University of Nairobi

Department of Mathematics

ISSN: 2410-1397

Master Project in Applied Mathematics

Model reduction for partial differential equations

Research Report in Mathematics, Number 01, 2022

Jane Ndinda Ndambuki

June 2022

Submitted to the Department of Mathematics in partial fulfilment for a degree in Master of Science in Applied Mathematics

Model reduction for partial differential equations

Research Report in Mathematics, Number 01, 2022

Jane Ndinda Ndambuki

Department of Mathematics
Faculty of Science and Technology
Chiromo, off Riverside Drive
30197-00100 Nairobi, Kenya

Master Thesis

Submitted to the Department of Mathematics in partial fulfilment for a degree in Master of Science in Applied Mathematics

Submitted to: The Graduate School, University of Nairobi, Kenya

Abstract

This research project considers Model Order Reduction (MOR) techniques known as Proper Orthogonal Decomposition (POD) method and Discrete Empirical Interpolation Method (DEIM) for Partial Differential Equations (PDEs). First, Proper Orthogonal Decomposition is used to formulate a low dimensional basis that can preserve the dynamics of the system. Then, POD-Galerkin approach is employed to obtain a reduced-order model. However, POD method is not efficient when dealing with nonlinear systems and therefore DEIM is used to minimize the computational complexity of the nonlinear term. We will apply POD and DEIM to estimate solutions of high dimensional dynamical systems that arise from finite difference discretization of PDEs. Practically, we will apply POD-DEIM approach to Fisher's equation and POD method to Diffusion-advection equation.

Master Thesis in Mathematics at the University of Nairobi, Kenya.
ISSN 2410-1397: Research Report in Mathematics

DISTRIBUTOR: Department of Mathematics, University of Nairobi, Kenya

Declaration and Approval

I the undersigned declare that this dissertation is my original work and to the best of my knowledge, it has not been submitted in support of an award of a degree in any other university or institution of learning.

09/09/2022

Signature Date

JANE NDINDA NDAMBUKI
Reg No. I56/38757/2020

In my capacity as a supervisor of the candidate's dissertation, I certify that this dissertation has my approval for submission.

09/09/2022

Signature Date

Dr Victor Ogesa Juma
Department of Mathematics,
University of Nairobi,
Box 30197, 00100 Nairobi, Kenya.
E-mail: vjuma23@uonbi.ac.ke

09/09/2022

Signature Date

Dr James Katende
Department of Mathematics
University of Nairobi,
Box 30197, 00100 Nairobi, Kenya.
E-mail: jkatende@uonbi.ac.ke

Dedication

This project is dedicated to my son, Mark Ryan.

Contents

Acknowledgments

First, I thank Almighty God for guiding me throughout this research project. His grace has been sufficient. Further, I would like to thank the University of Nairobi for offering me a chance to study for my masters degree and the International Science Programme (ISP) of Uppsala University, Sweden for granting me a scholarship to pursue this degree.

I gratefully acknowledge my supervisors Dr Victor Ogesa Juma and Dr James Katende for their willingness to offer helpful insights, comments and suggestions on my research project.
I am indebted to Professor Stephen Luketero for his undeviating support and guidance.

My special gratitude goes to my family, specifically my parents, for their endless encouragement, warm support and unconditional love.

Finally, I thank all my friends and classmates for making my time in campus quite enjoyable at times.

Jane Ndinda Ndambuki

Nairobi, 2022.

Notations

0.1 Variables and symbols

Y^* conjugate transpose of matrix Y
$u*$ conjugate transpose of vector u
Y_{kl} Element at the k^{th} row and l^{th} column of matrix Y
$\|u\|_F$ Frobenious norm of a vector :$\sqrt{u^*u}$, where $u \in \mathbb{C}^n$
$\|u\|_W$ Weighted norm: $\|u^*Wu\|$, where $u \in \mathbb{C}^n, W \in \mathbb{C}^{n\times n}$
$\langle u, \bar{u} \rangle$ inner product between two vectors: $u^*\bar{u} = \bar{u}^*u$, where $u, \bar{u} \in \mathbb{C}^n$
$\langle u, \bar{u} \rangle_W$ Weighted inner product between two vectors: $u^*W\bar{u} = \bar{u}^*Wu$, where $u, \bar{u} \in \mathbb{C}^n$, $W \in \mathbb{C}^{n\times n}$

0.2 Acronyms

POD Proper Orthogonal Decomposition
ROM Reduced Order Model
DEIM Discrete Empirical Interpolation Method
PDE Partial Differential Equations
SVD Singular Value Decomposition
ODE Ordinary Differential Equations
MOR Model Order Reduction
FE Finite Element
FD Finite Difference

1 Introduction

Partial Differential Equations are ubiquitous in everyday phenomena to describe biomedical, physical, chemical, engineering and technical processes. Partial Differential Equation (PDE) problems exhibit complex dynamics stemming from nonlinearities and instabilities. For this reason, numerical methods for PDEs like Finite Element (FE), Finite Volume (FV) and Finite Difference (FD) methods are essential in studying numerical solutions of PDEs in order to reduce computational costs and time. However, these numerical methods require the domain to be partitioned into meshes resulting to many degrees of unknowns, that corresponds to the number of nodes of mesh division [14]. This requires a lot of CPU time on a powerful computer which poses an outstanding challenge on these classical numerical approximation techniques. Actually, these methods demand huge computational efforts incase accuracy is necessary.

We want to reduce this computational complexity by removing unnecessary degrees of unknowns. Therefore, model reduction technique that is able to minimize computational time and memory capacity is suitable in order to boost computing efficiency. We can achieve this speed up by trading off the model accuracy for reduced computational complexity. Reduced order models describe a dynamical system by a *smaller* representation such that main characteristics of the system are conserved with adequate correctness. In this research project, model reduction technique named Proper Orthogonal Decomposition(POD) method has been extensively adopted .

Proper Orthogonal Decomposition is a technique used to provide low order models for Ordinary differential equations. This is attained by projecting the full-order system to subspaces that consists of optimal basis elements containing important features of the expected solution. This is in contrary to numerical methods whereby the elements are uncorrelated to the physical properties of the dynamical system they approximate. The POD method extracts basis functions that are used in construction of galerkin projection leading to low order models with fewer degrees of unknowns.

In spite of that, POD method may not reduce the simulation time for solving nonlinear dynamical system because the complexity of nonlinear term still rely on the high dimensionality of the original system. Thus, there is need to consider additional nonlinear model reduction technique. We will focus on Discrete Empirical Interpolation Method

(DEIM) because it can minimize complexity of general nonlinear terms. DEIM is an improvement of the POD algorithm. It estimates nonlinear term by finding basis from POD method and selecting the interpolation indices by a greedy algorithm. Goal of DEIM is to choose the interpolation indices by trying to minimize the error empirically.

1.1 Objectives

The main goal of this research project is to reduce computational complexity of PDEs.

Specific objectives are

- To study and understand Proper Orthogonal Decomposition method.
- To study and understand Discrete Empirical Interpolation Method.
- Find POD modes and POD-DEIM modes that preserve the dominant characteristics of the original system.
- Reduce relative average error in relation to the full order model.

1.2 Literature Review

In 1901, Karl Pearson presented eigenvector analysis method which was used to remove essential components of *big data*. During same period, Karl Pearson introduced Proper Orthogonal Decomposition(POD) method, as its successor. Sample analysis, Pearson's data mining and data processing techniques are appropriate upto date [9]. POD method is identified by different names in other scientific fields, namely; Karhunen-Loève decomposition, Singular systems analysis, Principal Component Analysis and Singular Value Decomposition . Over the years, POD method was introduced by numerous people independently, such as; Kosambi (1943), Loève (1945) , Karhunen (1946) , Pougachev (1953) and Obukhov (1954). The Proper Orthogonal Decomposition method was introduced in fluid dynamics, particularly turbulence, by Lumbley in 1967 to identify coherent structures in turbulent flows [3].

Later on, Sirovich presented the method of snapshot of POD in 1987. The Proper Orthogonal Decomposition method has been widely adopted in different fields such as optimal control, signal analysis and pattern recognition, computational fluid dynamics, statistics and biomedical engineering. After 1987, the POD method was basically applied in statistical computations to carry out principal component analysis. In 2001, Kunisch and Volkweind proposed an excellent work on POD-Galerkin method for PDEs where it was

used to find certain major behavior of dynamical systems [2]. Since then, this model reduction technique known as POD has been widely used in computing numerical solutions for PDEs, laying out enhanced efficiency as compared to numerical computational methods [4].

Initially, Kunisch and Volkweind derived error estimates for a POD reduced systems of numerical solutions of nonlinear parabolic PDEs, whereby those error estimates contained uncertain matrix norms. Particularly, the numerical solutions of the Galerkin method defined on time span [0, T] were used to formulate the POD reduced systems and recompute the numerical solutions of the PDEs using the similar time span [0, T]. Zhendong Luo developed interest on POD for PDEs in the year 2003 but only incomplete introductions about POD were available. In 2006, Zhendong Luo and his coauthors published their POD method first papers in which they discussed oceanic models and data assimilation [5]. Later, they were able to establish some POD reduced-order FD schemes and FE formulations. Since 2007, they derived the error approximations for solutions of POD reduced system for PDEs of numerous kinds [8].

POD method faced a big challenge even though it was promising at first due to the nonlinear nature of some systems. In 2010, Serenson and Chaturantabut developed the Discrete Empirical Interpolation Method (DEIM) which can deal with models with nonlinear terms in order to reduce computational complexity [16]. DEIM has been used in many applications, such as 2-D shallow water equations, Navier-Stokes equations and four dimensional variational data assimilation [17]. In 2018, Norapon Sukuntee and Saifon Chaturantatut used POD in conjunction with DEIM to study Sine-Gordon equation. They investigated the end result of using different snapshots for formulating the POD basis from discretization of the equation and applied the POD-DEIM method to predict numerical solution of the sine-Gordon equation [18].

Type text here

2 Preliminaries

In this chapter, we will discuss some important notions that will be used throughout this project. Particularly, we will discuss Singular Value Decomposition, low rank matrix approximation, dynamical system and reduced order models.

2.1 Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) is a unique matrix decomposition used to obtain low-rank approximations to matrices.
Let $Y = [y_1, y_2, \ldots, y_m]$ be a $n \times m$ complex valued matrix with rank $r \leq \min\{n, m\}$ and $n \geq m$. The columns of the matrix $y_l \in \mathbb{C}^n$, $1 \leq l \leq m$, are known as snapshots while m represents number of snapshots in Y.
Thus, Singular Value Decomposition (SVD) factorizes matrix Y into a product of other three matrices. That is,

$$Y = U\Sigma V^* \tag{2.1.1}$$

whereby $U \in \mathbb{C}^{n\times n}$ and $V \in \mathbb{C}^{m\times m}$ are unitary matrices with orthonormal columns $\{u_l\}_{l=1}^n$ and $\{v_l\}_{l=1}^m$ respectively and

$$\Sigma = \begin{bmatrix} \tilde{\Sigma} & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{n\times m}$$

where $\tilde{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \ldots \sigma_r) \in \mathbb{C}^{r\times r}$ with $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_r \geq 0$ which are known as singular values. Here, adjoint * denotes the complex conjugate transpose.

The vectors $\{v_k\}_{k=1}^r$ and $\{u_k\}_{k=1}^r$ are called eigenvectors of Y^*Y and YY^* respectively with corresponding eigenvalues $\lambda_k = \sigma_k^2 > 0$, $k = 1, 2, \ldots r$ and satisfy the relation

$$Yv_k = \sigma_k u_k \quad \text{and} \quad Y^*u_k = \sigma_k v_k, \quad k = 1, 2, \ldots r.$$

Using the fact that Σ has decreasing and non-negative diagonal entries, it is possible to cut out some rows and columns, truncating the singular vectors corresponding to small singular values. By doing this, we are able to get the best possible rank-r approximation to Y.

It follows that equation (2.1.1) can be written as

$$Y \approx \tilde{U}\tilde{\Sigma}\tilde{V}^* \tag{2.1.2}$$

whereby matrices $\tilde{U} \in \mathbb{C}^{n\times r}$ and $\tilde{V} \in \mathbb{C}^{m\times r}$ denote the truncated matrices formed by choosing the leading r columns of U and V matrices respectively.

Letting $\tilde{Y}$ = $\tilde{\Sigma}\tilde{V}^* \in \mathbb{C}^{r\times m}$, equation (2.1.2) can be expressed in the form

$$Y = \tilde{U}\tilde{Y} \tag{2.1.3}$$

This infers that the space spanned by columns of matrix Y can be expressed interms of r columns of matrix $\tilde{U}$ that are linearly independent. Hence, the expansion of the columns $y_l, l = 1,2,\ldots m$, with respect to the basis $\{u_k\}_{k=1}^{r}$, results to coefficients which are given by the l^{th} column of $\tilde{Y}$. Since matrix U is a unitary, then we can deduce that

$$\begin{aligned} y_l &= \sum_{k=1}^{r} \tilde{Y}_{kl}\tilde{U}_k = \sum_{k=1}^{r} (\tilde{\Sigma}\tilde{V}^*)_{kl}u_k = \sum_{k=1}^{r} (\tilde{U}^*\tilde{U}\tilde{\Sigma}\tilde{V}^*)_{kl}u_k \\ &= \sum_{k=1}^{r} (\tilde{U}^*Y)_{kl}u_k = \sum_{k=1}^{r} (\sum_{l=1}^{n} \tilde{U}^*_{ik}Y_{il})u_k \\ &= \sum_{k=1}^{r} \langle u_k, y_l\rangle u_l \end{aligned}$$

Thus, y_l is the Fourier series representation with the form

$$y_l = \sum_{k=1}^{r} \langle u_k, y_l\rangle u_k \tag{2.1.4}$$

2.1.1 Low Rank Matrix Approximation

Given a $n \times m$ complex valued matrix of rank r, low rank matrix approximation permits us to find an approximate matrix $\tilde{Y}$ which has rank less or equal to r. Therefore, we can say low rank approximation is a minimization problem:

$$\text{argmin}\left\|Y - \tilde{Y}\right\|_F \quad \text{subject to} \quad \text{rank}(\tilde{Y}) \leq r.$$

Theorem 2.1.1. *(Eckart-Young)*
The optimal rank-d approximation to Y, in a least square sense, is given by the rank-d SVD truncation $\tilde{Y}$:

$$argmin\left\|Y - \tilde{Y}\right\|_F = \tilde{U}\tilde{\Sigma}\tilde{V}^*$$

Here, $\tilde{U}$ and $\tilde{V}$ stand for the leading d columns of matrices U and V respectively, $\tilde{\Sigma}$ contains the leading $d \times d$ sub-block of Σ. $||.||_F$ is the frobenius norm and $\tilde{Y} = \tilde{U}\tilde{\Sigma}\tilde{V}^*$ denotes the truncated SVD basis.

2.2 Dynamical systems and Reduced Order Models

2.2.1 Dynamical system (Nonlinear Ordinary Differential Equation)

The general form of a nonlinear ODE is given by

$$\frac{d}{dt}\mathbf{y}(t) = \mathbf{f}(\mathbf{y}, t; \mu)$$

where $\mathbf{f}$ is the vector field, vector $\mathbf{y}$(t) $\in \mathbb{C}^n$ is known as the state of the dynamical system and μ are constants. We can say $\mathbf{f}$ is Lipschitz continuous function.

2.2.2 Nonlinear Partial Differential Equations

Definition 2.2.1. *Nonlinear PDE is defined as partial differential equation with nonlinear terms.*

The general form of the equation is

$$\mathbf{u}_t = \mathbf{f}(\mathbf{u}, \mathbf{u}_x, \mathbf{u}_{xx}, \ldots . x, t)$$

where $\mathbf{u}$ is the state of the PDE, $\mathbf{f}$ denote the nonlinear operator, x denote the spatial variables while t is the temporal variable and the subscripts denote partial differentiation.

2.2.3 Galerkin Projection

Galerkin projection is a method applied to high fidelity dynamical systems to obtain reduced models. This Galerkin projection is given by

$$u(x,t) \approx \sum_{i=1}^{d} a_i(t)\psi_i(x)$$

where the functions $a_l(t)$ are known as temporal coefficients and $\psi_i(x)$ are spatial modes.

In the case of high fidelity discretized state, the Galerkin projection is

$$\mathbf{u}(x,t) \approx \sum_{i=1}^{d} a_i(t)\Psi_i(x)$$

where spatial modes $\Psi_i \in \mathbb{C}^n$ are the columns of $\tilde{U} = \Psi$.

3 POD Method

3.1 Eigenvalue problem

Suppose we have a data set $Y \in \mathbb{C}^n$ of rank r that is a function of both space and time. The objective of POD method is to compute an orthonormal basis $\{u_k\}_{k=1}^d$, that is optimal and approximates the data set in some subspace X with dimension $d << n$. These orthonormal basis $\{u_k\}_{k=1}^d$ are called POD basis of rank d and u_k are POD modes or POD basis vectors.

Let $\{u_k\}_{k=1}^d$ denote an orthonormal basis of subspace X, then it is possible to write a finite dimensional representation of the form

$$y_l = \sum_{k=1}^{d} a_{kl} u_k \quad k = 1, \ldots, d, l = 1, \ldots, m \tag{3.1.1}$$

that describe Y better than presentation of the similar size in different basis. Here, a_{kl} are unknown coefficients of the expansion.

The mathematical statement of optimality is that we are supposed to choose u such that the error in least square sense due to the approximation of y_l is minimized.

$$e = \left\| y_l - \sum_{k=1}^{d} a_{kl} u_k \right\|_2$$

where $\|.\|$ is the induced l_2 norm.
The sum (squared error) is given by

$$E = \sum_{l=1}^{n} \left\| y_l - \sum_{k=1}^{d} a_{kl} u_k \right\|_2^2$$

We can rewrite this equation as

$$E = \sum_{l=1}^{n} \|y_l\|_2^2 - 2 \sum_{l=1}^{n} \sum_{k=1}^{d} a_{kl} y_l^* u_k + \sum_{l=1}^{n} \sum_{k=1}^{d} a_{kl}^2 \tag{3.1.2}$$

We can now minimize the error with respect to both a_{kl} and u_k.
First, let's minimize E with respect to the unknown coefficient a_{kl}. This is done by obtaining partial derivatives of E with respect to a_{rs}

$$\frac{\partial E}{\partial a_{rs}} = -2y_r^* u_s + 2a_{rs} = 0$$

From the equation, $a_{rs} = y_r^* u_s = \langle y_r, u_s \rangle$

Therefore, Equation (3.1.1) is called a *Generalized Fourier series* where a_{kl} are called the *Fourier coefficients* defined by $a_{kl} = y_l^* u_k = \langle y_l, u_k \rangle$.

We substitute the value of a_{kl} in Equation (3.1.2) to get

$$\begin{aligned} E &= \sum_{l=1}^{n} \|y_l\|_2^2 - 2\sum_{l=1}^{n}\sum_{l=1}^{d} (y_l^* u_k) y_l^* u_k + \sum_{l=1}^{n}\sum_{k=1}^{d} (y_l^* u_k)^2 \\ &= \sum_{l=1}^{n} \|y_l\|_2^2 - \sum_{l=1}^{n}\sum_{k=1}^{d} (y_l^* u_k)^2 \\ &= \sum_{l=1}^{n} \|y_l\|_2^2 - \sum_{k=1}^{d} u_k^* \left(\sum_{l=1}^{n} y_l^* y_l \right) u_k \end{aligned}$$

Let $\beta = \sum_{l=1}^{n} \|y_l\|_2^2$ and $W = YY^*$

Then, the equation reduces to

$$E = \beta - \sum_{k=1}^{d} u_k^* W u_k$$

Further, we minimize the error E with respect to u_k. This is equivalent to maximizing the second term

$$\max \sum_{k=1}^{d} u_k^* W u_k \quad \text{subject to} \quad u_k^* u_k = 1, k = 1, \ldots, d \tag{3.1.3}$$

This equality constitute a problem in the calculus of variations: to extremize $\sum_{k=1}^{d} u_k^* W u_k$ subject to the constraint $u_k^* u_k = 1, k = 1, \ldots, d$. In this case, we will use the *method of Lagrange multipliers.*

The Lagrange functional associated to this constrained variational problem is

$$\mathscr{L}(u_k,\lambda) = \sum_{k=1}^{d} u_k^* W u_k + \lambda(1 - u_k^* u_k)$$

Considering first-order necessary optimality condition $\nabla\mathscr{L}(u_k,\lambda) = 0$, we then find the gradient of $\mathscr{L}$ subject to u_k:

$$\begin{aligned}\frac{\partial\mathscr{L}}{\partial u_k} &= \frac{\partial}{\partial u_k}\left(\sum_{k=1}^{d} u_k^* W u_k + \lambda(1 - u_k^* u_k)\right)\\ &= 2Wu_k - 2\lambda u_k\\ &= 2YY^* u_k - 2\lambda u_k\end{aligned}$$

Thus,

$$\nabla\mathscr{L}(u_k,\lambda) = 2(YY^* u_k - \lambda u_k) = 0 \tag{3.1.4}$$

Equation (3.1.4) yields the eigenvalue problem

$$YY^* u_k = \lambda u_k \quad \text{in} \quad \mathbb{C}^n$$

Hence, the optimal basis is given by the eigenfunctions $\{u_k\}_{k=1}^{d}$ of YY^* that is defined from computing SVD of empirical data Y. This optimal basis are consequently called POD modes and therefore Equation(3.1.1) is called the Proper Orthogonal Decomposition of y_l.

This result can be summarized in Theorem 3.1.1

Theorem 3.1.1. *(POD basis) Suppose $Y = [y_1, y_2, \ldots, y_m] \in \mathbb{C}^{n\times m}$ is a matrix of rank $r \leq$ min{n, m}. Moreover, let $Y = U\Sigma V^*$ be the SVD of Y, where $U = [u_1, u_2, \ldots, u_n] \in \mathbb{C}^{n\times n}$, $V = [v_1, v_2, \ldots, v_m] \in \mathbb{C}^{m\times m}$ are unitary matrices and diagonal matrix $\Sigma \in \mathbb{R}^{n\times m}$ has real and non-negative entries . Then, for any $1 \leq d \leq r$, the solution to the maximization problem*

$$\max_{\bar{u_1},\ldots\bar{u_d}\in\mathbb{C}^n} \sum_{k=1}^{d}\sum_{l=1}^{m} |\langle y_l, \bar{u_k}\rangle|^2 \quad s.\ t \quad \langle\bar{u_k},\bar{u_l}\rangle = \delta_{kl}, \quad for \quad 1 \leq k,l \leq r \tag{Q1}$$

where

$$\delta_{kl} = \begin{cases} 0, & k \neq l \\ 1, & k = l \end{cases}$$

is given by the singular vectors $\{u_k\}_{k=1}^d$*, i.e, by the leading* d *columns of matrix* U*. Further,*

$$argmax(Q1) = \sum_{k=1}^{d} \sigma_k^2 = \sum_{k=1}^{d} \lambda_k \tag{3.1.5}$$

`Proof.` The Equality (Q1) is a constrained optimization problem, therefore we can use the method of Lagrange multipliers.

The Lagrangian $\mathscr{L}$ is given by

$$\mathscr{L}(u_1, u_2, \ldots, u_d, \Lambda) = \sum_{k=1}^{d} \sum_{l=1}^{m} |\langle y_l, u_k \rangle|^2 + \sum_{k,l=1}^{d} \lambda_{kl}(\delta_{kl} - \langle u_k, u_l \rangle)$$

for $u_1, u_2, \ldots u_d \in \mathbb{C}^n$ and $\Lambda = (\lambda_{kl}) \in \mathbb{R}^{d \times d}$. The first-order optimality necessary condition for this problem is given by

$$\frac{\partial \mathscr{L}}{\partial u_i}(u_1, u_2, \ldots, u_d, \Lambda)\delta u_i = 0 \quad \text{for all} \quad \delta u_i \in \mathbb{C}^n \quad \text{and} \quad i \in \{1, \ldots, d\} \tag{3.1.6}$$

Then from

$$\begin{aligned}
\frac{\partial \mathscr{L}}{\partial u_i}(u_1, u_2 \ldots, u_d, \Lambda)\delta U_i &= 2 \sum_{k=1}^{d} \sum_{l=1}^{m} \langle y_l, u_k \rangle \langle y_l, \delta u_i \rangle \delta_{ki} \\
&\quad - \sum_{k,l=1}^{d} \lambda_{kl} \langle u_k, \delta u_i \rangle \delta_{li} - \sum_{k,l=1}^{d} \lambda_{kl} \langle \delta u_i, u_l \rangle \delta_{ik} \\
&= 2 \sum_{l=1}^{m} \langle y_l, u_i \rangle \langle y_l, \delta u_i \rangle - \sum_{k=1}^{d} (\lambda_{ki} + \lambda_{ik}) \langle u_k, \delta u_i \rangle \\
&= \left\langle 2 \sum_{l=1}^{m} \langle y_l, u_i \rangle y_l - \sum_{k=1}^{d} (\lambda_{ki} + \lambda_{ik}) u_k, \delta u_i \right\rangle
\end{aligned}$$

and Equation (3.1.6) we deduce that

$$\sum_{l=1}^{m} \langle y_l, u_i \rangle y_l = \frac{1}{2} \sum_{k=1}^{d} (\lambda_{ki} + \lambda_{ik}) u_k \quad \text{in} \quad \mathbb{R}^n \quad \text{for all} \quad i \in \{1, \ldots, d\} \tag{3.1.7}$$

Note that

$$YY^* u = \sum_{l=1}^{m} \langle y_l, u_l \rangle y_l \quad \text{for} \quad u \in \mathbb{C}^n$$

Therefore, Equation (3.1.7) can be expressed as

$$YY^*u_i = \frac{1}{2}\sum_{k=1}^{d}(\lambda_{ki}+\lambda_{ik})u_k \quad \text{in} \quad \mathbb{C}^n \quad \text{for all} \quad i \in \{1,\dots,d\} \tag{3.1.8}$$

We then continue by induction. When $d = 1$, $i = 1$. Then Equation (3.1.8) becomes

$$YY^*u_1 = \lambda_1 u_1 \quad \text{in} \quad \mathbb{C}^n \tag{3.1.9}$$

with $\lambda_1 = \lambda_{11}$.
Now for $d \geq 1$, suppose that necessary optimality conditions are identified by

$$YY^*u_i = \lambda_i u_i \quad \text{in} \quad \mathbb{C}^n \quad \text{for all} \quad i \in \{1,\dots,d\} \tag{3.1.10}$$

We want to show that the first order necessary condition for optimality for a POD basis $\{u_k\}_{k=1}^{d+1}$ of rank $d+1$ is given by

$$YY^*u_i = \lambda_i u_i \quad \text{in} \quad \mathbb{C}^n \quad \text{for all} \quad i \in \{1,\dots,d+1\} \tag{3.1.11}$$

By assumption, Equation (3.1.10) is true. Hence, we have remained to prove that

$$YY^*u_{i+1} = \lambda_{i+1} u_{i+1} \quad \text{in} \quad \mathbb{C}^n \tag{3.1.12}$$

Using Equation (3.1.8), we have

$$YY^*u_{d+1} = \frac{1}{2}\sum_{k=1}^{d+1}(\lambda_{k,d+1}+\lambda_{d+1,k})u_k \quad \text{in} \quad \mathbb{C}^n \tag{3.1.13}$$

Since $\{u_k\}_{k=1}^{d+1}$ is a POD basis, we have $\langle u_{d+1}, u_k\rangle = 0$ for $1 \leq k \leq d$. YY^* is hermitian, so using Equation (3.1.10), for any $k \in 1,2,\dots,d$ we have

$$
\begin{aligned}
0 &= \lambda_k \langle u_{d+1}, u_k \rangle = \langle u_{d+1}, YY^* u_k \rangle = \langle YY^* u_{d+1}, u_k \rangle \\
&= \frac{1}{2} \sum_{k=1}^{d+1} (\lambda_{k,d+1} + \lambda_{d+1,k}) \langle u_k, u_k \rangle = (\lambda_{k,d+1} + \lambda_{d+1,k})
\end{aligned}
$$

which gives

$$
\lambda_{d+1,k} = -\lambda_{k,d+1} \quad \text{for any} \quad k \in \{1, \ldots, d\} \tag{3.1.14}
$$

Plugging in Equation (3.1.14) to Equation (3.1.13) we get

$$
\begin{aligned}
YY^* u_{d+1} &= \frac{1}{2} \sum_{k=1}^{d} (\lambda_{k,d+1} + \lambda_{d+1,k}) u_k + \lambda_{d+1,d+1} u_{d+1} \\
&= \frac{1}{2} \sum_{k=1}^{d} (\lambda_{k,d+1} - \lambda_{k,d+1}) u_k + \lambda_{d+1,d+1} u_{d+1} \\
&= \lambda_{d+1,d+1} u_{d+1}
\end{aligned}
$$

When we set $\lambda_{d+1} = \lambda_{d+1,d+1}$, we obtain Equation (3.1.12). Thus, the necessary conditions for (Q1) are given by the eigenvalue problem

$$
YY^* u_k = \lambda_k u_k \quad \text{for} \quad k \in \{1, \ldots, d\}. \tag{3.1.15}
$$

From SVD, we can conclude that $\{u_k\}_{k=1}^d$ solves Equation (3.1.15). The proof that $\{u_k\}_{k=1}^d$ is a solution to Equality (Q1) and that argmax(Q1) = $\sum_{k=1}^d \sigma_k^2$ holds is analogous to the proof for (Q1).

□

3.2 Optimality of the POD Basis

Given $d \in \{1, \ldots, r\}$, the vectors $\{u_k\}_{k=1}^d$ are known as POD basis with rank d. The results below shows that for each $d \leq r$ the approximation of the columns of matrix Y by the leading d singular vectors $\{u_k\}_{k=1}^d$ is optimal amid each and every rank d approximations to the columns of Y in average sense.

Corollary 3.2.1. *Suppose all hypothesis of Theorem 3.1.1 are satisfied and let $\hat{U}^r \in \mathbb{C}^{n\times r}$ be a matrix with pairwise orthonormal vectors $\hat{u}_k$. Then columns of Y can be expanded in the basis $\{\hat{u}_k\}_{k=1}^r$ as*

$$Y = \hat{U}^r D^r \quad \textit{where} \quad D^r_{kl} = \langle \hat{u}_k, y_l \rangle \quad \textit{for} \quad 1 \le k \le r, 1 \le l \le m$$

Therefore for every $d \in \{1,\dots,r\}$, we have

$$\left\| Y - U^d B^d \right\|_F \le \left\| Y - \hat{U}^d D^d \right\|_F \tag{3.2.1}$$

whereby $\|.\|_F$ is the Frobenius norm identified by

$$\|Y\|_F = \sqrt{\sum_{k=1}^{n}\sum_{l=1}^{m} |Y_{kl}|^2} = \sqrt{trace(Y^*Y)} \quad for \quad Y \in \mathbb{C}^{n\times m},$$

the matrix U^d represents the leading d columns of U, B^d denotes the leading d rows of B and likewise for both $\hat{U}^d$ and D^d.

Notice that

$$\begin{aligned}
\left\| Y - U^d B^d \right\|_F^2 &= \sum_{k=1}^{n}\sum_{l=1}^{m} |Y_{kl} - \sum_{i=1}^{d} U^d_{ki} B_{il}|^2 \\
&= \sum_{l=1}^{m}\sum_{k=1}^{n} |Y_{kl} - \sum_{i=1}^{d} \langle u_i, y_l \rangle U^d_{ki}|^2 \\
&= \sum_{l=1}^{m} \left\| y_l - \sum_{i=1}^{d} \langle y_l, u_i \rangle u_i \right\|^2
\end{aligned}$$

Analogously,

$$\begin{aligned}
\left\| Y - \hat{U}^d D^d \right\|_F^2 &= \sum_{k=1}^{n}\sum_{l=1}^{m} |Y_{kl} - \sum_{i=1}^{d} \hat{U}^d_{ki} D_{il}|^2 \\
&= \sum_{l=1}^{m}\sum_{k=1}^{n} |Y_{kl} - \sum_{i=1}^{d} \langle \hat{u}_i, y_k \rangle \hat{U}^d_{ki}|^2 \\
&= \sum_{l=1}^{m} \left\| y_l - \sum_{i=1}^{d} \langle y_l, \hat{u}_i \rangle \hat{u}_i \right\|^2
\end{aligned}$$

Thus, Equation (3.2.1) implies that

$$\sum_{l=1}^{m} \left\| y_l - \sum_{i=1}^{d} \langle y_l, u_i \rangle u_i \right\|^2 \leq \sum_{l=1}^{m} \left\| y_l - \sum_{i=1}^{d} \langle y_l, \hat{u}_i \rangle \hat{u}_i \right\|^2$$

for a different set $\{\hat{U}_k\}_{k=1}^{i}$ of i pairwise orthonormal vectors.

Proof.

$$\begin{aligned} \left\| Y - \hat{U}^d D^d \right\|_F^2 &= \left\| \hat{U}^r (D^r - D_o^r) \right\|_F^2 = \| D^r - D_O^r \|_F^2 \\ &= \sum_{k=d+1}^{r} \sum_{l=1}^{m} |D_{kl}^d|^2, \end{aligned}$$

where $D_o^r \in \mathbb{C}^{r \times m}$ results from $D \in \mathbb{C}^{r \times m}$ after we replace the last $r-d$ rows by 0.

Correspondingly,

$$\begin{aligned} \left\| Y - U^d B^d \right\|_F^2 = \left\| U^d (B^r - B_o^d) \right\|_F^2 &= \left\| B^r - B_O^d \right\|_F^2 \\ &= \sum_{k=d+1}^{r} \sum_{l=1}^{m} |B_{kl}^r|^2 \\ &= \sum_{k=d+1}^{r} \sum_{l=1}^{m} |\langle y_l, u_k \rangle|^2 \\ &= \sum_{k=d+1}^{r} \sum_{l=1}^{m} \langle \langle y_l, u_k \rangle y_l, u_k \rangle = \sum_{k=d+1}^{r} \langle YY^* u_k, u_k \rangle \\ &= \sum_{k=d+1}^{r} \sigma_k^2 \end{aligned}$$

By Theorem 3.1.1, the vectors $u_1, \ldots, u_d$ solves Q1.

From

$$\|Y\|_F^2 = \left\| \hat{U}^r D^r \right\|_F^2 = \|D^r\|_F^2 = \sum_{k=1}^{r} \sum_{l=1}^{m} |D_{kl}^r|^2$$

and

$$\|Y\|_F^2 = \|U^r B^r\|_F^2 = \|B^r\|_F^2 = \sum_{k=1}^{r} \sum_{l=1}^{m} |B_{kl}^r|^2$$

we deduce that

$$\begin{aligned}\left\|Y-U^dB^d\right\|_F^2 &= \sum_{k=d+1}^{r}\sigma_k^2 = \sum_{k=1}^{r}\sigma_k^2 - \sum_{k=1}^{d}\sigma_k^2 = \|Y\|_F^2 - \sum_{k=1}^{r}\sum_{l=1}^{m}|\langle y_l,\hat{u_k}\rangle|^2 \\ &\leq \|Y\|_F^2 - \sum_{k=1}^{d}\sum_{l=1}^{m}|\langle y_k,\hat{u_k}\rangle|^2 = \sum_{k=1}^{r}\sum_{l=1}^{m}|D_{kl}^r|^2 - \sum_{k=1}^{d}\sum_{l=1}^{m}|D_{kl}^r|^2 \\ &= \sum_{k=d+1}^{r}\sum_{l=1}^{m}|D_{kl}^r|^2 = \left\|Y-\hat{U}^dD^d\right\|_F^2\end{aligned}$$

which gives Equation (3.2.1). □

Proposition 3.2.2. *From Corollary 3.2.1, the following results hold:*

1. $\sum_{k=1}^{d}\sum_{l=1}^{m}|\langle y_l,u_k\rangle|^2 = \sum_{k=1}^{d}\sigma_k^2 = \sum_{k=1}^{d}\lambda_k \geq \sum_{k=1}^{d}\sum_{l=1}^{m}|\langle y_l,\hat{u_k}\rangle|^2$
   for a different set of orthonormal vectors $\{\hat{u_k}\}_{k=1}^d$
2. $\sum_{l=1}^{m}\langle y_l,u_k\rangle\langle y_l,u_i\rangle = \sum_{l=1}^{m}B_{kl}^dB_{il}^d = \sigma_k^2\delta_{ki}$ *for* $1\leq k,i\leq d$. *That is, the POD coefficients are not associated.*

This proposition demands that optimal POD basis of rank d is core for rebuilding a signal y_l. This insinuates that it is the most competence amongst each and every linear decompositions, in the sense of illustrating the columns $\{y_l\}_{l=1}^m$ of matrix Y as a linear combination by an orthonormal basis of rank d in an average sense. Furthermore, the time series of the POD coefficients are not associated.

We can summarize the computation of POD basis using alorithm 1 as shown below.

Algorithm 1: POD basis in $\mathbb{C}^n$

INPUT: Snapshot vectors $\{y_l\}_{l=1}^m \in \mathbb{C}^n$
OUTPUT: POD basis $\{u_k\}_{k=1}^d \in \mathbb{C}^{n\times d}$
1. Create snapshot matrix: $Y=[y_1,\ldots,y_m]\in\mathbb{C}^{n\times m}$ and $r\leq\min\{n,m\}$.
2. Compute SVD: $Y=U\Sigma V^*$ and choose dimension $d\leq r$.
3. POD basis of rank d : $\{u_k\}_{k=1}^d=\{u_1,\ldots,u_d\}$.

3.3 POD with weighted inner product

We can define the weighted inner product described on the unitary space $\mathbb{C}^n$ by

$$\langle u, \bar{u} \rangle_W = u^* W \bar{u} = \langle u, W\bar{u} \rangle = \langle Wu, \bar{u} \rangle \quad \text{for} \quad u, \bar{u} \in \mathbb{C}^n \tag{3.3.1}$$

where $W \in \mathbb{C}^{n\times n}$ is a hermitian, positive definite matrix and the corresponding induced norm is $\|u\|_W = \sqrt{\langle u^*, u \rangle_W}$ for $u \in \mathbb{C}^n$. Suppose W is an identity matrix in $\mathbb{C}^n$, then the weighted inner product (3.3.1) coincides the unitary inner product.

Let us replace (Q1) by

$$\max_{u\in\mathbb{C}^n} \sum_{l=1}^{m} |\langle y_l, u \rangle_W|^2 \quad \text{s. t} \quad \|u\|_W = 1, \tag{Q2}$$

Similarly to section 3.1, (Q2) constitute a problem in the calculus of variations. The Lagrangian $\mathscr{L} : \mathbb{C}^n \times \mathbb{C} \to \mathbb{C}$ for (Q2) is described by

$$\mathscr{L}(u, \lambda) = \sum_{l=1}^{m} |\langle y_l, u \rangle_W|^2 + \lambda(1 - \|u\|_W^2) \quad \text{for} \quad (u, \lambda) \in \mathbb{C}^n \times \mathbb{C}.$$

Let $u \in \mathbb{C}^n$ represent a solution to (Q2), then a first-order necessary optimality condition is defined by

$$\nabla \mathscr{L}(u, \lambda) = 0 \quad \text{in} \quad \mathbb{C}^n \times \mathbb{C}.$$

We now find the gradient of $\mathscr{L}$ with reference to u. Using the condition that W is hermitian, we obtain

$$\begin{aligned}
\frac{\partial \mathscr{L}}{\partial u_k}(u, \lambda) &= \frac{\partial}{\partial u_k} \left(\sum_{l=1}^{m} \left| \sum_{p=1}^{n} \sum_{q=1}^{n} Y_{lq}^* W_{qp} u_p \right|^2 + \lambda \left(1 - \sum_{p=1}^{n} \sum_{q=1}^{n} u_q W_{qp} u_p\right) \right) \\
&= 2 \sum_{l=1}^{m} \left(\sum_{p=1}^{n} \sum_{q=1}^{n} Y_{lq}^* W_{qp} u_p \right) \left(\sum_{\mu=1}^{n} Y_{l\mu}^* W_{\mu k} \right) - \lambda \left(\sum_{q=1}^{n} u_q W_{qk} + \sum_{p=1}^{n} W_{kp} u_p \right) \\
&= 2 \sum_{p=1}^{n} \sum_{q=1}^{n} \sum_{\mu}^{n} W_{k\mu} \sum_{l=1}^{m} Y_{\mu l} Y_{lq}^* W_{qp} u_p - 2\lambda \left(\sum_{p=1}^{n} W_{kp} u_p \right) \\
&= 2(WYY^*Wu - \lambda Wu)_k
\end{aligned}$$

Thus,

$$\nabla_u \mathscr{L}(u,\lambda) = 2(WYY^*Wu - \lambda Wu) = 0 \quad \text{in} \quad \mathbb{C}^n. \tag{3.3.2}$$

This Equation (3.3.2) generates the eigenvalue problem

$$(WY)(WY)^* u = \lambda Wu \tag{3.3.3}$$

Because W is hermitian, positive definite matrix, then $W = PDP^*$ is the eigenvalue decomposition whereby $D = \text{diag}(\tau_1, \ldots, \tau_n)$ consist of the eigenvalues $\tau_1 \geq \cdots \geq \tau_n > 0$ of W and $p \in \mathbb{C}^{n\times n}$ is a unitary matrix. We define

$$W^\alpha = P\text{diag}(\tau_1^\alpha, \ldots, \tau_n^\alpha)P^* \quad \text{for} \quad \alpha \in \mathbb{C}$$

Note that $(W^\alpha)^{-1} = W^{-\alpha}$ and $W^{\alpha+\beta} = W^\alpha W^\beta$ for $\alpha, \beta \in \mathbb{C}$.
Further, we have

$$\langle u, \bar{u} \rangle_W = \left\langle W^{\frac{1}{2}} u, W^{\frac{1}{2}} \bar{u} \right\rangle \quad \text{for} \quad u, \bar{u} \in \mathbb{C}^n$$

and $\|u\|_W = \left\| W^{\frac{1}{2}} u \right\|$ for $u \in \mathbb{C}^n$.
Let $\bar{u} = W^{\frac{1}{2}} u \in \mathbb{C}^n$ and $\bar{Y} = W^{\frac{1}{2}} Y \in \mathbb{C}^{n\times m}$ and multiply Equation (3.3.3) by $W^{\frac{1}{2}}$ from the left, we deduce $n \times n$ eigenvalue problem.

$$\bar{Y}\bar{Y}^* \bar{u} = \lambda \bar{u} \quad \text{in} \quad \mathbb{C}^n \tag{3.3.4}$$

From $\frac{\partial \mathscr{L}}{\partial \lambda}(u,\lambda) = 0$, we can express the constraint $\|u\|_W = 1$ as

$$\|\bar{u}\| = 1$$

Thus, the first-order optimality conditions Equation (3.3.4) for (Q2) are the same as for (Q1), but the matrix Y and vector u need to be weighted by $W^{\frac{1}{2}}$.
Also, we can show that

$$u_1 = W^{-\frac{1}{2}} \bar{u}_1,$$

solves Q2, where $\bar{u_1}$ is the eigenvector of $\bar{Y}Y^*$ that corresponds to the largest eigenvalue λ_1 with $\|\bar{u_1}\| = 1$. Using SVD, we can determine vector u_1 by first working out $n \times n$ eigenvalue problem

$$\bar{Y}^*\bar{Y}\bar{v}_1 = \lambda_1 \bar{v}_1$$

where $\bar{Y}^*\bar{Y} = Y^*WY$
Setting

$$u_1 = W^{-\frac{1}{2}} \bar{u}_1 = \frac{1}{\sqrt{\lambda_1}} W^{-\frac{1}{2}} \bar{Y}\bar{v}_1 = \frac{1}{\sqrt{\lambda_1}} Y\bar{v}_1$$

we can look at a second vector $u \in \mathbb{C}^n$ with $\langle u, u_1 \rangle_W = 0$ that optimizes $\sum_{j=1}^m |\langle y_j, u \rangle_W|^2$

Let us derive Theorem 3.1.1 as follows:

Theorem 3.3.1. *Assume that $Y = [y_1, y_2, \ldots, y_m] \in \mathbb{C}^{n\times m}$ is a matrix of rank $r \leq min\{n,m\}$, W is a Hermitian, positive definite matrix, $\bar{Y} = W^{\frac{1}{2}}Y$ and $d \in \{1,\ldots,r\}$. Moreover, let $\bar{Y} = \bar{U}\Sigma\bar{V}^*$ be the Singular Value Decomposition of $\bar{Y}$, whereby both $\bar{U} = [\bar{u}_1,\ldots,\bar{u}_m] \in \mathbb{C}^{n\times n}$, $\bar{V} = [\bar{v}_1,\ldots,\bar{v}_m] \in \mathbb{C}^{m\times m}$ are unitary matrices and*

$$\Sigma = \begin{bmatrix} \tilde{\Sigma} & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{n\times m}$$

Then, the solution to

$$\max_{\bar{u}_1,\ldots,\bar{u}_r \in \mathbb{C}^n} = \sum_{k=1}^{d}\sum_{l=1}^{m} |\langle y_l, \bar{u}_k\rangle_W|^2 \quad s.\ t \quad \langle \bar{u}_k, \bar{u}_l\rangle_W = \delta_{kl} \quad for \quad 1 \leq k,l \leq d,, \qquad \textbf{(Q3)}$$

is described by the vectors $u_k = W^{-\frac{1}{2}}\bar{u}_k, k = 1,\ldots,d$
Further,

$$argmax(Q3) = \sum_{k=1}^{d}\sigma_k^2 = \sum_{k=1}^{r}\lambda_k$$

Proof. The proof follows same arguments as shown in the proof of Theorem 3.1.1. □

Because of SVD and $\bar{Y}^*Y = Y^*WY$, the method of snapshot can be used to obtain the POD basis $\{u_k\}_{k=1}^d$ of rank d by solving the eigenvalue problem

$$Y^*WY\bar{v}_k = \lambda_k\bar{v}_k \quad \text{for} \quad k = 1,\ldots,d$$

and setting

$$u_k = W^{-\frac{1}{2}}\bar{u}_k = \frac{1}{\sqrt{\lambda_k}}W^{-\frac{1}{2}}(\bar{Y}\bar{v}_k) = \frac{1}{\sqrt{\lambda_k}}W^{-\frac{1}{2}}W^{\frac{1}{2}}Y\bar{v}_k = \frac{1}{\sqrt{\lambda_k}}Y\bar{v}_k, \quad \text{for} \quad k = 1,\ldots,d.$$

Notice that

$$\langle u_k, u_l\rangle_W = u_k^*Wu_l = \frac{\delta_{kl}\lambda_l}{\sqrt{\lambda_k\lambda_l}} \quad \text{for} \quad 1 \leq k,l \leq d.$$

For $n >> m$, the method of snapshot is faster than computing POD basis using Equation (3.3.4). The matrix $W^{\frac{1}{2}}$ is not a requirement for the method of snapshots.

We can summarize the computation of POD basis with weighted inner product as follows.

Algorithm 1: POD basis in $\mathbb{C}^n$ with weighted inner product

---

INPUT: Snapshot vectors $\{y_l\}_{l=1}^m \in \mathbb{C}^n$ and hermitian, positive definite matrix $W \in \mathbb{C}^{n\times n}$
OUTPUT: POD basis $\{u_k\}_{k=1}^d \in \mathbb{C}^{n\times d}$
1. Create snapshot matrix: $Y = [y_1, \ldots, y_m] \in \mathbb{C}^{n\times m}$ and $r \leq \min\{n, m\}$.
2. Determine $\bar{Y} = W^{\frac{1}{2}} Y \in \mathbb{C}^{n\times m}$
3. Compute SVD: $Y = U\Sigma V^*$ and choose dimension $d \leq r$.
4. POD basis with weighted inner product of rank d: $u_k = W^{-\frac{1}{2}} \bar{u}_k, k = 1, \ldots, d$.

3.4 Continuous formulation of the POD

In this section we talk about some extra points of importance concerning the POD method. To solve Equality (Q3), the techniques used in Section 3.1 and Section 3.3 are applied. That is, we use the method of Lagrangian.

$$\mathscr{L}(u_1, \ldots, u_d, \Lambda) = \sum_{l=1}^{m} \alpha_l \left\| y_l - \sum_{k=1}^{d} \langle y_l, u_k \rangle_W u_k \right\|_W^2 + \sum_{k=1}^{d}\sum_{l=1}^{d} \Lambda_{kl}(1 - \langle u_k, u_l \rangle_W)$$

for $u_1, \ldots, u_d \in \mathbb{C}^n$ and $\Lambda \in \mathbb{R}^{d\times d}$ with elements Λ_{kl}, $1 \leq k, l \leq d$. Thus, the solution to Q3 is given by the first-order optimality conditions

$$\nabla_{u_k} \mathscr{L}(u_1, \ldots, u_d, \Lambda) = 0 \quad \text{in} \quad \mathbb{C}^n, 1 \leq k \leq d, \tag{3.4.1}$$

and

$$\langle u_i, u_l \rangle_W = \delta_{kl}, \quad 1 \leq k, l \leq d. \tag{3.4.2}$$

From Equation (3.4.1), we derive

$$Y\Sigma Y^* W u_k = \lambda_k u_k \quad \text{for} \quad k = 1, \ldots, d \tag{3.4.3}$$

where $\Sigma = \text{diag}(\alpha_1, \ldots, \alpha_m) \in \mathbb{R}^{m\times m}$. Inserting $u_k = W^{-\frac{1}{2}} \bar{u}_k$ in Equation (3.4.3) and multiplying it by $W^{\frac{1}{2}}$ from the left yields

$$W^{\frac{1}{2}} Y\Sigma Y^* W^{\frac{1}{2}} \bar{u}_k = \lambda_k \bar{u}_k \tag{3.4.4}$$

From equation (3.4.2) we find

$$\langle \bar{u}_k, \bar{u}_l \rangle = \bar{u}_k^* \bar{u}_l = u_k^* W u_l = \langle u_k, u_l \rangle_W = \delta_{kl}, \quad 1 \le k, l \le d. \tag{3.4.5}$$

Suppose $\bar{Y} = W^{\frac{1}{2}} Y \Sigma^{\frac{1}{2}} \in \mathbb{C}^{n \times m}$ and $W^* = W$ same as $\Sigma^* = \Sigma$, we infer from equation (3.4.4) and (3.4.5) that the solution $\{u_k\}_{k=1}^d$ to equality (Q3) is given by the $n \times n$ eigenvalue problem

$$\bar{Y}\bar{Y}^* \bar{u}_k = \lambda_k \bar{u}_k, 1 \le k \le d \quad \text{and} \quad \langle \bar{u}_k, \bar{u}_l \rangle = \delta_{kl}, 1 \le k, l \le d.$$

Notice that

$$\bar{Y}^* \bar{Y} = \Sigma^{\frac{1}{2}} Y^* W Y \Sigma^{\frac{1}{2}} \in \mathbb{C}^{m \times m}.$$

Hence, we can find the POD basis of rank d using the method of snapshots as follows. We first work out the $m \times m$ eigenvalue problem

$$\bar{Y}^* \bar{Y} \bar{v}_k = \lambda_k \bar{v}_k, 1 \le k \le d \quad \text{and} \quad \langle \bar{v}_k, \bar{v}_l \rangle = \delta_{kl}, 1 \le k, l \le d.$$

Using SVD, we then set

$$u_k = W^{-\frac{1}{2}} \bar{u}_k = \frac{1}{\sqrt{\lambda_k}} W^{-\frac{1}{2}} \bar{Y} \bar{v}_k = \frac{1}{\sqrt{\lambda_k}} Y \Sigma^{\frac{1}{2}} \bar{v}_k, 1 \le k \le d;$$

Notice that $u_1, \ldots, u_d$ are POD basis vectors which are orthonormal in $\mathbb{C}^n$ with reference to the inner product $\langle ., . \rangle_W$, i.e.,

$$\langle u_k, u_l \rangle_W = u_k^* W u_l = \frac{1}{\sqrt{\lambda_i \lambda_l}} \bar{v}_k^* \underbrace{\Sigma^{\frac{1}{2}} Y^* W Y \Sigma^{\frac{1}{2}}}_{\bar{Y}^* \bar{Y}} \bar{v}_l = \frac{\lambda_k}{\sqrt{\lambda_k \lambda_l}} \bar{v}_k^* \bar{v}_l = \frac{\lambda_k \delta_{kl}}{\sqrt{\lambda_k \lambda_l}}$$

The snapshot ensemble $\{y_l\}_{l=1}^m$ for Q3 and the snapshot set span $\{y_1, \ldots, y_m\}$ are determined by the selected time instances $\{t_l\}_{l=1}^m$. Accordingly, the POD basis vectors $\{u_k\}_{k=1}^d$ and their corresponding eigenvalues $\{\lambda_k\}_{k=1}^d$ are defined on the same time instances, that is,

$$u_k = u_k^m \quad \text{and} \quad \lambda_k = \lambda_k^m, \quad 1 \le k \le d.$$

Further, how to select best time instances for the snapshots and how to determine appropriate positive weights $\{\alpha_l\}_{l=1}^m$ are the two questions that are yet to be discussed. In

order to compute the POD basis of rank d that describe the whole trajectory $\{y(t)|t \in [0,T]\} \in \mathbb{C}^n$ very well, then we have to consider the minimization problem given by

$$\min_{\tilde{u}_1,\ldots,\tilde{u}_d \in \mathbb{C}^n} \int_0^T \left\| y(t) - \sum_{k=1}^{d} \langle y(t), \tilde{u}_k \rangle_W \, \tilde{u}_k \right\|_W^2 dt \quad \text{s.t.} \quad \langle \tilde{u}_k, \tilde{u}_l \rangle_W = \delta_{kl}, \quad 1 \le k,l \le d. \quad \textbf{(Q4)}$$

In order to solve Equality Q4, similar arguments used in section 3.1 and 3.3 are applied. When $d = 1$, Q4 is obtained instead of the minimization problem

$$\min_{\tilde{u} \in \mathbb{C}^n} \int_0^T \|y(t) - \langle y(t), \tilde{u} \rangle_W \, \tilde{u}\|_W^2 dt \quad \text{s.t.} \quad \|\tilde{u}\|_W^2 = 1, \qquad \textbf{(3.4.6)}$$

If $\{\tilde{u}\}_{k=2}^n$ is selected in an approach that $\{\tilde{u}_1, \ldots, \tilde{u}_n\}$ is an orthonormal basis in $\mathbb{C}^n$ with reference to inner product $\langle ., . \rangle_W$, then we have

$$y(t) = \langle y(t), \tilde{u} \rangle_W \, \tilde{u} + \sum_{k=2}^{n} \langle y(t), \tilde{u}_k \rangle_W \, \tilde{u}_k \quad \text{for all} \quad t \in [0,T].$$

Therefore,

$$\begin{aligned} \int_0^T \|y(t) - \langle y(t), \tilde{u} \rangle_W \, \tilde{u}\|_W^2 dt &= \int_0^T \left\| \sum_{k=2}^{n} \langle y(t), \tilde{u}_k \rangle_W \, \tilde{u}_k \right\|_W^2 dt \\ &= \sum_{k=2}^{n} \int_0^T |\langle y(t), \tilde{u}_k \rangle_W|^2 dt \end{aligned}$$

We can conclude that Equation (3.4.6) is same as the following maximization problem

$$\max_{\tilde{u} \in \mathbb{C}^n} \int_0^T |\langle y(t), \tilde{u} \rangle_W|^2 dt \quad \text{s.t.} \quad \|\tilde{u}\|_W^2 = 1. \qquad \textbf{(3.4.7)}$$

The Lagrange functional $\mathscr{L} : \mathbb{C}^n \times \mathbb{C} \to \mathbb{C}$ associated with Equation (3.4.7) is given by

$$\mathscr{L}(u, \lambda) = \int_0^T |\langle y(t), u \rangle_W|^2 dt + \lambda(1 - \|u\|_W^2) \quad \text{for} \quad (u, \lambda) \in \mathbb{C}^n \times \mathbb{C}$$

The necessary optimality conditions are described by

$$\nabla \mathscr{L}(u,\lambda) = 0 \quad \text{in} \quad \mathbb{C}^n \times \mathbb{C}$$

Thus, the partial derivative of $\mathscr{L}$ with reference to the kth component u_k of the vector u is given by

$$\begin{aligned}\frac{\partial \mathscr{L}}{\partial u_k}(u,\lambda) =& \frac{\partial}{\partial u_k}\left(\int_0^T |\sum_{p=1}^n \sum_{q=1}^n y_p(t) W_{pq} u_q|^2 dt + \lambda\left(1 - \sum_{p=1}^n \sum_{q=1}^n u_p W_{pq} u_q\right)\right) \\ =& 2\int_0^T \left(\sum_{p=1}^n \sum_{q=1}^n y_p(t) W_{pq} u_q\right) \sum_{\mu=1}^n y_\mu(t) W_{\mu k} dt - 2\lambda \sum_{p=1}^n W_{kp} u_q \\ =& 2\left(\int_0^T \langle y(t), u\rangle_W W y(t) dt - \lambda W u\right)_k\end{aligned}$$

for $k \in \{1,\ldots,n\}$. Therefore,

$$\nabla_u \mathscr{L}(u,\lambda) = 2\left(\int_0^T \langle y(t), u\rangle_W W y(t) dt - \lambda W u\right) = 0 \quad \text{in} \quad \mathbb{C}^n$$

which gives

$$\int_0^T \langle y(t), u\rangle_W W y(t) dt = \lambda W u \quad \text{in} \quad \mathbb{C}^n \tag{3.4.8}$$

We now multiply Equation (3.4.8) by W^{-1} from the left to yield

$$\int_0^T \langle y(t), u\rangle_W y(t) dt = \lambda u \quad \text{in} \quad \mathbb{C}^n \tag{3.4.9}$$

We can define the operator $\mathfrak{E} : \mathbb{C}^n \to \mathbb{C}^n$ as

$$\mathfrak{E} u = \int_0^T \langle y(t), u\rangle_W y(t) dt \quad \text{for} \quad u \in \mathbb{C}^n \tag{3.4.10}$$

Lemma 3.4.1. *The operator $\mathfrak{E}$ is linear and bounded. Further,*

1. *$\mathfrak{E}$ is positive:*
$$\langle \mathfrak{E}u, u\rangle_W \geq 0 \quad \textit{for all} \quad u \in \mathbb{C}^n$$
2. *$\mathfrak{E}$ is self-adjoint(or symmetric):*
$$\langle \mathfrak{E}u, u\rangle_W = \langle u, \mathfrak{E}\tilde{u}\rangle_W \quad \textit{for all} \quad u, \tilde{u} \in \mathbb{C}^n$$

Proof. Let $u, \tilde{u} \in \mathbb{C}^n$ be arbitrary and $\beta, \tilde{\beta} \in \mathbb{C}$ we have

$$\begin{aligned}\mathfrak{E}(\beta u + \tilde{\beta}\tilde{u}) &= \int_0^T \left\langle y(t), \beta u + \tilde{\beta}\tilde{u}\right\rangle_W y(t)dt \\ &= \int_0^T (\beta \langle y(t), u\rangle_W + \tilde{\beta}\langle y(t), \tilde{u}\rangle_W) y(t)dt \\ &= \beta \int_0^T \langle y(t), u\rangle_W y(t)dt + \tilde{\beta}\int_0^T \langle y(t), \tilde{u}\rangle_W y(t)dt = \beta\mathfrak{E}u + \tilde{\beta}\mathfrak{E}\tilde{u},\end{aligned}$$

so that $\mathfrak{E}$ is linear. Using cauchy-Schwarz inequality we obtain

$$\begin{aligned}\|\mathfrak{E}u\|_W &\leq \int_0^T \|\langle y(t), u\rangle_W y(t)\|_W dt = \int_0^T |\langle y(t), u\rangle_W| \|y(t)\|_W dt \\ &\leq \int_0^T \|y(t)\|_W^2 \|u\|_W dt = \left(\int_0^T \|y(t)\|_W^2 dt\right)\|u\|_W = \|y\|^2_{L^2(0,T;\mathbb{C}^n)}\|u\|_W\end{aligned}$$

for an arbitrary $u \in \mathbb{C}^n$. Since $y \in C([0,T];\mathbb{C}^n) \subset L^2(0,T;\mathbb{C}^n)$ holds, the norm $\|y\|_{L^2(0,T;\mathbb{C}^n)}$ is bounded. Thus, $\mathfrak{E}$ is bounded. Since

$$\begin{aligned}\langle \mathfrak{E}u, u\rangle_W &= \left(\int_0^T \langle y(t), u\rangle_W y(t)d(t)\right)^* Wu = \int_0^T \langle y(t), u\rangle_W y(t)^* Wu dt \\ &= \int_0^T |\langle y(t), u\rangle_W|^2 dt \geq 0\end{aligned}$$

for all $u \in \mathbb{C}^n$ holds, $\mathfrak{E}$ is positive. Finally, we deduce from

$$\begin{aligned}\langle \mathfrak{R}u, \tilde{u}\rangle_W &= \int_0^T \langle y(t), u\rangle_W \langle y(t), \tilde{u}\rangle_W d(t) = \left\langle \int_0^T \langle y(t), \tilde{u}\rangle_W y(t)dt, u\right\rangle_W \\ &= \langle \mathfrak{E}\tilde{u}, u\rangle_W = \langle u, \mathfrak{E}\tilde{u}\rangle_W\end{aligned}$$

for all $u, \tilde{u} \in \mathbb{C}^n$ that $\mathfrak{E}$ is self-adjoint
. Utilizing the operator $\mathfrak{E}$, Equation (3.4.9) can be written as an eigenvalue problem.

$$\mathfrak{E}u = \lambda u \quad \text{in} \quad \mathbb{C}^n.$$

From Lemma 3.4.1, it follows that $\mathfrak{E}$ possesses eigenvetors $\{u_k\}_{k=1}^n$ and associated real eigenvalues $\{\lambda_k\}_{k=1}^n$ such that

$$\mathfrak{E}u_k = \lambda_k u_k \quad \text{for} \quad 1 \leq k \leq n \quad \text{and} \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0. \tag{3.4.11}$$

Note that

$$\int_0^T |\langle y(t), u_k \rangle_W|^2 dt = \int_0^T \langle \langle y(t), u_k \rangle_W y(t), u_i \rangle_W dt = \langle \mathfrak{E}u_i, u_k \rangle_W = \lambda_k \|u_k\|_W^2 = \lambda_k$$

for $k = \{1, \ldots, n\}$ so that u_k solves Equation (3.4.6).

Proceeding as in Section 3.1 and 3.3 we obtain the following result.

Theorem 3.4.2. *Suppose $y \in C([0,T]; \mathbb{C}^n)$ is the unique solution to a dynamical system. Then the minimization problem Q4 is solved by the POD basis of rank d described by the eigenvectors $\{u_k\}_{k=1}^d$ of $\mathfrak{E}$ corresponding to the leading d eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d$.*

Remark 3.4.3. *Let $\mathscr{R} : L^2(0,T) \to \mathbb{C}^n$ be linear and bounded operator defined by*

$$\mathscr{R}v = \int_0^T v(t) y(t) dt \quad \textit{for} \quad v \in L^2(0,T).$$

The adjoint $\mathscr{R}^* : \mathbb{C}^n \to L^2(0,T)$ satisfying

$$\langle \mathscr{R}^* u, v \rangle_{L^2(0,T)} = \langle u, \mathscr{R}u \rangle_W \quad \textit{for all} \quad (u,v) \in \mathbb{C}^n \times L^2(0,T)$$

is given as

$$(\mathscr{R}u)(t) = \langle u, y(t) \rangle_W \quad \textit{for} \quad u \in \mathbb{C}^n \quad \textit{and almost all} \quad t \in [0,T].$$

we then have

$$\mathscr{R}\mathscr{R}^* u = \int_0^T \langle u, y(t) \rangle_W dt = \int_0^T \langle y(t), u \rangle_W y(t) dt = \mathfrak{E}u$$

for all $u \in \mathbb{C}^n$, i.e.,$\mathfrak{E} = \mathscr{R}\mathscr{R}^*$ holds. Furthermore,

$$(\mathscr{R}^*\mathscr{R}v)(t) = \left\langle \int_0^T v(s)y(s)ds, y(t) \right\rangle_W = \int_0^T \langle y(s), y(t)\rangle_W \, v(s)ds = (\mathscr{H}V)(t)$$

for all $v \in L^2(0,T)$ and almost all $t \in [0,T]$. Hence, $\mathscr{H} = \mathscr{R}^*\mathscr{R}$. It can be shown that the operator $\mathscr{H}$ is linear,bounded, positive and self-adjoint. Further, $\mathscr{H}$ is compact. Hence, we can compute for POD basis as follows:

$$\mathscr{H}v_k = \lambda_k v_k \quad \text{for} \quad 1 \leq k \leq d, \quad \lambda_1 \geq \cdots \geq \lambda_d > 0, \quad \int_0^T v_k(t)v_l(t)dt = \delta_{kl} \qquad \textbf{(3.4.12)}$$

and set

$$u_k = \frac{1}{\sqrt{\lambda_k}}\mathscr{R}v_k = \frac{1}{\sqrt{\lambda_k}}\int_0^T v_k(t)y(t)dt \quad \text{for} \quad k = 1, \ldots, d.$$

Note that Equation (3.4.12) is an eigenvalue problem in the infinite-dimensional function space $L^2(0,T)$.

4 Discrete Empirical Interpolation Method (DEIM)

In chapter 3, we have discussed POD method which is core in finding reduced-order systems. However, this POD-Galerkin technique can only approximate solutions to problems with linear terms because it cannot reduce the complexity of nonlinear terms.
Thus, it is efficient to use POD approximation with DEIM when dealing with nonlinear problems.

The Discrete Empirical Interpolation Method (DEIM) is an improvement of the POD method that minimizes the computational complexity for generating reduced-order models for PDEs.

Consider nonlinear ODE in the form

$$\frac{d}{dt}\bar{y} = A\bar{y}(t) + N(\bar{y}(t)), \quad y(0) = y_0 \tag{4.0.1}$$

Let $U_d \in \mathbb{C}^{n\times d}$ be matrix consisting of orthonormal basis of dimension d, $d < n$.
Then on applying POD combined with Galerkin projection to Equation (4.0.1), it reduces into a new equation with fewer unknowns.

$$\frac{d\bar{a}}{dt} = U_d^* A U_d \bar{a}(t) + U_d^* N(U_d \bar{a}(t)) \tag{4.0.2}$$

Notice that the computational complexity of evaluating the nonlinear term $U_d^* N(U_d \bar{a}(t))$ still depends on the full dimension n. Thus, it is necessary to eliminate this dependence by combining POD-Galerkin approach with DEIM.

Assume that $\{\wp_1, \wp_2, \ldots, \wp_m\} \in \mathbb{C}^n$ is the set of nonlinear snapshots and $\bar{N} = [\wp_1, \wp_2, \ldots, \wp_m] \in \mathbb{C}^{n\times m}$ denote the nonlinear snapshot matrix. Then, we use SVD on $\bar{N}$ to compute POD basis of rank $s \leq \min\{n, m\}$ of the nonlinear term. Suppose SVD of $\bar{N}$ is $\bar{N} = Z\Sigma W^*$ where $Z = [z_1, z_2, \ldots, z_n] \in \mathbb{C}^{n\times n}$, $W = [w_1, w_2, \ldots, w_m] \in \mathbb{C}^{m\times m}$ and $\tilde{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \ldots \sigma_m) \in \mathbb{R}^{m\times m}$.
Therefore, the first s columns of matrix Z is the POD basis of rank s of the nonlinear term

denoted by Z_s. Then we can approximate the nonlinear function $N(U_d\bar{a}(t))$ by a subspace spanned by the basis $\{z_1, z_2, \ldots z_s\}$ which is written in the form

$$N(U_d\bar{a}(t)) \approx Z_s c(t) \tag{4.0.3}$$

where $c(t)$ denote the corresponding coefficient vector. We apply DEIM technique here to specify $c(t)$ by selecting the s rows of Equation (4.0.3). $c(t)$ is found by using the following interpolation method. Let's consider the matrix

$$P = [e_{\eta 1}, \ldots, e_{\eta s}] \in \mathbb{C}^{n\times s} \tag{4.0.4}$$

where $e_{\eta i} = [0, \ldots, 0, 1, 0, \ldots, 0]^T \in \mathbb{C}^n$. Assuming that $P^T Z$ is nonsingular, we can solve for $c(t)$ from

$$P^T N(U_d\bar{a}(t)) = \underbrace{(P^T Z_s)}_{s\times s} c(t)$$

and so,

$$c(t) = (P^T Z_s)^{-1} P^T N(U_d\bar{a}(t))$$

Hence, the approximation of Equation (4.0.3) is given by

$$N(U_d\bar{a}(t)) \approx Z_s c(t) = Z_s (P^T Z_s)^{-1} \underbrace{P^T N(U_d\bar{a}(t))}_{s\times 1}$$

but when the nonlinear function F is componentwise, $f(t)$ becomes

$$N(U_d\bar{a}(t)) \approx Z_s c(t) = Z_s (P^T Z_s)^{-1} \underbrace{N(P^T U_d\bar{a}(t))}_{s\times 1}$$

Discrete Empirical Interpolation Method (DEIM) approximates the nonlinear terms by computing the projection basis from POD method and selecting the interpolation indices by a greedy algorithm. We can obtain the interpolation indices $\eta_1, \ldots, \eta_s$ using the following DEIM algorithm.

Algorithm 3 DEIM

INPUT:$\{z_\ell\}_{\ell=1}^{n} \in \mathbb{C}^n$ linearly independent
OUTPUT: $\vec{\eta}_s = [\eta_1, \ldots, \eta_s]^T \in \mathbb{R}^s$
1. $[|\rho|, \eta_1] = \max\{|z_1|\}$
2. $\bar{Z} = [\bar{z}_1], \bar{P} = [e_{\eta_1}], \vec{\eta} = [\eta_1]$
3. for $\ell = 2 : n$ do
4. solve $(P^T Z)_C = P^T z_\ell$;
5. $r = z_\ell - Z_c$
6. $[|\rho|, \eta_\ell] = \max\{|r|\}$
7. $Z \leftarrow [Z \quad z_\ell], P \leftarrow [P \quad e_{\eta_\ell}], \vec{\eta} \leftarrow \begin{bmatrix} \vec{\eta} \\ \eta_\ell \end{bmatrix}$
8. end for
9. P = $\bar{P}(:, 1 : s), \vec{\eta}_s = \vec{\eta}(1 : s)$

From Algorithm 3, the procedure is used to construct a set of indices on the input basis. We begin the process by choosing the first interpolation index η_1 that corresponds to the first input basis z_1 entry with the biggest magnitude. The remaining indices η_l for $l = 2, 3, \ldots, s$ are chosen from the entry of the residual $r = z_\ell - Z_c$ with the largest magnitude. The input basis $\{z_l\}_{l=1}^{s}$, which is linearly independent, guarantees that in each iteration, both r and ρ are nonzero vectors. This implies that $P^T Z_s$ is always nonsingular and therefore the DEIM procedure is clearly stated. Also, this insinuates that the interpolation indices $\{\eta_\ell\}_{\ell=1}^{n}$ are not repeated. Then the output matrix P is used to build a low-dimensional estimation of the nonlinear term. After that, POD technique is used together with the DEIM technique to formulate a low order model.

$$\frac{d\bar{a}}{dt} = U_d^* A U_d \bar{a}(t) + U_d^* Z_s (P^T Z_s)^{-1} P^T N(U_d \bar{a}(t))$$

5 Application of Model Reduction on PDEs

Let's consider a system of 1 dimensional nonlinear PDE of the form

$$u_t = N(u, u_x, u_{xx}, x, t) \tag{5.0.1}$$

where $N(.)$ describe nonlinear evolution and the subscripts denote partial differentiation. We also have both initial and boundary conditions defined on a domain $x \in [-T, T]$.

First, consider a standard spatial discretization of Equation (5.0.1) evaluated at n discrete points for n large enough.

$$u(x,t) \to u(x_i,t) = u_i \quad \text{for} \quad i = 1, \ldots, n$$

using standard finite difference formulas, we can evaluate the spatial derivatives using neighboring spatial points.

$$u_x = \frac{u(x_{i+1},t) - u_(x_{i-1},t)}{2\Delta x}$$

$$u_{xx} = \frac{u(x_{i+1},t) - 2u_(x_i,t) + u_(x_{i-1},t)}{\Delta x^2}$$

Therefore, the governing PDE (5.0.1) transforms to a set of n ODEs.

$$\frac{du_i}{dt} = N(u(x_{i+1},t), u(x_i,t), u(x_{i-1},t), x_i, t), \quad \text{for} \quad i = 1, 2, \ldots, n. \tag{5.0.2}$$

We then apply Galerkin projection

$$u(x_i,t) = \sum_{i=1}^{d} a_i(t)\Psi_i(x) \tag{5.0.3}$$

where $\Psi_i(x)$ form a set of $n >> 1$ basis modes and $a(t)$ denote the unknown time dynamics

Substituting Equation (5.0.3) into the Equation (5.0.2) gives

$$\sum \Psi_i \frac{da_i}{dt} = N(\sum a_i \Psi_i, \sum a_i (\Psi_i)_x, \sum a_i (\Psi_i)_{xx}, x, t), \quad i = 1, \ldots, n \tag{5.0.4}$$

The set of the basis functions should be orthonormal to each other, that is,

$$\langle \Psi_i, \Psi_j \rangle = \delta_{ij} = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$$

where δ_{ij} is the kronecker delta function.
$\langle \Psi_i, \Psi_j \rangle$ is the inner product defined as

$$\langle \Psi_i, \Psi_j \rangle = \int_{-T}^{T} \Psi_i \Psi_j^* dx$$

where * denote complex conjugate.
Multiplying both sides of Equation (5.0.4) by $\Psi_j(x)$ and integrating over the domain $x \in [-T, T]$, we get

$$\frac{da_i}{dt} = \langle N(\sum a_j \Psi_j, \sum a_j (\Psi_j)_x, \sum a_j (\Psi_j)_{xx}, x, t), \Psi_i \rangle, \quad i = 1, \ldots, n \tag{5.0.5}$$

Hence, we can now solve the reduced system Equation (5.0.5) and use its reduced-order solution to approximate the solution of the high dimensional system Equation (5.0.2). Let's demonstrate this reduction method further by solving the fisher's equation and diffusion-advection equation

5.1 Fisher's Equation

Fisher's equation is a nonlinear parabolic PDE. It is writen in the form:

$$u_t = \alpha u(1-u) + D u_{xx}$$

where α is the reactive factor (positive), D is the diffusion coefficient (positive), t and x are time and spatial location respectively and $u = u(x,t)$ is the state variable. The term $\alpha u(1-u)$ is also called logistic growth. This equation was first proposed by Fisher as a model for propagation of a gene within a population.

Here, consider the equation

$$u_t = u(1-u) + u_{xx}$$

with $x \in [-20, 20]$, $t \in [0, 5]$

subject to

$$\text{B.C} : u(-20, t) = u(20, t) = 0$$
$$\text{1.C} : u(x, 0) = 2\text{sech}(x)$$

By using finite difference (FD) discretization evaluated at n discrete points, we have

$$\frac{du_i}{dt} = u_i - u_i^2 + \frac{d^2 u_i}{dx^2}, \quad i = 1, \ldots, n \tag{5.1.1}$$

We then apply Galerkin projection

$$u(x_i, t) = \sum_{i=1}^{d} a_i(t)\Psi_i(x)$$

to this n set of ODEs and we get

$$\sum \Psi_i \frac{da_i}{dt} = \sum a_i \Psi_i - (\sum a_i \Psi_i)^2 + (\sum a_i \Psi_i)_{xx}, \quad i = 1, \ldots, n$$

multiplying both sides by Ψ_j ,we get the following reduced system

$$\frac{da_i}{dt} = \langle \Psi_j, \Psi_i \rangle a_i - \langle (\Psi_j)^2, \Psi_i \rangle a_i^2 + \langle (\Psi_j)_{xx}, \Psi_i \rangle a_i, \quad i = 1, \ldots n. \tag{5.1.2}$$

5.1.1 Numerical results

Let n = 4000, then the Equation (5.1.1) contains 4000 ordinary differential equations. The following figure represents the solution for the full order model using 4000 dimension.

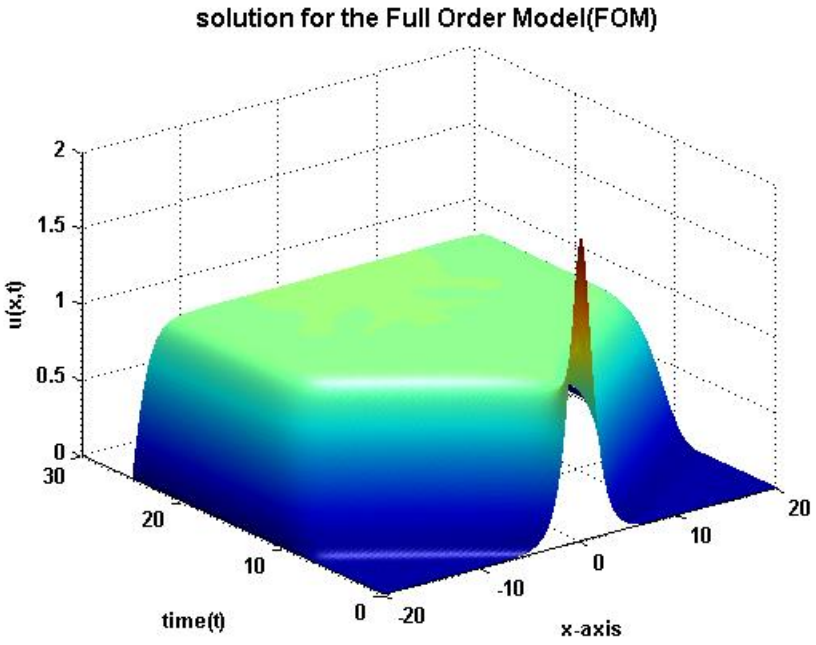

Figure 1. Numerical solution of the Full Order Model (FOM) of Fisher's equation

Now, let's use the 4000 solutions to construct reduced basis by using SVD. We have to take snapshots of the system at m time instances. Suppose $m = 100$, then we have a 4000×100 snapshot matrix.
We compute the SVD for the snapshot matrix 4000×100 to determine the POD basis of snapshots . The SVD decomposes the matrix into product of three matrices as shown in chapter 2. In this case, the blockwise diagonal matrix Σ consists of 100 non zero singular values arranged in descending order. Figure 2 shows how the singular values are distributed which are obtained from full-order finite difference discretization of the fisher's equation.

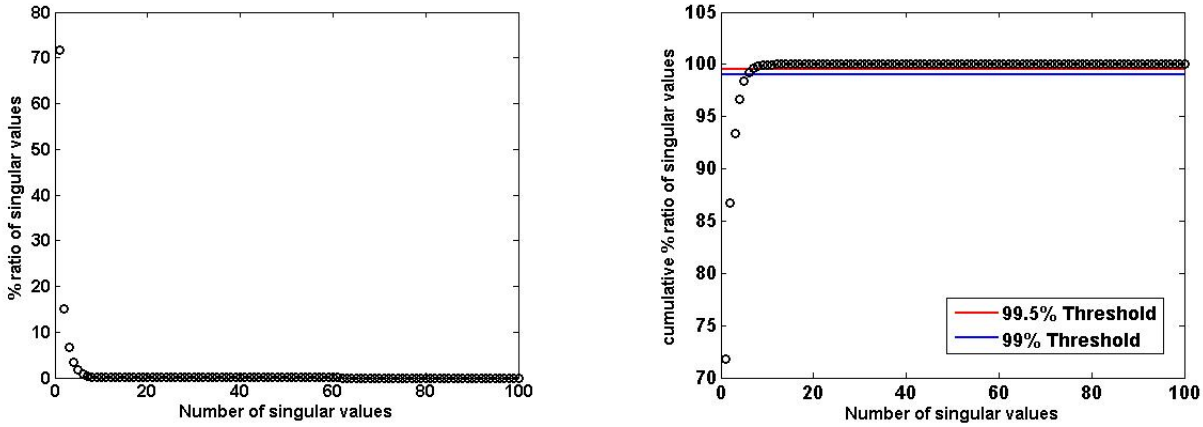

Figure 2. Distribution of singular values from the full order FD discretization of the Fisher's equation

In Figure 2, the rapid decay occurs around the first 5 singular values of the 100 snapshots taken. This implies that the first 5 left singular vectors can be used to represent the dominant features of the whole set of snapshots. Also, threshold is used to determine the percentage of the information we want to retain. In this case, the first 5 singular values retains 95% of the information.

So, the first $d = 1,2,3,4,5$ columns of matrix U from SVD are considered as the modes of different POD basis sets. The values of d are considered as the POD reduced system. Here, we apply the POD-Galerkin approach as discussed previously for each reduced system.
For instance, if we take $d = 1$, then 1 POD mode is used to approximate the numerical solution. The Galerkin projection becomes

$$u(x,t) = a_1(t)\Psi_1(x)$$

Then Equation (5.1.2) becomes

$$\begin{aligned}\frac{da_1}{dt} &= \langle \Psi_1, \Psi_1 \rangle a_1 - \left\langle (\Psi_1)^2, \Psi_1 \right\rangle a_1^2 + \left\langle (\Psi_1)_{xx}, \Psi_1 \right\rangle a_1 \\ &= \alpha a_1 - \beta a_1^2 + \eta a_1\end{aligned}$$

where $\alpha = \langle \Psi_1, \Psi_1 \rangle$, $\beta = \left\langle (\Psi_1)^2, \Psi_1 \right\rangle$ and $\eta = \left\langle (\Psi_1)_{xx}, \Psi_1 \right\rangle$.

Hence, we only have 1 first order ordinary differential equation to solve. This is easier and takes less CPU time to solve compared to the system with 4000 ordinary differential equations.

Figure 3 shows the numerical solution of the POD reduced system of the Fisher's equation with different POD modes. The results are obtained from Matlab ODE 45.

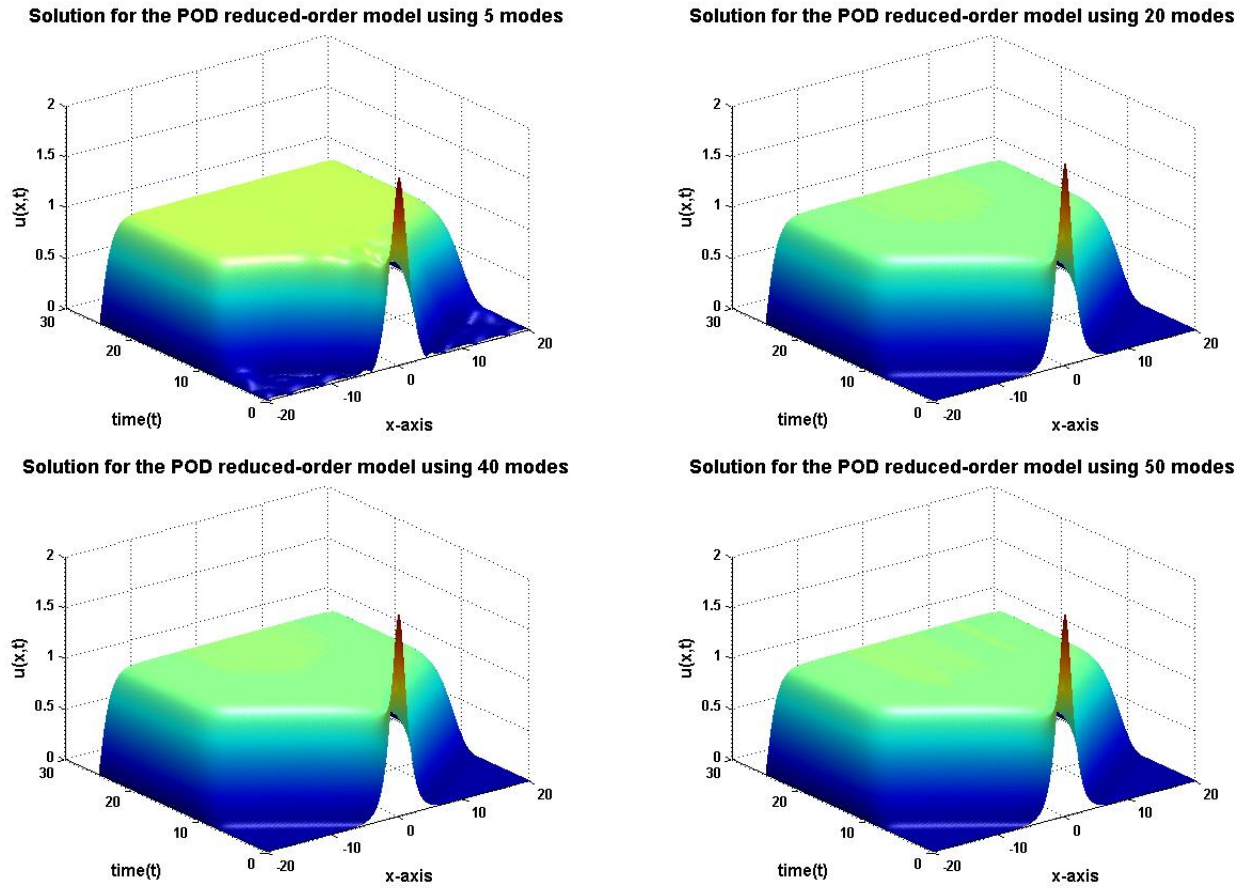

Figure 3. Solution of POD reduced model of Fisher's equation with different POD modes

From Figure 3, we observe that the dominant features of the solution of Fisher's equation are preserved after reducing the system using POD.

Now let's observe numerical solution of Fisher's equation using POD combined with DEIM.

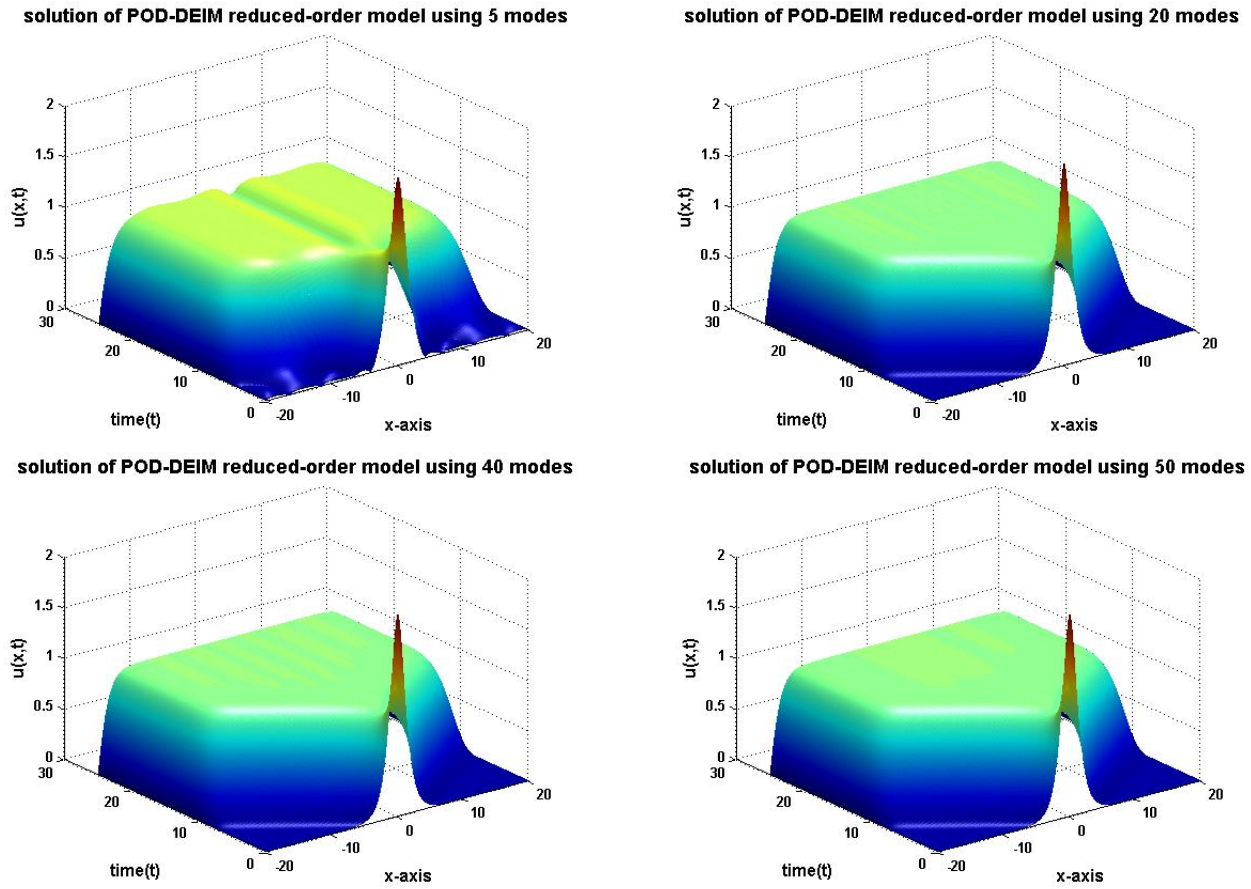

Figure 4. Solution of POD-DEIM reduced model of Fisher's equation with different modes

Again, the dominant features are still preserved even after using POD combined with DEIM. Now, lets compare POD and POD-DEIM simulation time and their relative error using the Table 1.

Model	**Relative average error**		**CPU time (sec)**		**% information retained**
FOM	0		7.481475		100%
	POD	**POD-DEIM**	**POD**	**POD-DEIM**	
ROM (5 modes)	0.6503668	2.209273	0.258731	0.128493	99%
ROM (20 modes)	7.2967e-04	0.0367634	0.357261	0.138304	99.78%
ROM (40 modes)	3.6928e-04	0.0013877	0.622867	0.186537	99.943%
ROM (50 modes)	3.3074e-04	0.0010444	0.788952	0.204632	99.971%

Table 1. Comparison between Relative average error, CPU time and percentage information retained for POD and POD-DEIM approach using different modes.

From Table 1 we observe that the relative error due to the POD reduced-order model with 20 modes which is approximately $\mathcal{O}(10^{-4})$ is less than than the error of the POD-DEIM reduced-order system with 20 modes which is approximately $\mathcal{O}(10^{-2})$. Also, as we increase the number of modes, the relative error decreases. Therefore, the POD and POD-DEIM methods becomes more accurate as the number of modes increases.
The CPU time of the POD reduced-order model decreases by a factor of approximately 29 while the simulation time of the POD-DEIM reduced-order model decreases by a factor of approximately 58. Hence, the POD-DEIM reduced-order model spends less time than the POD reduced-order model making it computationally efficient.

This trend is illustrated by the plots in Figure 5. We can see why the Model reduction technique is a trade off between accuracy and computational complexity.

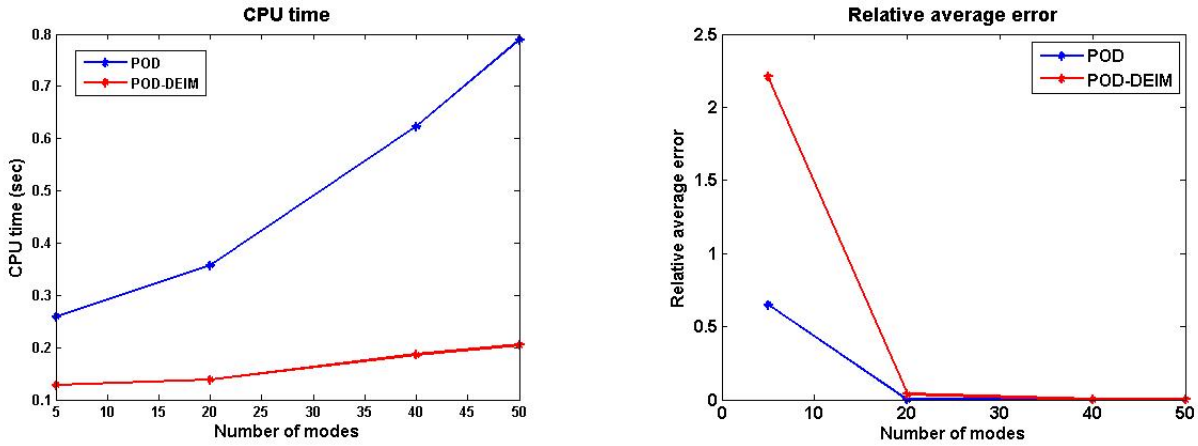

Figure 5. CPU time and Relative average error of POD and POD-DEIM approach

5.2 Advection-Diffusion Equation

The Advection-Diffusion equation is a PDE combining advection (plug-flow motion) and a diffusion term. It is applied in fluid dynamics to model diffusive waves.

One dimensional advection-Diffusion equation is given by

$$u_t = -au_x + \alpha u_{xx}$$

where α denote the coefficient of diffusivity and a is the advection velocity.
Let's solve this equation with $a = 1$ and $\alpha = 0.01$.

The Initial Boundary Value Problem is

$$u_t = -u_x + 0.01u_{xx}$$

subject to B.C: $u(-40,t) = u(40,t) = 0$ and I.C: $u(x,0) = 2\text{sech}(x)$

This equation is linear therefore we will use POD-Galerkin method only. Using the same technicalities used to reduce the Fisher's equation, we end up with the following reduced system.

$$\frac{da_k}{dt} = 0.01\left\langle \frac{d^2\Psi_k}{dx^2}, \Psi_k \right\rangle a_k - \left\langle \frac{d\Psi_k}{dx}, \Psi_k \right\rangle a_k, \quad k = 1, \ldots, n.$$

5.2.1 Numerical results

Now, we use Matlab ODE 45 to compute the solution for this reduced system. The results are as follows.

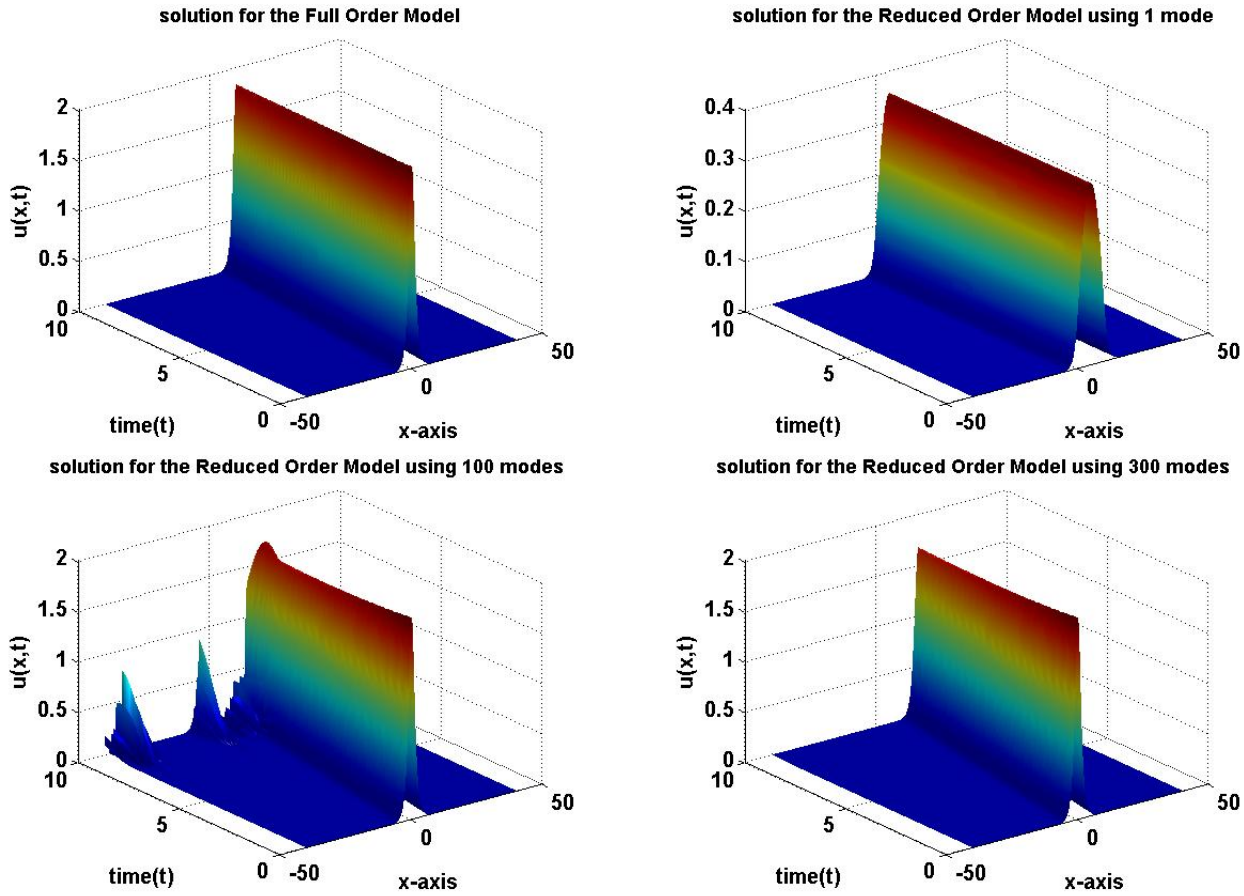

Figure 6. Solutions for the Full Order Model and POD reduced order model of Advection-Diffusion equation with different modes

From Figure 6, we observe that using 1 mode will not produce best results. Even as we increase the number of modes to 100 modes, the solution is not yet sufficient to describe the solution of Advection-Diffusion equation. In fact by using 300 modes, we can approximate our solution since the graph appear similar to the graph of full order model. At this point, the error is smaller than using 1 mode as shown in Table 2.

Model	Relative average error	CPU time(sec)	% information retained
FOM	0	3.740166	100%
ROM (1 mode)	0.8998	0.517654	90%
ROM (100 modes)	0.0723	0.722157	95.5%
ROM (200 modes)	0.0692	0.856862	99.51%
ROM (300 modes)	0.0010	0.935541	99.95%

Table 2. Relative average error and CPU time of solutions of Full Order Model (FOM) and POD reduced order models with different modes

As we have previously seen in fisher's equation, as the number of modes increases, the relative error decreases and CPU time increases. For instance, the error of the ROM with 300 modes, which is approximately $\mathcal{O}(10^{-3})$ is less than the error of ROM with 1 mode, which is approximately $\mathcal{O}(10^{-1})$.
The CPU time of the ROM decreases by a factor of approximately 7. Hence, POD method is computationally efficient.

Figure 7 summarizes Table 2 in graphical form.

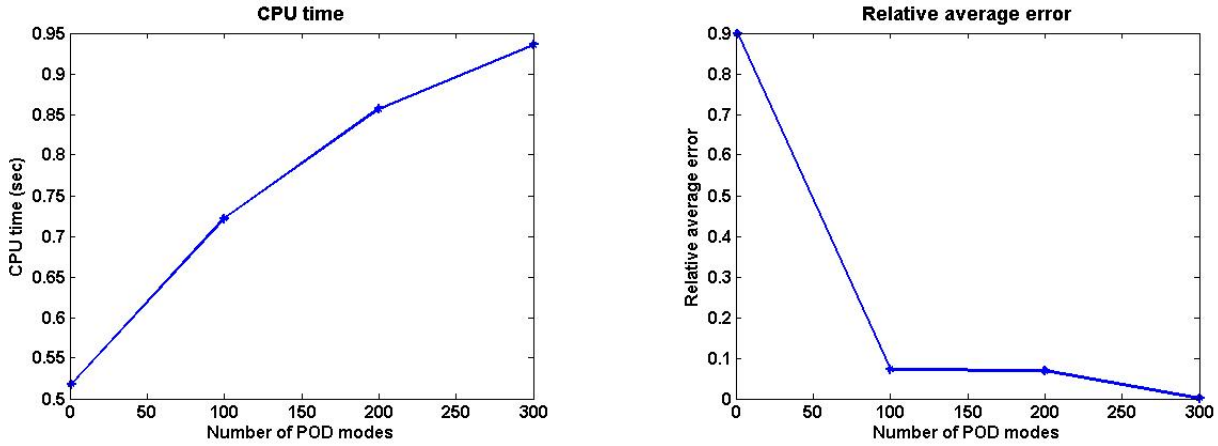

Figure 7. CPU time and Relative average error of the POD method.

6 Conclusion and Future Research

6.1 Conclusion

This research paper focused on applying Model Reduction techniques to reduce the computational complexity of PDEs. We used Finite Difference (FD) method to discretize the PDE problem, which resulted to high fidelity dynamical system. We then used SVD to construct a set of POD basis and then proceeded with Galerkin projection to deduce a POD reduced system equation. This POD-Galerkin approach was only limited to linear PDEs. Therefore, when dealing with nonlinear PDEs, POD-Galerkin approach was not efficient since the computational complexity of the nonlinear terms depends on the original system. We then combined POD with DEIM.

In particular, we applied POD-DEIM approach to Fisher's equation. Dominant features of the full-order system of Fisher's equation were preserved using different POD and POD-DEIM modes. We have shown that POD-DEIM approach is the most effective in minimizing computational complexity of the Fisher's equation as compared to POD-Galerkin approach. However, the relative average error due to the POD-DEIM approach was higher than the POD-Galerkin approach. Hence, there was a trade off between accuracy and computational complexity.

6.2 Future Research

The goal of this research project was to study Model Reduction techniques called POD and DEIM and apply these methods appropriately to PDE problems. In particular, we concentrated on one dimensional PDEs. This can be extended to 2 and 3 dimensional partial differential equations and study their stability. Additionally, POD method can be extended to real data in order to study certain aspects.

Bibliography

[1] Steven L. Brunton, J.Nathan Kutz: *Data-Driven Science and Engineering:machine learning,Dynamical systems and control*, 2019

[2] K. Kunisch, S. Volkwein: *Galerkin Proper Orthogonal Decomposition methods for parabolic problems.* Numerische Mathematik 90 (2001) 117-148.

[3] John L.Lumley: *The Proper Orthogonal Decomposition in the analysis of turbulent flows*, November 2003.

[4] P. Benner, A. Cohen, M. Ohlberger, A.K. Willcox: *Model Reduction and Approximation .* Theory and Algorithm, Computational Science and Engineering, SIAM, 2017.

[5] Y.H. Cao, J. Zhu, Z.D. Luo, I.M. Navon: *A reduced order approach to four-dimensional variational data assimilation using proper orthogonal decomposition.* International Journal for Numerical Methods in Fluids 53 (2007) 1571-1583.

[6] *Modal analysis of fluid flows: Applications and outlook*, AIAA Journal Vol 58, No 3, March 2020.

[7] Philip Holmes, John L. Lumley, Gahl Berkooz and Clarence W.Rowley: *Turbulence, coherent structures, dynamical systems and symmetry.* Second edition.

[8] J. An, Z.D. Luo: *A reduced finite difference scheme based on POD basis and posteriori error estimate for the three dimensional parabolic equation.* Acta Mathematica Scientia 31A (3) (2011) 769-775.

[9] Zlendog Luo, Goong Chen: *Proper Orthogonal Decomposition methods for partial differential equations.* 2019.

[10] L. Sirovich: *Turbulence and the dynamics of coherent structures.* Part I- III, quarterly of applied mathematics 45(1987), 561-590.

[11] Chatterjee, Anindya: *An introduction to the proper orthogonal decomposition*, 2000.

[12] Kunihiko Taira, Maziar S. Hemati, Stephen L. Brunton, Yiyang Sun, Karthik Duraisany, Shervin Bagheni, Scott T. M. Dawson and Chi - An Yeh. *Modal analysis of fluid flows: Applications and outlook.* AIAA JOURNAL,vol.58, No. 3, March 2020.

[13] Toni Lassila, Andrea Manzoni, Alfio Quarteroni and Gianluigi Rozza. *Model Order Reduction in fluid dynamics:Challenges and perspectives.* 2014.

[14] Abbasi, Farshid, and Javad Mohammadpour: *Nonlinear model order reduction of Burgers' equation using proper orthogonal decomposition.* American Control Conference (ACC), 2015.

[15] S. Volkwein: *Model reduction using proper orthogonal decomposition.* Lecture notes, April 2008.

[16] Saifon Chaturantabut and Danny C. Sorensen: *Nonlinear Model Reduction via Discrete Empirical Interpolation.* SIAM J.SCI.COMPUT Vol. 32, No. 5, pp. 2737-2764, 2010.

[17] Suchinthra Rungpitaxmana and Saifon Chaturantabut *Model Reduction for Fisher's Equation with an Error Bound.* Thai Journal of Mathematics, Vol. 19, No. 4, pg. 1729-1762, 2021.

[18] Norapon Sukuntee and Saifon Chaturantabut: *Model Order Reduction for Sine-Gordon Equation using POD and DEIM.* Thai Journal of Mathematics: 222-256, 2018.

[19] Ioannis P Stavroulakis, Stepan A Tersian: *Partial Differential Equations*, (second Edition) An introduction with mathematica and MAPLE.

[20] Giuseppe Regal and Hans Troger: *Dimension Reduction of Dynamical Systems:* Methods, Models, Applications, Article in Nonlinear Dynamics. January 2005.

[21] Michael Reed, Barry Simon: *Methods of modern mathematical physics*, Functional Analysis.

[22] Astrid, P.: *Reduction of process simulation models: a proper orthogonal decomposition approach,* Eindhoven: Technische Universiteit Eindhoven DOI: 10.6100/IR581728, 2004.

Appendix: Matlab codes

Fisher's Equation

Fishers.m

```
clear all; clc; close all;
tic
m = 0;
x = linspace(-20,20,4000);
L=max(x)-min(x); n=length(x);
t = linspace(0,25,100);
p=length(t);
sol = pdepe(m,@Fisherspde,@Fisherspdeic,@Fisherspdebc,x,t);
usol = sol(:,:,1);

figure;
surfl(x,t,usol,'light');
title('solution for the Full Order Model(FOM)');
xlabel('x-axis');
ylabel('time(t)');
zlabel('u(x,t)');
shading interp
set(gca,'FontSize',10,'fontweight','b','fontname','arial')
    set(gcf, 'Color', 'w');
toc

Y = usol.';
[U,S,V]=svd(Y);

figure;
plot((diag(S)/sum(diag(S))*100),'ko','Linewidth',2);
xlabel('Number of singular values');
ylabel('% ratio of singular values');
set(gca,'FontSize',12,'fontweight','b','fontname','arial')
    set(gcf, 'Color', 'w');

x1=[0 20 40 60 80 100];
y1=[99.5 99.5 99.5 99.5 99.5 99.5];
```

```
34
35 x2=[0  20  40  60  80  100];
36 y2=[99  99  99  99  99  99];
37
38 figure;
39 plot(x1,y1,'r','Linewidth',2);
40 hold on
41 plot(x2,y2,'b','Linewidth',2);
42 legend('99.5% Threshold','99% Threshold');
43 hold on
44 plot(cumsum(diag(S)/sum(diag(S))*100),'ko','Linewidth',2);
45 xlabel('Number of singular values');
46 ylabel('cumulative % ratio of singular values');
47 set(gca,'FontSize',13,'fontweight','b','fontname','arial')
48      set(gcf, 'Color', 'w');
49
50 k=(2*pi/L)*[0:(n/2-1) -n/2:-1].';
51
52 tic
53 %POD Method
54 Y = usol.';
55 [U,S,V] = svd(Y);
56 modes = 20;
57 phi = U(:,1:modes);
58
59 for j=1:modes
60     phixx(:,j) = -ifft((k.^2).* fft(phi(:,j)));
61     a0(j)= (2*sech(x))*conj(phi(:,j));
62 end
63 L=phi'*phixx;
64
65 [t,asol]=ode45('Fishers_POD',t,a0,[],phi,L);
66
67 %Reconstructs the solution
68 us1=zeros(n,length(t));
69 for j=1:length(t)
70      for jj=1:modes
71          us1(:,j)=us1(:,j)+asol(j,jj)*phi(:,jj);
72      end
73 end
74
75 figure;
```

```
surfl(x,t,abs(us1.'),'light');
title('Solution for the POD reduced-order model using 20
    modes');
xlabel('x-axis');
ylabel('time(t)');
zlabel('u(x,t)');
shading interp
set(gca,'FontSize',15,'fontweight','b','fontname','arial')
     set(gcf, 'Color', 'w');
toc

%POD combined with DEIM
Y=((abs(usol.').^2).*(usol.'));
Y1=Y(:,1:p);
tic
[W,T,Z]=svd(Y1);
Wk=W(:,1:modes);

%DEIM algorithm
[n,m]=size(Wk);
gamma=zeros(m, 1);
[~, gamma(1)]=max(abs(Wk(:, 1)));
e=eye(n);
P=e(:,gamma(1));
B(:, 1)=Wk(:, 1);
for l=2:m
     uL=Wk(:, l);
     c=(P'*B)\(P'*uL);
     r=uL-B*c;
     [~, gamma(l)]=max(abs(r));
     B(:, l)=uL;
     P(:, l)=e(:, gamma(l));
     Q(:, l)=phi(gamma(l),:);
     R(:, l)=Wk(gamma(l),:);
end

gamma= sort(gamma);
P=e(:, gamma);
Q=phi(gamma,:);
R=Wk(gamma,:);
M=((phi'*Wk)/(R));

```

```
117 [t,asol]=ode45('Fishers_DEIM',t,a0,[],phi,L,Wk,P,M,Q);
118
119 us2=zeros(n,length(t));
120 for j=1:length(t)
121     for jj=1:modes
122         us2(:,j)=us2(:,j)+asol(j,jj)*phi(:,jj);
123     end
124 end
125
126 figure;
127 surfl(x,t,abs(us2.'),'light');
128 title('solution of POD-DEIM reduced-order model using 20
      modes');
129 xlabel('x-axis');
130 ylabel('time(t)');
131 zlabel('u(x,t)');
132 shading interp;
133 set(gca,'FontSize',10,'fontweight','b','fontname','arial')
134     set(gcf, 'Color', 'w');
135 toc
136
137 T1=usol.';
138 T2=us1;
139 T3=us2;
140 q=20;
141 output=zeros(1,q);
142 output1=zeros(1,q);
143 firsterror=norm(abs(T1(:,1)-T2(:,1)));
144 firsterror1=norm(abs(T1(:,1)-T3(:,1)));
145 output(1)=firsterror;
146 output1(1)=firsterror1;
147 sum = 0;
148 sum1 = 0;
149 for q=1:length(t)
150     output(q)=norm(abs(T1(:,q)-T2(:,q)))/(norm(abs(T1(:,q)))
        );
151      output1(q)=norm(abs(T1(:,q)-T3(:,q)))/(norm(abs(T1(:,q)
         )));
152      T=output;
153      O=output1;
154      sum=sum+T(q);
155      sum1=sum1+O(q);
```

```
end
relativerror=(1/20)*sum;
relatverror1=(1/20)*sum1;

%plotting Relative error aand computation time

x2=[5 20 40 50];

y2=[0.258731 0.357261 0.622867 0.788952];
z2=[0.6503668 0.00072967 0.00036928 0.00033074];

y3=[0.128493 0.138304 0.186537 0.204632];
z3=[2.209273 0.0367634 0.0013877 0.0010444];

figure;
plot(x2,y2,'-b*','Linewidth',2);
hold on
plot(x2,y3,'-r*','Linewidth',2);
title('CPU time');
xlabel('Number of modes');
ylabel('CPU time (sec)');
legend('POD','POD-DEIM');
set(gca,'FontSize',15,'fontweight','b','fontname','arial')
    set(gcf, 'Color', 'w');

figure;
plot(x2,z2,'-b*','Linewidth',2);
hold on
plot(x2,z3,'-r*','Linewidth',2);
title('Relative average error');
xlabel('Number of modes');
ylabel('Relative average error');
legend('POD','POD-DEIM');
set(gca,'FontSize',15,'fontweight','b','fontname','arial')
    set(gcf, 'Color', 'w');
```

Fisherspde.m

```
function [c,f,s] = Fisherspde(x,t,u,DuDx)
c = 1;
f = DuDx;
s = u-u.^2;
```

Fisherspdeic.m

```
function u0 = Fisherspdeic(x)
u0=2*sech(x);
```

Fisherspdebc.m

```
function [pl,ql,pr,qr] = Fisherspdebc(xl,ul,xr,ur,t)
pl = ul;
ql = 0;
pr = ur;
qr = 0;
```

Fishers_POD.m

```
 function rhs=Fishers_POD(t,a,dummy,phi,L)

rhs= L*a+phi'*(phi*a)-phi'*(phi*a).^2;
```

Fishers_DEIM.m

```
function rhs=Fishers_DEIM(t,a,dummy,phi,L,Wk,P,M,Q)

rhs=L*a+M*(Q*a)-M*(Q*a).^2;
```

Advection-Diffusion Equation

ADE.m

```
clear all; close all; clc
tic
x=linspace(-40,40,400);
L=max(x)-min(x); n=length(x);
t=linspace(0,10,100);
p=length(t);
m=0;
sol=pdepe(m,@ADEpde,@ADEpdeic,@ADEpdebc,x,t);
usol=sol(:,:,1);

figure;
surfl(x,t,usol,'light');
title('solution for the Full Order Model');
xlabel('x-axis');
ylabel('time(t)');
```

```
zlabel('u(x,t)');
shading interp
set(gca,'FontSize',15,'fontweight','b','fontname','arial')
    set(gcf, 'Color', 'w');
toc

Y = usol.';
[U,S,V]=svd(Y);
figure;
plot((diag(S)/sum(diag(S))*100),'ko','Linewidth',2);
xlabel('Number of singular values');
ylabel('% ratio of singular values');
set(gca,'FontSize',15,'fontweight','b','fontname','arial')
    set(gcf, 'Color', 'w');

x1=[0 20 40 60 80 100];
y1=[99.5 99.5 99.5 99.5 99.5 99.5];

x2=[0 20 40 60 80 100];
y2=[99 99 99 99 99 99];

figure;
plot(x1,y1,'r','Linewidth',2);
hold on
plot(x2,y2,'b','Linewidth',2);
legend('99.5% Threshold','99% Threshold');
hold on
plot(cumsum(diag(S)/sum(diag(S))*100),'ko','Linewidth',2);
xlabel('Number of singular values');
ylabel('cumulative % ratio of singular values');
set(gca,'FontSize',13,'fontweight','b','fontname','arial')
    set(gcf, 'Color', 'w');


k=(10/L)*[0:n/2-1 -n/2:-1].';

%POD method
tic
Y=usol.';
[U,S,V]=svd(Y);
modes = 300;
phi=U(:,1:modes);
```

```
58
59 for j=1:modes
60     phix(:,j)=ifft((1i*k).*fft(phi(:,j)));
61     phixx(:,j)=-ifft((k.^2).*fft(phi(:,j)));
62     a0(j)=(2*sech(x))*conj(phi(:,j));
63 end
64 L=0.01*phi'*phixx;
65 [t,asol]=ode45('ADE_POD',t,a0,[],phi,phix,L);
66
67 us1=zeros(n,length(t));
68 for j=1:length(t)
69     for jj=1:modes
70         us1(:,j)=us1(:,j)+asol(j,jj)*phi(:,jj);
71     end
72 end
73
74 figure;
75 surfl(x,t,abs(us1.'),'light');
76 title('solution for the Reduced Order Model using 300 modes
    ');
77 xlabel('x-axis');
78 ylabel('time(t)');
79 zlabel('u(x,t)');
80 shading interp
81 set(gca,'FontSize',15,'fontweight','b','fontname','arial')
82     set(gcf, 'Color', 'w');
83 toc
84
85 tic
86 phi=U(:,1:400);
87
88 for j=1:400
89     phix(:,j)=ifft((1i*k).*fft(phi(:,j)));
90     phixx(:,j)=-ifft((k.^2).*fft(phi(:,j)));
91     a0(j)=(2*sech(x))*conj(phi(:,j));
92 end
93 L=0.01*phi'*phixx;
94 [t,asol]=ode45('ADE_POD',t,a0,[],phi,phix,L);
95
96 us=zeros(n,length(t));
97 for j=1:length(t)
98     for jj=1:400
```

```
99              us(:,j)=us(:,j)+asol(j,jj)*phi(:,jj);
100         end
101     end
102
103     figure;
104     surfl(x,t,abs(us.'),'light');
105     title('solution for the FOM');
106     xlabel('x-axis');
107     ylabel('time(t)');
108     zlabel('u(x,t)');
109     shading interp
110     toc
111
112     T1=us;
113     T2=us1;
114     q=400;
115     output=zeros(1,q);
116     firsterror=norm(abs(T1(:,1)-T2(:,1)));
117     firsterror1=norm(abs(T1(:,10)-T2(:,10)));
118     output(1)=firsterror;
119     sum=0;
120     for q=1:length(t)
121         output(q)=norm(abs(T1(:,q)-T2(:,q)))/(norm(abs(T1(:,q)))
            );
122         T=output;
123         sum=sum+T(q);
124     end
125     relativerror=(1/400)*sum;
126     %plotting Relative error and CPU time
127
128     x2=[1 100 200 300];
129     y2=[ 0.517654 0.722157 0.856862 0.935541];
130     z2=[0.8998 0.0723 0.0692 0.0010];
131
132     figure;
133     plot(x2,y2,'-*','Linewidth',2);
134     title('CPU time','FontSize');
135     xlabel('Number of POD modes');
136     ylabel('CPU time (sec)');
137     set(gca,'FontSize',12,'fontweight','b','fontname','arial')
138         set(gcf, 'Color', 'w');
139     figure;
```

```
140 plot(x2,z2,'-*','Linewidth',2);
141 title('Relative average error');
142 xlabel('Number of POD modes');
143 ylabel('Relative average error');
144 set(gca,'FontSize',12,'fontweight','b','fontname','arial')
145     set(gcf, 'Color', 'w');
```

ADEpde.m

```
1 function [c f s]=ADEpde(x,t,u,DuDx)
2 c=1;
3 f=0.01*DuDx;
4 s=-1*DuDx;
```

ADEpdeic.m

```
1 function u0=ADEpdeic(x)
2 u0=2*sech(x);
```

ADEpdebc.m

```
1 function [pl,ql,pr,qr]=ADEpdebc(xl,ul,xr,ur,t)
2 pl=ul;
3 ql=0;
4 pr=ur;
5 qr=0;
```

ADE_POD.m

```
1 function rhs=ADE_POD(t,a,dummy,phi,phix,L)
2
3 rhs=L*a-phi'*(phix*a);
```