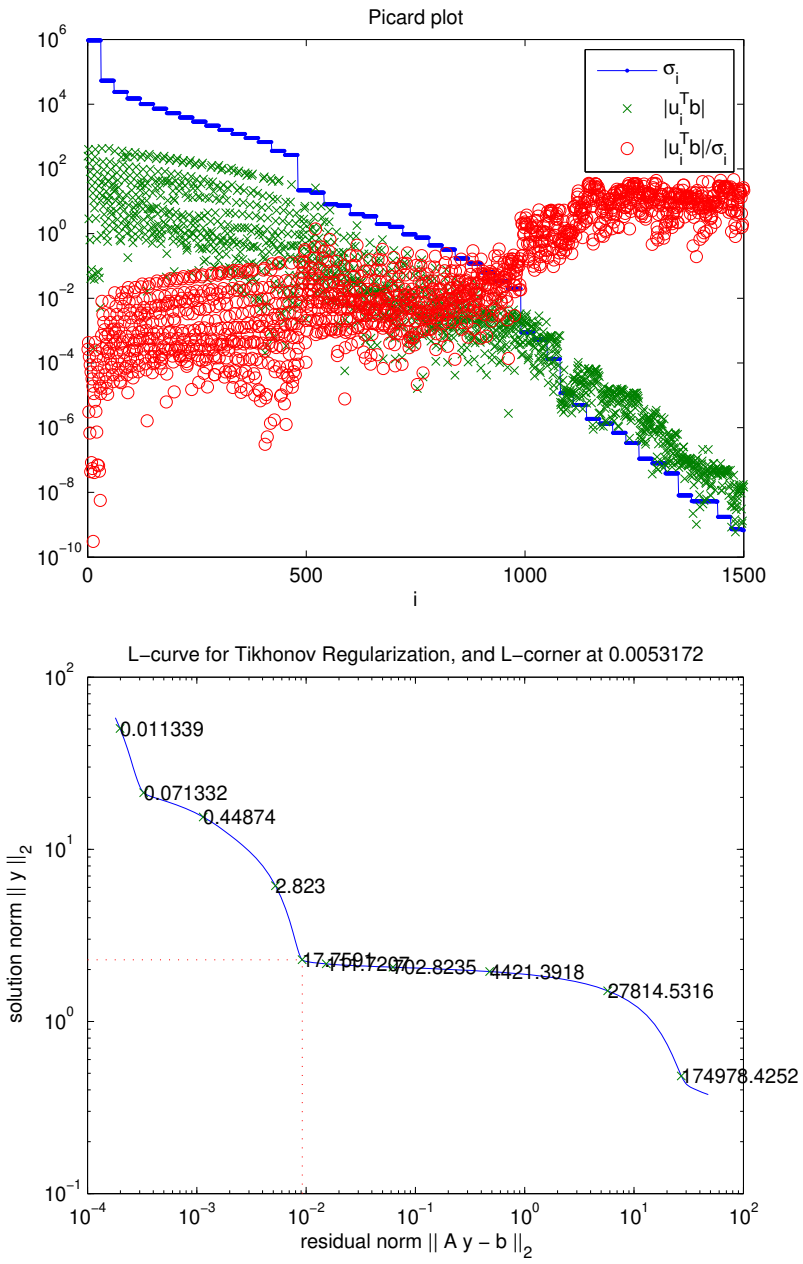

Figure 10: Check of the Picard condition and L-curve

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 Re, 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.

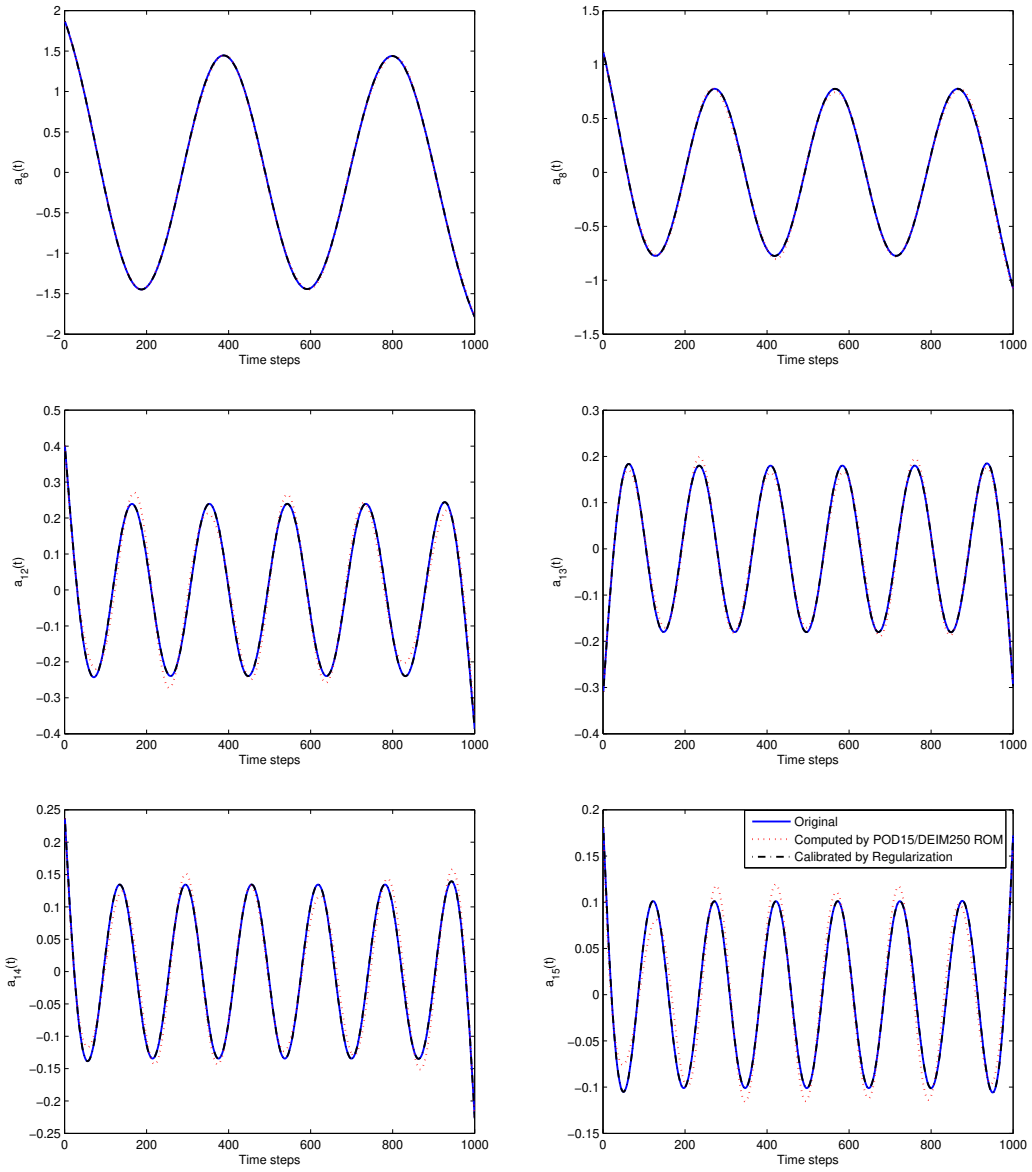

Figure 11: Comparison of the temporal coefficients before and after calibration.

Acknowledgement

This work reported here is supported by the National Natural Science Foundation of China (Grant No.41375115). 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 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

$$Kx = y \tag{A.1}$$

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

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

where K is a compact operator between the real Hilbert spaces X and Y, and (σ_i, u_i, v_i) is a singular system of K. The infinite sum in (A.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 σ_i for $i = 1, 2, \cdots$ [47].

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

$$Kx = y \tag{A.3}$$

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 square solution of the linear equation (A.3)[47] is given by

$$x_{ols} = K^\dagger y = V\Sigma^\dagger U^T y = \Sigma_{i=1}^{p} \frac{u_i^T y}{\sigma_i} v_i \tag{A.4}$$

From the relation (A.4), we observe that due to the presence of a very small σ_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.

$$y^\delta = y + \beta u_i^* \tag{A.5}$$

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

$$x_{ols}^\delta = x + \frac{\beta}{\sigma_i^*} v_i^* \tag{A.6}$$

so the following relation results in [47,48]

$$\frac{\|x_{ols}^\delta - x\|_2}{\|y^\delta - y\|_2} = \frac{1}{\sigma_i^*} \tag{A.7}$$

if σ_i^* is very small, the computed solution by (A.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 the 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.4), the discrete version of the Picard condition is therefore introduced. This was pointed out by Per Christian Hansen in the literature [49]. Let τ denote the level at which the computed singular values σ_i level off due to rounding errors, The discrete Picard condition is satisfied if, for all singular values larger than τ, the corresponding Fourier coefficients $\mid u_i^T y \mid$, on average, decay faster than the σ_i .The discrete Picard condition will play an important role in dealing with the discrete ill-posed problems to insure solution stability. We note that it is 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, then the Picard condition is not satisfied, for example, the case shown in this paper, see figure 10. The violation of the Picard condition may also be regarded as an explanation of the instability for the linear inverse problem under consideration.

At the same time, the insight we gain from a study of the quantities associated with the SVE gives a hint on how to deal with noisy problems that violate the Picard condition, that is, we need regularization methods that can compute less sensitive approximations to exact solution. Our hope is to filter out the unwanted part of (A.4). In view of the similarity of two representative method: TSVD and Tikhonov regularization[49], we only take Tikhonov regularization method as example to make an interpretation.

The Tikhonov solution x_{ols}^{λ} is defined as the solution to the problem

$$min\{\|Kx - y\|_2^2 + \lambda^2 \|x\|_2^2\} \tag{A.8}$$

The first term measures the goodness-of-fit, and the second term measures the regularity of the solution. The regularization parameter λ controls the weighting between the two terms. The hope is therefore that if we 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.8)

$$x_{ols}^{\lambda} = \Sigma_{i=1}^{p} f_i^{\lambda} \frac{u_i^T y}{\sigma_i} v_i \tag{A.9}$$

Comparing the (A.4) and (A.9), we see that the filter factors are introduced here for $i = 1, 2, \cdots, p$, which satisfy

$$f_i^{\lambda} = \frac{\sigma_i^2}{\sigma_i^2 + \lambda^2} = \begin{cases} 1 & \sigma_i \gg \lambda \\ \frac{\sigma_i^2}{\lambda^2} & \sigma_i \ll \lambda \end{cases} \tag{A.10}$$

We stress here that the selection of λ is very crucial for the regularization process. The method to determine it in the current paper is L-curve method. For the details, please see [47,48,49].

References

[1] Paul Arminjon, Claude Beauchamp, Numerical solution of Burgers' equations 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 Burgers equation, *J. Comput. Appl. Math*, 231(2009): 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 Mani 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] Hongqing Zhu, Huazhong Shu and Meiyu Ding, Numerical solutions of two-dimensional Burgers' equations by discrete Adomian decomposition method, *Computers and Mathematics with Applications*, 2010, (60): 840-848.

[7] P.Arminjon, C.Beauchamp, Numerical solution of Burgers equations in two space dimensions,*Comput. Meth. Appl. Mech. Eng.*, 1979, 19:351-365.

[8] C.A.J. Fletcher, Generating exact solutions of the two-dimensional Burgers equation, *Int. J. Numer. Meth. Fluids*, 1983 (3): 213C216.

[9] Nejib Smaoui, Boundary and distributed control of the viscous Burgers equation, *Journal of Computational and Applied Mathematics*, 2005, 182: 91-104.

[10] H.M.Park, Y.D.Jang, Control of Burgers equation by means of mode reduction,*International Journal of Engineering Science*, 2000, 38: 785-805.

[11] 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.

[12] 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.

[13] Razvan Stefanescu, Adrian Sandu and Ionel M. Navon, Comparison of POD reduced order strategies for the nonlinear 2D shallow water equations, *Int. J. Numer. Meth. Fluids*, 2014, 76:497-521.

[14] 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.

[15] S.Ravindran, A reduced-order approach for optimal control of fluids using proper orthogonal decomposition,*Int. J. Numer. Meth. Fluids*, 2000, 34: 425-448.

[16] S.Ravindran, Reduced-order adaptive controllers for fluid flows using POD,*J.Sci, Comput.*, 2000, 15(4): 457-478.

[17] Loève M., Probability Theory. Van Nostrand: Princeton, NJ, 1955.

[18] D.H.Chambers, R.J.Adrian, P.Moin, D.S.Stewart, and H.J.Sung, Karhunen-Loeve expansion of Burgers model of turbulence,*Phys. Fluid*, 1988; 31 (9): 2573-2582.

[19] Hotelling H. Analysis of a complex of statistical variables with principal components. *Journal of Educational Psychology*, 1933; 24:417-441.

[20] Lorenz EN. Empirical orthogonal functions and statistical weather prediction. Technical Report, Massachusetts Institute of Technology, Dept. of Meteorology: Cambridge, MA, 1956.

[21] R.C.Gonzalez, P.A. Wintz, Digital Image Processing, Addison-Wesley, Reading, MA, 1987.

[22] Chaturantabut S. Dimension Reduction for Unsteady Nonlinear Partial Differential Equations via Empirical Interpolation Methods. Technical Report TR09-38, CAAM, Rice University: Houston, TX, 2008.

[23] Chaturantabut S, Sorensen DC. A state space error estimate for POD-DEIM nonlinear model reduction.*SIAM Journal on Numerical Analysis*, 2012; 50(1):46-63.

[24] Chaturantabut S, Sorensen DC. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing* , 2010; 32(5):2737-2764.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] M. Couplet, C. Basdevant, P. Sagaut, Calibrated reduced-order POD-Galerkin system for fluid flow modelling, *Journal of Computational Physics*, 2005, 207: 192-220.

[31] M. Bergmann, C. H. Bruneau, A. Iollo, Enablers for robust POD models,*Journal of Computational Physics*, 2009, 228: 516-538.

[32] 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.

[33] 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.

[34] Omer San and Traian 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.

[35] Laurent Perret, Erwan Collin and J. Delville, Polynomial identification of POD based low-order dynamical system, *Journal of Turbulence* , 2006, 7(17): 1-15.

[36] V.L. Kalb, A.E. Deane, An intrinsic stabilization scheme for proper orthogonal decomposition based low-dimensional models, *Phys. Fluids*, 2007, 19: 054106.

[37] 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.

[38] Bernardo Galletti, Alessandro Bottaro, Charles-Henri Bruneau, Angelo Iollo, Accurate model reduction of transient and forced wakes,*European Journal of Mechanics B/Fluids*, 2007 (26) : 354-366.

[39] H.M.Park, Y.D.Jang, Control of Burgers equation by means of mode reduction, *International Journal of Engineering Science*, 2007, 38: 785-805.

[40] Mehmet Onder Efe, Hitay Ozbay, Low dimensional modelling and Dirichlet boundary controller design for Burgers equation, *Int. J. Control*, 2004, 77 (10): 895-906.

[41] Samir Sahyoun, Seddik 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.

[42] 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.

[43] Hung V. Ly, Hien T. Tran, Modeling and Control of Physical Processes Using Proper Orthogonal Decomposition, *Mathematical and Computer Modelling*, 2001, 33 : 223-236

[44] Masaaki Sugihara, Seiji 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] Philip Holmes, John L. Lumley, Gahl Berkooz,Clarencew Rowley, Turbulence, Coherent Structures,Dynamical Systems and Symmetry (2nd edition), Cambridge University Press, 2012.

[47] Adrian Doicu, Thomas Trautmann, and Franz Schreier, Numerical Regularization for Atmospheric Inverse Problems, Springer-Verlag Berlin Heidelberg, 2010.

[48] Hansen, Per Christian, Discrete inverse problems: insight and algorithms, SIAM, Philadelphia, 2010.

[49] Hansen, Per Christian, The discrete Picard condition for discrete ill-posed problems, BIT, 1990, 30(4): 658-672.

[50] Hansen PC. Regularization tools: a Matlab package for analysis and solution of discrete ill-posed problems. *Numerical Algorithms*, 1994, 6: 1-35.